Does the Online Card Payment Landscape Unwittingly Facilitate Fraud?

Mohammed Aamir Ali | Newcastle University
Budi Arief | University of Kent
Martin Emms and Aad van Moorsel | Newcastle University

An extensive study of the current practice of online payment using credit and debit cards reveals the intrinsic security challenges caused by differences in how payment sites operate.

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 online sales in 2014 were worth £45 billion, representing a 16 percent growth between 2013 and 2014.[1] In the same time period, the value of online fraud in the UK increased by 33 percent to £217 million.[1] Online fraud is now the single largest category of card fraud in the UK, representing 45 percent 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 centralized bank security mechanisms such as 3D Secure.[3–5] We address several questions: Do these differences cause a security problem? If so, 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 the important observation that the difference in various websites' security solutions introduces a practically exploitable vulnerability in the overall payment system. An attacker can exploit these differences to build a distributed guessing attack that generates usable card payment details (card number, expiry date, card verification value [CVV2], 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 might inadvertently weaken the whole system by creating an opportunity to guess the value of another field, as we explain later.

We demonstrate the practicality of exploiting the vulnerabilities with software that implements the distributed guessing attack. The vulnerabilities described in this article apply to cards that don't enforce centralized checks across transactions from different sites. Our experiments were conducted using Visa and MasterCard only.

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, customers must provide their card information during checkout. These pieces of information are then passed to the card-issuing bank, which will process the information further before authorizing or rejecting the payment request. This process involves several parties, each with different responsibilities.

Online Payment: Process and Parties

Figure 1 illustrates the actions and parties involved in processing online payments. The process involves the

 Copublished by the IEEE Computer and Reliability Societies

customer/cardholder entering payment card details on the payment page of the online merchant's website (action A). The merchant controls which data fields are used to authorize the payment.

The merchant then passes the card details to a chosen payment gateway, which provides a service of authorizing 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. 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 approves the payment (action E). The issuer maintains the 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 deposits the customer's money to the merchant's bank account (actions F, G, and H).

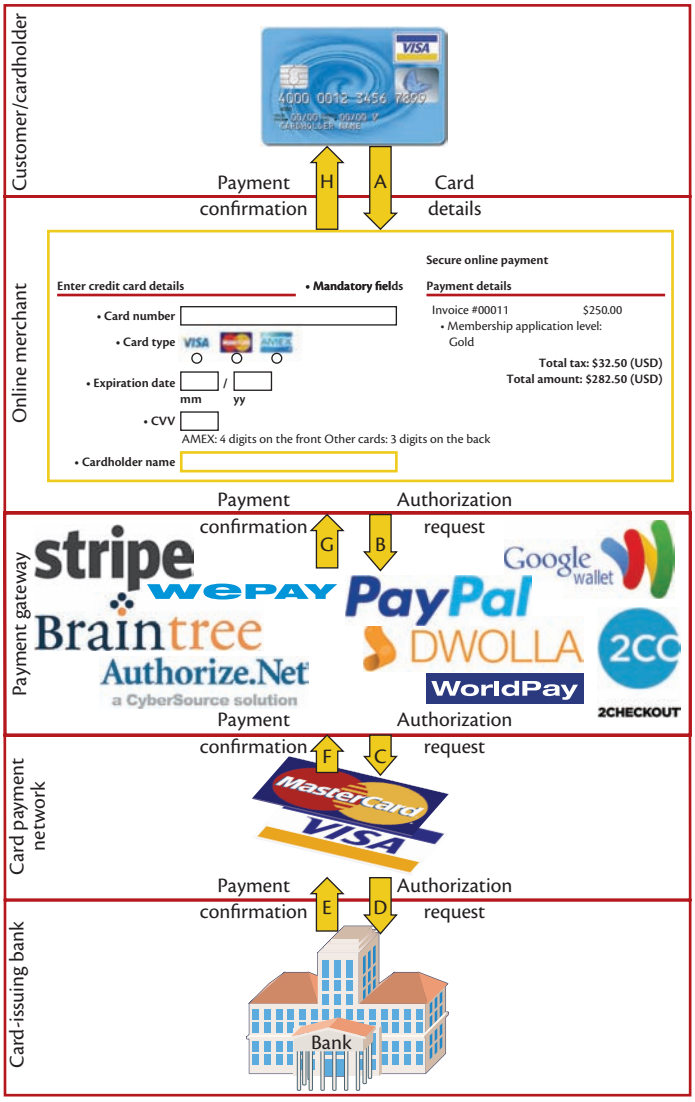

Figure 1. Parties in online payment. The customer/cardholder enters payment card details (action A). The online merchant passes the card details to a chosen payment gateway (action B). The payment gateway then connects the merchant to the card payment network, which provides the link between the payment gateway and the card-issuing bank (actions C and D). The card issuer holds the customer's bank account and approves the payment (action E). Finally, the card-issuing bank deposits the customer's money to the merchant's bank account (actions F, G, and H).

Payment Card Data Fields

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

The industry's Payment Card Industry Data Security Standard (PCI DSS) provides a comprehensive set of rules and controls for the secure handling and storage of sensitive card data.[7] However, there's no requirement for the merchant to request all the data fields during an online payment authorization, nor is there a 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 an entered name is correct.
- Sixteen-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.
- CVV2: a three-digit number printed on the reverse side of the card. It's meant to be known only to the person possessing the card. It shouldn't be stored electronically anywhere in the payment ecosystem.[7]
- Cardholder address: not visible on the card but sometimes used for payment authorization 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]

Distributed Guessing Attack

To obtain card details, an adversary 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. The reason this attack works in practice is due to two weaknesses. Each isn't too severe on its own, but when used together, they present a serious risk to the global payment system.

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

Second, 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:

- two fields: PAN + expiry date (the absolute minimum),
- three fields: PAN + expiry date + CVV2, and
- four fields: PAN + expiry date + CVV2 + address.

Starting with a valid card number (PAN), to guess the expiry date, an attacker can use 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 three 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 guessing the three-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 might 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, 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 four 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 a cardholder's address for the countries that have a long postcode (more than three numerical digits) isn't as straightforward as getting the expiry date or CVV2 because the attacker will first 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 (www.bindb.com/structure.html) and ExactBins (www.exactbins.com/features), 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 has used near-field communication (NFC) skimming to obtain the PAN and expiry date. 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.

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's indeed possible and practical to obtain all information on the card. Included were seven Visa cards with a spread of PAN, expiry date, and CVV2 values. We selected 400 Alexa top-rated commercial websites for our investigation; these included many global websites such as iTunes, Google, PayPal, and Amazon.[10]

Software Tools

The software tools implemented for the experiments consisted of a website bot and automated scripts written in Java Selenium browser automation framework (docs.seleniumhq.org/docs). All the experiments were run on the Mozilla Firefox web browser. Figure 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.

Obtaining Card Data

The PAN is the starting point for the generation of all other card data fields. There are at least two known

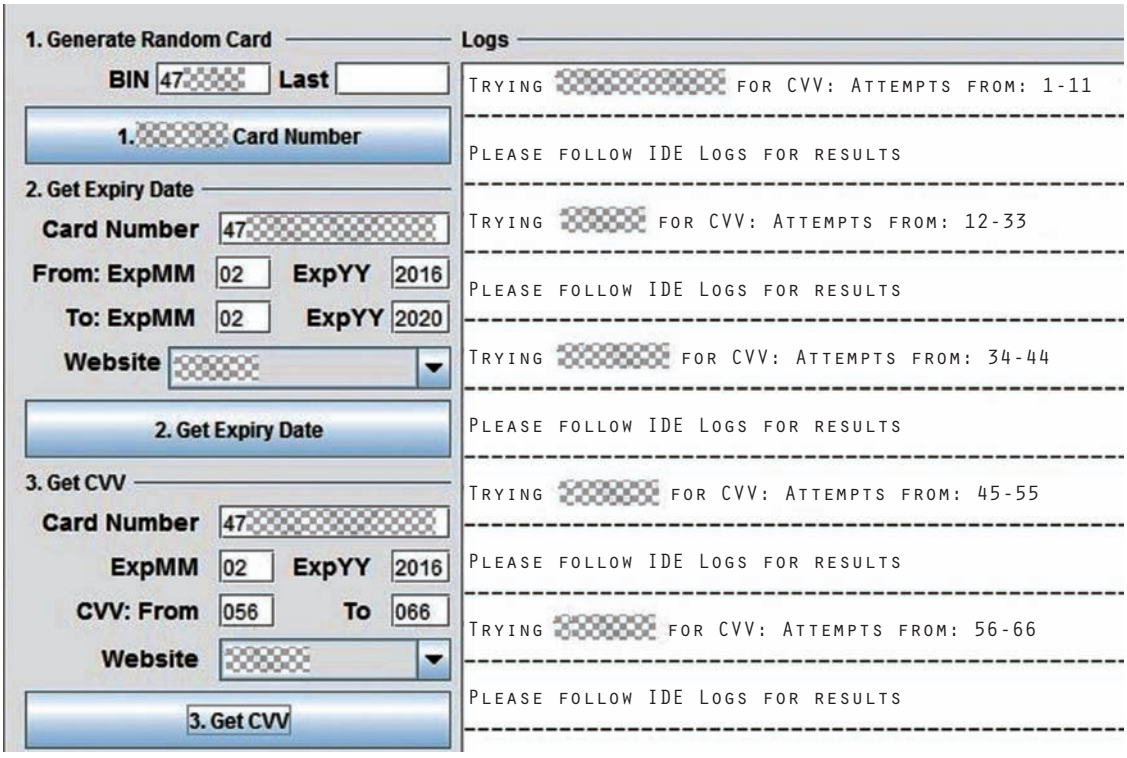

Figure 2. Screenshot of the website bot, farming card verification values (CVV2) from multiple sites. The bot cycles through the possible values for each field to find the correct information.

methods of obtaining valid PANs. Criminals sell bulk lists of card details online. These lists are considered less valuable when they don't 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 exploiting the contactless feature common in recently issued payment cards. NFC skimming provides an attacker with the PAN, the expiry date, and in some cases, the cardholder's name.[11] It's also possible to generate PAN using the first six digits of a PAN and the Luhn's algorithm, and getting it verified.[12] However, we didn't take this approach because it crosses the boundary of ethical research—we used only 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 isn't possible, the bot can be used to systematically guess the expiry date of a given PAN on the websites that don't 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, whereas most other payment sites allowed 5, 10, or even 50 attempts to enter a correct CVV2. In our scenario, we "farmed out" the brute-force guessing attack to tens or even hundreds of payment systems, which meant we could carry out practically unlimited guesses.

The final step generates the cardholder's address. An attacker can exploit the different variants of address verification systems to find the cardholder's full address.

Transferring Money

Once 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 card's owner, 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 and

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. Therefore, it's desirable from the attacker's point of view that the funds are transferred to an account outside the country (because it's 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 Western Union.

In our experiment, we used the card information extracted using our bot 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.

Results

Our results (detailed in Table 1) show that the distributed guessing attack we describe is indeed practical and so a credible threat. We studied and tested the payment websites of 389 of Alexa's most visited sites (we looked at the Alexa top-400 sites, but 11 of them didn't reveal sufficient useful information for our experiment). As shown in Table 1, 26 sites use only two fields for card payment, and an attacker would use these sites to guess the expiry date. Two hundred ninety-one sites use three fields, which an attacker can use for guessing the CVV2, and 25 sites use four fields, which lets an attacker guess the address postcode. Finally, of the 389 sites, 47 merchants (that is, 12 percent) had implemented 3D Secure payments (these sites are impervious to the distributed guessing attack).

There's 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 1, the number of sites that allows a 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.

In our experiments, we 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's 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 addresses (only postcode digits) to get numerical postcode digits. We extended our experiments and ran instances of our bot on another set of payment sites (which verify the door number and the postcode digits) to get the full address of all our Visa test cards.

These experiments have also shown that it's 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 authorized within two 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 four seconds.

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 categorized them into three groups: expiry date, CVV2, and postcode. Because 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 we found suitable contacts for each website, 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, which served as a means for these merchants to trace us back, so that they could verify our authenticity. This would also allow them to request more detailed and technical information about our experiments should they want 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 a machine-generated response mostly confirming the receipt of our notification. All the human responses requested more technical details, and some asked us to suggest solutions. Of the 36 websites we contacted, eight never responded. When web merchants requested more information, we offered them an initial draft of this article, which explained the

Table 1. Variation in payment security settings of online payment websites.

No. of attempts allowed	No. of sites with two fields (guess expiry date)	No. of sites with three fields (guess CVV2)	No. of sites with four fields (guess address postcode)	No. of 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

experiments and the attack to help them understand the actual problem. We followed the disclosure policy requested by the websites and anonymized the affected sites in our article.

As a result of our disclosure process, 8 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. Table 2 illustrates the nature of the patching of the notified websites. Of the eight websites that modified their approach (labeled A to H), four used two fields (labeled Expiry date in the Information leak column) and four used three fields (labeled CVV2).

In most cases, we learned about the patching behavior through manual observations, but in two cases (websites B and G), the affected websites notified us about the changes they made. Websites A and B patched their checkout system by adding an address verification field. However, this wasn't a good idea because it didn't provide additional security but instead opened up a new avenue for guessing, as we discuss later.

Typically, an online payment request is authorized almost instantly (within two seconds). From our observation, we noticed that websites C and D (both with expiry date leaks) had introduced additional delays to the payment authorization processing times. They did it in a staggered manner: few attempts were processed instantly, but after certain incorrect attempts took place, the time taken for payment confirmation 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 its 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 an IP-based velocity filter to limit the number of attempts to get CVV2 from 50 to 10 in 24 hours. Initially, websites G and 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 tradeoffs that payment websites need to consider when deciding which fields and filters to use, our ethical disclosure protocol didn't 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, which added CAPTCHA. CAPTCHAs prevent automated attempts in getting the sensitive card information but might adversely affect the usability of those websites.[13] To help websites G and H 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 2 for these websites).

The overall result of our study on the nature of patching on the notified websites revealed that the vast majority (78 percent) didn't make a change. We don't 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 was either to add a filter to make it more cumbersome to try many times (six of eight sites that patched added delay or velocity filters) or to add a field (websites A and 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 behaviors isn't so obvious. As we already pointed out, the sensible measure of limiting the number of attempts won't stop guessing attacks if it's not done on all websites. Furthermore, adding a card validation field might be a reasonable idea for a site for various reasons but might inadvertently weaken the protection against

Table 2. Nature of patching on the notified websites.

Website	Information leak	Patching behavior				
		Adding address field	Adding delay filter	Adding velocity filter (PAN* based)	Adding velocity filter (IP based)	Adding CAPTCHA
A	Expiry date	x				
B	Expiry date	x				
C	Expiry date		x			
D	Expiry date		x			
E	CVV2			x		
F	CVV2				x	
G	CVV2				x	x
H	CVV2				x	x

*PAN is primary account number.

guessing attacks of the payment system as a whole. After all, the added field might be a welcome opportunity to attempt guesses on this additional card detail.

Solving the Problem: The Challenges

Improving online payment system security is a complicated challenge for various reasons. One could argue that payment card security mechanisms are bound to remain unsatisfactory because they haven't been designed for operation over the distributed Internet. Solutions such as 3D Secure can be seen as afterthoughts and 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 Figure 1.

Customer/Cardholder

Because the distributed guessing attack described in this article uses merchant websites and card payment networks to get all the card details, there isn't much a cardholder can do to prevent it. At the same time, the cardholder is severely impacted by the attack: money might be lost and cards might have to be blocked; 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 organized as a group, or have representatives in various bodies, to put pressure on the other stakeholders. As individuals, cardholders could "vote with their feet" and select cards from card payment networks that aren't exposed to the distributed guessing attack. At the moment, the payment system is too complex and nontransparent to expect customers to be able to make such choices.

Online Merchant

On its 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 couldn't be staged.

At the same time, a merchant can avoid being exploited in the attack either by using only cards that use a payment network that isn't vulnerable from the attack or by using 3D Secure technologies recommended by the payment card industries,[7] such as American Express's SafeKey,[3] Verified by Visa,[4] and MasterCard's SecureCode.[5] If 3D Secure is implemented, the card-issuing bank is responsible for authenticating a cardholder before authorizing 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 transactions' frequency or value appears to be unusual. Our experiments confirmed that 3D Secure payments are protected from the distributed guessing attack described in this article because the issuing bank can see all transaction requests directed at a single card, even if those requests are distributed across many websites.

From the merchant's perspective, 3D Secure has several drawbacks; only 47 merchants in the Alexa top-400 sites have elected to implement 3D Secure. First, the proportion of the customers who don't complete the transaction can be high when the customer

encounters the 3D Secure login screen: up to 43 percent in the US and 55 percent in China.[14] Second, there are additional costs associated with implementing 3D Secure.

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.

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 utilize. One can't expect all 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 important, gateways might 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 be circumvented easily by switching to a website that uses a different payment gateway.

Card Payment Network

Responsibility for authorizing online payment requests ultimately resides with the bank that issued the credit or debit card. However, our experiments have shown that the distributed guessing attack described in the article works only 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 if the network is globally integrated.[15]

The most obvious defense against the distributed guessing attack would be at the level of the card payment network. However, we aren't 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.

Card-Issuing Bank

The bank comes into play at the final stage of the payment process to approve the transfer of funds, but it wouldn't 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 acquire card information. Many issuing banks are now running intelligent fraud detection systems that identify transactions outside their customer's normal spending habits.[6] The issuing bank then has the option to block the payment, ask the customer for confirmation, or accept the payment and take a calculated risk that a transaction might 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.

Fundamentally, much of the problem with card payment stems from the fact that the payer's identity needs to be established in the card-not-present mode. This is inherently problematic because it's 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 isn't available to establish the payer's identity. This is exacerbated by the fact that the Internet facilitates distribution of guesses for data fields over many merchant sites.

To prevent this attack, either standardization or centralization can be pursued (some card payment networks already provide this). Standardization would imply that all merchants need to offer the same payment interface, that is, the same number of fields. Then the attack doesn't scale. Centralization can be achieved by payment gateways or card payment networks possessing a full view over all payment attempts associated with its network. Neither standardization nor centralization naturally fits the flexibility and freedom of choice one associates with the Internet or successful commercial activity, but the two will provide the required protection. It's up to the various stakeholders to determine the case for and timing of such solutions. ■

Acknowledgments

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 and the UK EPSRC EP/L002213 Academic Centre of Excellence in Cyber Security Research.

References

1. "Retail Sales, February 2015," UK Office for Nat'l Statistics, 26 Mar. 2015; www.ons.gov.uk/ons/dcp171778_399119.pdf.
2. "Fraud the Facts 2015: The Definitive Overview of Payment Industry Fraud and Measures to Prevent It," Financial Fraud Action UK, London: UK Cards Association, 2015;

www.financialfraudaction.org.uk/download.asp?file=2979.
3. "Product Capability Guide," American Express SafeKey, 2014; network.americanexpress.com/en/globalnetwork/Images/SafeKeyProductCapabilityGuide_2014.pdf.
4. "3D Secure System Overview," Visa, 2001; www.visanet.com.pe/verified/demovisanet-web/resources/3DS_70015-01_System_Overview_external_v1.0.3.pdf.
5. "MasterCard Secure Code, Merchant Implementation Guide," MasterCard, 2014; www.mastercard.us/content/dam/mccom/en-us/documents/SMI_Manual.pdf.
6. "Card Acceptance Guidelines for Visa Merchants," Visa, 2014; usa.visa.com/download/merchants/card-acceptance-guidelines-for-visa-merchants.pdf.
7. "PCI DSS Applicability in an EMV Environment," Payment Card Industry, 2010; www.pcisecuritystandards.org/documents/pci_dss_emv.pdf.
8. "Gateway Developer Guide and Reference," PayPal, 2014; www.paypalobjects.com/webstatic/en_US/developer/docs/pdf/payflowgateway_guide.pdf.
9. "Transaction Processing Rules," MasterCard, 2014; www.mastercard.com/us/merchant/pdf/TPR-Entire_Manual_public.pdf.
10. "The Top 500 Sites on the Web—Shopping," Alexa, 2015; www.alexa.com/topsites/category/Top/Shopping.
11. M. Emms et al., "Harvesting High-Value Foreign Currency Transactions from EMV Contactless Credit Cards without the PIN," *Proc. ACM SIGSAC Conf. Computer and Communications Security* (CCS 14), 2014, pp. 716–726.
12. "Validating a Credit Card Number Using Luhn's Algorithm," Symantec, 2015; support.symantec.com/en_US/article.TECH221769.html.
13. A. El Ahmad, J. Yan, and W. Ng, "CAPTCHA Design: Color, Usability, and Security," *IEEE Internet Computing*, vol. 16, no. 2, 2012, pp. 44–51.
14. "Analysis Reveals Worldwide Impact of 3D Secure on Transaction Conversion Rates," Adyen, 2014; www.adyen.com/home/about-adyen/press-releases/2014/3d-secure-worldwide-impact-conversion.
15. "The MasterCard Network Advantage," MasterCard, 2011; newsroom.mastercard.com/wp-content/uploads/2011/09/MasterCard-Network-Advantage.pdf.

Mohammed Aamir Ali is a PhD research student in the Centre for Cybercrime and Computer Security at Newcastle University. His research centers on exploiting the potential vulnerabilities in the payment ecosystem (especially near-field communication mobile payments, payment applications, and online payments). Contact him at m.a.ali2@ncl.ac.uk.

Budi Arief is a lecturer in the School of Computing at the University of Kent. His research interests include cybercrime, the security and dependability of computer-based systems, cybersecurity education, and the Internet of Things (IoT), with a strong overarching element of interdisciplinary research. Arief received a PhD in computing science from Newcastle University. Contact him at b.arief@kent.ac.uk.

Martin Emms is a cybersecurity researcher at Newcastle University. His research interests include the security impacts of new payment technologies, user authentication using ubiquitous and IoT technologies, and related security issues caused by unintentional privacy leakage. Emms' work is enhanced by his industry experience as a solutions architect in payments and as a systems designer of safety-critical embedded control systems for satellite communications, nuclear power, and transport. Contact him at martin.emms@ncl.ac.uk.

Aad van Moorsel is a professor and Head of School in the School of Computing Science at Newcastle University. His research interests include cybersecurity and cybercrime, with an emphasis on decision making, quantification, and risk management. van Moorsel received a PhD in computer science from the University of Twente. Contact him at aad.vanmoorsel@ncl.ac.uk.