Does the Online Card Payment System Unwittingly Facilitate Fraud?

Mohammed Aamir Ali

School of Computing Science, Newcastle University

Newcastle Upon Tyne, UK

April 2018

This thesis is submitted for the degree of

Doctor of Philosophy

ABSTRACT

The research work in this PhD thesis presents an extensive investigation into the security settings of Card Not Present (CNP) financial transactions. These are the transactions which include payments performed with a card over the Internet on the websites, and over the phone. Our detailed analysis on hundreds of websites and on multiple CNP payment protocols justifies that the current security architecture of CNP payment system is not adequate enough to protect itself from fraud. Unintentionally, the payment system itself will allow an adversary to learn and exploit almost all of the security features put in place to protect the CNP payment system from fraud. With insecure modes of accepting payments, the online payment system paves the way for cybercriminals to abuse even the latest designed payment protocols like 3D Secure 2.0.

We follow a structured analysis methodology which identifies vulnerabilities in the CNP payment protocols and demonstrates the impact of these vulnerabilities on the overall payment system. The analysis methodology comprises of UML diagrams and reference tables which describe the CNP payment protocol sequences, software tools which implements the protocol and practical demonstrations of the research results. Detailed referencing of the online payment specifications provides a documented link between the exploitable vulnerabilities observed in real implementations and the source of the vulnerability in the payment specifications.

We use practical demonstrations to show that these vulnerabilities can be exploited in the real-world with ease. This presents a stronger impact message when presenting our research results to a non-technical audience. This has helped to raise awareness of security issues relating to payment cards, with our work appearing in the media, radio and TV.

Acknowledgements

Undertaking this PhD has been a truly life-changing experience for me and it would not have been possible to do without the support and guidance that I received from many people.

Special mention goes to my supervisor, Aad van Moorsel. My PhD has been an amazing experience and I thank Aad wholeheartedly, not only for his tremendous academic support, but also for giving me so many wonderful opportunities throughout. I mean, not many PhDs involves meetings with leading corporates or lavish travels to St. Kitts, NY and many other places to get to a conference.

Similar, profound gratitude goes to Martin Emms, who has been a truly dedicated technical support. I am particularly indebted to Martin for his continuous support in my lab work. Martin's guidance in the area of my research has been invaluable and he is above all a good friend.

I am also hugely appreciative to Leo Freitas for sharing his expertise on payments so willingly, Budi Arief and Maryam Mehernezad for collaborative work on several research projects, Uchechi Nwadike and Richie Jenkins for continually encouraging me to embark on payments research path, and for providing me with friendly support when needed.

I am greatly indebted to Sara Nurain for all good she has done for me and I dedicate this thesis to her. Life without you Sara will always be incomplete.

Last but not the least, I would like to thank my family: my mother and to my brother and sisters for always believing in me, encouraging me to follow my dreams, and helping in whatever way they could during this challenging period.

Table of Contents

List of Tables

List of Figures

Glossary of Terms

AC	Application Cryptogram – the EMV protocol utilises 3-DES encoded cryptograms as a secure method of communication between the EMV payment card and the Issuing Bank. There are four different types of cryptogram; TC (transaction approved), AAC (transaction declined), ARPC (request online approval from the Issuer) an ARPC (Issuer authorisation decision).
AAC	Application Authentication Cryptogram – is the application cryptogram generated by an EMV card indicating that the transaction has been cancelled.
ARPC	Authorisation Response Cryptogram – in the protocol sequence for transactions that require online authorisation, the card generates an Authorisation Request Cryptogram to signify that it wishes to complete the transaction online. The Issuer responds with and ARPC, which encodes the Issuers authorisation response to the transaction request. The ARPC is 3-DES or AES encrypted using the Issuers private key which allows the card to ensure that the card to validate that the response came from the Issuer and has not been altered.
Payment System	A payment system is an inter-network of globally connected systems that facilitate the settlement of the financial transaction.
Tokenisation	Tokenisation is a process in which sensitive data is transferred between parties in reordered strings of number or tokens. Tokenisation uses mathematical formulas and random number generators that creates characters of no value to an attacker.
Payment Card Tokenisation	When the process of Tokenisation is applied on a payment card, it is then known as payment card tokenisation. When payment card tokenisation is implemented, the actual card numbers are replaced with random numbers which masks the actual card numbers.
Firewall	Firewall is a software or a hardware component that acts as a defence layer implemented to secure the system against known cyber-attacks.

PAN	Primary Account Number (PAN). Refers to 16-digit payment card number that links the card to the customer account.
Cardholder/Customer	The cardholder/customer is an authorised person entitled by the card issuing bank as the owner of the card
Acquirer	The payment acquirer maintains contractual relationships with the merchants and provides the merchant with a trading account to collect the cardholder transactions.
Payment Card	Token provided by the financial institution (bank for example) to the cardholder/customer that can be used by facilitate payments.
Account Number	The account number is an 8-digit code that links to the cardholder account and is used during check payments and internet money transfers.
Sort Code	The sort-code is a 6-digit numeric code that identifies the bank and the branch where the cardholder account is held.
Magnetic Stripe	Magnetic Stripe is made up of tiny iron-based magnetic particles in plastic like film. The purpose of a magnetic stripe is store payment application data and allow the cardholder to swipe the card through the merchant POS terminal.
CVV	Cardholder Verification Value (CVV) is a three-digit value generated by the card issuing banks and is embedded in card data before the card is issued to the cardholder.
CVV2	Cardholder Verification Value-2 is a three or four-digit code printed on the front or at the back of the card.
Cardholder Data	Includes user payment card data including any authentication information which can identify a cardholder in the payment system.
Cryptography	Cryptography is a science of securing data or communication in the presence of adversaries.
PCI	Payment Card Industry – A consortium of five card payment networks American Express, Discover Financial Services, JCB International, MasterCard, and Visa Inc. that works towards defining, maintaining and evolving payment card standards for global interoperability.
PCI DSS	Payment Card Industry Data Security Standard (PCI DSS). A framework defined by PCI for secure handling and storage of payment card data.
Encryption	A cryptographic technique of obscuring information to make it unreadable.
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure

Card Issuer	Card Issuer or simply an issuer is a financial body that issues a payment card and maintains contractual relationship with the cardholder
MO/TO	Mail Order/Telephone Order
Payment Processor(s)	Entities that process user payment card details. Includes but are not limited to: merchants, payment acquirers, card payment networks, card issuing banks and any intermediate service providers.
Protocol	Set of rules and procedures with which the computing systems communicate and operate.
PoS terminal/Card reader	Point of Sale terminal. Provided by the merchant where the cardholder inserts their payment card to initiate payments.
PIN	Personal Identification Number
Smart Card	A token that has a capability to secure store and process the data.
SSL	Secure Socket Layer
TLS	Transport Layer Security
EMVCo	Europay MasterCard Visa
RFID	Radio Frequency Identification
SDA	Static Data Authentication
DDA	Dynamic Data Authentication
CDA	Combined Data Authentication
PKI	Public Key Infrastructure
3-DES	Triple DES – is the common name for Triple Data Encryption Algorithm which applies the Data Encryption Standard (DES) three times to each data block. 3-DES is a 64-bit block-cipher
AC	Application Cryptogram – the EMV protocol utilises 3-DES encoded cryptograms as a secure method of communication between the EMV payment card and the Issuing Bank. There are four different types of cryptogram; TC (transaction approved), AAC (transaction declined), ARPC (request online approval from the Issuer) an ARPC (Issuer authorisation decision).
Acquirer	Refers to the bank that holds the destination bank account for the transaction, which is typically the bank that issued the POS terminal to the merchant. Also referred to as the “acquiring bank”.
ARPC	Authorisation Response Cryptogram – in the protocol sequence for transactions that require online authorisation, the card generates an Authorisation Request Cryptogram to signify that it wishes to complete the transaction online. The Issuer responds with and ARPC, which encodes the

	Issuers authorisation response to the transaction request. The ARPC is 3-DES or AES encrypted using the Issuers private key which allows the card to ensure that the card to validate that the response came from the Issuer and has not been altered.
ARQC	Authorisation Request Cryptogram – is generated by the card to indicate that it wants to complete the transaction online (see ARPC).
ATM	Automatic Teller Machine – commonly termed cash machines, they allow bank customers to withdraw cash from their bank account.
CA	Certificate Authority – in EMV each of the card scheme providers Visa, MasterCard, American Express, JCB, Discover and Union Pay act as the Certificate authority for their own branded cards. The CAs generate the Issuer's RSA private keys thereby allowing the Issuers to generate RSA public private key pairs for their cards. ATMs and POS terminals can validate the cards RSA signature using the CAs public key.
Cardholder	This is a generic term for the person with an EMV credit or debit card who is making a payment in a shop / restaurant or is withdrawing cash from an ATM. The term also implies that the person has a bank account to which the card is attached.
Chip & PIN	The common name used in the UK to refer to EMV payment system. The EMV specifications defines the operation of the payment cards, POS terminals and ATMs.
EMV	Europay MasterCard Visa is a global standard for card payments (commonly termed "Chip & PIN"). The standard ensures the interoperability of EMV payment cards, ATMs and POS terminals across different banks and in different countries.

Chapter 1. Introduction

The Internet is no doubt a historic evolution in communication for the human race. Free from limits of time, distance and territories, the Internet today has reshaped our society's behaviour in a way that many aspects of our lives are now connected online. This includes electronic commerce (e-commerce) where we buy or sell goods online further facilitated by various internet merchants like Amazon, Apple, PayPal to name a few.

The convenience of making purchases suing online payment makes it even more attractive for customers experience. In 2016, a total of 1.8 billion transactions were made which amounted to a total of £154 billion by UK based merchants. This is an 18% increase in the online spending patterns recorded in 2014 where 1.3 billion transactions were recorded totalling to an amount of £120 billion. All this convenience of making payments over the Internet is possible because of networks provided by payment processors and a piece of plastic – a payment card: which is now common-practice and the most widely accepted form of payment by online merchants.

One should consider that an element of fraud exists in systems where the money flows. Payment fraud over e-commerce is a $20bn business worldwide [1]. In 2017, e-commerce payment fraud has surpassed the thresholds fraud rate recorded in any of the previous years amounting to a total of £432, for cards issued in the UK [2]. The payment industry is effective in mitigating other categories of payment card frauds through the introduction of more secure technologies. However, the statistics [2] suggests that, for e-commerce payments, the cybercriminals are well ahead of the intelligence of payment system security designers. Cybercriminals have also adapted to the changing and ever evolving payment systems and invest their time to identify the weakest links in order to exploit the system.

There are several challenges[1] identified by the payment industry which need to be considered while designing an online payment system (we will discuss these challenges in Chapter 3). However, superseding every other challenge is the business need to provide convenience to the customer at the checkout while making an online card payment.

This has resulted in the online payment system being loosely regulated with its security implementations left mostly upon the online merchants. As online merchants work more on attracting customers by offering more choices and convenience to make payments, it may impact the security of online payment system, benefitting cybercriminals to exploit weaknesses and practice fraudulent purchases. Choices of implementation options have also resulted in the modes of payment being inconsistent from one web merchant to another. Several researchers [3][4] claim that the online payment system is badly designed for the customers and even recently Sir Tim Berners Lee stressed that we are way behind in designing a conventional method of making card payments on the Internet [5].

Technically, in the payment system terms, e-commerce payments are referred to as Card-Not-Present (CNP) payments and the transactions performed in physical stores are referred to as Card Present (CP) payment systems. This distinction becomes much clearer in Chapter 2 where we expand our knowledge of different components of the card payment system. In what follows next is a description of our aims and objectives with this PhD.

1.1 Aims and Objectives

With this research work, we primarily aim to answer the following question: *"Does the philosophy of providing excessive convenience to the customer at the checkout have any effect on the security of the CNP payment system?".* In particular, we aim to explore the security settings of each method that is presently in use by the payment industry stakeholders to accept CNP payments. Our objectives are to explore the following questions:

- What are the root causes and consequences of increasing online CNP payment fraud?
- How securely does each party in the CNP payment eco-system process and handle the payment card data?
- What is the minimum card data required by the card issuing banks to process a CNP payment?
- How does the card issuing bank establishes the identity of a cardholder making a CNP payment (how secure is user-authentication in the CNP payment system), and

[1] Challenges includes customer authentication, payment credential confidentiality, payment data integrity, customer non-repudiation explained in Chapter 2.

- How do cybercriminals obtain the payment card and user authentication information required to make payments even in the most secure modes of CNP payment systems?

To answer these questions, we have followed a structured analysis of payment protocols which are currently in use by the payment industry to accept CNP payments online.

At this point, it is important to mention that the research work done in this PhD and the experiments we performed are from a standard customer accessible platform on online merchant websites (the checkout page from where payments are made) and without any escalated privileges from payment processors[2]. This, to a substantial extent, has demonstrated that the vulnerabilities that our research has discovered can be exploited effortlessly.

For many reasons research into the security of card-not-present is projected as a business activity. Therefore, the research work in this PhD sets out to bridge the gap in between the academia and the business worlds and provide a mixed-method approach to an investigation into the security of CNP payment systems.

1.2 Contributions of the Thesis

The research work carried out in this PhD represents a significant contribution to the subject of CNP payment security research in the following ways:

- Reviewing and updating already insufficient academic research on the security analysis of CNP payment systems.
- Development of structured architecture framework which serves as a blueprint for further research into the security analysis of CNP payment system.
- Critical evaluation and comparison of strengths and weaknesses of existing CNP payment protocols.
- Implementation of a merchant platform which can link to multiple payment acquirers (merchant payment service providers) and assess the security provided by the payment acquirers.
- Implementation of automated desktop tools and Android-based contactless card reader software which are programmed to perform security assessments over online merchant websites and contactless payment cards.
- First public description of 3D Secure 2.0 frictionless authentication protocol in use by card-issuing banks to provide more security to the online CNP payment system.

[2] Payment processors includes card issuing banks, payment networks, payment acquirers and online merchants.

Following our methodology, we have identified several previously undocumented vulnerabilities in the online CNP payment system. Combining these vulnerabilities we empirically demonstrate the workings of at least three attack scenarios on the online CNP payment system: *Distributed guessing attack, cardholder impersonation attack and betrayal attack.*

- Our structured responsible disclosure exercise assisted online merchants, payment acquirers, card payment networks and card-issuing banks in identifying and patching the vulnerabilities that we discovered during our analysis on the CNP payment system.
- Passing our research finding to the general public and more importantly educating them in how to securely handle their payment card details online.

1.3 Collaborations and Publications

Chapters 6, 7 and 8 in this PhD thesis reflect research papers that were published as conference papers, journal articles and as technical reports. A list of our publication with co-authors and acknowledgements is listed below:

- Ali, M.A., Arief, B., Emms, M. and van Moorsel, A., 2017. "Does the Online Card Payment Landscape Unwittingly Facilitate Fraud?" In IEEE Security & Privacy, 15(2), pp.78-86. [6]
- Mehrnezhad, M., Ali, M. A., Hao, F., & van Moorsel, A. (2016, December). "NFC payment spy: a privacy attack on contactless payments." In *International Conference on Research in Security Standardisation*, (pp. 92-111). Springer . [7]
- Ali, M.A., Emms, M., Arief, B. and van Moorsel, A, (2015, July). "Extracting Credit Card Details from the Online Payment System. (TR 1475)," Newcastle University – School of Computing Science *Technical Report Series*, July 2015.
  Ali, M.A. and van Moorsel, A., 2018 "Designed to be Broken: Security Analysis of 3D Secure 2.0 Frictionless Authentication," in *23rd International Conference on Financial Cryptography and Data Security (FC 2019)*, St. Kitts, 2019.
  Ali, M.A., Centeno, M.P, Azad, M.A., Hao, F. and van Moorsel, A., 2018, Consumer Facing Technology Fraud: Economics, Attack Methods and Potential Solutions" in *Special Issue on Economic Aspects of Cybersecurity and Privay, Future Genration of Computer Systems, Elsevier*.

The security of CNP payment system is not only crucial for payment processors but also has an impact on the lives of customers who are the primary driving factors for the success of CNP payment systems. Therefore, with an intent to educate the current security posture of CNP payment systems, we publicise our findings in a mode accessible to general public. We have not only spoken during several public visiting events at the university but also used the media to carry this message of secure handling of their payment credentials while making purchases online. Following are some of the media articles and blog posts by renowned researchers appraising our research work:

- BBC Radio 5, “Six seconds – all it takes a fraudster to get your Visa details,” 2 December 2016
- BBC Radio Newcastle, “Alfie and Anna at Breakfast – Visa card hack,” 2 December 2016
- BBC UK, “Frighteningly easy' for criminals to get Visa card details, study claims”, 2 December 2016 [8]
- The Telegraph UK, “Hacked in just six seconds: How criminals only need moments to guess card number and security code,” 2 December 2016 [9]
- msn.com, “Six seconds – all it takes a fraudster to get your Visa details,” 2 December 2016 [10]
- Canada TV (ctvnews.ca), “How hackers can guess your credit card information in just 6 seconds,” 2 December 2016 [11]
- BBC Radio 4, “Criminals take just six seconds to guess Visa card number and code, experts find,” 2 December 2016
- The Sun, “Gone in six second – Internet crooks can now hack your credit card details in just seconds”, 2 December 2016 [12]
- The Times, “Fraudsters take six seconds to steal bank card details” 2 December 2016 [13]
- AOL, “Criminals take just six seconds to guess Visa card number and code, experts find,” 2 December 2016 [14]
- The Guardian, “Tesco Bank cyber-attack involved simply guessing details, a study claims”, 2 December 2016 [15]
- ITV Lunchtime News, “Hacking takes seconds, say experts at Newcastle University,” 2 December 2016 [16]
- **Schneier on Security**, “Guessing credit card security details,” 5 December 2016 [17]
- **AT&T ThreatTraq** #225, “Distributed Guessing Attack,” 14 December 2016 [18]

The importance of a part of our work was recognised when the cyber-criminals exploited distributed guessing attack, managing to steal at least £2.5 million by abusing at least 9000 customer accounts belonging to Tesco Bank. We reported the distributed guessing attack to the payment processors (including Visa payment network) at least 10 months before the attack was exploited by the fraudsters. This to an extent demonstrates the practical impact of our research work.

- SCMagazine UK, “Tesco Bank allegedly ignored warnings of hack from Visa, ” 29 November 2016 [19]
- The Times, “Tesco Bank failed to heed the warning on cyberattack,” 28 November 2016 [20]

1.4 Presentations

During the course of my Ph.D., I was privileged to be invited as a guest speaker at following research institutes and payment security conferences:

- "Trends in Online Payment Security." Computer Laboratory Security Talks, **University of Cambridge**, UK. Jan-2017.
- "Does the Online Card Payment System Unwittingly Facilitate Fraud?" **University of Kent**, UK. Jan-2017.
- "We Have Your Payment Card Details." Research Exhibition, **Home Office**, London, UK. June 2016.
- "Trends in Payment Security (Masterclass)." **CESG CyberUK in Practice**, Liverpool, UK. May 2015.
- "Trends in Card Payment Security." **Indian Institute of Technology (IIT)**, Kanpur, India. April-2017.
- "Exploiting Vulnerabilities in the Online Payment System." **Smart Payment Conference**, Frankfurt, Germany. Sept-2017.
- "Six Seconds to Hack a Credit Card." FINSEC - **The Banking Security Summit**, Dubai, UAE. Feb-2017.

1.5 Structure of Thesis

Chapter 2 presents an introduction to the context of global card payment system. We highlight the security features of a payment card and introduce the reader to various ways in which a payment can be performed using the payment card. We then categorise the card payment systems in two types: Card Present (CP) and Card Not Present (CNP). Furthermore, we provide a detailed description of protocols involved in CP transactions and perform a comparative analysis of their security features and limitations.

Chapter 3 presents literature review which is relevant to the research performed on the card payment systems. It will focus on the vulnerabilities that were explored and practically proven to exist in the payment systems by the researchers.

Chapter 4 presents our analysis methodology following which we have achieved identifying certain vulnerabilities in the online CNP payment system. It draws best practices from cybersecurity literature and assembles them into a structured blueprint which can be applied by current and future researchers interested in the analysis of CNP payment security.

Chapter 5 comprehensively examines protocols involved in the CNP payment transaction. It will broadly classify CNP payment protocols into two categories: authorisation-only and authentication-enabled CNP protocols and will further detail the working of available

protocols within each category. It will conclude by assessing the security strengths and weaknesses associated with each protocol in the CNP payment system. The conclusions from this chapter are foundations for establishing research questions that we aim to answer in this PhD.

Chapter 6 demonstrates a distributed guessing attack on the authorisation-only online payment system. The attack effectively turns the process that is meant to validate the card payment details into a process which delivers to the attacker all card data required to make an online payment. It will also detail the vulnerability disclosure exercise that we performed on the affected parties in the online CNP payment system. Finally, it will conclude by suggesting countermeasures to mitigate the distributed guessing attack.

In the next two chapters, we focus our study on the security of authentication-enabled CNP payment protocols.

Chapter 7 describes our work in applying reverse engineering techniques to study certain obscure components of 3DS 2.0 authentication-enabled CNP payment protocol. It will provide the first public description of components of 3DS 2.0 protocol which enable the card issuers to efficiently accept online CNP payments without requesting any passcodes from the customer. It will also address the vulnerabilities or weaknesses that were identified in the 3DS 2.0 protocol.

Chapter 8 will introduce two attack scenarios on the 3DS 2.0 payment protocol. Firstly, it will demonstrate the cardholder impersonation attack which allows an attacker to use stolen cardholder details by circumventing the security features of 3DS 2.0 in place to protect cardholder payments from fraud. Secondly, it describes a betrayal attack on 3DS 2.0 protocol which will enable a 'trusted' adversary to perform transactions using any random card details. The chapter will conclude with a discussion on the challenges of solving the vulnerabilities and suggesting some countermeasures that can be adopted to mitigate exploited attacks.

Chapter 9 concludes this PhD thesis with a summary of the contributions that the research work has provided to the community. It evaluates the results that we achieved against our research question: "*Does the philosophy of providing excessive convenience to the customer at the checkout has (have) any effect on the security of the CNP payment system?*". Finally, it will suggest the directions for future explorations in the area of payment systems research.

Chapter 2. Card Payment Systems

This chapter starts with a brief introduction to the parties and processes involved in the card payment process. It will look at how the payment card evolved from a simple paper to include much-advanced processors embedded in simple plastics. Next, the chapter defines payment transaction types supported by the card, outlines the messaging standards and protocols used across the parties in the payment system.

Once the context of the system is established, this chapter continues with an overview of payment card fraud which evolved over the years. We will look at the techniques adopted by fraudsters and the chain under which these cybercriminals operate. Finally, the chapter concludes with a discussion on the insights that were obtained from payment fraud trends which helped me to select a topic of interest for the research.

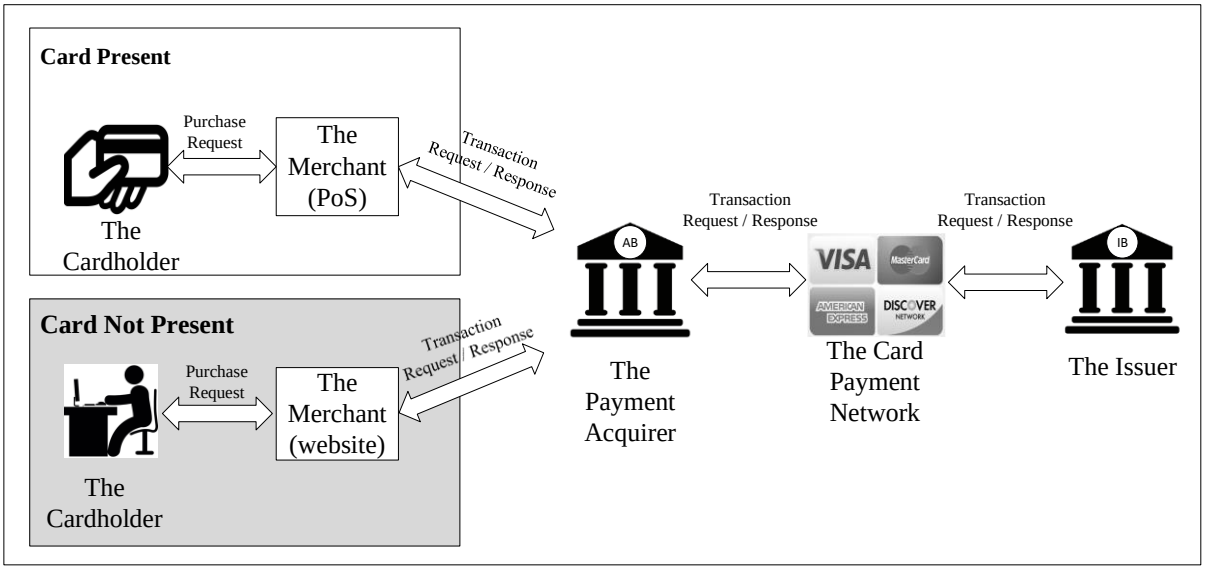

Figure 1 – Parties and process involved in the card payment transaction.

2.1 The Card Payment System

The card payment system uses a buyer's existing credit/debit card to transfer funds from the buyer's bank account into merchant's bank account. To understand how the card payment system operates, let

us start with a basic overview of the system. Figure 1 shows the parties and the processes involved in the card payment transaction.

The cardholder initiates the transaction process by presenting his card to the merchant's payment acquiring devices such as a Point of Sale (PoS) terminal / card reader or a website. In the next step, the merchant submits a payment request to the cardholder's card issuing bank. The payment request to the card issuing bank travels through a series of networks provided by the acquirer (the merchant payment processor) and the payment card network (Visa, Mastercard, American Express etc). The card issuing bank verifies the cardholder's card details and account balances and responds back to the merchant with the result of the transaction (approved or declined), through the same route as of transaction request. Below is the description of parties involved in a card payment transaction:

The cardholder/customer is an authorised person entitled by the card issuing bank as the owner of the card. The cardholder purchases goods either through a card-present, or a card-not-present transaction. For the card-present transactions, the cardholder inserts their card into the PoS terminal and completes a transaction by entering the PIN or by signing the merchant receipt. For the Card-Not-Present (CNP) transactions, the cardholder enters their payment card details on merchant websites.

Within the CNP payment system, there are Mail Order (MO) and Telephone Order (TO) transactions. For MO transactions, the card details are provided in the email to the merchants, and for Telephone Order (TO) type transactions, the cardholders provide their card details verbally to the merchant over the phone.

The Merchant is an organisation accepting payments, usually in exchange for goods or services. The merchant provides the cardholder with an interface which can be used by the cardholder to initiate a payment.

The Issuer is the bank or an organisation that issues the payment card to the cardholder. The payment card provided by the issuer can be used as a token to initiate payments from the merchant's provided payment interface. The issuer holds the cardholder bank account and has final approval of the payment. The issuer has access to cardholder information such as account balance, name, full address, any other personally identifiable information belonging to the cardholder, which is not visible to the rest of the payment network.

The Payment Acquirer facilitates the communication between the merchants and the card payment networks. The payment acquirer maintains contractual relationships with the merchants and provides the merchant with a trading account to collect the cardholder transactions. The payment acquirer offers a range of services to the merchants which includes:

- Providing merchant with a trading bank account

- maintaining the merchant compliant programs, Payment Card Industry Data Security Standard (PCI DSS) [21], General Data Protection Regulation (GDPR) [22] as an example
- producing a line of defence against fraud
- in some cases, accepting liability for fraudulent transactions.

In payment industry terms, the acquiring bank is also known as merchant's payment processor, the payment gateway or sometimes as merchant's acquiring bank or simply an acquirer.

The Card Payment Network manages the network that connects thousands of payments acquiring banks to the card issuing banks worldwide. Visa, MasterCard and American Express (Amex) are some examples of the most heard card payment networks. Some card payment networks like Amex, Discover, Japan Card Brand (JCB) issue the cards to the cardholders and thus verifies the payment requests as well. Such type of a payment network is said to be operating in a closed-loop network. Visa and MasterCard, on the other hand, support open-loop network. Card payment networks are sometimes referred to as Card Brands.

Figure 2 - ChargeIt credit card

Service Providers. There can be additional business entities involved in the processing of a transaction (not shown in Figure 1). For example, certificate authorities, tokenisation servers, managed firewalls or card data storage servers. Payment industries generally categorise them as service providers.

2.2 Evolution of Payment Cards

It all started in the year 1940's where the first bank card "Charg-It" was introduced locally by Baggins bank [23]. Figure 2 illustrates the physical properties of a Charg-It card which shows the card as a metal token with account number embossed on the front. When the customers used Charg-It for purchases, the merchants accepting the card forwarded the customer bill to the Baggins bank which reimbursed the money to the merchants. Customers were charged for the purchases on a monthly basis.

Later in 1950, the Diner club credit card was introduced [23] with advanced physical properties of a payment card. The Diner club card was more a cardboard piece, with account numbers embossed on the front. A decade later, the cardboard was replaced with a plastic card.

Around the same time in 1958, BankofAmerica introduced their first credit card [23][24]. Earlier cards than BankofAmerica were regional specific, operated locally and only with certain merchants. However, BankofAmerica expanded their network and began issuing the card to multiple banks across the country. This created a chain of the network that connected banks to banks. Later in 1970,

Figure 3 – Magnetic stripe card prototype proposed by IBM

BankofAmerica had a spin-off which later became Visa Inc as we know it. MasterCard came along the same lines in 1966 as the Interbank Charge Association [23].

To accept payments using cards, merchants began using a card-imprinter device "Zap-Zap machine" which took an impression of raised card numbers on payment cards The impression of card numbers was taken over two pressure sensitive carbon papers/bills. One of the bill was handed to the card issuing bank where a person would type the transaction information into their payment processing systems. This system took days, was insecure and was prone to errors. All these limiatations paved the way for Magnetic stripe technology that allowed magnets to store and recover the information onto a tape or stripe.

Figure 3 shows the first magnetic stripe card prototype unveiled by IBM in 1970 [24]. The cardholder was required to swipe the stripe cards across a magnetic reader provided by the merchant. The payment industry was slow in the adoption of magnetic stripe card technology. However, further enhancements to the magnetic stripe interface, low production cost and durability of magnetic stripe cards over cardboard cards made them more palatable to the financial industry, and magnetic stripe cards started to become a universal media for making payments. It was around the same time when ATM machines were introduced. In 1967, Barclays became one of the earliest banks to install ATM machines across the streets for public use in the UK [23].

Since the universal adoption magnetic stripe cards in the 1980s, the payment card mostly remained unchanged until in 1990s when cards with embedded smart chips were introduced [24]. With cryptographic processing intelligence and secure storage technology, these smart cards offered robust electronic media to host banking applications and secure communication standards. In 1994 – Europay, Mastercard and Visa (EMV) [25], a consortium of card payment networks, came up with an EMV protocol [26]: a secure communication protocol for payment applications using smart cards.

Around early 90's the Internet was released for public use. Merchants saw business potential in the technology, and soon webstores began to appear on the Internet [23]. Musical records and Pizza were few of earliest made orders on the Internet using a payment card [27].

In 2003, the UK payment industry announced the replacement of magnetic stripe cards with EMV enabled smart payment cards. Smart cards with EMV protocol require the cardholder to enter a four-

Figure 4 - Visible Elements on the Payment Card

digit Personal Identification Number (PIN) on the PoS terminal provided by the merchant to complete the payment: this is the reason why EMV is also referred as 'Chip and PIN'. The PIN is given to the cardholder by the card issuing bank.

Another technological evolution in the payment card in the form of support for making touchless (contactless) payments was introduced in 2003. Unlike earlier smart cards which require a point of contact for communication with the reader, contactless payment cards exchange messages with the reader terminal wirelessly using RFID technology [28]. Contactless payments are designed for low-cost in-store payments usually about £30 in the UK and do not require PIN verification for authentication. The five-year period from 2003-2008 saw rapid development in the payment card, especially RFID technologies. The ease with an RFID component can be build made it possible to have a payment application on wristbands and paper stickers. Sometimes contactless payments are also called proximity payments because the card is required to be in very close proximity (around 15 cm) to the reader [29].

All these technological evolutions to a payment card brought it to what we see it today (shown in Figure 4). It is to be noted that one of the primary features of the payment card has never evolved and is still in operation today i.e., the presence of card number and expiry date embossed or printed on the front of the card.

2.3 Payment Card Overview

A payment card is a token provided by the card issuing bank to the cardholder which is used to initiate payments. A payment card contains data in several places and provides support for multiple payment interfaces. Let us start by inspecting the visual elements of a payment card. Figure 4 shows the front and back of a typical payment card.

Visual elements on the front side of a payment card include:

1. **Card issuer name.** The name of the card issuer that issues the payment card to the cardholder.

2. **Chip/Secure Element.** The chip is an electronic medium which hosts the card issuing bank's payment application. With cryptographic processing intelligence and secure storage technology these chips provide robust electronic media for EMV [25] to host and run banking applications and communication standards.
3. **16-digit card number.** Also called the Primary Account Number (PAN), this field is a unique 16-digit payment card identifier assigned by the card issuing bank. The PAN links the card to the cardholder's bank account and is usually printed or embossed on the front of the card. There can exist multiple PANs belonging to a single cardholder account. The first six digits of a PAN collectively is called as the Bank Identification Number (BIN) or the Issuer Identification Number (IIN). The BIN identifies the type and brand of a payment card.
4. **Card expiry date.** The card expiry date denotes the date beyond which the card is rendered unusable. Some cards also have a starting date which indicates when the card is valid from.
5. **Cardholder name** is printed or embossed on the front card. This information is stored by the card issuing bank when the cardholder opens their bank account.
6. **Account number and bank sort code.** This account number should not be confused with the PAN. The account number is an 8-digit code that links to the cardholder's account and is used during check/paper payments and the Internet money transfers. The sort-code is a 6-digit numeric code that identifies the bank and the branch where the cardholder account is held.
7. **Payment card network/Payment brand.** Identifies the network under which the payment card operates. There are five internationally recognised payment card bands. They are MasterCard, Visa, American Express, JCB and Discover. There can by many more card brands as it is also country specific. This field also denotes the type of the card among credit and debit card.
8. **Support for contactless.** The logo indicates that the card supports contactless interface.

Visual elements on the back of a payment card include:

9. **Magnetic stripe.** The magnetic stripe is made up of tiny iron-based magnetic particles in plastic like film. The purpose of a magnetic stripe is to store payment application data and allow the cardholder to swipe the card through the merchant PoS terminal.
10. **Signature stripe:** Each card also has a signature panel, and this is used to verify the identity of the cardholder making in-store purchases. During a transaction, the merchant requests the cardholder to sign the merchant receipt. The signature on the card is then compared with the signature on the merchant receipt. The transaction is only accepted in case of a signature match.
11. **Card Verification Value (CVV2):** CVV2 is a three or four-digit code printed on the front or at the back of the card. The purpose of CVV2 is to identify the cardholder when the

transaction is made either online or over the phone. The CVV2 is almost always requested on every transaction made online, and it drastically reduces the number of fraudulent transactions. American Express prints a four-digit Card Identification Code (CID) on the front of the card.

Many cards also have various other security features, such as holograms, on either the front or back of the card. Also, one might not be able to see it, but every card that supports contactless payments has a Radio Frequency Identification (RFID) antenna.

2.4 Payment Card Technologies

In this section, we will review the technology behind three interfaces available in a modern payment card: magnetic stripe, chip cards and EMV, and contactless cards and EMV.

2.4.1 Magnetic Stripe Technology

Magnetic stripe has a capability to store information which can be read electronically by magnetic stripe reader head. Each magnetic stripe payment card comes with a pre-loaded payment application that contains information about the user payment account as embedded by the card issuing bank.

There are two tracks containing payment data located within the magnetic stripe – Track 1 and Track 2. Track 1 is the longer track, up to 79 alphanumeric characters, where Track 2 is the shorter, up to 42 numeric characters and mainly used for the older dial-up transmissions. Occasionally, the high-coercivity magnetic stripe cards contain an additional track which can hold up to 107 numeric characters.

Track 1 includes all fields of Track 2 plus the cardholder's name and additional fields for exclusive use by the card issuer. Figure 5 illustrates the structure and contents of Track 1 inside magnetic stripe. Mainly we have a field for PAN, cardholder name, card's expiry date, a service code which specifies the interchange rules and controls risk management functions and the final field is discretionary data which is used to provide security functions to a magnetic stripe transaction. Discretionary data includes one or more of the following fields: PIN Verification Key Indicator (PKVI) [30] , PIN Verification Value (PVV) [30], Card Verification Value (CVV) [31] and Card Validation Code (CVC) [31].

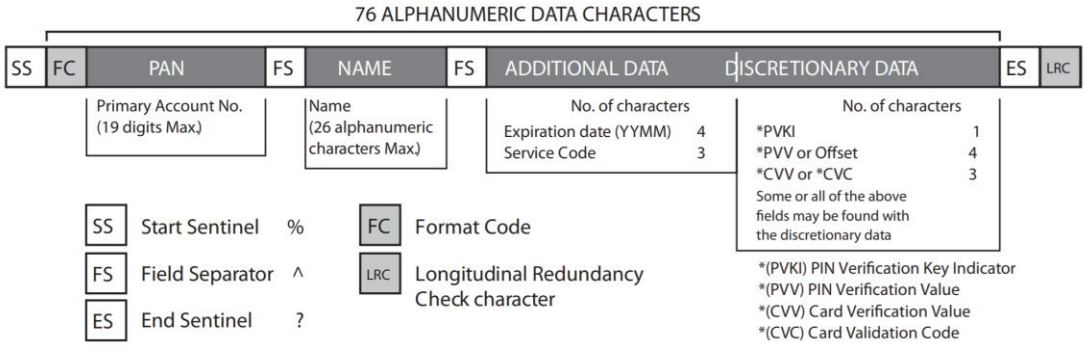

Figure 5 - Structure and contents of a Track 1 magstripe

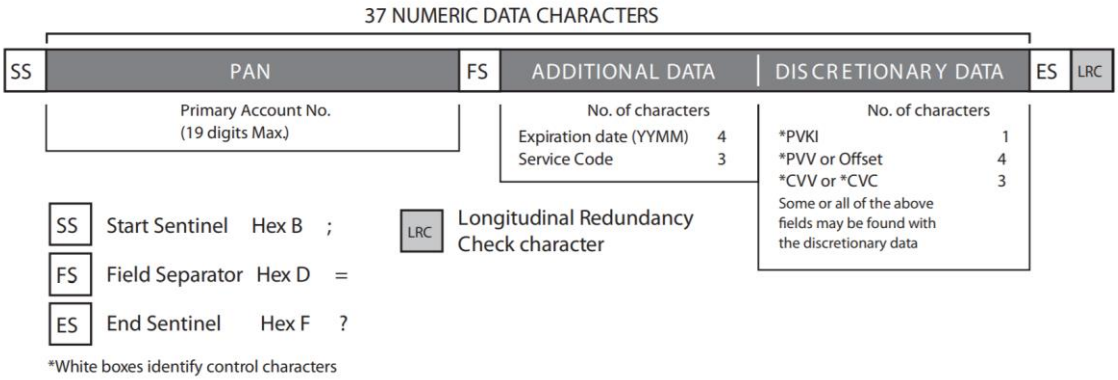

Figure 6 - Structure and contents of a Track 2 magstripe

With each magnetic stripe read, the card provides stored payment application data which is fetched by the PoS terminal to process a transaction. The terminal then applies security protocols for cardholder verification and risk management to complete a transaction.

2.4.2 Magnetic Stripe Card Security

Each magnetic stripe card has a cryptographically derived Card Verification Value (CVV) which makes magnetic stripe cards much secure than just having the PAN and expiry date [32]. CVV prevents counterfeit cards from being generated by using the cardholder data obtained from paper receipts. CVV is a three-digit value generated by the card issuing banks and is embedded in card data before the card is issued to the cardholder. It is calculated by encrypting the PAN and expiry date with a symmetric key only known to the card issuer. The output cryptogram from the encryption is passed through a conversion function which reduces the cryptogram size to a three-digit numeric value. CVV is kept to three-digit because of the storage restrictions in the magnetic stripe. Every time, the payment data is received in the authorisation request, the card issuer extracts the PAN and expiry date and generates a local copy of CVV using the same symmetric key. If the CVV received during authorisation request matches to the locally generated CVV, the card issuer ensures that the transaction data is coming from the real-card issued to the cardholder.

Figure 7 – Skimming devices found attached to ATM machines

2.4.2.1 PIN Validation

One of the most common methods used by the card issuing banks to generate and validate the card PIN is by using VISA PIN Verification Value (PVV) algorithm [30]. The PVV is a four-digit cryptographic signature encoded on the payment card by the card issuing banks. The PVV is derived by encrypting together three elements (i) payment card number (ii) a key identifier and (iii) a four-digit PIN, with a PVV generation key is only known to the card issuer.

When the cardholder enters the PIN, the PoS terminal combines the transaction data, the card data including PVV and the cardholder entered PIN; encrypts it and forwards it to the card issuing bank as an authorisation request. To validate the PIN, the card issuing bank either retrieves a local copy of PVV stored in their database or generates a PVV from the PIN and PAN received in the authorisation request. The generated PVV is compared with a reference PVV received in the authorisation request. A match in the PVV indicates that the cardholder has entered the correct PIN at the PoS terminal.

2.4.2.2 Magnetic Stripe Card Limitation

Magnetic stripe cards work simply as memory sticks and are best suited for the applications like ticketing, loyalty pass where security is not of prime importance. Apart from storing and retrieval of static data, not many operations could be performed on the magnetic stripe cards. The other problem associated with magnetic stripe card is the amount of information that can be read. The access control policies on fetching the amount of data cannot be defined on the magnetic stripe. This means any reader with magnetic head, can read all the contents stored within a magnetic stripe. This came as an opportunity for attackers to practice a trivial type of fraud on magnetic stripe cards: Skimming. In skimming fraud, the magnetic stripe technology cannot prove the difference in the actuality of a real and counterfeit card generated through skimming.

Skimming: One of the methods used by card thieves is skimming. In this type of fraud, ATMs are physically modified with a minimal effort in a manner that is difficult for the cardholder to detect. The way skimming works is that thieves put a card scanner on top of the little slot where the payment card is typically inserted in an ATM machine. These skimmers allow the card to pass through them into the ATM slot while also scanning the card and stealing the numbers off it. This happens so discretely that

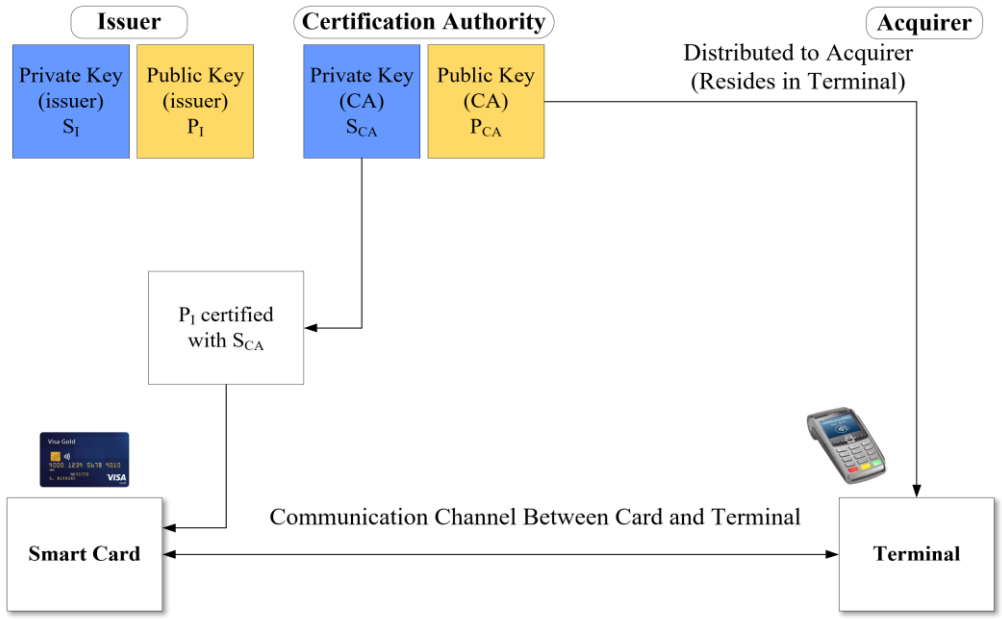

Figure 8 - The process of Static Data Authentication (SDA)

many victims have no idea that something is a miss until they look at their bank statements probably weeks later. And because many ATM card slots use similar designs, there are plenty of skimmers that are designed to look almost exactly identical to legitimate card slots making it even harder for a customer to realise what is going on.

Of course, though, the magnetic stripe transactions are protected with a PIN, and to do anything notoriously useful with the skimmed card. To steal the PIN, miscreants also install small pinhole cameras in inconspicuous locations on the ATM to capture footage of cardholder keying the PIN. To capture PIN, there are also number pad overlayers available in the black market which just look like the keypad on the ATM.

Nowadays due to this becoming common practice, the public and banks have become more aware and the scammer may get caught when they try to retrieve their scamming equipment from the ATMs. To resolve this, more advanced skimming devices on the black market transmit stolen card information and PINs wirelessly making it much easier for the fraudsters to practise their scheme without getting caught.

2.4.3 Chip Cards and EMV

Chip cards or smart cards fill the gap of ensuring the security and integrity of data stored inside the card's memory. Assisting this secure storage is a microprocessor capable of performing much advanced cryptographic tasks such as support for AES and DES algorithms. Data in smart cards is stored as files with the root file called as Master file and branch files are called elementary files [32].

EMVCo [25] is a consortium of card payment networks that was set-up to maintain interoperability between payment card operation. EMVCo created the "Integrated Circuit Card Specifications for Payment Systems". These specifications are related to ISO7816 [33] and create a common technical

basis for card and system implementation of a payment system. Integrated Circuit Card Specifications for Payment Systems can be obtained from an EMV website [25]. These specifications define an EMV chip and pin protocol [26], a messaging standard using which the payment cards operate and communicate with compatible readers. EMV draws its key functionality using features provided by these smart cards. Let us look at how the EMV protocol provides armour against cloning and counterfeit attacks. EMV specifications define three security features in the chip and pin protocol:

- Card authentication,
- cardholder authentication, and
- transaction authorisation.

2.4.3.1 Card Authentication

Validates that the card and the Issuer involved in the transaction are authentic, and not counterfeit. This step also ensures the uniqueness of card-specific data set up by the card issuing bank has not been altered by anyone. EMV specifications define at least three modes of card authentication [26]:

- Static Data Authentication (SDA),
- Dynamic Data Authentication (DDA) and
- Combined Data Authentication (CDA).

All these methods rely on the signature scheme based on asymmetric cryptography as established by relevant payment card network. Let us explore in detail the card authentication mechanisms.

PKI in EMV is a 3-tier architecture with root Certification Authority (CA) signing public keys for the card issuer which consecutively signs relevant data and public keys for the card [34].

Static Data Authentication (SDA). SDA was designed for initial versions of smart cards that had limited processing capability and can securely store only limited data. SDA validates the integrity of the application data stored within the smart card IC. However, SDA does not authenticate the card itself.

The process of SDA is shown in Figure 8. During the card personalisation phase (before the card is issued to the cardholder), the card issuer prepares the payment Application Data (AD) which is relevant to the cardholder account. The AD is signed with the card issuer's private key (S_1) and is stored in the smart card IC. The card issuer's public key (P_1) is signed by the CA's private key (S_{CA}) and this issuer's public key certificate is stored in the smart card IC.

During the transaction process, when the cardholder inserts the card into the PoS terminal, the CA's public key (P_{CA}) (which is issued to the acquirer and resides with the PoS terminal) is used to validate the issuer public key certificate which resides within the card. The PoS terminal extracts the issuer public key (P_1) from the certificate. In the next step, the PoS extracts the signed application data

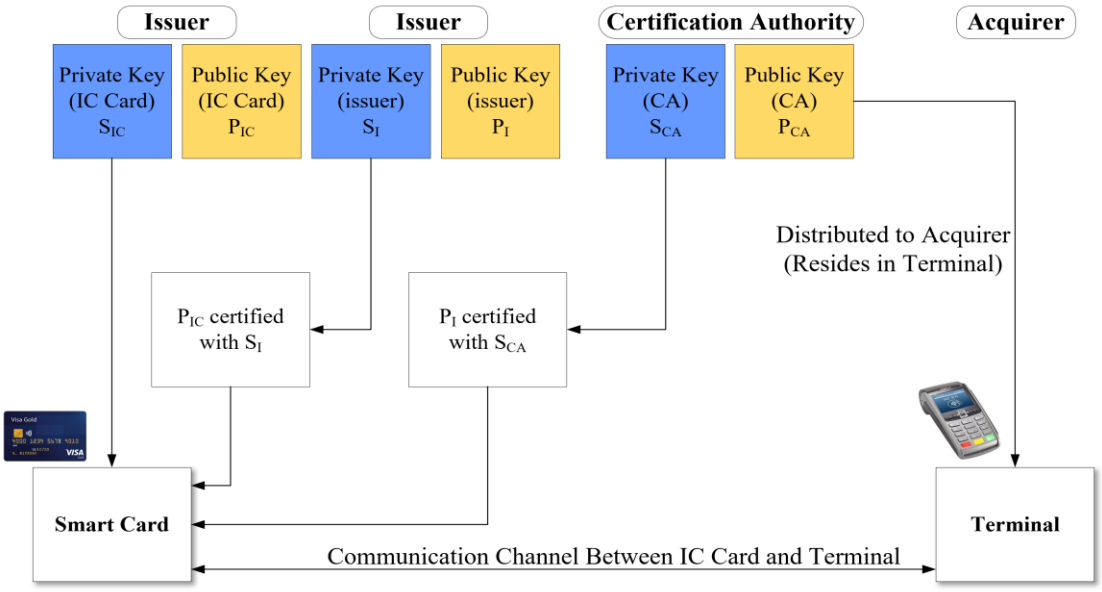

Figure 9 - The process of Dynamic Data Authentication

(signed by S_1) from the card and validates it using P_1. Once the signed application data is found valid, the reader and the issuer can be assured that the data in the smart card IC can be trusted and has not been altered.

Dynamic Data Authentication: DDA is an advance scheme of card authentication, where each card is personalised with its private key used by the card to generate a signature – Signed DDA (SDDA). The signature encodes transaction data and a random number given to the card by the PoS terminal which guarantees uniqueness for every transaction. In this scheme of card authentication, the card's public key (P_{IC}) is signed by the issuer's private key (S_1) and card public key certificate is stored in the smart card IC. The issuer public key (P_1) is further signed by CA's private key (S_{CA}), and issuer public key certificate is stored in the smart card IC. The CA's public keys are distributed to the PoS terminals. During a transaction, the validation of SDDA by the reader indicates that the card is authentic and is issued to the cardholder by the card issuer.

Combined Data Authentication. With CDA, both the POS terminal and the card issuer verify the integrity of the payment card. Much similar to DDA, in CDA the card generates a transaction specific signature (SDDA) which is used by the PoS to verify the transaction. In addition, an Application Cryptogram (AC) is also generated by the issuing bank to be signed by the card using a shared secret key. An AC consist of transaction specific data and a random number which guarantees transaction uniqueness. During a transaction, a request to sign the AC is sent to the card by the card issuing bank. Having validated the received signed AC from the card, the issuing bank can guarantee the card authentication.

2.4.3.2 Cardholder Authentication

Cardholder Authentication guarantees the identity of the actual cardholder making a transaction. This is achieved by using either of the three methods: Signature, PIN and no CVM (Cardholder Verification Method). PIN verification can either be online or offline. Online PIN verification is much similar to what we saw in Section 2.4.2.1 with a magnetic stripe. However, with offline PIN verification process, the PIN entered by the cardholder is compared to the PIN loaded into the card.

2.4.3.3 Transaction Authorisation

The third security feature defined in EMV chip and pin specification [26] is transaction authorisation which indicates whether the transaction was approved or declined. This is achieved by using either of the two modes: offline and online method.

Transaction authorisation relies on Transaction Certificate (TC) which is a cryptogram generated in a similar manner as of AC. TC combines the data used for the transaction, the type of cardholder authentication used for the transaction and the mode of transaction authorisation used for the transaction. Therefore, TC can be regarded as a receipt which guarantees transaction non-repudiation [35]. Figure 50 in Appendix A, details the EMV transaction sequence diagram. Markantonakis K., Mayes K. (2008) “Smart cards for Banking and Finance” [34], provides a more detailed analysis of EMV Chip and Pin transaction protocol.

2.4.4 Contactless Cards and EMV

Advancements in the EMV payment ecosystem towards fast and secure payments were achieved with the introduction of contactless cards. Unlike chip and PIN card which require point of contact for communication with the reader, contactless card talks to POS terminal wirelessly using RFID technology. Contactless payments are designed for low-cost in-store payments usually about £30 in the UK and do not require PIN verification for cardholder authentication.

The EMV contactless transaction protocol is derived from the EMV chip and pin protocol and is further enhanced to minimise the transaction processing times at the PoS terminal. The EMV contactless specifications define at least two variations of contactless transaction protocol. Both of these protocol sequences derive three security features from EMV chip and pin protocol as defined in Section 2.4.3.

Figure 10 - An example checkout page showing the card data fields required to make an online CNP payment transaction

2.4.5 Card Not Present (CNP) Transactions

These are transactions which are performed by the merchant in physical absence of both the cardholder and the card. All CNP transactions require the cardholder to submit: 16-digit card number, the expiry date and sometimes the CVV2 and cardholder billing address information. The identity of the cardholder making a purchase cannot be established in CNP transactions. CNP transactions can be categorised into two types: online CNP transactions and Mail Order/Telephone Order (MO/TO) transactions.

Online CNP Transactions. Transactions performed online on the merchant web store are categorised as online CNP transactions. The cardholder enters their payment card details on the merchant

provided checkout page. Figure 10 illustrates an example checkout page provided by the merchant website. Typically, the checkout page contains the following pieces of information for the customer.

1) Selected product for purchase, type of payment method selected by the customer, payment currency type and amount.
2) Type of online CNP payment protocol supported (Chapter 4 gives more details on types of CNP protocols)
3) Fields for the customer to input her payment card details. An asterisk (*) symbol indicates minimum fields required by the merchant to process this transaction. Note that there is no (*) symbol at 'Security Code' field which indicates that CVV2 is not mandatorily required for this transaction.
4) Cardholder address fields which serves two purposes for the merchant: firstly, it is used by the merchant to ship the purchased items to the customer and secondly to compare the cardholder address with the card issuing bank for fraud protection purposes. In Chapter 6, we will detail most of the fraud protection filters (also called as transaction safeguards) used by merchants and card issuing banks.
5) Use of captcha for cardholder validation is an example of how to not design a checkout page as it adds inconvenience to the customer at the checkout. Providing convenience to the customer at checkout is one of the primary objectives of the online merchant. We have several patents (Amazon one-click [36] for example) filed by online merchants that preserve ways of providing customer convenience at checkout.

The research work in this PhD mainly focuses on the security of online CNP payment systems and transactions. The online CNP payment system is complex and has multiple protocols, therefore, we provide a complete description of CNP payment protocols in Chapter 5.

Mail Order/Telephone Order (MO/TO). The payments performed over an e-mail or over the phone are categorised as MO/TO. For Mail order transactions, the cardholder provides her payment card details to the merchant via an e-mail. Telephone order transactions require the cardholder to reveal her payment card details to the merchant, verbally and over the voice. Merchants for MO/TO maintains a virtual terminal where they insert the cardholder provided payment card details to initiate payments.

2.5 Payment Card Fraud Overview

In this section, we review fraud over card payments as it affects the global payments system. Payment card fraud is an international issue that spans across nations, states and borders. Fraud overpayment cards has amounted to a total of $22.80 billion globally for the year 2016 [1]. This is 4.4% increase in the global card payment fraud rate as compared to the year 2015 where it was recorded $21.79 billion [37]. The United States (US) alone accounts for an overall of two-fifths (38.7%) of the global card

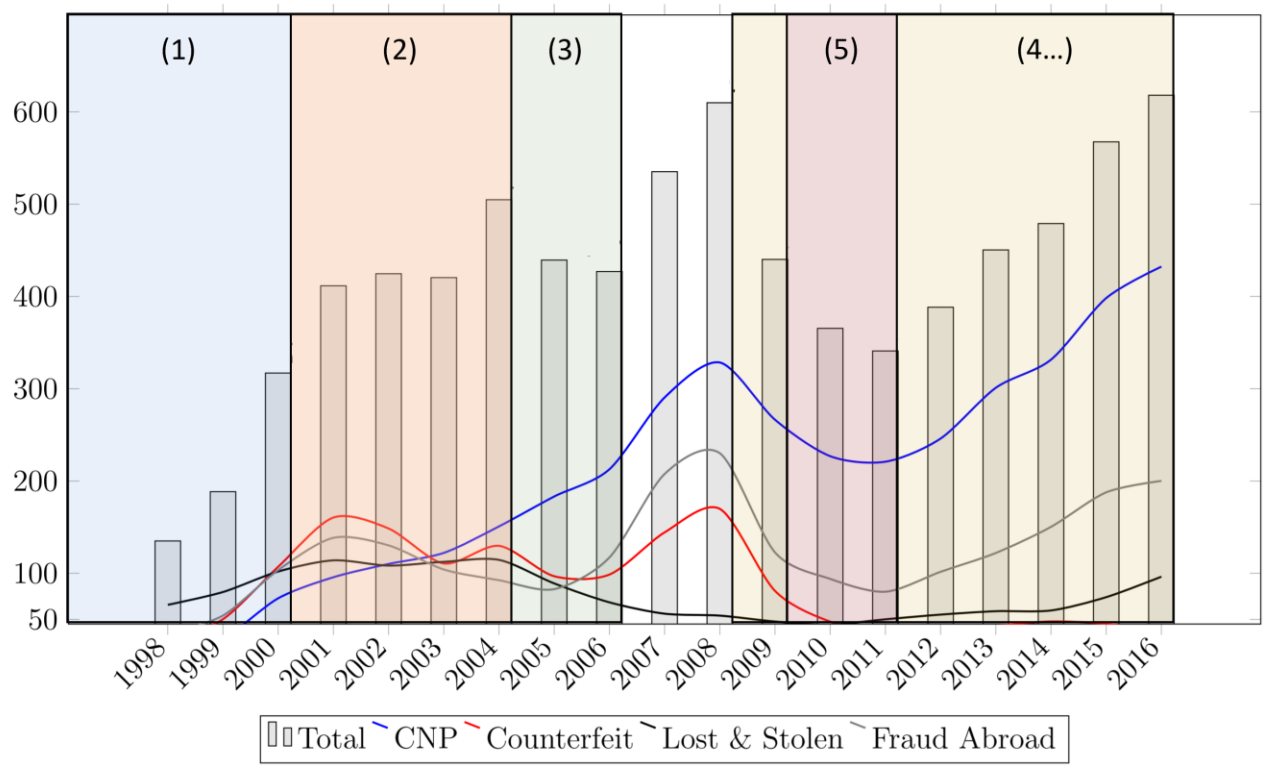

Figure 11 - UK Card Fraud by Type from the year 1998 to 2016

payment fraud rate amounting to a total of $8.45bn for the year 2016 and it is estimated that by the year 2020 the US card payment fraud could surpass $28bn [38]. To the contrary, card fraud losses for Europe in 2016 reached $2.12bn and about 73% of the European card fraud came from the United Kingdom (UK) and France [39]. In fact, for Europe, card payment fraud in one of the EMPACT priority, under Europols priority crime areas (2018- 2021 EU Policy Cycle [40]). Over the last six years it is established that the costs associated with the losses on financial systems constituted to the largest single category of fraud across the globe and over the Internet [37].

So how does a perpetrator achieve in practising fraud over electronic payment systems? A simple answer to this question is fraudsters in most cases targets weakness in the payment system technologies and exploits them for their interests and/or monetary gain. The methods used by fraudsters to abuse the payment system varies and depends upon the type of system (among card present and CNP) being targeted. Fraudsters methods can be well understood by mapping the payment card fraud patterns over the evolution/improvements of card payment technologies.

For this study, we take the payment card fraud patterns from the UK as an example. Figure 11 shows UK card fraud statistics from 1998 to 2016 [2]. The statistics reveal that ratio between the different types of card fraud changes year on year. In the figure, the red line represents fraud losses on card present payment types and blue line signifies the fraud that occurred over CNP payment interface. Figure 11 also shows the introduction for significant security improvements in the card payment system technology listed below:

(1) Online CNP payment systems were introduced in 1994 enabling merchants to accept card payment on their virtual stores on the Internet. Customers paid for the purchased items by entering the card details as printed on their cards.
(2) 3 Domain (3D) Secure protocol was introduced to secure online CNP payments (discussed in next chapter). This enabled the card issuers to secure CNP transactions with the use of static passwords only known to the registered cardholders.
(3) EMV Chip & PIN cards introduced into the UK replacing the existing magnetic stripe cards. The introduction of Chip & PIN cards was phased, starting in 2004 with the majority of cards replaced by the start of 2006.
(4) 2009 reported the first year on year decrease in total card fraud reported in the UK, this decrease is mainly attributed to adoptions of sophisticated fraud screening detection tools by the Issuing banks [2].
(5) In the UK the EMV Static Data Authentication (SDA) cards were replaced by Dynamic Data Authentication (DDA) and Combined Data Authentication (CDA) cards which perform additional cryptographic authentication, making them more secure than the original SDA cards [32][41].

Figure 11 illustrates that each time a new security feature is introduced to the card payment system the pattern of card fraud changes. It also shows that when a new security feature is successful in reducing card fraud in one particular area, in the following years, card fraud will increase in other areas. The overall result being that the total value of card fraud has continued to steadily increase, despite a number of significant improvements in EMV card security during the period 1998 to 2016.

For example, prior to 2004 magnetic stripe payment cards were vulnerable to cloning because of the magnetic stripe, section 2.4.1, and cards authorised by signature were very vulnerable to being lost in the mail before the customer had signed the card. After 2004 the type of fraud committed moved towards cloned magnetic stripe cards being used overseas and “card not present” fraud (e.g. telephone payments). Both of these fraud types side-stepped the new Chip and PIN security features of EMV by taking advantage of weaknesses in non-EMV payment streams. Prior to 2004, before the introduction of EMV chip and pin, fraudsters have shown to target the magnetic-stripe based card present transactions. CNP fraud, on the other hand, shows a gradual increase from the year 1998 to 2008 just before transaction risk profiling was introduced. However, fraudsters have shown to bypass CNP transaction safeguards as this is reflected by the growing fraud rates from the year 2011 of CNP payment systems. Presently, CNP fraud standouts to be the single largest category of fraud amounting to a total of 70% of the total card fraud rate for the year 2016 [2].

2.6 Conclusion

Online card payments are the first of many new payment technologies being introduced to the payment system. By analysing the vulnerabilities that online CNP payments have introduced we can better understand how those vulnerabilities impact the patterns of card payment fraud in the future. The new technologies, described in this section, will both combat fraud and create new opportunities for fraud, as shown by Figure 11.

In the following chapters we will study the literature of card payment systems to analyse the vulnerabilities introduced to the protocols. We outline an analysis methodology that we used to analyse future changes to online card payments protocols required for the introduction of subsequent payment technologies and identify potential vulnerabilities.

Chapter 3. Literature Review of Card Payment Protocols and their Vulnerabilities

This chapter presents a structured literature review performed on the security of card payment protocols and their associated vulnerabilities. The literature work selected in this chapter sorts some of the leading scientific research conducted on the real world and deployed payment protocols. With this literature review, we aim to learn the research gaps explored by the literature and explore the vulnerabilities identified by each research outcomes. We also discuss the methodology adopted by the previous research that has achieved in obtaining the required data to overcome any restrictions on access to data and protocols. The chapter will conclude with a discussion on the takeaways from the literature review which is then applied to performing security analysis on the online payment protocols.

Online CNP payment protocols are part of the payment system which uses cardholder's data that can be obtained from another mode of payments like EMV chip and PIN, EMV contactless and Magnetic stripe payments. This means the security of online payment protocols must be analysed in the broader context of the payment system. Therefore, our literature review is primarily divided into three sections, each exploring the analysis on a different form of payment systems. The sections include security analysis of *EMV chip and PIN protocol, EMV contactless protocol and CNP payment protocols.*

3.1 Research Categories

This section lists the academic research papers included in this literature review and identifies which of the four categories to which they are applicable.

Exploitable Vulnerabilities in the EMV Protocol

Murdoch et al. (2010) "Chip & PIN is Broken" [43]

Roland and Langer (2013) "Cloning credit cards: a combined pre-play and downgrade attack on EMV contactless" [44]

Bond et al. (2014) "Chip and Skim: cloning EMV cards with the pre-play attack" [45]

Barisani et al. (2011) “Chip & PIN is definitely broken” [46]

Degabriele et al. (2011) “On the Joint Security of Encryption and Signature in EMV” [47]

Anderson et al. (2005) “Chip & SPIN” [48]

Exploitable Vulnerabilities in EMV Contactless Payment Technology

Emms et al. (2014) “Harvesting High-Value Foreign Currency Transactions from EMV Contactless Credit Cards Without the PIN” [28]

Emms et al. (2013) “Risks of Offline Verify PIN on Contactless Cards” [49]

Francis et al. (2012) “Practical Relay Attack on Contactless Transactions by Using NFC Mobile Phones” [50]

Roland and Scharinger (2013) “Applying Relay Attacks to Google Wallet” [51]

Roland et al. (2012) “Relay Attacks on Secure Element-Enabled Mobile Devices: Virtual Pickpocketing Revisited” [52]

Roland et al. (2012) Practical Attack Scenarios on Secure Element-Enabled Mobile Devices [53]

Kfir and Wool (2005) “Picking Virtual Pockets using Relay Attacks on Contactless Smartcard Systems” [52]

Diakos et al. (2015) “Eavesdropping near-field contactless payments: a quantitative analysis” [54]

Hancke (2011) “Practical Eavesdropping and Skimming Attacks on HF RFID Tokens.” [55]

Security Analysis of CNP Payment Protocols

Murdoch et al. (2008) “3D-Secure: or How to not design an authentication system”

Drimer et al. (2007) “Optimised to Fail: Card readers for online banking”

Redteam-pentesting. (2005) “Online Banking: Warning about the deceptive security of the iTAN process”

Redteam-pentesting. (2009) “Man-in-the-Middle Attacks against the chipTAN comfort Online Banking System”

Redteam-pentesting. (2005) “New banking security system iTAN not as secure as claimed”

3.2 Security Analysis of EMV Chip and PIN Protocol

EMV Chip and Pin is an open-source and well-documented protocol. The proven complexity of EMV Chip and Pin protocol and its widespread use across the globe made the protocol much attractive for research communities. There is a substantial amount of research addressed on the security analysis of EMV Chip and Pin protocol. In this section, we will focus on the research papers which have identified practical exploitable vulnerabilities in the EMV protocol.

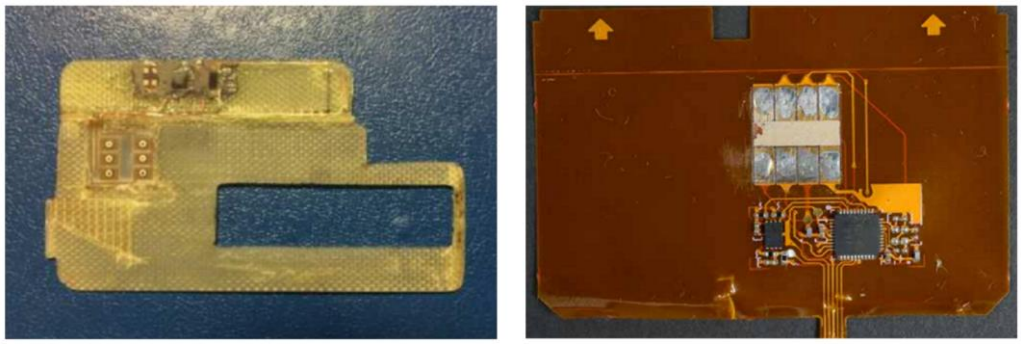

Figure 12 - Shows two real EMV shimmer devices found in EMV enabled ATM machines

As discussed in the previous sections, EMV defines functions that establish the authenticity of the card, however, there is no mechanism defined which verifies the authenticity of the reader that the card communicates with. This provides ample opportunity for an attacker with a rogue reader to communicate with the EMV card.

Figure 12 shows two EMV shimmer devices found in EMV enabled ATM machines. These shimmers are attacker tools that intercept communication between the EMV card and a PoS terminal or an ATM. Although there is no known possibility for an attacker to create a cloned copy of victim's EMV card, shimmed details can be used to create a magnetic stripe version of victim's card.

EMV smart card chip contains all components of cardholder payment application data found in magnetic stripe except for CVV. EMV interface contains its own version of Card Verification Value generally referred to as iCVV or dynamic CVV. iCVV which is different to magnetic stripe CVV prevents the shimmed data being copied and used over magnetic stripe interface. The rationality behind the success of shimming can be related to the negligence of some card issuing banks while validating the CVV: they do not validate the CVV while authorizing a magnetic stripe transaction.

Murdoch et al. (2010) [43] identifies a vulnerability in the EMV payments system which allows an attacker to authorise a payment whilst entering an incorrect PIN. A man-in-the-middle device can subvert the cardholder verification process, see Figure 13. The MITM device tells the POS terminal that the PIN entered by the attacker is correct, whilst telling the EMV card that this is a transaction verified by signature and therefore no PIN required. This bypasses the primary security of the EMV Chip & PIN protocol i.e. the cardholder PIN.

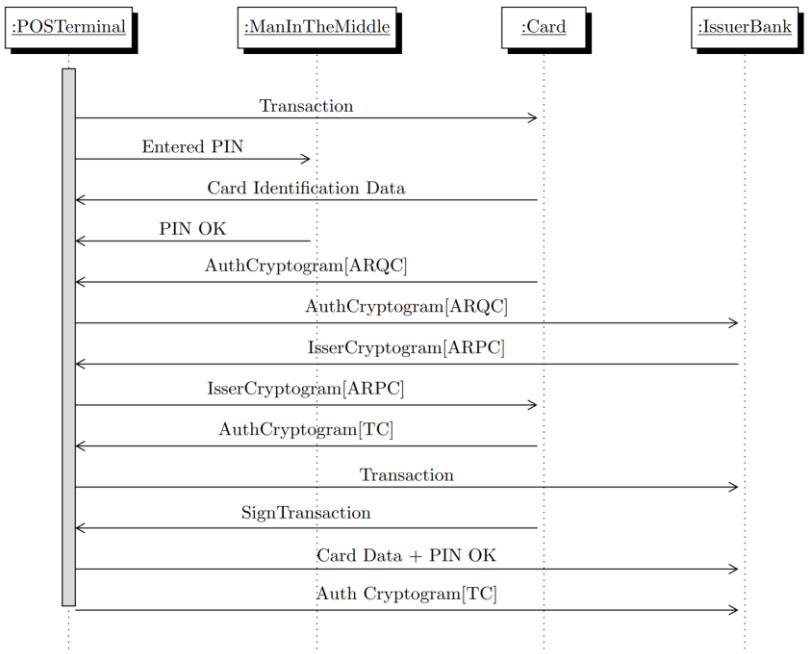

Figure 13 – Murdoch et al. (2010) protocol sequence

The research team performed practical experiments to demonstrate that the vulnerability was present in UK issued credit / debit cards and UK POS terminals. The importance of this research was highlighted in 2012 when criminals were arrested in France, they had exploited the vulnerability to conduct 6,000 fraudulent purchases with a total value of more than €500,000 [56]. This research also uncovers critical failings in the banks transaction validation processes. The transaction data transmitted to the issuing bank includes the Terminal Verification Results (TVR) and the Issuer Application Data (IAD), which together encode the results of the cardholder verification carried out by the POS terminal and card. These data are signed by the card, so the man-in-the-middle cannot alter them. However, despite this, the data required to clearly identify the fraud are split across several data fields some of which are visible to the POS terminal and others are visible to the issuing bank. This creates an ambiguity in the data which makes it difficult to detect this type of attack at either the POS or the issuer.

The EMV transaction protocol is designed to ensure that the EMV payment cards issued by many different issuing banks are accepted at any of the POS terminals / ATMs worldwide. This is a challenge as the cards, POS terminals and ATMs support multiple authorisation modes (online / offline), authorisation methods (PIN, signature, contactless) and cryptographic authentication technologies (SDA, DDA, CDA). To make any EMV card compatible with any POS / ATM, the protocol includes a negotiation at the start of the transaction to decide on authorisation mode, method and cryptography. The POS / ATM will select the most secure combination of mode, method and cryptography available to both the card and the POS / ATM.

This negotiation process is a significant weakness in the protocol. There are a number of research papers that prove it is possible for a man-in-the-middle to alter the capabilities of the card or the POS / ATM, to cause the POS / ATM to select an exploitable authentication mode, method or crypto. This type of attack is called a downgrade attack, where vulnerability is discovered in the EMV protocol and the attack must put the POS /ATM into a given mode to enable the vulnerability to be exploited. Two such attacks are Roland and Langer (2013) [44] and Barisani et al. (2011) [46].

The vulnerability discovered by Roland and Langer (2013) [44] allows the attacker to create cloned EMV contactless cards. In normal operation a cloned EMV contactless card should not be accepted, because the private key on the original EMV contactless card cannot be copied. This means that the cryptographic signature produced by the cloned card will not be validated by the POS terminal. However, the downgrade element of the attack alters the capabilities of the cards to fool the POS terminal into performing a magnetic stripe mode contactless transaction rather than the EMV mode transaction. The cryptographic protection on the magnetic stripe contactless transaction is much weaker than the combined RSA / 3-DES protection employed in EMV mode transactions. This paper demonstrates that the CVC3 code, used to authorise magnetic stripe mode transactions, can be manipulated to reduce the number of possible values to 999. This allows a cloned card to be created with the 999 possible CVC3 responses encoded upon it.

Degabriele et al. (2012) [47] describes a theoretical partial oracle attack on the RSA cryptography used by DDA / CDA cards. This paper demonstrates that it is theoretically possible to forge the DDA / CDA cards digital signature. However, even at best, the attack would need to run 4,639 partial transactions against a card to generate the forged digital signature, making the attack impractical. Given that each transaction takes approximately 500ms, to complete the attack would require access to the card for 38 minutes.

Barisani et al. (2011) [46] was a black hat presentation at DEFCON 19 which covered a number of known issues in EMV such as the Murdoch et al. (2010). Part of which introduced a downgrade attack in which POS terminal can be convinced to reveal the PIN entered by the cardholder in plain text rather than in enciphered form, thereby allowing a man-in-the middle to record the PIN.

3.3 Security Analysis of EMV Contactless Payment Protocol

There are a number of known vulnerabilities in the underlying technologies which support EMV contactless payments, these can be split into the following categories:

- eavesdropping contactless payments
- extending the effective range of NFC
- Flaws with the EMV contactless protocol

These categories of vulnerability have one thing in common, they take advantage of the underlying contactless / NFC technology. They are very difficult to guard against and or prevent.

Eavesdropping, Skimming and Extended Range Reading

Contactless payments utilise the ISO-14443 wireless communications standard (International Standards Organisation, 2011), which is an open standard used in many different contactless applications on smartcard and mobile devices. Use of this common standard leaves contactless payments vulnerable to data hijacking attacks such as eavesdropping, skimming and extended range reading. The data gathered by these attacks include the 16 digit card number (PAN) and the card expiry date, which research shows is sufficient to create a new account on Amazon.com and make online purchases, as we will see in Chapter 6. This is due to the minimal security checks on some websites which do not enforce a full check on all of the card-not-present security fields recommended by EMV, i.e. the PAN, expiry date, CVV2, cardholder name and cardholder address. Therefore despite the cryptographic security that prevents cloning of EMV cards based on the data obtained through contactless eavesdropping, skimming and extended range reading; the data collected are still useful in performing card-not-present attacks.

Eavesdropping

A number of research projects have looked into the practicalities of eavesdropping the ISO-14443 wireless communications. These projects show that it is possible to eavesdrop the data from a contactless payment at a distance of 1 metre. The research does prove that eavesdropping produces exploitable data, thereby making the contactless EMV cards vulnerable to attack. However, the research also shows that the equipment required to perform contactless eavesdropping is very specialised requiring a great deal of electronics expertise to build. For instance Diakos et al. (2015) [54] presents excellent research which builds the eavesdropping equipment into everyday objects such as a shopping trolley. However, as the research also shows, the RF receiver and the signal processing equipment required are complex and would require a great deal of work to make the equipment portable enough to be used in real-word attack scenario.

This would make eavesdropping a much less attractive method of collecting credit card data when compared with skimming attacks using an NFC enabled mobile phone. Research by Francis et al. and research by Emms et al. show that skimming attack can be performed using off-the-shelf Android mobile phones which are very portable and discreet.

Hancke et al. (2011) [55] makes a comparison between eavesdropping and skimming attacks using the same equipment. The result of the comparison between the eavesdropping and skimming concludes that eavesdropping has the potential to read from a greater distance, however, the skimming provides

more reliable data reading. With eavesdropping being more susceptible to atmospherics, environmental conditions and RF interference.

Skimming

The popularity of NFC enabled Android mobile phones provides a perfect attack platform for contactless skimming as demonstrated in (Francis L, Hancke G, Mayes K, Markantonakis K., 2012) [57]. However, that is not the only potential attack vector, an attack platform was developed that masquerades as an NFC door reader by [49]. The door reader accesses all of the cards in a victim's wallet before activating the door opener. Our multiple card reader software utilises the standard anti-collision functionality present in the ISO 14443 standards [59] (part 3). Emms et al. (2013) [27] exploit the EMV *offline* Pin verify command from contactless interface. Contactless transactions do not require the cardholder to enter their PIN. However, the researchers discovered the offline PIN verify command is functionally available on most of the UK issued payment cards. This PIN verify command can be exploited by an attacker to guess the card PIN without blocking the card. The research demonstrated a viable attack scenario where a contactless physical access control reader is programmed with part of an EMV transaction protocol. When the user scans a wallet with payment card onto the access control reader, it selects a payment application on the card. Figure 14 shows a Verify PIN protocol sequence implemented by the reader. It can be seen from the figure that, the reader, after selecting the payment application, gets the number of PIN attempts left of the card. If not zero, the reader attempts a PIN verify command with a random PIN on the card. The command is executed until the right PIN for the card is guessed or until the PIN counter is zero.

Extended Range Contactless Reading

The maximum practical communication range of EMV contactless payments cards is approximately 10cm. EMV uses the restricted communications range of ISO 14443 as a design security feature. The cardholder authorises the payment by tapping their card on the POS terminal, the assumption being that the cardholder must be present at the merchant location to authorise the payment.

There has been significant research into the extending the read range of contactless payment cards. Kirshenbaum and Wool (2006) [60] demonstrates that ISO 14443 cards can be read at a distance of 30cm which is 6 times the design distance. The experiments show that to increase the effective reading range of ISO-14443 cards that reader must increase the transmission power from 200mW to 4 Watts and increases the antenna size from 5cm diameter to 50cm diameter.

Hancke et al. (2011) [55] introduces an interesting concept, it utilises two separate antennas to extend the reading range. A standard ISO 14443 reader uses a single antenna to power the card, transmit data to the card and to receive the card responses. The two antenna approach uses one antenna to power and transmit, it uses the second antenna to receive the card responses. Using a second receiving

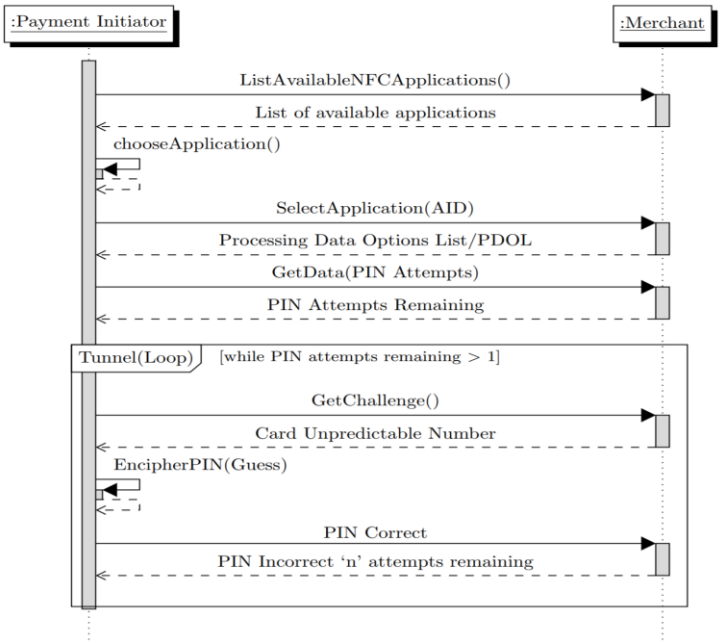

Figure 14 – Verify PIN protocol sequence

antenna allows the attack to increase the reading range of ISO 14443 whilst using less signal power and smaller antenna diameters.

One of the attack scenarios explored in Oren et al. (2013) [61] is a "mafia fraud attack" scenario. The POS terminal ("ghost") which is dedicated to receiving fraudulent transactions. An extended range contactless reader ("leech") is used to capture transactions from passing victims at a range of 1 meter whilst the contactless payment card is still in the victim's wallet. The "ghost" and the "leach" are connected by the relay allowing them to be many kilometres apart.

Emms et al. (2014)[28] demonstrated another practically viable attack on the EMV contactless cards. The researchers were able to bypass the contactless transaction limit from £30 in the UK to a million Euros. The flaw in the protocol is exacerbated by the fact that the EMV contactless specifications does not define the transaction value limits for transactions made in foreign currency to the card. For example, if the card is issued in the UK, an attacker can practically bypass the transaction limit by attempting a transaction in currency type other than GBP.

3.4 Security Analysis of CNP Payment Protocols

Murdoch et al. (2008) [4], analysed the architecture and design of 3D-Secure 1.0 and termed the protocol as an example of "how to not design an authentication protocol". The authors exploited two design flaws associated with the 3DS 1.0 protocol: *Activation during shopping* and *the use of static passwords for authentication*. With activation during shopping, the customers were enrolled into the system at the time of making a purchase. The demerits were, the customers were not educated about the protocol and were not used to interacting with integrated frames loaded as pop-ups which made it difficult for customers to distinguish between an actual 3DS 1.0 window and a phishing window. Cyber fraudsters designed similar looking 3DS 1.0 pop-ups to steal customer card details to be able to

register the card and use it over the 3DS 1.0 system. Also, due to stringent password policies, it was difficult for the customers to maintain their password remembered. The password-based scheme was proven to be costlier for the card issuing banks as they were required to maintain a dedicated customer support team for changing or resetting of 3DS 1.0 passwords.

Later, the password-based scheme was made optional by the payment regulators and the card issuing banks were allowed a freedom to choose the type of authentication method to be implemented across the 3DS 1.0 protocol. This resulted in the card issuing banks deploying inconsistent and unpopular authentication methods which were more burdensome for customers. One such example is the Chip Authentication Protocol (CAP) that has been introduced in the Europe for securing transactions over the Internet. CAP cryptographically binds an online transaction with a one-time token generated by a handheld reader provided by the card issuing banks. Each time the transaction is initiated, the cardholder submits the transaction specific token (CAP response) along with the payment card data on 3DS 1.0 window. Once the token corresponding the transaction is successfully verified, the card issuing bank accepts the transaction. The CAP extended the features of EMV Chip and Pin protocol and there were no specifications maintained to explain how the CAP should be implemented for online cardholder authentication.

Drimer et al. [3] performed an analysis of CAP by reverse engineering the protocol and found that there were variations in the ways the CAP was implemented from one card issuing bank to another. Their detailed analysis on UK versions of the CAP protocol unveiled that the protocol was vulnerable to EMV chip and PIN middleman attack [43]. The vulnerability allowed the middleperson to circumvent the PIN security and generate CAP responses required for authentication over 3DS 1.0.

The CAP protocol may be further abused by criminals during mugging. There is a serious case reported in July 2018, where two French students were attacked to death for not revealing their card's PIN. Furthermore, there were inconsistencies in the ways in which the card readers from different vendors generated CAP responses. For example, card readers from Racal Watchword, generated incorrect CAP response in case of PIN mismatch but this was not informed to the users. Two technical reports from redteam-pentesting [37] [38] claimed that, TAN enabled protocols (static TAN, iTAN and chipTAN) which makes use of static or dynamic passcodes provided by the card issuing banks were vulnerable to phishing attacks. It was no surprise that TAN systems were further made optional.

3.5 The Contribution of Literature Review to this PhD Research

The literature review in this chapter has focused on three areas of academic research related to the security of the EMV chip and PIN protocol, EMV contactless protocol and CNP payment protocols. The research in the literature review influenced the research presented in this PhD thesis as follows:

3.5.1 Security Analysis of EMV Chip and PIN

Our methodology, which is discussed in the next chapter is influenced by the research into exploitable vulnerabilities in the EMV protocol. My guide (Aad van Moorsel) encouraged the use of controlled practical experiments in my research. This allowed me to demonstrate that the vulnerabilities highlighted in the protocol were exploitable in the real world, and thereby increase the impact of the research. This approach has proven very successful in our papers which are discussed in Chapter 6, Chapter 7 and Chapter 8.

Papers in this category looked at the system wide impacts of the individual vulnerabilities in the EMV Chip and Pin payment systems. This gives context to our research and has helped us to more fully understand the impact of the vulnerabilities identified by my research and assisting me to convey this message to a non-academic audience; the general public, law enforcement and the payment industry stakeholders.

Learning from the papers in this section, during the first of my research, I developed a part of EMV chip and PIN specification that helped me understand the EMV chip and PIN transaction process which benefitted my research to understand how the payment protocols are designed and implemented.

3.5.2 Security Analysis of EMV Contactless

Research into the exploitable vulnerabilities in the EMV contactless technology, have both influenced and confirmed the technology choices made in our experimental work. Work by Hancke (2011), Roland and Scharinger (2013), Roland et al. (2012) demonstrated the use of NFC enabled smartphones as a practical attack platform against EMV contactless payments. Learning from their research, I developed an application on NFC enable android phone which emulated the EMV contactless POS terminal. The NFC application developed was helpful in extracting the payment card details from its contactless interface. There are cases where features in the contactless transaction protocol / technology does not affect the security of contactless payments but has a potential to affect entire CNP payment system. This is further explained in Chapter 6.

Research presented by Emms et al. (2014) and Emms et al. (2015) helped me in understanding how the payment authorisation messages are handled by the card issuing banks in the backend.

Work by Kfir and Wool (2005) and Diakos et al. (2015) on extending the range of NFC comprehensively explores extended range reading and eavesdropping contactless payments which supported my assertions made on contactless cards that it is not difficult for criminals to steal user payment card details from a distance using contactless interface.

I started my research applying the lessons learned from EMV contactless technology on security assessment of university-based RFID access control systems. This project work [64] exploited a vulnerability which would allow attackers to create cloned copy of the university issued smart cards.

3.5.3 Security Analysis of CNP Payment Protocols

As discussed earlier, and even pointed by Murdoch et al (2014) [4], the deployed CNP payment systems has escaped academic scrutiny to a major extent. The research papers detailed in this section are only a broader overview of the challenges faced by payment protocol designers. The research articles available for CNP payment system do not provide an extensive technical detail on every aspects CNP protocols that are currently in use. Before I started my research work, there were a lot of questions still needed to be answered for example:

- What are the standards which are needed to be followed while accepting a card payment online?
- If the freedom is allowed for the merchants to choose the type of payment protocol (authorisation-only and authentication-enabled), then what is the minimum information required by the card issuing banks to process an online payment transaction?
- Do the distributed protocol choices will have any effect on security of CNP payment system? How secure such a payment system be?
- If the authorisation-only protocol is dependent upon static card details, then what is the data used by merchants and card issuers for their fraud protection algorithms?
- Is the user machine (mobile and PCs) trusted enough to host card issuer bound keys and digital certificates?
- As with EMV chip and PIN and EMV contactless, is the online payment transaction cryptographically bounded? How is transaction authentication performed in the CNP payments?

3.6 Conclusion

There are several leading academic research teams actively analysing the security of payment systems and transaction protocol and researching potentially exploitable vulnerabilities in the payment protocols. EMV Chip and Pin and EMV Contactless have been especially given a substantial importance because of the availability of clearly defined protocol specifications. However, the research gaps associated with CNP payment systems were clearly defined and with the increasing fraud rates, it can be withdrawn that more research is needed in the area of CNP payment systems.

In this literature review we have established a link between the existing academic research and the areas of weaknesses in CNP payment systems, which were of my potential interest of this research. Some of the weakness identified in the payment system which drove my research were:

- Wireless interface and the data available in EMV contactless interface introduce new categories of attack (i.e. skimming, eavesdropping and relay) for online payment systems.
- The data in the EMV chip and PIN interface can easily be read by false readers making it possible for attackers to use the stolen card details over online payment system.
- The 3DS 1.0 which required the cardholder to enter static passwords on a pop-up screen was more burden to the payment industry than a solution. This allowed a freedom for the online merchant to have options on the protocol they want to implement on their checkout systems.
- Cryptographically bounded one-time passcodes for online payments using EMV readers were vulnerable to chip and PIN attacks where an attacker can generate one-time passcodes from a stolen payment cards.

The literature review supports the assertion made in this PhD thesis that the security of the online payment system is fundamentally weakened by the philosophy of providing convenience to the customer at the checkout. Also, the requirement for backward compatibility makes it essential for the card data to be available in plain text across other interfaces which reduces the security to the least secure technology supported by the system.

Chapter 4. Analysis Methodology

This chapter details the methodology developed as part of this PhD for the security analysis of the CNP payment system. We analysed studies conducted on a range of systems, including payment systems, money mules, OpenSSL and OAuth implementation in Android. This allows us to construct a framework of terminology, methods, and recommendations for the use of attack landscaping and the associated disclosure processes as a research method. This is the main drive for our research into vulnerability discovery and attack landscaping. By analysing several case studies (papers recently published in the topics of attack and disclosure, as reference), we have formulated a framework which consists of following stages: Security Architecture Assessment, Attack Landscaping and Vulnerability Disclosure.

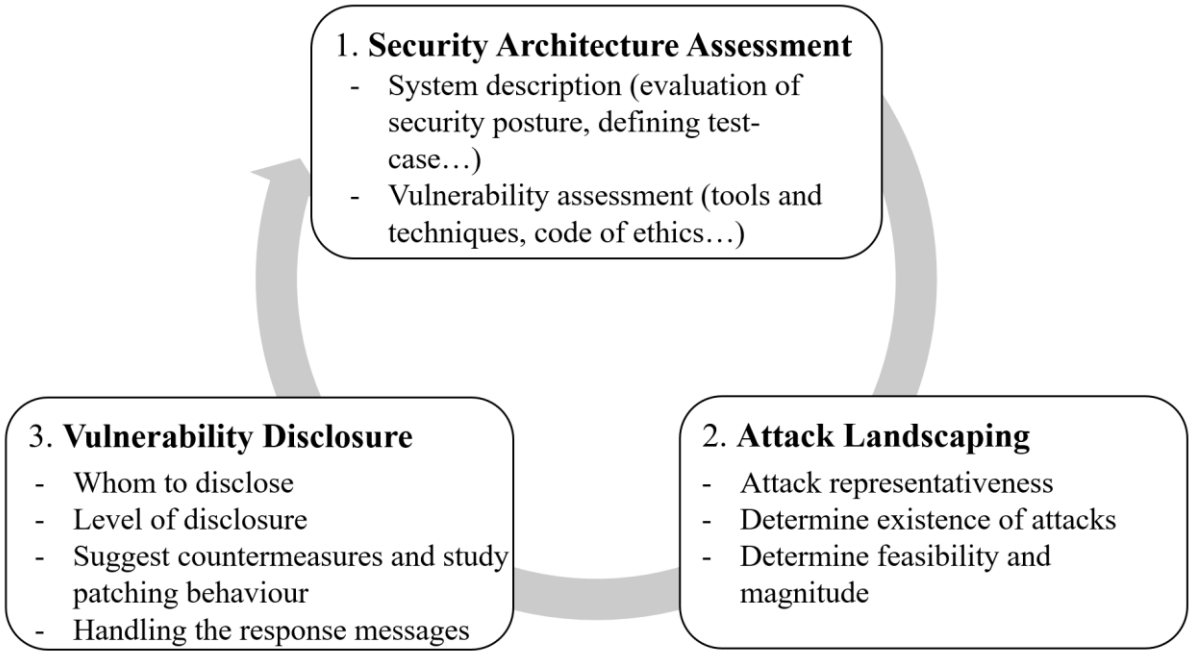

Figure 15 - Overview of Analysis Methodology

Figure 15. shows the overview of analysis methodology. It identifies three stages and numbers them according to their occurrence.

1. **Security architecture assessment:** refers to the process of identifying, quantifying and assessing the vulnerabilities associated with a system under study. With security architecture

assessment we capture the CNP payment system information in a meaningful way to evaluates its security posture against potential vulnerabilities. For an efficient security architecture assessment, our methodology defines the following approaches to be followed:

- **System description:** Comprehensively explains the processes and parties involved in a system including any messaging standards, communication protocols and their security features.
- **Vulnerability assessments:** This phase will outline the vulnerability assessment objectives, define the test cases, details technique used for vulnerability scanning and identifies what controls are in place against a vulnerability database. We evaluate strengths and weaknesses of existing protocols by assessing the fraud infiltration process and techniques.

2. **Attack Landscaping**: In the case when a vulnerability is discovered, the next step we follow in determining the feasibility and magnitude of the attack is to identify a representative instance of the attack. A systematic approach for this process is followed to ensure that an accurate likelihood is determined for each attack scenario.
3. **Vulnerability disclosure:** After discovering a vulnerability, the next step is to pass the findings to the affected parties (i.e. the developer, maintainer or owner of the system with the vulnerability). In an approach for consistent and coherent procedure to achieve effective vulnerability exercise, we will discuss in the following sub-sections.
    - Disclosing the findings: Identifies whom to disclose the vulnerabilities, determines the timing of disclosure and details the level of disclosure.
    - Post vulnerability disclosure: Apart from describing the patching nature, this area of our research gives insides into the behaviour of stakeholders on patching which includes timings taken by stakeholders to patch the vulnerability, technical abilities of stakeholders involved in patching the system.

Arguably, the above three discussed research methods are combined into a comprehensive methodology, providing scientific rigour to elements of the research of the attack, and providing scientific value by documenting the current state of the art (much like measuring a system in scientific experiments).

4.1 Security Architecture Assessment

Ensuring the security of CNP payment is a system wide concern. To date, resources in securing the CNP payment system have been heavily focussed on the following four aspects: data transport security, web client security, web server security and operating system security. Securing data during transport is important to preventing unauthorised parties from capturing transaction data, but it is only one part of the solution in securing the CNP payment system. The client and server-side security risks

are equally important to understand and address. The last security topic – certainly not the least in importance – is the security of operating system of the transaction processing systems. The operating system is the foundation on which commercial online application are built. Weakness in the foundation can be exploited to compromise the server regardless of the security attributes of the applications or databases that stores payment card data. We have Payment Card Industry Data Security Standard (PCI DSS) which governs the security requirement of the above four aspects for secure handling of payment card data.

However, one component which is inadequately addressed by the academic literature while assessing the security of CNP payment system is the security assessment of CNP payment protocol itself. Payment protocols defines rules under which the CNP payment should be processed. There is a common misconception that data transport security through the Internet protocols like HTTP and TLS are sufficient enough for the proper and secure functioning of the CNP payment system. However, HTTP and TLS are somewhat general in that they can be applied to any web-based session, whether or not it involves commercial transactions. Payment system protocols go beyond simply securing the communication channel; they also establish rules by which commercial transactions will be paid and provide a means for transferring payment between merchants and consumers bank account. To fulfil this gap in literature on the security of CNP payment protocols and to answer questions on growing CNP payment fraud, in this research we investigate the security of CNP payment protocols currently in use by the payment industry to accept card payments over the Internet.

Selecting target protocols for research. A number of protocols for supporting card payment transactions have been defined and implemented for the CNP payment system, the result has been a plethora of confusing and competing mechanisms to accept payments. For our research we will focus on payment protocols currently in use by the payment industry while accepting card payments online. We organise CNP payment protocols into two meaningful categories and present an overview of each protocol that have established themselves in today's burgeoning CNP payment system. Based on the type of payment processes supported, we classify CNP payment protocols into two categories and they are authorisation-only, and authentication enabled (which further includes 3 Domain Secure 3DS 1.0, 3DS 2.0 and Secure Electronic Transaction (SET)).

4.1.1 System description.

In the next step we capture CNP protocol information in a meaningful way and evaluate its security posture against potential vulnerabilities. We start by identifying the processes and parties involved in a CNP payment protocol under study. Further, we evaluate the security strengths and evaluates the effectiveness of security architecture of a protocol with an aim to explore its weaknesses. We aim to do this with a consistent representation for all architecture descriptions of protocols, so that the process of discovering attacks can be simplified. However, consistent and concise description of CNP

payment protocols is challenging because CNP protocols definitions are lengthy and security architectures are complex. As an example, for authorisation-only CNP payment protocol, there are no strict requirements in payment acceptance guidelines [39][40][41][42] (over 3000 pages in length when combined for all payment networks) and standards [43][44][68] that rules how payment should be accepted consistently. This has resulted in freedom for payment acquirers and merchants to have their own variations of protocol for accepting payments (see Chapter 6 for more details).

Security architectures of CNP payment protocols are further perplexed by the influence of regulatory standards which are country and regional specific. For example, European Commission has mandated the use of 3DS 2.0 for every CNP payment performed over the Internet, where-as, Australian Competition and Consumer Commission (ACCC) has rejected a proposal to mandate any specific any fraud product for online payment including 3DS 2.0 [71].

Such complexities and inconsistencies demand a suitable method to be defined to describe a system. Use of modelling languages which can express a system concisely is preferred. Modelling languages such as behaviour trees [72], UML sequence diagram [26][47], petri nets [74] and more are found to be applied by previous research on security analysis of computing protocols. For the description of CNP payment protocols we make use of UML sequence diagrams. Figure 16, shows an example use of UML sequence diagram from our study on 3DS 2.0 protocol (further detailed in Chapter 7). The above UML sequence diagram collates information from multiple sources in the 3DS 2.0 specifications [75] and combines them into a single model. Each message, response and function are numbered to provide an easy to follow up to documented link with the reference tables. Use of reference tables can clarify interpretations and can give a link between system description and system implementations.

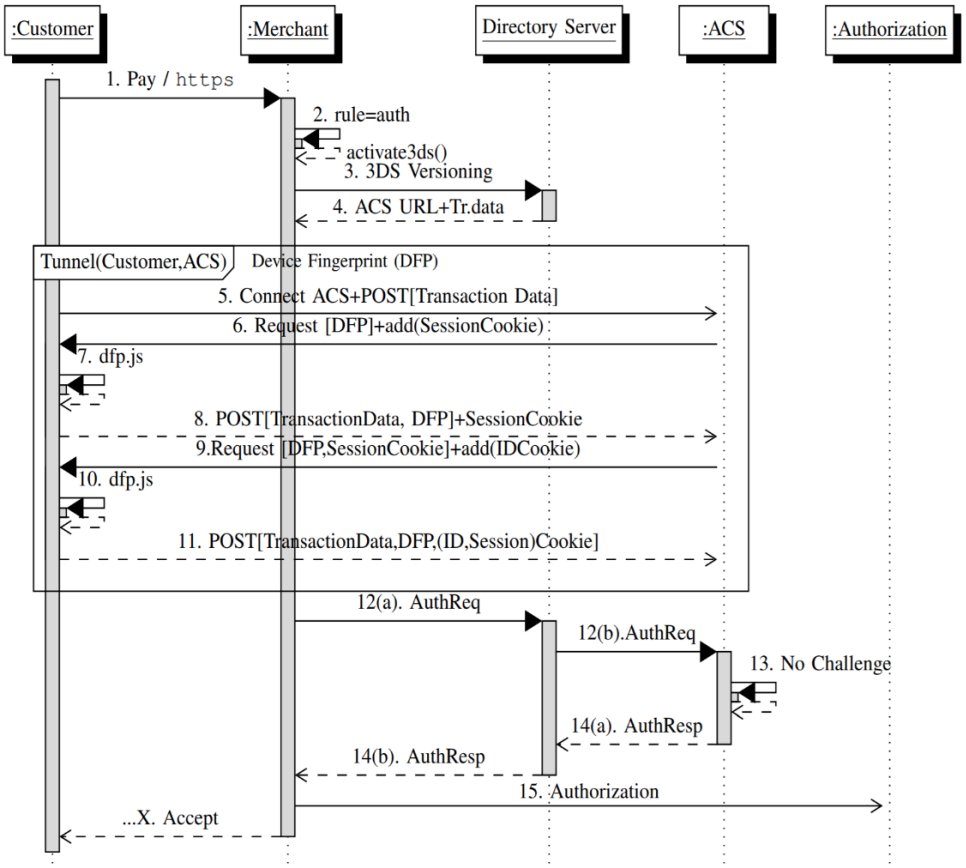

Figure 16 - Sample UML sequence diagram

Comparison tables are used on two occasions firstly while comparing the security features and messaging standards of each latest protocol with its previous versions and secondly when comparing the fraud filters supported by payment acquires and card issuing banks.

4.1.2 Vulnerability Assessment

Once the security architecture of CNP payment protocols has been captured, vulnerabilities can be assessed, and attack scenarios generated. Vulnerability assessment refers to the process of identifying, analysing and quantifying the potential vulnerabilities in the system under study. With vulnerability assessments on CNP payment protocols we

- outline the vulnerability assessment objectives,
- define the vulnerability assessment test-cases,
- detail techniques and tools used for vulnerability scanning and
- identify what controls are in place against a vulnerability database.

Our overall objective with vulnerability assessment on CNP payment protocols is to scan, identify and investigate weaknesses that are open to vulnerabilities without actually compromising the current security posture of CNP payment system. To achieve this, the first challenge we have is to select a suitable platform for research. We approached card issuing banks and payment networks requesting

for test accounts, test cards and more detailed specifications on the CNP payment system than what is available on the Internet. Having received a very limited response, we designed our research from a standard customer accessible platform. For the CNP payment system the customer accessible platform comes in two forms: a merchant account with acquiring banks and a customer account with online merchants.

A merchant accounts is offered by CNP payment acquirers to accept card payment online. For vulnerability assessments, we subscribed to the merchant accounts from payment acquirers and designed software tools which includes: 'tshirtshop' – a merchant webstore and a 'toolkit' that interfaces and performs security assessments on subscribed services. We also developed an android based NFC application to communicate with EMV contactless cards when required. To start with, we select at least 15 random online payment acquirers that provide payment services across countries and purchased the authorisation-only and authentication-enabled checkout services with real merchant trading accounts. Subscription also includes a package of payment fraud protection filters as used by real online merchants to defend their checkout platform against payment fraud.

For vulnerability assessments over merchant webstores, we select at least Alexa top 400 merchant websites and create customer accounts on each website so that the merchant website allows us to make card payments.

Experimental techniques we use for vulnerability assessments differ as it depends on the type of CNP protocol and the system being assessed. For example, for vulnerability assessments using merchant accounts and on authentication enabled CNP protocols we applied reverse engineering techniques using man-in-the-middle tools (network and application proxies including Wireshark, npcap and Fiddler) in the case where 'no' or 'partial information' about the protocol was available. This allowed us to understand variations in protocol implementation and enabled us to investigate any improper configurations and vulnerabilities. For authorisation-only CNP payment protocols, we applied security failure analysis [76] which is a test to systems to assess its ability to protect itself from deliberate attacks. We have designed several software tools for vulnerability assessments over merchant accounts and merchant websites which includes a merchant webstore, a website bot detailed next. We have also developed NFC android application to read contactless cards description of which is given in Chapter 6.

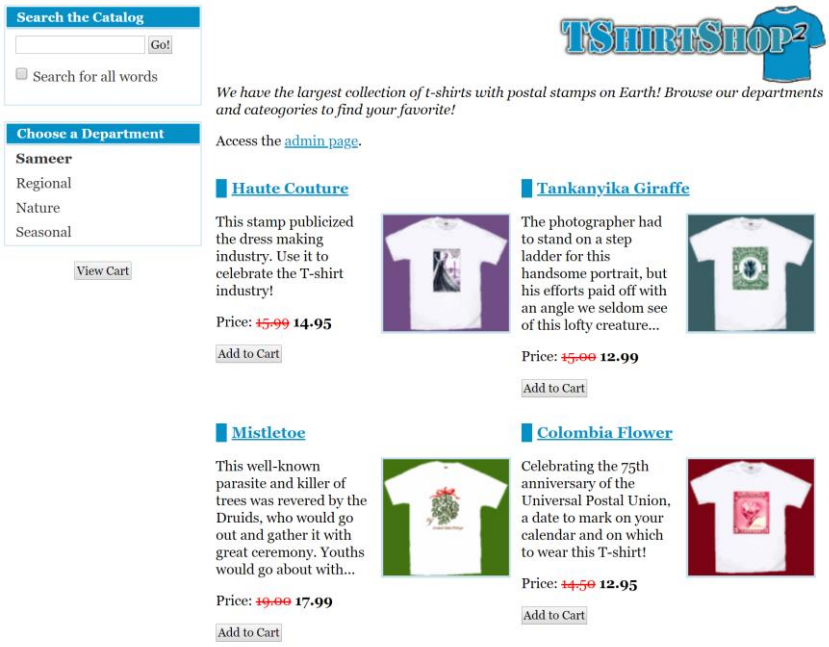

Figure 17 – a screenshot of tshirtshop homepage showing four of the 308 products available for purchase

tshirtshop. The merchant webstore we developed is a 'tshirtshop' (not made available for public use as it was not PCI DSS compliant) which allowed us to establish ourselves as online merchants and test the security settings provided by the payment acquires and card issuing banks. Figure 17 shows a screenshot of tshirtshop homepage, with four of the 308 products available for purchase. For each payment acquirer, the tshirtshop is capable to accept payment through authorisation-only and authentication enabled protocols. Typically, CNP payment acquirers offers two types of checkout systems to online merchants: hosted checkout and standalone checkout. Small online business websites use hosted checkout system also called as third-party checkout systems which is custom build to handle checkout operations (add and remove products) and to reduce the programming overhead involved in dealing with the response messages from payment acquirers which are forwarded to them by the card issuing banks. Standalone checkout systems on the other hand required the online merchants to setup their own checkout system which can handle the payment response messages send by the card issuing banks. Our tshirtshop is configured to support both of these checkout systems.

The checkout system on tshirtshop as shown in Figure 18, is standalone and is built from scratch, supports VISA and MasterCard brands of payment cards and can handle payment response messages without using any third-party checkout providers. An example payment response as received by the tshirtshop from PayPal as payment acquirer is as shown in textbox below.

```
RESULT=0&PNREF=VFHA0FF94691&RESPMSG=Approved&AUTHCODE=245PNI&AVSADDR=Y&AVSZIP=Y&HOSTCODE
=A&PROCAVS=Y&VISACARDLEVEL=12&TRANSTIME=2011-01-123:54:35&AMT=1.00&ACCT=1111&EXPDATE =1
215&CARDTYPE=0&IAVS=N
```

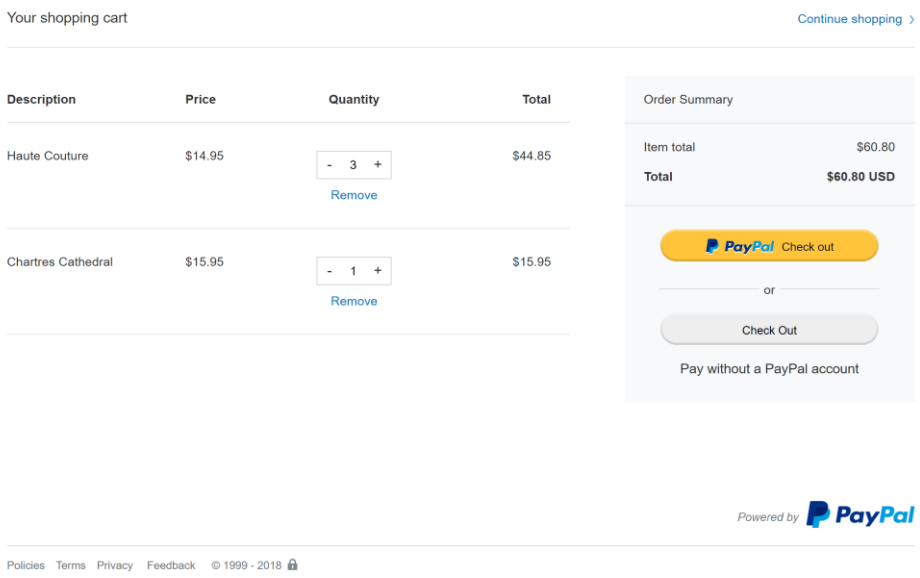

Figure 18 - PayPal hosted checkout system as supported by tshirtshop

As we will study in the next chapter, these payment responses are called as an authorisation response (AuthzRes) and contains a string which indicates the status of transaction of payment initiator and as processed by the customer's card issuing bank. The AuthzRes also indicates to the merchant the transaction status and any card data field put incorrect by the card holder. We started our vulnerability assessment process by studying AuthzRes codes from attacker mindset. For example, to make AuthzRes codes readable for the customers at checkout, these response codes are parsed by the merchants into the user understandable language. Perceiving the parsed AuthzRes from a rogue merchant's mindset; the codes conversely could be used to learn all card data fields.

Table 1 - Authroisation Response-code for tshirtshop with PayPal as Payment Processor

RESULT=114&PNREF=VXYZ01234567&RESPMSG=114&AVSADDR=Y&AVSZIP=N&IAVS=N&CVV2MATCH=N		
RESULT	> 0	Result > 0 indicates the transaction was declined. RESPMSG gives a brief reason for the decline of the transaction.
PNREF	Value	A unique value that identifies a transaction
RESPMSG	114	The transaction was declined (Card security code doesn't match). RESPMSG 113 implies that the transaction was declined because of the invalid card number.
AVSADDR	Y	Address of the cardholder was verified for the transaction
AVSZIP	N	Postcode provided at the checkout does not matches with cardholder's bank file
IAV	N	Cardholder country code is local
CVV2MATCH	N	Card security code mismatch

Table 1 shows an AuthzRes code for the merchant with PayPal as its payment processor. It can be derived from Table 1 that the transaction was declined because CVV2 supplied at checkout by the customer does not match with the actual card holder file with the bank's authorization server. This AuthRes code also implies that the card number and the expiry date were valid. In the next step, the AuthRes code is simplified at the checkout in user natural language. For example, during our experiments with valid card numbers when the expiry was not entered correctly while making a purchase on website x (name masked), the parsed response string as shown in Figure 19 explicitly

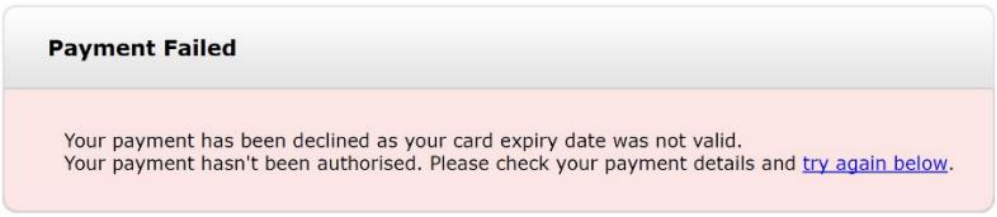

Figure 19 - Response code revealing the validity of a card number and expiry date

stated "Your payment has been declined as your card expiry date was not valid". Educating a user/attacker about the incorrect card data element.

As we will see in Chapter 6, VISA authorization network does not detect multiple invalid attempts when distributed across multiple payment gateways, attacker has countless attempts to guess expiry date and all the other card data required to make an online payment. Using the AuthRes codes an attacker could easily be able to obtain card data fields for all Visa cards.

In chapter 6, we will further decode and perform a detailed analysis of authorisation response codes of hundreds of online merchant websites from a customer checkout page where we study the authorisation-only CNP payment protocols. The level of information encoded in these authorisation response code allowed us to formulate an attack we call 'Distributed Guessing Attack' which will allow fraudsters to learn all the payment card details belonging to the target cardholder.

Website bot. We designed a website bot – a web crawler that automates the vulnerability assessment process on selected group of websites. Website bot can perform the following three functions:

- generates and verifies the validity of payment card numbers
- automates repeated filling of payment card data fields
- determines the number of invalid attempts allowed

Generates and verifies the validity of payment card numbers. The website bot is programmed to use Luhn's check algorithm to generate a database random payment card number. Although widely available, the Luhn's algorithm is not well known. The algorithm provides a basic check of whether a payment card number is valid. The purpose of the algorithm is not to authorise a payment card transaction but to provide a first line of defence against misspellings and wrong input. The algorithm works as follows:

Multiply every digit in the credit card number by its weight. The weights are either 1 or 2 depending on the position of the digit. If the card has an even number of digits, the first digit has a weight of 2, otherwise it has a weight of 1. Weights alternate: 1,2,1,2....., or 2,1,2,1… If the weighted value of a digit exceeds 9, subtract 9. Add together the weighted values of all digits, modulo 10. If the credit card number is correct, the result must be 0. To generate payment card numbers belonging to specific bank and brand (Visa, MasterCard, Debit or Credit), we provide BIN as an input to the website bot.

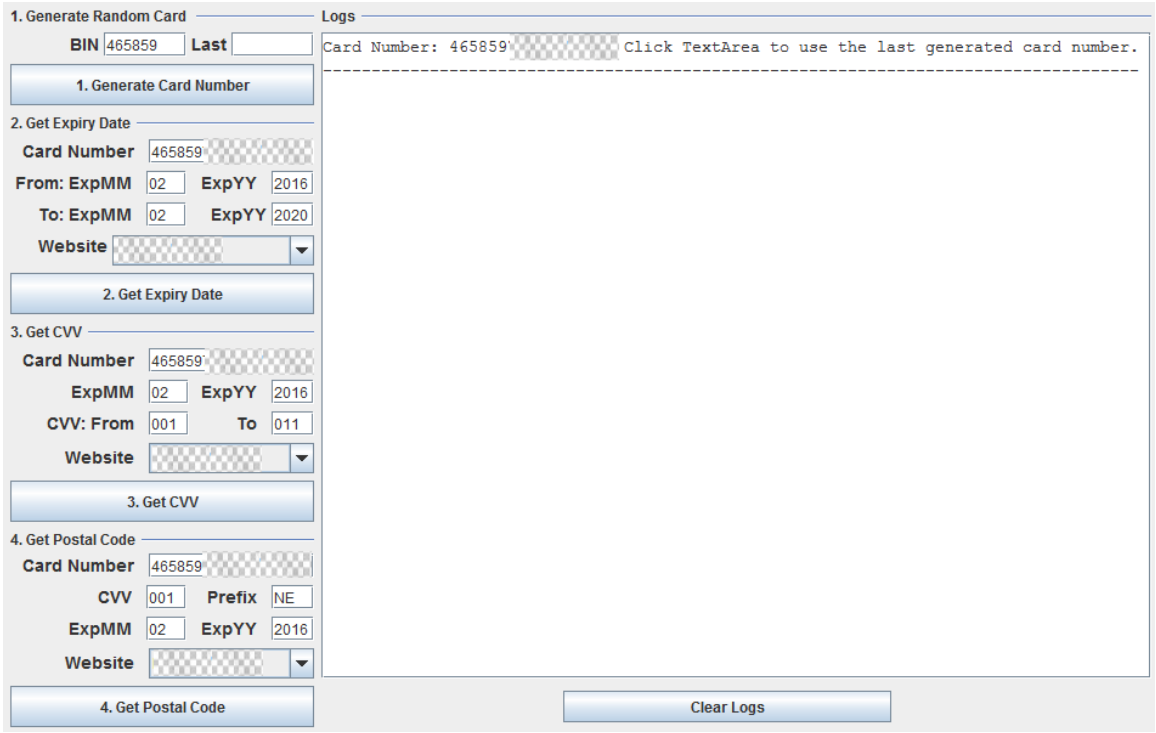

Figure 20 - A screenshot of the website bot generating Barclay's PLC bank's Visa debit card numbers

```
    public static boolean luhnTest(String number) {
        int s1 = 0, s2 = 0;
        String reverse = new StringBuffer(number).reverse().toString();
        for (int i = 0; i < reverse.length(); i++) {
            int digit = Character.digit(reverse.charAt(i), 10);
            if (i % 2 == 0) {//this is for odd digits, they are 1-indexed in the algorithm
                s1 += digit;
            } else {//add 2 * digit for 0-4, add 2 * digit - 9 for 5-9
                s2 += 2 * digit;
                if (digit >= 5) {
                    s2 -= 9;
                }
            }
        }
        return (s1 + s2) % 10 == 0;
    }
}
```

Figure 20 shows screenshot of the website bot generating Barclay's PLC bank's Visa debit card number using the Luhn's check logic shown in the above text box. The validity of card numbers generated by the website bot can be established by using authorisation response codes i.e. when a transaction is attempted using these card number the authorisation response for the card issuing banks will indicate whether the transaction is attempted with a valid card number or not.

Automates repeated filling of payment card data fields. The website bot can simplify manual filling of payment card data on selected websites by automating the process. The website bot uses selenium web drivers to select card input fields on the checkout page of the selected website and populates the selected input field with the given value. For a card number, the website bot can initiate transaction on a range of expiry dates, CVV2's and address fields.

	Website Name	CardNumber	Expiry Date	CVV2	Postal Code	Number of	Leak	Comments	Wallet Support
3	Amazon	YES	YES	NO	NO	Unlimited	Expiry Date		..
4	Ebay (Paypal)	YES	YES	YES	YES	10	Postal Code		Paypal
5	Netflix	YES	YES	YES	NO	10	CVV2	Verified Number of Attempts	Paypal
6	Walmart	YES	YES	YES	NO	10	CVV2	Verified Number of Attempts	Paypal
7	Etsy	YES	YES	YES	NO	20	CVV2	PayPal/ Retry after 1 hour	paypal
8	Bestbuy	YES	YES	YES	No	10	CVV2	Verified Number of Attempts	Visa checkout
9	Ikea	YES	YES	YES	YES	4	CVV2/Postalcode	Verified by (3D secure) / blocks card after 4 attempts	..
10	Target	-	-	-	-	-	-	-	paypal
11	homedepot	-	-	-	-	-	-	-	
12	Steampowered	YES	YES	YES	NO	10	CVV2	PayPal/ Retry after few hours	Paypal
13	Newegg	YES	YES	YES	NO	4	CVV2/PostalCode	Verified by (3D secure) / blocks card after 4 attempts	Amex express checkout/Google checkout/Visa checkout/MasterPass/Paypal
14	Macys	YES	YES	YES	NO	20	CVV2	Verified Number of Attempts	Paypal
15	Lowes	YES	YES	YES	NO	5	CVV2	Verified Number of Attempts	..
16	Nordstrom	YES	YES	YES	NO	20	CVV2	Verified Number of Attempts	..
17	Kohls	YES	YES	YES	NO	10	CVV2	Blocks IP after 10 attempts for some time	Visa Checkout
18	Gap	YES	YES	YES	NO	4	CVV2/PostalCode	Verified by (3D secure) / blocks card after 4 attempts	Visa Checkout
19	Costco	YES	YES	YES	NO	4	CVV2/PostalCode	Verified by (3D secure) / blocks card after 4 attempts	..
20	Hm	YES	YES	YES	NO	4	CVV2/PostalCode	Verified by (3D secure) / blocks card after 4 attempts	..
21	Sears	YES	YES	YES	NO	10	CVV2	Verified Number of Attempts	Paypal
22	6pm	YES	YES	NO	NO	10	Expiry Date	Verified Number of Attempts	..
23	Nike	YES	YES	YES	NO	20	CVV2	Verified Number of Attempts	Paypal
24	Bodybuilding	YES	YES	YES	NO	20	CVV2	Verified Number of Attempts	Paypal
25	Overstock	YES	YES	YES	NO	4	CVV2/Postalcode	Verified by (3D secure) / blocks card after 4 attempts	Paypal/bitcoin
26	Staples	YES	YES	YES	NO	25	CVV2	Verified Number of Attempts	..
27	Bhphotovideo	YES	YES	YES	no	10	CVV2	Verified Number of Attempts	Paypal/Google wallet
28	Groupon	YES	YES	YES	NO	25	CVV2	Verified Number of Attempts	..
29	Forever21	YES	YES	YES	NO	4	CVV2/PostalCode	Verified by (3D secure) / blocks card after 4 attempts	Paypal
30	Ticketmaster	YES	YES	YES	NO	10	CVV2	Verified Number of Attempts	..
31	Jcpenney	YES	YES	YES	NO	10	CVV2	Verified Number of Attempts	..
32	Zappos	YES	YES	YES	NO	10	CVV2	Verified Number of Attempts	..
33	Sky	YES	YES	YES	NO	10	CVV2	Verified Number of Attempts	..

Figure 21 - An example table comparing the security features of each merchant website

As we will see in Chapter 6, not all the four card data fields (card number, expiry data, CVV2 and address) are validated by each online merchant. The website bot determines what card data fields are used by the merchant website to accept a transaction.

Determine the number of invalid attempts allowed. One of our experiments on the online payment system involved determining the number of incorrect attempts allowed on each merchant website. For a given card number, the website bot initiates repeated transaction on each merchant website with random card data fields until the merchant website stops the website bot from making further transactions.

Once the details about websites and their checkout systems were captured, we maintained a table that compares the security features on each of the 400 online merchant websites. Figure 21 shows an example table comparing the security features of each merchant website. It can be seen from the first entry of the table that 'Amazon' web store requests card number and expiry data from their customers but does not request CVV2 and Postal Code. Similarly, for entry two the Ebay merchant website supported by PayPal payment acquirer requests for four card data fields and additionally supports Pay with PayPal or PayPal wallet system. Rows with orange background indicate that the experiments were not performed on that selected website (either due to high cost of products or costs associated with cancelling an order once its placed).

4.2 Attack Landscaping

After discovering the attacks, the attack scenarios can be analysed to determine their likelihood. For our research we identify the following stages for attack landscaping.

Representativeness. The first step in determining the feasibility and magnitude of the attack is to identify a representative instance of the attack landscape as it can be found in actual systems. For instance, if the attack considers online payments, the obvious way to proceed is to select the most

popular websites from the Alexa listing as a representative sample of online merchants. For each merchant in our representation sample we demonstrate the feasibility of the attack by practically exploiting the attack using our real test card and test accounts.

Given a representative landscape, the study of the attack can then go in different directions. In some cases, **the existence** of the vulnerability needs to be demonstrated [3][51–54], in some cases the practical **feasibility** of the attack needs to be demonstrated [26][35][54–59] (our research on authorisation-only protocol required distributed attacks not to be spotted by the credit card payment acquirers and also required enough differences in the information requested for online purchases).

The next stage in the attack landscaping exercise that we follow is to determine **the magnitude** of the problem. For instance, if the cause of the attack is outdated software [77] or insufficient patching on checkout system provided a payment acquirer or is a systemic problem. Chapter 6, Section 6.7 and Chapter 8 demonstrates the attack landscaping exercise we performed on the online CNP payment system.

Finally, **the disclosure** exercise takes place, which is a closely monitored process of disclosing the vulnerability/attack and tracking the actions taken by various stakeholders.

In addition to these primary five stages, we also look at the bigger picture of the landscape by considering various stakeholders involved:

- **System Owners:** this could be the developer, maintainer, seller or whoever in charge of the system. They have an ability to make a modification to the affected system, for example by issuing patches to address known vulnerabilities.
- **Researchers:** conduct a systematic analysis of the system, or they may accidentally stumble upon some problems when using the system that they believe might result in a vulnerability. They cannot fix any vulnerability themselves, so they need to get in touch with the system owner to pass their findings for the latter to take action on.
- **Attackers:** When an attacker finds out about a vulnerability, they will try to exploit if for whatever reason (financial reward, just for fun, political motives, to name but a few). We will not discuss or explore this stakeholder much further in this as there are plenty of existing literature on them already.

Each of these stakeholders has their methods – in relation to security architecture assessment (discussed in section 4.1), attack landscaping (described above) and vulnerability disclosure (discussed in the next section).

4.3 Vulnerability Disclosure

Learning from the case studies and leveraging our previous experience in performing vulnerability disclosure, in this section we look at how we communicate the vulnerabilities to the vulnerable system owners with the aim of getting the vulnerabilities efficiently fixed. We broadly categorise vulnerability disclosure into two categories: (1) Full Disclosure (publishing the vulnerabilities on the Internet without any restriction), and (2) Responsible Disclosure (all relevant stakeholders agree to allow a period of time for the loophole/vulnerability to be patched before publishing the details on the Internet).

Once we decide to notify the vulnerable stakeholders, there are several other crucial decisions to be made in order to provide a proper structure to the disclosure exercise. All of these decisions need to consider the ethics and morals of passing vulnerability information to the affected third party, and these considerations are not widely explored in the scientific community yet. There are questions that need to be addressed, such as: To whom do we disclose the vulnerability information? How should we handle the responses from the affected third party? How much information shall we divulge in the notification message?

In an approach to find a consistent and coherent procedure to achieve effective vulnerability exercise, we will discuss in the following sub-sections two key steps: disclosing the findings and following this up with post disclosure activities.

4.3.1.1 Disclosing the Findings

The following issues were considered when disclosing vulnerability findings.

Identifying whom to disclose to. It may be complicated to determine which parties to disclose the vulnerability to, and even if stakeholders have been identified, it is not always straightforward to actually contact the party. Large companies (like card issuing banks and payment acquirers) typically provide customer service contact details, but they might not be the right people to approach for vulnerability disclosure. In some cases, there might be a dedicated contact email or number for security-related issues (such as for reporting phishing and spam), but in others, there is nothing obvious as a port of call.

The timing of the disclosure. Passing the vulnerability information forward to the affected stakeholders could have helped them to put down the adversarial activities. However, not disclosing or delaying passing the vulnerability information to the affected stakeholders may benefit the research in many other ways. For example, this would allow researchers to measure the behaviour and intentions of attackers, or to study how long the attackers would take to exploit the vulnerability [77][83]. This is a key factor in determining the likelihood of an attack and will give research a prominence dimension from which cyber-security can be studied, and if not followed may scale down eminence of the research. In general, data collection strategies and studies on attacker interest and

behaviour in exploiting the vulnerability are recommended by research on vulnerability disclosures [51][52][58][59]. However, it should not be extended beyond once the conclusions are made. Researchers should avoid interference with middle-level personnel and should commute with the most relevant managers whenever available[77].

Importance of the First Notice. With our research on the CNP payment system we intended to perform multiple disclosure exercises with the hosts that remained vulnerable did not respond to our first notification. We derived two important findings.

Firstly, those contacts who accepted the vulnerability on our first notice patched their systems we found that only the earliest notice was likely to show more effect on the patching rate. Secondly, systems that did not patch or remediate after the first notification chose not to remediate and remained vulnerable.

Level of Detail of Disclosure. Disclosure reports that lack the details of compromise will have a very limited response from the notified hosts [77]. For efficient notification process, the vulnerability disclosure report in our research included

- A detailed description of every successful attack or any weakest link found during the assessment process
- Features of the attack whether if it's practical or theoretically demonstrated. Disclosure report also mentioned the cost and time required to devise the attack.
- The report clearly detailed the technical sophistication, and in detail, the tools used or are necessary to practice the attack.
- A detailed description of any publicly available information or insider helps that may be required to make the attack practice. Is there any inside information about the system needed for the attacks to be exploited?
- Samples of the defeated security devices were provided if practical and appropriate.
- A statistical summary of the level of effort made during the vulnerability assessment was detailed. The number of times the attack was successfully performed, time to develop them, time to execute them, type of defeats.

Suggested Counter Measures. The vulnerability disclosure report included not just discovering and detailing the weakness, but also suggested effective countermeasures, if practical. By providing suggested countermeasures, we will lend more credibility, but have to be careful that these potential fixes will not make the situation worse [84].

During the course of our disclosure exercise research, we had learned that the affected hosts were more amenable to deal with the vulnerability when a solution was offered along. It is essential that the recommendations for improvement should consider the business incentives and not distract the

vulnerable hosts from their core deliverables. Apparently, there are other researchers [85][86] that had expanded in detail the effects of patching when the solution was suggested. Li et al. in [86] found that the hosts were more open to fixing their system when the solutions were suggested.

Tracking the Patching Behaviour. We could collect relevant data on characteristics of the defender. Defender features might include attributes of the hosting provider (e.g., large vs small, shared vs dedicated hosting, country headquarters), site owner (company size, company vs individual, country headquarters) or associated registrar.

Handling the Response Messages. We argue that those involved in take-down should consider how to protect individuals from harm while creating an opportunity for research to advance the understanding of how to better perform take-down. Opting to keep information private can be even more dangerous than the reckless publication of information that aids attackers. The harm may be harder to observe directly (slowed take-down speed, lack of pressure to improve practices, etc.) but equally destructive.

However, not all issues addressed during vulnerability assessments are mitigated because a lot may depend on the business decisions involved. In such a case, publishing the details of a vulnerability in the public will 'name and shame' responsible parties and victims [84].

4.3.1.2 Post Vulnerability Disclosure

While most of the research we have looked into so far have only carried out the assessment until the disclosure of the vulnerability, a select few sought a further investigation into post vulnerability disclosure. Experiments carried out after the disclosure are useful to determine the adversarial attractiveness against the affected systems. Understanding the insidious tactics against the affected systems at this stage will provide an in-depth insight into the methods and psychology of the attackers. In that regard, there are three useful activities that can be performed post vulnerability disclosure:

- Selecting hosts for further analysis: this will allow the research to continue to explore other, similar systems that might be affected by the same vulnerability, and to see if they get attacked too, or if some remedial actions can effectively thwart potential attacks.
- Adversary Identification: in an ideal world, it would be desirable to be able to identify the attackers, so that they can be brought to justice. This activity will be closely related to electronic forensic investigation that law enforcement agencies or certain security companies have the capability of performing.
- Adversary Characterisation: when it is not possible to identify the attackers, it would still be useful to be able to characterise attackers so that we can understand their profile better in order to come up with a more effective countermeasure.

4.4 Conclusion

In this Chapter we constructed a framework of terminology, methods, and recommendations for the use of vulnerability assessment, attack landscaping and the associated disclosure processes as a research method.

In the next part of this PhD we will demonstrate the application of our analysis methodology, to evaluate the security of CNP payment system. In Chapter 5 and Chapter 7, we will apply the first requirement of our methodology and perform security architecture assessment of the CNP payment system. In Chapter 6 and Chapter 8 we demonstrate the application of attack landscaping and vulnerability disclosure over CNP payment system.

Chapter 5. Online Card Not Present (CNP) Payment Protocols

This chapter presents our security analysis on one of the most ignored components of the CNP payment system – the payment protocol. Although a great deal of resources has been spent on developing secure data communication through the Internet, there are plethora of confusing and competing mechanisms to handle payment information. This chapter organises the different protocols into meaningful categories and presents their strengths and weaknesses. To securely participate in the CNP payment, it is important to realise what security attributes are and are not provided by a given protocol. For example, which protocols encrypt data, which authenticate cardholders as well as servers, which handle payments, and which offer nonrepudiation? How are these security attributes provided for a given protocol? This chapter objectively answers the questions asked above.

5.1 Introduction

The CNP payment protocol defines mechanisms for transferring of funds from the customer bank account to the merchant bank account for purchases over the Internet. The protocol defines the customer data and merchant data that is needed to be included in the payment request and defines rules for the merchants to be followed while accepting the payment.

As discussed, there are broadly two fundamental protocols touted for CNP payment systems: authorisation-only and authentication-enabled CNP protocols. Authorisation-only CNP protocol is designed for providing speed and convenience to the customer at the checkout, and the payment process is straightforward, in that, the customer only must submit their payment card details during the purchase. In contrast, authentication-enabled CNP protocols adds an extra layer of user-authentication on top of the authorisation-only protocol. With authentication-enabled protocols, the card issuer requires the cardholder to prove their identity through established methods (like passwords, security questions, one-time passcodes) before accepting the transaction.

Before launching into the CNP payment protocols, one must recognise existing protocols widely in place to secure internet-based transactions. The Transport Layer Security (TLS) is a cryptographic

protocol used as a standard for securing Internet-based transactions today. Although TLS does not provide mechanisms for handling payment card data and processes involved in handling payment, it does offer confidentiality in web sessions.

With SSL/TLS, the communication between the payment processors and the customer is encrypted, so that an attacker cannot intercept the payment card details. Further, the merchant is authenticated (customer authentication is optional) to prevent spoofing attacks, where an attacker sets up a fake merchant server. Because SSL/TLS is covered in many other research projects, to keep this chapter practical, we do not provide its technical description here. For good technical coverage of SSL/TLS, see [87].

In the context of measuring the security features provided by payment protocols, it is measured along the following properties by the payment service providers:

- **Payment Data Confidentiality.** Protocols must ensure payment credentials confidentiality during the transmission and storage. The customer is an entity requiring the confidentiality and the stakeholders processing the payment are providing the confidentiality service.
- **Payment Data Integrity.** These ensure that only authorised parties (cardholder or card issuer) are able to modify the payment data when stored and transmitted.
- **Card Authentication / Machine Authentication.** These confirm the identity of the card (token) or, when appropriate, the machine (computer of the payment initiator) that is linked to the cardholder account.
- **Transaction Authentication.** These authenticate specific transactions, typically cryptographically binding a random one-time code to the data of the transaction, achieved by the card or machine signing the transaction data.
- **Second Factor User Authentication.** These are additional user authentication techniques that use information about a second authentication factor to establish the identity of the cardholder.
- **Payment Authentication.** Techniques to ensure that the origin of any payment transaction message is correctly identified, combining (i) payment card / machine authentication and (ii) Transaction Authentication
- **Payment Non-repudiation.** These ensure that the customer/cardholder cannot deny the fact that she has completed a transaction. When required, she must be able to provide a proof of the transaction. Typically, this is achieved by signing the transaction data with user specific keys.

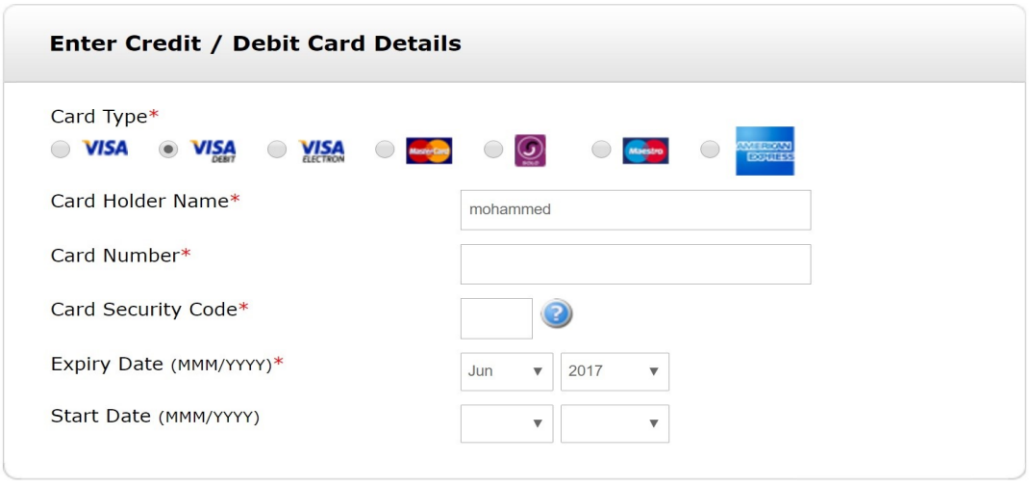

Figure 22 – An example checkout page from a merchant website supporting authorisation-only CNP protocol.

In the following sections we assess the security architecture of authorisation-only and authentication-enabled CNP payment protocols. For each protocol we describe the system either by using UML sequence diagrams or by simplified payment transition diagrams (in case there are more than five participants in the protocol). For each protocol we study, we detail the processes involved in a transaction, security features and limitations of each protocol.

5.2 Authorisation-only CNP protocol

Payment made using the authorisation-only protocol undergoes three phases: payment-authorisation, payment-clearance and payment-settlement. Each phase has its own responsibility and is discussed in detail below. An example checkout page from a merchant supporting authorisation-only CNP protocol is shown in Figure 22.

5.2.1 Payment-authorisation

Payment-authorisation is a process which verifies the customer card details provided at the checkout against the customer record file held with the card issuing bank. Having validated the card details, the card issuer checks whether the customer account has enough money to purchase the requested goods. If the customer account is in good standings, the transaction request is accepted, and a hold is put on the customer money for the purchased item and the authorisation response is forwarded to the customer through the merchant. The payment-authorisation phase involves the following steps:

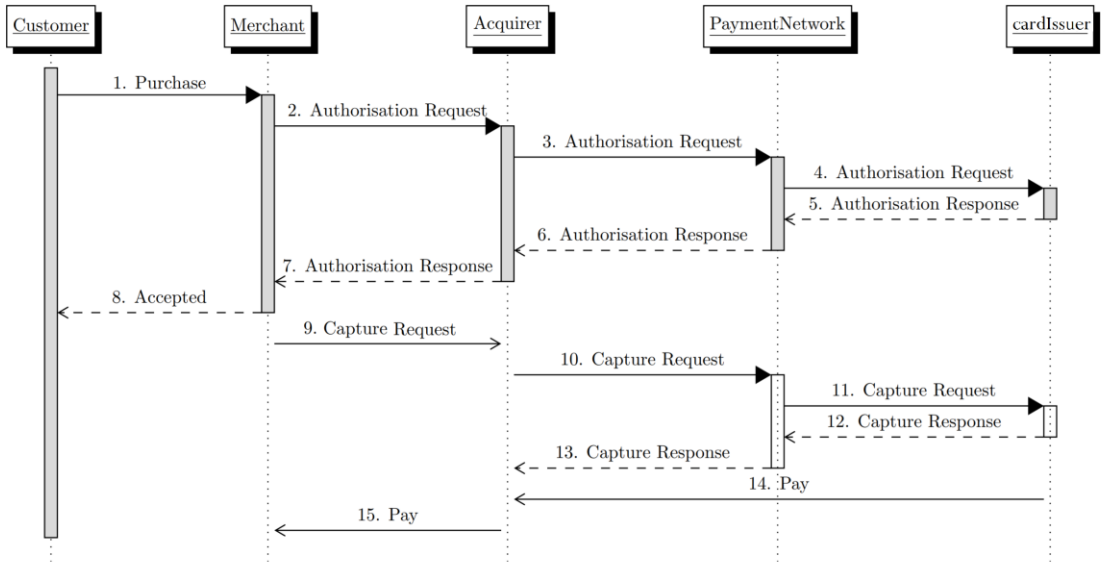

Figure 23 - Parties and processes involved in an authorisation-only CNP payment

1	The customer starts the transaction process by filling his payment card details, shipping and billing information on the checkout page provided on the merchant website. The communication channel between the customer browser and the merchant server is TLS protected and provides confidentiality and integrity to the transaction data.
2	The merchant determines the payment card type to verify if the given card type is accepted and once the card type is determined, the merchant formats an authorisation request message into a format understandable by the acquirer, shown in step 2, the authorisation request is submitted to the acquirer. It is also possible for a merchant to maintain multiple acquirers for different card brands.
3	In step 3, the acquirer adds additional information to the received authorisation request message and formats the message to the structure understandable by the card payment brand (ISO 8583 [88], Standard 70 [89] or ISO 20022 [90]). The information included by the acquiring bank includes acquirer bank details and additional information required by the card issuing banks during the payment-settlement phase. The authorisation request is forwarded to the acquirer.
4	In step 4, the card payment network determines the card issuer and forwards the payment authorisation request to it. This request is typically sent over a protected financial services network.
5	In step 5, the card issuer authorises the payment and the result of the authorisation either accepted or declined is forwarded to the card payment network. If accepted, the authorisation code is provided, and the customer money of purchased item value is put on hold by the card issuing bank. If declined, the card payment network is notified as such with a decline reason code.

6	Shown in step 6, the card payment network forwards the authorisation response to the acquirer. When required, the acquirer parses the received authorisation response into the merchant system's understandable language.
7	The parsed authorisation response is then forwarded to the merchant in step 7.
8	Finally, in step 8, the merchant completes the payment-authorisation by notifying the customer.

Payment-authorisation is the only stage visible to the cardholder. Payment-clearing and payment-settlement are the processes internal to the acquirer/acquiring bank, the payment network and the card issuing bank. Unlike the payments over the Internet, customers over Mail Order/Telephone Order (MO/TO) transactions are requested to provide their payment card details to the merchant verbally over the phone or over the email. Merchants that accept MO/TO transactions maintain a virtual platform where they enter the customer's provided card details manually and attempt making a transaction from the platform.

5.2.2 Payment-clearance

Payment-clearance is a process which provides a reconciliation to the acquirer. In this process, the customer bank bills the customer for the purchased item**.** Although the authorisation phase captures the payment on the cardholder account, the payment is not forwarded to the merchant account unless a clearance request is submitted by the merchant. The payment-clearance process has the following steps:

9	The merchant requests the shipped item value from the customer account. To do this, the merchant creates and submits a transaction clearance file to the acquirer. The acquirer collects all the clearance files from the merchant and combines them into a single batch file.
10	The acquirer sorts the received batch files and forwards them to relevant card payment network as a capture request file.
11	The capture request file from card payment network is forwarded to the card issuing bank.
12	Having received the capture request file, the card issuer verifies each record in the capture file to ensure the value are correct, and the requested amount matches to the acquirer clearance records as maintained by the card-issuing banks. If the records are found reliable, the card issuer provides a complete reconciliation to the acquirer (in the form of clearance code) for each record in the clearance file. The capture response is forwarded to the payment network.
13	The capture response is forwarded from the payment network to the acquirer. The acquirer can use the capture code provided in the capture response in case of any disputes.

5.2.3 Payment-settlement

During the payment-settlement process, the customer bank bills the customer for the purchased item and the merchant bank is credited with the value of customer purchase. This process is simplified to

exclude the message sequences passing through a payment network. The steps involved in payment settlement includes:

14	The issuer processes the batch with payment-clearance files, identifies the acquirer and sends the payment to the acquirer account.
15	The acquirer pays into the merchant account the value of cardholder's purchase. This indicates the completion of an online payment process.

Table 2 below shows an example code snippet for payment authorisation request from the merchant to the acquirer. It can be seen from the first row (marked as 1) of the table that merchant adds a unique order code to each transaction (with <order> XML tag) and mentions the type of currency in which the transaction is needed to be made. The second row (2) shows the payment details passed from the merchant to the acquirer. The amount of card data required for the transaction may vary and this depends on the contract between the merchant and the acquirer. The third row (3) shows an additional transaction data.

Table 2. An example code snippet for payment authorisation request

1	`<?xml version="1.0" encoding="UTF-8"?>`
	`<!DOCTYPE paymentService PUBLIC "-//Worldpay//DTD Worldpay PaymentService v1//EN"`
	`"http://dtd.worldpay.com/paymentService_v1.dtd">`
	`<paymentService version="1.4" merchantCode="098749304394857"> <!--Enter your own merchant code-->`
	`<submit>`
	`<order orderCode="000432"> <!--Enter a unique order code each time-->`
	`<description>TSHIRTSHOP</description> <!--Enter a description useful to you-->`
	`<amount currencyCode="GBP" exponent="2" value="50"/>`

2	`<paymentDetails>`
	`<CARD-SSL>`
	`<cardNumber>4763672350641010</cardNumber>`
	`<expiryDate><date month="01" year="2020"/></expiryDate>`
	`<cardHolderName>Mohammed</cardHolderName>`
	`<cardAddress>`
	`<address>`
	`<address1>2</address1>`
	`<address2>Sidney Grove</address2>`
	`<address3>Newcastle upon Tyne</address3>`
	`<postalCode>NE45PD</postalCode>`
	`<city>Newcastle upon Tyne</city>`
	`<state>Tyne and Wear</state>`
	`<countryCode>GB</countryCode>`
	`</address>`
	`</cardAddress>`
	`</CARD-SSL>`
	`<session shopperIPAddress="xxx.xxx.xxx.xxx" id="0215ui8ib1"/>`
	`</paymentDetails>`

3	`<shopper>`
	`<shopperEmailAddress>cccsncl@gmail.com.com</shopperEmailAddress>`
	`<browser>`
	`<acceptHeader>text/html</acceptHeader>`
	`<userAgentHeader>Mozilla/5.0 ...</userAgentHeader>`
	`</browser>`
	`</shopper>`
	`<dynamicMCC>5045</dynamicMCC> <!--The merchant category code that applies to this transaction-->`
	`<dynamicInteractionType type="ECOMMERCE"/> <!--The type of shopper interaction for this transaction-->`
	`</order>`
	`</submit>`
	`</paymentService>`

Table 3. shows an example authorisation response for an accepted transaction request from the acquiring bank to the merchant. As it can be seen, tag <orderStatus> indicates that the transaction for the card number <cardNumber> is authorised by the card issuing bank.

Table 3. An example authorisation response for an accepted transaction request

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService v1//EN"
  "http://dtd.worldpay.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="098749304394857"> <!--The merchantCode you supplied in
the order-->
  <reply>
    <orderStatus orderCode="000432"> <!--The orderCode you supplied in the order-->
      <payment>
        <paymentMethod>VISA-SSL</paymentMethod>
        <amount value="050" currencyCode="GBP" exponent="2" debitCreditIndicator="credit"/>
        <lastEvent>AUTHORISED</lastEvent>
        <AuthorisationId id="666"/>
        <balance accountType="IN_PROCESS_AUTHORISED">
          <amount value="5000" currencyCode="GBP" exponent="2" debitCreditIndicator="credit"/>
        </balance>
        <cardNumber>4763672350641010</cardNumber>
        <riskScore value="0"/>
      </payment>
    </orderStatus>
  </reply>
</paymentService>
```

Table 4 shows an example authorisation response for a transaction marked as declined/refused by the card issuer. As it can be seen, <CVCResultCode> and <AVSResultCode> represents the CVV2 and the address verification results as verified by the card issuer. The <ISO8583ReturnCode> indicates the status of a transaction response and in this case, code="41" indicates that the transaction is rejected by the card issuer because the CVV2 supplied by the customer at the checkout does not match the actual CVV2 belonging the card (‘description’ field in the <ISO8583ReturnCode> indicates the reason of decline of a transaction)

Table 4. An example authorisation response for a transaction marked declined

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE paymentService PUBLIC "-//Worldpay//DTD Worldpay PaymentService v1//EN"
  "http://dtd.worldpay.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="098749304394857"> <!--The merchantCode you supplied in
the order-->
  <reply>
    <orderStatus orderCode="000432"> <!--The orderCode you supplied in the order-->
      <payment>
        <paymentMethod>VISA-SSL</paymentMethod>
        <amount value="050" currencyCode="GBP" exponent="2" debitCreditIndicator="credit"/>
        <cardNumber>4763672350641010</cardNumber>
        <lastEvent>REFUSED</lastEvent>
        <CVCResultCode description="C"/>
        <AVSResultCode description="E"/>
        <ISO8583ReturnCode code="41" description="WRONG CVV"/>
        <riskScore value="0"/>
      </payment>
    </orderStatus>
  </reply>
</paymentService>
```

Authorisation-only CNP payment security relies upon the data entered by the cardholder at the checkout. Additionally, fraud detection filters are used by merchants, payment acquirers and card-issuing banks to detect CNP payment fraud. Most of these fraud protection filters rely on the three-

digit CVV2 code printed on the back of the card and the verification of the address information of the cardholder.

Apart from fraud detection filters, the PCI DSS specification[21] and merchant specification from card payment scheme providers (such as Visa [66] and MasterCard[67]) outline the security measures that should be employed by any merchant. The PCI DSS[21], provides a comprehensive set of rules and controls for the secure handling and storage of sensitive card data. For example Table 5 shows a PCI DSS requirement 3 [21] which states that all parties in a payment ecosystem are prohibited from storing a card's sensitive data (full track data and CVV2) even in an encrypted form. The card number may be stored; however, it should be rendered unreadable using strong cryptography. Merchants, payment acquirers, payments networks and banks involved in processing online payments are supposed to adhere to PCI DSS,

Table 5 - PCI DSS' rule on which card data may be stored

Data Element	**Storage Permitted**
Primary Account Number (PAN)	Yes
Cardholder Name	Yes
Expiration Date	Yes
Full Track Data	No
CAV2/CVC2/CVV2/CID	No

5.3 Security Limitations of Authorisation-only CNP Protocol

The convenience and simplicity with which the customer can make a purchase make authorisation-only protocol the most widely used protocol across the globe for accepting CNP payments. However, the authorisation-only protocol does not provide complete confidentiality of customer's card data leaving many opportunities for an attacker to steal the customer payment card data. Primarily, this is because the identity of the customer making a transaction is not established by the card issuer and the security of the transaction relies on the cardholder entering their static card details as requested by the merchant. The card details used for making a payment are static, and the same card details are requested by every merchant over the Internet or across MO/TO platforms. This makes the security of authorisation-only protocol distributed where each participant processing the payment possess a local copy of customer payment card data. Following are the statistics from PCI council which illustrates how the payment card data is stolen:

- 76% exploited stolen credentials including **phishing**
- 40% incorporated **malware**
- 52% involved some form of hacking including **identity theft**

- 35% involved physical attacks
- 29% leveraged social tactics

Let us discover some of these attacker tactics in more detail:

5.3.1 Phishing

Phishing might seem as a very primitive yet very effective way for fraudsters to deceive a victim. The primary aim of phishing is to get valuable private information by contacting a victim via email, text message, phone call, social media or via any other communication means. Scammers try to disguise their messages as official communication from authorities, banks or similar organisations to gain trust of their victims and derive private information from them to further use with criminal intentions. Such phishing emails can also contain a link to a fake phishing website, which is veiled to appear as a legitimate website of victim's online bank or any other organisation.

It is now a common practice by the card issuing banks to advice to its customers to check that the web merchant page address contains "https" and a lock sign on the address bar of a web browser. This is important as it indicates the legitimacy of a website.

What can make phishing even more deceiving for victim is spoofing technique. This technique involves using real e-mail or IP addresses. Fraudsters are able to mask their email address as somebody else's e-mail address and send their phishing messages from this spoofed e-mail. This technique allows to gain absolute trust of victims, who will think that the e-mail is coming from an authorised sender. Same applies to IP addresses, when fraudsters create IP packets that use someone else's IP address to appear as a legitimate host. Quite often a web site can also be spoofed in combination with a pharming method. Pharming is redirecting victim to a spoofed web site when they are clicking on a link, which was supposed to lead to a legitimate site. Such technique can easily be used to scam buyers on spoofed online merchant web sites by collecting their card information at check out.

5.3.2 Malware

In a nutshell, malware is a malicious software - computer program, which is created with fraudulent intentions. Malware brings such threats like viruses, worms, Trojan horses, ransomware and bots. Such malicious agents can attack and gain control over operating system, database, network or computer on behalf of a fraudster. They aim to collect any sensitive information of victims such as passwords, personal information, payment card detail and more. Malwares can be very advanced and can automate numerous attacks. Such fraud poses a big threat and contributes to the increasing number of financial internet crime every year.

Potentially Unwanted Programs (PUPs), which usually contain malware, is a substantial threat for internet users' private information. As a rule, victims are fooled into downloading these malicious

applications on social networks and websites where upon the PUPs will install themselves on victims' computers even without their consent. Furthermore, they are not easy to be removed from computer once installed. The PUPs and their purposes can be diverse. They may gain control over the computer and lead to disabling security software and infecting computer with even more malware. With the use of browser parasite criminals can manipulate victim's browser. Such PUP might not only track visited pages, but if very advanced might also steal passwords and any sensitive information (including card data) that was input in the browser. Another dangerous PUP is a spyware, which is designed specifically to collect private information like e-mails and messages, record keystrokes and take screenshots to steal any valuable data. PUPs are very effective criminal tools for identity theft.

5.3.3 Identity Theft

Payment card details for authorisation-only systems are usernames and passwords which are shared with every merchant on the Internet that customers do business with. When a customer makes a transaction, he has no assurance that the payment card details will be safeguarded by the merchant and the security of customer's payment card details is only as secure as the merchant server. Consider a scenario where a merchant either an Internet-based or MO/TO based, who plays a rogue. All the customer payment card data is visible to the rogue merchant who has the freedom to use to across myriad of CNP platforms. This makes the system more vulnerable to identity theft attacks where rogue merchants, if any, would misuse the customer payment card details.

Once payment card information has been collected with the use of identity theft techniques or purchased from another criminal group, then fraudsters move on to committing online payment fraud (trial stage) by simply making purchases on online stores. One of variations of online payment fraud is account takeover. The technique involves fraudster gaining control over victim's account in online payment website or even in online banking system. Account numbers and passwords are obtained with the use of any of the identity theft techniques. Then a fraudster will change the personal information of the victim in the account such as e-mail address and home address and will use the provided card details for making purchases. After the unauthorised purchases are done it might take some time for the victim to find it out and identify the account takeover.

5.3.4 No Real-Time Exchange of Fraud Information

The other limitation with authorisation-only CNP payment system is that there is no real-time exchange of fraud information. Fraud detection in the CNP payment system can either be detected by the merchant or the card issuers. Additionally, if customers find any illicit activity on their account or find any unknown transaction the fraud is then reported to the card issuing banks.

Fraud detection by merchants. If a merchant detects that it is a fraudulent transaction, the information is kept locally with the merchant unless the acquirer is providing the fraud monitoring

service. The fraud data with acquirer is only available to all the subscribed merchants, and this comes at an extra cost from the acquirer to the merchants.

Fraud detection by customers and card issuers. If the customer discovers any false statement on their bank account, the customer files a complaint with the card-issuing banks which further investigates if the card details have been misused after being stolen. If found stolen and misused, the card is blocked by the card issuer. It is important to note the time it may take for a victim to notice an unknown transaction in their bank statement and report it and by that time the fraudster has already completed an illicit transaction.

Once fraud is reported by the customer, the card issuer generates a record in the TC-40 (for Visa and MasterCard) report file [91]. The TC-40 is an electronic file that audits all the fraud identified by the card issuing banks, and the card number is recorded on to the file. The file is distributed across the acquirers typically once in a week who can update their fraud database. If a transaction is attempted using a card found in TC-40, the attempt is rejected by the acquirer and is not sent to the issuer for authorisation.

To summarise, authorisation-only protocol provides a method for securely handling the payment for goods purchased through the Internet. This protocol also goes beyond securing the Web session; rather, the protocol also provides assurance to merchants and consumers that the payment is authorised by a payment card issuer, that the merchant's bank will be paid, that the order was correctly received by the merchant.

5.4 Authentication-Enabled CNP Protocols

After the problems associated with authorisation-only protocols were realised, the card payments networks started working on a more secure version of online CNP payment protocol which came in the form of authenticating a cardholder for each transaction. Cardholder authentication or simply authentication is a process which verifies the identity of a cardholder. Authentication-enabled CNP protocols require the customers making a transaction at the checkout to establish their identity with the card issuer before the transaction is being accepted for authorisation. Once the customer is authenticated, the merchant is provided with an authentication code which can later be submitted to claim authorisation and settlement. Cardholder authentication in CNP payments is achieved by using either of the two protocols: 3D Secure 1.0 (3DS 1.0) and 3D Secure 2.0 (3DS 2.0).

5.4.1 3D Secure 1.0 (3DS 1.0)

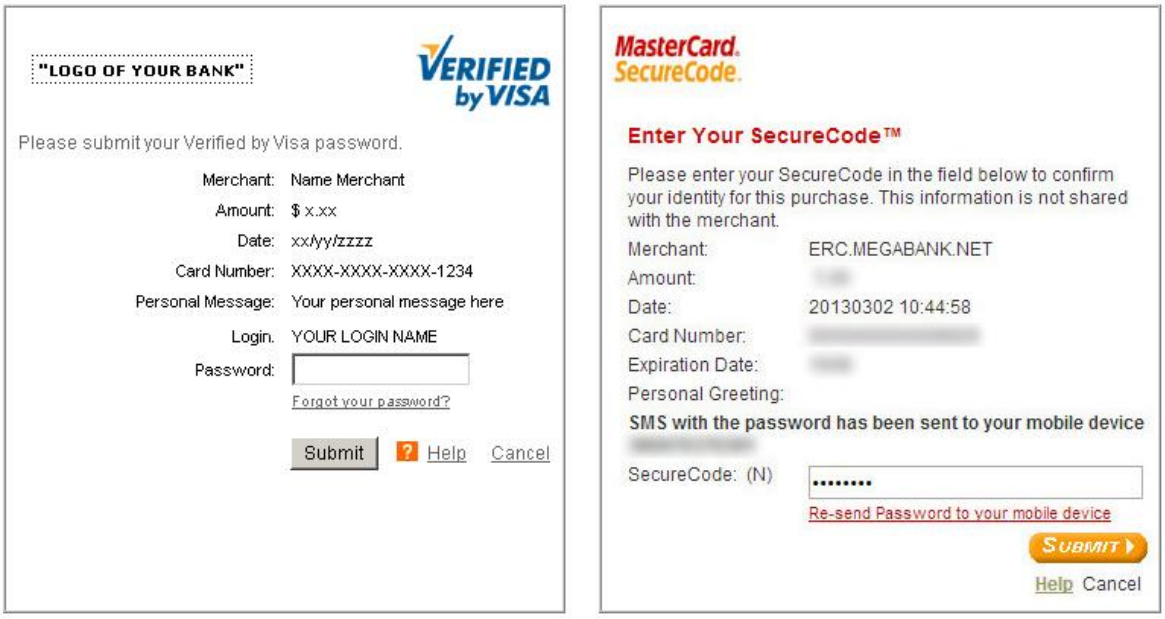

Figure 24 - 3D Secure 1.0 checkout windows from Visa and MasterCard

3DS 1.0 aims to provide more security as opposed to earlier "authorisation-only" systems which relied on static card data to verify the customers at the checkout. Along with the card details, 3DS 1.0 requires the customer to enter password for each online payment transaction. 3DS 1.0 got the economics right, in that, the fraud liability was shifted to the card issuing banks. Where-as, the merchants with authorisation-only systems were forced to bear the losses associated with fraud. In the year 2001, Visa introduced the 3DS 1.0 protocol which was later adopted by MasterCard. Each card payment brand has their alias name for the technology. Visa calls it as Verified by Visa, MasterCard refers to the protocol as SecureCode, and American Express referred to their implementation of the technology as SafeKey. 3DS 1.0 transaction process involves a number of parties, each with different responsibilities. Figure 24 - 3D Secure 1.0 checkout windows from Visa and MasterCard. The display of these screens on the merchant website indicates that the merchant supports 3DS 1.0 protocol on their checkout system. It can be observed from MasterCard SecureCode screen that 3DS 1.0 can be extended to support Out-Of-Band authentication where a password is sent to the customer to their registered mobile devices. We will further discuss OOB authentication in Chapter 7 where we discuss the authentication methods supported by 3D Secure 2.0.

Figure 25 on the next page shows actions and parties involved in a 3DS 1.0 payment process. As discussed, a 3DS 1.0 enabled transaction adds an extra phase of user-authentication over the authorisation-only protocol. Therefore, the phases of a 3DS 1.0 enabled transaction includes: user-authentication, payment authorisation, payment clearance and payment settlement.

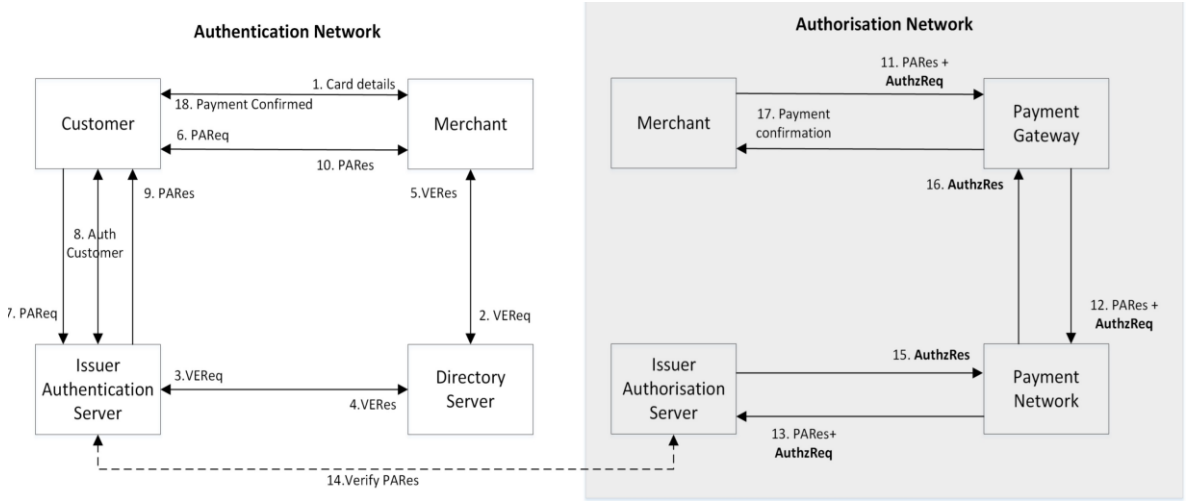

Figure 25 - Actions and parties involved in a 3DS 1.0 payment process

5.4.1.1 User-authentication Phase

1)	The first step involves the customer/cardholder entering their payment card details on the payment page of the online merchant website. The merchant controls which data fields are used to authorise the payment. At this step, merchant has enough card data to start transaction process.
2)	The merchant server hosts Merchant Plugin (MPI): a software module that process payer authentication and authorisation messages. To ensure that the card number is enrolled for authentication, MPI builds a Verification Request (VEReq) message and forwards it to the Directory Server.
3)	The Directory Server (DS) maintains a list of all card issuing banks that support payer authentication schemes. If the card number is in the participating range, DS requests Issuers Authentication Server to determine whether the card number is enrolled for authentication.
4)	Issuer Authentication Server (IAS) serves two integral functions: Firstly, it will validate the card number in VEReq is enrolled for authentication. Secondly, it stores all the customer authentication data (passwords, security questions or PIN) to facilitate the actual cardholder authentication process. IAS responds to DS, indicating whether authentication is available for the card number. If the cardholder is not enrolled in the system, the card issuing bank connects to the cardholder and registers the cardholder with the 3DS 1.0 system. For this transaction, we assume the cardholder is enrolled with 3DS 1.0 and the authentication is available.
5)	The MPI receives a Verification Response (VERes) from the DS.
6)	MPI combines the transaction data and customer card details in a single Payer Authentication Request (PAReq) to IAS via the customer's browser.

7)	IAS receives PAReq from the MPI. IAS maintain a history list of all previous transactions for the given card number. The history list can be used as records in case of any disputed transactions.
8)	IAS authenticates customer as appropriate owner for the card number, then formats the Payer Authentication Response (PARes) message with a cryptographic hash (CAVV) which is later used by authorisation server to verify the integrity of a message.
9)	IAS returns PARes to the MPI via shopper's browser. IAS maintains a copy of Payer Authentication Response for the transaction
10)	The MPI receives PARes. Merchant now has all the data required to submit an authorisation request.

5.4.1.2 Authorisation Phase

11)	The MPI passes the PARes + Authorisation request (AuthzReq) to their chosen payment acquirer, which provides a service of authorising and processing the merchant's payment request. The payment acquirer on behalf of the merchant can also implement additional security filters at this point.
12)	The Payment acquirer then connects the merchant to the card payment network to request payment from the customer's bank account held at the card issuing bank. The payment networks provide the link between payment acquirers and the thousands of card-issuing banks.
13)	The payment network now forwards the PARes + Authz request to the Issuer Authorisation Server. The Issuer Authorisation Server holds the customer's bank account and makes the final approval of the payment. The issuer has access to information such as account balance, customer name and full address not visible to the rest of the payment network.
14)	In this step, the Issuer Authorisation Server checks the cryptographic hash received in PARes matches with a copy stored in the IAS.
15)	The Issuer authorisation server responds payment network with the authorisation response.
17)	The payment confirmation from the payment acquirer is passed to the merchant from where it is forwarded to the customer.

The payment-clearance and payment-settlement phase of a 3DS 1.0 transaction are similar to the authorisation-only protocol and so are omitted in this section for brevity. Table 6 shows the 3DS 1.0 transaction data that is passed from the when the customer clicks the "pay" button on the 3DS 1.0 enabled merchant website. The merchant collects this data to frame a PaReq message and the transaction is processed as described in the authentication phase.

Table 6. 3DS 1.0 transaction data passing from merchant to acquirer

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE paymentService PUBLIC "-//Worldpay//DTD Worldpay PaymentService v1//EN"
  "http://dtd.worldpay.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="098749304394857<!--Enter your own merchant code-->
<submit>
  <order orderCode="9842408"> <!--Enter a unique order code each time-->
    <description>YOUR DESCRIPTION</description>
    <amount value="2000" currencyCode="EUR" exponent="2"/>
    <orderContent>
      <![CDATA[]]>
    </orderContent>
    <paymentDetails>
      <CARD-SSL>
        <cardNumber>4444333322221111</cardNumber><!-This is a test card number-->
        <expiryDate>
          <date month="01" year="2020"/>
</expiryDate>
 <cardHolderName>3D</cardHolderName>
  <cardAddress>
    <address>
            <address1>Worldpay</address1>
            <address2>270-289 The Science Park</address2>
            <address3>Milton Road</address3>
            <postalCode>CB4 0WE</postalCode>
            <city>Cambridge</city>
            <countryCode>GB</countryCode>
          </address>
        </cardAddress>
      </CARD-SSL>
      <session shopperIPAddress="127.0.0.1" id="SESSION_ID"/> <!--Session id must be unique for each
order-->
    </paymentDetails>
    <shopper>
      <shopperEmailAddress>jshopper@myprovider.com</shopperEmailAddress>
      <browser>
        <acceptHeader>text/html</acceptHeader>
        <userAgentHeader>Mozilla/5.0 ...</userAgentHeader>
      </browser>
    </shopper>
  </order>
</submit>
</paymentService>
```

Table 7 shows an example PAReq message as generated by MPI to request user-authentication via 3DS 1.0 enabled IAS. The PAReq message is forwarded to the IAS (<issuerURL> shown in Table 7 with a grey background represents the IAS URL) via the customer browser.

Table 7. An example PaReq message

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService v1//EN"
  "http://dtd.worldpay.com/paymentService v1.dtd">
<paymentService version="1.4" merchantCode="098749304394857"> <!--The merchantCode you supplied in
the order-->
  <reply>
    <orderStatus orderCode="ExampleOrder1"> <!--The orderCode you supplied in the order-->
      <requestInfo>
        <request3DSecure> <!--PaRequest must be supplied as-is. Do not truncate-->
          <paRequest>eJxVUsFuwjAM/ZWK80aSUgpFJogNpHEo2hjTzlVr0Uq0KUkYsK+fUwpl72Q/2y/Jc2B2LvfeD2pTqGr
aE33e82YStrlGXHxietQoIUZjkh16RUYdQTge8DAII3846kl4n2/wIKFVkCTQ94HdUhrVaZ5UVkKSHl5Wayk6AGs5KFGvFo8lh+e
u71qHOjHmpHQmhT8IhuFoDOxOQZWUKL+V3md1cvEWWCqTqxpYw0OqjpXVFzn0aeiWwFHvZW5tPWGsUtqqylilEZjjgXV3fz+6yJD
Ouchk/DsX8XZ3Wi+WPP6YP2IKzHVAlliUPhchHwnf4+FkGEz8CFjDQ1K6C8jl18YTT5yTD1cCanfO/JoIV3gkgJahsUovMnIvv2e
A51pVSB1k/D2GDE0qLRrrkT2o6WxHAOve8vrmtpJasjYgDAg+4baapuDEC7JKRDxs1F0CzI2ydvWs/R4U/fs2f8B1wXg=</paReq
uest>
          <issuerURL>
```

```
            <![CDATA[https://secure-
test.worldpay.com/jsp/test/shopper/ThreeDResponseSimulator.jsp]]>
          </issuerURL>
</request3DSecure>
      </requestInfo>
      <echoData>13742444099876913 95</echoData> <!--For compatibility with older integrations - can
be ignored-->
    </orderStatus>
  </reply>
</paymentService>
```

As discussed, once the card issuer authenticates the cardholder, it frames an ARes message and forwards it to the merchant MPI through the customer browser. Table 8 shows an example PaRes message.

Table 8. An example PaRes message

```
?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE paymentService PUBLIC "-//Worldpay//DTD Worldpay PaymentService v1//EN"
  "http://dtd.worldpay.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="YOUR_MERCHANT_CODE">
  <submit>
    <order orderCode="jsxml3792597179"> <!--The order code from the first message-->
      <info3DSecure> <!--PaResponse must be supplied as-is. Do not truncate-->
        <paResponse>eJx9UlFvgjAQft+vILxrQQHBnDUsaMKDxkyW7JXABZsIuBaM+/e7VicuZvte2n739e67a2F5qY/WGaUS
bbOw3bFjW9gUbSmaamG/Z+tRaC85ZAeJmOyx6CVy2KBSeYWWKOmGF/hR6LmzwA9dm8MufkP1HBkrUTVYkuBWi1Op8QTYz5GSyuKQ
Nx2HvPh8TbfcHQDsxkGNMk0eQxqjQXeNAxuy7Xq9U2T7Ikqexn9iAUwroMw75BPHDZzA9S0nmnvRfBoAMzycdLq4bnvK7ToOsEcC
aD6SxvfFo1kI7H4CvJzaBklBHd/3wAZvu3jLPcKUMCFcu9EsZB8cOlE/efKmwAwPqsu7XvE0WW2zdJ2uEmA3Cor8fOabf5s2EsBC
cMcnd7SaW/GxaqXoDrX2/JsApj0x89Qc9vSyVEwiM3M3X0NLHr/MyzdYf7wy</paResponse>
      </info3DSecure>
      <session id="ssn792597179"/> <!--The session id in the first message-->
    </order>
  </submit>
</paymentService>
```

5.4.1.3 3D Secure 1.0 Challenges

Although 3DS 1.0 helps to reduce number of fraud and offers an extra layer of security for both cardholder and merchant, it does not solve all the problems. First of all, 3DS adds an extra step in the checkout process that complicates the payment process. During the checkout, the cardholder needs to enter his password in a separate pop-up window or enrol to the 3DS service if this has not been done before. This introduces a disruption on customer journey, which confuses customers and can lead to a reduction in the conversion rate. Additionally, 3DS 1.0 supports only browser-based purchases while in-app (mobile based) purchases were not supported.

When implementing 3DS 1.0, the merchant can benefit from the liability shift from merchant to the issuer in case of fraud for 3DS 1.0 authorised transactions. On the other hand, the merchant needs to invest in additional components which may even increase the scope of their PCI-DSS certification. Moreover, the risk of dropping the conversion rate due to complication of the checkout process holds some merchants from implementing 3DS.

Lastly, 3DS 1.0 can potentially be vulnerable to phishing and “man-in-the-middle” attacks. This is caused by the redirection to another URL for the 3DS process. An attacker can generate a similar

looking pop-up window that can steal personal information or download malicious content on the computer of the customer.

5.4.2 3D Secure 2.0 (3DS 2.0)

As online fraud rates were steadily increasing, the European Commission (EC), in the year 2015 emanated with their interest to secure the card payments. The EC published the Payment Services Directive 2015/2366 (PSDII) [92], a regulatory standard, which ruled card issuing banks within Europe to mandate Strong Customer Authentication (SCA) for each online payment transaction. With the PSD-II, SCA is required to be carried out by combining two out of the three following elements: something you are (e.g. biometrics), something you have (e.g. a token generator) and something you know (e.g. a password). However, payment industry stakeholders expressed concerns that the methods proposed in the PSD-II for SCA ignored the objectives of user-friendliness. As proven with the adoption of 3DS1.0, more steps in the checkout process will have serious implication on the customer experience. The stakeholders rather argued that the PSD-II should exempt the compulsion of SCA on every transaction and allow card issuing banks freedom to perform selective authentication through Transaction Risk Assessments (TRAs). For card issuing banks, TRA prompted SCA only on payment transactions categorized as high risk of fraud. After a long negotiation of almost six months with over 200 payment industry stakeholders, the SCA requirements of PSD-II were replaced with Transaction Risk Assessment.

With 3DS 2.0, the card issuing banks perform TRA and authenticate cardholders using either of the two schemes: Challenged and Frictionless authentication. Challenged authentication which is typically for a purchase with a high risk of fraud will have the card issuers prompting authentication challenge to cardholders (through software tokens like one-time pass-codes (OTP) or security questions). Schemes of user-authentication with OTP's and security questions, are still vulnerable to real-time Manin-the-Middle (MitM) attacks. As demonstrated by Drimer et. al.[3] and RedTeam researchers [62], [63], attackers can initiate transaction with card issuing banks and synchronously request the cardholder to enter their one-time passcode information on phishing websites. However, real-time MitM kind of attacks would require a real time co-ordination between the attacker, the card-issuing bank and the cardholder.

The payment networks and card-issuing banks believes that 3DS 2.0 is a better solution to combat online payment fraud because as with the data provided by the payment networks [40][41] more than 80% of the CNP transactions are processed through frictionless authentication which does not require any OTP yet provides a secure way of making purchase online. With Frictionless authentication, the customer will not be bothered to authenticate themselves in the checkout flow by inputting the information, rather, the card issuer will lean on "some" data from the customer machine to provide frictionless authentication. However, the 3DS 2.0 data used by the card issuer for frictionless

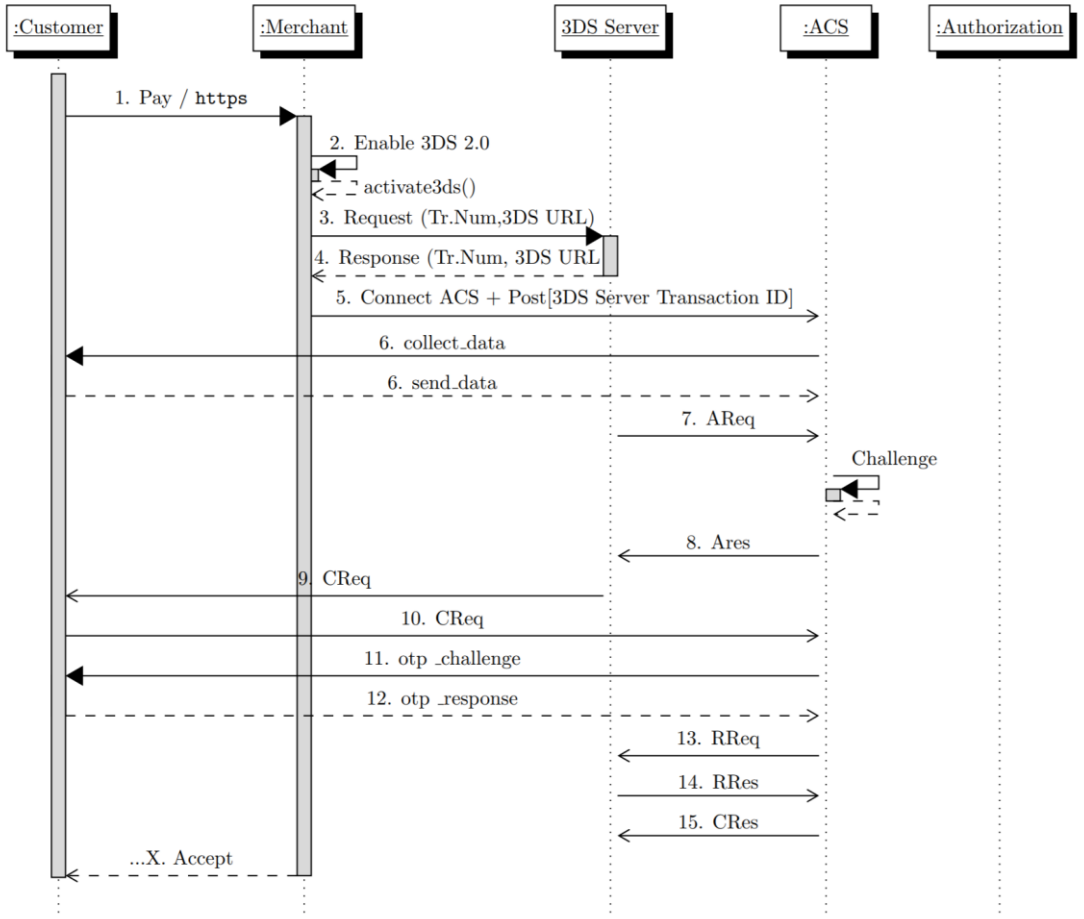

Figure 26 - Parties and processes in 3DS 2.0 transaction

authentication is not defined in either the PSD II technical standard or the EMV 3DS 2.0 specifications. We will learn more about 3DS 2.0 frictionless authentication in Chapter 7 but what follows is the technical description of 3DS 2.0 challenged authentication as defined by the EMV 3DS 2.0 specifications.

A 3DS 2.0 challenged transaction can be broadly classified into four phases based on the order in which they occur. They are: User-authentication, Payment-authorisation, Payment-clearance and Payment-settlement. As we are primarily dealing with user-authentication and payment-authorisation, for brevity, we consider the other two payment process as out of scope of this section. Figure 26 - Parties and processes in 3DS 2.0 transaction

5.4.2.1 User-authentication phase

1)	Figure 26 shows parties and process involved in a 3DS 2.0 transaction. After filling the checkout page and as the 'Pay' button is clicked, the payment card details along with customer browser's (B1's) HTTP headers and merchant session cookies are posted to the merchant.
2)	The merchant decides to have user-authentication enabled for this transaction. The merchant enables 3DS 2.0 protocol.

3)	For the payment card provided at the checkout, the merchant in this step requests the ACS version number, 3DS Method URL (address of the ACS) and a unique 3DS Server Transaction ID from the 3DS Server.
4)	The 3DS Server responds back to the merchant with the ACS version number, 3DS Method URL and a unique 3DS Server Transaction ID. The merchant sends the ACS version number and 3DS Method URL to B1 for routing to the ACS.
5)	In step 5, B1 is connected to card issuer ACS, and this connection is made via the 3DS Method URL received from the previous step. The 3DS Server Transaction ID along with B1's HTTP headers are passed to the ACS in this connection.
6)	The ACS can use this link to collect data from the customer machine. However, "The manner in which the issuer obtains the device information and which information is gathered is outside the scope the specification" (section 3.3, step 4 of 3DS 2.0 specifications [75]).
7)	The 3DS Server collects the necessary data from the merchant and frames an AReq message which is forwarded to the ACS.
8)	The Issuer performs Risk-Based Authentication (RBA) AReq and decides to have a challenged authentication for this transaction. Therefore, the issuer through Authentication Response (ARes) message responds back the 3DS Server indicating that the challenged authentication is required to further process the transaction. If the customer was authenticated using Frictionless authentication, the ARes will indicate a successful user-authentication.
9)	The 3DS Server initiates a Challenge Request message (CReq) and posts it to the customer browser to redirect it to the ACS
10)	The CReq travels through the customer browser to the ACS. The link to CReq is achieved by connecting the customer session to the issuer using a URL that the merchant received in the ARes.
11)	The issuer having received a CReq message sends a challenge user interface (UI) to the customer browser. This UI is an interaction platform where the issuer can interact with the customer to obtain challenge responses. At this point, the card issuer prompts authentication challenge to the cardholders (through software tokens like One-Time Passcodes (OTP) or biometrics).
12)	Once successfully authenticated, the issuer determines the customer as appropriate owner of the payment card.
13)	The issuer formats the Results Request (RReq) message with a cryptographic hash which is forwarded to the 3DS Server. The RReq and the hash is later used by the Authorisation network to verify the integrity of authentication messages.

14	To acknowledge the receipt of the RReq, the 3DS server prepares the Results Response (RRes) and forwards it to the ACS.
15	Finally, the issuer formats the Challenge Response (CRes) message and shuttles it to the 3DS Server which later forwards the CRes to the merchant (not shown in figure) through the link it received in the CReq message. The CRes indicates the completion of challenged authentication for the cardholder.

5.4.2.2 Payment-authorisation

During the payment-authorisation phase, the merchant builds the Authorisation Request message (AzReq) which includes the RReq message and a hash received in the previous phase. The AzReq is forwarded to the Issuer Authorisation Server which checks the correctness of RReq message with a copy stored in the ACS. If the customer account has enough money for the item at the checkout, and if the account is in good standings, the authorisation response (AzRes) is sent back to the 3DS server indicating a confirmation of payment. The merchant and the customer are further notified of the completion of a purchase. Later the merchant and the acquirer process the relevant authorisation and settlement messages.

5.4.2.3 3DS 2.0 Advantages

The introduction of EMV 3DS 2.0 has several advantages that allows enhanced security of online CNP payment transactions while optimising the cardholder's experience. Merchants will be able to make their checkout process smoother and available through different channels/devices without compromising security. 3DS 2.0 allows the non-payment authentication flow that enables merchants to offer additional secure non-payment services. Furthermore, 3DS 2.0 allows for further development of risk-based authentication techniques for cardholder authentication. Based on their internal rules, issuers would be able for example to authorise low value transaction without additional interaction with the cardholder.

Dismissing of the 3DS 1.0 requirement to authenticate the customer in a different screen than the merchant's website will not only enhance the user experience but also reduce the chance of phishing and "man-in-the-middle" attacks. Moreover, not relying on the static password will allow the use of new authentication options such as biometrics through Out-Of-Band (OOB) or One-Time Password (OTP). 3DS 2.0 focuses on interoperability not just across various card association services but across both the desktop and mobile based platforms. Besides, the powerful driver of EMV 3DS 2.0, at least in the European Union is the Payment Service Directive 2 (PSD 2) that mandates the use of strong customer authentication for both types of payments: mobile and browser based.

Ultimately, 3DS 2.0 tends to solve multiple technical pain points of 3DS 1.0. Such as the reduction of customer confusion, making the checkout process smoother for both browser based and mobile based purchases, the introduction of a frictionless authentication flow, non-payment authentication flow and

enhanced security. However, to ensure success of 3DS 2.0 both issuers and merchants need to actively participate in the 3DS 2.0 program. Otherwise there is a risk that EMV 3DS 2.0 may share similar technical and business challenges as 3DS v1.0. To summarise, 3DS 2.0 have the following advantages over 3DS 1.0 protocol:

- Support of in-app purchases on mobile phone and other customer devices.
- Enable merchants to integrate the authentication process into their checkout experiences, for both app and browser-based implementations.
- Enable the issuing banks to perform risk-based decisions on the transaction authorisation that enables frictionless consumer authentication when the customer is not required to perform an additional authentication to the bank.
- Enables non-payment customer authentication that allows services like Identification & Verification (ID&V) for mobile wallets and secure request of tokens for card on file.

5.5 Secure Electronic Transaction (SET) Protocol

Secure electronic transaction (SET) is a secure protocol jointly developed in 1996 by MasterCard and Visa with the backing of Microsoft, Netscape, IBM, GTE, SAIC, and other companies to facilitate credit card transactions over the Internet. The objective of SET is to provide security for credit card payments as they traverse the Internet from the customer to the merchant sites and onto the processing banks. SET has been developed to secure the entire credit card payment process, including verifying that the consumer is indeed the owner of the credit card. Although Visa and MasterCard have publicly stated that the goal of proposing the SET protocol is to establish a single method for consumers and merchants to conduct payment card transactions on the Internet, acceptance of the standard has been slow.

SET was never implemented practically, it is still nonetheless, but it laid foundational requirement for a much secure CNP payment eco-system. The SET specification uses public key cryptography and digital certificates for validating all participants in the transaction. In contrast to the SSL protocol, which only provides confidentiality and integrity of credit card information while in transit, the SET protocol provides confidentiality of information, payment data integrity, user and merchant authentication, consumer nonrepudiation, and payment clearinghouses (certificates).

The major components of SET are as follows:

1. The issuer (or customer’s bank) is a financial institution that issues bankcards (credit cards or debit cards).
2. The customer (or cardholder) is an authorised user of the bankcard who is registered with SET. The customer participates in the SET transaction by use of an electronic wallet. SET calls this the cardholder wallet. This is where the customer/consumer’s credit card

information is stored in an encrypted manner. The wallet software is able to communicate and interoperate with other SET components.

3. The merchant is the seller of goods and services. The merchant uses the merchant server software to automatically process credit card authorisations and payments.
4. The payment gateway processes merchant authorisation requests and payment messages including payment instructions from cardholders. It is operated by either an acquirer or some other party that supports acquirers (for example, the merchant's bank). It interfaces with financial networks to support the capture of SET transactions.
5. One or more certification authorities' issue and verify digital certificates related to public keys of customers, merchants, and/or the acquirers or their gateways. The SET public key infrastructure proposes a top-down hierarchy of certification authorities comprising the following types:
    - *Root certification authority:* All certification paths start with this authority's public key. It is operated by an organization that the entire industry agrees to trust. The initial root key is built into the SET software with provision for replacing it in the future. It issues certificates to brand certification authorities.
    - *Brand certification authority*: These authorities are operated by different credit card brand owners like Visa and MasterCard. Each brand has considerable autonomy as to how it manages the certificate subtree rooted at it.
    - *Geo-political certification authority:* This is an optional level of certification authority. It allows a brand to distribute responsibility for managing lower level certificates across different geographic or political regions. This is to account for variations in how financial systems operate in different regions.
    - *Cardholder certification authority:* These authorities generate and distribute cardholder certificates to cardholders. Depending on brand rules, the certification authority maybe operated by an issuer or other party.
    - *Merchant certification authority:* These authorities issue certificates to merchants based on approval by an acquirer.

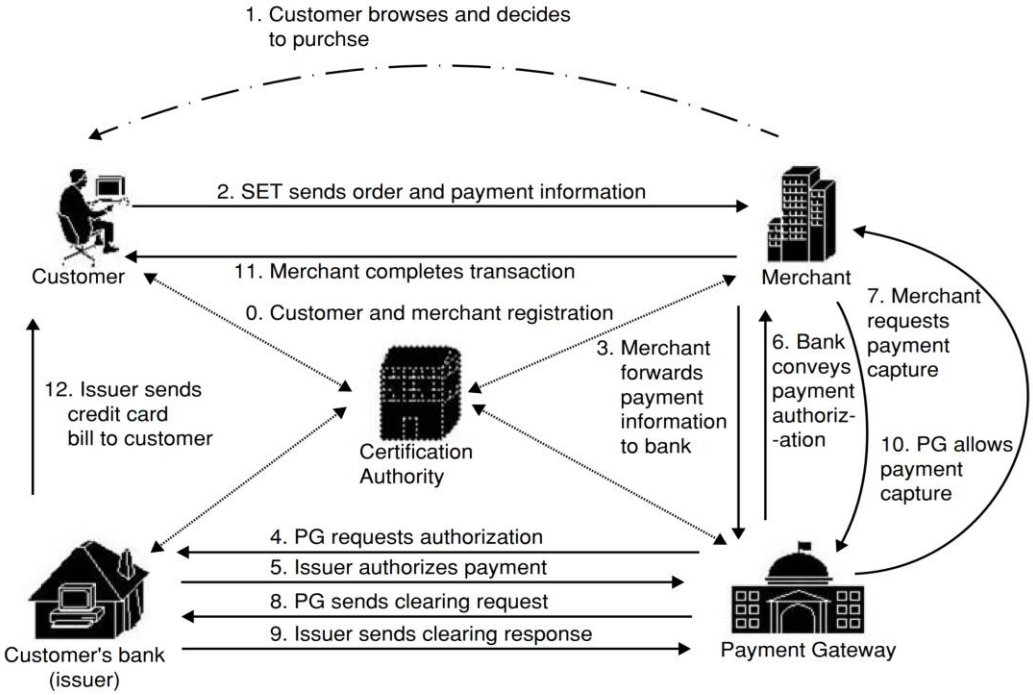

Figure 27 SET Steps

The SET infrastructure is deliberately planned not to interoperate with any other payment infrastructure other than the bankcard system. Although this may be considered a restriction of SET, it ensures that the operating organisations are not subjected to any unknown risks. The main steps of SET are described in Figure 27. The sequence can be summarized as follows:

1. *Customer and merchant registration:* In this step (messages 0 in Figure 27), the customer and the merchant acquire relevant certificates from the corresponding authorities that will allow them to participate in the transaction. This needs to be done once before any SET transaction and needs to be re-executed if the certificates expire or are revoked.
2. *Browse and negotiate purchase:* This step (message 1) proceeds in an offline manner; it allows the customer to select the product and negotiate on a price. SET is not involved in this phase.
3. *Purchase request:* Once the customer has completed the product selection process, it invokes the cardholder wallet software on its machine. This is where the main SET protocol starts (message 2). The cardholder wallet initiates a SET session with the merchant server; it sends its certificate to the merchant and requests a copy of the merchant's certificate and the payment gateway's certificate. On receipt of these, the cardholder wallet creates a payment information (PI) and an order information (OI). The PI includes the cardholder's credit card account information and public key certificate, among others. The OI includes necessary information about the order. Two digests are computed, one each for PI and OI. They are concatenated and signed by the customer's private key. The PI is encrypted with the payment gateway's public key. Next, the OI and encrypted PI are together encrypted with the

merchant's public key. Finally, the entire message is signed and sent to the merchant server. The merchant server verifies the cardholder's certificate. It verifies the signature on the PI and OI. If it agrees to OI, it forwards the encrypted PI to the payment gateway for authorisation (message 3). While waiting for authorisation, the merchant prepares an order confirmation (OC), signs it with its private key, and sends it encrypted with the customer's public key.

4. *Payment authorisation:* To request payment authorisation, the merchant sends the encrypted PI (received in step 3) signed with its private key (message labeled 3) to the payment gateway. The payment gateway verifies the merchant's signature on the message, decrypts the PI with its private key, and retrieves the customer's certificate. It then verifies the relevant signature and performs an authorisation at the issuer (message 4). When the issuer authorises payment (message 5), the payment gateway generates an authorisation response (AR) and a capture token (CT). Both are signed by the payment gateway's private key and sent to the merchant encrypted with the latter's public key. The merchant, on receipt, stores the payment authorisation response and capture token for later use.
5. *Payment capture:* The merchant prepares a payment capture request with the transaction identifier from the original OI and the CT obtained earlier. It signs the capture request (CR) and sends it encrypted with the payment gateway's (PG) public key (message 7). The payment gateway verifies the capture request message and sends a clearing request message to the issuer (CLR) (message 8). When the issuer clears the payment (message 9), the payment gateway generates a capture response (CR) and sends it encrypted with the merchant's public key. The merchant stores the CR and completes the transaction with the customer.

A major advantage of using SET over the SSL protocol for electronic credit card payment is that SET does not allow the merchant to view the customer's complete credit card information, which provides an additional degree of protection. In addition, SET prevents the payment gateway and/or the issuer bank from being able to view the terms of the transaction established between the merchant and the customer. This way the privacy of the customer is also enhanced. However, SET has received a lukewarm reception in the payment industry and, so far, has not attracted a large number of merchants and consumers, though it seems to be the "best" protocol for securing payment over the otherwise unsecured Internet.

5.6 Conclusion

Ensuring the security of online CNP payment transactions is a fundamental requirement for commerce over the Internet. Lack of trust in an online store, fear of financial loss through theft of a credit card or other banking information, and other concerns over the privacy of information transmitted to and

stored by an online store are just some of the concerns faced by consumers in CNP payment transactions.

Secure communication protocols such as SSL and TLS can be used to secure Web based sessions including commercial data transactions. Although these protocols can authenticate both parties and provide privacy in the transaction, they do not provide protocols for payment. To fill this void, a number of payment systems have emerged, from authorisation-only systems to 3D Secure 2.0.

The differences between the two systems point to the advantages and disadvantages of both approaches. Authorisation-only systems provide high accountability, real-time online verification of payment for merchants, and the ability to use the existing back-end financial infrastructure for settling payments between banks. Authorisation-only systems also provide a high level of security because all transactions are traceable, and the value of the transactions never actually leaves the banks. The trade-off for consumers, however, is the lack of confidentiality and privacy provided in authorisation-only transactions. All transactions are mediated by financial institutions or a third party, so it is entirely possible that complete spending profiles for consumers can be collected and potentially used for purposes that are often equated with “big brother” intrusive oversight. Also, the payment card details are shared with each merchant that the customer shops with.

Authentication-enabled payment systems, on the other hand, offer the ‘potential’ for anonymity. Although authentication-enabled systems have numerous advantages, the usability and security risk associated with the protocol have help up widespread adoption. The combination of online verification and the ability to support small-value transactions over the internet and in specialised applications may end up being the most viable economic model for launching the widespread adoption of the protocol globally.

Chapter 6. Distributed Guessing Attack

In this chapter we present our work in conducting an in-depth investigation of authorisation-only CNP protocol and the state-of-affairs with respect to payment security. In this process, we have discovered certain weakness in the online CNP payment system which allowed us to formulate an attack we call "Distributed Guessing Attack". The attack effectively turns the process that is meant to validate the card payment details into a process which delivers to the attacker all card data required to make an online payment. The attack has a potential to subvert security filters in place to protect CNP payment system from fraud.

The chapter is divided into following sections; section 6.1 gives a brief description of the vulnerabilities and introduces the distributed guessing attack. Section 6.2 details the controls that are in place to protect the authorisation-only CNP payment from fraud. Section 6.4 will describe the techniques and tools we used for vulnerability assessment against authorisation-only protocol, section 6.6 will outline the attack scenario and experimental work carried-out to demonstrate the attack, section 6.7 will detail the attack landscape, in section 6.8 we will detail the vulnerability disclosure exercise that we initiated to disclose the vulnerabilities to affected parties and finally in section 6.9 we will discuss potential solutions to address the vulnerabilities and conclusion.

6.1 The Attack

A CNP payment is effectively a "card-not-present" credit or debit card transaction, which means the merchant cannot physically verify that the customer actually has the card. The security of CNP payment is therefore dependent upon the customer correctly entering a number of data fields – which only the valid owner of the card should know. Our experimental work has shown that many online payment systems can be subverted to generate the payment validation data fields (card number, expiry date, CVV2 and cardholder address) when these data are not known.

To obtain card details, one can use a web merchant's payment page to guess the data: the merchant's reply to a transaction attempt will state whether the guess was correct or not. The reason this attack works in practice is due to two weaknesses, each not too severe on its own, but when used together present a serious risk to the global payment system.

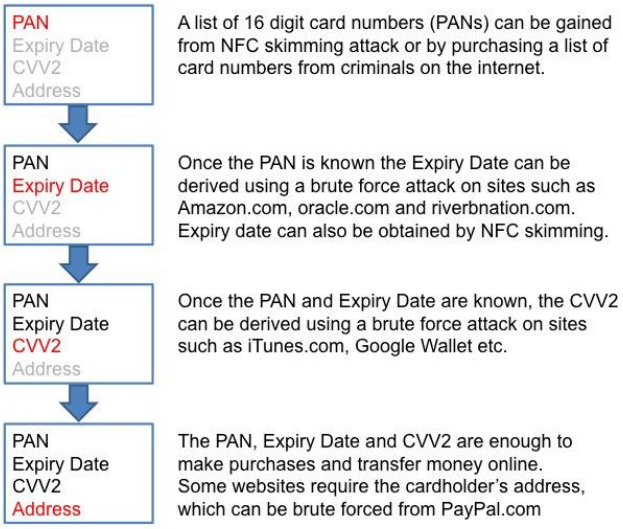

Figure 28 - Distributed guessing attack process

The first weakness is that in many settings, *the current online payment system does not detect multiple invalid payment requests on the same card from different websites*. Effectively, this implies that practically unlimited guesses can be made by distributing the guesses over many websites, even if individual websites limit the number of attempts.

Secondly, the attack scales well because *different web merchants requests the customer to provide different card data fields*, and therefore allow the guessing attack to obtain the desired card information one field at a time. To understand how essential the scaling issue is, we look at the differences in websites in some more detail. The data fields that web merchants use can be divided into three categories:

- 2 fields: PAN + Expiry date (the absolute minimum)
- 3 fields: PAN + Expiry date + CVV2
- 4 fields: PAN + Expiry date + CVV2 + Address

Figure 16 shows the distributed guessing attack processes. Starting with a valid card number (PAN), to guess the expiry date an attacker can utilise several merchants' websites that check only two fields: the card number and the expiry date. Once the expiry date is known, the attacker can use it along with the card number to guess the CVV2 information using another set of websites that check 3 fields (the card number, the expiry date, and the CVV2).

For many purposes, knowing the PAN, expiry date and CVV2 is sufficient to use a card online, but for some purchases, an attacker would also need to obtain address information. To guess address information, the attacker needs to use websites that ask for 4 fields.

We will demonstrate that the potential impact of the attack is substantial because by exploiting the vulnerabilities an attacker can generate a database of usable payment card details in a matter of few seconds.

The vulnerabilities described in this chapter apply to cards that do not enforce centralised checks across transactions from different sites. Our experiments were conducted using Visa and MasterCard only. Whereas MasterCard's centralised network detects the guessing attack after fewer than 10 attempts (even when those attempts were distributed across multiple websites), Visa's payment ecosystem does not prevent the attack. Because Visa is the most popular payment network in the world, the discovered vulnerabilities greatly affect the entire global online payments system.

The vulnerabilities and the attack scenario that we discuss later in this chapter involves exploiting the weakness in the fraud protection filters that are in place to protect the CNP payment system from fraud. Therefore, to have a complete understanding on working of the distributed guessing attack, in what follows is a brief introduction of online CNP transaction safeguards.

6.2 Online CNP Transaction Safeguards

Card issuing banks and payment acquirers regularly monitor activities on customers' accounts to check if any of those have a risk of being fraudulent. Investigating every transaction in real time is a complex task which can only be achieved by employing a suite of fraud protection software. In this section, we describe the fraud protection tools and filters (often called "controls") provided by online merchant, payment acquirers and card issuing banks.

6.2.1 Address Verification System (AVS)

AVS can be used by merchants to validate the billing address provided by the customer at the checkout page against the address information stored at the card-issuing bank. AVS is an issuer-side control, which means enquiries are made during authorisation phase. The issuing bank will respond to the payment acquirer's request with the result of the authorisation process, along with an AVS response code. A response code can either be a "full match", "no match" or "partial match" and it is completely up to the merchant to decide whether to evaluate the response code and take the appropriate action based on their risk management plan, or not. Regardless of the value of the AVS response, the merchant can still proceed with accepting the transaction as long as there is an acknowledgement of successful authorisation from the issuing bank. AVS can be further improved at the payment acquirer level through the provision of:

- **International Shipping/Billing Address Checks (ISF):** This flag is used by the merchant to screen orders requested to be shipped or completed to foreign country. Such transactions trigger ISF flag and a merchant may either decline a transaction or evaluate it with further verification checks.

- **International AVS Check:** Some merchants limit their services only to domestic customer. For example, Visa checkout [95] only allows US based customers to register their cards on their payment system. Any transactions made using cards issued by foreign banks are declined.
- **Shipping/Billing Address Mismatch Checks:** Orders with different shipping and billing address are either declined or evaluated with further verification checks.
- **Postal code Risk List Match Checks:** Payment acquirers maintain a history of all fraudulent transactions made at specific address zones. New orders involving risk zones trigger this flag, indicating the possibility of new fraudulent transaction. However, there is a problem here. Our experiments have confirmed that different websites perform varying levels of validation upon the address field and the address validation is only on the numeric digits:
    - **Full Validation:** the website validates the full postcode digits and the house number, against the address details held by the issuing bank for the cardholder.
    - **Postcode validation:** the website validates the full postcode digits ignoring the door number against the details held by the issuing bank.
    - **Postcode prefix validation:** the website validates only the prefix of the postcode, for example if the cardholder's full postcode was AB1 2CD the website would validate as correct any postcode starting with 12.

6.2.2 Card Security Code Checks

Issuing banks can also perform CVV2 checks during payment authorisation. For a given PAN, the system generates the CVV2 using (keys) and compare it against value provided by the customers during checkout. Results are later made available for merchants (match or no-match) to decide the outcome for purchase request.

At this point, it is very important to note the shift of liability. Regardless of the outcome from the AVS and CVV2 filters, banks take no responsibility, therefore merchants have to reimburse the card holders for any transactions that are later marked as fraudulent. Card Issuer banks takes fraud liability only for 3D Secure enabled transactions.

6.2.3 Email Service Provider (ESP) and IP Risk List Match Filter

This filter screens for the transaction requests originating from specific domain email addresses. Standard email service providers sometimes keep track of their users' behaviour which may also include traceable information such as IP address and location information [96]. Not to get traced, fraudsters can use completely random and disposable email addresses which are valid only for a few minutes. Such email id's are freely available online through domains like [97] and requires no prior user registration. ESP put a hold on any transaction made using fake and short-lived email addresses and warn merchants about the likelihood of cheat.

6.2.4 Velocity Checks

One way to detect potentially fraudulent activities is by monitoring the number of invalid login attempts made in a certain time span. Attackers can develop and use automated tools against checkout systems to learn passwords and another login information. PCI DSS in requirement 8.1.6 [21] describes testing procedures and guidance to be adopted against brute force attacks to gain unauthorised control of a user account. The requirement states that the procedure to be taken is to restrict continuous login attempts and lock out the affected user account temporarily after five or more attempts. This is called "velocity check" and there are two types of velocity checks that are of importance here:

- **IP Address Velocity Checks:** Blocking legitimate access to a customer account can adversely affect the customer's experience and ultimately might cause loss of business to the merchants. Merchants and acquirers must, therefore, operate an intelligent approach, for example by blocking IP address of known malicious customers attempting to brute force the login, while letting through legitimate customers. Any consecutive login requests originating from blacklisted IP addresses are either blocked or are evaluated under PAN velocity rules which is explained later on.

  Limitations: IP address velocity filter works well for login requests originating from static IP addresses. However, attackers can trick the IP velocity filters by making use of dynamic IP addresses. Moreover, many proxy servers and internet service providers forward requests from customers by routing them to merchant websites through a static IP. In such transactions, IP addresses recorded are of proxy servers' and not of the actual customers'. This will trigger the IP velocity filter to reject the transactions. The bottom line is that fraudsters using a proxy account are harder to track.

- **Account/PAN Velocity Checks.** Just as IP velocity filters tracks IP addresses, acquirers or merchants can use the account number velocity checks to keep an eye on the frequency of use for card numbers. The advantage of using PAN velocity filters is that: if an attacker bypasses the IP filters, multiple purchase requests targeting compromised card number can still be tracked. For example, in the latest payment processing schemes like Verified by Visa and MasterCard Secure Code, irrespective of customers' IP address, cards are blocked for making payment after four invalid transaction attempts.

6.2.5 Transaction Amount Checks

When it comes to electronic purchase on the Internet, transactions can be of any value and/or currency. Largest single online CNP transaction recorded is amounted to 40000000 US DOLLAR(S) by Mark Cuban in 1999 [98]. Card details if compromised, attackers can effort buying a high-value product or can make multiple shots of lower value. To safeguard customers from such attacks,

banking systems usually marks higher value purchase as suspicious until the transaction is actually verified by real cardholders.

6.3 Limitations of CNP Safeguards

With this short survey on fraud filters we observed several inconsistencies and limitations within the implementation of fraud filters.

- Fraud protection controls are not made mandatory by card issuing banks to be implemented by the online merchants. This is also reflected in every merchant online payment implementation guide [34, 35, 46, 49].
- For online merchants several fraud protection controls like Address Verification Systems (AVS), email service provider and IP risk list match filter comes at an extra cost. The online merchant may opt-out of choosing advanced fraud filters, but this comes at a risk of less secured checkout system.
- As standards do not define any requirement for velocity filters (IP address and PAN velocity checks), this leaves payment acquirers and online merchants more open options for determining precise limits of invalid attempts.
- The more security filters implemented by the online merchant, the more fields a customer has to fill in the checkout and less convenient it is for the customer to make payment.
- There are already established limitations with IP velocity filters on their limited ability to keep track of attackers operating from dynamic IP addresses.

All these inconsistencies in the ways that an authorisation-only transaction is assessed made us to design experiments to assess their security.

6.4 Vulnerability Assessment Landscape

6.4.1 Selection of Websites

This study was conducted on over a total of 400 global commercial websites. The selection of websites was based on a maximum number of visitors as provided by Alexa web traffic analysis. In some cases, it was required for us to study the websites that we have been regularly using, this answers our question on whether the loyalty of returning customers over the web merchant have any change in efficiency of security filters. Since some web merchants redirect or host the payment page to the supported payment acquirers (hosted checkout), for websites we studied, we made sure that they accept and support at least one or more type of payment card - credit card, debit card, prepaid card. We also created new accounts on various payment sites with account holder's name that do not match the names printed on the payment cards. This is to replicate the situation where the attacker might not necessarily know the cardholders name, and to illustrate the issue that hardly any of existing online payment systems cross check the cardholder's name.

Table 9 - Test cards used for experiments

Card Ref	Card Type	Card Number	Expiry Date	CVV2
V1	Visa	4xxx-xxxx-xxxx-1010	06/2017	126
V2	Visa	4xxx-xxxx-xxxx-4009	03/2018	577
V3	Visa	4xxx-xxxx-xxxx-2617	05/2017	349
V4	Visa	4xxx-xxxx-xxxx-9010	02/2018	022
V5	Visa	4xxx-xxxx-xxxx-8649	04/2016	927
V6	Visa	4xxx-xxxx-xxxx-5027	10/2015	954
V7	Visa	4xxx-xxxx-xxxx-1719	07/2018	614
M1	MC	5xxx-xxxx-xxxx-8565	08/2016	090
M2	MC	5xxx-xxxx-xxxx-0106	07/2015	130

6.4.2 Test cards

In total 11 test cards were included in the research: divided between nine Visa cards and two MasterCard cards. Details of the type of card used, masked card numbers, range of expiry dates and CVV2's are shown in Table 9. All test cards used belonged to our research team. It is true that we have all the card details, but we configure the bots to replicate the situation whereby these pieces of information are not known to the attacker at the beginning.

All the experiments we performed were from a Windows 10 machine, with Chrome web browser. To automate the interaction with web stores where required, we designed a website bot using Selenium web drivers and AutoIt scripts.

6.4.3 Software Tools

To test the websites and validate our research, we have implemented a number of software tools, which demonstrate the viability and practicality of the attack. (We stress that we only tried the tools on our own credit/debit cards.) The tools consist of two separate applications:

Automated web crawlers to automate the process of guessing the remaining card details. There are two types of bots implemented:

- **Website bot,** as introduced in Chapter 4, a website bot can run the experiments against fraud prevention filters. Website bot can also be used to brute force expiry dates, CVV2 values and postcodes until the selected website or group of websites API returns a true value (indicating a successful attack) or the website's limit on the number of attempts has been reached (failed attack).
- **AutoIt scripts.** An AutoIt [99] script, which interfaces with a well-known service provider in order to brute force CVV2 guesses until this service provider accepts the card for a correct

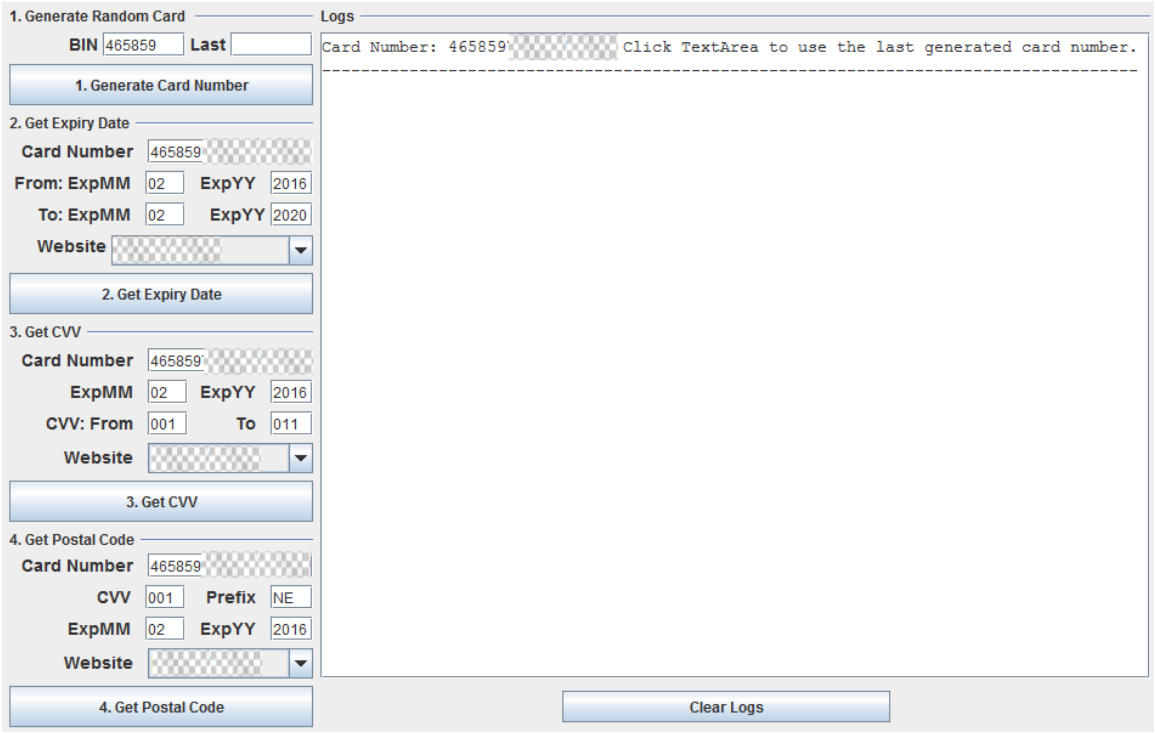

Figure 29 - A snapshot of the Website bot

value of CVV2 (there is no limit enforced by this service provider regarding the number of guesses allowed, so this attack will eventually return a success all the time). An example AutoIt script to automate iTunes application is shown in the textbox below:

```
$cvv = 001
$password = "************"
Run ("C:\Program Files\iTunes\iTunes.exe", "C:\Program Files\iTunes\")
Sleep (5000)
AutoItSetOption ("MouseCoordMode", 0)
WinWait ("iTunes")
WinActivate ("iTunes")
MouseClick ("primary", 1046, 23, 1, 100)
MouseClick ("primary", 1046, 172, 1, 0)
Sleep (5000)
Send ($password)
Sleep (2000)
Send ("{Enter}")
Sleep (20000)
MouseClick ("primary", 938, 273, 1, 50)
Sleep (10000)
MouseClick ("primary", 585, 281, 1, 200)
Sleep (2000)
Send ("100")
Sleep (5000)
Send ("{Enter}")
For $cvv = 001 to 999 Step +1
Sleep (10000)
MouseClick ("primary", 594, 352, 1, 200)
Sleep (5000)
Send ("{BS 3}")
Sleep (5000)
Send ($cvv)
Sleep (5000)
Send ("{Enter}")
Sleep (5000)
Next
```

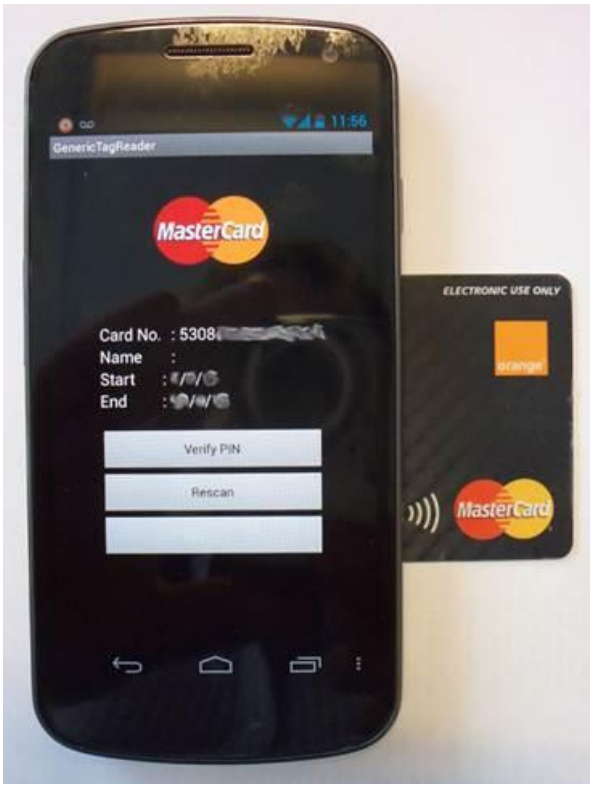

Figure 30 - Android app for NFC skimming

- NFC Android app: additionally, we programmed an NFC android skimming application capable of reading an EMV contactless payment card. When a card was scanned, the android application is capable to read the card number, the card's expiry date and any additional payment application information available on the card's contactless interface.

6.5 Identification of the Vulnerabilities

We started our assessments by selecting 50 websites from our 400 global commercial websites and created purchase orders of values ranging from £1-£500. Once we were directed by the website to the checkout page to pay for the checked items, we recorded and compared several features from websites which included:

- Type of payment options available (card payment, cash-on-delivery)
- the number of card data fields requested
- the type of CNP payment protocol implemented by the web merchant (authorisation-only or authentication-enabled)
- the support of multiple payment options

Figure 31 - Illustrating the components of website-x checkout page. It can be observed from the figure that: (1) the type of payment cards supported, (2) Number of card data fields requested and (3) indicates the type of protocol used for accepting payment (as there is no 'verified by payment-network icon, the website supports authorisation-only payment protocol). This created one entry in set of

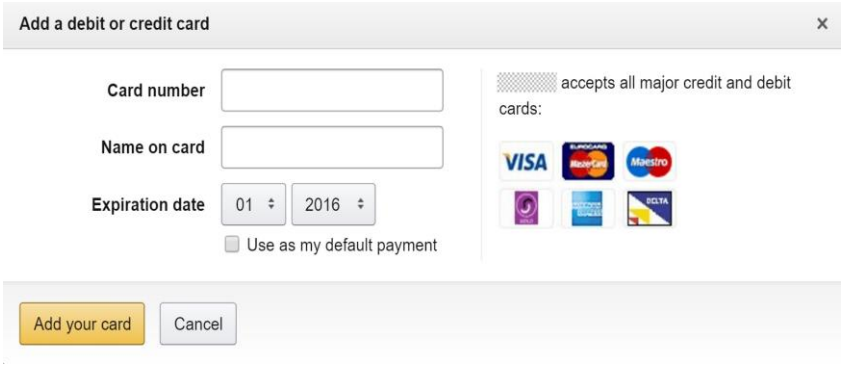

Figure 31 - Illustrating the components of website-x checkout page

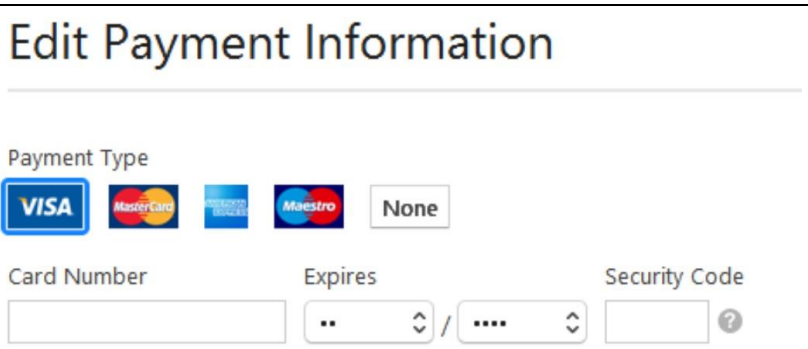

Figure 32 - Illustrating the components of website-y checkout page

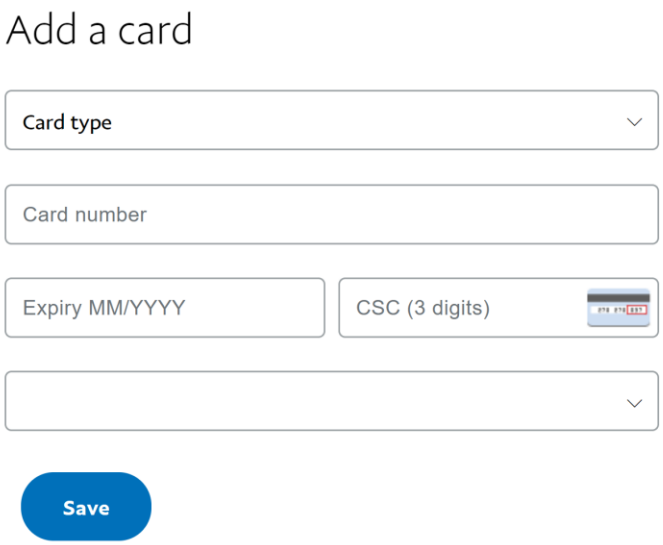

Figure 33 - illustrating the components of website-z checkout page

reference tables maintained to record checkout information from all the online merchants that we visited. Appendix D gives a reference table which documents the details that we captured from each website. Comparing Figure 31-33, it can be observed that different websites request different pieces of card information from their customers.

6.6 Attack Scenario

This section describes the distributed guessing attack scenario which is composed of three stages: *generating the card data fields, creating a fraudulent account, and finally transferring the money*.

6.6.1 Generating the card data fields

Card data fields have a limited number of possible values (analysis of which is shown in Table 10). This means an automated web script (a "bot") can cycle through these values in a short time to find the correct information.

For example, to find the correct CVV2 value, the bot will simply need to cycle through the possible values starting at 001 until the website blocks further attempts (i.e. the attack has failed) or the website API returns a success. A handful of payment sites allow unlimited attempt, while most of the other payment sites allow ten or 50 attempts to enter a correct CVV2 value. This may not sound like a big number but coupled with another limitation in the global payment system (there is no evidence of coordination or synchronisation among various payment sites regarding a particular payment card), there exists a scenario of "farming out" the brute force attack to tens or even hundreds of payment systems, which practically means we can carry out unlimited guesses.

Table 10 - Possible values of generating card data fields

Field	Possible Values
Expiry Date	Up to 60 possible values (12 months * 5 years) –average lifetime of credit cards is 3 years while debit cards are usually valid for 5 years.
CVV2	999 possible values (001 to 999).
Address of cardholder	999 possible values in the United Kingdom. Address validation is performed for only the numeric digits of the postcode, a full list of which can be obtained from the internet.
Name of cardholder	Any random value can be entered. Our research has found that the cardholder name is not validated on any of the payment websites we used in our experiments.

6.6.1.1 Obtain PAN

In the attack scenario described, the 16-digit card number (PAN) is the starting point for the generation of all of the other fields. We identify three methods of obtaining valid PANs:

Prevalence of Card Number as a Plain text: The design of Europay MasterCard Visa (EMV) protocol [23][25] mandates the card number to be stored as plain text within the card's memory; this enables even an illegitimate card reader to communicate and interpret the card details. Such an unusual design for a payment protocol offers opportunities for an attacker to obtain card details. It is

Figure 34 - Merchant receipts obtained from different acquirers

well-known for a contactless card that card number and expiry date could be *skimmed* from a distance with any NFC enabled device [29][32] and in fact, in a single google play search, we located 38 freely available Android apps which could be used by an attacker to read the contactless payment cards.

Merchant Receipts. Another channel that for an attacker to follow to obtain the card number is from the merchant's sales receipt from reader Point of Sale (POS) terminal.

To maintain the sales records made using payment cards, in-store merchants store merchant copy of customer's transaction. We found that the merchant copies from a number of high street retailers revealed their customer's complete credit card number and expiry date (shown in Figure 34). Worst of all, none of the merchants were educated about the risks of losing merchant copies, and few of the merchants even agreed to sell several merchant copies for under a pound. This means, whenever a customer uses their card with in-store retailers, there is a risk of card number being stolen. Just from around our organisation, we found 23 such retailers whose merchant copy revealed full card number and expiry date.

Guessable Card Numbers We further investigated the possibility of an attacker generating payment card numbers and explored that an adversary can easy produce and validate a database of active payment card numbers as discussed below:

The payment card numbering specifications are governed by the ISO/IEC 7812-1:2017 [100] and the ISO 10202-6:1994 [101]. Table 11 enumerates the useful insights that can be obtained from a credit card number. It can be learned from a card number that customer account number fills nine spaces and therefore, the maximum number of possible active card numbers for a bank would be one less to 10^9 (a billion). An attacker starts after selecting target banks BIN (bank with a high number of customers would give high positives), randomly generates thousands of accounts number using Luhn's check algorithm [102](or with automated bot) and makes transactions using the generated card numbers on online payment websites.

Table 11 - Card number information fields (Numbering is from left to right)

Card number: 4658 – 5900 – 0000 – 000C
• First six digits: called as Bank Identification Number (BIN), identifies the card brand and issuing bank • Digits 7 to (15): assigned by the card issuing bank and denotes personal account number (shown as zero's) • Last digit: akin to checksum (indicated by 'C'), used by a computer to verify the card number entered is correct

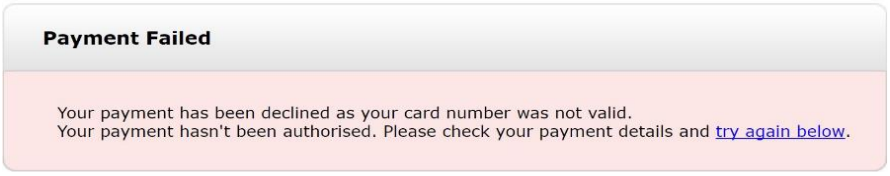

Figure 35- Response code revealing the validity of a card number

Figure 36 - Payment cards belonging to the same cardholder and card numbers are shown issued in a sequence

When a transaction is made, a transaction authorization request is sent by the merchant to the card issuing bank. The card issuing bank, through authorization response message (further discussed in next section), indicates to the merchant that the card number used while trying to make a purchase is not correct. For an attacker, the Authorization response will reveal the validity of a card number. For example, when we made a transaction with invalid card number on a merchant website-x (website name masked), we received the response as shown in Figure 35. If the card number was valid, the authorization only changes to indicate any other invalid card data element. A recent investigation into Tesco bank breach revealed that attackers used similar technique to exploit payment card details of around 9000 customers [20]. Additionally, we observed a weak security practice by a leading card issuer (name masked) while issuing the payment card numbers to their customers. We found that the card issuer issued payment card numbers in a serial guessable sequence. Shown in Figure 36, are three payment cards belonging to the same customer and card numbers are shown issued in a sequence with a difference of 8.

Underground forums. It is well known that criminals sell credit card details in bulk online. They are pretty cheap to buy (especially if you only need the PAN and the expiry date information), so an

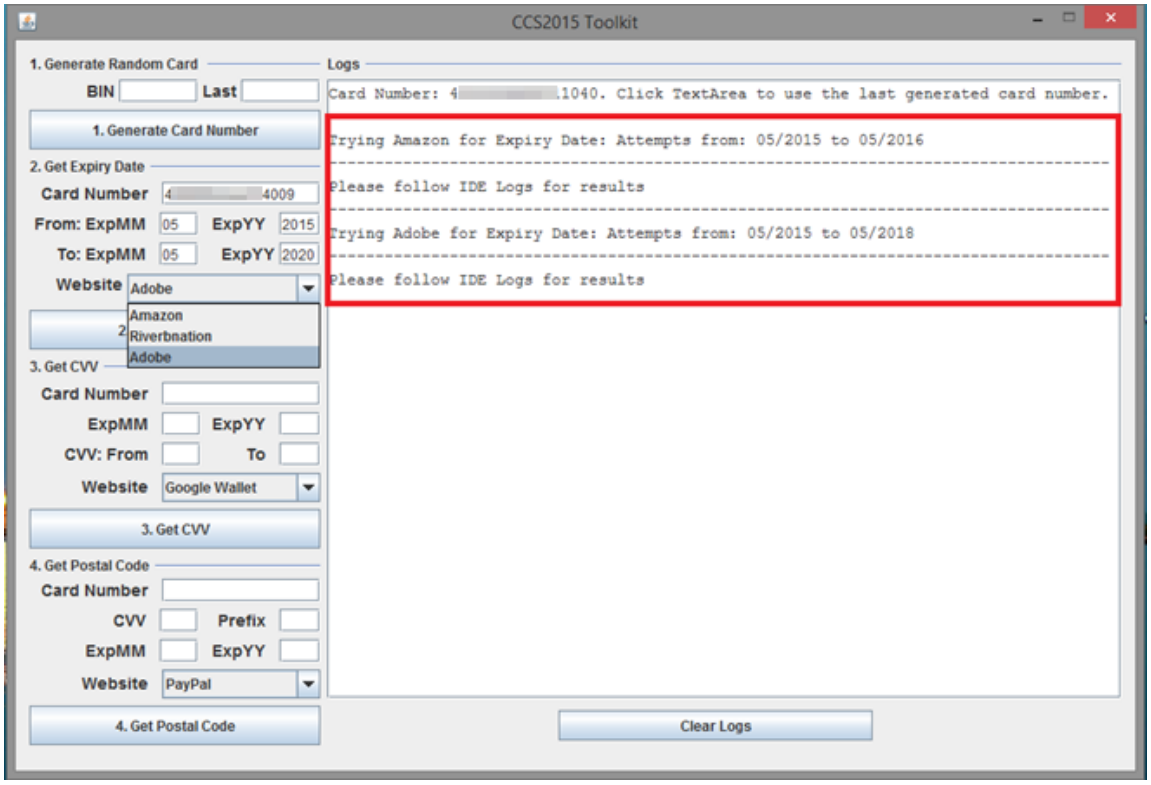

Figure 37 - A website bot instance for finding expiry date

attacker can easily take this route to come up with the PAN (and the expiry date). In Appendix C, we identify 25 live underground forums where payment card details are still traded.

6.6.1.2 Obtain Expiry Date

Once the PAN is known, an attempt to obtain the expiry date can commence. In many cases, the expiry date can be obtained at the same time as the PAN, for example by using the NFC skimming method described above. But if that is not possible, a distributed brute force attack can be carried out. A bot or a script can be programmed to systematically guess the expiry date of a given PAN. As shown in Table 10, a maximum of 60 guesses will be needed to find the correct expiry date. Many payment sites are prone to this attack because they do not limit the number of attempts one can make to guess the expiry date, and they do not require CVV2 or address to be entered.

Our experiments were successful in getting a valid expiry dates for each of our Visa test cards. Among 26 websites which were used to get expiry dates, two highly popular websites allowed us unlimited attempts to verify card number and expiry date match.

6.6.1.3 Obtain CVV2

The third stage of data generation involves getting the card's CVV2 using the PAN and the expiry date information obtained in the first two stages. To achieve this, we need to configure the web bot to switch to payment websites that do not implement AVS filters for card verification or payment capture. For a given range of CVV2 guesses, the bot iterates over the list of selected websites and multiple instances can be run in parallel to speed up the experiments and to overcome the attempts

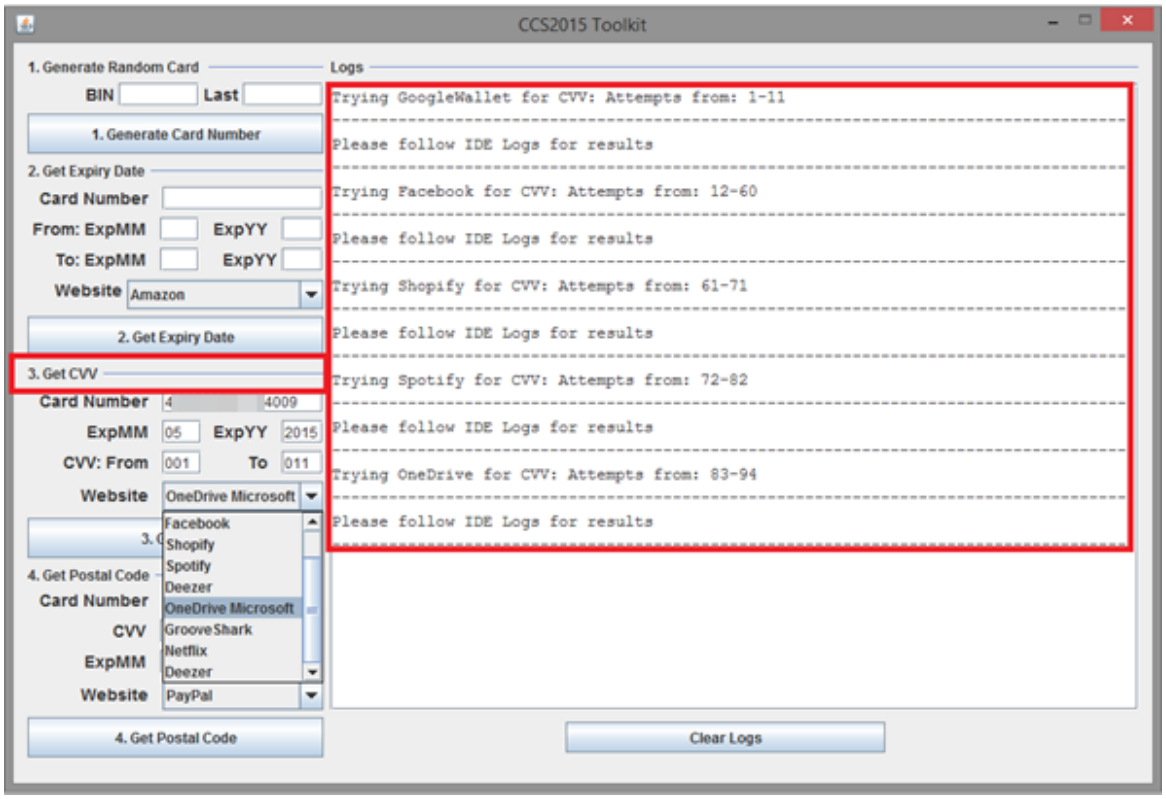

Figure 38 - A website bot instance for finding CVV2

limit imposed by IP and PAN velocity filters. Figure 38 shows a screenshot of the bot trying to find a card's CVV2 from 291 websites.

More than 11,000 CVV2 iterations were performed using the web bot and scripts, and our experiments have found that there is no standard (or centrally imposed) limit on CVV2 attempts. We have managed to find valid CVV2 for all of our Visa test cards.

6.6.1.4 Obtain Postcode

The final stage is to get the postcode. BinDb [103] and ExactBins [104] are two well-known online databases that feature a list of card-issuing banks based on the 6-digit BIN. The free version of such online databases allow an attacker to learn about the card's brand, issuing bank name, and card type. An example output from ExactBins is shown in Table 12 below.

Once the issuing bank is known, the attacker can increase the probability of guessing the right postcode by assuming that the victim might be registered with one of the branches nearby – this is particularly relevant if the attacker uses NFC skimming to obtain the PAN and expiry date in the first place. Now, the web bot just needs to start brute force guesses from a list of issuing bank postcodes for a particular city where the card details have been skimmed from.

We performed more than 3000 iterations to get first four characters of the postcode. We extended our experiments running multiple instances using our bot on the hosts that verify postcode and complete address. Hence, we achieved in getting the complete address for all our visa test cards.

Table 12 - A sample of the information given by ExactBins

BIN	465859
Card Brand	VISA
Issuing Bank	Barclays Bank PLC
Card Type	Debit
Card Level	Classic
ISO Country Name	United Kingdom

There is no need to generate the cardholder name because this field is not verified by any payment systems, i.e. any value will be accepted by these systems. At this point, the attacker is ready to make fraudulent transaction or money transfer.

6.6.2 Transferring Money

Once either two, three, or four fields of the card data have been obtained, the attacker can use them to purchase goods on a website. This is damaging enough for the owner of the card, but we looked at even more impactful attacks. Rather than buying online goods from an online merchant website, we created an attack scenario that uses the card details to open a money transfer account, sends the money to an anonymous recipient abroad, where the money is picked up within minutes of issuing the transfer. The attacker needs to be able to clear the funds before the issuing bank reverses the payment and thwarts the attack. It is therefore desirable from the attacker's point of view that the funds are transferred to an account outside the country (because it is more time consuming and costly to reverse payment across countries) or be conducted through a wire transfer to an anonymous cash recipient by using services such as the Western Union.

In our experiment, the card information extracted using our bot was used to create a bogus account from which we transferred money to a recipient in India. Within minutes, we received a confirmation email for the order made, and our contact confirmed the pick-up of the money. The time it took from the process of creating an account to collecting the money at the destination was only 27 minutes, which is short enough to avoid the bank reversing the payment.

6.7 Guessing Attack as Systemic Problem

Our brute force attack experiments have demonstrated an important vulnerability in online payment systems whereby the variations in payment security settings across multiple online merchants' websites could be exploited to make fraudulent online payments.

In order to further illustrate the feasibility of such attack, we carried out a comprehensive survey of existing online payment technology, in which we examined the payment security settings of the top 400 Alexa rated commercial websites [105], including many top global websites such as iTunes, Google, PayPal and Amazon. Our survey has documented the following aspects:

- the data fields requested by each merchant to authorise an online payment
- the number of incorrect attempts for each of these fields
- the presence (if any) of additional security features selected by online merchants such as Verified by Visa and MasterCard SecureCode
- sites which supported alternative payment methods, such as PayPal, where the customer was not asked for their credit card details

The aim of the survey is to assess the nature and scope of the vulnerability described in Section 6.6. We would like to find out whether this vulnerability only affect a few merchants with poor implementation of their online payments (in which case our attack scenario would only have a very limited impact), or whether this was as a result of a systemic problem created by the underlying regulations and design of the payment networks (which makes our attack scenario feasible and practical).

Unfortunately, our survey shows that this is indeed a systemic problem, affecting all websites including the most popular online retailers.

6.7.1 Survey Results

Our survey results (detailed in Table 13 - Variation in payment security settings of online payment websites) show that the attack described in Section 6.6 is a credible threat. The variations in online merchants' payment security settings exist in large enough numbers to be exploitable. We used data from 389 of Alexa's most visited sites. There are 26 sites from which an attacker can generate the expiry date, 291 sites for generating the CVV2 and 25 sites for generating the postal code. There is also a variation in the number of attempts allowed at each of these sites, ranging from 4, 5, 10, 20, 25, 50, or even unlimited. Of these 389 sites, there were only 47 merchants (i.e 12%) who had implemented 3D Secure payments [106][107][108], which would have prevented our attack.

Table 13 - Variation in payment security settings of online payment websites

Number of attempts	Expiry Date	CVV2	Postal Code	3D Secure	Total
0 to 5	2	23	2	-	27
6 to 10	20	238	18	-	276
11 to 50	2	28	3	-	33
Unlimited	2	2	2	-	6
3D Secure	-	-	-	47	47
Total	**26**	**291**	**25**	**47**	**389**

Table 13 shows that the majority of online merchants (291, i.e. 74.8%) chose to use the 16-digit card number, the card expiry date and the CVV2 as the payment security fields for authorising payments.

There are 72 merchants who opted for additional security, of which 25 are requesting the verification of the customer's postal code and 47 are using 3D Secure. Twenty-six merchants, including one merchant in the top ten, opted for much lower security settings, just requiring the 16-digit card number and expiry date. It is these merchants that allow the attack described in Section 6.6 to be accomplished.

Table 14 - Variations in payment security data fields required

Alexa Rank	Expiry Date	CVV2	Postal Code	3D Secure	Total
0 to 100	6	68	5	21	100
101 to 200	5	81	2	12	100
201 to 300	7	79	2	12	100
301 to 389	8	63	16	2	89
Total	**26**	**291**	**25**	**47**	**389**

One of the primary aims of the survey was to determine if the vulnerabilities described in this chapter were systemic. To assess this, we compared the security profiles of the most popular websites with the less popular websites, grouping the surveyed websites into a popularity order according to Alexa top 100, followed by 100 to 200, 200 to 300 and 300 to 400.

We observed that the more popular online merchants were more likely to implement additional security features, such as address verification and 3D Secure (Table 14). There were two notable outliers to this observation in the top 10 merchants, one of which allowed unlimited attempts to input the CVV2 and another who only required the 16-digit card number plus the expiry date.

An interesting finding from our survey was that the majority of the merchants (276, i.e. 71.0%) opted to give between 6 and 10 incorrect data entry attempts. This is in contrast to 3D Secure which only allows 4 incorrect data entry attempts. Table 15 shows that more popular online merchants were more likely to implement 3D Secure, which limits the number of attempts to 4.

Table 15 - Variations in number of incorrect payment data input attempts

Alexa Rank	0-5	6-10	11-50	Unlimited	3D Secure	Total
0 to 100	9	53	16	1	21	100
101 to 200	12	63	13	0	12	100
201 to 300	1	86	1	0	12	100
301 to 389	5	74	3	5	2	89

Total	**27**	**276**	**33**	**6**	**47**	**389**

Furthermore, our survey counted the number of merchant websites which accepted alternative methods of payment such as PayPal [96], Google Wallet [109], Visa Checkout [95], MasterPass [110] and Amazon Payments [111]. These digital wallet systems have the advantage that the customer enters their credit / debit card details in a single trusted system (the digital wallet), which makes it possible to pay at the merchant's website without revealing the card details. Table 16 presents the figures of accepted alternative payment methods among the top 400 Alexa ranked websites [105].

Table 16 - Merchants supporting alternative methods of payment

Alexa Rank	PayPal	Visa	Master Card	Amazon	Others	Total	None
0 to 100	61	10	4	3	6	84	16
101 to 200	67	8	3	6	5	89	11
201 to 300	67	2	1	6	7	83	17
301 to 389	67	2	1	6	4	80	20
Total	**262**	**22**	**9**	**21**	**22**	**336**	**53**

For merchants, digital wallet systems have an advantage that they (the merchants) do not have to store payment cards and shipping details on their own PCI-certified secure server, thereby reducing their regulatory burden and the risk of online card fraud and identity theft losses which could put small merchants out of business. The disadvantage of digital wallets is that customers who are not signed up for that particular brand of wallet, e.g. a customer who has Google Wallet but not PayPal, may be discouraged from completing their purchase, thereby causing a lost sale for the merchant. Our survey shows that digital wallets are accepted by 336 (86.4%) merchants out of the 389 observed. The most popular digital wallet is PayPal, which is accepted by 262 (67.4%). Surprisingly, only 51 (13.1%) merchants accept multiple digital wallets.

More detailed survey results are given in Appendix D.

6.8 Responsible Disclosure

Two weeks after we completed the distributed guessing attack experiments, we initiated an ethical/responsible disclosure exercise, notifying Visa and a selection of affected sites. Based on the number of fields that a website checks, we categorised them into three groups: expiry date, CVV2 and postcode. Since the total number of vulnerable websites was very high, we selected the 12 biggest

Table 17 - Nature of patching on the notified websites

		Patching Behaviour				
Website	**Information Leak**	**Adding Addr. field**	**Adding Delay filter**	**Adding velocity filter (PAN based)**	**Adding velocity filter (IP based)**	**Adding CAPTCHA**
A	Exp. date	√				
B	Exp. date	√				
C	Exp. date		√			
D	Exp. date		√			
E	CVV2			√		
F	CVV2				√	
G	CVV2				√	√
H	CVV2				√	√

players from each category (in terms of the highest number of users), taking the total number of notified websites to 36.

Once a suitable contact person or team for each website was found, we presented them with the disclosure information that featured the experiments we performed and the type of vulnerabilities on their site. We used our official work/university email address, and this served as a means for these merchants to trace us back, so that they can verify our authenticity. This would also allow them to request more detailed and technical information about our experiments should they wish to find out more.

We recorded the responses received from these websites over the duration of four weeks after we disclosed the vulnerabilities to them. Altogether, we received 20 human responses from 10 websites and 18 websites came back to us with machine generated response mostly confirming the receipt of our notification. All the human responses requested more technical details while some asked us to suggest solutions. Out of the 36 websites we contacted, eight never responded. When a web merchant requested more information, we offered them an initial draft of our research experiments and results, which explained the experiments and the attack to help them understand the actual problem. We followed the disclosure policy requested by the websites and anonymised the affected sites in our research publications. As a result of our disclosure process, eight of the 36 websites changed their online security settings, but the other 28 websites remained unchanged four weeks after the disclosure. We call such changes ‘patches’ in what follows, and Table 17 illustrates the nature of the patching of the notified websites. Of the eight websites that modified their approach (labelled A to H), four used two fields (labelled ‘Exp. Date’ in the ‘Information Leak’ column) and four used three fields (labelled ‘CVV2’).

In most cases, we learned about the patching behaviour through manual observations, but in two cases (Website B and Website G), the affected websites notified us about the changes they made. Website A and Website B patched their checkout system by adding an address verification field. However, this

was not a good idea because it did not provide additional security, but instead opened up a new avenue for guessing and will be discussed at the end of this section.

Typically, an online payment request is authorised almost instantly (within 2 seconds). From our observation, we noticed that Website C and Website D (both with expiry date leak) had introduced additional delays to the payment authorisation processing times. They did it in a staggered manner: few attempts were processed instantly but after certain incorrect attempts had taken place, the time taken for payment confirmation were increased. In this manner, fewer attempts were available (at least practically speaking) to enter the right expiry date without setting a hard upper bound to the number of attempts.

We found that Website E (one of the Alexa top-10 websites in terms of the number of visitors) patched their checkout system by adding PAN velocity filters, reducing the number of attempts allowed (based on the PAN) from unlimited to 100 attempts within 24 hours. Website F followed a similar approach and added IP-based velocity filter to limit the number of attempts to get CVV2 from 50 to 10 in 24 hours. Initially, Website G and Website H added CAPTCHA on their checkout page, thus disrupting our bot from carrying out the attack. Our experiment protocol limited the interaction with the administrators of notified websites. Due to complex trade-offs that payment websites need to consider when deciding which fields and filters to use, our ethical disclosure protocol did not volunteer advice about what actions to take to deal with the vulnerabilities. However, in one situation we felt we needed to depart from the protocol, namely in the case of Websites G and H, who added a CAPTCHA. CAPTCHAs prevent automated attempts in getting the sensitive card information but may adversely affect the usability of those websites [112]. To help Websites G and H to better understand the implications of adding a CAPTCHA, we provided these two websites with more detailed information about the attacks. This resulted in the CAPTCHA being replaced with IP address velocity filters, which allowed five attempts per IP address in 24 hours (hence a mark in two cells in Table x, for these websites).

The overall result of our study on the nature of patching on the notified websites revealed that the vast majority (78%) did not make a change. We do not know the reason behind this and further research will be needed to find the explanation. Of the eight that patched, the general approach taken by merchants is either to add a filter to make it more cumbersome to try many times (6 of 8 sites that patched added delay or velocity filters), or to add a field (Website A and Website B). Perhaps surprisingly, none of the sites reacted by simply putting a hard limit on the number of allowed attempts. The effect of these patching behaviours is not so obvious. As we already pointed out, the sensible measure of limiting the number of attempts will not stop the guessing attack if it is not done on all websites. Furthermore, adding a card validation field may be a reasonable idea for a site for various reasons, but inadvertently may even weaken the protection against the guessing attack of the

payment system as a whole. After all, the added field may be a welcome opportunity to attempt guesses on this added card detail.

6.9 The Challenges in Solving the Problem

Improving the security of the online payment system is a complicated challenge for a variety of reasons. One could argue that payment card security mechanisms are bound to remain unsatisfactory since they have not been designed for distributed operation over the distributed Internet. Many of the solutions, such as 3D Secure can be seen as afterthoughts, and they struggle to gain widespread adoption. Any suggested improvement or solution faces the challenge that the online landscape contains many players that all have their own – at times competing – incentives for or reasons against change. Any solution would have to combine technical concerns with financial and business operational concerns, and its adoption will depend on legal and economic dynamics. We explore and discuss these issues from the perspectives of the five parties involved in authorisation-only transaction process.

6.9.1 Customer / Cardholder

Since the distributed guessing attack described in this article uses merchant websites and card payment network to get all the card details, there is not much a cardholder can do to prevent it. At the same time, the cardholder is severely impacted by the attack: money may be lost, cards may have to be blocked, and the result is a waste of time and effort and a decreased sense of security. Arguably, it would be beneficial for cardholders if they could get organised as a group, or would have representatives in various bodies, to put pressure on the other stakeholders. As an individual, cardholders could ‘vote with their feet’ and select cards from card payment networks that are not exposed to the distributed guessing attack. At the moment, the payment system is too complex and non-transparent to expect customers to be able to make such choices.

6.9.2 Online Merchant

On their own, a merchant can do very little to prevent distributed guessing attacks. All merchants would have to agree or be forced to use the same number of fields so that the guessing attack cannot be staged as explained in Section 6.6. At the same time, a merchant can avoid being exploited in the attack either by only using cards that use a payment network that is not vulnerable from the attack, or by using 3D Secure technologies recommended by the payment card industries such as the American Express ‘SafeKey’ [68], ‘Verified by Visa’ [94] and MasterCard ‘SecureCode’ [93]. If 3D Secure is implemented, the card issuing bank is responsible for authenticating a cardholder before authorising the payment and it monitors the frequency of transactions and the total value of purchases for each card or bank account. The system will initiate additional security checks such as IP address and/or request an additional password if the frequency or value of the transactions appears to be unusual. Our experiments confirmed that 3D Secure payments are protected from the distributed guessing attack

described in this article since the issuing bank has visibility of all transaction requests directed at a single card, even if those requests are distributed across many websites. From the perspective of the merchant, 3D Secure has several drawbacks, and these are reflected in that only 47 merchants in the Alexa top-400 have elected to implement 3D Secure. First, the proportion of the customers who do not complete the transaction can be high when the customer encounters the 3D Secure login screen: up to 43% in the United States and 55% in China [113]. Second, there are additional costs associated with implementing 3D Secure. We reiterate that from the whole payment system's perspective, we would need a very high adoption rate of 3D Secure technology to prevent the distributed attack, because the attack would still work as long as there are sufficient vulnerable websites not using 3D Secure.

6.9.3 Payment Acquirer

There are many payment acquirers, which charge web merchants different rates depending on the number of fields and filters they ask to check and utilise. One cannot expect all of these acquirerss to be able to coordinate sufficiently to prevent the distributed guessing attack. Nevertheless, payment acquirers can provide advanced features to their merchants, and these features should at least make it more difficult to exploit a website for the attack. Most importantly, payment acquirers may use IP address velocity filters [34, 35, 46], which are implemented to detect repeated invalid attempts made within a certain time span from the same IP address. But with no coordination between different payment acquieres, these velocity filters can easily be circumvented just by switching to a website that uses a different payment acquirer.

6.9.4 Card Payment Network

Our experiments have shown that distributed guessing attack described in the chapter only works on Visa cards, independent of which bank issued the card. When the attack is applied to a MasterCard, the distributed attack is detected. This suggests that the payment networks have the capability to detect and prevent a distributed attack where the network is globally integrated [114].

The most obvious defence against the distributed guessing attack would be at the level of the card payment network. However, we are not in a position to know whether payment network providers could modify their network infrastructure to detect payment requests from multiple, globally spread payment acquirers, looking for suspicious activities on a single card distributed across multiple merchant websites.

6.9.5 Card Issuing Banks

The bank comes into play at the final stage of the payment process, to approve the transfer of funds, but it would not be party to each individual guess (unless 3D Secure is used). Banks play an important role in limiting the damage that can be done if attackers get hold of card information. Many issuing banks are now running intelligent fraud detection systems which detect transactions which are outside

their customer's normal spending habits [66]. The issuing bank then has the option to block the payment, or ask the customer for confirmation, or accept the payment taking a calculated risk that a transaction may be found to be fraudulent later. A complicated set of considerations comes to the fore in the bank's decisions, from ease of use to financial risks. However, one would expect that if they so desire, banks could have considerable influence on the payment acquirers and card payment networks in protecting against the distributed guessing attack.

6.10 Conclusion

In this chapter, we studied 400 of the most popular e-commerce websites and surveyed their web payment interface, identifying that different websites present different sets of fields to identify the cardholder. It turns out that this disparity between different websites inadvertently creates conditions for a scalable distributed guessing attack. By conducting a guessing attack one field at the time – using a set of appropriate websites at each stage – the attack becomes practical. With the obtained data, the attacker can make purchases or transfer funds, as we have demonstrated.

We showed that the attack works if the card payment network is not able to relate card activities from different websites. Fundamentally, much of the problem with card payment stems from the fact that the identity of the payer needs to be established in the 'card-not-present' mode. This is inherently problematic since it is at odds with the original use of cards (where the card and cardholder are present at the moment of purchase). It also implies that, for instance, Chip-and-PIN is not available to establish the identity of the payer. This is exacerbated by the fact that the Internet facilitates distribution of guesses for data fields over many merchant sites.

To prevent the attack, either standardisation or centralisation can be pursued (some card payment networks already provide this). Standardisation would imply that all merchants need to offer the same payment interface, that is, the same number of fields. Then the attack does not scale anymore. Centralisation can be achieved by payment gateways or card payment networks possessing a full view over all payment attempts associated with its network. Neither standardisation nor centralisation naturally fit the flexibility and freedom of choice one associates with the Internet or successful commercial activity, but they will provide the required protection. It is up to the various stakeholders to determine the case for and timing of such solutions.

Chapter 7. Reverse Engineering the 3D Secure 2.0 Frictionless Authentication

3 Domain Secure 2.0 (3DS 2.0) is a user-authentication payment protocol introduced to combat online card payment fraud while keeping online payment simple and fast. 3DS 2.0 can challenge the payment initiator, who provided the payment card details, for second factor authentication information (e.g., a passcode). For convenience, 3DS 2.0 also offers a frictionless authentication mode, in which the authenticator can decide not to challenge the payment initiator depending on perceived transaction risk. Transaction risk is assessed based on rules set in the 3DS authentication service using several sources of information, including data fingerprinting the browser. The 3DS 2.0 standard itself does not specify how to implement transaction risk assessment for frictionless authentication. The research question addressed in this chapter therefore are: how is transaction risk assessment implemented for current payment cards; do there exist practical attacks against 3DS 2.0 exploiting the frictionless option; and, if so, what alternative designs exist to avoid the security problem while maintaining reasonable ease-of-use.

This chapter therefore conducts a detailed reverse engineering study of frictionless authentication in 3DS 2.0 for payment using a browser. We identify the data that card issuers use for transaction risk assessment, for five payment cards from Visa as well as Mastercard, used at a number of different web sites. In addition, we conduct experiments to identify if the 3DS authentication service uses additional rules in the risk assessment.

7.1 Revisiting 3DS 2.0

3DS 2.0 protocols was introduced to combat the “phishing” and “malware” attacks that were still possible on 3DS 1.0 protocol. These attacks enabled fraudsters to steal the static passwords used by 3DS 1.0 for cardholder authentication. With 3DS 2.0, as shown in Figure 39, the card issuing bank performs user-authentication either by challenged authentication or frictionless authentication. With challenged authentication, cardholders were authenticated through one-time tokens such as passcodes over the phone. However, such schemes of user-authentication are still vulnerable to real-time man-

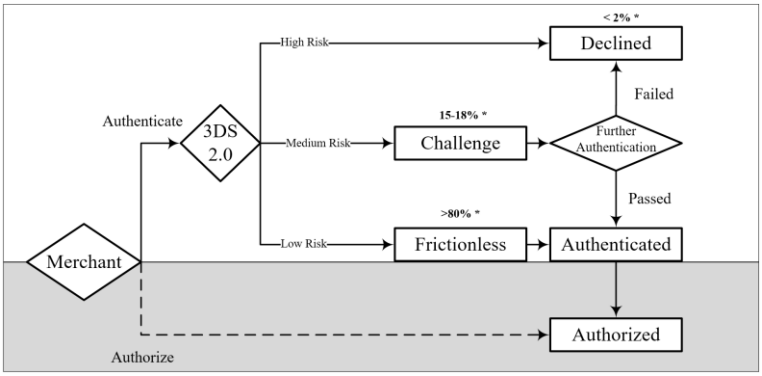

Figure 39 - 3DS 2.0 supported transaction schemes

in-the-middle attacks. As demonstrated by Drimer et. al. [3] and RedTeam research [2][3], attackers can initiate transaction with card issuing banks and synchronously prompt the cardholder to enter their one-time passcode information on phishing websites. Such one-time passcodes can also be easily stolen by malware hidden within the cardholder machines [3].

However, such kind of attack requires real time co-ordination between the attacker, the card-issuing bank and the cardholder. The payment networks and card-issuing banks believe that 3DS 2.0 is a better solution to combat online payment fraud because as with the data provided by the payment networks [94][93] more than 80% of the CNP transactions are processed through frictionless authentication which does not require any OTP yet provides a secure way of making purchase online.

In the following section, we aim to assess the security provided by the 3DS 2.0 frictionless authentication for CNP transactions. Before we start our assessment process, the first challenge we have is - the definitions provided by 3DS 2.0 specification for frictionless authentication are not well defined. EMV 3DS 2.0 specifications does not define standards or methods on the operational workings of frictionless authentication.

However, section 3.3 (step 4) of EMV 3DS 2.0 specifications [75] hints that “The methods used and any data elements that are extracted from a customer machine during the 3DS 2.0 frictionless authentication are out of the scope of EMV specification and is left to card issuer's implementation [75]”. This implies frictionless authentication relies on some data extracted from the customer machine by the card issuing bank during the checkout process. The security of this data is crucial because it is this data that makes the transaction to go frictionless and authenticate the cardholder securely without any challenge.

7.2 Reverse Engineering Transaction Risk Assessment

3DS 2.0 specifies very little about how card issuers should implement Transaction Risk Assessment. To understand how merchants and card issuers assess the risk of consumer payments we therefore reverse engineer existing implementations. To that end we place a proxy between web browser and merchant/card issuer to intercept communication between payment initiator and authentication

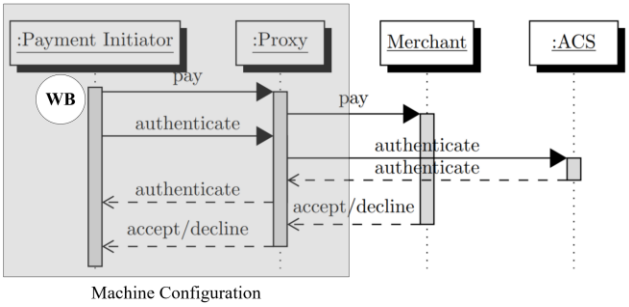

Figure 40 - Shows the reverse engineering set-up

service, as we describe in Section 7.2.1. Section 7.2.2 introduces the 3DS 2.0 transaction sequence, combining specification information with that obtained from the reverse engineering. Section 7.2.3 presents the data the risk assessor collects to inform its decision whether to challenge the customer. Section 7.3 discusses differences we found across the implementation of 3DS 2.0 we encountered.

7.2.1 Reverse Engineering System Set-up

Figure 40 shows the reverse engineering set-up. Within 3DS 2.0, a number of services and stakeholders are involved in the frictionless authentication protocol: the payment initiator using a browser, the merchant providing the check-out page at every purchase, and a set of services and servers for the authentication, termed the Access Control Server (ACS). The ACS maintains payment initiator's data which can be used to authenticate the cardholder during a purchase. The ACS functionality is usually carried out by or on behalf of the card issuer (such as Visa [66] or Mastercard [67]), but some large vendors like PayPal [96] and Amazon [111] use their own solutions.

To intercept communication, we use the Fiddler proxy, which is available as open-source at [115]. The proxy runs on the machine of the payment initiator. We configure the machine's web browser (WB) to send its HTTP(S) requests to Fiddler, which then forwards the traffic to the communicating merchant or ACS. The responses are returned to Fiddler, which passes the traffic back to WB. When HTTPS decryption is enabled, the Fiddler proxy generates a self-signed root certificate and a matching private key. The root certificate is used to generate HTTPS server certificates for each secure site that is visited from WB.

Apart from intercepting the browser communication we use two other techniques. First, using Fiddler, we challenge WB as if we were the merchant or the card issuer. Secondly, from Fiddler, we challenge the merchant as if the challenge was originating from WB. To handle ('tamper' in Fiddler terminology) a challenge, Fiddler provides a breakpoint function, which invokes a pause to the communication, as shown in Figure 41. Once paused, we can tamper or edit the changes to the communication data. In Figure 41 the arrow labelled '1' adds a breakpoint when the user navigates to the payment URL on the merchant website 'hps.datacash.com', and the arrow labelled '2' points to where we edit the communication data to the merchant. Using this platform, we are able (1) to sniff the communication, (2) control the input to WB and (3) control the output from WB. Implementations

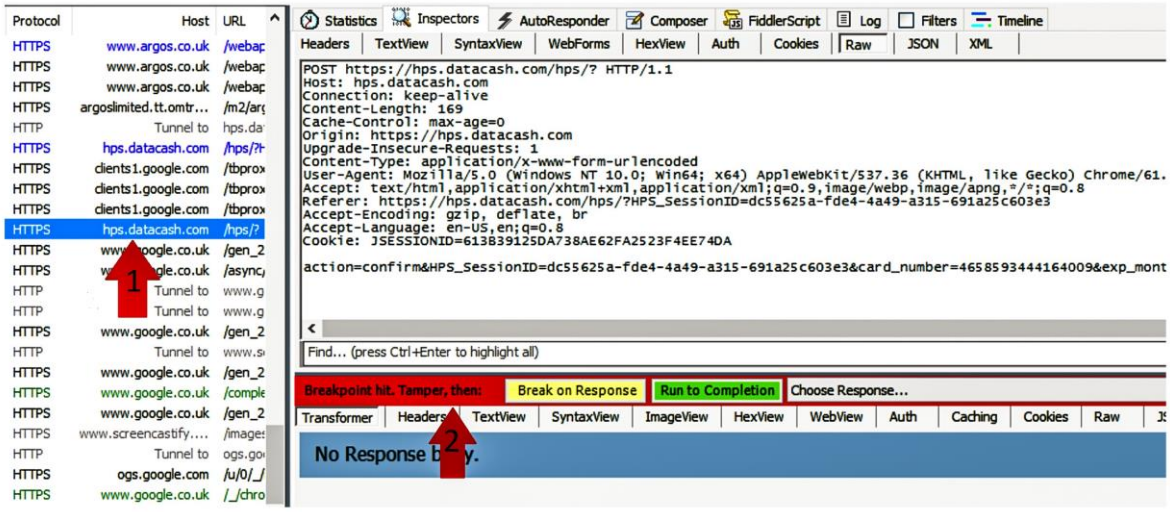

Figure 41 - Screenshot of Fiddler proxy tool

differ between different ACS providers, and for the same providers may differ depending on the merchant or cards. However, from our experiments it seems that there is considerable overlap between implementations. In total, we used five test cards for our experiments, three Visa cards (C1-C3) and two MasterCard cards (C4, C5). To make sure that 3DS 2.0 does not have any machine identifiers pre-installed on the machine, we had a fresh installation of Windows 10 operating system and Chrome 59.x web browser.

The merchant web sites we used were all enabled with 3DS 2.0 checkout and were selected from Alexa list of merchant web sites [116]. The 'Verified by (payment-network)' icon on the merchant web site indicates that it is 3D Secure enabled. To ensure that we have a representative sample of merchant websites, during our experiments we kept track of the ACS URL's to which our transactions were redirected. All 'Verified by (payment-network)' websites redirected us to the same ACS URL indicating that the implementation of 3DS is issuer based. For each test card, we made several legitimate transactions and recorded the complete checkout session for each transaction with Fiddler. We decided to stop making further transactions once authenticated by the ACS using frictionless authentication. This ensures that the ACS trusts WB enough for frictionless authentication. We decoded the 3DS 2.0 transaction data as necessary and analysed the outcomes in detail.

Using our test cards, we made several legitimate frictionless transactions on websites enabled with 3DS 2.0 checkout. Fiddler recorded the complete checkout session for each transaction which is later retrieved for analysis or decoding a part of 3DS 2.0 transaction visible from B1.

After assessing the checkout sessions obtained from Fiddler, we establish that card issuing banks have at least two variations in the implementation of frictionless authentication. We name these variations as *frictionless authentication over 3DS 2.0* and *frictionless authentication over 3DS 1.0*. In the following two sections, we will detail each variation of the frictionless authentication implementation. For each type, we start with providing a detailed description of the sequence of steps involved in the

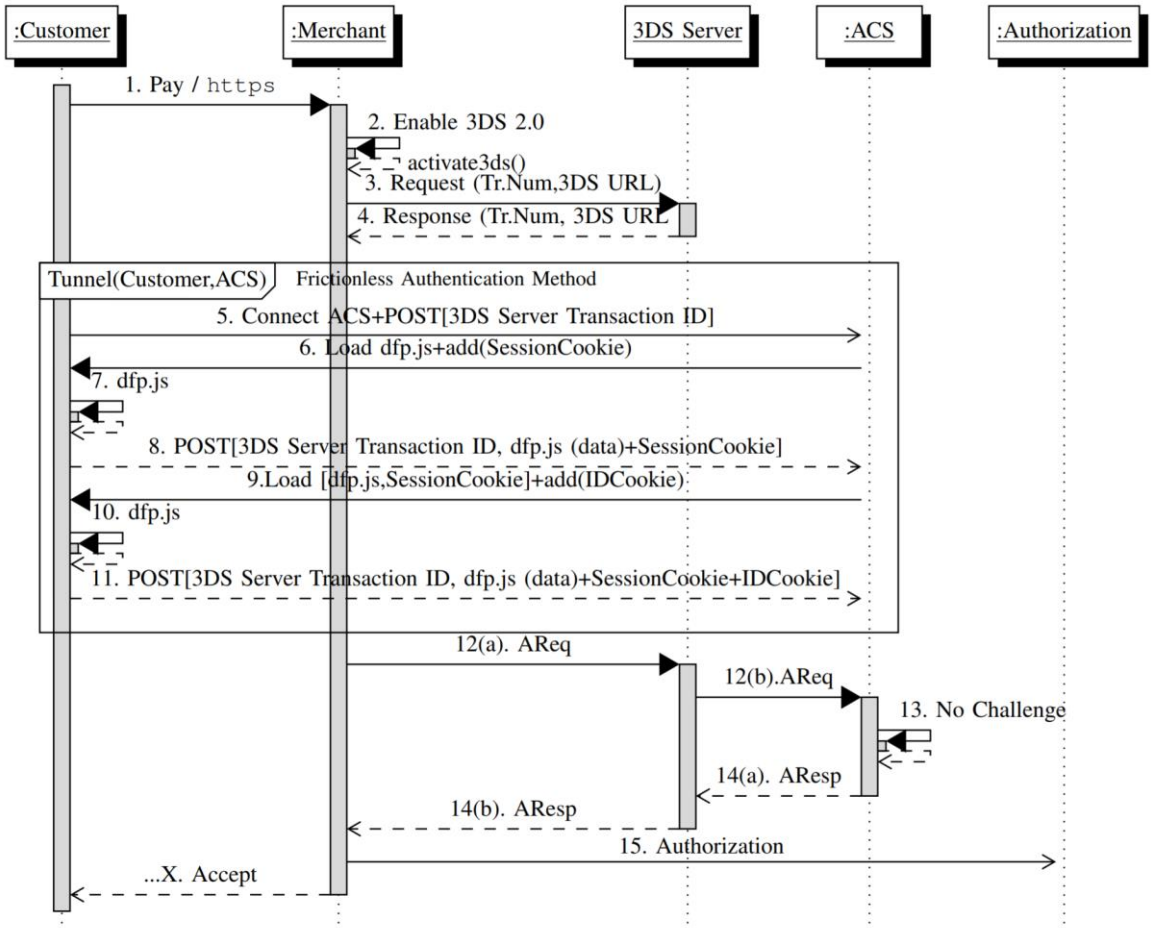

Figure 42 - Transaction sequence for frictionless authentication over 3DS 2.0

transaction. We will then explain the methods that are used by each implementation to perform frictionless authentication and finally we will detail the data that is used by each type to perform frictionless authentication.

7.2.2 3DS 2.0 Frictionless Authentication Protocol

Figure 42 shows the transaction sequence for frictionless authentication over 3DS 2.0, collating 3DS 2.0 specification with the transaction information extracted from Fiddler. The box labelled 'Tunnel (Customer,ACS)' represents the reverse engineered part of transaction visible from WB, while the transaction sequence steps for the rest of the parties are derived from 3DS 2.0 specifications.

Steps	Description
1	After filling the checkout page and as the 'Pay' button is clicked, the payment card details along with the Customer Web Browser's (WB's) HTTP headers are posted to the merchant.
2	The merchant decides to have user-authentication enabled for this transaction. The merchant enables 3DS 2.0 protocol.

3	For the payment card provided at the checkout, the merchant in this step requests the ACS version number, 3DS Method URL (address of the ACS) and a unique 3DS Server Transaction ID from the 3DS Server.
4	The 3DS Server responds back to the merchant with the ACS version number, 3DS Method URL and a unique 3DS Server Transaction ID. The merchant sends the ACS version number and 3DS Method URL to WB for routing to the ACS.
5	In step 5, WB is connected to card issuer ACS, and this connection is made via the 3DS Method URL received from the previous step. The 3DS Server Transaction ID along with WB's HTTP headers are passed to the ACS in this connection.
6	Having received WB's session and 3DS Server Transaction ID, the ACS adds a session cookie to the transaction and loads a JavaScript (named dfp.js) in B1.
7	WB executes the methods included in JavaScript (dfp.js) which has a logic to extract data from WB and post the dfp.js extracted data to the ACS (explained below the table is the analysis of dfp.js code).
8	As shown in the figure, through step 8, the dfp.js extracted data is posted to the ACS.
9-11	Interestingly, for first transaction, we found that steps (6 -8) were repeated to form steps (9-11). The only difference was observed in step 9 where the ACS additionally store two persistent cookies (ID cookies) in WB and fetch it in every subsequent transaction, as shown in step 11 of Figure 42.
12	Hereafter, the transaction is processed according to the rules specified in EMV 3DS 2.0 specifications that states the 3DS Server to submit an Authentication Request (AReq) to the ACS.
13	The Issuer performs Risk-Based Authentication (RBA) on the data it receives from the dfp.js and AReq and authenticates the customer at WB as the valid owner of the card. The Issuer's decision to not challenge the customer is added to Authentication Response (ARes).
14	The ARes is forwarded to the merchant via the 3DS Server.

Table 18 - dfp.js data extracted from B1 during frictionless authentication

Method	Attribute description	Source	Example values
nav.userAgent()	User agent(UA), OS	JavaScript	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36
Test()	Accepted MIME types/ Documents	HTTP header	text/html, application/xhtml+xml, application/xml;q=0.9, image/webp, image/apng, */*q=0.8
	Accepted Charsets	HTTP header	utf-8, iso-8859-1;q=0.5
	Accepted Encodings	HTTP header	gzip deflate
	Accepted Languages	HTTP header	en-US, en; q=0.8
	ActiveX, GeckoActiveX	HTTP header	?1:0
	Adobe Reader and components	HTTP header	?1:0
	XMLHttpReqest, Serializer, Parser support	HTTP header	Yes/No
deviceprint_browser()	UA(Version, cpuClass, language)	JavaScript	5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36; Win32; en-US
	navigator.appName	JavaScript	Netscape
	navigator.appCodeName	JavaScript	Mozilla
	navigator.appVersion	JavaScript	5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/61.0.3163.100 Safari/537.36
	navigator.appMinorVersion	JavaScript	5.0
	navigator.vendor	JavaScript	GoogleInc
	navigator.userAgent	JavaScript	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/65.0.3325.181 Safari/537.36
	navigator.oscpu	JavaScript	Windows NT 10.0
	navigator.platform	JavaScript	Win32
	navigator.securityPolicy	JavaScript	US & CA domestic policy or Export Policy
	navigator.onLine	JavaScript	True
	info_browser.name	JavaScript	Chrome
	info_browser.version	JavaScript	61.0.3163.100
	info_layout.name	JavaScript	Webkit
	info_layout.version	JavaScript	536.36
	info_os.name	JavaScript	win
	navigator.geoLocation	JavaScript	?1:0
deviceprint_display()	Screen's (colorDepth, width, height, availHeight, availWidth, HDPI, VDPI, Pixel Depth, ColorDepth, bufferDepth, FontSmoothing, Update interval)	JavaScript	2560*1440; 2560*1400; 24; 24
Window Information	innerWidth, innerHeight, outerWidth, outerHeight, length	JavaScript	675,473,1392,760,3
DoNotTrack	navigator.doNotTrack	JavaScript	?1:0
Useofadblock	alert_test	javaScript	?1:0
deviceprint_software()	Plugins installed	JavaScript	Adobe Acrobat, Macromedia Flash, Java, MS office, Cortana...
deviceprint_timez()	TimeZone	JavaScript	-60
deviceprint_java()	Java enabled	JavaScript	?1:0
	Java Supported	JavaScript	?1:0
	Java Version	JavaScript	Please fill
	javaScript cookies support	JavaScript	?1:0
	Server cookies support	JavaScript	?1:0
	HTTP only support	JavaScript	?1:0
Flash	Flash Version	FlashScript	WIN 28,0,0,126
	Flash Version	JavaScript	28,0,0
	Flash Details	FlashScript	Platform, Major Version, Minor Version, Capabilities (Audio, Accessibility, Audio support, MP3 support, Language, Manufacturer, OS, Pixel aspect, Color support, Dot per inch, Horizontal size, Vertical size, Video
	Number of Fonts	FlashScript	226
	List of Fonts	FlashScript	List of Fonts
deviceprint_cookie()	Cookie enabled	JavaScript	?1:0
	Session cookie	HTTP header	!yEpKXp9eMDojNcc7zSC3a5YqJJYDrqVB23 H1Cy/yThmhX+omXVM933/..... AIr8S7ldvbA==
	Test cookie	HTTP header	TESTCOOKIE=Y
	IDCookie	HTTP header	35BWzcxFkUu1aDdY%2B%2FxvL3VrDuvgo Xau%2FAgU%2BJqzYvZZoWiGPKKeYruvs GaPTeecduMcSLa%2FUlf1QGU07S89bddR 3dVSFT2dwVeUOd%2FkXvaw7JknHxjFlk4 ..... GY4I7drTK0nTCNJ%2BhHYW8Y5Wis%3D

7.2.3 3DS2.0 Transaction Risk Assessment Data

The reverse engineering exercise we conducted shows how the ACS builds up a fingerprint of the payment initiator's machine based on interaction with the browser. The ACS uses three pieces of information to establish a fingerprint for the payment initiator's machine:

1. the fingerprint information extracted from the browser using JavaScript
2. the 3DS 2.0 ID cookies fetched from the browser
3. the HTTP headers from payment initiator's browser forwarded by the merchant to the ACS as well as network data

In addition, the ACS may have other sources of data, such as customer or card type information or IP headers, but centre the discussion on the above three

7.2.3.1 Fingerprint Data using JavaScript.

The JavaScript fingerprinting scripts that we analysed contain functions to (i) collect browser-supplied information from the end-user device, and (ii) forward the collected data to the 3DS 2.0 server as a single Base-64 encoded string (the 3DS 2.0 specifications [75] requires all the data messages to be in Base-64 format). Table 18 shows an exhaustive list of device attributes from card C1 to C5 that are passed from WB to the ACS. The loading and execution of dfp.js by the ACS as a part of the checkout process is similar for all our test cards that we used. The 'Method' column indicates the functions implemented in the dfp.js that extract information from WB (for readability, in some cases we have simplified the method name). The details that are fetched in each function are shown in 'Attribute description' column of the table. The 'Source' column marks the origin of each attribute (JavaScript or HTTP). Finally, the rightmost column shows an example output value of each function. It would go too far to discuss all attributes in detail, but a number of them are particularly interesting. The data obtained is quite diverse, from browser and operating system information, to display, time, geo-location and some plug-in software information. What follows are the details description of the methods and attributes listed in Table 18.

- **deviceprint.userAgent().** B1 sends an HTTP request to the server it communicates. The request structure is defined as a part of HTTP protocol [16] and has a field userAgent that carries detailed information about the B1's configurations. This method returns the value of the user-agent field send by B1 HTTP request.
- **test().** Attributes within this methods extracts fields from the HTTP header and indicates to the server B1's support for MIME type, accepted: charsets, encodings, languages, support for activex, geckoactivex, adobereader and other components.
- **deviceprint_browser().** This method extracts more information about B1's settings. Some properties within this method navigator.appVersion, navigator.platform and navigator.browserLanguage extracts the B1 version, platform (operating system) on which the

B1 is installed and the language installed on the B1 and its operating system. It also has a geolocation attribute that checks if the location settings on a B1 is enabled.

- **deviceprint_display():** This function captures the B1 machine's detailed screen information like colour depth, screen width, height and available height.
- **window_information():** collects attributes like window's inner width, inner height, outer width, outer height and length.
- **DoNotTrack:** The attribute navigator.doNotTrack within this method retrieves B1 tracking preference. Do Not Track attribute is discussed in [117].
- **Useofadblock:** If an advertisement blocker plugin is installed on a user browser, this attribute will have a value of 1.
- **deviceprint_software().** This function captures all the browser plugins and types installed on the user machine. The navigator.plugin attribute extracts the full path of all the plugin files. However, the dfp.js script only extracts the filename ignoring the description and length or size if available.
- **deviceprint _timez() and deviceprint java().** These functions within dfp.js extract the client machine's timezone and verify if whether Java is installed/enabled.
- **Deviceprint_cookie().** This function has a logic to test whether the browser settings have cookies enabled.
- **Flash:** Along with dfp.js some frictionless implementation may additionally invoke a flash script on B1 browser. With our experiments, we found that only test card C4 had this script enable. This flash script can collect details about the status (enabled or disabled) and version of a flash plugin installed on a user browser. It also collects the number and complete list of fonts from the user browser. The collected attributes are again sent to the ACS in a base-64 format.

There are two other methods that we found were implemented in dfp.js: encode_deviceprint() and asyncpost_deviceprint(url)

- **encode_deviceprint()** combines the collected data into a single string. It formats the string by removing whitespace, add delimiters and other characters as requires by the ACS. Table shows an example output from encode_deviceprint function.
- **asyncpost_deviceprint(url)** posts the data to the ACS URL. The data is converted to base-64 before being sent as a form element to the ACS.

An example of resulting encoded device fingerprint is displayed in Figure 43.

Figure 43 - Device fingerprint information encoded and sent to ACS

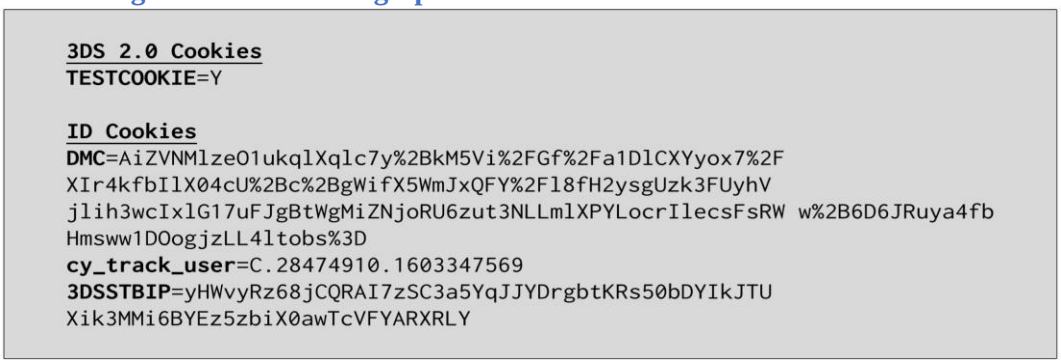

Figure 44 - Cookies installed by the ACS on our Machines

7.2.3.2 The 3DS 2.0 ID Cookies fetched from the browser

We found three types of cookies installed by the ACS on our machines. These are also described in Table 18, bottom rows. Full cookies are displayed in Figure 44.

- **Session cookie.** The cookie is deleted after a user closes the session.
- **Test cookie:** A test cookie with a name TESTCOOKIE and a value of Y was observed in an exchange during the transaction. This is set by the ACS server to determine if the user browser settings allowed cookies to be set.
- **IDCookies.** When the cardholder first enrols into the 3DS 2.0 system, a token in the form of ID Cookie(s) is placed on the cardholder browser. The number of cookies installed varied from one to three. These cookies we found have a validity of three years from the date of installation and also have an HTTP-only security tag. The HTTP-only tag on a cookie protects it from being accessed by cross-domain websites, meaning, only the website that has tagged the cookie on the cardholder browser can access this cookie.

```
HTTP headers
POST https://hps.datacash.com/hps/? HTTP/1.1
Host: hps.datacash.com
Connection: keep-alive
Content-Length: 151
Cache-Control: max-age=0
Origin: https:hps.datacash.com
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla5.0 (Windows NT 10.0; Win64; x64) AppleWebKit537.36
(KHTML, like Gecko) Chrome59.0.3071.115 Safari537.36
Content-Type: applicationx-www-form-urlencoded
Accept:                                  texthtml,applicationxhtml+xml,
applicationxml;q=0.9,imagewebp,imageapng ,;q=0.8
Cookie: JSESSIONID=2DA07357AEFF38EE6A8DEDD1FCC96628

Card details
card_number        =4xxx-xxxx-xxxx-xxxx&exp_month=04&        exp_year=
2020&Name=Mxxxxxxxxx&cvv2_number=271
```

Figure 45 - Browser information passed on from merchant to ACS

Understandably the dfp.js script data and the persistent ID cookies generate a fingerprint of the B1 profile that will enable the card issuer ACS to frictionless-ly authenticate the cardholder at the checkout. Further examination of fingerprint data reveals that the ACS makes use of the Browser Object Model (BOM) [118] which is a larger representation of information provided by the browser through JavaScript.

The BOM deals with browser components like history (limited), location, navigator, screen any other detailed functionality the browser may expose to JavaScript. Because no standards exist for the BOM at the time of this writing, each browser has its own implementation of BOM.

7.2.3.3 Data Passed from Merchant to ACS.

Data passed by the merchant in AReq message (step 12 of Figure 39) contains elements that identify payment initiator browser configuration. For instance the EMV 3DS 2.0 specifications [75] (Table A.1), mentions the merchants to include browser accept headers, language, screen details and user agent to be passed in the AReq message whenever available. The browser configuration helps the ACS to render correct iframe for the cardholder device and may be used by the ACS to compare the information passed with dfp.js. To inspect the methods by which the merchant collects data to frame the AReq message, we referred to the merchant developer guides from payment networks Visa [65] and MasterCard [70] and payment service providers like PayPal [96], which suggest to use the HTTP headers passed on by the merchants during checkout as part of WB's authentication data.

7.3 Frictionless Authentication Over 3DS 1.0

Figure 46, shows a sequence diagram detailing the process of frictionless authentication over 3DS 1.0 that we obtained from test card C1 and C3. To have a clear understanding of the process the sequence diagram above collates the information extracted from Fiddler and 3DS 1.0 specification. Data from Fiddler represent the part of the transaction visible from B1.

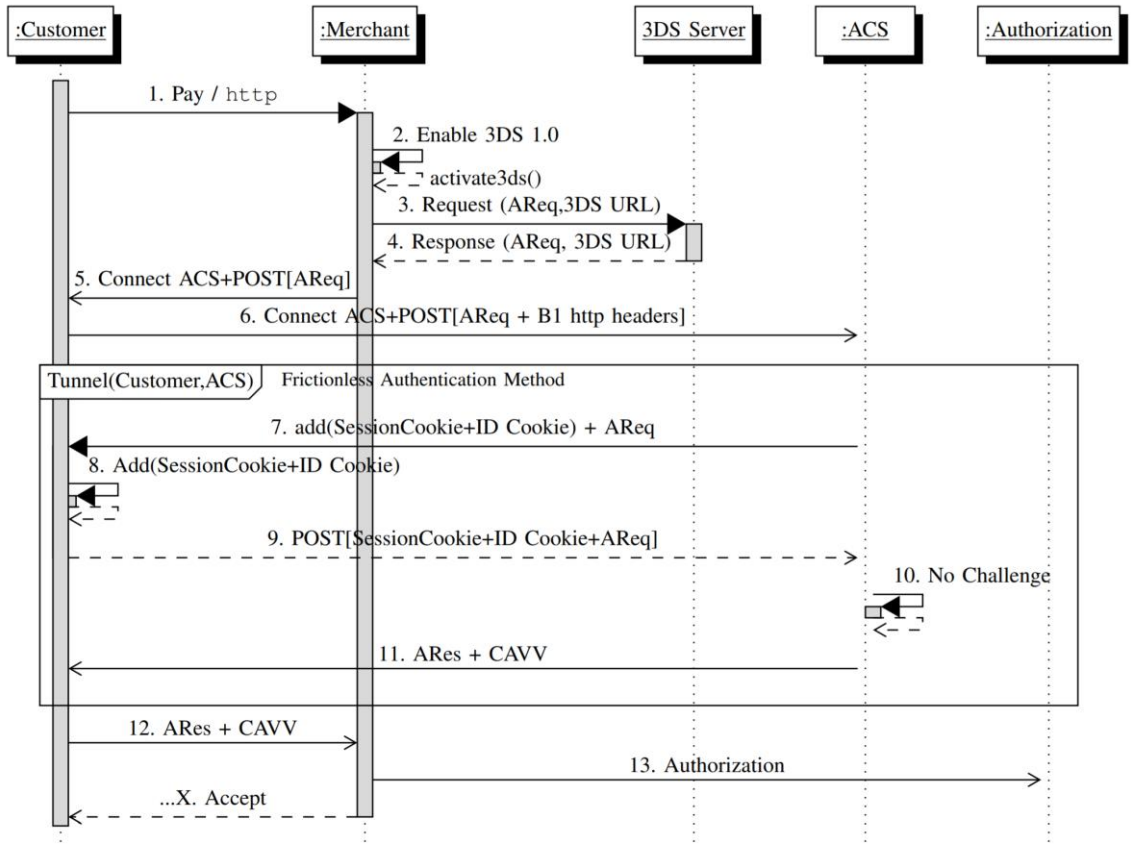

Figure 46 - Frictionless flow sequence diagram

Steps	Description
1	After filling the checkout page and as the 'Pay' button is clicked, the payment card details along with the Customer's Web Browser's (WB's) HTTP headers and merchant session cookies are posted to the merchant.
2	The merchant decides to have user-authentication enabled for this transaction. The merchant in this case enables 3DS 1.0 protocol. Although the 3DS protocol version is internal to the merchant implementation, we can derive the 3DS version number from the data passed in step 5 through the WB.
3	The transaction is processed according to the 3DS 1.0 specifications. For the payment card provided at the checkout, the merchant in this step requests the 3DS Method URL (address of the ACS) and AReq message from the 3DS Server. The data that is passed from the merchant to the 3DS Server includes merchant data and transaction data.
4	The 3DS Server responds back to the merchant with the 3DS Method URL and an AReq message.
5	The merchant sends the 3DS Method URL and AReq message to the WB for routing it to the ACS.

6	The WB is connected to card issuer ACS, and the connection is made via the 3DS Method URL received from the previous step. The AReq message along with WB's HTTP headers are passed to the ACS in this connection.
7	Having received the WB's session and AReq message, the ACS examines the data received in AReq message and adds session and ID cookies to the transaction.
8	The WB is tagged with the cookies from the ACS
9	Interestingly, for the first transaction, we found that steps 6 was repeated as step 9. The only difference observed was, in step 9, ID cookies were fetched by the ACS along with AReq and WB's HTTP headers
10	The Issuer performs Risk-Based Authentication (RBA) on the data it receives from the AReq message, WB's headers and ID cookies and authenticates the customer at WB as the valid owner of the card. The Issuer's decision to not challenge the customer is added to Authentication Response (ARes).
11-13	The ACS after authenticating the customer, responds back with the ARes message and a CAVV.

7.4 Discussion of 3DS 2.0 Implementations

There exist a number of notable differences between different implementations of 3DS 2.0. These differences can be categorized as follows:

1. Difference in the use of 3DS protocol version
2. Difference in transporting the device fingerprint: obfuscated versus plain-text
3. Difference in amount of data collected as a fingerprint: JavaScript based versus HTTP headers and cookies only.

7.4.1.1 Difference in the use of 3DS protocol version

We observed that the ACE associated with card C2 use an extended version of 3DS 1.0 protocol for frictionless authentication. The optimisation is achieved by adding an extra layer of frictionless authentication over the 3DS 1.0 protocol. As opposed to 3DS 2.0, in Message 4-5, the browser collects and submits the AReq message with the transaction identifier, as defined in the 3DS 1.0 specification. Through Message 6 and 8, using device fingerprinting JavaScript, the ACS collects the fingerprint data from the

browser. Like 3DS 2.0 frictionless authentication, if this is the first 3DS 1.0 transaction from the machine, the ACS repeats the messages 9-11 to install IDCookie. Hereafter the transaction is processed according to the 3DS 1.0 specification. The ACS decision (to not challenge) is added to the ARes which is then forwarded to the merchant via the browser. More importantly, comparing the

frictionless authentication of 3DS 1.0 and 3DS 2.0, both of these protocols capture static fingerprint data in base-64 encoded format and use HTTP-only IDCookies for transaction risk assessment.

7.4.1.2 Difference in device fingerprint implementation

The other variant that we observed is in the technique with which the device fingerprint JavaScript is implemented. In two cases (C2 and C5) we noticed that code obfuscation techniques were applied to make the JavaScript difficult to read and analyse. However, obfuscated codes has certain general limitations, in that, it is an encoding technique (not encryption) and needs to make sure that the code does not loose its functionality when executed over the system. The 3DS 2.0 device fingerprint JavaScript can still be run to obtain base-64 device fingerprint values. Additionally, code obfuscation is a technique that has long been used by malware writers to hide their malicious code. Therefore, there are plethora of security tutorials and freely available security tools designed to de-obfuscate JavaScript. The most reliable deobfuscater that we discovered for our research is available as open source from Intelligent Systems Lab, Zurich [119].

7.4.1.3 Difference in amount of machine data collected

Table 18 shows an exhaustive list of all the data elements collected by the fingerprinting scripts and HTTP headers, the amount of data collected by each implementation of the JavaScript varies and in some cases to a substantial extent. We noticed that some of the card issuers have no device fingerprinting JavaScript implemented at all. For example, card issuer of C3 implements frictionless authentication over 3DS 1.0 and only relies on the data received in the AReq message. The transaction sequence that we obtained with C3 is detailed in Section 7.3 Frictionless Authentication Over 3DS 1.0.

In this chapter we focus predominantly on the transaction phase of the 3DS 2.0 protocol, in which device fingerprint and ID cookies are collected and then used together with the transaction data to decide whether to challenge the payment initiator. However, the 3DS 2.0 protocol also defines an enrolment phase during which the card issuer firstly collects the fingerprints from the card issuer computer and signs the fingerprint data to create ID cookies. The card holder computer is then 'tagged' through the usual cookie mechanism with these ID cookies. This enrolment phase is imperfect, in that it cannot be determined if the payment initiator who enrols a certain card is a legitimate user of the card–this vulnerability existed in 3DS 1.0 as well.

7.5 Conclusion

Our reverse engineering exercise on 3DS2.0 protocol indicates several inconsistencies in the EMV 3DS 2.0 and regulatory standards which are listed below:

- The PSD II standards do not regulate the technical requirements transaction risk assessments. This leaves the protocol designers and the merchants to have their own
- The EMV 3DS 2.0 standard, does not define the data elements to be used for frictionless authentication for browser-based transactions.

- Frictionless authentication is optional for the card issuing banks and card issuing banks even being compliant to the PSD 2 by adopting EMV are unsure of how to consistently perform frictionless authentication

Also with 3DS 2.0, access control policies can only be defined for cookies which can be marked 'http-only'. Apart from this the frictionless authentication data simply relies on the data extracted by java scripts. This raises a question on the security of frictionless authentication. Although the transfer of javascript data is SSL protected, the actual data is a plaintext base-64 data which can be easily be extracted by malicious websites using the same javascript code as used by the issuer ACS. This will have the device fingerprint data in the same format as accepted by the ACS. There is no uniqueness to the transaction leaving from the cardholder machine. 3DS 2.0 was introduced to defeat malware's that infests more PCs to steal user payment data. However, the implementation of frictionless authentication is not secure as it relies on static and plain text data stored as persistent cookies.

Chapter 8. Designed to be broken: Vulnerabilities and Attacks on 3DS 2.0

In this chapter we device a realistic impersonation attack, where an attacker uses obtained data described in section 7.2.3 and avoids being challenged for a second factor of authentication information. We first describe the precise attack model in Section 8.2, and then explain in Section 8.2.2 how the attack can be implemented, particularly related to obtaining the data. We carried out a number of experiments with different machines to demonstrate that the impersonation attack indeed succeeds, as we will describe in the rest of this chapter. Lastly, in section 8.5, we identify a betrayal attack in which an attacker can perform friendly fraud to cause an actual business loss to the online merchant. Let us start with the introduction to the cardholder impersonation attack on 3DS 2.0 protocol.

8.1 Introduction – Cardholder Impersonation.

The design of 3DS 2.0 also suggests an obvious vulnerability, in that the authentication service may decide incorrectly not to challenge a payment. We will demonstrate an impersonation attack, in which a perpetrator impersonates a payment initiator, thus 'tricking' the authentication service into allowing a transaction to complete without being challenged for second factor authentication information. This attack is practically feasible if one manages to install malware or similar on the payment initiator's machine to obtain the necessary data. A main function of the malware would be to recreate the machine fingerprint and then transport the fingerprint to the attacker, who can use it for a purchase impersonating the card holder (see Section 8.2 for details). By the design characteristics of 3DS 2.0, the security level of 3DS 2.0 then defaults to that of old-style authorization-only online payment.

We complete the security analysis of 3DS 2.0 by comparing different payment protocols, and by discussing design alternatives. From this analysis we conclude that alternatives to 3DS 2.0 exist but that the challenge remains to effectively balance security and usability on browser platforms. An interesting question is whether, in hind-sight, the regulator (who originally proposed stricter two-

factor authentication in PSD II) is satisfied with emerging approaches from the industry to transaction risk assessment, such as those reported in this chapter.

8.2 Attack Model – Cardholder Impersonation

The objective of the attack is to use the credit card of another party to successfully complete an online purchase, despite the fact that the merchant uses 3DS 2.0. We assume that the attacker has no manner in which it could respond successfully to a challenge for a second factor of authentication information. Therefore, the objective of the attacker is to avoid a challenge and be allowed to complete a frictionless transaction. We consider the attack successful if an attacker avoids being challenged in situations the ACS actually wants to challenge. To demonstrate the success of the attack, we therefore have to show that there exist transactions that trigger a challenge in normal operations, but that allow the attacker to complete the transaction without challenge. To succeed, the attacker needs to obtain the credit card details, the cookies and the fingerprint data used for Transaction Risk Assessment, as described in section 7.2.3. We do not assume any insider administrative access privileges of the attacker, neither at the payment initiator's machine nor at any of the 3DS 2.0 services. The attack assumes a perpetrator manages to install malware or plug-in that collects the necessary data from the payment initiator's machine, which includes running the JavaScript fingerprinting scripts–we will argue in the next section that that is not far-fetched. Shipping this data to the attacker allows the attacker to impersonate the cardholder's identity by crafting its 3DS 2.0 authentication data to be identical to the 3DS 2.0 frictionless authentication data belonging to that of the payment initiator.

8.2.1 Attack Implementation

The attack implementation needs to complete two stages: (1) obtaining the card and transaction risk assessment data, and (2) using the card and transaction risk assessment data.

Obtaining Card and Transaction Risk Assessment Data. In this stage, the attacker needs to obtain credit card details and machine fingerprint data (including cookies). There is a variety of reasons why this can only be done through a Man in the Browser. A challenge is that the ID cookies (see Section 7.2.3) are http-only protected, that is, they cannot be read by any cross-domain web pages or through JavaScript. Browsers allow access to http-only cookies to extensions (including malware) because extensions are considered "trusted" once installed, whereas regular JavaScript is not. Cross-site scripting (XSS) [96, 97, 98], in which a script from a web site different than the merchant or 3DS 2.0 server attempts to access information such as cookies, is therefore not possible. The most feasible attack will therefore be through making the browser user install malware or similar in the browser, which obtains the data from the browser.

The most basic approach to obtain the required data is a browser plug-in that can sniff the browser communication to steal http-only cookies, record keystrokes to steal user payment data and execute

device fingerprint JavaScript to capture the device fingerprints. More advanced malwares incorporate also have such features, and are commonly available at [123], see for instance the ZeUS, SpyEye, Dridex and Tinba malwares. Once such malware is installed, it has an ability to obtain card transaction data for a purchase, the associated transaction risk assessment data described in section 7.2.3, as well as the http-only cookies [100–102]. Malware SpyEye, for example, gets into a browser by prompting them to install a pdf reader or a flash player plug-in. Once into the browser, it updates itself as needed to configure fake entity certificates into the browser storage, record keystrokes, sniff the browser communication, records browser sessions and also can capture screen shots of user screen [127][128].

Using the Obtained Card and Transaction Risk Assessment Data. The task in exploiting the obtained data is to impersonate the card holder in the attacker's browser. The attacker copies the cookies to their own browser, and initiates a transaction with the merchant of choice, even if the merchant uses 3DS 2.0. It also receives credit card details and machine fingerprint data, per the above. At payment, the attacker creates or replays the correct responses in the protocol of Figure 42. Since there is no randomness in the fingerprint data, the same string of dfp.js data and HTTP headers obtained from the payment initiator's machine can be replayed on the attacker's machine using Fiddler (if required). To tamper the data using fiddler breakpoints are added whenever the merchant and the ACS connects to the attacker's browser.

8.2.2 Attack Demonstration

The demonstration of the attack aims to identify if it indeed is possible to impersonate from a different machine a legitimate payment initiator. In this demonstration we use the data obtained from machine M1, using the experiment set-up from Figure 40. We randomly selected a merchant with 3DS 2.0 enabled checkout and repeated transactions using all test cards C1 to C5 until M1 was trusted enough for frictionless authentication. The payment sessions made from M1 were recorded by the Fiddler proxy and were reused on different machines M2 and M3.

The detailed configuration of Machine M1 and M3 is shown in Figure 6. It can be seen from the figure these machines were configured with chrome 59.x browser and the same version of operating system i.e. Windows NT 10.0. Machine M2 differed both in operating system and the web browser configurations. For M2, the operating system is Windows 8 and the web browser installed Mozilla FireFox. We note that M2 and M3 were on networks different from M1, so that the IP source address is different.

The approach behind our experiments is as follows. We conduct the experiment for the five credit cards mentioned. First, we ran an experiment to verify that transactions from machine M2 with the five cards are indeed challenged if one only enters card information (and does not impersonate the card holder with the risk assessment data). This verification was successful in all of the cases except

for card C1 where only lower value transactions below £10 were approved (we will get back to this in the next section). Then, we ran an experiment in which we used the obtained transaction risk assessment data to impersonate the card holder, to see if we were allowed to complete the purchase unchallenged, i.e., in frictionless mode. We initiated transactions where we selected products with values ranging between a £1 to £300, on an online merchant that uses 3D Secure 2.0 at checkout.

We were successfully able to execute the attack for all our test cards (C1-C5), in that the transactions were approved without any challenge by the card issuing bank's ACS. Interestingly, only for test card C5, the card issuer ACS issued challenges when the value of transaction reached above £200 (a typical transaction threshold set for frictionless authentication).

We ran a second experiment, using different machine M3, with the same hardware and software as M1. In so doing, we wanted to see if different machines construct the same Fingerprint data. This is to simulate a scenario where an attacker is unable the device fingerprint data but was able to get the ID cookies. In that case, we did not use the Fiddler proxy to recreate the M1 Fingerprint data, but simply completed the transaction normally. In all cases, the transactions were allowed to go on without being challenged. Close inspection of the data that M2 sent to the merchant and ACS revealed that the transaction risk data was essentially identical for M1 and M2.

Reflection. For consumers it would be important to know how merchants and card issuers respond if the above attack took place. To that end, we communicated with the card issuing banks to understand how it would react if we were to report the fraudulent transactions that were made from the attacker machine. The card issuer for C3 asks cardholders to identify some previous transactions made from the victim's machine and would not register the transactions made from attacker machine as fraud. The card issuer for C3 also blocks and re-issues a new payment card to the card holder. However, in two cases (C4 and C5), the card issuer argued that the transactions must have originated from the actual card holder's machine. They argued the card holder is trying to perform a 'friendly fraud', and so is denied a refund of any reported losses. This chapter makes clear that this situation can easily arise when the situation is in fact an actual impersonation attack which can create an actual monetary loss to the cardholder.

8.3 Further Re-Engineering of Transaction Risk Assessment

Section 7.2.3 established which data 3DS 2.0 implementations used in their transaction risk assessment, and we showed that with that data alone, one can execute an impersonation attack. However, this does not yet provide us with full understanding of the way risks are being assessed by the ACS. First, the ACS may use additional sources of data, for example, it may use header info from the protocol stack such as the IP source address or some other data about the card holder available from the card issuer. Secondly, the ACS will set certain rules about when to invoke a challenge. These

Table 19 - Experimental simulation and results with C1 and C2

Transaction number	Scenario	Machine data	Cookie ID	Value (£)	Region	Website	Card	Challenged?	Transaction status	Blocked
T1	S1	✓	✓	10	✓	W1	C1	×	Approved	×
T2							C2	×	Approved	×
T3						W2	C1	×	Approved	×
T4							C2	×	Approved	×
T5	S2	✓	✓	309	✓	W1	C1	×	Approved	×
T6							C2	×	Approved	×
T7						W2	C1	×	Approved	×
T8							C2	×	Approved	×
T9	S3	✓	✓	10	×	W1	C1	×	Approved	×
T10							C2	✓	Declined	×
T11						W2	C1	×	Approved	×
T12							C2	×	Approved	×
T13	S4	✓	✓	309	×	W1	C1	×	Approved	×
T14							C2	✓	Declined	✓
T15						W2	C1	×	Declined	×
T16							C2	✓	Declined	✓
T17	S5	×	×	10	✓	W1	C1	×	Approved	×
T18							C2	×	Approved	×
T19						W2	C1	×	Approved	×
T20							C2	✓	Declined	×
T21	S6	×	×	309	✓	W1	C1	✓	Declined	×
T22							C2	✓	Declined	✓
T23						W2	C1	✓	Declined	×
T24							C2	✓	Declined	✓
T25	S7	×	×	10	×	W1	C1	×	Approved	×
T26							C2	✓	Declined	×
T27						W2	C1	×	Approved	×
T28							C2	✓	Declined	×
T29	S8	×	×	309	×	W1	C1	✓	Declined	×
T30							C2	✓	Declined	✓
T31						W2	C1	×	Declined	✓
T32							C2	✓	Declined	✓

rules will stipulate which data to consider when checking the fingerprint and specifies bounds on data outside which the transaction will be challenged (e.g., a limit for the transaction amount).

There is a number of questions of interest motivating further re-engineering of the risk assessment approach. First, it provides information about which variants of the impersonation attack would succeed and thus allows us to assess the security and risks behind online payment. Secondly, it serves as a suggestion for a possible methodology to assess consumer implications of Transaction Risk Assessment. Approval of TRA as a mechanism shifts liability to the card issuer but nevertheless still exposes consumers to possible distress when an impersonation attack is carried out. Arguably, it would be in the interest of the public if there is visibility in the implementation of Transaction Risk Assessment. The re-engineering experiments in this section demonstrates how to provide such visibility.

The experiments in this section obtain responses from the ACS for transactions in 8 different scenarios. These scenarios provide all combinations of the following three features:

1. submitting the machine data and IDCookie or not (from Section 7.2.3.1 and 7.2.3.2)
2. submitting different transaction values
3. submitting transactions from different regions

Table 19. shows selected results from our experiments on two test cards C1 and C2. Our set-up was identical to Section 7.2.1, with data obtained from machine M1 used on an alternative machine M2. Payments were initiated on two merchant websites (W1 and W2) that enforce 3DS user authentication. W1 is a web merchant local to country of where the victim card is issued and W2 is an overseas merchant for a victim's card.

The columns specify the scenario. For instance, Scenario S1 copies the machine data and the ID Cookie, for a low value transaction, within the region. With respect to the region, experiments for C1 and C2 were made from UK and Germany. Region (✓) indicates the transaction attempts were made from same country as of the Victim.

We see from Table 19 that different card issuers make different risk-based decisions. In particular, card issuer for C1 provides more frictionless options and is more focused on providing convenience to the customer. Whereas, the card issuing bank for C2 makes more security-oriented decisions, challenging the payment initiator more often. Interestingly, comparing transaction T4 and T10 from C2 it can be derived that C2 card issuer challenges every transaction if the web merchant is established in a country overseas to C2.

Figure 47 and Figure 48 summarise the findings of Table 19. The 'states' are phases in the 3DS 2.0 transaction, where Pay indicates initiating payment, while the other refer to possible outcomes, either approved, challenged/declined or blocked. Note that for our purposes we do not have to differentiate between challenge and declined, they both imply that the transaction has not gone through as frictionless.

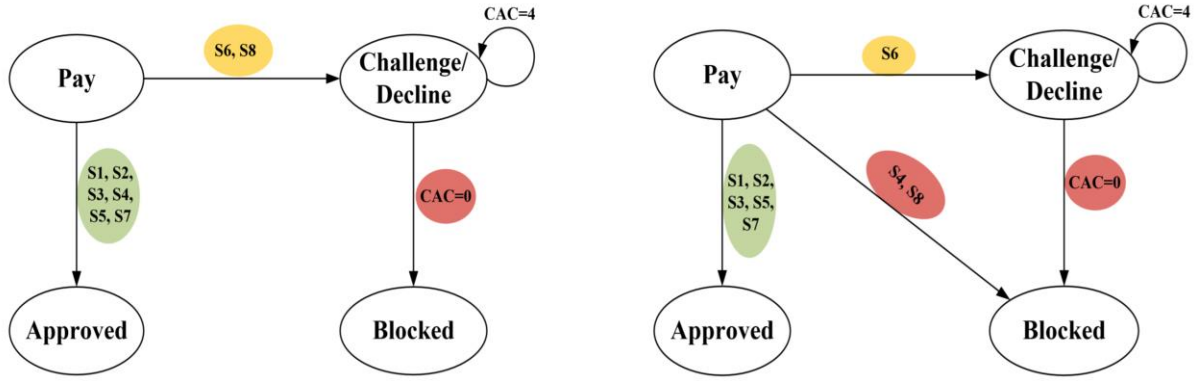

Figure 47 - Summarising C1's risk assessment outcomes over merchants W1 and W2

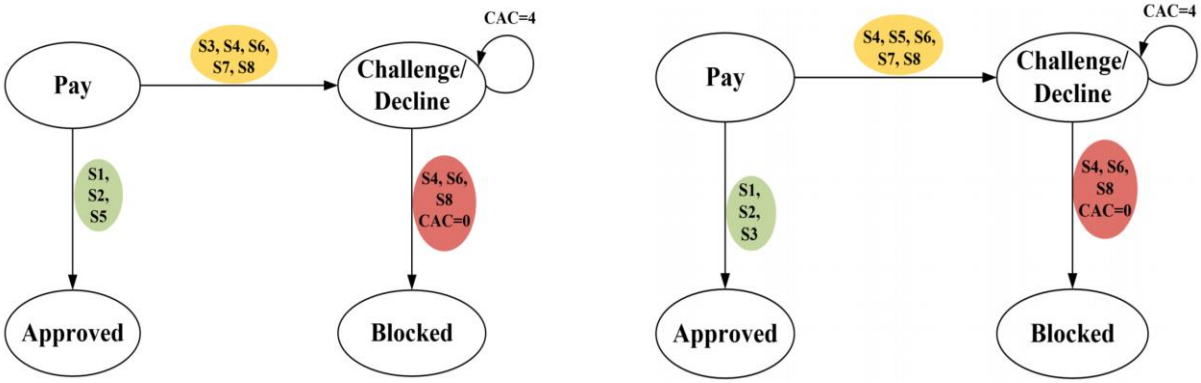

Figure 48 - Summarising C2's risk assessment outcomes over merchants W1 and W2

The arcs are labelled with the scenario given in the second column of Table 2. CAC stands for challenge limit counter, which counts down from the limit to zero. Here, the limit is 4, and at the fifth attempt the card is blocked.

8.4 Discussion of Card Payment Systems Security

In this section we discuss security of card payment systems from various angles. First, we discuss in Section 8.4.1 practical aspects of the attack on 3DS 2.0 introduced in this chapter, and will see that strong security solutions can be devised but for various reasons have not gained widespread adoption of the technologies. In Section 8.4.2 we discuss 3DS 2.0 in relation to the various security approaches used over the years for card payment.

8.4.1 Impersonation Attack in 3DS 2.0

The problem of authenticating cardholders in the online payment system is exacerbated by the desire to cause minimal friction during the checkout. The introduction of 3DS 2.0 addresses this security/usability challenge through the use of Transaction Risk Assessment, and it is clear that the industry strongly favours such risk-based approaches, given that in the US about 75% of the card issuers have adopted risk-based authentication [129]. However, as we have seen in this chapter, the remaining security bottleneck is the secure storage and transfer of machine authentication data and http-only cookies from the customer machine to the authentication service.

Once 3DS 2.0 is common and authorisation-only transactions can no longer be exploited, the impersonation attack presented in this chapter is potentially attractive for perpetrators. Its net effect would be that perpetrators can use stolen 3DS 2.0 frictionless authentication data in online shops without the cardholder being negligent, exactly as was the case with authorisation-only systems before the introduction of 3D Secure. The attack does not require to synchronize fraudulent purchase with that of an unwitting customer (as a relay attack would). Malware could easily be designed to sniff the 3DS 2.0 transaction data and later forward it to the attacker server. In fact, there are a number of such open source browser extension available and installed be thousands of browsers, e.g., HTTPWatch [130] and LiveHTTPHeaders [131]. Other developments, such as FraudFox [132], are also cause of concern. FraudFox aims to make it faster and easier to change a browser's fingerprint to one that matches that of a victim, for instance through profile generator scripts.

Attempts to complicate executing the attack through JavaScript obfuscation, as some implementations do, cannot be expected to be of much help. Techniques exist that mask the JavaScript functionality and prevent it from being stolen (code stealing). JavaScript obfuscation typically works by replacing the white spaces in the JavaScript code and renaming the variable and function. There exist several tools and tutorials on the Internet which can be useful to re-establish the original data and script obfuscation is therefore far from sufficient.

More helpful is the manner in which cookies are stored in the observed implementations. All ID cookies we discovered were secure enabled, which means the cookies are only passed on secure connections (HTTPS). Secondly, the cookies were tagged http-only, which implies that the cookie is not readable to JavaScript. This prevents the cookies from being accessed by the cross-domain websites, i.e., prevents cross-site scripting attacks (XSS). Nevertheless, cookie storage in browsers remains non-secure unless the machine uses secure storage.

Technologically, an obvious solution for secure transfer would be to use private/public key approaches to encrypt and sign messages between the payment initiator and the 3DS Server. However, for such a solution to gain acceptance would require a separate trusted secure storage environment for cryptographic keys and certificates. The payment industry standards [21] require payment credentials, including keys and certificates to be stored in 'TamperResistant Security Module' which is defined as the set of hardware, software, firmware, or some combination thereof that implements cryptographic logic or process (including cryptographic algorithms and key generation) and is contained within the cryptographic boundary. Today's computer systems and their software systems are not provably secure enough. This issue has come up before, when Google first introduced Android pay with the concept of Host Card Emulation with Android KitKat 4.4 [133] in 2014. The key storage security model for Host Card Emulation was software controlled and contained the threat that an attacker may compromise the mobile OS to steal the credentials. This approach was therefore not found suitable to host EMV payment applications [57].

Compared to the desk top browser, the situation with respect to security is more positive when using mobile payment apps. Using hardware security elements [57][134] one combines aspects of tamper resistance and payment security with tokenization [134]. With tokenization, the actual card number is replaced with a 'meaningless' token and each transaction performed is signed with securely stored device-specific keys. Not surprising, EMV supports this approach to security of payments and defined a 3DS 2.0 SDK [75]. Such approaches do not allow attackers to steal the payments credential or any party intercepting communication to alter the transaction data and so is not vulnerable to the impersonation attack discussed.

With the desktop browser being the security bottleneck, the industry is working on evolving the security of client authentication mechanisms on desktop platforms. Apple recently introduced biometric fingerprint readers and secure enclaves in their laptops [134], which meets payment industry standards for storing payment credentials. Lenovo, alongside with Intel [135], and payment service provider PayPal are already working with Fast Identity Online (FIDO) [136] to develop PKI based solution for the customers identity using a hardware security module.

8.4.2 Security Solutions across Card Payment Systems

Table 20 provides an overview of the security techniques utilized in various card payment protocols that have been in us. The protocols considered are the following:

1. Card Present technologies:
    a. Magnetic stripe
    b. EMV Chip and PIN, with three data authentication variants: static (SDA), dynamic (DDA) and combined (CDA)
    c. EMV Contactless, with three data authentication variants (SDA, fast DDA and fast CDA)
2. Card Not Present technologies:
    a. 3D Secure 2.0
    b. 3D Secure 1.0
    c. Chip Authentication Programme or CAP (generates one-time tokens)
    d. Transaction Authentication Number or TAN (a variant of CAP mostly used in Germany)

We compare the technologies with respect to eight types of techniques they may or may not provide. These are:

1. **Payment Data Confidentiality techniques.** These ensure that the cardholder payment data (authentication data, cryptographic keys) or any data used to access a cardholder account are not made available to unauthorised individuals [11].

Table 20 - Comparison of card payment protocols and their security features

	Mag Stripe	EMV Chip & PIN			EMV Contactless			3DS 2.0		3DS 1.0	CAP	TAN	Attacks
		SDA	DDA	CDA	SDA	fDDA	fCDA	Chal.	Fric.				
Data confidentiality	×	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓	ID theft / Impersonation [42]
Data Integrity	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Cloning / Skimming [27]
Card / machine authentication	×	×	✓	✓	×	✓	✓	✓	✓	×	✓	✓	Cloning / Skimming [5]
Transaction authentication	×	×	✓	✓	×	✓	✓	×	×	×	✓	✓	ID theft / Replay[10]
Second Factor User authentication	✓	✓	✓	✓	×	×	×	✓	×	✓	✓	✓	ID theft [12]
Payment authentication	×	×	✓	✓	×	×	×	×	×	×	✓	✓	ID theft / Impersonation [5, 10]
Non-repudiation	×	×	✓	✓	×	×	×	×	×	×	✓	✓	Friendly Fraud[27, 45]

2. **Payment Data Integrity techniques.** These ensure that only authorised parties (cardholder or card issuer) are able to modify the payment data when stored and transmitted.
3. **Card Authentication / Machine Authentication approaches.** These confirm the identity of the card (token) or, when appropriate, the machine (computer of the payment initiator) that is linked to the cardholder account.
4. **Transaction Authentication.** These authenticate specific transaction, typically cryptographically binding a random one-time code to the data of the transaction, achieved by the card or machine signing the transaction data [30].
5. **Second Factor User Authentication.** These are additional user authentication techniques that use information about a second authentication factor to confirm the identity of the cardholder.
6. **Payment Authentication.** Techniques to ensure that the origin of any payment transaction message is correctly identified, combining (i) Card / Machine Authentication and (ii) Transaction Authentication.
7. **Payment Non-repudiation techniques.** These ensure that the customer / cardholder cannot deny the fact that they completed a transaction. When required, they must be able to provide a proof of the transaction. Typically, this is achieved by signing the transaction data with user specific keys.

Table 20 details the comparison of card payment protocols and the security technologies they utilise. The table also highlights reported attacks on card payments that are made possible when any security feature is not included in the protocol. We provide a summary discussion of the salient points.

8.4.2.1 Solutions for Card Present

This category corresponds to the left four columns in Table 20, providing payment when the card is physically present. With magnetic stripe cards, data integrity and card authentication features were not placed on the actual card itself. The data stored in a magnetic stripe is static and is kept in plain text which made magnetic stripe cards vulnerable to identity theft attacks [4], cardholder impersonation attacks [41] and card cloning attacks [137].

EMV extended the features of smart cards which provided a secure, “tamper proof”, storage for the card’s private cryptographic keys. The Chip and Pin protocol defined by EMV to secure card

payments makes use of RSA public key infrastructure in three variants. The SDA card has a static signature which is generated by the issuer signed by using the issuer's private key and written to the SDA card during manufacture. However, static signatures are used to approve every transaction, which makes SDA cards vulnerable to cloning attacks [138]. DDA payments on the other hand generate a unique 'challenge-response' RSA signature (SDAD) for each transaction, including a nonce. CDA improves upon DDA by encoding the Application Cryptogram into the signature rather than the transaction data. This makes DDA and especially CDA highly robust against any form of attack.

EMV contactless provides convenience to the customer by authenticating the card instead of actually prompting the cardholder to approve the transactions. Fast DDA (fDDA) and CDA (fCDA) are enhanced versions of DDA and CDA of EMV chip and PIN, excluding the cardholder authentication methods from the protocol. Both DDA and SDA offer protection against known attacks on the payment system, however, each DDA and SDA enabled transactions would require the cardholders to prove their identity, thus adversely affecting usability. This was further addressed with enhanced versions of fDDA and fCDA in EMV contactless [28]. However, fDDA and fCDA still fail to provide payment authentication and non-repudiation of payment transactions.

8.4.2.2 Solutions for Card Not Present

If the card is not present, the situation is very challenging, as we have seen in this chapter. As discussed in the introduction, the complications associated with the implementation of the 3DS 1.0 protocol made it possible for attackers to bypass its security features and perform identity theft attacks [4][117]. Chip Authentication Programme and Transaction Authentication Numbers [36][37] are two token generation technologies that consumers use to produce the answer to a challenge from the authorisation system. Typically, this is done with a little machine that reads a credit card and/or uses a PIN to generate a response to a challenge. These are increasingly commonly provided by banks, but in many cases are limited to payments through banking transactions.

In conclusion, different payment protocols have been developed for different purposes. Satisfactory solutions find a successful combination of usability and security, and also manage the exposure to risk were something to go wrong. For instance, transaction limits on contactless cards as well as the frictionless 3DS 2.0 payment limit both manage the risk by limiting loss exposure of consumers. Not surprisingly, sound approaches challenge for a second factor of authentication information, either through a PIN such as in Chip & PIN as well as Challenged Authentication in 3DS 2.0 or using token generators such as in CAP and TAN. However, these do not always satisfy the usability wishes of merchants, leaving consumer with systems such as 3DS 2.0 that are designed to allow less secure payments and therefore inherently (and by design) expose consumers and card issuers to fraud.

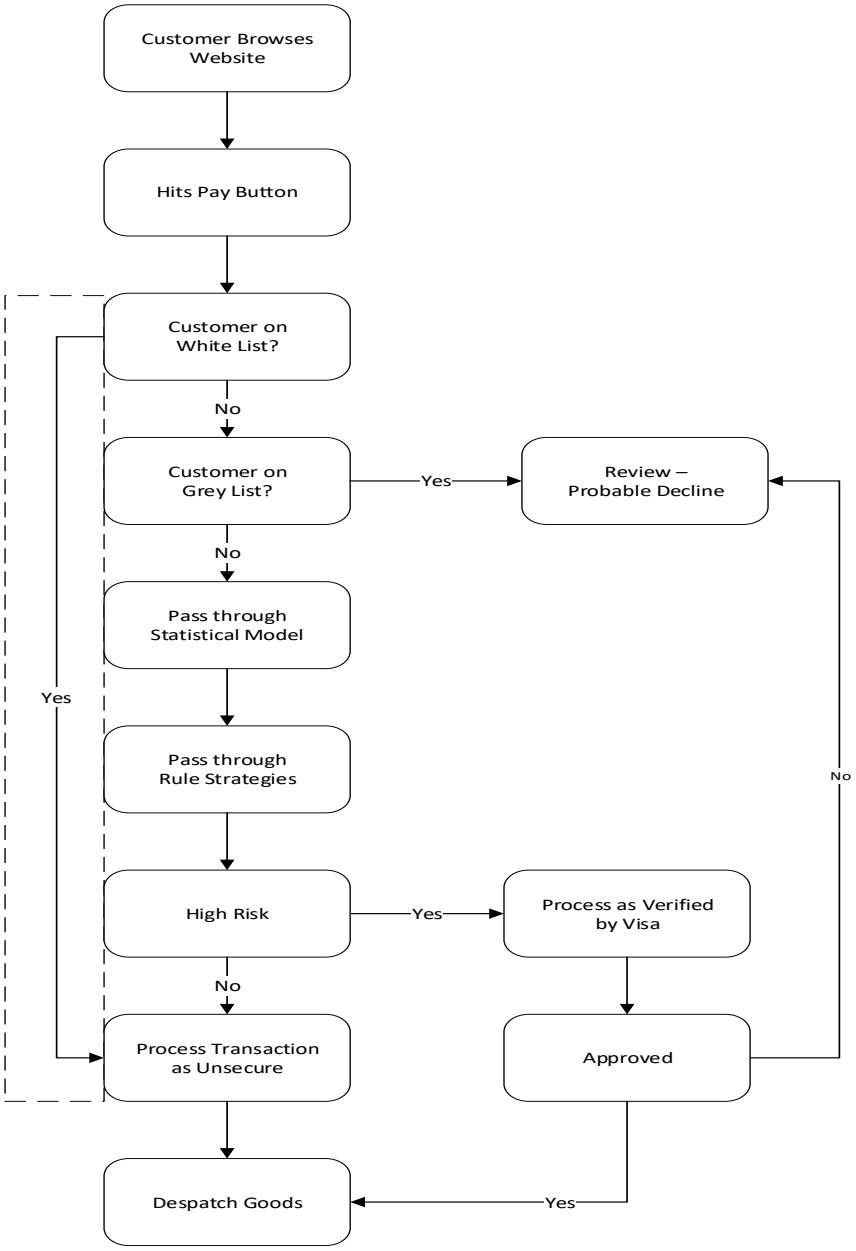

Figure 49 - The topology in which VbV capable merhant using Rules based and TRA/Statistical system

8.5 Betrayal Attack on 3DS 2.0 Transaction Risk Assessment

For customers in the white list (detailed definition below), Visa's 3DS 2.0 system (Verified by Visa (VbV)) does not instantly authorise the card details used while making the purchase. Once in the white list, an attacker can use randomly generated card details to make any number of fraudulent online transactions. The fraudulent transactions will stay undetected until the next payment settlement request which is usually about 24 hours after the transaction [96].

Figure 49. taken from Visa's official guide to merchants [140], shows an approach in which a Verified by Visa (VbV) capable merchant uses the TRA authentication system (TRA is referred to as statistical model by Visa payment network). As discussed before, TRA is a tool used by card issuers to help them evaluate if a transaction is risky. Card issuers evaluate a purchase request for its fraud potential using specific thresholds and risk indicators set by the merchant and/or acquirers. Rule-based authentication, which is introduced by Visa as an additional layer to 3DS 2.0, gives card issuer an ability to rely on the user's TRA information and determine if the merchant wants to authenticate a cardholder. With the Rule-based system, merchants can have their own rules giving them a control on when to deploy 3DS 2.0, which in some cases will minimise authentication to only when required. One such rule is devised to be based on the grey list and the white list of customers.

The white list will include the customers that are valuable to the merchant's business. For white list customers, transactions will pass authentication without any cardholder authentication [140] and the merchant would process orders from white list customers regardless of what products they buy. The white list customers are selected based on TRA data assigned by the issuers to their customers. The grey list of customers includes the converse i.e. known bad IP addresses, email addresses, account numbers, customer names and addresses. Purchase requests from grey list customers fall into higher risk transactions that may fail authentication (transaction declined) or will be challenged using dynamic methods such as One Time Passcodes (OTPs) or secret questions.

Our experiments[3] with VbV unveiled few security concerns over a rule-based system which led us to formulate an attack, we call it as a Betrayal attack. We found that for white list customers the VbV's Statistical and the Rule-based system does not instantly verify the card details entered during the purchase. To fall into the white list category, attacker first needs to make 'x' number of valid transactions (x is decided by the merchants or card issuing banks). Once in the white list, the attacker can make unlimited number of small value transactions using randomly generated card details. The fraudulent transactions will stay undetected until the next payment settlement request (almost about 24 hours or in some cases a week -as it depends on acquiring banks). The attack works because transactions from the white list customers are not passed through any of the statistical checks or fraud detection controls (highlighted with a dashed line in Figure 49). The attack significantly affects the customers that process instant services, for instance, mobile phone top-ups.

8.6 Conclusion

3D Secure 2.0 is a standard that proposes a risk-assessment based approach to securing online payments. Its objective is to move the payment industry from authorisation-only transactions (using

[3] Link to the experimental video: https://goo.gl/UgfQee

credit card details only) to two-factor user authentication. To avoid making the payment laborious for consumers it contains the option to complete a transaction through frictionless authentication. That is, if the authorisation service considers the transaction low risk, it carries out frictionless authentication, which does not challenge the payment initiator for second factor authentication information. 3DS 2.0 is pushed hard by the regulator, especially in Europe through the Payment System Directive (PSD II) [92], and is gaining considerable traction.

This chapter presents the first sizeable experimental study of reallife implementations of 3DS 2.0. Through a reverse engineering study, we map out the transaction sequences for frictionless transactions and we identify two main types of implementations. In most implementations we encountered, the payment initiator's machine is fingerprinted through JavaScripts, except for the implementation based on 3DS 1.0. In our experiments we obtained further insights in the decision making of the authorisation service, experimenting with transaction amounts and the region from which payment was initiated. We found that card issuers differ in terms of their risk appetite, with some issuers considerably more liberal in allowing transaction to proceed unchallenged.

We also demonstrated an impersonation attack against 3DS 2.0, which is more or less a direct implication of the vulnerabilities associated with a risk-based design. This impersonation attack is practically feasible and exploits that fingerprinting information from the payment initiator's machine can be recreated by malware or plug-ins, if installed on that machine. Private/public key and secure storage solutions to such attacks are commonplace, and for instance integrated in mobile payment apps, but these have not found wide-spread acceptance on desktop platforms, leaving payment through browsers vulnerable to attacks.

Giving merchants freedom to choose the security settings may prove useful for merchants' business, but our experimental results and observations with the betrayal attack confirm that it can be detrimental to the overall security of the online payment system.

A key question for the regulator is whether it was justified to allow risk-assessment based approach to online payment security as result of the PSD II negotiations. To answer that question, one needs to consider a variety of factors, including technological feasibility and acceptance, ease-of-use, liability, as well as vulnerabilities and threats. In addition, one would need deeper insight into the specifics of the risk assessment carried out by the card issuer. This and the previous chapter provides the reverse engineering tools to probe the authorisation service and divine which data is used in the decision making and for what values of transaction amount or other parameters it challenges the payment initiator for second factor authorisation information.

Chapter 9. Conclusion

Our research on the security analysis of CNP payment systems started with an aim to answer the following question: *"Does the philosophy of providing excessive convenience to the customer at the checkout have any effect on the security of the CNP payment system?."* The vulnerabilities and the attack scenarios that emerge from our analysis demonstrates that 'lack of security standardisation' and 'freedom of choices given to card issuers and online merchants in ways to accept online CNP payments' are in fact credible threats to the overall security of CNP payment system. The element that makes these vulnerabilities exist in the CNP payment system is the ultimate business need to provide convenience to the customer at the checkout; inadvertently benefitting the cybercriminals.

Specifically, the security of CNP payment system is further worsened by:

- The use of guessable (if not obtainable) static payment card number as a primary element that links the customer to their online payment account.
- Availability of card number as plain text across myriad of platforms for example at magnetic stripe interface, merchant receipts, EMV chip and pin and EMV contactless interface.
- Introduction of the wireless interface on payment cards enabling false readers to communicate with victim's card from a distance to steal payment card details.
- Insecure methods of assigning payment card numbers (sequential card numbers) to their customers by some card issuing banks.
- Lack of real-time sharing of online payment fraud information among payment processors enabling fraudsters to use stolen payment card details across global platforms (discussed in Section 5.3.4).
- Lack of education among online merchants on the risks involved in processing online payments with variable card data fields as reflected in distributed guessing attack (Chapter 6). Furthermore, lack of education among payment acquirers and in-store merchants on security risks of losing merchant receipts.

- Insecure design of authentication and authorisation response codes from the card issuing banks revealing information about user payment card details to the online merchant.
- The incapabilities and limitations of payment processing systems in not detecting simple distributed brute force attacks.
- Use of static data for customer authentication for 3DS 2.0 frictionless authentication, stored as plain-text on the customer machine.

All these weakness and vulnerabilities when combined introduce several new categories of exploitable attacks that were not present previously in the CNP payment system. The attack scenario that we have practically demonstrated by exploiting the above vulnerabilities include but are not limited to: distributed guessing attack on authorisation-only CNP payment system, cardholder impersonation attack and betrayal attack on 3DS 2.0 frictionless authentication.

This PhD thesis "*Does the Card Payment System Landscape Unwittingly Facilitate Fraud?"* presents our security analysis of the CNP payment system. The vulnerabilities and attack scenarios that we have identified in this research are made possible because of the systemic weaknesses of the card payment system. These vulnerabilities have even proven to aid fraudsters to contribute to growing online payment fraud and has also curtailed the loss of customer confidence in the technology where even Sir Tim Berners Lee is no exception [5].

For many reasons, research into the security of card-not-present is projected as a business activity, and payment system stakeholders (card issuing banks and online merchants) are too sensitive to talk about their fraud detection systems. Even as pointed by several researchers there is a need for academic scrutiny into the security of CNP payment systems [4]. Therefore, the research work in this PhD sets out to bridge the gap in between the academia and the business worlds, provides an approach to an investigation into the security of CNP payment systems and fill in the needs for literature into the security of the CNP payment systems. At the core of this PhD research is the systemic methodology that we adapted for responsible research into practically sensitive payment systems. The approval of our findings by the academia and even by the payment processing entities, to a substantial extent has demonstrated the practical impact of our research work.

9.1 Summary of Contributions

Contributions from my PhD research are:

- The contribution of the literature review: a detailed literature review that binds the research on the security analysis of various card payment systems and protocols.
- The contribution of Methodology: A framework for current and future research looking into the security analysis of CNP payment systems

- The contribution of software tools: Creation of multiple software tools for the protocol security analysis
- The contribution of identified vulnerabilities: The identification of several weak practices in CNP payment system, undocumented vulnerabilities and practical demonstration of at least three attack scenarios: *Distributed guessing attack, cardholder impersonation attack and betrayal attack*
- The contribution of Distributed Guessing Attack
- The contribution of Reverse engineering the 3DS 2.0 frictionless authentication
- The contribution of Cardholder impersonation attack on 3DS 2.0 frictionless authentication
- The contribution of betrayal attack
- Practical experimental research with impact

9.1.1 The Contribution of the Literature Review

In the literature review, we have established a link between the existing academic research into payments security and the areas of weaknesses in CNP payment systems, which were of potential interest of this research. With literature review on payments security, we established certain weakness in the architecture of card payment system. Specifically,

- The wireless interface on payment cards introduces new categories of attack (i.e. skimming, eavesdropping and relay) and the data were stolen from the wireless interface can be used to make fraudulent CNP payments.
- Just like contactless payment cards, the data in the EMV chip and PIN interface can easily be read by false readers. This data includes the complete card number and card's expiry date which can be used to make fraudulent CNP payments.
- The 3DS 1.0 protocol which required the cardholder to enter static passwords on a pop-up screen was more burden to the payment industry than a solution. This allowed freedom for the online merchant to have options on the protocol they want to implement on their checkout systems.
- Cryptographically bounded one-time passcodes for online payments using EMV readers were vulnerable to chip and PIN attacks where an attacker can generate one-time passcodes from stolen payment cards.

The literature review supports the assertion made in this PhD thesis that the security of the online payment system is fundamentally weakened by the philosophy of providing convenience to the customer. Also, the requirement for backward compatibility makes it essential for the card data to be available in plain text across other interfaces which might have an impact on the security of the CNP payment system.

9.1.2 The Contribution of the Methodology

The analysis methodology used for this PhD research consist of four systematic steps (i) the use of UML sequence diagrams and comparison table which enabled us to concisely provide the description of CNP payment system (ii) involved the use of various vulnerability identification techniques like security failure analysis and reverse engineering techniques which allowed us to define the test cases and identify the vulnerabilities of CNP payment system. (iii) involved demonstrating the existence of attacks in the real world and determining the feasibility, magnitude and representativeness of the attack and (iv) examining how disclosure exercise can be done ethically and effectively to mitigate the vulnerabilities.

9.1.3 The Contribution of Software Tools

To assess the CNP payment system for vulnerabilities, we have designed several software tools which involve (i) tshirtshop - a web store enabled us to link our understandings on CNP payment protocols with practical implementations. It provided us with an experimental platform which was used to assess the security of CNP payment protocols and fraud filters offered by payment acquirers (ii) website bot – enabled us to execute our test case on tens of online merchants. Website bot was programmed to link to online websites under study with an aim to explore vulnerabilities in authorisation-only CNP payment system. (iii) AutoIt scripts – which automated our experiments on tens of windows applications and (iv) Android NFC skimming app – which was developed to understand the EMV contactless protocol. This enabled us to explore the vulnerabilities in contactless payment protocols that can be linked to the CNP payment system.

9.1.4 The Contribution of Distributed Guessing Attack

The distributed guessing attack demonstrated that the current security model of the payment networks is ill-suited to dealing with online CNP payments. The networks lack the ability to correlate information from multiple sources to detect distributed guessing attacks. Our survey of the online payment landscape highlights that the vulnerabilities we described in Chapter 7 are systemic.

The variations in merchants' security settings, which create the conditions for the attack, are present in the 400 of the most popular e-commerce websites. We present an attack scenario involving payment systems such as iTunes, Google Wallet, and PayPal that allows attackers to subvert the payment functionality from its intended purpose of validating entered credit or debit card details, into helping the attackers to generate all of the security data fields required to create an online payment account. Even worse, these data will allow attackers to transfer money to an anonymous recipient, on top of an ability to fraudulently purchase items online.

Our experimental work has shown that it is possible to implement a web bot which will generate all of the fields required to create an online account. We have proved that it is possible to circumvent all of the security features (including separation of printed and electronic data and rule for storage of data

by the merchants) put in place to protect the cardholders. We have also demonstrated that it is possible to refine the web bot so that it will generate data from multiple websites, circumventing the limitations of the number of attempts one can make to enter the correct value, which is imposed by some payment systems. As a result of our ethical disclosure process, a number of the top 10 Alexa rated online merchants have changed their online security settings. This shows that the research is relevant and impactful.

9.1.5 The Contribution of Reverse Engineering the 3DS 2.0 Frictionless Authentication

By reverse engineering the frictionless instance of 3DS 2.0 protocol, we provided the first public description of the working of the protocol. The research work done in this section was a substantial contribution to our PhD as it helped us to identify the vulnerabilities associated with the 3DS 2.0 protocol. We demonstrated the use of publicly available tools like Fiddler that can be employed for performing security research into online CNP payment systems.

9.1.6 The Contribution of Cardholder Impersonation Attack

The work done to explore cardholder impersonation attack was a significant advancement in our understanding of attacks against 3DS 2.0 because it is applicable to the payment protocols recommend currently by the payment industry. it may be one of the most realistic and attractive attacks for criminals, if and when authorisation only transactions are no longer permitted. It could even be used at the moment, by criminals who wish to make purchases on the merchant and in the regions like Europe which now mandate 3DS 2.0 on merchant websites. If this attack becomes more widely used, its net effect will be that criminals can use stolen cards in online shops without the cardholder being negligent exactly as was the case with authorisation only systems before the introduction of 3D Secure.

9.1.7 The Contribution of Betrayal Attack

Giving merchants freedom to choose the security settings may prove useful for merchants' business, but our experimental results and observations with the betrayal attack confirm that it can be detrimental to the overall security of the online payment system. We notified to the payment industry stakeholders about the existence of betrayal attack on 3DS 2.0 system as a result of which we saw that the 3DS 2.0 merchant guidelines were change reflecting the patching of betrayal attack. By this, we conclude that our vulnerability disclosure with betrayal attack contributed to patching of vulnerabilities that existed within the 3DS 2.0protocol.

9.1.8 Practical Experimental Research

The research work in this PhD was focussed on protocols that are currently in use by the payment industry to accept CNP payment over the Internet. Unlike EMV protocol, the security features of the CNP payment protocols are not documented in the available implementation guides and standards by the payment processor. As we have demonstrated with betrayal attack that, theoretically, there may be

several features in the design of CNP payment protocols but in practice such features, if implemented incorrectly can result in exploitable vulnerabilities.

Even though there were limitations identified with the access of CNP payment system documentation and test cards required for experiments, we still took this challenge and designed our experiments on real implementations of CNP payment protocols and by remaining ethically moral with the sensitive CNP payment processing systems. By documenting our results to public accessible platforms, we passed our message of security in CNP payment systems and educated public some best practices to be followed while making payments over CNP payment systems.

9.2 Future Work

The introduction of IoT devices making purchases over the Internet [141] has brought new security challenges to the CNP payment system [142]. My future research on the security of CNP payment systems will continue to adopt the best practices learnt from this research and apply it to perform security assessments of IoT devices making CNP payments.

The W3C consortium of web standardisation [143] is also working towards enhancing and standardising the payment request API's. The payment request API's are the modern methods supported by the browser to eliminate the cumbersome fields required to be filled in by the customers while making online purchases. The idea behind payment request API is to support one-click payments in which browser takes control of the merchant checkout page and fill in the required data for making a payment. With my future work, I aim to assess the security of these payment request API's and potential CNP payment protocols that may appear in the future.

BIBLIOGRAPHY

[1] Nilson, “Card Fraud Losses Reach $21.84 Billion,” 2016. [Online]. Available: https://nilsonreport.com/upload/content_promo/The_Nilson_Report_10-17-2016.pdf [Accessed: 30-Apr-2018]

[2] Financial Fraud Action UK, “The Definitive Overview of Payment Industry Fraud,” 2017. [Online]. Available: https://www.financialfraudaction.org.uk/fraudfacts17/assets/fraud_the_facts.pdf. [Accessed: 30-Apr-2018]

[3] Drimer S., Murdoch S.J., Anderson R. (2009) Optimised to Fail: Card Readers for Online Banking. In: Dingledine R., Golle P. (eds) Financial Cryptography and Data Security. FC 2009. Lecture Notes in Computer Science, vol 5628. Springer, Berlin, Heidelberg

[4] Murdoch S.J., Anderson R. (2010) Verified by Visa and MasterCard SecureCode: Or, How Not to Design Authentication. In: Sion R. (eds) Financial Cryptography and Data Security. FC 2010. Lecture Notes in Computer Science, vol 6052. Springer, Berlin, Heidelberg

[5] TheNewYorkTimes, “What if ‘One Click’ Buying Were Internetwide? - The New York Times.” [Online]. Available: https://www.nytimes.com/2016/09/26/business/dealbook/what-if-one-click-buying-were-internetwide.html. [Accessed: 30-Apr-2018].

[6] M. A. Ali, B. Arief, M. Emms, and A. van Moorsel, “Does the Online Card Payment Landscape Unwittingly Facilitate Fraud?,” *IEEE Secur. Priv.*, vol. 15, no. 2, pp. 78–86, Mar. 2017.

[7] M. Mehrnezhad, M. A. Ali, F. Hao, and A. van Moorsel, “NFC Payment Spy: A Privacy Attack on Contactless Payments,” Springer, Cham, 2016, pp. 92–111.

[8] BBC, “‘Frighteningly easy’ for criminals to get Visa card details, study claims - BBC News.” [Online]. Available: http://www.bbc.co.uk/news/uk-england-tyne-38181149. [Accessed: 30-Apr-2018].

[9] The Telegraph, “Hacked in just six seconds: How criminals only need moments to guess card number and security code.” [Online]. Available: https://www.telegraph.co.uk/news/2016/12/02/hacked-just-six-seconds-criminals-need-moments-guess-card-number/. [Accessed: 30-Apr-2018].

[10] msn, “It takes only six seconds to hack a credit card.” [Online]. Available: https://www.msn.com/en-in/money/topstories/it-takes-only-six-seconds-to-hack-a-credit-card/ar-AAl2QeY. [Accessed: 30-Apr-2018].

[11] CTV, “How hackers can guess your credit card information in just 6 seconds | CTV News.”

[Online]. Available: https://www.ctvnews.ca/sci-tech/how-hackers-can-guess-your-credit-card-information-in-just-6-seconds-1.3185772. [Accessed: 30-Apr-2018].

[12] The Sun, “Internet crooks can now hack your credit card details in just SECONDS.” [Online]. Available: https://www.thesun.co.uk/news/2310669/internet-crooks-can-now-hack-your-credit-card-details-in-just-seconds/. [Accessed: 30-Apr-2018].

[13] T. Times, “Fraudsters take six seconds to steal bank card details | News | The Times.” [Online]. Available: https://www.thetimes.co.uk/article/fraudsters-take-six-seconds-to-steal-bank-card-details-670wtrp67. [Accessed: 30-Apr-2018].

[14] FirstData, “‘Flaw’ in Visa contactless cards could help criminals steal thousands - AOL.” [Online]. Available: https://www.aol.co.uk/2014/11/04/flaw-in-visa-contactless-cards-could-help-criminals-steal-thou/. [Accessed: 30-Apr-2018].

[15] The Guardian, “Tesco Bank cyber-thieves stole £2.5m from 9,000 people | Business | The Guardian.” [Online]. Available: https://www.theguardian.com/business/2016/nov/08/tesco-bank-cyber-thieves-25m. [Accessed: 30-Apr-2018].

[16] I. News, “Hacking takes seconds, say experts at Newcastle University | Tyne Tees - ITV News.” [Online]. Available: http://www.itv.com/news/tyne-tees/2016-12-02/hacking-takes-seconds-say-experts-at-newcastle-university/. [Accessed: 30-Apr-2018].

[17] B. Schneier, “Guessing Credit Card Security Details - Schneier on Security.” [Online]. Available: https://www.schneier.com/blog/archives/2016/12/guessing_credit.html. [Accessed: 30-Apr-2018].

[18] A. ThreatTraq, “ThreatTraq #225 - Distributed Guessing Attack | AT&T ThreatTraq (Security for Security Pros) on acast.” [Online]. Available: https://www.acast.com/attthreattraq/threattraq-225---distributed-guessing-attack. [Accessed: 30-Apr-2018].

[19] SC Magazine, “Tesco Bank allegedly ignored warnings of hack from Visa.” [Online]. Available: https://www.scmagazineuk.com/tesco-bank-allegedly-ignored-warnings-of-hack-from-visa/article/575762/. [Accessed: 30-Apr-2018].

[20] The Times, “Tesco Bank ‘failed to heed warning on cyberattack’ | Business | The Times.” [Online]. Available: https://www.thetimes.co.uk/article/tesco-bank-failed-to-heed-warning-on-cyberattack-rpgvhrh8j. [Accessed: 30-Apr-2018].

[21] PCI-DSS, “Payment Card Industry Data Security Standard,” *Security*, no. April, p. 139, 2015. [Online]. Available: https://www.pcisecuritystandards.org/document_library?category=pcidss&document=pci_dss .

[Accessed: 30-Apr-2018]

[22] The European Union, “General Data Protection Regulation (GDPR) – Final text neatly arranged,” 2016. [Online]. Available: https://gdpr-info.eu/. [Accessed: 30-Apr-2018].

[23] MasterCard, “History of the Card Payments System.” [Online]. Available: http://www.mastercard.com/us/company/en/docs/history of payments.pdf. [Accessed: 30-Apr-2018]

[24] TheUKCardsAssociation, “Discover The History of Payment Cards| The UK Cards Association.” [Online]. Available: http://www.theukcardsassociation.org.uk/history_of_cards/index.asp. [Accessed: 28-Apr-2018].

[25] EMVCo, “Home - EMVCo.” [Online]. Available: https://www.emvco.com/. [Accessed: 28-Apr-2018].

[26] EMVCo, “EMV Integrated Circuit Card Specifications for Payment Systems,” *B. 1-4*, vol. 4.3, 2011. [Online]. Available: https://goo.gl/xfwJn1

[27] New York Times, “Attention Shoppers: Internet Is Open - The New York Times.” [Online]. Available: https://www.nytimes.com/1994/08/12/business/attention-shoppers-internet-is-open.html. [Accessed: 28-Apr-2018].

[28] M. Emms, B. Arief, L. Freitas, J. Hannon, and A. van Moorsel, “Harvesting High Value Foreign Currency Transactions from EMV Contactless Credit Cards Without the PIN,” in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security - CCS ’14*, 2014, pp. 716–726.

[29] Martin Emms, “Contactless payments :usability at the cost of security?,” Newcastle Univeristy. [Online]. Avaialbe: https://theses.ncl.ac.uk/dspace/bitstream/10443/3304/1/Emms,%20M.%202016.pdf. [Accessed: 30-Aug-18]

[30] IBM, “VISA PIN Algorithms.” [Online]. Available: https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.1.0/com.ibm.zos.v2r1.csfb400/csfb4za2598.htm. [Accessed: 28-Apr-2018].

[31] I. 7813, “ISO/IEC 7813:2006 - Information technology -- Identification cards -- Financial transaction cards.” [Online]. Available: https://www.iso.org/standard/43317.html. [Accessed: 28-Apr-2018].

[32] K. Mayes and K. Markantonakis, Eds., *Smart Cards, Tokens, Security and Applications*. Cham: Springer International Publishing, 2017.

[33] ISO, “ISO/IEC 7816-4:2013 - Identification cards -- Integrated circuit cards -- Part 4: Organization, security and commands for interchange.” [Online]. Available: https://www.iso.org/standard/54550.html. [Accessed: 30-Apr-2018].

[34] K. Markantonakis and D. Main, “Smart Cards for Banking and Finance,” in *Smart Cards, Tokens, Security and Applications*, Cham: Springer International Publishing, 2017, pp. 129–153.

[35] C. Boyd and A. Mathuria, *Protocols for Authentication and Key Establishment*. Springer Berlin Heidelberg, 2003.

[36] Amazon.com INC, “US5960411A - Method and system for placing a purchase order via a communications network - Google Patents,” *US Grant*, 2006. [Online]. Available: https://patents.google.com/patent/US5960411A/en. [Accessed: 30-Aug-2018].

[37] “The Nilson Report,” 2017. [Online]. Available: https://nilsonreport.com/upload/content_promo/The_Nilson_Report_Issue_1118.pdf. [Accessed: 30-Apr-2018]

[38] “Statista - U.S. payment card fraud losses by type 2018.” [Online]. Available: https://www.statista.com/statistics/419628/payment-card-fraud-losses-usa-by-type/. [Accessed: 30-Apr-2018]

[39] FICO, “Evolution of Card Fraud In Europe 2016,” 2018. [Online]. Available: http://www.fico.com/europeanfraud/. [Accessed: 30-Apr-2018]

[40] Europol, “Payment Card Fraud In the European Union - Perspective of Law Enforcement Agencies,” 2012. [Online]. Available: http://data.consilium.europa.eu/doc/document/ST-12811-2017-INIT/en/pdf. [Accessed: 30-Apr-2018]

[41] S. J. Murdoch and R. Anderson, “Security Protocols and Evidence: Where Many Payment Systems Fail,” Springer, Berlin, Heidelberg, 2014, pp. 21–32.

[42] Darwin Charles, “On The Origin of Species by Means of Natural Selection,” *D. Appleton and Company*, 1946. [Online]. Available: http://darwin-online.org.uk/converted/pdf/1861_OriginNY_F382.pdf. [Accessed: 31-Aug-2018].

[43] S. J. Murdoch, S. Drimer, R. Anderson, and M. Bond, “Chip and PIN is Broken,” in *2010 IEEE Symposium on Security and Privacy*, 2010, pp. 433–446.

[44] M. Roland and J. Langer, “Cloning credit cards: a combined pre-play and downgrade attack on EMV contactless,” *Proceedings of the 7th USENIX conference on Offensive Technologies*. USENIX Association, pp. 6–6, 2013.

[45] M. Bond, O. Choudary, S. J. Murdoch, S. Skorobogatov, and R. Anderson, “Chip and Skim:

Cloning EMV Cards with the Pre-play Attack," in *2014 IEEE Symposium on Security and Privacy*, 2014, pp. 49–64.

[46] A. Barisani, D. Bianco, A. Laurie, and Z. Franken, "Chip & PIN is definitely broken Credit Card skimming and PIN harvesting in an EMV world," 2011.

[47] J. P. Degabriele, A. Lehmann, K. G. Paterson, N. P. Smart, and M. Strefler, "On the Joint Security of Encryption and Signature in EMV," Springer, Berlin, Heidelberg, 2012, pp. 116–135.

[48] A. Laurie, "Chip and spin," *Infosecurity*, vol. 4, no. 8, pp. 38–40, Nov. 2007.

[49] M. Emms, B. Arief, N. Little, and A. van Moorsel, "Risks of Offline Verify PIN on Contactless Cards," Springer, Berlin, Heidelberg, 2013, pp. 313–321.

[50] L. Fancis, Hancky P. Gerard, K. Mayes, and K. Markantonakis, "Practical Relay Attack on Contactless Transactions by Using NFC Mobile Phones. | Request PDF," *Cryptol. Inf. Secur. Ser.*, 2011.

[51] M. Roland, J. Langer, and J. Scharinger, "Applying relay attacks to Google Wallet," in *2013 5th International Workshop on Near Field Communication (NFC)*, 2013, pp. 1–6.

[52] Z. Kfir and A. Wool, "Picking Virtual Pockets using Relay Attacks on Contactless Smartcard," in *First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05)*, pp. 47–58.

[53] M. Roland, J. Langer, and J. Scharinger, "Practical Attack Scenarios on Secure Element-Enabled Mobile Devices," in *2012 4th International Workshop on Near Field Communication*, 2012, pp. 19–24.

[54] T. P. Diakos, J. A. Briffa, T. W. C. Brown, and S. Wesemeyer, "Eavesdropping near-field contactless payments: a quantitative analysis," *J. Eng.*, vol. 2013, no. 10, pp. 48–54, Oct. 2013.

[55] G. P. Hancke, G. P. Hancke, and S. C. Centre, "Practical Eavesdropping and Skimming Attacks on High-Frequency RFID Tokens."

[56] LeParisien, "L'imparable escroquerie à la carte bancaire - Le Parisien," 2012. [Online]. Available: http://www.leparisien.fr/faits-divers/l-imparable-escroquerie-a-la-carte-bancaire-24-01-2012-1826971.php. [Accessed: 13-Jul-2018].

[57] Z. Ahmad, L. Francis, T. Ahmed, C. Lobodzinski, D. Audsin, and P. Jiang, "Enhancing the Security of Mobile Applications by Using TEE and (U)SIM," in *2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing*, 2013, pp. 575–582.

[58] T. Chothia, F. D. Garcia, J. de Ruiter, J. van den Breekel, and M. Thompson, "Relay Cost

Bounding for Contactless EMV Payments," Springer, Berlin, Heidelberg, 2015, pp. 189–206.

[59] "ISO/IEC 14443-3:2011 - Identification cards -- Contactless integrated circuit cards -- Proximity cards -- Part 3: Initialization and anticollision." [Online]. Available: https://www.iso.org/standard/50942.html. [Accessed: 14-Jul-2018].

[60] I. Kirschenbaum, "How to Build a Low-Cost, Extended-Range RFID Skimmer." [Online]. Available: https://www.usenix.org/legacy/events/sec06/tech/full_papers/kirschenbaum/kirschenbaum.pdf. [Accessed: 30-Apr-2018]

[61] Y. Oren, D. Schirman, and A. Wool, "Range Extension Attacks on Contactless Smart Cards." [Online]. Available: https://pdfs.semanticscholar.org/e7ab/0245bc9d824d37f2538cb9d29fab122d58be.pdf. [Accessed: 30-Apr-2018]

[62] R. P. GmbH, "New banking security system iTAN not as secure as claimed," 2009. [Online]. Available: https://www.redteam-pentesting.de/en/advisories/rt-sa-2005-014/-new-banking-security-system-itan-not-as-secure-as-claimed. [Accessed: 30-Apr-2018]

[63] R. P. GmbH, "Man-in-the-Middle Attacks against the chipTAN comfort Online Banking System," 2009. [Online]. Available: https://www.redteam-pentesting.de/publications/2009-11-23-MitM-chipTAN-comfort_RedTeam-Pentesting_EN.pdf. [Accessed: 30-Apr-2018]

[64] M. A. Ali, "University Smartcard Protocol Analysis MSc Thesis," Newcastle University, 2013.

[65] Visa, "Visa Developer Centre," 2018. [Online]. Available: https://developer.visa.com/pages/visa-3d-secure. [Accessed: 08-May-2018].

[66] Visa, "Card Acceptance Guidelines for Visa Merchants," 2015. [Online]. Available: https://usa.visa.com/dam/VCOM/download/merchants/card-acceptance-guidelines-for-merchants.pdf. [Accessed: 30-Apr-2018]

[67] MasterCard, "U.S. Merchant Acceptance Guide," 2010. [Online]. Available: http://cdn2.hubspot.net/hubfs/400547/MerchantAcceptanceGuide_Manual-MasterCard.pdf. [Accessed: 30-Apr-2018]

[68] AmericanExpress, "3D Secure Online Payments | Safekey." [Online]. Available: https://www.americanexpress.com/uk/content/safekey-information.html. [Accessed: 01-May-2018].

[69] Visa, "Verified by Visa (VbV) - Merchant Implementation Guide," 2005. [Online]. Available: https://www.visa.gr/media/images/verifiedbyvisa_3dsecure-42-10230.pdf. [Accessed: 30-Apr-

2018]

[70] MasterCard, “3D Secure Integration Guides.” [Online]. Available: https://www.mastercard.com/gateway/integration-guides.html. [Accessed: 08-May-2018].

[71] Australian Competition and Consumer Commision, “ACCC proposes to deny authorisation to APCA for 3D Secure arrangements | ACCC,” 2016. [Online]. Available: https://www.accc.gov.au/media-release/accc-proposes-to-deny-authorisation-to-apca-for-3d-secure-arrangements. [Accessed: 30-Aug-2018].

[72] T. Abbes, A. Bouhoula, and M. Rusinowitch, “Protocol analysis in intrusion detection using decision tree,” in *International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004.*, 2004, p. 404–408 Vol.1.

[73] J. Jurjens, “Sound methods and effective tools for model-based security engineering with UML,” in *Proceedings. 27th International Conference on Software Engineering, 2005. ICSE 2005.*, pp. 322–331.

[74] T. Murata, “Petri nets: Properties, analysis and applications,” *Proc. IEEE*, vol. 77, no. 4, pp. 541–580, Apr. 1989.

[75] EMVCo, *EMV 3-D Secure Protocol and Core Functions Specification*, Version 2. 2017. [Online]. Available: https://www.emvco.com/emv-technologies/3d-secure/. [Accessed: 30-Apr-2018]

[76] A. D. John A. Wise, *Information Systems: Failure Analysis*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1987.

[77] Z. Durumeric *et al.*, “The Matter of Heartbleed,” in *Proceedings of the 2014 Conference on Internet Measurement Conference - IMC ’14*, 2014, pp. 475–488.

[78] H. Wang *et al.*, “Vulnerability Assessment of OAuth Implementations in Android Applications,” in *Proceedings of the 31st Annual Computer Security Applications Conference on - ACSAC 2015*, 2015, pp. 61–70.

[79] R. Wang, S. Chen, X. Wang, and S. Qadeer, “How to Shop for Free Online -- Security Analysis of Cashier-as-a-Service Based Web Stores,” in *2011 IEEE Symposium on Security and Privacy*, 2011, pp. 465–480.

[80] J. Vanegue and S. K. Lahiri, “Towards Practical Reactive Security Audit Using Extended Static Checkers,” in *2013 IEEE Symposium on Security and Privacy*, 2013, pp. 33–47.

[81] R. J. Anderson and R. J., “Why cryptosystems fail,” *Commun. ACM*, vol. 37, no. 11, pp. 32–40, Nov. 1994.

[82] M. Vasek and T. Moore, “Do malware reports expedite cleanup? an experimental study,”

Proceedings of the 5th USENIX conference on Cyber Security Experimentation and Test. USENIX Association, pp. 6–6, 2012.

[83] M. Zhao, J. Grossklags, and P. Liu, "An Empirical Study of Web Vulnerability Discovery Ecosystems," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security - CCS '15*, 2015, pp. 1105–1117.

[84] T. Moore and R. Clayton, "Ethical Dilemmas in Take-Down Research," in *Proceedings of the 2011 international conference on Financial Cryptography and Data Security*, Springer-Verlag, 2012, pp. 154–168.

[85] A. Nappa, R. Johnson, L. Bilge, J. Caballero, and T. Dumitras, "The Attack of the Clones: A Study of the Impact of Shared Code on Vulnerability Patching," in *2015 IEEE Symposium on Security and Privacy*, 2015, pp. 692–708.

[86] F. Li *et al.*, *You've got Vulnerability: Exploring Effective Vulnerability Notifications*. USENIX Association, 2005.

[87] Hossein Bidgoli, *E-Commerce Vulnerabilities - Handbook of Information Security: Threats, Vulnerabilities, Prevention, Detection, and Management, Volume 3 [Book]*. .

[88] International Organisation for Standardisation, "ISO 8583:2003(en), Financial transaction card originated messages — Interchange message specifications," *ISO*, 2003. [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso:8583:-1:ed-1:v1:en. [Accessed: 30-Aug-2018].

[89] The UK Cards Association, "Standard 70 - Payment Card Standards - UK Cards Association," 2013. [Online]. Available: http://www.theukcardsassociation.org.uk/technical_services_standards/card_standards.asp. [Accessed: 30-Aug-2018].

[90] ISO20022, "Universal Financial Industry Message Scheme," 2014. [Online]. Available: https://www.iso20022.org/. [Accessed: 30-Aug-2018].

[91] E. Council, "First Data E-commerce Payments Gateway." [Online]. Available: https://www.firstdata.com/downloads/international/PSP_e-commerce_solution.pdf. [Accessed: 30-Apr-2018]

[92] P. S. D.- II, "Directive (EU) 2015/2366 of the European Parliament." [Online]. Available: http://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32015L2366&from=EN. [Accessed: 30-Apr-2018].

[93] MasterCard, "Advantages of a risk based authentication strategy for mastercard securecode." [Online]. Available: https://globalrisk.mastercard.com/wp-content/uploads/2015/12/Advantages-of-Risk-Based-Authentication.pdf. [Accessed: 30-Apr-

2018]

[94] Visa, “Frictionless Experience with Verified by Visa Risk-based authentication case study.” [Online]. Available: https://www.emc.com/collateral/customer-profiles/44933-visa-vbv-case-study.pdf. [Accessed: 30-Apr-2018]

[95] Visa, “Visa Checkout.” [Online]. Available: https://www.visa.co.uk/visa-checkout/. [Accessed: 01-May-2018].

[96] PayPal, “Payflow Link User’s Guide Payflow Link User’s Guide Payflow Link User’s Guide.” [Online]. Available: https://www.emc.com/collateral/customer-profiles/44933-visa-vbv-case-study.pdf. [Accessed: 30-08-2018]

[97] Temp-mail, “Temp Mail - Disposable Temporary Email.” [Online]. Available: https://temp-mail.org/. [Accessed: 04-Jul-2018].

[98] A. K. Ghosh, *E-commerce security : weak links, best defenses*. John Wiley, 1998.

[99] A. Flesner, *AutoIt v3 : your quick guide*. O’Reilly, 2007.

[100] “ISO/IEC 7812-1:2017 - Identification cards -- Identification of issuers -- Part 1: Numbering system.” [Online]. Available: https://www.iso.org/standard/70484.html. [Accessed: 15-Jul-2018].

[101] “ISO 10202-6:1994 - Financial transaction cards -- Security architecture of financial transaction systems using integrated circuit cards -- Part 6: Cardholder verification.” [Online]. Available: https://www.iso.org/standard/18233.html. [Accessed: 15-Jul-2018].

[102] “Hans Peter Luhn and the Birth of the Hashing Algorithm - IEEE Spectrum.” [Online]. Available: https://spectrum.ieee.org/tech-history/silicon-revolution/hans-peter-luhn-and-the-birth-of-the-hashing-algorithm. [Accessed: 15-Jul-2018].

[103] Bindb, “Bank Identification Numbers Database, BIN Lookup by BinDB.” [Online]. Available: https://www.bindb.com/. [Accessed: 01-May-2018].

[104] Exactbins, “Credit Card BIN Numbers Database | Bank Identification Number Lookup | BIN List | ExactBin.” [Online]. Available: https://www.exactbins.com/bin-lookup. [Accessed: 01-May-2018].

[105] Alexa, “Alexa - Top Sites by Category: Business/E-Commerce.” [Online]. Available: https://www.alexa.com/topsites/category/Top/Business/E-Commerce. [Accessed: 01-May-2018].

[106] Braintreepayments, “Fraud Tools | 3D Secure - Braintree Support Articles.” [Online]. Available: https://articles.braintreepayments.com/guides/fraud-tools/3d-secure#processing. [Accessed: 30-Apr-2018].

[107] Visa, “Verified by Visa Acquirer and Merchant Implementation Guide,” 2011. [Online]. Available: https://www.visa.gr/media/images/verifiedbyvisa_3dsecure-42-10230.pdf. [Accessed: 30-Apr-2018]

[108] Mastercard, “Information about this New Manual SecureCode Merchant Implementation Guide,” 2005. [Online]. Avaialable: https://www.mastercard.com/uk/wce/PDF/smi-manual.pdf. [Accessed: 30-Apr-2018]

[109] Google, “Google Pay.” [Online]. Available: https://pay.google.com/send/home?authuser=0. [Accessed: 01-May-2018].

[110] Mastercard, “Masterpass - Digital Wallet by Mastercard.” [Online]. Available: https://masterpass.com/en-us.html. [Accessed: 01-May-2018].

[111] Amazon, “Amazon Pay.” [Online]. Available: https://pay.amazon.com/uk. [Accessed: 01-May-2018].

[112] E. Bursztein, M. Martin, and J. Mitchell, “Text-based CAPTCHA strengths and weaknesses,” in *Proceedings of the 18th ACM conference on Computer and communications security - CCS ’11*, 2011, p. 125.

[113] A. S. El Ahmad, J. Yan, and W.-Y. Ng, “CAPTCHA Design: Color, Usability, and Security,” *IEEE Internet Comput.*, vol. 16, no. 2, pp. 44–51, Mar. 2012.

[114] MasterCard, “The MasterCard Network Advantage Multinational Issuers.”

[115] Telerik, “Fiddler - Free Web Debugging Proxy - Telerik.” [Online]. Available: https://www.telerik.com/fiddler. [Accessed: 29-Apr-2018].

[116] Alexa, “Alexa - Top Sites by Category: Business/E-Commerce.” [Online]. Available: https://www.alexa.com/topsites/category/Top/Business/E-Commerce. [Accessed: 08-May-2018].

[117] Mozilla, “DNT - HTTP | MDN.” [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/DNT. [Accessed: 29-Apr-2018].

[118] Microsoft, “The Browser Object Model.” [Online]. Available: https://msdn.microsoft.com/en-gb/library/ms952643.aspx. [Accessed: 29-Apr-2018].

[119] Intelligent Systems Lab, “JS NICE: Statistical renaming, Type inference and Deobfuscation,” 2018. [Online]. Available: http://jsnice.org/. [Accessed: 09-May-2018].

[120] OWASP, “Cross-site Scripting (XSS) - OWASP,” 2018. [Online]. Available: https://www.owasp.org/index.php/Cross-site_Scripting_(XSS). [Accessed: 31-Aug-2018].

[121] A. Barth, J. Caballero, and D. Song, “Secure Content Sniffing for Web Browsers, or How to

Stop Papers from Reviewing Themselves," in *2009 30th IEEE Symposium on Security and Privacy*, 2009, pp. 360–371.

[122] M. Ter Louw and V. N. Venkatakrishnan, "Blueprint: Robust Prevention of Cross-site Scripting Attacks for Existing Browsers," in *2009 30th IEEE Symposium on Security and Privacy*, 2009, pp. 331–346.

[123] MalShare, "Malware Repository for Researchers." [Online]. Available: https://malshare.com/. [Accessed: 08-May-2018].

[124] N. Etaher, G. R. S. Weir, and M. Alazab, "From ZeuS to Zitmo: Trends in Banking Malware," in *2015 IEEE Trustcom/BigDataSE/ISPA*, 2015, pp. 1386–1391.

[125] D. Kim, B. J. Kwon, and T. Dumitraş, "Certified Malware," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security - CCS '17*, 2017, pp. 1435–1448.

[126] K. Thomas *et al.*, "Data Breaches, Phishing, or Malware?," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security - CCS '17*, 2017, pp. 1421–1434.

[127] Harshit Nayyar, "Clash of the Titans: ZeuS v SpyEye," *SANS Inst. InfoSec Read. Room*, 2010.

[128] A. K. Sood, S. Zeadally, and R. J. Enbody, "An Empirical Study of HTTP-based Financial Botnets," *IEEE Trans. Dependable Secur. Comput.*, vol. 13, no. 2, pp. 236–251, Mar. 2016.

[129] Commerce Cardinal, "Use of Consumer Authentication in eCommerce, Annual Survey 2017: The Fraud Practice," 2017.

[130] HTTPWatch, "HttpWatch 11: HTTP Sniffer for Chrome, IE, iPhone and iPad." [Online]. Available: https://www.httpwatch.com/. [Accessed: 08-May-2018].

[131] Esolutions, "Live Http Header." [Online]. Available: https://www.esolutions.se/. [Accessed: 08-May-2018].

[132] WickyBay.Inc, "FRAUDFOX VM | WickyBay Store." [Online]. Available: http://store.wickybay.xyz/product/fraudfox-vm/. [Accessed: 08-May-2018].

[133] AndroidPay, "Google Pay (UK) – Pay in apps, on the web, and in stores," 2014. [Online]. Available: https://pay.google.com/about/. [Accessed: 31-Aug-2018].

[134] A. Inc, "iOS Security Guide - iOS 11.4," 2018. [Online]. Available: https://www.apple.com/business/site/docs/iOS_Security_Guide.pdf. [Accessed: 30-Apr-2018].

[135] FIDO2018, "FIDO Alliance Intel, Lenovo, PayPal and Synaptics Collaborate to Accelerate FIDO Adoption on the Desktop - FIDO Alliance." [Online]. Available:

https://fidoalliance.org/intel-lenovo-paypal-and-synaptics-collaborate-to-accelerate-fido-adoption-on-the-desktop/. [Accessed: 08-May-2018].

[136] FIDO, "FIDO Alliance Specification," 2018. [Online]. Available: https://fidoalliance.org/specs/. [Accessed: 31-Aug-2018].

[137] K. Markantonakis, M. Tunstall, G. Hancke, I. Askoxylakis, and K. Mayes, "Attacking smart card systems: Theory and practice," *Inf. Secur. Tech. Rep.*, vol. 14, no. 2, pp. 46–56, May 2009.

[138] J. Van Den Breekel, D. A. Ortiz-Yepes, E. Poll, and J. De Ruiter, "EMV in a nutshell," 2016. [Online]. Available: https://www.cs.ru.nl/E.Poll/papers/EMVtechreport.pdf. [Accessed: 30-Apr-2018]

[139] C. Herley, P. C. van Oorschot, and A. S. Patrick, "Passwords: If We're So Smart, Why Are We Still Using Them?," Springer, Berlin, Heidelberg, 2009, pp. 230–237.

[140] Visa, "A guide for merchants Managing the card not present fraud environment," 2014.

[141] J. Manyika *et al.*, "The Internet of Things: Mapping the value beyond the hype," *McKinsey Glob. Inst.*, no. June, p. 144, 2015.

[142] R. Alur *et al.*, "Systems Computing Challenges in the Internet of Things," 2015. [Online]. Available: https://arxiv.org/ftp/arxiv/papers/1604/1604.02980.pdf. [Accessed: 30-Apr-2018]

[143] w3c.org, "Payment Request API," 2018. [Online]. Available: https://www.w3.org/TR/payment-request/. [Accessed: 30-Apr-2018].

Appendix A.

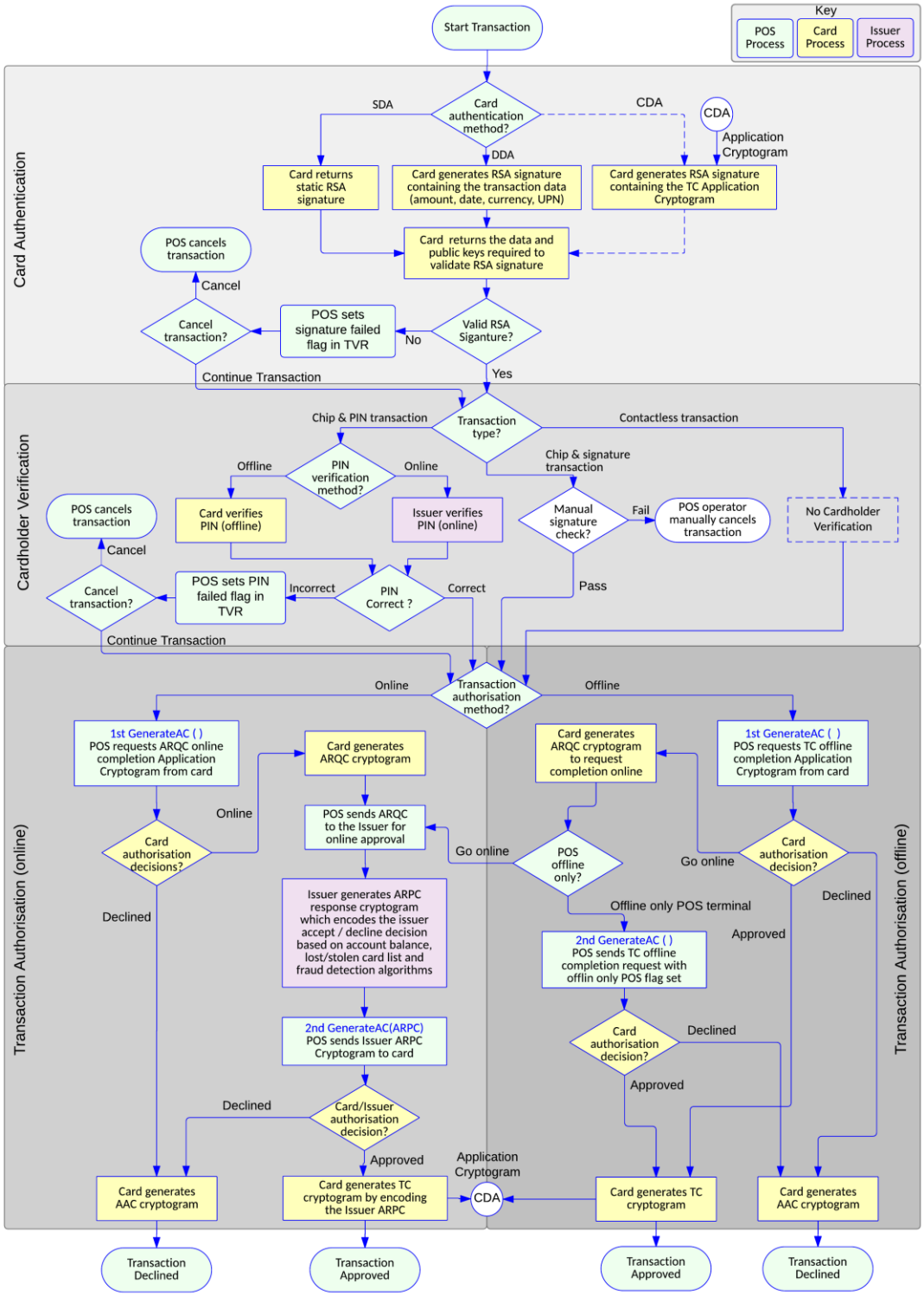

Figure 50 - EMV Transaction Sequence (This figure is taken from [29])

Appendix B.

Table 21 - The website-bot sample code to automate vulnerability assessements over Google wallet website

```
package com.uk.ac.ncl.Bot;
        import com.gargoylesoftware.htmlunit.ElementNotFoundException;
        import org.openqa.selenium.By;
        import org.openqa.selenium.ElementNotVisibleException;
        import org.openqa.selenium.WebDriver;
        import org.openqa.selenium.WebElement;
        import org.openqa.selenium.firefox.FirefoxDriver;
        import java.util.NoSuchElementException;
/**
 * Created by Mohammed on 19/04/2015.
 */
public class GoogleWallet implements Runnable {
    static String email = "xxxxx";
    static String passwd = "xxxxx";
    static String cardNum;
    static String expMM;
    static String expYY;
    static String csc;
    public boolean flag=false;
    int from, to;

    public GoogleWallet(String cardNum, String expMM, String expYY, String csc, int from,
int to) {
        this.cardNum = cardNum;
        this.expMM = expMM;
        this.expYY = expYY;
        this.csc = csc;
        this.from = from;
        this.to = to;
    }

    public void run() {
            //System.out.println("I Started with
values"+cardNum+"\n"+expMM+"\n"+expYY+"\n"+csc+"\n"+from+"\n"+to);
            final WebDriver driver = new FirefoxDriver();
            driver.get("https://accounts.google.com/ServiceLoginAuth");
            WebElement element1;
            element1 = driver.findElement(By.id("Email"));
            element1.sendKeys(email);
            element1 = driver.findElement(By.id("Passwd"));
            element1.sendKeys(passwd);
            element1 = driver.findElement(By.id("PersistentCookie"));
            element1.click();
            element1 = driver.findElement(By.id("signIn"));
            element1.submit();
driver.navigate().to("https://wallet.google.com/manage/#CreateInstrumentPlace:");
            element1 = driver.findElement(By.xpath("//*[@id='gwt-debug-
CreateInstrumentView-addCreditCardWidget-creditCardWidget-ccForm-creditcard-number-
field']"));
            element1.sendKeys(cardNum);
            element1 = driver.findElement(By.xpath("//*[@id='gwt-debug-
CreateInstrumentView-addCreditCardWidget-creditCardWidget-ccForm-creditcard-month-
field']"));
            element1.sendKeys(expMM);
            expYY = expYY.substring(2,4);
            element1 = driver.findElement(By.xpath("//*[@id='gwt-debug-
CreateInstrumentView-addCreditCardWidget-creditCardWidget-ccForm-creditcard-year-
field']"));
            element1.sendKeys(expYY);
            element1 = driver.findElement(By.xpath("//*[@id='gwt-debug-
CreateInstrumentView-addCreditCardWidget-creditCardWidget-ccForm-creditcard-cvc-field']"));
            element1.sendKeys(csc);
            element1 = driver.findElement(By.xpath("//*[@id='gwt-debug-
CreateInstrumentView-addCreditCardWidget-saveButton']"));
            element1.click();
            System.out.println("Trying CVV on Google Wallet... Please be patient");
            for (int i = from; i <= to; i++) {
                String cse = Integer.toString(i);
```

```
                try {
                    Sleep.sleep(10000);
                    element1 = driver.findElement(By.xpath("//*[@id='gwt-debug-
CreateInstrumentView-addCreditCardWidget-creditCardWidget-ccForm-creditcard-cvc-field']"));
                    element1.clear();
                    element1.sendKeys(cse);
                    element1 = driver.findElement(By.xpath("//*[@id='gwt-debug-
CreateInstrumentView-addCreditCardWidget-saveButton']"));
                    Sleep.sleep(2000);
                    element1.click();
                    Sleep.sleep(10000);
                }
                catch (Exception e) {
                    System.err.println("CVV for the Card: " + cardNum);
                    System.err.print(i - 1);
                    flag = true;
                    break;
                }
                if(flag=false){
                    System.out.println("Not Found");
                }
            }
    }

}
```

Appendix C.

Table 22 - List of 25 active underground forums where payment card details are traded

Forum name	Forum address	Forum name	Forum address
Agoraforum	Lacbzxobeprssfx.onion	Bus1nezz	Bus1nezz.biz
Altenen	Altenen.com	Cardingmafia	Cardingmafia.ws
Crdpro	Crdpro.su	Bpcsqad	Bpcsquad.com
Crimenetwork	Crimenc5wxi63f4r.onion	Procarder	Procarder.ru
Cardingforum	Cardingforum.org	Cardersforum	Cardersforum.se
Hackingforum	Hackingforum.ru	Crimes	Crimes.ws
Unixorder	Unixorder.com	Carderbase	Carderbase.su
Crdclub	Crdclub.ws	Carder	Carder.me
Carderscave	Carderscave.ru	Darkstuff	Darkstuff.net
Infraud	Infraud.cc	Coinodeal	Coinodeal.com
Lampeduza	Lampeduza.so	Tuxedocrew	Tuxedocrew.biz
Blackstuff	Blackstuff.net/forum.php	Privatemarket	Privatemarket.us
Omerta	Omerta.cm		

Appendix D.

Table 23 - Survey results showing distributed attack landscaping exercise performed on over 400 commercial websites.

Alexa Rank	Website Name	PAN	Expiry Date	CVV2	Address	No of Attempts	Leak
1.	Amazon	YES	YES	NO	NO	Unlimited	Expiry Date
2.	Ebay (Paypal)	YES	YES	YES	YES	10	Postal Code
3.	Netflix	YES	YES	YES	NO	10	CVV2
4.	Walmart	YES	YES	YES	NO	10	CVV2
5.	Etsy	YES	YES	YES	NO	20	CVV2
6.	Bestbuy	YES	YES	YES	No	10	CVV2
7.	Ikea	YES	YES	YES	YES	4	CVV2/Postalcode
8.	Target	-	-	-	-	-	-
9.	homedepot	-	-	-	-	-	-
10.	Steampowered	YES	YES	YES	NO	10	CVV2
11.	Newegg	YES	YES	YES	NO	4	CVV2/PostalCode
12.	Macys	YES	YES	YES	NO	20	CVV2
13.	Lowes	YES	YES	YES	NO	5	CVV2
14.	Nordstrom	YES	YES	YES	NO	20	CVV2
15.	Kohls	YES	YES	YES	NO	10	CVV2
16.	Gap	YES	YES	YES	NO	4	CVV2/PostalCode
17.	Costco	YES	YES	YES	NO	4	CVV2/PostalCode
18.	Hm	YES	YES	YES	NO	4	CVV2/PostalCode
19.	Sears	YES	YES	YES	NO	10	CVV2
20.	6pm	YES	YES	NO	NO	10	Expiry Date
21.	Nike	YES	YES	YES	NO	20	CVV2
22.	Bodybuilding	YES	YES	YES	NO	20	CVV2
23.	Overstock	YES	YES	YES	NO	4	CVV2/Postalcode
24.	Staples	YES	YES	YES	NO	25	CVV2
25.	Bhphotovideo	YES	YES	YES	no	10	CVV2
26.	Groupon	YES	YES	YES	NO	25	CVV2
27.	Forever21	YES	YES	YES	NO	4	CVV2/PostalCode
28.	Ticketmaster	YES	YES	YES	NO	10	CVV2
29.	Jcpenney	YES	YES	YES	NO	10	CVV2
30.	Zappos	YES	YES	YES	NO	10	CVV2
31.	Sky	YES	YES	YES	NO	10	CVV2
32.	Bedbathandbeyond	YES	YES	YES	NO	10	CVV2
33.	Walgreens	YES	YES	NO	NOT/SURE	20	Expiry Date
34.	Barnesandnoble	YES	YES	YES	NO	20	CVV2
35.	Legacy	YES	YES	YES	NO	20	CVV2
36.	Google.com/shopping	-	-	-	-	-	-
37.	Qvc	YES	YES	NO	NO	5	Expiry Date
38.	Wiley	YES	YES	YES	NO	10	CVV2
39.	Gamestop	YES	YES	YES	YES	4	CVV2/PostalCode
40.	Autotrader	YES	YES	YES	NO	10	CVV2

41.	Samsclub	-	-	-	-	-	-
42.	Cvs	YES	YES	YES	NO	10	CVV2
43.	Victoriasecret	YES	YES	YES	NO	10	CVV2
44.	Cars	-	-	-	-	-	-
45.	Rei	YES	YES	YES	NO	10	CVV2
46.	Rakuten	YES	YES	YES	NO	10	CVV2
47.	Sephora	YES	YES	YES	NO	15	CVV2
48.	Shutterfly	YES	YES	YES	NO	10	CVV2
49.	Iherb	YES	YES	YES	NO	10	CVV2
50.	Officedepot	YES	YES	YES	NO	10	CVV2
51.	Livingsocial	YES	YES	YES	NO	4	CVV2/PostalCode
52.	Jcrew	YES	YES	YES	NO	10	CVV2
53.	Dx	YES	YES	YES	YES	10	Postal Code
54.	Humblebundle	YES	YES	YES	NO	10	CVV2
55.	Kmart	YES	YES	YES	NO	10	CVV2
56.	Hsn	YES	YES	YES	NO	4	CVV2/PostalCode
57.	Cambridge	YES	YES	YES	NO	4	CVV2/PostalCode
58.	Tigerdirect	YES	YES	YES	NO	10	CVV2
59.	Dickssportinggoods	YES	YES	YES	NO	4	CVV2/PostalCode
60.	Directv	YES	YES	YES	YES	10	Postal Code
61.	Trademe	YES	YES	YES	NO	10	CVV2
62.	Urbanoutfitters	YES	YES	YES	NO	10	CVV2
63.	Sierratradingpost	YES	YES	YES	NO	4	CVV2/PostalCode
64.	Frys	YES	YES	YES	YES	10	Postal Code
65.	Cabelas	YES	YES	YES	NO	10	CVV2
66.	Ralphlauren	YES	YES	YES	No	4	CVV2/PostalCode
67.	Redbubble	YES	YES	NO	NO	10	Expiry Date
68.	Yoox	YES	YES	YES	YES	4	CVV2/PostalCode
69.	eshop.Sonymobile.com	YES	YES	YES	NO	25	CVV2
70.	Mango	YES	YES	YES	NO	4	CVV2/PostalCode
71.	Ae	YES	YES	YES	NO	8	CVV2
72.	Landsend	YES	YES	YES	YES	10	Postal Code
73.	Drugstore	YES	YES	YES	NO	10	CVV2
74.	Ulta	YES	YES	YES	NO	10	CVV2
75.	Blu-ray	YES	YES	YES	No	10	CVV2
76.	Cargurus	-	-	-	-	-	-
77.	Bloomingdales	YES	YES	YES	NO	10	CVV2
78.	Llbean	YES	YES	YES	NO	10	CVV2
79.	Neimanmarcus	YES	YES	YES	NO	10	CVV2
80.	Asda	YES	YES	YES	NO	5	Postal Code
81.	Marksandspencer	YES	YES	YES	NO	4	CVV2/PostalCode
82.	Potterybarn	YES	YES	YES	NO	20	CVV2
83.	Net-a-porter	-	-	-	-	-	-
84.	Harborfreight	-	-	-	-	-	-
85.	Shopbop	YES	YES	YES	NO	20	CVV2
86.	Modcloth	YES	YES	YES	NO	10	CVV2

87.	Petsmart	-	-	-	-	-	-
88.	Gnc	YES	YES	YES	NO	4	CVV2/PostalCode
89.	Anthropologie	YES	YES	YES	NO	10	CVV2
90.	Saksfifthavenue	YES	YES	YES	NO	4	CVV2
91.	Backcountry	YES	YES	YES	NO	4	CVV2/PostalCode
92.	Siriusxm	YES	YES	YES	NO	10	CVV2
93.	Carmax	-	-	-	-	-	-
94.	Mapsofindia	YES	YES	YES	NO	10	CVV2
95.	cafepress	YES	YES	NO	NO	10	Expiry Date
96.	Petco	YES	YES	YES	NO	5	CVV2
97.	Adorama	YES	YES	YES	NO	5	CVV2
98.	Finishline	YES	YES	YES	NO	10	CVV2
99.	Joann	YES	YES	YES	No	10	CVV2
100.	Mobikwik	YES	YES	YES	NO	4	CVV2/PostalCode
101.	Vitacost	-	-	-	-	-	-
102.	Disneystore	YES	YES	YES	No	4	CVV2/PostalCode
103.	Westelm	YES	YES	YES	No	10	CVV2
104.	William-sonoma	YES	YES	YES	No	5	CVV2
105.	Sportsauthority	YES	YES	YES	No	10	CVV2
106.	Dsw	YES	YES	YES	No	20	CVV2
107.	Crateandbarrel	YES	YES	YES	No	5	CVV2
108.	Wightwatchers	-	-	-	-	-	-
109.	Musiciansfriend	YES	YES	YES	No	10	CVV2
110.	Hottopic	-	-	-	-	-	-
111.	Childresplace	YES	YES	YES	No	5	CVV2
112.	Bonanza	YES	YES	YES	NO	10	CVV2
113.	Abercrombie	YES	YES	YES	No	20	CVV2
114.	Shop.lego.com	YES	YES	YES	No	10	CVV2
115.	Gunbroker	-	-	-	-	-	-
116.	Underarmour	-	-	-	-	-	-
117.	Ebags	YES	YES	YES	No	20	CVV2
118.	Worldmarket	YES	YES	YES	NO	10	CVV2
119.	Sweetwater	-	-	-	-	-	-
120.	Fineartamerica	YES	YES	YES	NO	20	CVV2
121.	Techbargains	YES	YES	YES	NO	4	CVV2/PostalCode
122.	Crutchfield	YES	YES	YES	NO	5	CVV2
123.	Pier1	YES	YES	YES	NO	20	CVV2
124.	Anntaylor	YES	YES	YES	NO	10	CVV2
125.	Jomashop	-	-	-	-	-	-
126.	Nespresso	-	-	-	-	-	-
127.	Oldnavy	YES	YES	YES	NO	10	CVV2
128.	Freepople	-	-	-	-	-	-
129.	Advanceautoparts	YES	YES	YES	NO	10	CVV2
130.	Basspro	YES	YES	YES	NO	10	CVV2
131.	Midwayusa	YES	YES	YES	NO	20	CVV2
132.	Dish	-	-	-	-	-	-

133.	Bathandbodyworks	YES	YES	YES	NO	20	CVV2
134.	Orientaltrading	YES	YES	NO	NO	10	Expiry Date
135.	Monoprice	YES	YES	YES	NO	20	CVV2
136.	Containerstore	YES	YES	YES	NO	5	CVV2
137.	Academy	YES	YES	YES	NO	10	CVV2
138.	Tirerack	YES	YES	YES	NO	4	CVV2/PostalCode
139.	Shoebuy	YES	YES	NO	NO	10	Expiry Date
140.	Scholasic	YES	YES	YES	NO	10	CVV2
141.	Autotrader	-	-	-	-	-	-
142.	Dillards	YES	YES	YES	NO	5	CVV2
143.	Ebay/motors	-	-	-	-	-	-
144.	Luluemon	YES	YES	YES	NO	4	CVV2/PostalCode
145.	Fingerhut	YES	YES	YES	NO	10	CVV2
146.	Ashford	YES	YES	YES	NO	10	CVV2
147.	Aeropostale	YES	YES	YES	NO	10	CVV2
148.	Customink	YES	YES	YES	NO	10	CVV2
149.	Shopgoodwill	-	-	-	-	-	-
150.	Gymboree	YES	YES	YES	NO	10	CVV2
151.	Polar	YES	YES	YES	NO	4	CVV2/PostalCode
152.	Snapfish	YES	YES	YES	NO	10	CVV2
153.	Lordandtaylor	YES	YES	YES	No	10	CVV2
154.	Swansonvitamins	YES	YES	YES	No	10	CVV2
155.	Lanebryant	YES	YES	YES	NO	10	CVV2
156.	Acehardware	YES	YES	YES	NO	4	CVV2/PostalCode
157.	Oakley	YES	YES	YES	NO	10	CVV2
158.	Rockauto	-	-	-	-	-	-
159.	Moo	YES	YES	YES	NO	10	CVV2
160.	Sportsmansguide	YES	YES	NO	NO	5	Expiry Date
161.	Northerntool	YES	YES	YES	NO	10	CVV2
162.	Tickets	-	-	-	-	-	-
163.	Express-scripts	-	-	-	-	-	-
164.	Zennioptical	YES	YES	YES	NO	10	CVV2
165.	Micromaxinfo	-	-	-	-	-	-
166.	Instyle	-	-	-	-	-	-
167.	Coach	-	-	-	-	-	-
168.	Revolveclothing	YES	YES	YES	NO	20	CVV2
169.	Pacsun	YES	YES	YES	NO	20	CVV2
170.	Brooksbrothers	YES	YES	YES	NO	20	CVV2
171.	Converse	YES	YES	YES	YES	20	Postal Code
172.	Beachbody	YES	YES	NO	NO	10	Expiry Date
173.	Tractorsupply	YES	YES	YES	NO	20	CVV2
174.	Bjs	YES	YES	YES	NO	10	CVV2
175.	Summitracing	YES	YES	YES	NO	4	CVV2/PostalCode
176.	Davidsbridal	-	-	-	-	-	-
177.	Gamestop	YES	YES	YES	NO	4	CVV2/PostalCode
178.	Payless	YES	YES	YES	NO	10	CVV2

179.	Buybuybaby	YES	YES	YES	NO	10	CVV2
180.	Eddiebauer	YES	YES	YES	NO	20	CVV2
181.	Adpost	YES	YES	YES	NO	10	CVV2
182.	Cheaperthandirt	YES	YES	YES	NO	10	CVV2
183.	Bose	YES	YES	YES	NO	5	CVV2
184.	Colehaan	YES	YES	YES	NO	5	CVV2
185.	Watchuseek	YES	YES	YES	NO	4	CVV2/PostalCode
186.	Drsfostersmith	YES	YES	YES	NO	10	CVV2
187.	Puritan	YES	YES	NO	NO	10	Expiry Date
188.	Movietickets	-	-	-	-	-	-
189.	Autoanything	YES	YES	YES	NO	10	CVV2
190.	Famousfootwear	YES	YES	YES	NO	10	CVV2
191.	Toryburch	YES	YES	YES	NO	4	CVV2/PostalCode
192.	Zumiez	YES	YES	YES	NO	10	CVV2
193.	Pandora	-	-	-	-	-	-
194.	Tennis-warehouse	YES	YES	YES	NO	5	CVV2
195.	Funimation	YES	YES	YES	YES	5	Postal Code
196.	Shop.mlb	YES	YES	YES	NO	4	CVV2/PostalCode
197.	Tradepub	-	-	-	-	-	-
198.	Pbteen	YES	YES	YES	NO	4	CVV2
199.	Tillys	YES	YES	YES	NO	10	CVV2
200.	Fabric	YES	YES	YES	NO	5	CVV2
201.	Bkstr	YES	YES	YES	NO	10	CVV2
202.	Keurig	YES	YES	YES	NO	10	CVV2
203.	Art	YES	YES	YES	No	10	CVV2
204.	Lulus	YES	YES	YES	No	10	CVV2
205.	Onlineshoes	YES	YES	YES	NO	4	CVV2/PostalCode
206.	Swimoutlet	-	-	-	-	-	-
207.	Cduniverse	YES	YES	YES	NO	5	CVV2
208.	Lampsplus	YES	YES	YES	NO	10	CVV2
209.	Focalprice	YES	YES	YES	NO	10	CVV2
210.	Restorationhardware	YES	YES	YES	NO	10	CVV2
211.	Autopartswarehouse	YES	YES	YES	NO	10	CVV2
212.	Vans	YES	YES	YES	NO	10	CVV2
213.	Tradesy	YES	YES	YES	NO	10	CVV2
214.	Campingworld	YES	YES	YES	NO	10	CVV2
215.	Lids	YES	YES	YES	NO	10	CVV2
216.	Newbalance	YES	YES	YES	NO	10	CVV2
217.	Chicos	YES	YES	YES	NO	10	CVV2
218.	Dollarshaveclub (Check this security)	YES	YES	YES	NO	5	CVV2
219.	Bluefly	YES	YES	YES	NO	10	CVV2
220.	Swappa	-	-	-	-	-	-
221.	Musicnotes	YES	YES	YES	NO	10	CVV2
222.	Torrid	YES	YES	YES	NO`	10	CVV2
223.	Oup	YES	YES	YES	NO	4	CVV2/PostalCode
224.	Performacebike	YES	YES	YES	NO	10	CVV2

225.	Gandermountain	YES	YES	YES	NO	10	CVV2
226.	Boostmobile	YES	YES	YES	NO	10	CVV2
227.	Dickblick	YES	YES	YES	NO	10	CVV2
228.	Golfsmith	YES	YES	YES	NO	20	CVV2
229.	Roomstogo	YES	YES	YES	NO	10	CVV2
230.	Cdbaby	-	-	-	-	-	-
231.	Womanwithin	YES	YES	YES	NO	10	CVV2
232.	Talbots	YES	YES	YES	NO	10	CVV2
233.	Shoes	YES	YES	YES	NO	10	CVV2
234.	Forsalebyowner	-	-	-	-	-	-
235.	Warbyparker	YES	YES	YES	No	10	CVV2
236.	Maccosmetics	YES	YES	YES	No	10	CVV2
237.	Primagames	YES	YES	YES	No	10	CVV2
238.	Dicksmith	-	-	-	-	-	-
239.	Grasscity	YES	YES	YES	No	10	CVV2
240.	Reebok	YES	YES	YES	No	10	CVV2
241.	Starcitygames	-	-	-	-	-	-
242.	Brownells	YES	YES	YES	No	10	CVV2
243.	Copart	-	-	-	-	-	-
244.	Repairclinic	YES	YES	YES	No	10	CVV2
245.	Bluenile	-	-	-	-	-	-
246.	Tiffany	YES	YES	YES	Yes	4	CVV2/PostalCode
247.	Vitamishoppe	YES	YES	YES	No	10	CVV2
248.	Surlatable	Yes	YES	YES	No	10	CVV2
249.	Quill	YES	YES	YES	No	10	CVV2
250.	Competitivecyclist	YES	YES	YES	No	10	CVV2
251.	Tumi	YES	YES	YES	No	10	CVV2
252.	Apmex	YES	YES	YES	No	10	CVV2
253.	Kingarthurflour	YES	YES	NO	No	10	Expiry Date
254.	Loccitane	YES	YES	YES	No	4	CVV2/PostalCode
255.	1800petmeds	YES	YES	YES	No	10	CVV2
256.	Uncommongoods	YES	YES	YES	No	10	CVV2
257.	Auctionzip	-	-	-	-	-	-
258.	Gamefly	-	-	-	-	-	-
259.	Autos.yahoo	-	-	-	-	-	-
260.	Opticsplanet	Yes	yes	yes	no	10	cvv2
261.	Bigbadtoystore	YES	YES	YES	No	10	CVV2
262.	Homedecorators	YES	YES	YES	No	10	CVV2
263.	Arcamax	-	-	-	-	-	-
264.	Sunglsshut	YES	YES	YES	No	10	CVV2
265.	Venus	YES	YES	YES	No	10	CVV2
266.	Sideshowtoy	YES	YES	YES	No	10	CVV2
267.	Etrailer	YES	YES	YES	No	10	CVV2
268.	Zavvi	YES	YES	YES	No	10	CVV2
269.	Jcwhitney	YES	YES	YES	No	10	CVV2
270.	Chapters.indigo	YES	YES	YES	No	10	CVV2

271.	Bonton	YES	YES	YES	s	10	CVV2
272.	Boscovs	YES	YES	YES	NO	4	CVV2/PostalCode
273.	Peapod	-	-	-	-	-	-
274.	Alibris	YES	YES	YES	No	10	CVV2
275.	Mandmdirect	YES	YES	YES	NO	4	CVV2/PostalCode
276.	Radioshack	-	-	-	-	-	-
277.	Duckindonuts	-	-	-	-	-	-
278.	Youmail	YES	YES	YES	NO	10	CVV2
279.	Threadless	YES	YES	YES	NO	10	CVV2
280.	Firemountaingems	YES	YES	YES	NO	10	CVV2
281.	Brookstone	YES	YES	YES	NO	10	CVV2
282.	Bananarepublic	YES	YES	YES	NO	10	CVV2
283.	Barenecessities	YES	YES	YES	No	10	CVV2
284.	Fossil	YES	YES	YES	No	4	CVV2/PostalCode
285.	Duluthtrading	YES	YES	NO	No	20	Expiry Date
286.	Westmarine	YES	YES	YES	No	10	CVV2
287.	Burlingtoncoatfactory	Yes	YES	YES	No	10	CVV2
288.	hallmark	YES	YES	YES	No	10	CVV2
289.	Giant-bicycles	YES	YES	YES	Yes	10	Postal Code
290.	Tributes	-	-	-	-	-	-
291.	Menswearhouse	YES	YES	YES	No	10	CVV2
292.	Bricklink	-	-	-	-	-	-
293.	Rockler	YES	YES	YES	NO	4	CVV2/PostalCode
294.	Broadway	-	-	-	-	-	-
295.	Abt	-	-	-	-	-	-
296.	Zales	YES	YES	YES	No	4	CVV2/PostalCode
297.	Hhgregg	-	-	-	-	-	-
298.	Framesdirect	Yes	YES	YES	No	10	CVV2
299.	Eurocarparts	YES	YES	YES	No	4	CVV2/PostalCode
300.	Minte	-	-	-	-	-	-
301.	Bergdorfgoodman	YES	YES	YES	No	10	CVV2
302.	Hemmings	-	-	-	-	-	-
303.	Roadrunnersports	YES	YES	YES	No	10	CVV2
304.	Orvis	YES	YES	NO	No	10	Expiry Date
305.	Jjill	YES	YES	YES	No	10	CVV2
306.	Otterbox	YES	YES	YES	No	10	CVV2
307.	Journeys	Yes	yes	yes	no	10	cvv2
308.	Therealreal	Yes	yes	yes	no	10	cvv2
309.	Autobytel	-	-	-	-	-	-
310.	Sheetmusicplus	Yes	yes	yes	No	10	CVV2
311.	Gazelle	YES	YES	YES	No	10	CVV2
312.	Nashbar	YES	YES	YES	No	10	CVV2
313.	Landofnod	YES	YES	YES	No	10	CVV2
314.	Skechers	Yes	YES	YES	No	10	CVV2
315.	Fragrancenet	YES	YES	YES	No	10	CVV2
316.	Timberland	YES	YES	YES	No	10	CVV2

317.	Mec.ca	YES	YES	YES	No	10	CVV2
318.	Harrods	YES	YES	YES	No	4	CVV2/PostalCode
319.	Informit	YES	YES	YES	No	10	CVV2
320.	Cdjapan	YES	YES	YES	No	4	CVV2/PostalCode
321.	Napaonline	YES	YES	YES	No	10	CVV2
322.	cooking	YES	YES	YES	NO	10	CVV2
323.	Entertainmentearth	YES	YES	YES	No	10	CVV2
324.	Schwans	YES	YES	YES	No	10	CVV2
325.	Brownpapertickets	YES	YES	YES	NO	10	CVV2
326.	Vodafone	-	-	-	-	-	-
327.	Kirklands	YES	YES	YES	NO	10	CVV2
328.	Epage	-	-	-	-	-	-
329.	Businessesforsale	-	-	-	-	-	-
330.	Jpcycles	YES	YES	YES	NO	10	CVV2
331.	Hermes	YES	YES	YES	NO	4	CVV2/PostalCode
332.	Frontgate	YES	YES	YES	NO	10	CVV2
333.	Footsmart	YES	YES	YES	No	10	CVV2
334.	Tivo	YES	YES	YES	No	10	CVV2
335.	Jensonusa	YES	YES	Yes	No	10	CVV2
336.	Appliancepartspros	YES	YES	NO	No	10	Expiry Date
337.	Builddirect	Yes	YES	YES	No	10	CVV2
338.	Lionbrand	YES	YES	YES	No	4	CVV2
339.	Cracker	YES	YES	YES	No	10	CVV2
340.	Fender	-	-	-	-	-	-
341.	Iboats	YES	YES	YES	No	4	CVV2/PostalCode
342.	Personalizationmall	YES	YES	YES	No	10	CVV2
343.	Glassesusa	YES	YES	YES	NO	10	CVV2
344.	Coolstuffinc	YES	YES	YES	No	10	CVV2
345.	Ballarddesigns	YES	YES	YES	No	10	CVV2
346.	Onofre	-	-	-	-	-	-
347.	Sonicelectronix	Yes	YES	YES	No	10	CVV2
348.	Luckyvitamin	Yes	YES	YES	No	10	CVV2
349.	Worldsoccershop	YES	YES	YES	No	10	CVV2
350.	Tinyprints	YES	YES	NO	No	10	Expiry Date
351.	Fnp	YES	YES	YES	No	4	CVV2/PostalCode
352.	Ajmadison	YES	YES	YES	No	10	CVV2
353.	Bizbuysell	-	-	-	-	-	-
354.	Weddingpaperdivas	YES	YES	NO	No	10	Expiry Date
355.	Wetseal	YES	YES	YES	No	10	CVV2
356.	Zzsounds	YES	YES	YES	No	10	CVV2
357.	Hlj	Yes	yes	yes	no	10	cvv2
358.	Replacements	Yes	yes	NO	No	10	Expiry Date
359.	Carandclassic	-	-	-	-	-	-
360.	Dyson	Yes	yes	yes	No	10	CVV2
361.	Atgstores	YES	YES	YES	No	10	CVV2
362.	Baseballexpress	YES	YES	YES	No	10	CVV2

363.	Jtv	YES	YES	YES	No	10	CVV2
364.	Jpc	-	-	-	-	-	-
365.	Tedbaker	YES	YES	YES	No	10	CVV2
366.	Channelfireball	YES	YES	YES	No	10	CVV2
367.	Ninewest	YES	YES	YES	No	4	CVV2/PostalCode
368.	Yandy	YES	YES	YES	No	10	CVV2
369.	Knifecenter	YES	YES	YES	No	10	CVV2
370.	Gohastings	Yes	YES	YES	No	10	CVV2
371.	Pandahall	YES	YES	YES	No	4	CVV2/PostalCode
372.	Roamans	YES	YES	YES	NO	10	CVV2
373.	Guess	YES	YES	YES	No	10	CVV2
374.	VisaCheckout	YES	YES	YES	No	Unlimited	CVV2
375.	MarterPass	YES	YES	YES	Yes	Unlimited	Postal Code
376.	Yahoo	YES	YES	YES	NO	5	CVV2
377.	Wikipedia	YES	YES	YES	NO	10	CVV2
378.	Oracle	YES	YES	NO	NO	Unlimited	Expiry Date
379.	6pm	YES	YES	NO	NO	10	Expiry Date
380.	Adobe	YES	YES	NO	NO	10	Expiry Date
381.	Apple	YES	YES	YES	NO	Unlimited	CVV2
382.	Google	YES	YES	YES	NO	10	CVV2
383.	Facebook	YES	YES	YES	NO	50	CVV2
384.	Dropbox	YES	YES	YES	No	10	CVV2
385.	OneDrive Microsoft	YES	YES	YES	No	5	CVV2
386.	Skype	YES	YES	YES	No	5	CVV2
387.	Twitter	YES	YES	YES	No	10	CVV2
388.	Netflix	Yes	YES	YES	No	10	CVV2
389.	Macys	YES	YES	YES	No	10	CVV2
390.	Paypal	YES	YES	YES	Yes	Unlimited	Postal Code
391.	Rdio	YES	YES	YES	Yes	50	Postal Code
392.	Udemy	YES	YES	YES	Yes	10	Postal Code
393.	Hellofresh	YES	YES	YES	Yes	10	Postal Code
394.	Yola	YES	YES	YES	Yes	10	Postal Code
395.	Strikingly	YES	YES	YES	Yes	10	Postal Code
396.	Weebly	YES	YES	YES	Yes	10	Postal Code
397.	Webs	YES	YES	YES	Yes	10	Postal Code
398.	Asda	Yes	YES	YES	Yes	5	Postal Code
399.	Converse	Yes	YES	YES	Yes	20	Postal Code
400.	Funimation	YES	YES	YES	Yes	10	Postal Code
401.	Proflowers	YES	YES	YES	No	10	CVV2
402.	Beallsflorida	Yes	Yes	Yes	No	10	CVV2
403.	Emusic	YES	YES	YES	NO	10	CVV2
404.	Garnethill	YES	YES	YES	NO	10	CVV2
405.	Hammacher	Yes	Yes	Yes	No	10	CVV2
406.	Parts-express	YES	YES	YES	NO	10	CVV2
407.	Tgw	YES	YES	YES	NO	10	CVV2
408.	Lakeside	YES	YES	YES	NO	10	CVV2

409.	1800contacts	YES	YES	NO	NO	10	Expiry Date
410.	Soma	YES	YES	YES	No	10	CVV2
411.	Dockers	YES	YES	YES	No	10	CVV2
412.	Partselect	YES	YES	Yes	No	10	CVV2
413.	Swimsuitsforall	YES	YES	YES	No	10	CVV2
414.	Digitalrev						
415.	Omahasteaks	YES	YES	NO	NO	10	Expiry Date
416.	Cirquedusoleil	..	..	..	..	..	..
417.	Citypass	Yes	Yes	Yes	No	..	CVV2
418.	nflshop	Yes	Yes	Yes	No	10	CVV2
419.	Towerhobbies	Yes	Yes	Yes	No	10	CVV2
420.	Avenue	Yes	Yes	Yes	No	10	CVV2
421.	1000bulbs	Yes	Yes	Yes	No	10	CVV2
422.	Clinique	Yes	Yes	No	No	10	Expiry Date
423.	Cigarsinternational	Yes	Yes	No	No	10	Expiry Date
424.	Smilebox	..	..	..	..	…	..
425.	Wickedweasel	Yes	Yes	Yes	No	10	CVV2
426.	Leevalley	Yes	Yes	No	No	10	Expiry Date
427.	Naturebox	Yes	Yes	Yes	No	10	CVV2
428.	Claires	Yes	Yes	Yes	No	4	CVV2/Postal Code
429.	Moma	Yes	Yes	Yes	No	10	CVV2
430.	Classiccars	..	..	…	..	..	..
431.	Horizonhobby	Yes	Yes	Yes	No	10	CVV2
432.	Yankeecandle	Yes	Yes	Yes	No	10	CVV2
433.	Casper	Yes	Yes	Yes	No	10	CVV2
434.	Suunto	Yes	Yes	Yes	No	4	CVV2/Postal Code
435.	Luckybrand	Yes	Yes	Yes	No	10	CVV2
436.	Bikebandit	Yes	Yes	Yes	No	10	CVV2
437.	Quadratec	Yes	Yes	Yes	No	10	CVV2
438.	Eyebuydirect	Yes	Yes	Yes	No	10	CVV2
439.	Karmaloop	Yes	Yes	Yes	No	10	CVV2
440.	Juno	Yes	Yes	Yes	No	10	CVV2
441.	Wine	Yes	Yes	Yes	No	10	CVV2
442.	Timbuk2	Yes	Yes	Yes	No	10	CVV2
443.	Travelsmith	Yes	Yes	Yes	No	10	CVV2
444.	Startrek	Yes	Yes	Yes	No	10	CVV2
445.	Globalhealingcenter	Yes	Yes	Yes	No	10	CVV2
446.	Brooksrunning.com	Yes	Yes	Yes	No	10	CVV2
447.	http://www.hannaandersson.com/	Yes	Yes	Yes	No	10	CVV2
448.	Roomandboard	Yes	Yes	Yes	Yes	10	Postal Code
449.	Americanmuscle	Yes	Yes	Yes	No	4	CVV2/Postal Code
450.	Usfreeads	..	..	..	..	..	..
451.	Skinstore	Yes	Yes	Yes	No	10	CVV2
452.	Overnightprints	Yes	Yes	Yes	Yes	10	Postal Code
453.	Herroom	Yes	Yes	Yes	No	10	CVV2

454.	Buckle	Yes	Yes	Yes	No	10	CVV2
455.	Gardeners	Yes	Yes	Yes	No	10	CVV2
456.	Onstar	..	..	..	..	..	..
457.	Gaiam	Yes	Yes	Yes	No	10	CVV2
458.	Christopherandbanks	Yes	Yes	Yes	No	10	CVV2
459.	Bevmo	Yes	Yes	Yes	Yes	10	Postal Code
460.	Globalgolf	Yes	Yes	Yes	No	10	CVV2
461.	Drjays	Yes	Yes	Yes	No	10	CVV2
462.	Gojane	Yes	Yes	Yes	No	10	CVV2
463.	Carandclassic	..	..	..	..	..	..
464.	Horchow	Yes	Yes	Yes	No	10	CVV2
465.	Well	..	..	..	..	..	..
466.	Jockey	Link to Amazon					
467.	ems	YES	YES	YES	No	10	CVV2
468.	Weathertech	YES	YES	YES	No	10	CVV2
469.	Urbandecay	YES	YES	YES	Yes	10	Postal Code
470.	Forzieri	YES	YES	YES	No	10	CVV2
471.	Gifts	Yes	YES	YES	No	10	CVV2