University of Kent

Kent Academic Repository

Ali, Mohammed Aamir, Arief, Budi, Emms, Martin and van Moorsel, Aad (2017) *Does The Online Card Payment Landscape Unwittingly Facilitate Fraud?* IEEE Security & Privacy, 15 (2). pp. 78-86. ISSN 1540-7993.

Downloaded from
https://kar.kent.ac.uk/58364/ The University of Kent's Academic Repository KAR

The version of record is available from
https://doi.org/10.1109/MSP.2017.27

This document version
Author's Accepted Manuscript

DOI for this version

Licence for this version
UNSPECIFIED

Additional information

Versions of research works

Versions of Record
If this version is the version of record, it is the same as the published version available on the publisher's web site. Cite as the published version.

Author Accepted Manuscripts
If this document is identified as the Author Accepted Manuscript it is the version after peer review but before type setting, copy editing or publisher branding. Cite as Surname, Initial. (Year) 'Title of article'. To be published in *Title of Journal* , Volume and issue numbers [peer-reviewed accepted version]. Available at: DOI or URL (Accessed: date).

Enquiries
If you have questions about this document contact ResearchSupport@kent.ac.uk. Please include the URL of the record in KAR. If you believe that your, or a third party's rights have been compromised through this document please see our Take Down policy (available from https://www.kent.ac.uk/guides/kar-the-kent-academic-repository#policies).

Does The Online Card Payment Landscape Unwittingly Facilitate Fraud?

Mohammed Aamir Ali, Budi Arief, Martin Emms, and Aad van Moorsel

Abstract—**This article provides an extensive study of the current practice of online payment using credit and debit cards, and the intrinsic security challenges caused by the differences in how payment sites operate. We investigated the Alexa top-400 online merchants' payment sites, and realised that the current landscape facilitates a distributed guessing attack. This attack subverts the payment functionality from its intended purpose of validating card details, into helping the attackers to generate all security data fields required to make online transactions. We will show that this attack would not be practical if all payment sites performed the same security checks. As part of our responsible disclosure measure, we notified a selection of payment sites about our findings, and we report on their responses. We will discuss potential solutions to the problem and the practical difficulty to implement these, given the varying technical and business concerns of the involved parties.**

Keywords—security; online payment; distributed attack; fraudulent transactions; survey; ethical disclosure.

I. INTRODUCTION

Cards are the *de facto* means of paying for online purchases. However, as the value of online sales has increased, so has the amount of online fraud. As an example, UK[1] online sales in 2014 was worth £45 billion, which represents a 16% growth between 2013 and 2014 [1]. In the same time period, the value of online fraud in the UK has increased by 33% to £217 million [1]. Online fraud is now the single largest category of card fraud in the UK, representing 45% of the total value of the fraud committed against UK credit and debit cards [2].

In this article, we present the online payment landscape in detail. In particular, we aim to highlight the different manners in which online payment is performed, and the varying security measures put in place by online merchants – from checking only the card number and the expiry date, to fully-fledged centralised bank security mechanisms such as 3D Secure [3][4][5]. There is a number of questions we would like to address: does the difference cause a security problem? if it does, how common is the problem and can it be exploited? how much damage can be done? and how could it be resolved in the future? To determine the extent of the problem, we survey the 'online payment landscape', creating a mapping of various merchant payment implementations.

We came to an important observation that the difference in security solutions of various websites introduces a practically exploitable vulnerability in the overall payment system. An attacker can exploit these differences to build a distributed guessing attack which generates usable card payment details (card number, expiry date, card verification value, and postal address) *one field at a time*. Each generated field can be used in succession to generate the next field by using a different merchant's website. Moreover, if individual merchants were trying to improve their security by adding more payment fields to be verified on their site, they potentially inadvertently weaken the whole system by creating an opportunity to guess the value of another field, as explained later in the article.

We demonstrate the practicability of exploiting the vulnerabilities with software that implements the distributed guessing attack. We will show that the potential impact of these vulnerabilities is substantial because the card details generated by this distributed attack can be used to transfer money from a victim's bank account to an anonymous recipient overseas using a financial services company such as the Western Union as a conduit.

The vulnerabilities described in this article 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 (see Section VI.D). Because Visa is the most popular payment network in the world, the discovered vulnerabilities greatly affect the entire global online payments system.

We also carried out a responsible disclosure exercise with the payment sites affected by these vulnerabilities. Of the 342 vulnerable websites, we presented our findings to the top-36 of these sites (in terms of the severity of the vulnerabilities and the size of their customer base), monitored their responses, and analysed the changes these websites have implemented to deal with our disclosure. Several websites, including some of the largest and most popular websites in the world, changed their approach to online payment processing after our disclosure, as we will report later in this article. To protect the affected sites,

[1] Sales and fraud statistics from regions other than the UK are less reliable but indicate the same pattern.

we refrain from specifically revealing their names and their vulnerabilities.

Finally, we discuss potential solutions to the problem. We will see that the vulnerabilities are systemic and cannot be protected against in isolation by any individual online merchant or by the issuing bank through improving their own security policies. But first, let us look into how current online payment system operates.

II. Overview of the Online Payment System

An online payment site uses a customer's existing credit or debit card to transfer funds from the customer's bank account into the merchant's bank account. For this to happen, the customer needs to provide their card information during checkout. These pieces of information are then passed to the card issuing bank, who will process the information further before authorising or rejecting the payment request. This process involves a number of parties, each with different responsibilities.

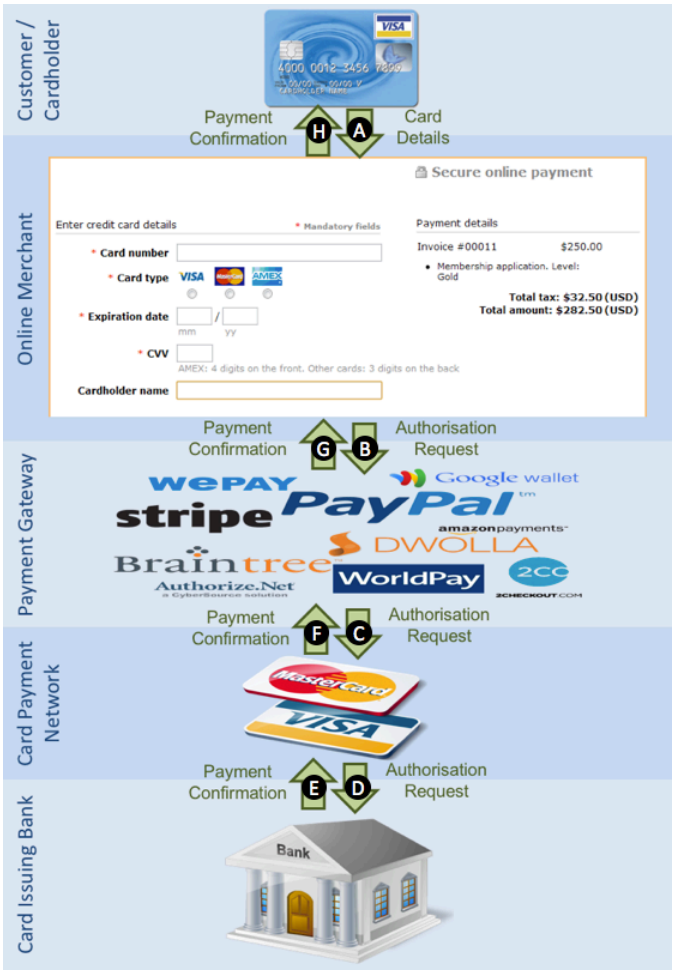

Fig. 1. Actions and parties in online payment.

A. Online Payment Process and Parties Involved

Fig. 1 illustrates the actions and parties involved in processing online payments. The process involves the customer/cardholder entering their payment card details on the payment page of the online merchant's website (action **A** in Fig. 1). The merchant controls which data fields are used to authorise the payment.

The merchant then passes the card details to their chosen payment gateway, which provides a service of authorising and processing the merchant's payment request (action **B**). The payment gateway, on behalf of the merchant, can also implement additional security filters at this point (further details can be found in Section VI.C). The payment gateway 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 (such as Visa and MasterCard) provide the link between payment gateways and the thousands of card issuing banks (actions **C** and **D**).

The card-issuer holds the customer's bank account and makes the approval of the payment (action **E**). The issuer maintains customer's card record file, which contains information such as account balance, customer name, full address, and other card details not visible to the rest of the payment network. In the final step, called a settlement, the card-issuing bank subsequently deposits the customer's money to the merchant's bank account (actions **F**, **G** and **H**).

B. Payment Card Data Fields

An online payment is a "card-not-present" credit or debit card transaction [6]. This implies the merchant cannot physically verify that the customer actually has the card. The security of online payment is therefore dependent upon the customer providing data that only the owner of the card could know.

The payment card industry has developed a Payment Card Industry Data Security Standard (PCI DSS) [7], which provides a comprehensive set of rules and controls for the secure handling and storage of sensitive card data. However, there is no requirement for the merchant to request all of the data fields during an online payment authorisation, nor is there a mandatory requirement for the merchant to implement any of the optional security filters. Five pieces of information are typically used when making an online payment:

- *Cardholder Name*: the account holder's name as printed on the card. We found that no website checks that a name entered is correct.
- *16-digit Card Number*: a unique identifier printed on the front of the card by the issuing bank. Referred to as the *Primary Account Number (PAN)*, it links the card to the customer's bank account.
- *Card Expiry Date*: printed or embossed on the front of the card. The expiry date and the PAN constitute the minimum set of card authentication data.
- *Card Verification Value (CVV2)*: a 3-digit number printed on the reverse side of the card. It is meant to be known only to the person possessing the card. It should not be stored electronically anywhere in the payment ecosystem [7].
- *Cardholder Address*: not visible on the card but sometimes used for payment authorisation purposes. Address verification is performed only on the numerical values of the street/house and postcode fields; any alphabetical characters

are ignored. Different websites perform varying levels of verification on the address field's numerical digits, ranging from verifying just the numerical digits in the postcode (partial match), to the complete numerical digits in postcode plus the door number (full match) [8].

III. Distributed Guessing Attack

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.

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 provide different 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

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).

Guessing an expiry date takes at most 60 attempts (banks typically issue cards that are valid for up to 60 months), and subsequently, guessing the 3-digit CVV2 takes fewer than 1,000 attempts. Hence, expiry date and CVV2 are guaranteed to be obtained within 60 + 1,000 = 1,060 guesses. If all merchants would use three fields and ask for expiry date as well as CVV2, then it may take as many as 60 x 1,000 = 60,000 attempts. The difference between 1,060 and 60,000 is the difference between a quick and practical attack, and a tedious, close to impractical attack.

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. The address field is used in a variety of manners, based on the *Address Verification System (AVS)*, which validates the billing address provided by the customer against the address information stored by the card-issuing bank [6][8][9]. The process of getting cardholder's address for the countries that have a long postcode (more than 3 numerical digits) is not as straightforward as getting the expiry date or CVV2 because first, the attacker will need to narrow down the possible postcodes of the cardholder's address. This can be done by querying the first six digits of a PAN through well-known online databases such as BinDb [10] and ExactBins [11], which will reveal the card's brand, issuing bank name, and card type. Once the issuing bank is known, the attacker can increase the probability of guessing the right postcode by assuming that the victim is likely to 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 (see Section IV.B). Now, the attacker 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.

IV. Experiments

We implemented a set of software tools to carry out the distributed guessing attack, using the research team's own cards to verify that it is indeed possible and practical to obtain all the information of the card. Included are seven Visa cards with a spread of PAN, expiry date, and CVV2 values. We selected 400 Alexa [12] top rated commercial websites for our investigation. These include many global websites such as iTunes, Google, PayPal, and Amazon.

A. Software Tools

The software tools implemented for the experiments consist of a website bot and automated scripts written in Java Selenium browser automation framework [13]. All the experiments were run on Mozilla Firefox web browser. Fig. 2 shows a screenshot of the website bot, which was used to automate the process of guessing relevant card information. The bot cycles through the possible values for each field to find the correct information.

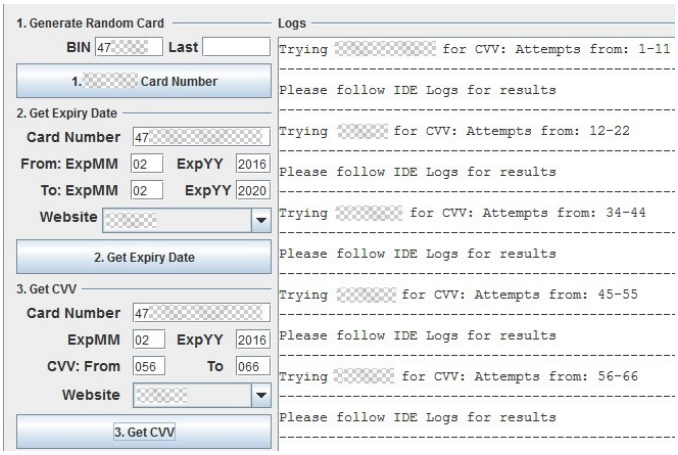

Fig. 2. Screenshot of the website bot, farming CVV2 from multiple sites.

B. Obtaining Card Data

The PAN is the starting point for the generation of all of the other card data fields. There are at least two known methods of obtaining valid PANs. Criminals sell bulk lists of card details online. These lists are considered less valuable when they do not contain the CVV2; nevertheless, such a list could be used as a source of PANs from which the expiry date, CVV2 and address information can be generated. Another method is by exploiting the contactless feature common in recently issued payment

cards. NFC skimming [14] provides an attacker with the PAN, the expiry date and in some cases, the cardholder's name. It is also possible to generate PAN using the first six digits of a PAN and the Luhn's algorithm [15] and getting it verified. However, we did not take this approach because it is crossing the boundary of ethical research—we only used our own cards.

Once the PAN is known, an attempt to obtain the expiry date can commence. We note that sometimes 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, the bot can be used to systematically guess the expiry date of a given PAN on the websites that do not require CVV2 to be entered. The next step in card data generation involves getting the card's CVV2. To find the correct CVV2, the bot will simply need to cycle through the possible values starting from 001 until the payment website blocks further attempts. A handful of payment sites allowed unlimited attempts while most of the other payment sites allowed 5, 10 or even 50 attempts to enter a correct CVV2. In our scenario, we "farm out" the brute force guessing attack to tens or even hundreds of payment systems, which practically means we can carry out unlimited guesses. The final step generates the cardholder's address. An attacker can exploit the different variants of address verification system (discussed in Section III) to find the full address of the cardholder.

C. Transferring the 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 e-commerce 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.

D. Results

Our results (detailed in Table I) show that the distributed guessing attack described in Section III is indeed practical and so a credible threat. We studied and tested the payment website of 389 of Alexa's most visited sites (we looked at the Alexa top-400 sites, but 11 of them did not reveal sufficient useful information for our experiment). As shown in Table I, 26 sites use only two fields for card payment and an attacker would use these sites to guess the expiry date. 291 sites use three fields, which one can use for guessing the CVV2, and 25 sites use four fields, which allows one to guess the postcode of the address. Finally, of the 389 sites, 47 merchants (i.e. 12%) had implemented 3D Secure payments (these sites are impervious to the distributed guessing attack, see Section VI.B).

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. In Table I, the number of sites that allow certain number of guesses is shown in the rows, for each type of site (as represented in the columns). We see that most sites (276) allow between 6 and 10 attempts, but 6 sites set no limit to the number of attempts. There were two notable outliers to this observation in the top-10 highly popular websites, one of which allowed unlimited attempts to guess the CVV2, while the other required only the 16-digit card number plus the expiry date.

Our experiments successfully obtained the valid expiry date for each of our Visa test cards, without exception. We also managed to find valid CVV2 information for our Visa test cards, again without exception. We performed more than 11,000 CVV2 iterations using our bot and scripts, and our experiments confirmed that there is no centrally imposed limit on the number of CVV2 attempts when distributing guesses over multiple websites. The final step is to obtain the address information. Our tests performed more than 3,000 iterations on the group of websites that verify partial address (only postcode digits), to get numerical digits of the postcode. We extended our experiments and run instances of our bot on another set of payment sites (which verify the door number and the postcode digits) in order to get the full address of all our Visa test cards.

TABLE I. VARIATION IN PAYMENT SECURITY SETTINGS OF ONLINE PAYMENTS WEBSITES

Number of attempts allowed	Sites with 2 fields (guess expiry date)	Sites with 3 fields (guess CVV2)	Sites with 4 fields (guess address postcode)	Sites with 3D Secure (safe from attack)	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**

These experiments have also shown that it is possible to run multiple bots at the same time on hundreds of payment sites without triggering any alarms in the payment system. Combining that knowledge with the fact that an online payment request typically gets authorised within 2 seconds makes the attack viable and scalable in real time. As an illustration, with the website bot configured cleverly to run on 30 sites, an attacker can obtain the correct information within 4 seconds.

V. 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 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 of 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 this article, 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 article.

TABLE II. Nature of Patching on the Notified Websites

		Patching Behaviour				
Web site	**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				√	√

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 II 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 as 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 [16]. 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 II 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.

VI. 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 shown in Fig. 1.

A. 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.

B. 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 III.

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 [7], such as the American Express 'SafeKey' [3], 'Verified by Visa' [4] and MasterCard 'SecureCode' [5]. 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 [17]. 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.

C. Payment Gateway

There are many payment gateways, 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 gateways to be able to coordinate sufficiently to prevent the distributed guessing attack. Nevertheless, payment gateways 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, gateways may use IP address velocity filters [6][8][9], 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 gateways, these velocity filters can easily be circumvented just by switching to a website that uses a different payment gateway.

D. Card Payment Network

Responsibility for authorising online payment requests ultimately resides with the bank which issued the credit / debit card. However, our experiments have shown that distributed guessing attack described in the paper 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 [18].

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 gateways, looking for suspicious activities on a single card distributed across multiple merchant websites.

E. 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 [6]. 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 gateways and card payment networks in protecting against the distributed guessing attack.

VII. Conclusion

In this paper, 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.

Acknowledgements

This material is based in part on research supported by the UK EPSRC EP/K006568 Research Institute for Science of Security—Choice Architecture for Information Security.

References

[1] UK Office for National Statistics, "Retail Sales, February 2015," http://www.ons.gov.uk/ons/dcp171778_399119.pdf, 26 March 2015, Accessed: 13 May 2016.

[2] Financial Fraud Action UK, "Fraud the Facts 2015: The definitive overview of payment industry fraud and measures to prevent it," London: UK Cards Association, 2015, www.financialfraudaction.org.uk/download.asp?file=2979, Accessed: 13 May 2016.

[3] American Express SafeKey, "Product Capability Guide," https://network.americanexpress.com/en/globalnetwork/Images/SafeKeyProductCapabilityGuide_2014.pdf, Accessed: 04 Mar 2016.

[4] Visa, "3D Secure System Overview," http://www.visanet.com.pe/verified/demovisanet-web/resources/3DS_70015-01_System_Overview_external_v1.0.3.pdf, 2001, Accessed: 12 May 2016.

[5] MasterCard, "MasterCard Secure Code, "Merchant Implementation Guide," https://www.mastercard.us/content/dam/mccom/en-us/documents/SMI_Manual.pdf, 2014, Accessed: 13 May 2016.

[6] Visa, "Card Acceptance Guidelines for Visa Merchants," http://usa.visa.com/download/merchants/card-acceptance-guidelines-for-visa-merchants.pdf, 2014, Accessed: 13 May 2016

[7] Payment Card Industry, "PCI DSS Applicability in an EMV Environment," https://www.pcisecuritystandards.org/documents/pci_dss_emv.pdf, 2010, Accessed: 13 May 2016.

[8] PayPal, "Gateway Developer Guide and Reference," https://www.paypalobjects.com/webstatic/en_US/developer/docs/pdf/payflowgateway_guide.pdf, 2014, Accessed: 11 Mar 2016.

[9] MasterCard, "Transaction Processing Rules," http://www.mastercard.com/us/merchant/pdf/TPR-Entire_Manual_public.pdf, 2014, Accessed: 13 May 2016.

[10] BinDb, "Bank Identification Numbers Database - Credit Card Bin Lookup," https://www.bindb.com/structure.html, Accessed: 13 May 2016.

[11] ExactBins, "Exact BIN Database," https://www.exactbins.com/features, Accessed: 13 May 2016.

[12] Alexa, "Alexa Top Shopping Sites," http://www.alexa.com/topsites/category/Top/Shopping, Accessed: 13 May 2016.

[13] SeleniumHQ, "Selenium framework documentation. http://docs.seleniumhq.org/docs/, Accessed: 17 May 2016.

[14] 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), ACM, 2014, pp. 716-726.

[15] Symantec, "Validating a Credit Card Number using Luhn's Algorithm. https://support.symantec.com/en_US/article.TECH221769.html [Accessed: 09 May 2016]

[16] A. El Ahmad, J. Yan and W. Ng, "CAPTCHA Design: Color, Usability, and Security", *IEEE Internet Computing*, vol. 16, no. 2, pp. 44-51, 2012.

[17] Adyen, "Analysis Reveals Worldwide Impact of 3D Secure on Transaction Conversion Rates," https://www.adyen.com/home/about-adyen/press-releases/2014/3d-secure-worldwide-impact-conversion, 2014, Accessed: 13 May 2016.

[18] MasterCard, "The MasterCard Network Advantage," http://newsroom.mastercard.com/wp-content/uploads/2011/09/MasterCard-Network-Advantage.pdf, Accessed: 12 May 2016