TRUSTWAVE GLOBAL SECURITY REPORT

Trustwave®

Trustwave®

INTRODUCING THE 2016 TRUSTWAVE GLOBAL SECURITY REPORT

Cybercrime is big business. We hear it so often that the words threaten to lose their impact. But the details still have the capacity to shock. In last year's report, we demonstrated how attackers launching a malware infection campaign could expect to earn a breathtaking $84,100 (USD) in profit from an initial investment of just $5,900—an ROI of 1,425 percent—in just 30 days. In parts of the world where many attacks originate, the prospect of that kind of money can mean a trip out of poverty to a life of riches and glamour.

Small wonder, then, that investigating cybercrime reveals something that looks very much like… a business. The biggest cybercrime operations are essentially computer software and services companies, albeit illicit ones. Developers create tools that they sell or rent to customers through online black markets, complete with sales, money back guarantees, and reputation systems to provide customers with assurances that they won't get ripped off. As enterprise software vendors have increasingly moved to the cloud, so too have malware vendors. Where once prospective cybercriminals bought exploit kits as packaged software, today they pay for access to a central server administered by the exploit kit maker, who keeps it stocked with the freshest exploits and all the tools one needs to exploit thousands of unsuspecting computers. Malware-as-a-service.

Understanding the motivations and resources of professional cybercriminals is key to defending against them. In that spirit, we present the 2016 Trustwave Global Security Report. In these pages, we have collected and organized statistics and analysis that Trustwave researchers have gathered from around the world, from breach investigations, incident reports, vulnerability research, and telemetry from Trustwave products and services. It provides information about data compromise incidents, vulnerabilities and exploits, attacks on web platforms, threats delivered through the web and email, and a range of other important and timely security topics. For an extra look at some of the internals of the cybercrime business, be sure to check out our special sections on malvertising and on attacks targeting the software supply chain.

If cybercrime is a business, you can consider this report your guide to its business plan. Use it to learn more about what the criminals are doing now, what they may do in the future, and the steps you can take to keep them away.

> "It's not personal, Sonny. It's strictly business."
>
> **—Michael Corleone, The Godfather**

EXECUTIVE SUMMARY

Data Compromise

- Hundreds of data compromise investigations conducted by Trustwave across 17 countries
  - **45%** of incidents were in North America
  - **27%** were in the Asia-Pacific region
  - **15%** were in Europe, the Middle East and Africa
  - **13%** were in Latin America and the Caribbean
  - **23%** of investigations were in the retail industry, 14% were in the hospitality industry, and 10% were in the food & beverage industry
- **40%** of investigations were of corporate and internal network breaches, and 38% were of e-commerce breaches
- **85%** of compromised e-commerce systems used the Magento open-source platform
- **60%** of breaches targeted payment card data
  - **31%** targeted card track (magnetic stripe) data recorded at point-of-sale (POS) terminals
  - **29%** targeted card data from e-commerce environments
- **41%** of breaches were detected by victims themselves
  - This is up from 19% in 2014
  - Median time between intrusion and detection was 15 days for internally detected breaches, compared to 168 days for breaches detected and reported by external parties
  - For breaches requiring containment efforts after detection, the median time between detection and containment was one day for internally detected breaches, compared to 28 days for externally detected breaches

Vulnerabilities and Exploitation

- **21:** The number of high-profile zero-day vulnerabilities Trustwave saw exploited in the wild in 2015
- Adobe Flash Player was the most heavily targeted software component of the year
  - **38%** of zero-day vulnerabilities targeted Flash Player
  - **86%** of vulnerabilities added to exploit kits targeted Flash Player
- **40:** The percentage share of exploit kit traffic detected by Trustwave attributed to Angler, the most commonly used exploit kit of the year
  - **4**: The number of zero-day Flash exploits first used by Angler
- **90:** Estimated percentage of traffic to the RIG exploit kit that originated from malicious advertisements
- **71:** The percentage of web attacks observed by Trustwave that targeted WordPress

Email Threats

- **54:** The percentage of all inbound email that was spam
- **34:** The percentage point drop in spam advertising pharmaceutical products, from 73% of total spam in 2014 to 39% in 2015
- **5:** The percentage of spam observed by Trustwave that included a malicious attachment or link

Malware

- **42%** of the malware observed by Trustwave used obfuscation
- **33%** of the malware observed by Trustwave used encryption

Application and Network Security

- **97%** of applications tested by Trustwave had one or more security vulnerabilities
  - **14:** The median number of vulnerabilities per application discovered by managed Trustwave application scanning services
  - The most common vulnerability types included session management vulnerabilities, information leakage vulnerabilities, and cross-site scripting vulnerabilities
  - **10%** of vulnerabilities discovered were rated critical or high risk
- Vulnerable SSL and TLS installations were the most common class of vulnerabilities detected by Trustwave network scanners

DATA SOURCES

Enhanced by our applied research and experiences from the field, Trustwave's large, global client base offers us unmatched visibility into security threats. We gain key insights from our analysis of hundreds of data breach investigations, threat intelligence from our global security operations centers, telemetry from security technologies and industry-leading security research.

For example, in 2015 we:

- Investigated compromised locations in **17** countries
- Logged **billions** of security and compliance events each day across our seven security operations centers (SOCs)
- Examined data from more than **tens of millions** of network vulnerability scans
- Accumulated results from **thousands** of web application security scans
- Analyzed **tens of millions** of web transactions for malicious activity
- Evaluated **tens of billions** of email messages
- Blocked **millions** of malicious websites
- Conducted **thousands** of penetration tests across databases, networks and applications

SECTION 1

DATA COMPROMISE

After 2014, when a spate of high-profile data breaches and vulnerabilities alerted the general public to the challenges of data security, we were keenly interested to see what attackers and data thieves had in store for 2015. The answer, it seems, was specialization. As the increasing adoption of EMV ("chip-and-PIN") payment card technology finally began to reduce the attack surface for point-of-sale (POS) systems, criminals shifted their focus slightly from broad-based attacks on retail to a tighter focus on specific industries and platforms.

In particular, 2015 was not a great year for the hospitality industry from a security perspective, with major breaches affecting many of the world's most prominent hotel chains. Significantly, in most of the cases Trustwave investigated, attackers were able to gain access to hospitality systems using well-known remote access applications. Of greater concern was the fact that often the compromised credentials were owned by third parties, such as software vendors.

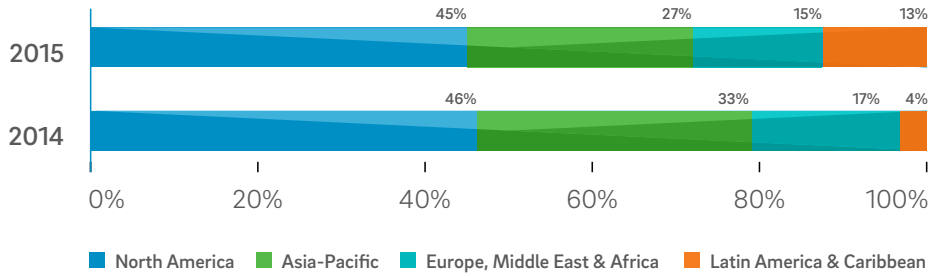
In the e-commerce space, the Magento open-source e-commerce platform was the target of choice for attackers, with Magento installations accounting for 85 percent of compromised systems. Considering that at least five critical Magento vulnerabilities were identified in 2015, this focus is not surprising. Depressingly – but predictably – most of the affected systems were not fully up to date with security patches, with some being behind by more than 12 months. With so much attention being paid to zero-day vulnerabilities, it's vital to remember that it doesn't matter how fast a vendor releases a patch if the patch is never applied.

In this section we discuss our findings from, and analysis of, Trustwave investigations of security compromises and data breaches affecting enterprise environments in 2015. While these statistics are highly dependent on the details of each investigation, we find that they provide an interesting picture of where and how attackers concentrated their efforts last year, and give some useful clues as to what the future might hold.

“The retail industry once again accounted for the largest single share of incidents investigated by Trustwave, at 23 percent of the total.”

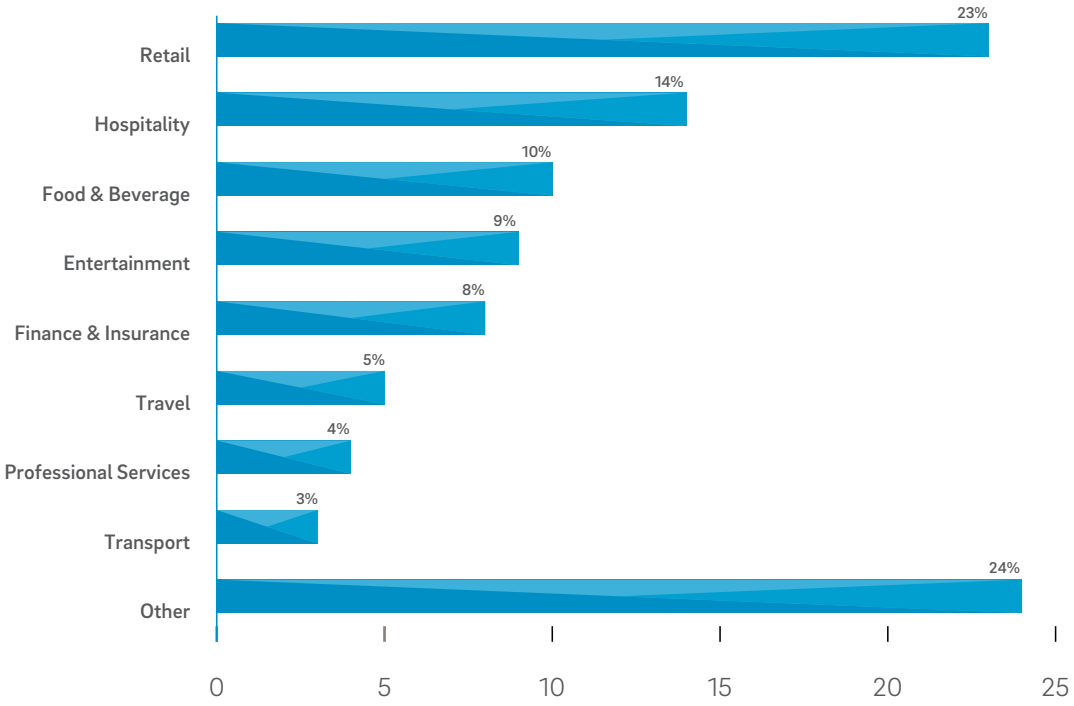
COMPROMISE DEMOGRAPHICS

COMPROMISES BY REGION



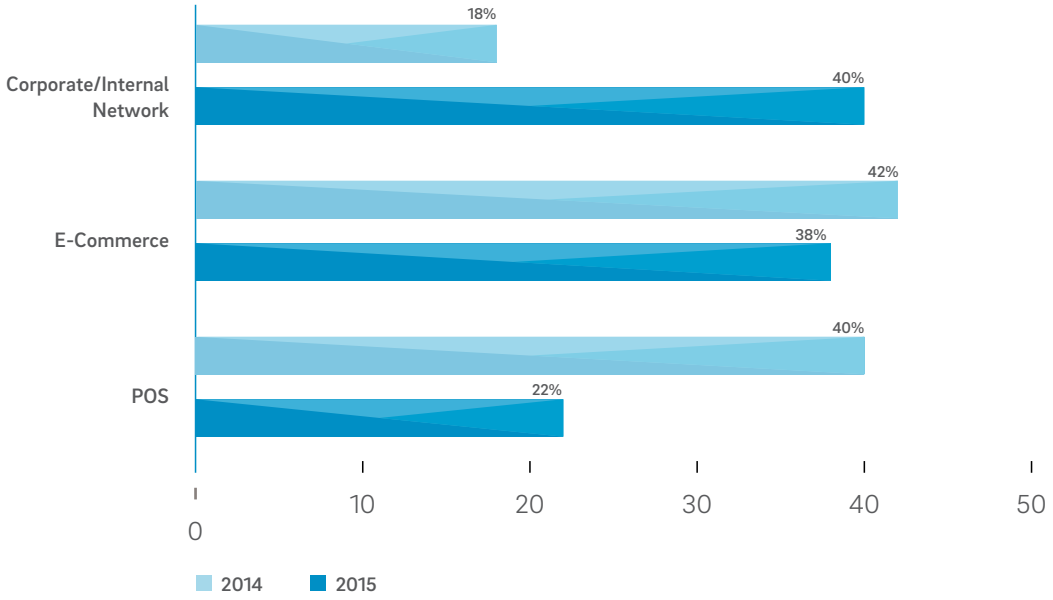
The observations in this section are derived from several hundred investigations involving malicious data breaches that the SpiderLabs team at Trustwave conducted across 17 countries in 2015. As in previous years, nearly half of the incidents took place in North America, with the rest distributed between the Asia-Pacific region (APAC); Europe, the Middle East, and Africa (EMEA); and Latin America and the Caribbean. Latin America, in particular, accounted for a significantly higher share of investigations than in 2014, an indicator of both the region's growing economic power and of how that power has made it a larger target for criminals.

COMPROMISES BY INDUSTRY



The retail industry once again accounted for the largest single share of incidents investigated by Trustwave, at 23 percent of the total, although that number fell from 43 percent in last year's report. The hospitality industry, which was targeted by a significant attack campaign that used point-of-sale (POS) malware to collect payment card data from travelers, accounted for the second largest share of incidents at 14 percent, followed by food and beverage (10 percent), entertainment (9 percent), and finance and insurance (8 percent).

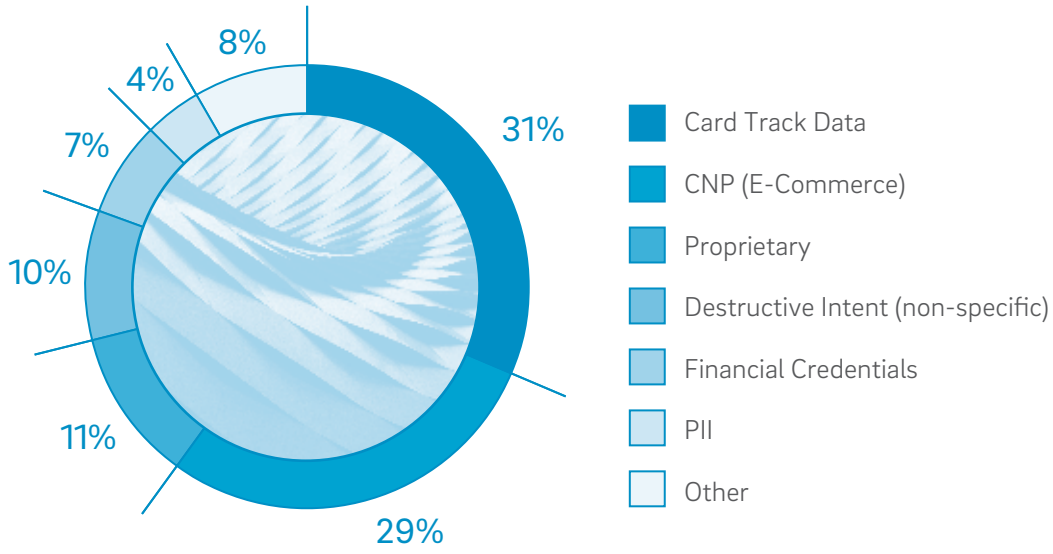
COMPROMISES BY ENVIRONMENT: 2014-2015



As the share of incidents involving the retail industry declined from 2014, so did the share of incidents affecting POS environments, which accounted for 22 percent of 2015 incidents, down from 40 percent in the previous year. The share affecting corporate and internal networks correspondingly increased to 40 percent, up from 18 percent in 2014. Incidents affecting e-commerce environments accounted for 42 percent of incidents in 2015, on par with the previous year.

“Compromises affecting corporate and internal networks increased to 40 percent in 2015, up from 18 percent in 2014.”

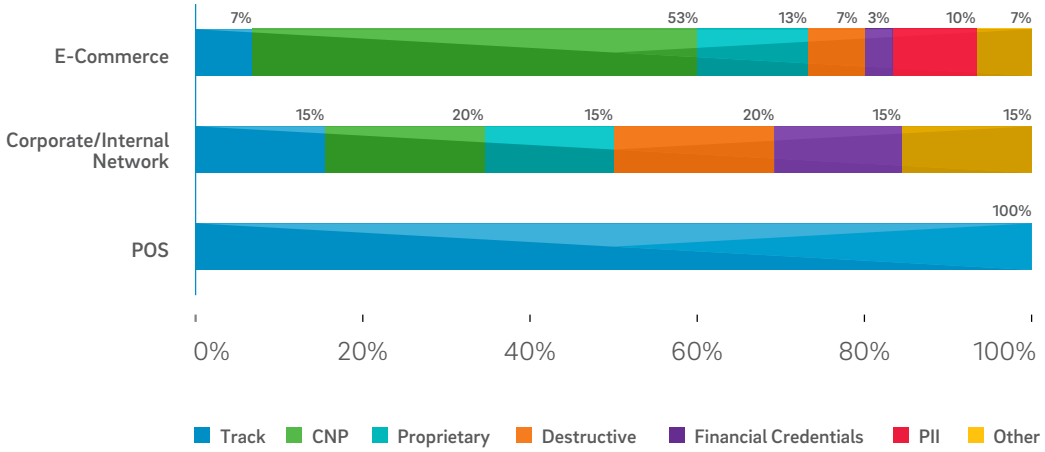
PRIMARY TYPES OF DATA TARGETED



In the majority of incidents, attackers were after payment card data, split about evenly between card track (magnetic stripe) data (31 percent of incidents), which came mainly from POS environments, and card-not-present (CNP) data (29 percent), which mostly came from e-commerce transactions. In 10 percent of cases examined, the attackers simply sought to destroy or damage information, rather than to collect it. Other attackers sought proprietary information (11 percent), financial credentials (7 percent), and personally identifiably information (PII) (4 percent). In some cases, multiple types of data were exposed and targeted, meaning that the exposure of any one type of data does not reflect the totality of the breach. For this particular statistic, we've reported the primary data type targeted.

COMPROMISES BY ENVIRONMENT

TYPES OF DATA COMPROMISED BY ENVIRONMENT



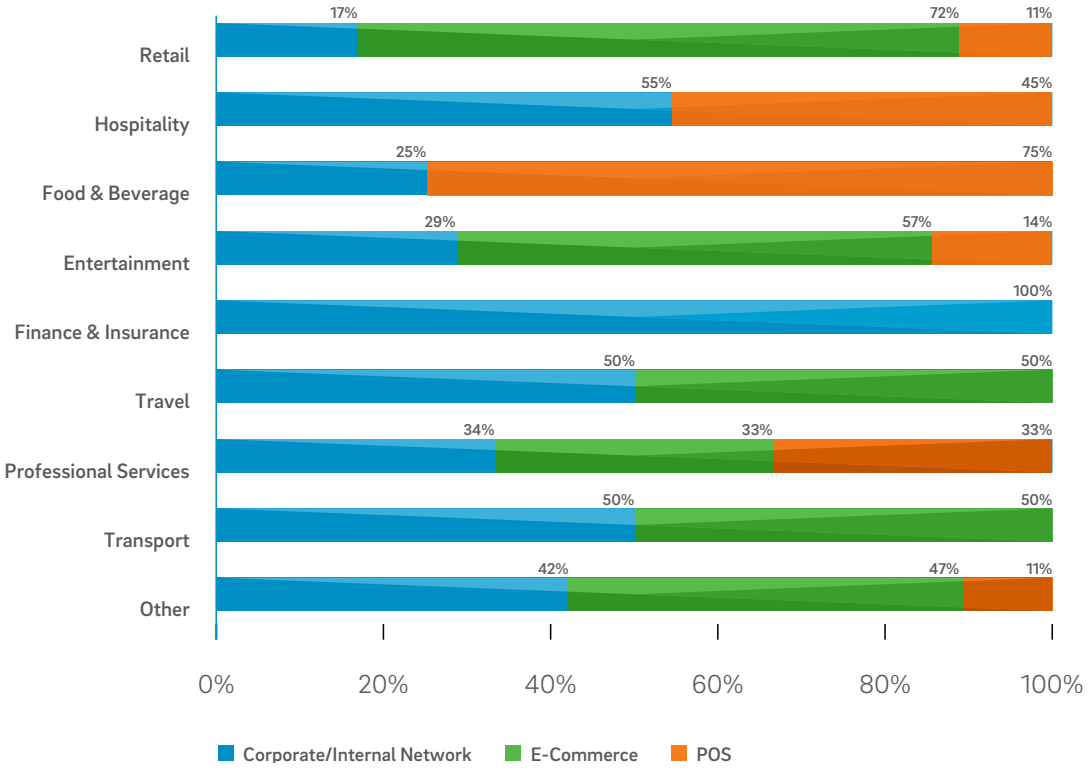
We classify the IT environments in which breaches take place into the following categories:

- 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 smart card readers. Most run versions of the Windows Embedded or Linux operating systems customized for POS devices, and they are usually networked to transmit card and sale data to a centralized location and/or a financial institution.
- E-commerce environments include web server infrastructures dedicated to websites that process payment information and/or personally identifiable information (PII).
- Corporate and internal network environments comprise enterprise networks in general, and can include sensitive data that was originally collected in a POS or e-commerce environment.

In keeping with their limited and specialized function, all of the incidents affecting POS environments targeted track data, the information encoded on a payment card’s magnetic stripe (but not on the smart cards used in chip-and-PIN transactions, which are significantly more secure). Most of the incidents affecting e-commerce environments targeted CNP data, while incidents involving corporate and internal networks targeted a range of different data types.

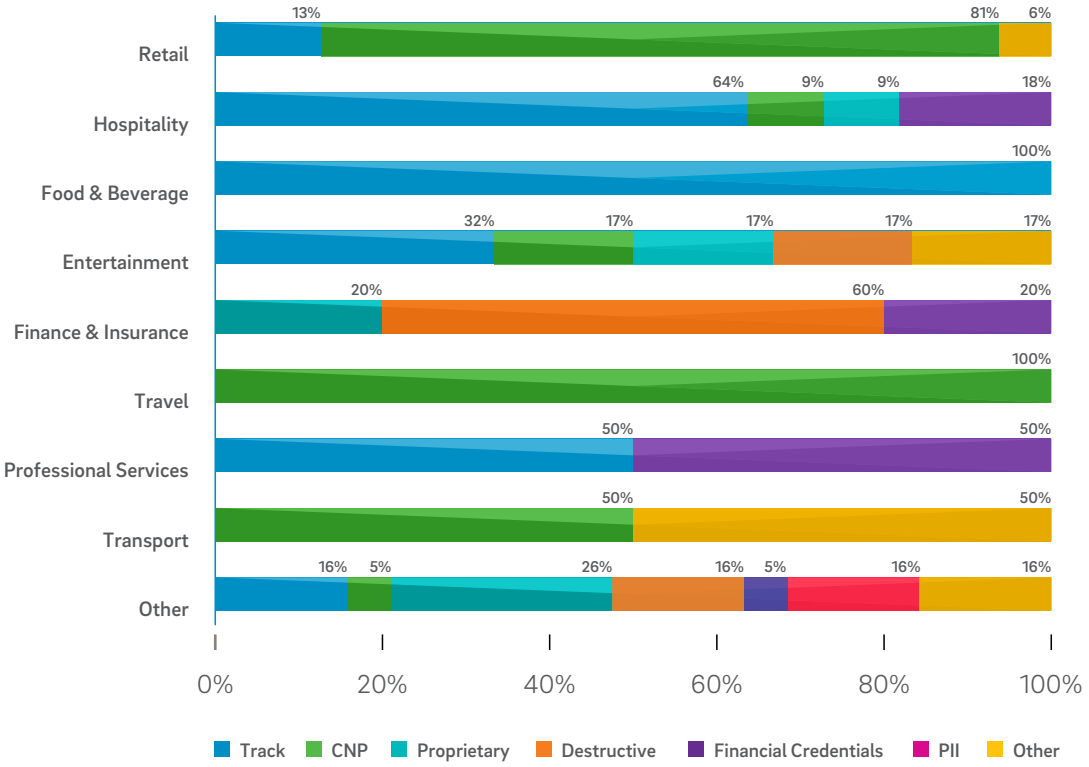
COMPROMISES BY INDUSTRY

IT ENVIRONMENTS COMPROMISED BY INDUSTRY



Unsurprisingly, the types of environments targeted were highly dependent on industry vertical. The industry with the highest share of POS incidents was the food and beverage industry—restaurants, grocery stores, and similar establishments—which conducts nearly all of its business through payment terminals. The retail industry, which includes e-commerce sites as well as brick-and-mortar stores, had the largest share of incidents that affected e-commerce assets. And all of the breaches we investigated that affected the finance industry, which does little retail-related business, involved corporate and internal networks.

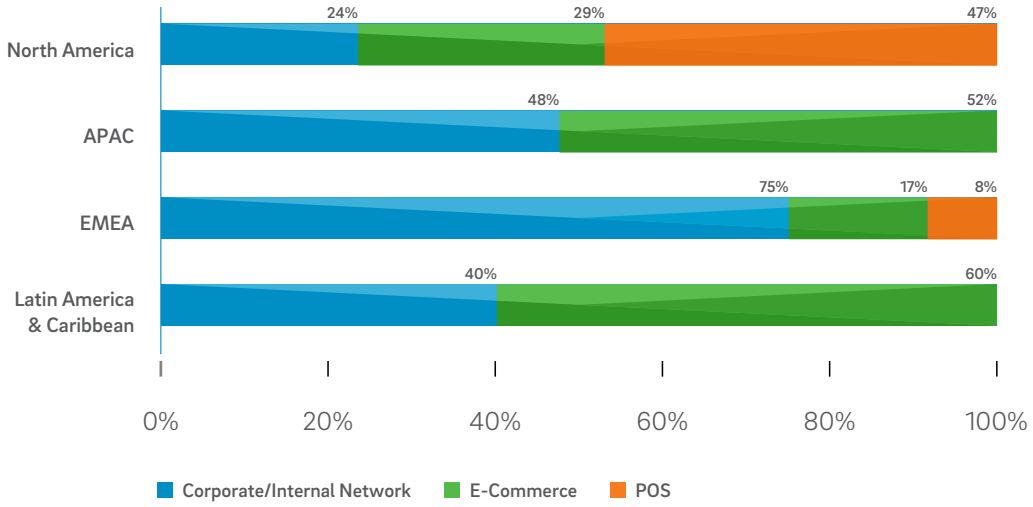
TYPES OF DATA COMPROMISED BY INDUSTRY



Likewise, the types of data targeted depended heavily on from whom the attackers were stealing. Compromises involving the travel industry—including airlines, travel agencies, and travel websites—and the retail industry were skewed heavily in favor of data from CNP transactions, while attackers targeting the hospitality and food and beverage industries mostly sought card track data. Most of the compromises affecting the finance and insurance industry were destructive in nature.

COMPROMISES BY REGION

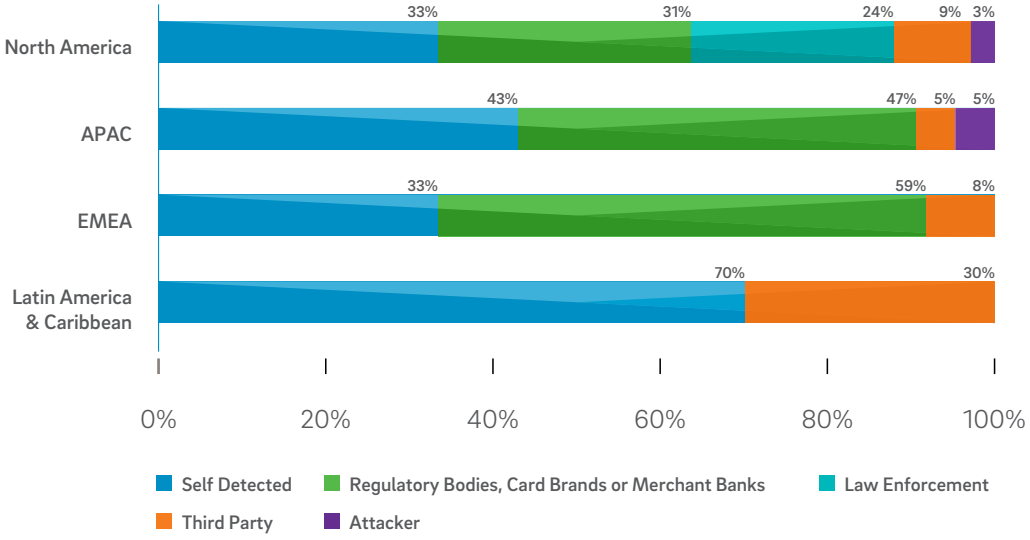
ENVIRONMENTS COMPROMISED BY REGION



The United States lags well behind most of the rest of the world in adopting the EMV payment card standard (often called chip-and-PIN) and consequently North America was the only region with significant POS environment compromises, all of which involved the older magnetic-stripe technology. In 2015, card issuers mailed millions of new EMV-compatible cards to U.S. cardholders, and an industry-imposed October 2015 deadline for businesses to install EMV-compatible equipment or assume liability for card fraud themselves saw retailers begin to upgrade their POS infrastructures. EMV adoption in the United States still has a long way to go, but we expect to see fewer POS incidents in North America in 2016 and thereafter.

"EMV adoption in the United States still has a long way to go, but we expect to see fewer POS incidents in North America in 2016 and thereafter."

METHOD OF DETECTION BY REGION



In the Latin America and Caribbean region, the share of self-detected compromises was twice that of the rest of the world, and none of the breaches that we investigated there were detected by card processors or regulatory bodies. While self-detection is usually the best and most reliable way to detect a breach, the high rate of self-detection in Latin America and the Caribbean may have as least as much to do with a relative lack of detection abilities on the part of the regional financial and regulatory infrastructures as with the security postures of the companies themselves. As the economic power of Latin America has grown, the financial security and regulatory apparatus has not always kept up, although we expect the gap with the rest of the world to close as the financial industry in that region focuses more on banking security.

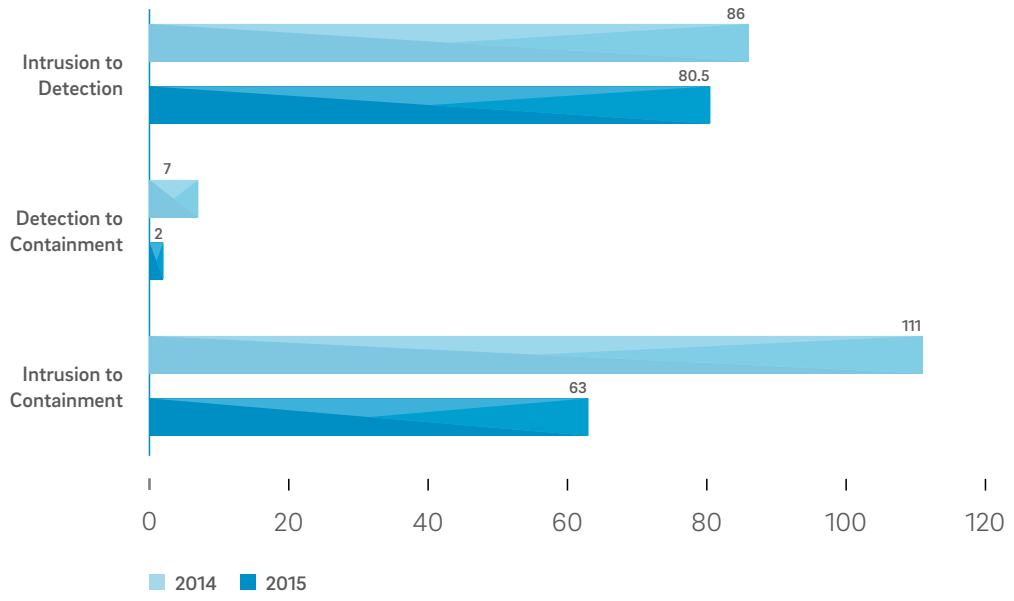
COMPROMISE DURATION

To understand how long it takes businesses to detect a breach and how long affected data records are exposed, Trustwave investigators record the dates of three milestones in a compromise's duration: initial intrusion, detection, and containment (wherever possible).

- **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 is the day the victim or another party identifies that a breach has taken place.
- **Containment:** The date of containment is the day the compromise has been cleaned, and records no longer remain exposed.

In some cases, the date of containment can occur before the date of detection, as when an attack is halted by a software upgrade before being discovered, or when investigators determine that the attacker left the network before evidence of the breach was detected.

MEDIAN DAYS BETWEEN COMPROMISE MILESTONES

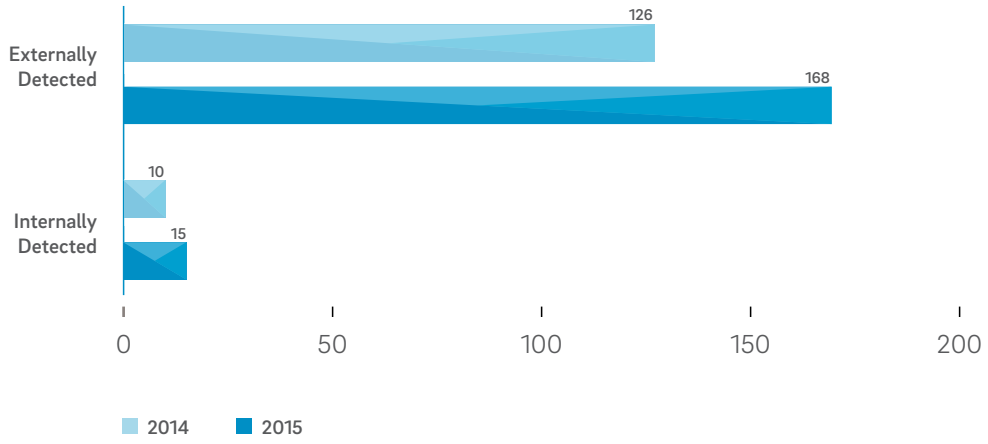


Durations varied greatly in the incidents we investigated. The median number of days from the first intrusion to detection of the compromise decreased from 86 days in 2014 to 80.5 days in 2015, with values ranging from zero days to 2,000 days (more than five years).

Once intrusions were detected, they were usually contained quickly. The median number of days from detection to containment decreased from seven days in 2014 to two days in 2015, with values ranging from –245 days (the intrusion ended 245 days before evidence was detected) to an outlier of 274 days in one exceptional case.

The median total duration between intrusion and containment decreased significantly to 63 days in 2015, down from 111 days in 2014. Interestingly, this figure is actually lower than the median duration between intrusion and detection, due to the number of cases in which the intrusion was contained before being detected.

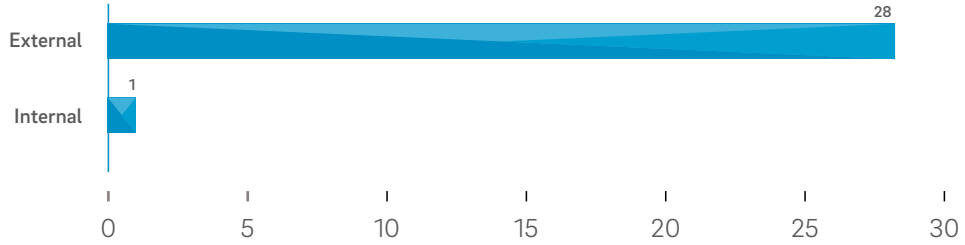
INTRUSION TO DETECTION IN DAYS



The longer a data compromise lasts, the more harm the attacker can do, and the more costly the breach can be. Our 2015 findings continue to support an assertion we've made for several years: Victims who are capable of detecting compromises internally, either on their own or in partnership with a managed security services provider, can detect breaches sooner and contain them more quickly than victims who are not.

When victims have the capability to detect compromises internally, they generally do so quickly: The median duration between intrusion and detection was just 15 days, up from 10 in 2014. In cases where the victims did not learn of the breach before being notified of it by regulatory bodies, law enforcement, or other third parties, the duration was usually much longer. The median time between intrusion and detection for externally detected compromises was 168 days in 2015, up from 126 in 2014.

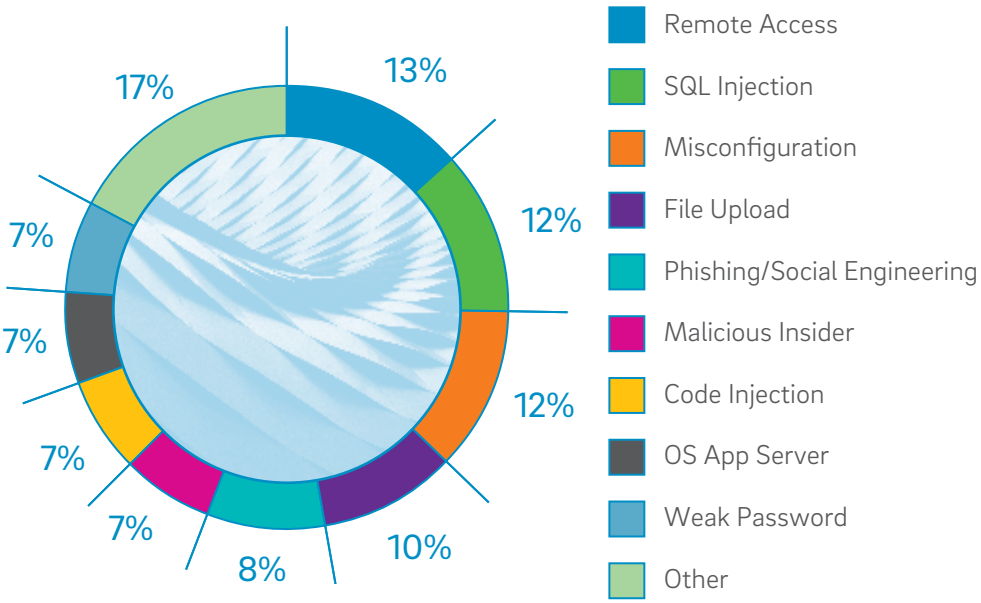
DETECTION TO CONTAINMENT IN DAYS
(excluding incidents in which containment preceded detection)



Even more interesting, perhaps, is the fact that internally detected compromises were also contained more quickly than externally detected ones. In cases where containment occurred after detection, the median duration between the two milestones was just one day for internally detected breaches, compared to 28 days for externally detected breaches. The same tools and techniques that enable businesses to detect breaches on their own often make it possible to respond to them quickly and easily—in fact, half of the internally detected compromises we investigated were contained the same day they were discovered. By contrast, a business that must be informed of a breach by an outside party is often not in a position to contain it quickly, and the compromise continues, sometimes for several crucial days.

METHODS OF INTRUSION

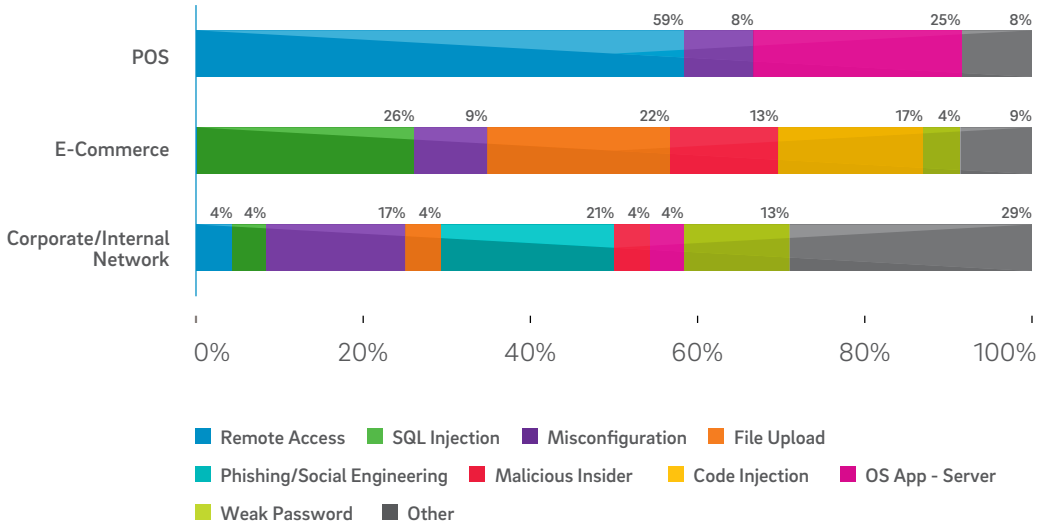
FACTORS CONTRIBUTING TO COMPROMISE



Insecure remote access software and policies, at 13 percent, contributed to the largest share of compromises we investigated in 2015, followed by SQL injection and general misconfiguration issues, at 12 percent each. Together with malicious file uploads (10 percent of incidents) and phishing and social engineering (8 percent), these factors accounted for more than half of the incidents investigated. The remainder included factors such as malicious insiders, weak passwords, code injection, and authentication bypass.

"Victims who are capable of detecting compromises internally, either on their own or in partnership with a managed security services provider, detect breaches sooner and contain them more quickly than victims who are not."

CONTRIBUTING FACTORS BY COMPROMISE TYPE

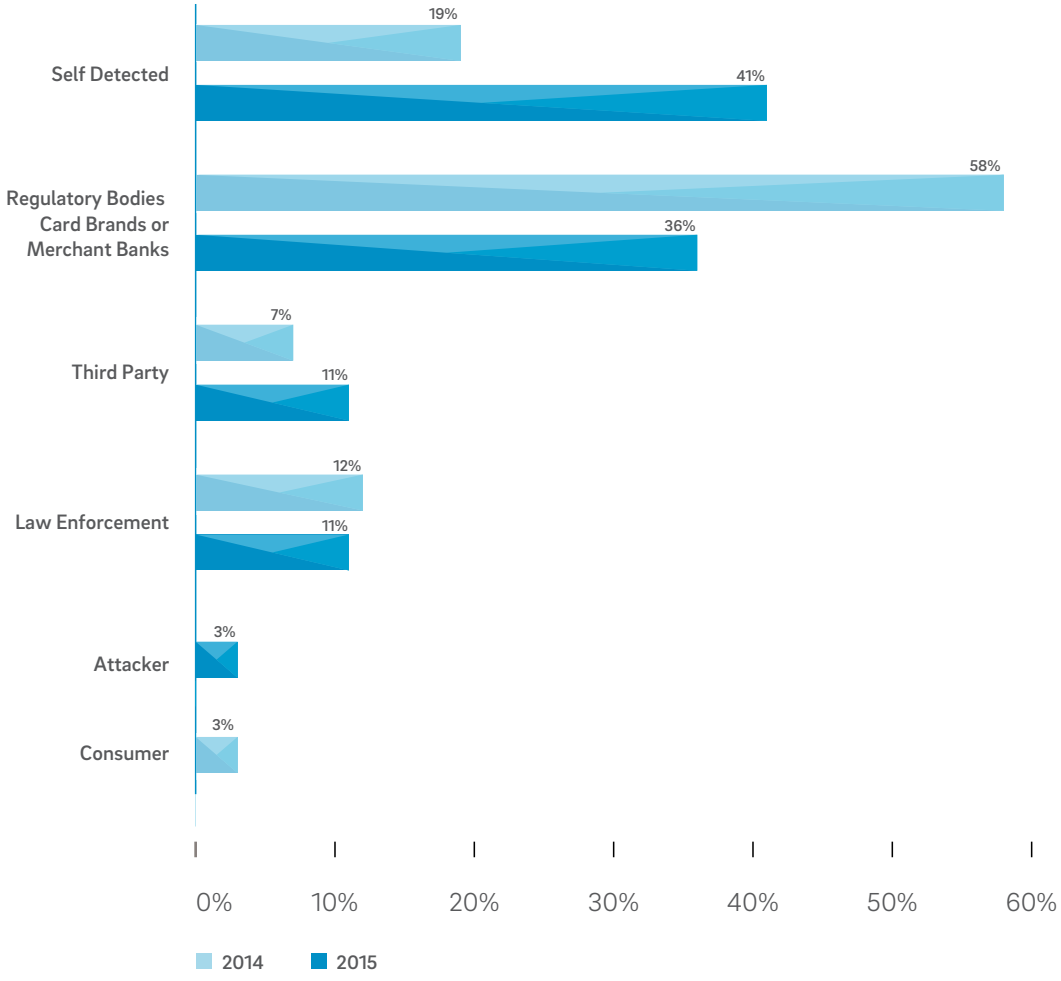


The majority of intrusions (59 percent) affecting POS environments involved malicious remote access—a significant hazard with networked POS devices. Misuse of applications that ship with the operating system, such as a web browser, contributed to the second-largest number of intrusions, with misconfiguration issues in third place.

For e-commerce environments, SQL injection was the biggest factor at 26 percent of intrusions, followed by malicious file uploads (22 percent), code injection (17 percent) and malicious insiders (13 percent).

Corporate and internal networks intrusions were facilitated by a wide range of factors, most prominently phishing (21 percent), misconfiguration (17 percent), and weak passwords (13 percent).

METHODS OF DETECTION



More than 40 percent of the compromises we investigated in 2015 were initially detected by the victims themselves, which is more than double the share of self-detected compromises in 2014. As noted earlier, self-detected compromises are typically identified, detected and contained much more quickly than compromises detected by outside parties, so we hope to see this trend continue in the future. Compromises detected by regulatory bodies, card brands and merchant banks accounted for 36 percent of incidents, with law enforcement agencies and other third parties accounting for 11 percent each. In an unusual and unwelcome development, a small percentage of the compromises in 2015 were first reported by the attackers themselves, often for blackmail purposes.

STOPPING DATA COMPROMISE NOW AND IN THE FUTURE

The cost and effort of securing a network against data compromise pales in comparison to the cost and effort of cleaning up after a breach. The following list includes security measures that Trustwave investigators recommend to customers to mitigate the risk they face from data compromise. These steps are based on the Payment Card Industry Data Security Standard (PCI DSS) for merchants that handle payment card data, but can be adapted for use by any organization that handles sensitive data.

Firewall Configuration

- Inbound and outbound access to and from the network should be properly restricted. Inbound access must be restricted to only those services (open ports) necessary to conduct business. Outbound traffic should be restricted to only trusted sites or IP addresses.
- Systems connected to a payment processing environment should not be allowed to “surf” the web.
- Systems that are not part of the payment processing environment or are required to conduct business should not be located within the same network segment.
- All firewalls should be audited for accessible ports and services.
- All firewalls should be hardware based and should provide stateful packet inspecting (SPI) capabilities.

Passwords

- All personal computers, servers, firewalls, routers, and other network devices should follow password complexity requirements. Passwords should be changed at least every 90 days.
- All passwords either stored or transmitted must be rendered unreadable using strong encryption.
- Each user should have their own unique account so that activities on a system can be tracked. Avoid using generic or default account names.
- When an employee leaves the company, change any passwords to which the employee had access.

System Configuration

- Ensure that system-hardening guidelines are in place to address known vulnerabilities and security threats. System configuration should be based on industry-standard best practices.
- For Windows environments, ensure the OS is configured to clear the pagefile.sys upon reboot.
- For Windows environments, ensure the OS is configured to have restore points disabled.
- Ensure no unauthorized modifications are made to systems in the environment (i.e. use of external storage, TrueCrypt volumes, unsupported software).
- Implement a strong change control process to track all changes made to systems in the environment.

Remote Access Solution

- Use two-factor authentication for all remote access into the environment. Two-factor authentication is normally defined as an authentication method requiring something a user knows (password) and something the user has (token, certificate).
- Third-party remote access must be an on-demand solution. It must be turned off by default and only enabled when needed.
- Enable auditing and logging for remote access into the environment.

Malware Removal

- If malware is, or was, suspected on a system, the system should be rebuilt to fully confirm the removal of the threat.
- Ensure anti-virus software is current on all systems and that it is set to update virus definitions. Also, ensure the virus definition license is valid and the software is properly accessing new definitions.

Logging and Monitoring

- Configure Windows event logs to capture security, application and system events on all systems.
- Ensure that the logs are retained for at least 90 days on the system and one year offline.
- Conduct a daily review of the logs from all devices. Procedures should be in place for escalations of critical alerts.
- Implement an intrusion detection system (IDS).
- Implement file-integrity monitoring (FIM) software.

Patch Management

- Patch the operating system within 30 days of vendor released security patches/hotfixes.
- Keep applications up to date with the latest vendor supplied security patches.

External and Internal Scanning

- Regularly conduct external and internal scanning to proactively find and remediate vulnerabilities.
- Conduct annual external and internal penetration testing at least once a year and after any significant infrastructure or application upgrade.

Policy and Procedures

- Conduct employee security awareness training to educate employees on information security best practices at least annually.
- Systems that handle sensitive data should only be used for business purposes. Implement policies and procedures, along with strict monitoring, to ensure misuse does not take place (i.e. installing computer games, unlicensed software).

SECTION 2

THREAT INTELLIGENCE

While the Data Compromise section details the final phases of a cybercriminal operation (namely, finding, accessing and extracting data), the Threat Intelligence section documents the commencement of the attack: the stages in which crooks find and exploit vulnerabilities, and then take possession of servers or clients to siphon valuable information.

We begin by examining some of the high-profile threats that made headlines and had security researchers talking in 2015. From there, we'll examine some of the attacks we've seen that affected popular web platforms during the year, including a special look at how attackers poison the supply chain used by software developers to build their applications. We'll also look at the latest trends in spam, phishing, and other threats delivered by email.

Exploit kits—all-in-one packages that criminals buy or rent to build their own attack campaigns—are one of the more popular mechanisms attackers use to commandeer lots of computers. We'll take a look at the biggest ones, and investigate the latest trends in exploitation tricks and techniques, including using "malvertisements" to attack computers from legitimate, respected websites. We'll also spend some time on some of the prominent malware families we've dealt with, including custom-built malware that attacks point-of-sale (POS) systems to steal payment card data.

HIGH-PROFILE THREATS

It's been a truism in marketing for decades: You can increase interest in your product by giving it a catchy name. In an effort to raise awareness about prominent threats, security researchers lately have begun giving their "products"—vulnerabilities they've discovered in popular applications and protocols—memorable names. While the increased attention this has garnered for dangerous vulnerabilities is commendable, it's important to realize that vulnerabilities without names can be just as worthy of attention—and, in the case of "zero-day" vulnerabilities for which no security patch is available, even more so.

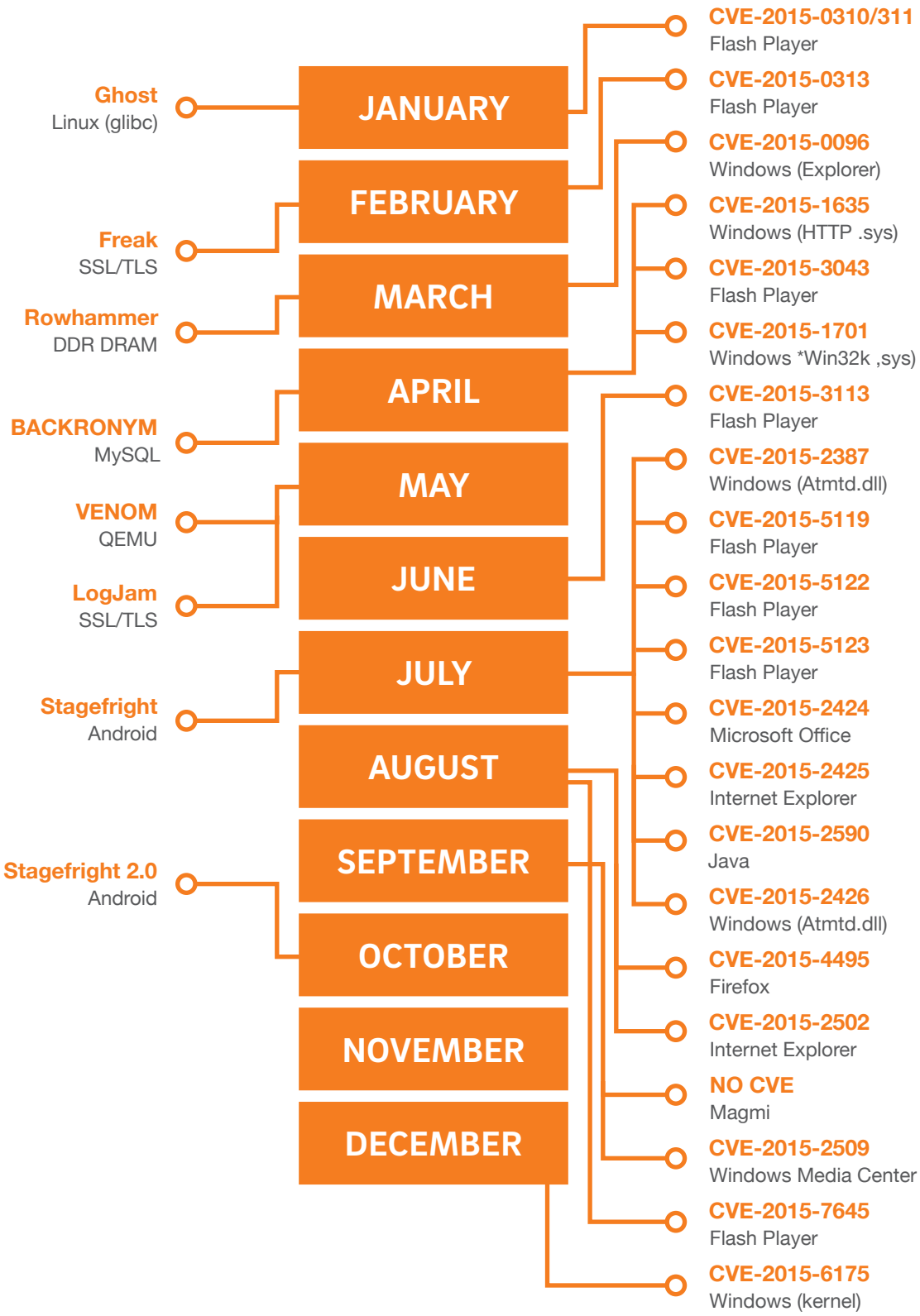
CELEBRITY VULNERABILITIES

The era of the "celebrity vulnerability" began in 2014 when researchers disclosed a serious vulnerability in the OpenSSL library that underlies much of the secure communication on the web. Formally designated CVE-2014-0160 under the Common Vulnerabilities and Exposures (CVE) system used to track vulnerabilities, it became far better known by a nickname given to it by its discoverers: "Heartbleed." News of the Heartbleed vulnerability and its impact spread widely in the technical press and even in the general news media. Mindful of the role the memorable nickname likely played in raising awareness about Heartbleed and prompting businesses to quickly patch the vulnerability, other researchers began assigning nicknames to significant vulnerabilities they discovered, including "Shellshock" and "Poodle," which also garnered a fair amount of media coverage… and the celebrity vulnerability was born.

2015 saw the introduction of several named vulnerabilities, but none of them approached the level of Heartbleed, a significant security risk that could be exploited easily on hundreds of thousands of web servers running the vulnerable OpenSSL software. The most severe of these vulnerabilities (dubbed "Ghost," "Venom" and "Stagefright") could only be exploited under certain exceptional conditions; others only affected a small number of systems, or were more theoretical than practical.

NAMED VULNERABLITIES

UNNAMED ZERO-DAYS



Ghost (CVE-2015-0235)

CVE-2015-0235, a buffer overflow vulnerability affecting all versions of the Linux GNU C library (glibc) through glibc-2.17, was disclosed in January. Researchers dubbed the vulnerability "Ghost," after the name of the vulnerable function, gethostbyname(),

The bug was patched in glibc-2.18 in May 2013, but was not marked as a security bug so the fix did not make it into many common Linux distributions. The researchers who discovered the vulnerability were able to use it to trigger remote code execution on Exim mail servers, but noted that the buffer overflow could not be triggered in most other popular Linux components. Trustwave SpiderLabs researchers were able to exploit the vulnerability through the XML-RPC pingback feature in WordPress to cause a segmentation fault in PHP, but did not achieve remote code execution.

So this Ghost turns out to be not very scary at all. Nevertheless, patching known bugs is a good idea on principle, so Linux users and administrators should check their package managers and make sure they've got glibc-2.18 or later, and update their systems if not. WordPress administrators who don't make use of XML-RPC should disable it.

VENOM (CVE-2015-3456)

VENOM, an acronym for "Virtualized Environment Neglected Operations Manipulation," is a nickname for CVE-2015-3456, a vulnerability disclosed in May that affects several open source virtualization platforms. The vulnerable component is the virtual floppy drive controller in QEMU, an open-source hypervisor that is used by a number of popular virtualization solutions, including Xen, Virtual Box and KVM. (The Microsoft Hyper-V and VMware virtualization solutions use proprietary hypervisors that do not contain the vulnerability.)

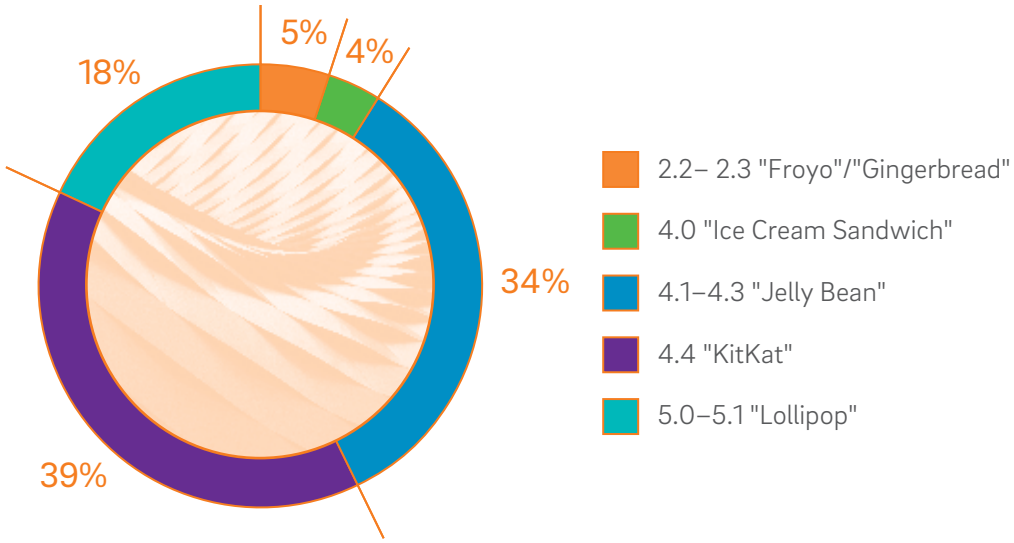
VENOM is one of the more serious virtual machine escape vulnerabilities that we've seen. It is distinguished both by the number of different virtualization platforms affected and the fact that it can lead to arbitrary code execution even with default configurations in some circumstances. Nevertheless, the fact that an attacker must have access to an existing virtual machine limits VENOM's utility somewhat, as does the fact that the vulnerability does not affect Microsoft Hyper-V and VMware, two of the most popular virtualization solutions in business environments. A typical attack scenario might involve an attacker targeting a hosting company that uses one of the vulnerable virtual environments. The attacker would purchase an inexpensive virtual private server instance and use VENOM to breach the hosting server, hoping to find high value targets hosted on the same hardware.

Stagefright (CVE-2015-1538 et al.)

"Stagefright" is actually the name of a code library in the Android OS that provides a number of low-level functions related to media playback. In July, researchers discovered several significant vulnerabilities in the Stagefright library that affect almost all Android users. To exploit the bug, an attacker need only send the target an MMS message containing a specially constructed media file. Because the Stagefright library performs tasks such as generating thumbnails and accessing metadata like file size, length and codec in addition to media playback, it's not necessary for the recipient to actually view or listen to the media to trigger the exploit—simply displaying the MMS message is often sufficient. If the recipient has configured Google Hangouts as the default text messaging app on the phone, in fact, simply receiving the message can trigger the exploit, as Hangouts processes the malicious message behind the scenes without any user interaction necessary.

The Stagefright vulnerabilities affect approximately 95 percent of all devices running Android version 2.2 ("Froyo") or higher, with more damage being possible on older versions. Android version 4.1 ("Jelly Bean") and later provides security protections like application sandboxing that make the vulnerability harder to exploit, although Android still runs the Stagefright library in the root context in Jelly Bean, thereby giving successful attackers full access to the device. Stagefright runs in a more limited context in Android version 4.4 ("Kit Kat") and later, but an attacker can still perform malicious actions like controlling the device's microphone and camera, as well as accessing files on external storage cards.

ANDROID VERSION SHARE, JULY 2015



Google published patches for the bug quickly, but because handset manufacturers decide when to push operating system updates out to devices, many Android users were left waiting for the fix.

Stagefright 2.0 (CVE-2015-6602, CVE-2015-3876)

In October, researchers disclosed two more vulnerabilities in the Android media processing libraries. CVE-2015-6602, a flaw in the libutils, affected every version of Android since version 1.0, released in 2008. This vulnerability can be triggered by exploiting CVE-2015-3876, another vulnerability in the Stagefright code library that caused so many problems earlier in the year. Together, these two vulnerabilities could be exploited to execute remote code on even the newest versions of Android.

B-LIST CELEBRITY VULNERABILITIES

We use the "B-list" label to describe the less significant named vulnerabilities of the year, but in truth none of the named vulnerabilities disclosed in 2015 rises to the level of 2014's Heartbleed. It might be more accurate to consider these vulnerabilities "C-listers," if that.

FREAK (CVE-2015-0204)

FREAK (for "Factoring RSA Export Keys"), disclosed in March, is an SSL man-in-the-middle attack that forces weak encryption negotiation, similar to 2014's POODLE. A successful attack forces the communicating parties to use weak 40-bit encryption, a legacy of United States government restrictions on the export of strong cryptography in the mid-1990s. Most web servers and browsers have dropped support for export-level encryption, significantly limiting the potential impact of this vulnerability.

Rowhammer (CVE-2015-0565)

Rowhammer, disclosed in March, is an attack on a physical vulnerability in certain varieties of DDR DRAM memory chips. Modern chips contain so much capacity at such high densities that it becomes possible for individual cells to interact with each other on a physical electric level, as the Rowhammer vulnerability demonstrates. The attack involves repeatedly accessing, or "hammering," a row of memory to cause a bit in an adjacent row to flip from 0 to 1 or vice versa. Researchers have demonstrated two proof-of-concept attacks using Rowhammer on a Linux system, one that gains kernel-level privileges and another that escapes a native sandbox used to isolate software privileges.

Rowhammer can be performed with five lines of assembly, and has been demonstrated to work on DDR3 chipsets. DDR4 and chips that perform Error Correcting Code are not vulnerable.

BACKRONYM (CVE-2015-3152)

The trend toward assigning vulnerabilities memorable, catchy names reached (intentionally) ridiculous new heights with "Bad Authentication Causes Kritical Risk Over Networks, Yikes MySQL," or BACKRONYM. First disclosed in April, BACKRONYM is a vulnerability in the popular MySQL database system that can be exploited by a man-in-the-middle attacker to disable SSL encryption on a session between a MySQL client and the database. The vulnerability was actually fixed in MySQL 5.7.3, released in December 2013, but at the time the vulnerability was disclosed, the 5.7 branch of MySQL was only available as a preview release, and consequently almost all MySQL production databases worldwide were still on 5.6 or earlier, which had no available fix for the vulnerability. Because the vulnerability requires that the attacker be in a position to perform a man-in-the-middle attack, and because most MySQL instances don't use SSL anyway, the impact of the vulnerability was limited.

Logjam (CVE-2015-4000)

Disclosed in May, Logjam is a vulnerability in TLS that can be used to force an encrypted session to use weak export-grade keys, similar to FREAK. The researchers who disclosed Logjam also determined that millions of servers use the same large prime numbers to initiate the Diffie-Hellman key exchange at the beginning of a session, thereby greatly reducing the complexity of the computation required to break them.

ZERO-DAY VULNERABILITIES

When “white hat” security researchers discover a vulnerability, they typically disclose it privately to the affected software vendor, either directly or through an intermediary. The vendor then develops a patch and releases it, along with details of the vulnerability. Upon learning of the vulnerability’s existence, criminal exploit writers rush to develop and deploy exploits to vulnerable computers before they can be patched. However, a computer user or administrator who installs all vendor patches promptly as they are released generally faces little danger from such vulnerabilities.

When cybercriminals discover a vulnerability before the vendor or a white-hat researcher does, they can sometimes create an exploit and deploy it to computers before anyone else knows the flaw exists. This is called a “zero-day” exploit. Zero-day exploits are the Holy Grail for cybercriminals because these exploits allow them to take control of systems without worrying about being detected and blocked by security products. It’s also possible to have a zero-day vulnerability become publicly known without a corresponding zero-day exploit existing in the wild. A zero-day vulnerability is simply a vulnerability that is disclosed before the vendor can release a corresponding patch.

In the present day, zero-day exploits usually become known to security researchers after being used in targeted attacks by advanced persistent threats (APTs). To preserve their value and delay the release of a patch that fixes the vulnerability, cybercriminals rarely use zero-day exploits outside of these situations. Most of the vulnerabilities and associated exploits discussed in this section were initially used in targeted attacks, although exploit kits adopted many of them shortly after initial disclosure for use in opportunistic attacks.

We counted 21 high-profile zero-day vulnerabilities in 2015. That’s one fewer than in 2014, but the 2015 zero-days were more dangerous for the average user overall, as a larger percentage of them can be used to remotely target users: 13 of the zero-days disclosed in 2015 could be used for remote client exploits through web browsers or browser add-ons, compared to just six in 2014.

Flash Player Zero-Days

Adobe Flash Player is a favored target for attackers because of its large installed base and the potential it offers for silently compromising computers with little user interaction required. Almost half of the high-profile zero-day vulnerabilities disclosed in 2015 affect Adobe Flash Player, a major increase from 2014, when only four Flash zero-days were disclosed during the entire year. (See the “Exploitation Trends” section for more information about this.)

Notably, at least two Flash zero-days first surfaced publicly in exploit kits, rather than being picked up by kits only after being used in known targeted attacks. CVE-2015-0311 was first disclosed in January when it showed up in the Angler exploit kit. The following month, CVE-2015-0313, a use-after-free vulnerability, was spotted in a lesser-known exploit kit called HanJuan. Similarities with CVE-2015-0311 lead us to suspect that the two exploits were crafted by the same author.

Other notable Flash zero-day exploits from 2015 include:

- CVE-2015-3113, a heap-based buffer overflow vulnerability disclosed in June and used by the “APT3” targeted attack group. (See the “Exploitation Trends” section for more information about this.)
- CVE-2015-5119, CVE-2015-5122, and CVE-2015-5123, a trio of severe vulnerabilities disclosed in July from a single source.
- CVE-2015-7645, a remote code execution vulnerability disclosed in October after being used by a targeted attack group.

Microsoft Windows Zero-Days

Seven significant zero-day vulnerabilities disclosed in 2015 affected versions of the Microsoft Windows operating system. Of these, the most interesting is probably CVE-2015-0096, a vulnerability in the way Windows Explorer processes .lnk (shortcut) files. The underlying issue was originally disclosed five years earlier as CVE-2010-2568, a vulnerability exploited by the Stuxnet family to sabotage uranium enrichment machinery in Iran. Microsoft published a patch to address the Stuxnet vulnerability in 2010, but the patch was revealed in March of 2015 to be ineffective at completely stopping exploitation. Other Windows zero-days from 2015 include:

- CVE-2015-1635, a vulnerability in Windows HTTP.sys disclosed in April that could be used to execute code remotely on web servers running Windows Internet Information Services (IIS).
- CVE-2015-2426, a vulnerability in the Windows TrueType font rendering engine, disclosed in July.

Other Zero-Days

In addition to the Flash and Windows zero-days discussed previously, two zero-days affecting Internet Explorer were disclosed in 2015, along with one each affecting Java, Mozilla Firefox, and the Magmi add-on for the Magento e-commerce platform.

- Two memory corruption vulnerabilities in Internet Explorer, CVE-2015-2425 (affecting Internet Explorer 11 only) and CVE-2015-2502 (affecting Internet Explorer 7 through 11), were disclosed in July and August, respectively.
- CVE-2015-2590, a vulnerability in Java, was disclosed in July after being used in targeted attacks by the group known as Pawn Storm.
- CVE-2015-4495, a vulnerability in the PDF reader feature of Mozilla Firefox, was disclosed in August.
- In September, researchers at Trustwave SpiderLabs discovered a vulnerability in the default configuration of Magmi, an importer utility for Magento. See the “Poison in the Well: Protecting Software from Vulnerable and Malicious Components” section for more information about this.

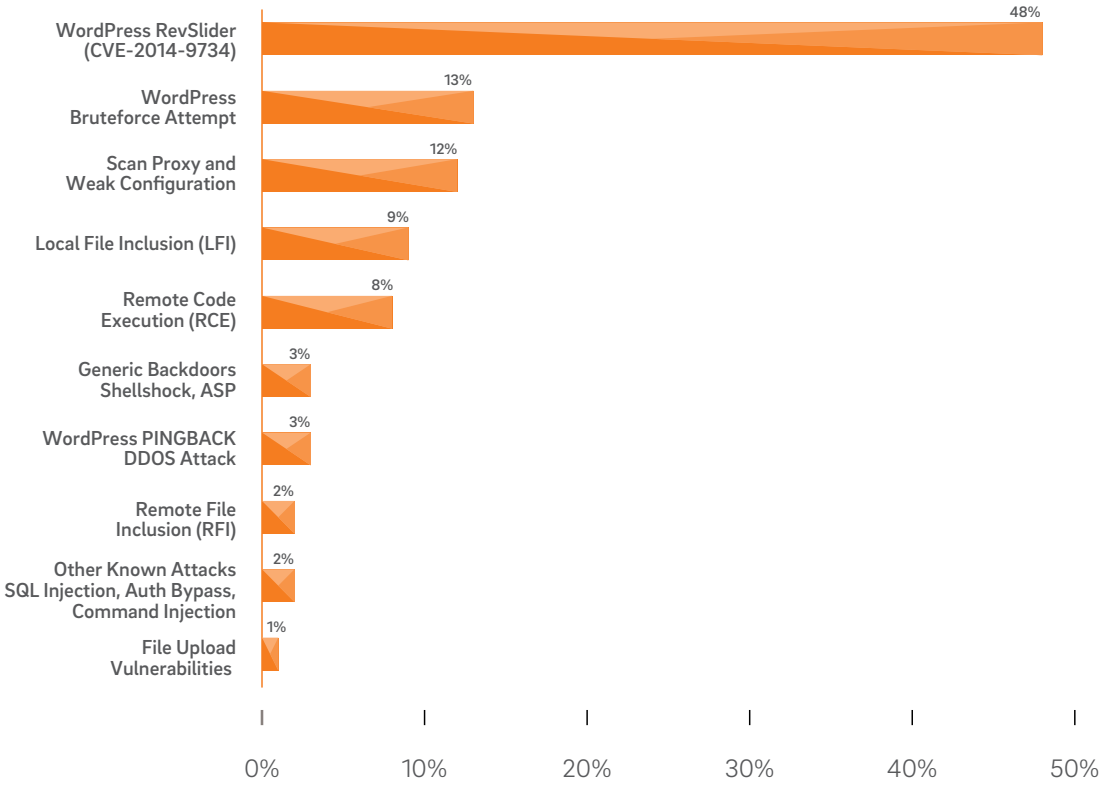
Zero-days might well be always with us, and there’s every reason to think that criminals will continue to invest in purchasing or discovering them. This is one of the main reasons that defense in depth is so vital to information security. Most traditional signature-based solutions like IDS and anti-virus can’t defend against unknown attacks, but network and system monitoring may pick up early signs of exploitation. Equally important is a good incident response process that has been regularly revisited and practiced. This will help your organization to quickly recover from any breach that may occur due to a zero-day exploit.

WEB ATTACKS

Analysis of web application attacks and compromises helps to identify the top attack methods used by cybercriminals in 2015. Our data set includes multiple sources:

- Alerts from Trustwave Managed Web Application Firewall
- Web-specific alerts from Trustwave Managed Intrusion Detection and Prevention systems
- Web honeypot systems
- Publicly available Apache web server log files
- Logs from ModSecurity Web Application Firewall instances deployed as part of the OWASP Web Application Security Consortium Distributed Web Honeypots project
- Trustwave incident response and forensic investigations

TOP OPPORTUNISTIC ATTACK EXPLOIT METHODS OBSERVED BY TRUSTWAVE



Opportunistic Attacks: Techniques and Targets

Security professionals distinguish between opportunistic attacks, in which attackers generally try to infect as many computers as possible for various unlawful purposes, and targeted attacks, in which attackers select a specific business or organization to compromise, typically to steal or damage valuable data. Opportunistic attackers will identify targets by performing search engine queries or by sequentially scanning network block ranges for listening web servers. This information helps an attacker determine which public-facing web servers are hosting applications that are vulnerable to the exploits they intend to use. Opportunistic attackers often don't make attempts to hide their actions from security monitoring systems because it's simply not worth the effort—there are always other potential victims to try.

The majority of opportunistic attacks will attempt to exploit vulnerabilities in well-known, popular web application software. We also see examples of this class of attacker going after older vulnerabilities in websites, hoping relevant updates or patches were never applied. Most of the time, attackers use publicly available exploits rather than zero-day exploits, which are much rarer and more valuable and tend to be used by targeted attackers who have the resources to develop or buy them. Once a usefully exploitable vulnerability is disclosed, however, attackers are quick to adopt them. For example, when Trustwave researchers discovered an SQL injection vulnerability in the Joomla content management system (CMS), attacks using the vulnerability started appearing less than two hours after disclosure.

Upon exploiting a web application or server, opportunistic attackers either install web shells/backdoors to redirect website visitors for the purposes of search engine optimization (SEO), or install an Internet Relay Chat (IRC) client for botnet recruitment.

WordPress: Big Target, Big Payoffs

WordPress is the most popular content management system (CMS) in the world, owning more than half of the market by some estimates. Originally developed as a blogging platform, WordPress's ease of use, flexibility, and large community of users have gained it legions of fans and millions of active websites. Unfortunately, this level of popularity makes it a highly tempting target for criminals, who can take advantage of unpatched WordPress installations to infect hundreds or even thousands of web servers at a time. A solid majority of the known web application attacks—attacks that could be traced to known, widely used exploits or techniques—that we observed in 2015 affected WordPress or WordPress add-ons.

"A solid majority of the known web application attacks—attacks that could be traced to known, widely used exploits or techniques—that we observed in 2015 affected WordPress or WordPress add-ons."

Of these, the largest share targeted a vulnerability in the popular Slider Revolution ("RevSlider") plugin, which displays a rotating gallery of images on a web page. A vulnerability discovered in 2014 enables an attacker to use the plugin to access files elsewhere on the web server, a technique called local file inclusion (LFI). In an audit log dump of the HTTP request from the ModSecurity web application firewall, an attacker attempts to download the WordPress master configuration file wp-config.php, which contains database credentials and other sensitive information and can often allow an attacker to compromise the website.

```
--fd0b151b-A--
[03/Sep/2014:04:23:23 --0500] VAbeC8Co8AoAABmBX5EAAAAL 85.25.242.250 34609 XXX.XXX.XXX.XXX 80
--fd0b151b-B--
GET //wp-admin/admin-ajax.php?action=revslider_show_image&img=../wp-config.php HTTP/1.1
TE: deflate,gzip;q=0.3
Connection: TE, close
Host: REDACTED
User-Agent: lwp-request/5.834 libwww-perl/5.834
```

Widespread exploitation of the vulnerability began in September 2014, despite the fact that the flaw had been patched eight months earlier. The risk was exacerbated by the fact that the plugin developer chose to patch the vulnerability silently, without informing its user base of the vulnerability's existence. Many site owners never updated the plugin, and their properties remained vulnerable when widespread exploitation began. By some estimates, more than 100,000 WordPress sites may have been compromised this way.

Attacks on the RevSlider plugin alone comprised nearly half (48 percent) of the known attacks on web servers observed by Trustwave in 2015. Other WordPress attacks, including distributed denial-of-service (DDoS) attacks and simple brute force cracking attempts, accounted for another 23 percent, meaning that WordPress was involved in more than 70 percent of known attacks that we observed. In light of the platform's attractiveness to attackers, WordPress users should carefully monitor their sites for suspicious activity and keep both the core WordPress engine and any installed add-ons up to date to reduce the risk of exploitation.

Other Attacks

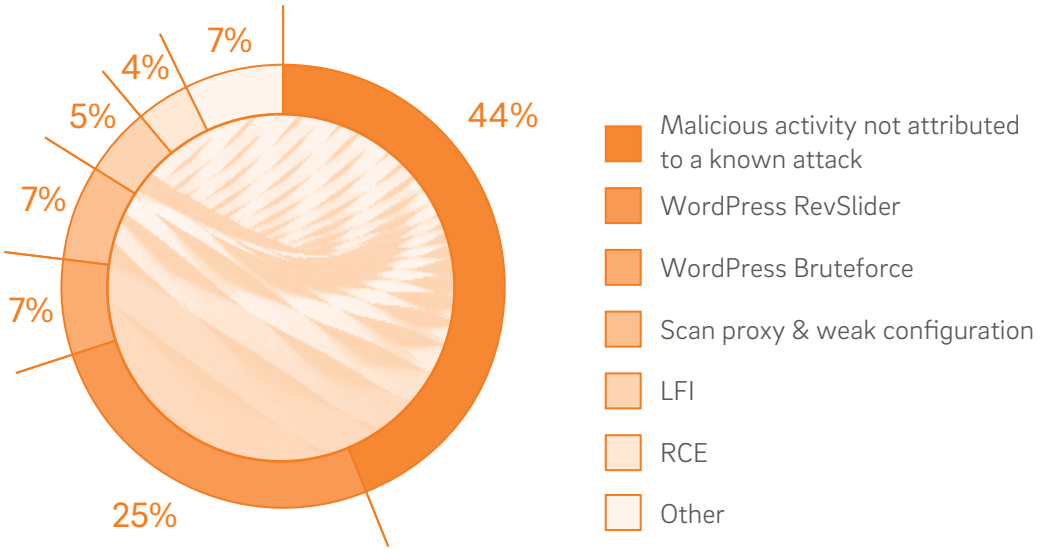
Scan proxy and weak configuration issues were involved in 12 percent of the known attacks on web servers that we observed in 2015. This is a catch-all category for attacks that take advantage of poorly configured web applications or web servers, such as a password file that is not adequately protected from compromise.

Local file inclusion attacks that did not involve the RevSlider plug-in accounted for 9 percent of known attacks. Remote code execution attacks, which allow an attacker to execute arbitrary code on the compromised server, accounted for another 8 percent. Several other attacks and techniques accounted for a few percent each.

Exploit Detection with Web Application Firewalls

The known attacks discussed here form only part of the web application security story. Almost 45 percent of all web application attacks observed by Trustwave in 2015 did not involve known attacks at all, but were detected by generic web application firewall rules that look for behaviors such as cross-site scripting (XSS) and blind SQL injection, and by the Trustwave Web Application Firewall learning-based anomaly detection.

KNOWN AND UNKNOWN ATTACKS IN 2015



Reviewing web application firewall events for detected malicious activity is instrumental in detecting, and defending against, new zero-day exploits in the wild. As attackers come to rely more and more on these supposedly "undetectable" threats, defenses like web application firewalls will become ever more important.

POISON IN THE WELL: PROTECTING SOFTWARE FROM VULNERABLE AND MALICIOUS COMPONENTS

In all but the simplest cases, every application built today includes code developed by third parties, in the form of libraries or modules. While this makes it possible for developers to do great things by building on the work of others, it complicates the job of making secure software. While developers may rigorously follow secure coding practices, the companies, open-source projects, and independent developers who produce the components they use may not. Open-source software can be particularly vulnerable to this kind of supply chain compromise due to the decentralized nature of its development and distribution. In the worst scenario, an attacker can even trick a vendor into compiling malicious code right into an application, giving the attacker a wide-open backdoor into every computer or server on which the application is installed.

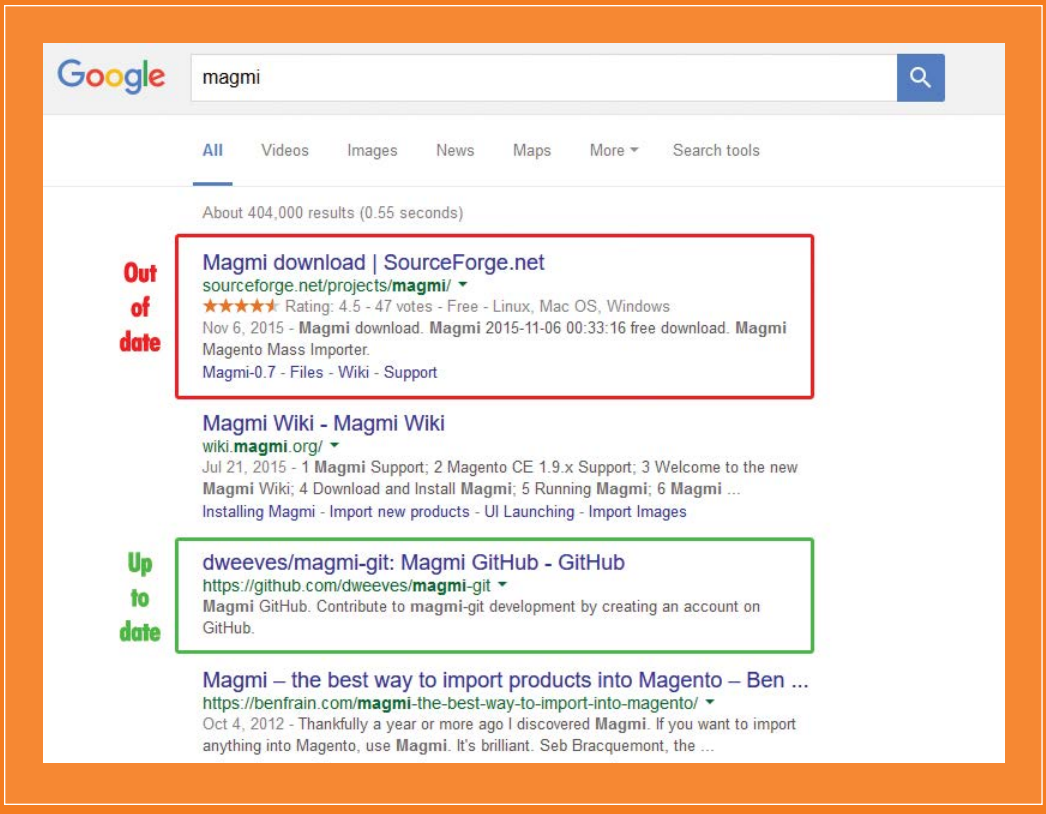
This section takes a look at some of the ways attackers can take advantage of vulnerable software supply chains, and what software developers and vendors should do about the problem.

The Repository Problem

A single open-source software project may be hosted in multiple places. Sometimes the project team may upload the software to multiple repositories itself, to maximize its availability and reach. In other cases, third parties might host copies of the software for download themselves without the knowledge or active participation of the project team, as either an attempt to monetize page views or an aid to users in a particular organization or geographic region. These multiple sources can easily get out of sync if an up-to-date version of the software is uploaded to one repository and is not immediately copied by all the others. This creates a significant security problem when the latest version includes a patch for a newly disclosed vulnerability. Exploit writers are often quick to take advantage of vulnerabilities in popular software after a patch is released, in a rush to infect as many computers as possible before the patch is applied. Unfortunately, developers who rely on an out-of-date repository for a component may not receive the security update, or even know it exists, for weeks or months—and the software they develop using the vulnerable component may itself remain vulnerable during that crucial period. An incident involving the popular Magento e-commerce platform is a case in point.

In September 2015, Trustwave SpiderLabs researchers discovered a vulnerability in the default configuration of Magmi, an independently developed open-source utility for Magento that is hosted on both GitHub and SourceForge. Trustwave disclosed the vulnerability privately to the Magmi developer, who uploaded a patch to GitHub, and to Magento, which issued a security notification and contacted site owners it believed to be vulnerable. The following month, we began to observe HTTP requests that targeted the vulnerable version of Magmi to perform directory traversal attacks.

At that time, the version of Magmi available for download at SourceForge still contained the vulnerability that had been patched a month previously at GitHub. The project page at SourceForge gave no indication that the version available for download there was not up to date—in fact, the project README file said that the two repositories are kept in sync. Even more troubling is that a search for “magmi” on Google listed the SourceForge repository first and more prominently than the GitHub repository, which was listed third.

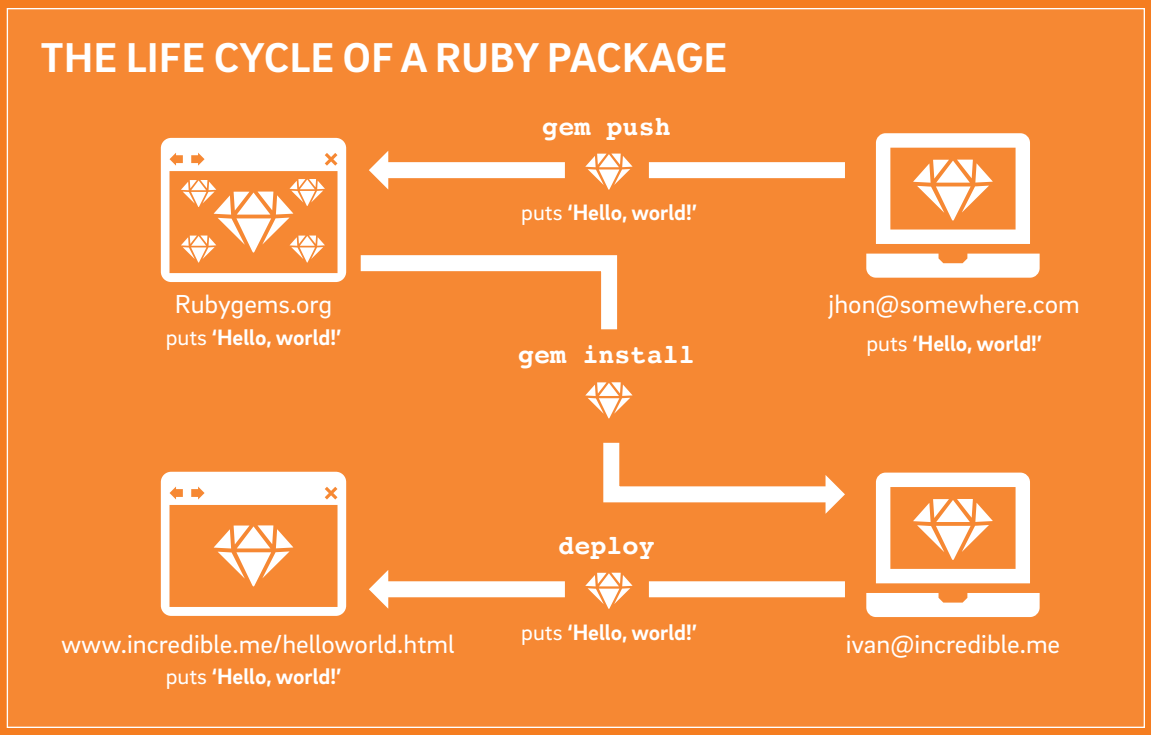


The patch was finally uploaded to SourceForge in November, two months after it was committed on GitHub and a month after we began observing exploit attempts in the wild. During that time, the SourceForge project page reported several hundred downloads of the vulnerable version per week.

While this is a particular case of two repositories being out of sync by a couple of months, a potentially more serious problem is created when an open-source project abandons a repository altogether but does not delete its files there. SourceForge, the oldest major open-source repository, offers numerous examples of this: Many projects have moved development work from SourceForge to other repositories in recent years, often leaving behind old, vulnerable versions for download. It's important to understand, however, that the problem is not unique to SourceForge Just as many projects left abandoned files on SourceForge after moving to GitHub and other repositories, projects might also abandon GitHub in the future if a more attractive option comes along. Developers need to ensure that they are obtaining their dependencies from up-to-date, actively maintained sources. For their part, open-source projects should do everything in their power to ensure that security updates get committed quickly wherever they host code.

The Package Problem

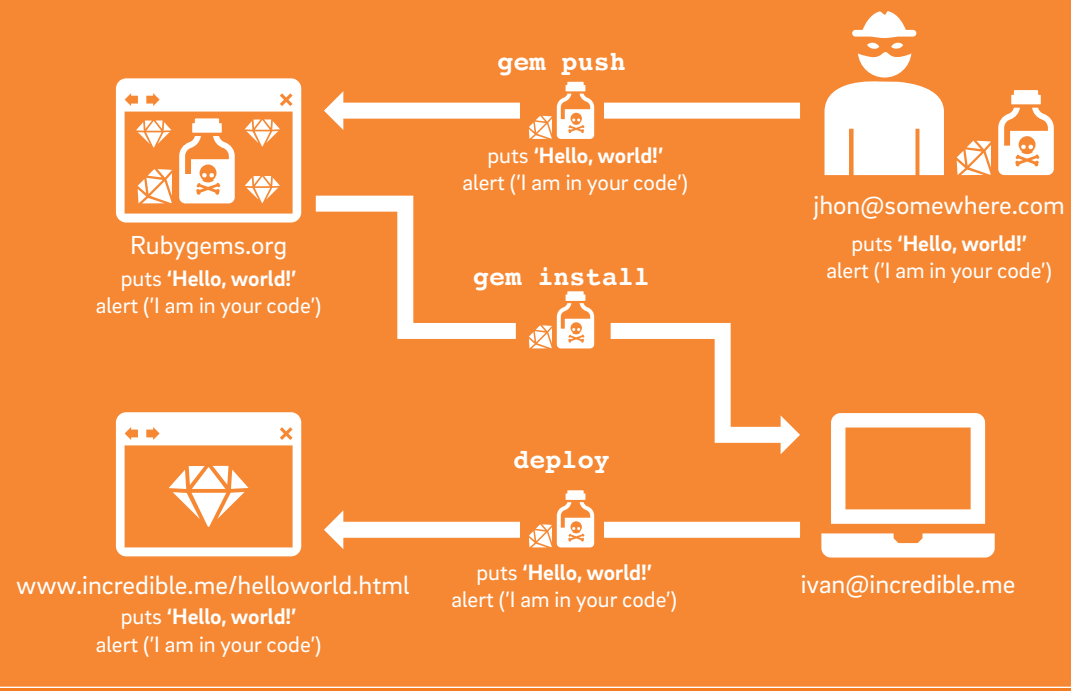
Package management infrastructures are attractive targets for attackers because of the potential they offer for building compromises directly into software. Most popular programming languages are supported by package managers that provide a way for developers to share code libraries, development tools, and other programs and files through a trusted community website. For example, users of the Ruby programming language can install the RubyGems package manager to discover and download packages (called "gems") from the RubyGems.org website. Developers can also package their own code into gems and upload them to the site to share with others. Package managers are built into many popular integrated development environments (IDEs) and provide access to a wide variety of packages, including community-created packages.



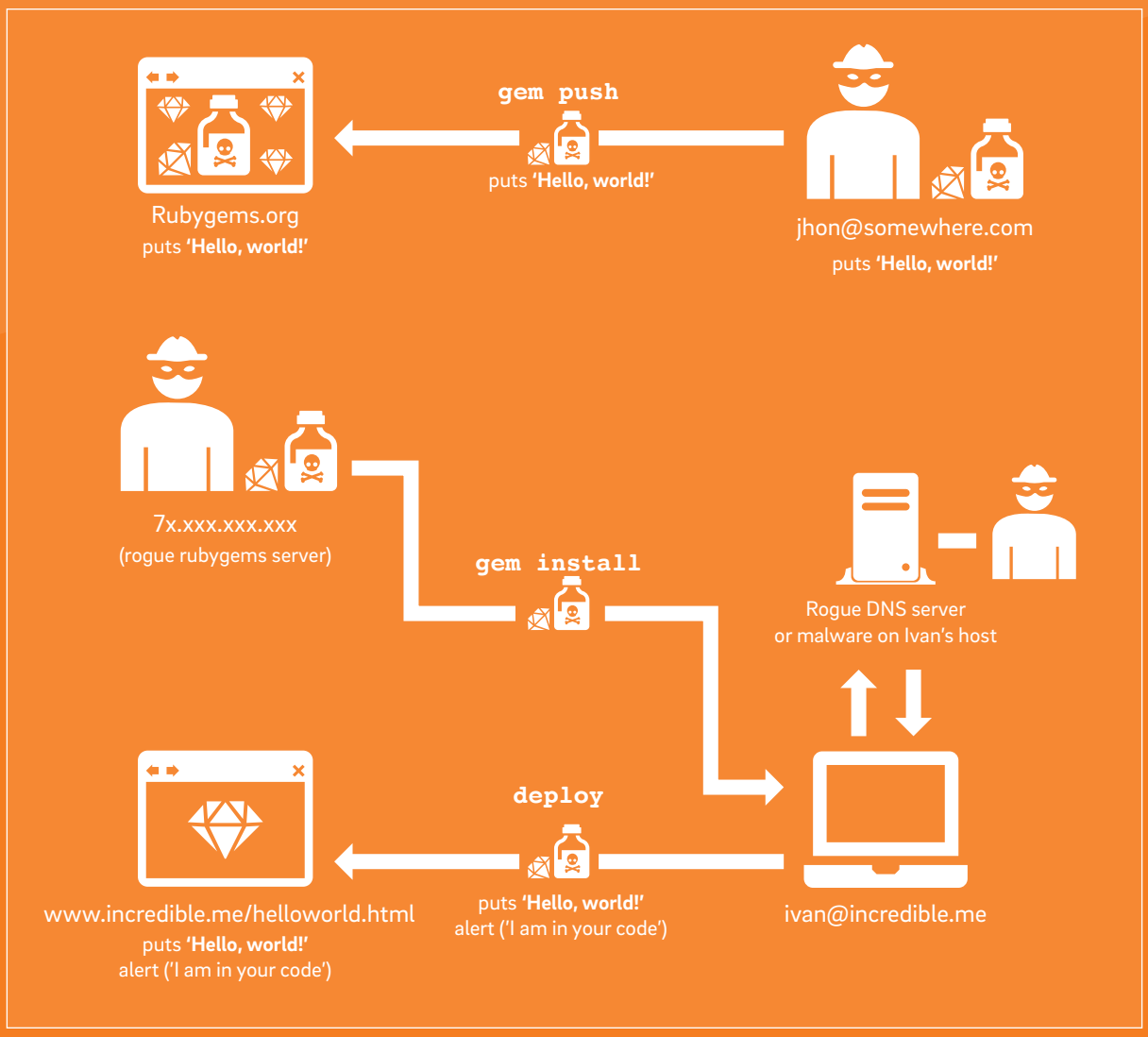
Successfully compromising the package management infrastructure, therefore, represents one of the most effective mechanisms for compromising a web application. Using a package in a web application essentially means making the package a part of the application. Code in the package executes in the context of the application, which could allow it to access a wide range of sensitive functions and data. An attacker who successfully induces a developer to install a malicious package on a web server could gain near-total control of the server.

A trivial attack would involve simply developing and uploading a malicious package and persuading other developers to download and install it. This could be an opportunistic attack aimed at random developers, or the attacker could target a specific application (for example, by crafting an email message luring a targeted developer to install the package). In general, coarse tactics like misleadingly named packages can be expected to have a low probability of succeeding. The community ratings systems and download counts employed by most package repositories tend to guide users toward widely used packages and away from unfamiliar ones. In at least one case, however, a proof-of-concept attack based on similarly named packages from two different repositories attracted some inadvertent downloads in 2015.

SIMPLE ATTACK VECTOR – PUBLISHING A POISONED PACKAGE



In 2015, Trustwave researchers discovered a more advanced attack vector that exploited a vulnerability in the RubyGems client (CVE-2015-3900, patched in May 2015). The RubyGems client has a gem server discovery function that uses a DNS SRV request to locate a gem server. Vulnerable versions of the client do not require that DNS replies come from the same security domain as the original gem source, allowing arbitrary redirection to attacker-controlled gem servers. In the attack scenario, the adversary uses malware or DNS hijacking to redirect requests intended for the legitimate RubyGems server to a rogue gem server under the attacker's control. When the victim uses the RubyGems client to search for a gem, the returned listing of gems and their details comes from the legitimate server, as intended—but downloading one will retrieve a malicious gem from the rogue server.



The Code Problem

It's important for developers to understand how the nature of open-source software affects its security profile. The open, decentralized, community-based nature of open-source software, often cited as being among its greatest strengths, also creates a number of unique security challenges that developers should keep in mind. The problem of out-of-date repositories is a consequence of the fact that open-source software can be freely redistributed. The risk created by potentially malicious packages arises from the bottom-up, community-based culture of shared code. While the threats we describe are fairly rare, a more commonplace risk is that open source is highly accessible to "white box" vulnerability discoveries by security researchers and attackers. Whereas vulnerabilities in proprietary software must usually be discovered through "black box" techniques like feeding a component malformed data and monitoring its output for irregularities, a knowledgeable person can often discover vulnerabilities in open-source software by simply reading the source code. In October 2015, for example, Trustwave SpiderLabs researchers discovered a SQL injection vulnerability (CVE-2015-7587) in Joomla, a popular CMS platform with three million active sites worldwide, by reading the source code. Fortunately, the same traits of openness and accessibility that make it easier to exploit vulnerabilities also make it easier to identify and resolve them—our researchers found the Magmi vulnerability quite readily by simply doing a visual comparison of the affected PHP file from both repositories.

Keeping the Supply Chain Safe

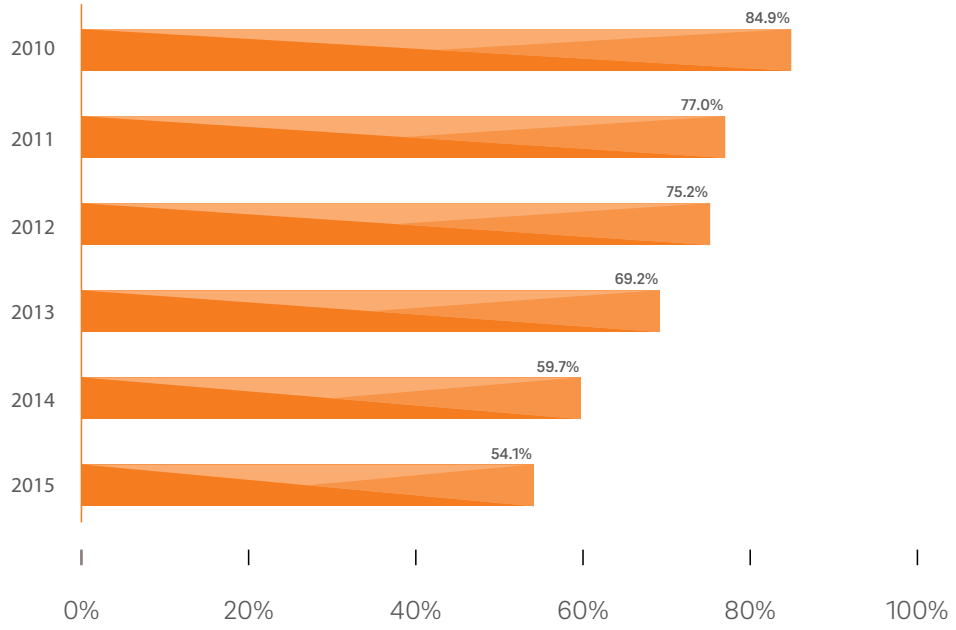
Web application developers should perform a security review of their development and continuous integration environments to help ensure that all components originate from trusted sources and all security updates are integrated as quickly as possible. The Open Web Application Security Project (OWASP) lists "Using Components with Known Vulnerabilities" as No. 9 on its recent list of the Top 10 security risks facing web applications. It advises software projects to have a process in place to:

- Identify all components and the versions you are using, including all dependencies
- Monitor the security of these components in public databases, project mailing lists, and security mailing lists–and keep them up to date
- Establish security policies governing component use, such as requiring certain software development practices, passing security tests, and requiring acceptable licenses
- Where appropriate, consider adding security wrappers around components to disable unused functionality and/or secure weak or vulnerable aspects of the component

EMAIL THREATS

The problem of unwanted email—spam—is familiar to anyone who has used the Internet. Opportunistic attackers connect compromised computers together into networks called botnets, which are the primary source of spam. Despite the significant global impact of spam, the spamming "industry" itself is not especially large. We estimate that fewer than a dozen operators are responsible for 80 to 90 percent of the spam sent today.

SPAM AS A PERCENTAGE OF TOTAL INBOUND EMAIL



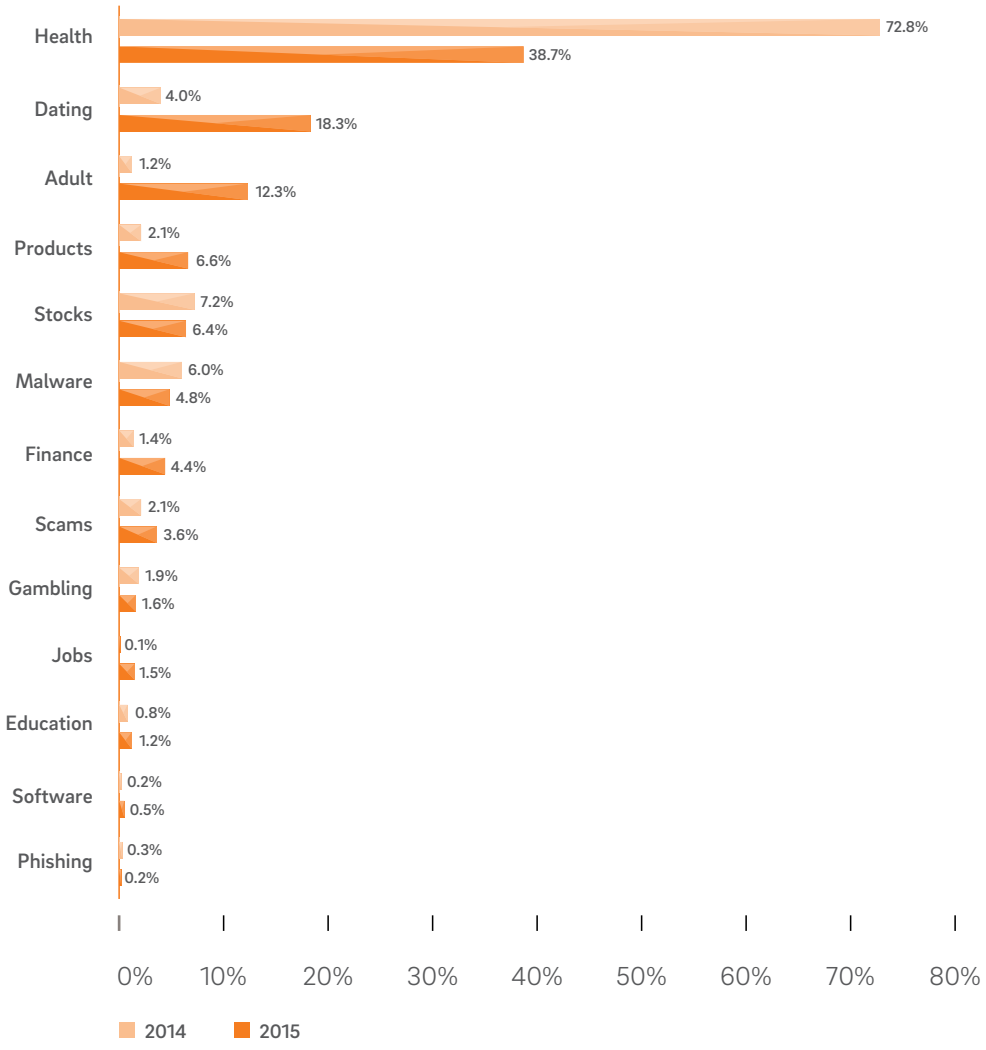
After several years of declines, spam volumes fell again in 2015 to just 54 percent of total inbound email processed by the Trustwave Secure Email Gateway Cloud service. This continuing decline suggests that the economics and/or risk of high-volume spamming may have fundamentally changed, due to a combination of general lack of profitability and a growing fear of prosecution on the part of large botnet operators. Several high-profile botnet takedowns occurred in 2015, including the Ramnit and Simda botnets. Authorities have successfully disrupted the operations of a number of large spammers over the past few years, while others appear to have left the junk mail business voluntarily.

"If the trend continues, spam will account for less than half of inbound email volume in 2016 for the first time in more than a decade."

Categories of Spam Subject Matter

Spammers make money by signing on to affiliate marketing programs. In these programs, affiliates distribute advertisements for the vendor's products or services, and the vendor pays the affiliates a percentage of each sale they bring in. The spammers switch freely between affiliate programs according to which ones they believe will make them the most money. When the profits from spamming messages for one program drop off, they switch to another that offers better terms or a more attractive product. This arrangement can lead to significant changes in the types of spam sent over time.

SPAM CATEGORIES 2014-2015



SPAM CATEGORIES WITH THE LARGEST YEAR-TO-YEAR CHANGES, 2014-2015

Spam Category	2014	2015	Percentage point change
Online dating sites	4.0%	18.3%	14.3%
Sex, x-rated	1.2%	12.3%	11.1%
Watches, electronics, general goods	2.1%	6.6%	4.5%
Loans, mortgages etc.	1.4%	4.4%	3.0%
Pharma, pills potions	72.8%	38.7%	-34.1%

In 2014, health-related spam peddling pills, potions, and other miracle cures made up almost three-fourths of the spam messages we analyzed. In 2015, that portion dropped dramatically, to 39 percent—which was still enough to make it the largest share of any category. Much of the rest was replaced by messages advertising online dating sites (18 percent), adult products and services (12 percent), and general products (7 percent), with several other categories each accounting for a small percentage of the total volume.

Malicious Spam

While the relative shares of spam devoted to subjects like health, dating, and sex changed significantly from 2014 to 2015, spam that included malicious payloads, either as attached files or as links to malware on websites, remained relatively constant. Malicious spam accounted for 5 percent of total spam volume in 2015, down slightly from 6 percent the previous year, suggesting that the economic drivers for spamming malware are a little different from spam advertising magic pills and other dubious products.

Common Families

Upatre, a downloader that can install other malware, remained one of the most common malware families distributed by spam. Upatre is most often seen downloading the Dyre banking Trojan, which steals account credentials for financial institutions, but has also been observed downloading the Zbot Trojan. The Cutwail botnet in particular is known to distribute Upatre through malicious attachments in spam messages.

Malicious Office Documents

Attackers often use Microsoft Word, Excel, and PowerPoint files to deliver malware through spam. Zero-day Microsoft Office exploits often originate with advanced persistent threat (APT) groups, which use them in targeted spam campaigns to infect specific individuals associated with the institution under attack. The Tsar Team group, also called APT28 and Operation Pawn Storm, used RTF files containing a zero-day heap corruption exploit in Office (CVE-2015-2424) to attack targets. Opening the RTF file in Word triggered the exploit and allowed Tsar Team to run its own shellcode. Meanwhile, opportunistic attackers use new Office exploits in their own spam campaigns quickly after they are disclosed, making it especially important to keep Office applications up to date as patches become available.

In addition to exploiting vulnerabilities, spammers also use Word documents with malicious macros to spread malware. To complicate detection, at least one attacker has begun saving these files in MHTML format, which is used to package web pages as single files, and then changing the file extension from .mht to .doc or .xls, as appropriate. The file opens normally and displays as expected in Word or Excel, giving no indication that it is actually an HTML-based document underneath, with the malicious macro stored as a Base64-encoded MIME attachment. Traditional email security tools looking for macros in Office documents may not detect macros saved in this unusual way. These files typically downloaded the Dridex banking Trojan until the Dridex network was disrupted by a takedown in October 2015, then shifted to distributing other malware, including Zbot.

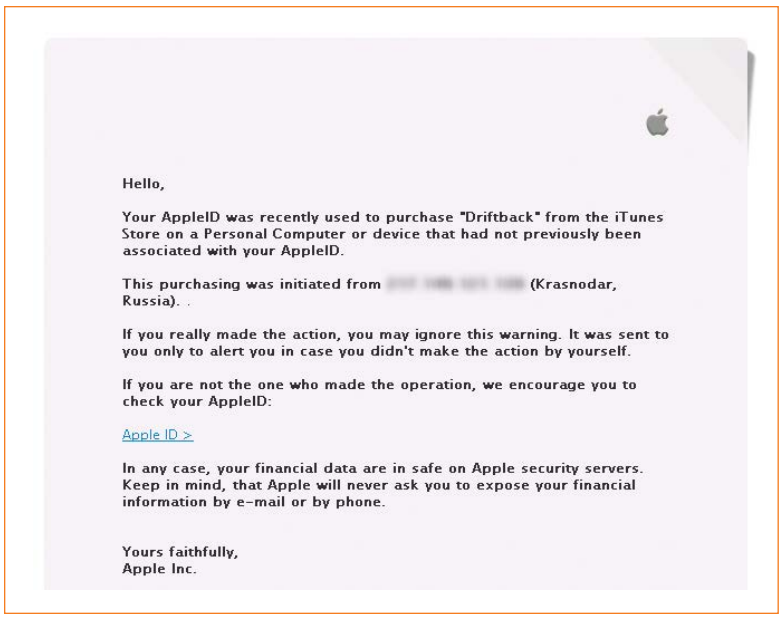
Java and Script Malware

We've also observed attackers using attached JavaScript and VBScript files and Java .jar packages to deliver malware in emails. The Java attachments typically contain remote access Trojans (RATs), like AlienSpy, QRAT, Nanocore, DarkComet and jRat.

Script file attachments are one of the oldest vehicles for delivering malware through email, and most attackers moved to more sophisticated methods years ago, but we continued to see use of this venerable tactic in 2015. The JavaScript files are usually obfuscated to deter detection and often deliver Cryptowall, a ransomware family that locks the infected computer and demands a ransom to restore access.

Phishing and Fraud

Once aimed primarily at collecting login credentials for financial institutions, phishers have expanded their efforts to target other kinds of accounts. At least one campaign targeted Apple users with phishing messages in 2015, the messages claiming to warn the recipient that their Apple IDs had been used to purchase items in distant countries. Following the link in the message leads to a realistic looking login page that asks users for their Apple ID credentials and credit card numbers.



Hello,

Your AppleID was recently used to purchase "Driftback" from the iTunes Store on a Personal Computer or device that had not previously been associated with your AppleID.

This purchasing was initiated from [redacted] (Krasnodar, Russia).

If you really made the action, you may ignore this warning. It was sent to you only to alert you in case you didn't make the action by yourself.

If you are not the one who made the operation, we encourage you to check your AppleID:

Apple ID >

In any case, your financial data are in safe on Apple security servers. Keep in mind, that Apple will never ask you to expose your financial information by e-mail or by phone.

Yours faithfully,
Apple Inc.

We also witnessed the resurgence of a tactic we first encountered a few years ago, in which the phishing message claims that the recipient's mailbox is full and the recipient must re-enter their login credentials to fix the problem. The phisher uses the collected login credentials for a variety of purposes, including sending spam from the victim's account. In fact, the phisher is often able to collect more credentials by sending the same "mailbox full" message from the compromised account to other users in the same organization, who may be more likely to trust messages that are clearly sent from their own domain.

Another tactic that we observed with increased frequency in 2015 is the wire transfer fraud scam, a targeted attack that involves sending fake messages, ostensibly from a company CEO asking an assistant or person in finance to transfer money to a third party. The messages often have slightly misspelled domain names or different Reply-To addresses.

Defending the Email Attack Surface

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

- Deploying an email security gateway – on-premises 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. Carefully consider inbound email policy. Quarantine or flag all executable files, including Java, JavaScript, and VBS attachments, as well as all suspicious and/or unusual file attachments, such as .cpl, .chm, and .lnk files. Create alternative plans for how to handle these types of files coming into the organization.
- Blocking or flagging macros in Office documents, or ensuring that macro protection is enabled in Office while making users aware of the threats.
- 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.
- Ensuring potentially malicious or phishing links in emails can be checked, either with an email security gateway or a web security gateway, or both.
- Educating users – from the rank-and-file up to the C-suite – on the nature of today's email attacks. Conducting mock phishing exercises against staff shows employees that phishing attacks are a real threat of which they need to be wary.

EXPLOITATION TRENDS

Though the tactics used by exploit kits and advanced persistent threats (APTs) change constantly, one thing remains constant: Exploit writers like to seek out the low-hanging fruit. When one technology becomes too difficult to exploit effectively, attackers move on to another. In recent years, Java was the component targeted most often, and most successfully, by exploit kits. Oracle and the major browser makers responded by adding key mitigations to reduce the risk from running Java in the browser, including a click-to-play feature that requires users to interact with each Java applet before it can execute, and blocking unsigned and self-signed applets from running in the browser. As a result, exploit kit authors moved on to other components—including 2015's biggest target, Adobe Flash Player.

Flash, Flash, Flash

To say that exploit kit makers targeted Adobe Flash Player in 2015 is a significant understatement. Of the 14 new exploits added to kits during the year, one targeted Internet Explorer, one targeted Silverlight, and 12 targeted Flash Player. Our observation of version 3.0 of the RIG exploit kit found that more than half of the exploitable traffic to the kit came from Flash. (See the "Exploit Kits" section for more information about RIG.) Advanced persistent threats (APTs), which involve sophisticated attackers who target particular institutions and individuals, also made heavy use of Flash exploits in their attacks. For example, Flash zero-day CVE-2015-3113 was first detected in a campaign waged by an attack group which researchers dubbed "APT3."

In part, the popularity of Flash Player among attackers has been self-reinforcing: When a particular component becomes known for having exploitable vulnerabilities, exploit writers tend to spend more time looking at the component to find more of them. Aside from that, though, Flash Player can be tricky to build mitigations into because of how it is used in the real world.

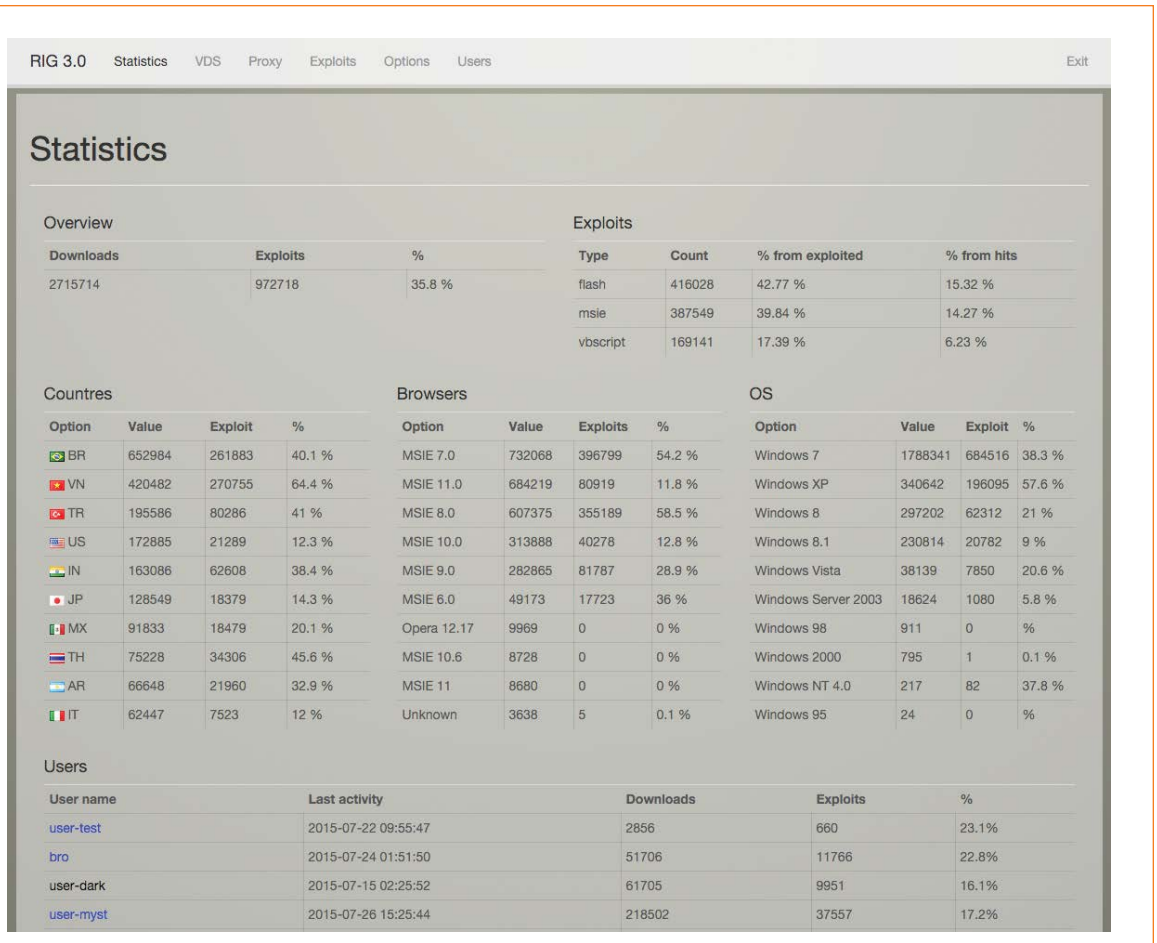
Java custodian Oracle and the major browser makers were able to successfully halt most Java exploitation using relatively simple mitigations such as click-to-play, which can be implemented without changing any of the internals of the Java virtual machine or the Java Runtime Environment (JRE) themselves. Unfortunately, adding click-to-play to Flash Player is not a realistic option, in part because Flash is used so heavily for advertising—requiring the user to click an ad before it can run would disrupt the revenue model on which many sites depend. Building mitigations into Flash requires going deeper, as when Adobe and the Google Project Zero team jointly developed a number of highly technical mitigations for Flash Player version 18.0.0.209, released in July 2015, aimed at foiling some of the techniques used by several recent Flash exploits. (Unfortunately, complex mitigations don't always close off all of the avenues used by attackers, as exploits soon appeared that could bypass at least one of the mitigations added by Adobe and Google.)

Exploit writers continue to target other components and programs, as with the Internet Explorer and Silverlight exploits mentioned earlier, but at a much lower rate than Flash. Browser vulnerabilities that can be profitably exploited are rare these days, owing to mitigations like security sandboxes that limit the browsers' ability to run other code. PDF documents, once a frequently used vehicle for delivering exploits, have become much harder to successfully exploit due to the addition of sandbox features to Adobe Reader and Acrobat. Despite the new Silverlight exploit adopted by kits in 2015, its relatively small installed base limits the plugin's potential for delivering malware.

One odd development in 2015 was the emergence of Microsoft Word Intruder, an exploit kit that produces Rich Text Format (RTF) files containing Microsoft Word exploits. Malicious Office documents are mostly used in APT attacks, as attachments to email messages targeted at individuals within a particular institution. Because they're not an online technology, they are of little use in the broad-based attacks for which most exploit kits are designed. In fact, Microsoft Word Intruder is the first exploit kit we've encountered that appears to be designed specifically for APTs—it even carries a user agreement that warns that the customer's license will be revoked if the kit is used in spam attacks. It will be interesting to see if Microsoft Word Intruder remains a curiosity or is a sign of things to come.

Exploit Kits "as a Service"

The well-known exploit kits of the past were sold like traditional desktop software, wherein the purchaser would receive a package to install on a server under the purchaser's control. In recent years, however, exploit kit authors have moved to cloud-based kits, mirroring the trend in the legitimate software industry—in essence, a criminal version of software-as-a-service (SaaS). Today, most of the major kits use a rental-based business model, wherein customers pay for an account on a server under the kit authors' control and manage their illicit "campaigns" through an administrative interface, such as the one shown here from the RIG exploit kit:



RIG 3.0 Statistics VDS Proxy Exploits Options Users Exit

Statistics

Overview

Downloads	Exploits	%
2715714	972718	35.8 %

Exploits

Type	Count	% from exploited	% from hits
flash	416028	42.77 %	15.32 %
msie	387549	39.84 %	14.27 %
vbscript	169141	17.39 %	6.23 %

Countres

Option	Value	Exploit	%
BR	652984	261883	40.1 %
VN	420482	270755	64.4 %
TR	195586	80286	41 %
US	172885	21289	12.3 %
IN	163086	62608	38.4 %
JP	128549	18379	14.3 %
MX	91833	18479	20.1 %
TH	75228	34306	45.6 %
AR	66648	21960	32.9 %
IT	62447	7523	12 %

Browsers

Option	Value	Exploits	%
MSIE 7.0	732068	396799	54.2 %
MSIE 11.0	684219	80919	11.8 %
MSIE 8.0	607375	355189	58.5 %
MSIE 10.0	313888	40278	12.8 %
MSIE 9.0	282865	81787	28.9 %
MSIE 6.0	49173	17723	36 %
Opera 12.17	9969	0	0 %
MSIE 10.6	8728	0	0 %
MSIE 11	8680	0	0 %
Unknown	3638	5	0.1 %

OS

Option	Value	Exploit	%
Windows 7	1788341	684516	38.3 %
Windows XP	340642	196095	57.6 %
Windows 8	297202	62312	21 %
Windows 8.1	230814	20782	9 %
Windows Vista	38139	7850	20.6 %
Windows Server 2003	18624	1080	5.8 %
Windows 98	911	0	%
Windows 2000	795	1	0.1 %
Windows NT 4.0	217	82	37.8 %
Windows 95	24	0	%

Users

User name	Last activity	Downloads	Exploits	%
user-test	2015-07-22 09:55:47	2856	660	23.1%
bro	2015-07-24 01:51:50	51706	11766	22.8%
user-dark	2015-07-15 02:25:52	61705	9951	16.1%
user-myst	2015-07-26 15:25:44	218502	37557	17.2%

Exploit Kit Innovations

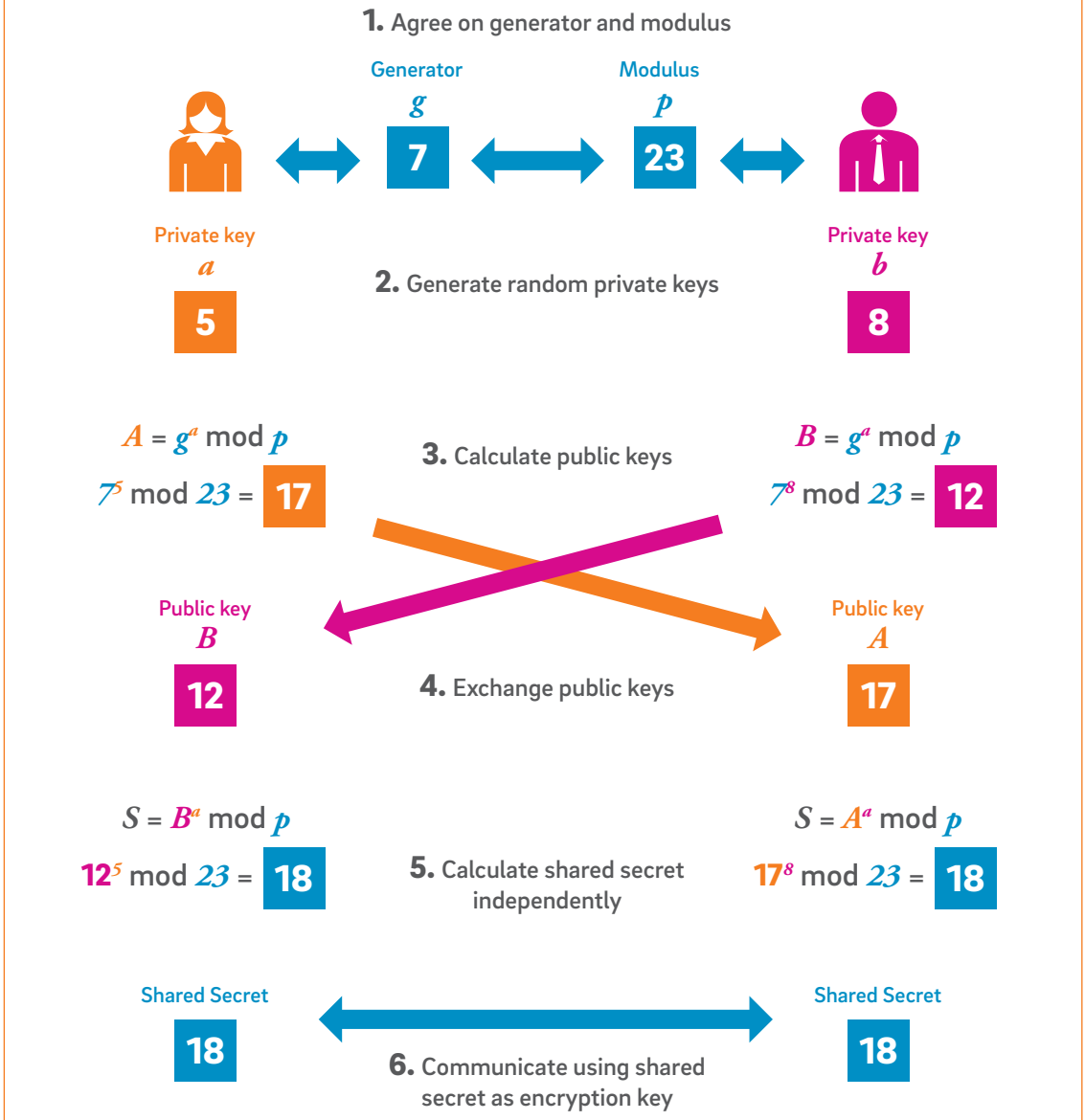
Attackers have turned to increasingly sophisticated techniques in an effort to disguise their traffic and presence from security researchers. Two such techniques, Diffie-Hellman key exchange and domain shadowing, were adopted by the most popular exploit kits in 2015.

Diffie-Hellman Key Exchange

In August 2015, the Angler exploit kit began using the Diffie-Hellman key exchange method to deliver encrypted malware from its landing pages, with the Nuclear exploit kit following suit shortly thereafter. Diffie-Hellman is a method that two parties can use to establish a secure communication channel by exchanging some preliminary information over an insecure one. In the Diffie-Hellman process, the two parties (conventionally called "Alice" and "Bob") use the insecure channel to agree on two prime numbers, a generator and a modulus, that will be used for calculation. Alice then generates a large random number to use as a private key, and uses it along with the two shared primes to calculate a result, which she sends to Bob. At the same time, Bob generates a private key of his own and calculates a different result, which he sends to Alice. Alice and Bob can then each use their own private key along with the public key received from the other party to independently calculate a shared secret: a single integer that both parties know, but which never gets transmitted from one to the other. Alice and Bob can then use the shared secret as a symmetric key to establish a secure channel for further communication. If done properly, any eavesdropper will find it too computationally expensive to calculate the shared secret, even if they manage to intercept all of the numbers that Alice and Bob exchange with each other. Diffie-Hellman is a well-known and widely used protocol, and is one of the key exchange protocols supported by TLS.

"Exploit kit authors have moved to cloud-based kits, mirroring the trend in the legitimate software industry—in essence, a criminal version of software-as-a-service (SaaS)"

A SIMPLIFIED EXAMPLE OF A DIFFIE-HELLMAN KEY EXCHANGE SESSION



If large-enough numbers are used, an eavesdropper will not be able to derive the shared secret from any intercepted numbers.

Angler and Nuclear use Diffie-Hellman to attempt to hide from security products that monitor traffic between client machines and the internet. A typical landing page uses a script to fetch exploit code from a remote server and use it to attack the client, thereby giving the security product an opportunity to recognize it as an exploit and block it before it can harm the machine. By using Diffie-Hellman, a kit can encrypt malicious code before sending it, rendering it unrecognizable to security products while also preventing them from decrypting the code on the fly for analysis.

In Angler's version, the landing page contains a JavaScript implementation of Diffie-Hellman and a 16-byte random number g that will be used as the generator. On the client computer, the JavaScript generates a 16-byte modulus number p and a 16-byte private key a, calculates a public key A, and sends g, p, and A to the server in a JSON object. Next, the server generates its own private key b and uses it along with the data received from the client to calculate public key B. The server then calculates the shared secret S, uses it to encrypt the exploit code, and sends the client a JSON object that includes B and the encrypted shellcode. The client then derives S and uses it to decrypt and execute the shellcode. If a security product or researcher captures the traffic and replays it later, the JavaScript will generate different random numbers, and the resulting shared secret will be different, rendering it useless for decrypting the shellcode or any captured encrypted traffic.

Domain Shadowing

Another technique newly used by Angler, dubbed "domain shadowing" by researchers, involves compromising domain registration accounts and using them to create subdomains of legitimate domains (like admrixlwof.example.com), which are then used to host Angler landing pages. The researchers identified hundreds of domain registration accounts in use by the Angler team, along with nearly 10,000 unique malicious subdomains.

In the past, exploit kits typically used domains of their own, which were often acquired from top-level domain (TLD) registrars that did not have strong countermeasures in place. Because these attacker-registered domains have little to no legitimate traffic associated with them, it is relatively easy for security products to identify them as malicious and block access. Using subdomains of legitimate domains allows the attacker to piggyback on the legitimate site's reputation, and complicates the job of identifying and blocking landing pages. This technique is also simpler for the attacker than fraudulently registering hundreds of new domains, as it requires only that the attacker manipulate the DNS records for existing domains. Angler rapidly rotates the subdomains it uses, with large numbers of subdomains pointing to a much smaller group of IP addresses. (Compare this to the fast flux technique, which involves pointing a single domain at a large and rapidly rotating list of IP addresses.)

Angler started heavily using the domain shadowing technique at the beginning of 2015, though there is evidence of the technique being used at a low level since at least 2011.

APT Techniques

Whereas the opportunistic attackers behind exploit kits primarily seek money, the targeted attackers behind APTs seek information: to steal, to alter, or to destroy. Unlike opportunistic attackers, they aren't trying to spread their malware as widely as possible to maximize their profits—they choose their targets carefully and attack them persistently until they succeed. As these two types of attackers have such divergent goals, it is hardly surprising that the tactics and techniques they use tend to differ significantly as well.

From a strictly technical standpoint, APTs tend to use more kernel mode exploits and other privilege escalation techniques than exploit kits do. Privilege escalation exploits are significantly harder to come by than other exploits, and exploit kits writers don't focus on finding and including them because, by and large, they don't need to. An attacker doesn't need to escalate privilege to inject browser processes, send spam or DDoS packages, or steal the active user's banking credentials. By contrast, APTs looking for sensitive institutional information to compromise need more privileges than are generally available to the low- to mid-level employees who are often the easiest and most reliable route inside an organization. Of course, once an exploit is publicly known, anyone can use it, and exploit kit authors are more than willing to integrate any exploit that serves their purposes. For example, CVE-2015-2426, a kernel-mode exploit that takes advantage of a vulnerability in the Windows TrueType font rendering engine, was adopted by exploit kits almost immediately after being discovered.

Looking to the Future

Flash Player is unlikely to remain the low-hanging fruit forever. As more vulnerabilities are patched and new mitigations are added, exploit writers will increasingly look elsewhere, toward other browser components, browsers themselves, other applications, or operating systems. And of course in the unlikely event that new exploits become too difficult to profitably find and use, attackers will simply move on to other techniques.

MALVERTISING

Malicious advertising has become the most prevalent method used for the mass distribution of malware by opportunistic attackers. Nearly all exploit kits use malvertisements as a distribution channel, and our analysis of the RIG exploit kit, one of the more prominent exploit kits used in 2015, suggests that approximately 90 percent of traffic to the kit originates from malicious advertisements. Even some of the largest ad networks, including networks run by Google and Yahoo, have been misused by attackers to spread malware to unsuspecting users visiting popular websites. Defending against advertising-borne malware is a challenge for ad networks and end-users alike.

Malvertising Techniques
Exploit kits streamline the process of creating and deploying malvertisements. Many of the tools provided by some of the more popular kits mirror those provided by ad networks themselves. Prospective attackers can target specific times of the day for their malicious ads to run, serve ads only to specific browsers or operating systems, avoid targeting mobile platforms, and so on.

DIFFERENT MALVERTISEMENT DELIVERY TECHNIQUES

Click
Clicking the ad redirects to an exploit kit landing page

No-Click
Visiting the page loads an ad with a Flash Player exploit; no interaction is required

Pop-Under
Script attached to the ad automatically opens an exploit kit landing page in a pop-under window

Malvertisements can use a number of different techniques to deliver payloads to the victim. The simplest way involves creating a legitimate-looking ad that leads to an exploit landing page, containing exploits for unpatched vulnerabilities in browsers and popular browser add-ons, when clicked. This technique is often used in malicious text-only advertisements such as those placed through the Google AdWords and AdSense programs, but can also be used in advertisements containing pictures and other media. The HTML and media assets for the ad and its home page are typically stolen from a legitimate advertiser and give no indication that anything is amiss. To deter detection by the ad network used, the page to which the advertisement leads is often completely innocuous when the ad buy is placed, and the attacker alters it shortly thereafter to deliver exploits, either by redirecting to a malicious landing page or loading it into an inline frame (iFrame). The attacker may add multiple redirects to the path in an effort to foil detection by security systems.

Although this technique is simple and easy to implement, its effectiveness is limited by the fact that the computer user must click the ad to be exposed to the exploit. A more dangerous technique involves building an exploit into a Flash-based advertisement itself. Anyone who visits a page containing such an ad using a computer with an unpatched version of Flash Player that is vulnerable to the exploit becomes compromised without having to click anything. As with the previous technique, attackers often begin a campaign with an innocuous ad that contains no exploits, then substitute the malicious version later, either by uploading a new version to the ad server or by changing the HTML of the inline frame that hosts the ad to point to a different Flash file. In other cases, the Flash file is scripted to display no malicious behavior at first, then begin using an exploit after a target date and time has passed. The Neutrino and Flashpack exploit kits are among those that have used no-click malvertisements.

A similar technique involves launching an exploit landing page from a pop-under window launched by script on the page. Loading the contents of the pop-under window with a vulnerable browser or plugin can trigger an exploit. To defeat the pop-up blockers built into most modern browsers, the script usually requires user interaction to open the new window. Clicking anywhere on the page—not just the ad itself—is enough to launch the pop-under containing the exploit landing page.

Challenges

For many years, the conventional wisdom among security experts has been that computer users can greatly reduce their exposure to compromise by staying away from the “bad neighborhoods” of the internet: porn sites, sites trading pirated entertainment and software, and so forth. The emergence of malvertising as a significant exploit delivery mechanism upends that conventional wisdom. Malvertisements have appeared on some of the largest and most well-known websites in the world in recent years. Going forward, users must be taught that legitimate, trustworthy sites can contain exploits through no fault of the sites themselves, and that the best way for individuals to remain safe is to check for security patches regularly and apply them quickly after they are published.

Ad networks and advertising-supported publishers also face significant challenges from malvertising. To save money and make detection less likely, attackers typically turn to small ad providers to buy impressions (displays of the ad on a web page), often for as little as $0.20 (USD) or less per 1,000 impressions. At such prices, most of these impressions are likely to end up on low-end sites, but from time to time they can “trickle up” to larger ad networks that serve popular websites. These networks use browser cookies to build profiles for each visitor, making educated guesses as to their age, gender, location, and other details. When a visitor to a typical large advertising-supported website loads a page, the ad network serving the site tries to match the visitor up with an ad that’s compatible with their profile, drawn from one of the many providers that supply ads to the network. Ad providers are usually smaller networks themselves, and these may try to resell the bid further down the chain. Once in a while, a malvertisement from a low-cost ad provider is the highest-bidding ad matching the visitor’s particular profile, and gets served to the visitor.

EXPLOIT KITS

An exploit kit is software that automates the identification and exploitation of vulnerabilities in a victim's computer (typically via their web browser) to then deliver a malware payload to the target machine. Exploit kits have become a popular method for client-side infection and were responsible for the majority of client-side exploits researched by Trustwave in 2015.

"Exploit kits were responsible for the majority of client-side exploits in 2015."

Exploit Kit Techniques

Exploit kits provide an illustrative example of the commercialization of computer crime over the past decade. Kit developers combine exploits for popular browsers, browser plugins, and other components together with tools to distribute and use them, and sell access to the kit to criminals in underground forums and similar venues. The developers compete with each other to offer customers the highest rates of successful infection, with the most successful kits commanding premium prices in the marketplace. Exploit kits make it possible for criminals to infect computers without having the technical sophistication necessary to develop and deploy exploits on their own, simply by purchasing access to a kit and configuring it to meet their needs.

Exploit kits work by generating special web pages, called landing pages, which contain exploits for vulnerabilities in popular browsers, browser plugins, and other software. When a prospective victim loads a landing page using a computer that is vulnerable to one of the exploits, the computer can get infected. Customers can typically configure the kit to target computers based on specific criteria, such as:

- Geographic region
- Specific operating system or browser versions
- The presence or absence of specific browser plugins
- The presence or absence of specific anti-malware software or other security countermeasures.

Exploit kit makers face a running battle with security professionals, and design their kits to minimize the likelihood of exploit attempts being detected and blocked by anti-malware software or IDS/IPS solutions. Popular kits compete with each other to provide access to the newest exploits before patches are widely available. Some security products recognize exploit attempts by monitoring web traffic for characteristic URL patterns used by landing pages generated by different kits, so kit makers frequently change these markings. Some of the more advanced kits include features that attempt to detect when they are executing in a virtualized or laboratory environment, which can indicate a security gateway product or research lab, and avoid running in such environments.

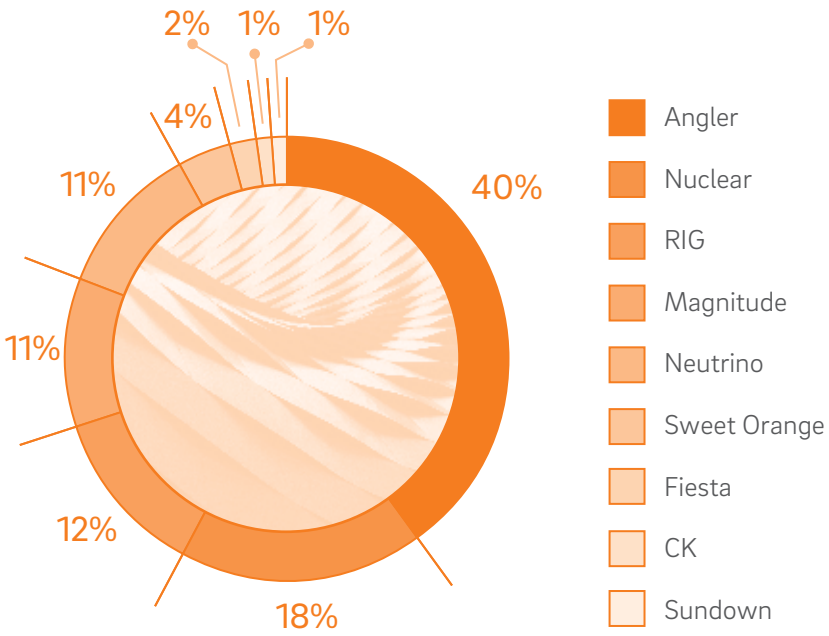
Exploit Kit Trends

Through the blocking of millions of malicious URLs each year, Trustwave is able to identify the trends in exploit kit incidents. In 2014, the biggest stories in the world of exploit kits were RIG and Nuclear, which together accounted for almost half of the incidents we observed (RIG at 25 percent, Nuclear at 23 percent). Angler, Fiesta, Magnitude, and Neutrino accounted for most of the other incidents.

In 2015, the exploit kit landscape shifted significantly. Angler, the most prevalent kit of 2015, accounted for 39.7 percent of exploit kit-related incidents we observed, more than twice as many as the next most prevalent kit, Nuclear. Behind Nuclear were RIG, Magnitude, and Neutrino, which each accounted for between 11 and 12 percent of incidents, followed by several other kits with much smaller percentages. In fact, all of the major kits from 2014 decreased in prevalence except for Angler, with Fiesta enduring the greatest fall from grace, from 13 percent of incidents in 2014 to less than 3 percent in 2015.

"Angler accounted for 39.7 percent of exploit kit-related incidents in 2015."

EXPLOIT KIT PREVALENCE IN 2015



Exploit Kits in 2015: Angler

In 2014 we characterized Angler as an up-and-coming exploit kit that was becoming popular among criminals due to its use of fresh exploits and innovative obfuscation techniques. This approach paid off in a big way for Angler's developers, as its share of incidents more than doubled in 2015 to make it the most prevalent exploit kit of the year.

Angler was the most advanced exploit kit available on the black market in 2015. It was the first kit to adopt techniques like Diffie-Hellman encryption and domain shadowing, as discussed earlier in the report. Angler also frequently changes the URL structure used for its landing pages and other resources, making it more difficult for security researchers and anti-intrusion products to identify Angler-related traffic. Angler's URL patterns changed at least five times during 2015, more than any of its competitors.

Aside from its technical sophistication, Angler was also the first exploit kit to integrate several newly disclosed exploits, including four zero-day exploits and seven "one-day" exploits, which target vulnerabilities for which patches have been released but have not yet been widely distributed. The lifetime of a high-value exploit can often be measured in days, as the effectiveness of newly disclosed exploits drops significantly as computers receive the software patches that address the underlying vulnerabilities. An exploit kit that develops a reputation for offering high-value exploits before its competitors can therefore gain a dominant position in the marketplace, as Angler has done.

SIGNIFICANT EXPLOITS ADOPTED FIRST BY ANGLER IN 2015

CVE	Product or component affected	CVSS v2 severity
CVE-2015-0310 (zero-day)	Adobe Flash Player	10.0
CVE-2015-0311 (zero-day)	Adobe Flash Player	10.0
CVE-2015-0359	Adobe Flash Player	10.0
CVE-2015-1671	Microsoft Silverlight	9.3
CVE-2015-2419	Microsoft Internet Explorer	9.3
CVE-2015-3090	Adobe Flash Player	10.0
CVE-2015-5119 (zero-day)	Adobe Flash Player	10.0
CVE-2015-5122 (zero-day)	Adobe Flash Player	10.0
CVE-2015-5560	Adobe Flash Player	10.0
CVE-2015-7645	Adobe Flash Player	9.3
CVE-2015-8446	Adobe Flash Player	9.3

Angler is known to have a relatively large number of exploit developers contributing to the kit relative to its competitors, although in many cases they simply take exploits that have been developed by others and repackage them with their own obfuscation and evasion techniques.

Exploit Kits in 2015: The Next Four

Following Angler, four other exploit kits attained significant levels of prevalence in 2015: Nuclear, which was involved in 18 percent of observed incidents; RIG, with 12 percent; Magnitude; with 12 percent; and Neutrino, with 11 percent. Although these kits generally did not exhibit the same level of technical innovation as the more expensive Angler, they were usually quite good at integrating newly discovered exploits and adopting new obfuscation techniques. Because of the premium that newly discovered exploits command, an exploit that appears in one kit usually makes its way to the others within a few days or even hours—in fact, developers working on one kit often simply copy exploit code from another and add their own obfuscation to it. For example, after Angler added a zero-day exploit for the Flash Player vulnerability CVE-2015-5122, the other four major kits all added the same exploit within four days. In a few cases, in fact, Angler was forced to play catch-up after one of the other kits debuted a new exploit or technique—which Angler then copied.

Nuclear, first observed in 2009, is the oldest of the five most prevalent kits of 2015. Though not the most sophisticated kit, it has been stable over the years and tends to provide good results for its customers. Like Angler, Nuclear targets a range of popular browser plugins, including Flash Player, Microsoft Silverlight, and the Adobe Reader plugin, in addition to Internet Explorer. Nuclear was the first kit to implement an exploit for CVE-2015-0336, a Flash Player vulnerability that was patched a few days earlier.

RIG was the most prevalent kit we observed in 2014, but slipped to third place in 2015. RIG took a significant hit in February 2015 after a disgruntled reseller leaked part of its source code and database, which gave security researchers a valuable look at not only the kit's technical infrastructure but also its business model. A few months later, the RIG authors introduced version 3.0 of the kit, which included a number of security updates, infrastructure changes that point to a discontinuation of the reseller model, and an updated administration interface.

Our observation of two RIG instances suggest that the RIG team has recovered quite well from the source code leak, with RIG 3.0 successfully infecting around 27,000 machines per day in the first few months after release. About 90 percent of traffic to RIG landing pages came from malvertisements (see the "Malvertising" sidebar in the "Trends in Exploitation" section for more information).

Neutrino depends less on malvertising to spread than the other kits and more on compromising popular web applications. Thousands of sites were compromised by Neutrino landing pages in two separate incidents in 2015, one involving WordPress sites and the other involving the Magento e-commerce platform. Recently, Neutrino has begun to adopt a technique previously employed by the Flashpack exploit kit in which all of the malicious code on the landing page is bundled into a single Flash file that handles all of the checks that landing pages typically perform, such as verifying client-side plugin versions and choosing which exploits to deliver. This makes the landing page harder for security products to recognize and makes it more likely that the exploit will succeed.

Magnitude uses an unusual traffic-sharing business model, whereby its customers trade a percentage of their traffic (up to 20 percent) with Magnitude's owners rather than paying to rent the kit directly. The Magnitude owners then use the traffic for their own criminal activities, usually involving infecting machines with ransomware. This arrangement has earned it a small but significant exploit kit market share, which remained consistent from 2014 to 2015.

Exploit Kits in 2015: The Rest

Angler, Nuclear, RIG, Magnitude, and Neutrino together accounted for more than 90 percent of the incidents we observed, with the remainder being split between **Sweet Orange, Fiesta, CK,** and **Sundown**. Of these, Fiesta was the only kit to have a significant impact in 2014. Fiesta-related incidents fell from 14 percent that year to less than 3 percent in 2015.

Exploit Kits: Constantly Evolving

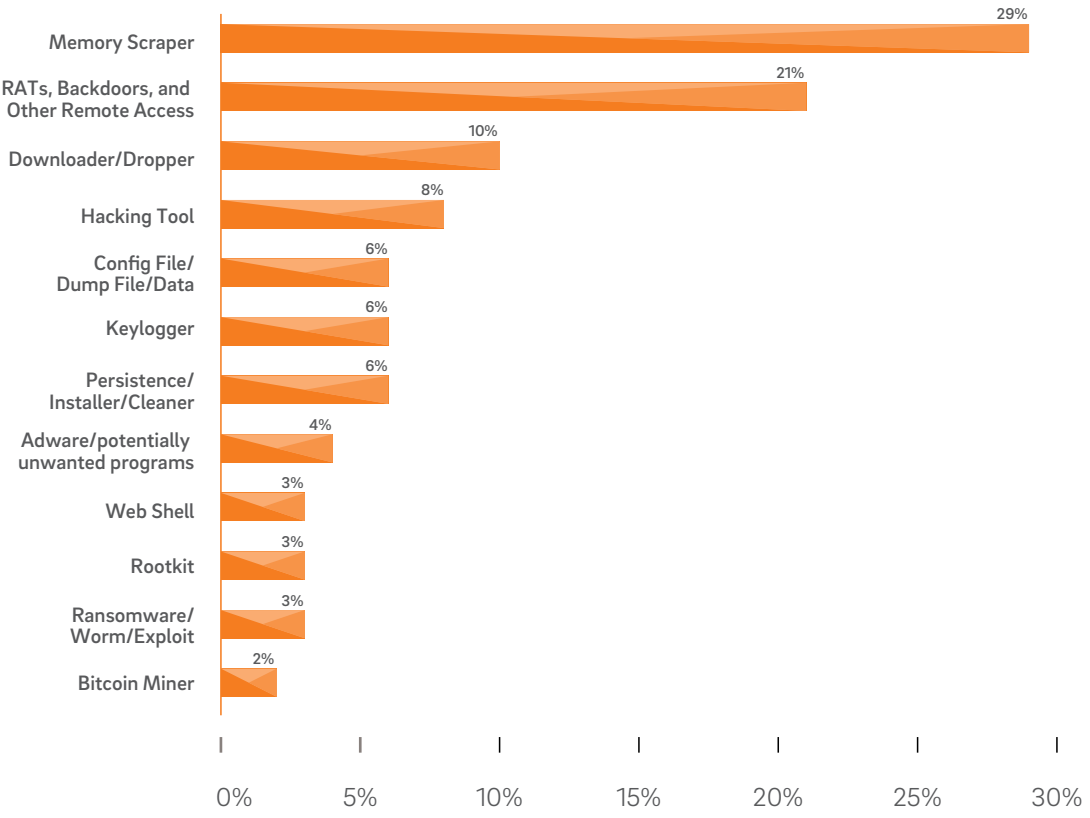
It's tempting to try and extrapolate these figures to predict what the exploit kit landscape will look like in the future, but such an exercise would probably be unwise. The marketplace is very competitive, and the kit that is on top today might be all but gone from the market tomorrow. The only prediction we can make with any certainty is that exploit kits, which put tools of considerable technical sophistication in the hands of criminals who have financial resources but limited technical ability, are likely to remain a significant challenge for security professionals for some time to come.

MALWARE

As part of Trustwave incident response engagements and collaborations with law enforcement, Trustwave conducts deep analysis and reverse engineering of malware. Most of the information presented in this section comes from investigations of compromise incidents affecting point-of-sale (POS) environments, as well as the specialized equipment used to collect payment card data from customers. As a result, most of the malware families discussed here are tailored specifically for stealing and exfiltrating data from POS systems, although some malware samples were also collected from general purpose computers in such environments.

Malware Types Encountered Through Data Compromise Investigations

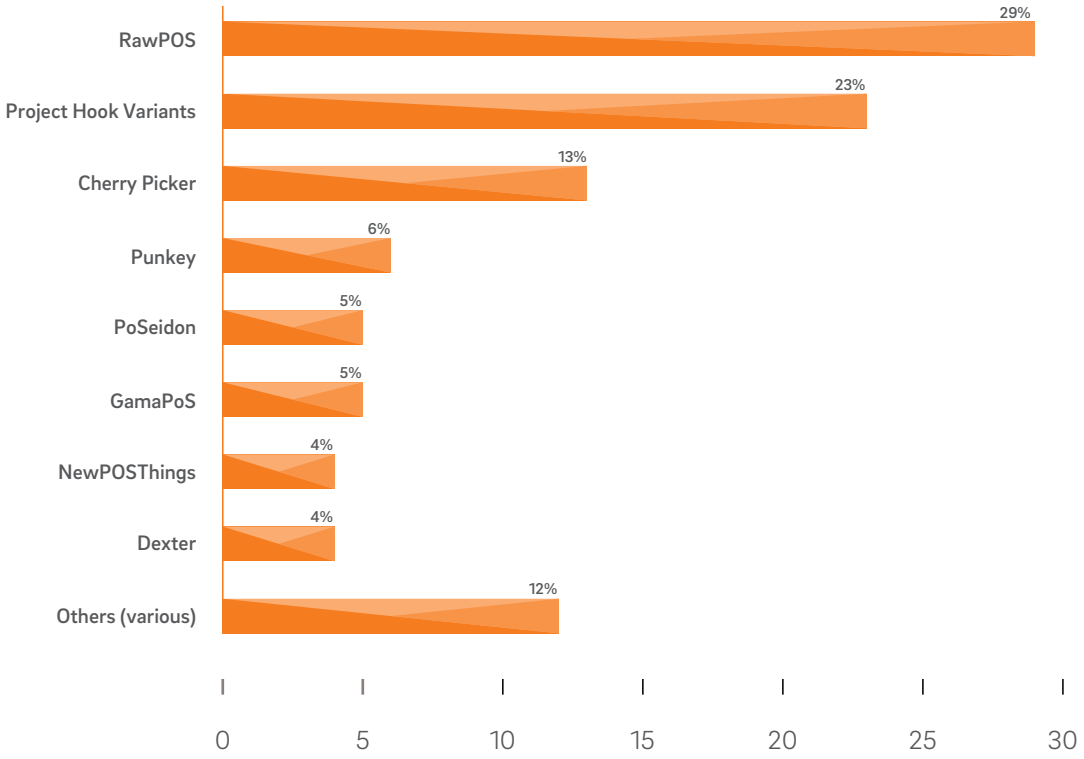
TYPES OF MALWARE ENCOUNTERED DURING INVESTIGATIONS



POS attackers tend to make heavy use of memory scrapers, which accounted for the largest share (29 percent) of the malware encountered during the incidents we investigated. A memory scraper is a simple program that searches the computer's memory for sequences of data that match particular patterns, such as a credit card number. POS terminals and other computers usually encrypt payment card data when storing and transmitting it, so attackers often use scrapers to locate card numbers in memory before they are encrypted, or after they are decrypted for processing. Most memory scrapers are very basic, although some make use of features such as whitelisting and blacklisting to focus the processes and memory address they scan, to save time and reduce false positives. Many scrapers lack remote communication functionality of their own, so the attacker must exfiltrate the data in another way, such as by remote administration tools, backdoors, or other remote access methods.

Malware Families Targeting POS Systems

POS MALWARE FAMILIES ENCOUNTERED DURING INVESTIGATIONS



The POS malware ecosystem continues to fluctuate rapidly. Several of 2014's prominent families, like Backoff and Alina/Spark, lost ground in 2015 and were replaced by several new families, as well as a few older ones. Most of the top families are simple memory scrapers that search for card stripe data, sometimes dumping the contents of RAM to disk before reading it. They are ineffective against the newer chip-and-PIN standard used by most card processors when the standard is implemented properly.

RawPOS

RawPOS is a memory scraper that has been around for several years but increased in prevalence dramatically in 2015 in connection with an apparent campaign targeting the hospitality industry. RawPOS is a fairly unsophisticated scraper that includes blacklist/whitelist functionality to help it focus its memory searches. It uses simple XOR encryption to cover its tracks.

Project Hook

Another large share of the POS malware we observed involved variants based on Project Hook, a POS family that has been around for several years. At one time advertised for sale for $1,000 (USD) on underground forums, it has since been customized into a number of different variants by attackers. Project Hook variants use the open-source UPX packer for obfuscation, and do not use encryption.

Cherry Picker

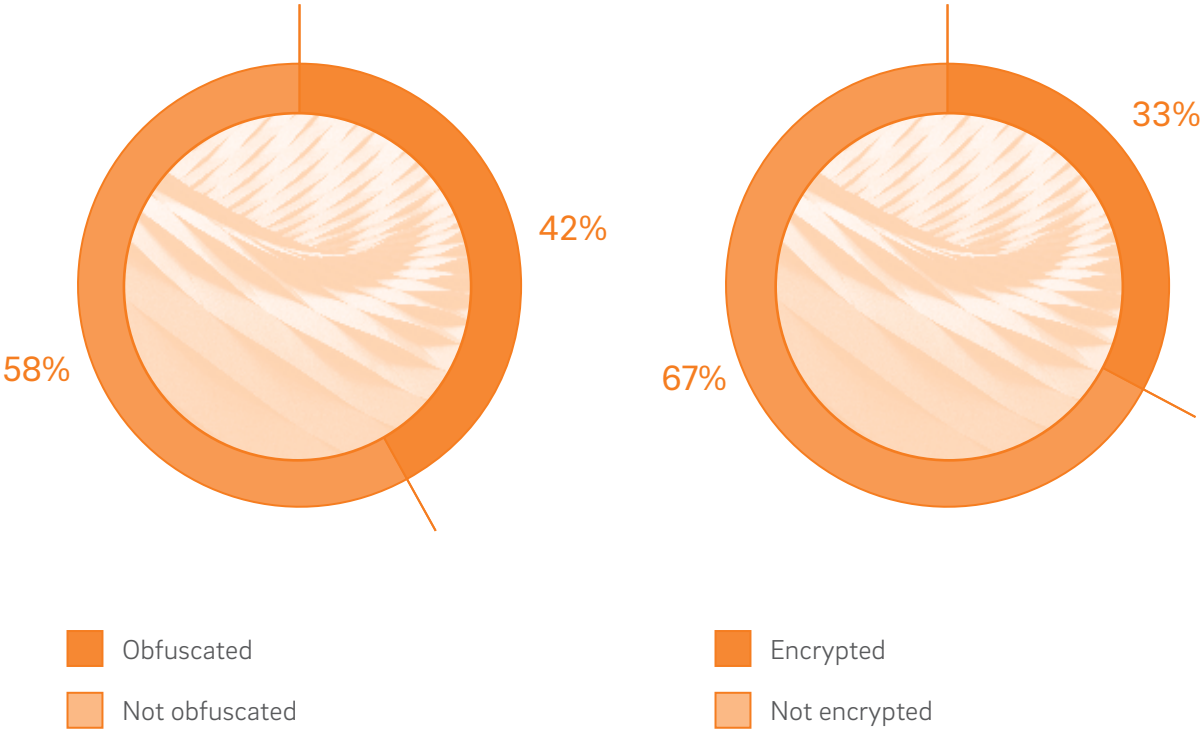
The Cherry Picker scraper was first detected more than five years ago, then went inactive for several years before a new version emerged in 2015. Cherry Picker uses a new memory scraping algorithm, a file infector for persistence, and cleaner malware that removes all traces of the infection from target systems. This sophisticated functionality and highly targeted victims have helped the malware remain under the radar of many anti-virus and security companies.

Punkey

Trustwave researchers discovered and named the Punkey family in April 2015. It appears to have evolved from the older NewPOSthings family, but differs from it in several significant ways. Unlike some of the other families, Punkey is a full-featured malware family that employs process injection and tries to connect to a command-and-control (C&C) server for communication with the attacker.

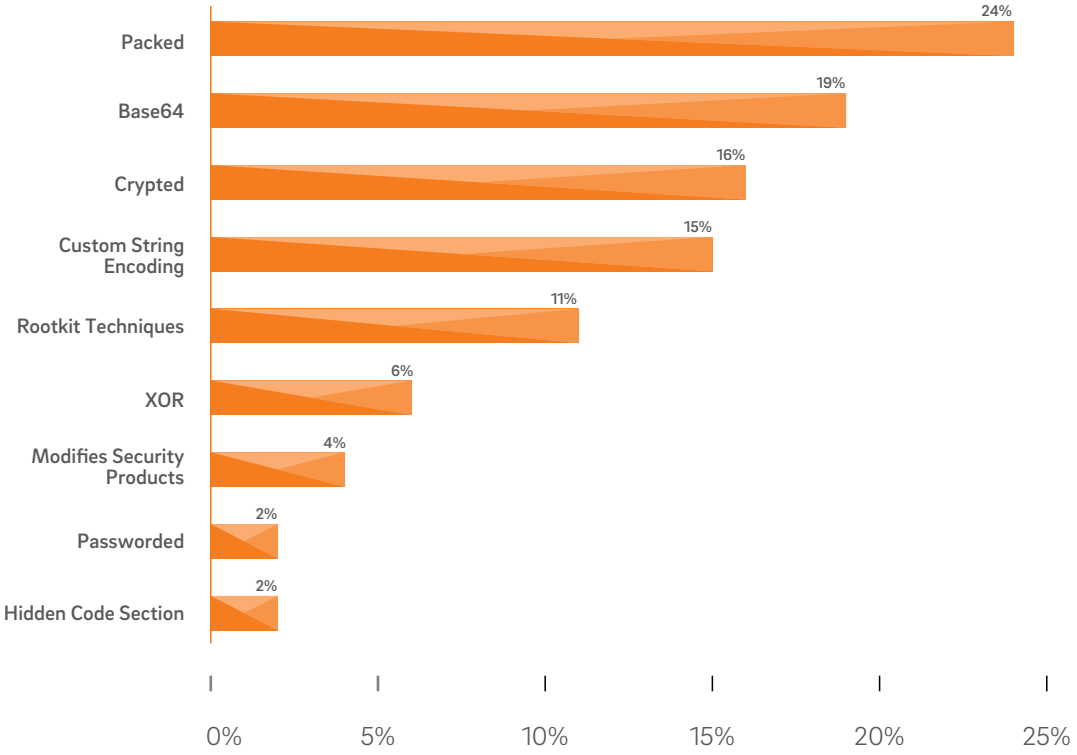
Malware Obfuscation and Encryption

Malware developers often use obfuscation and encryption techniques. In general, malware authors use obfuscation to attempt to hide the true nature of their code's functionality from security tools. They use encryption to hide the data they save to disk and/or send outside the victim network so that any monitoring controls would not flag the data or related communications.



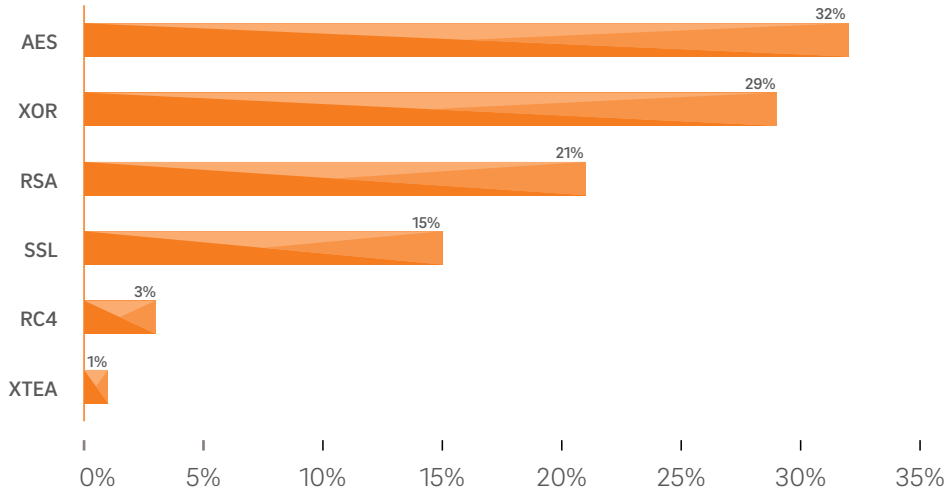
More than half of the malware samples encountered in the incidents we analyzed did not use any method of obfuscation at all, and two-thirds used no encryption. In many cases, these functions are performed by other components that are detected as different malware families. Malware that includes every desired function in a single executable can be more compact and easier to deploy than multi-component installations, but it's also easier for security tools to identify and fingerprint. We observed many single-purpose programs executing a simple technique or task, such as dumping memory to disk, and then quitting. These tools, used in succession, can be harder to detect than if packaged together in a single program.

OBFUSCATION TECHNIQUES USED



Of the samples that did use obfuscation, most used fairly simple techniques such as packing (which prepares an executable for distribution, optionally using compression) and Base64, an encoding scheme that translates binary data to ASCII text.

ENCRYPTION TECHNIQUES USED

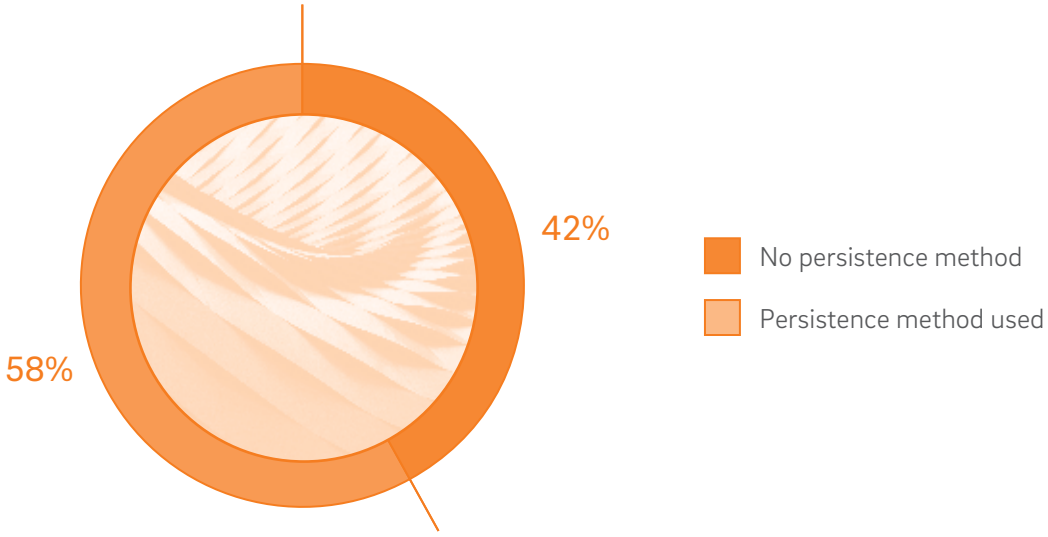


For the first time, malware using strong Advanced Encryption Standard (AES) encryption accounted for a larger share than malware using the trivial XOR method. Both Cherry Picker and Punkey use AES, which helps explain the encryption technique's rise.

Malware Persistence

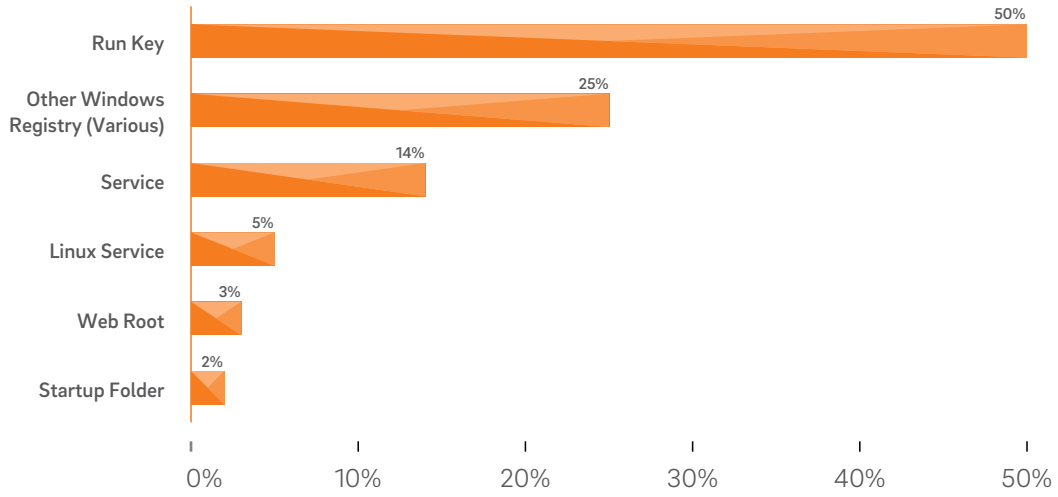
Persistence is a computer program's capability to continue to run after a system is restarted. If a piece of malware does not include persistence capabilities, a reboot of the infection system would effectively disable the attack until it was manually restarted.

PERSISTENCE



As with obfuscation and encryption, most of the samples we observed did not include any facility for persisting after a reboot. In most cases, persistence with these families would be handled by other malware components, or by the attacker interacting with the system remotely.

METHODS OF PERSISTENCE

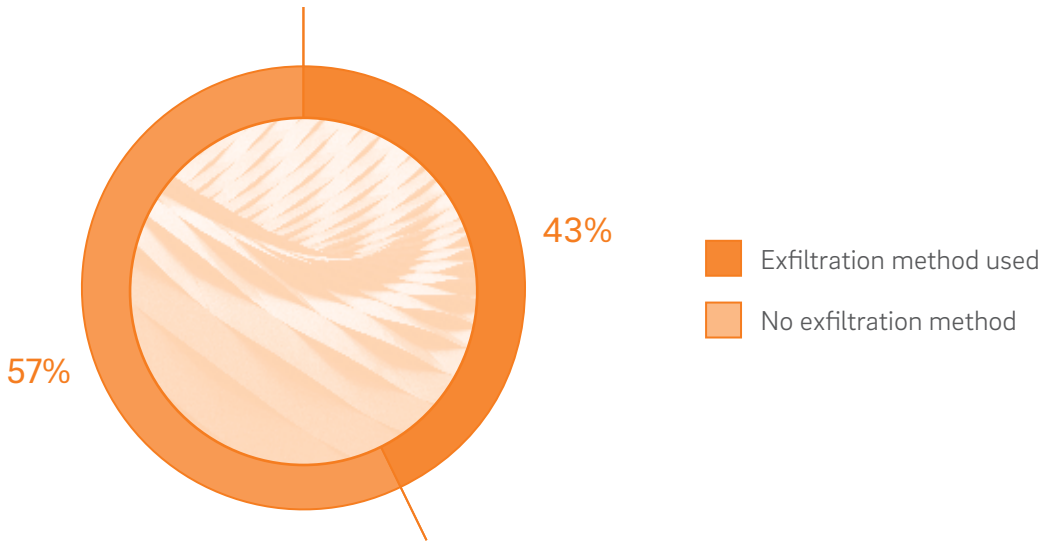


About half of the samples that employed persistence methods did so by adding themselves to the Run key (HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run and HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run) in the Windows registry, which causes Windows to execute the program as it starts up after a reboot. Most of the rest used other keys in the Windows registry, or Windows or Linux services.

Malware Exfiltration

Exfiltration is a malware feature that automates the sending of harvested victim data, such as login credentials and cardholder information, back to an attacker-controlled server.

EXFILTRATION



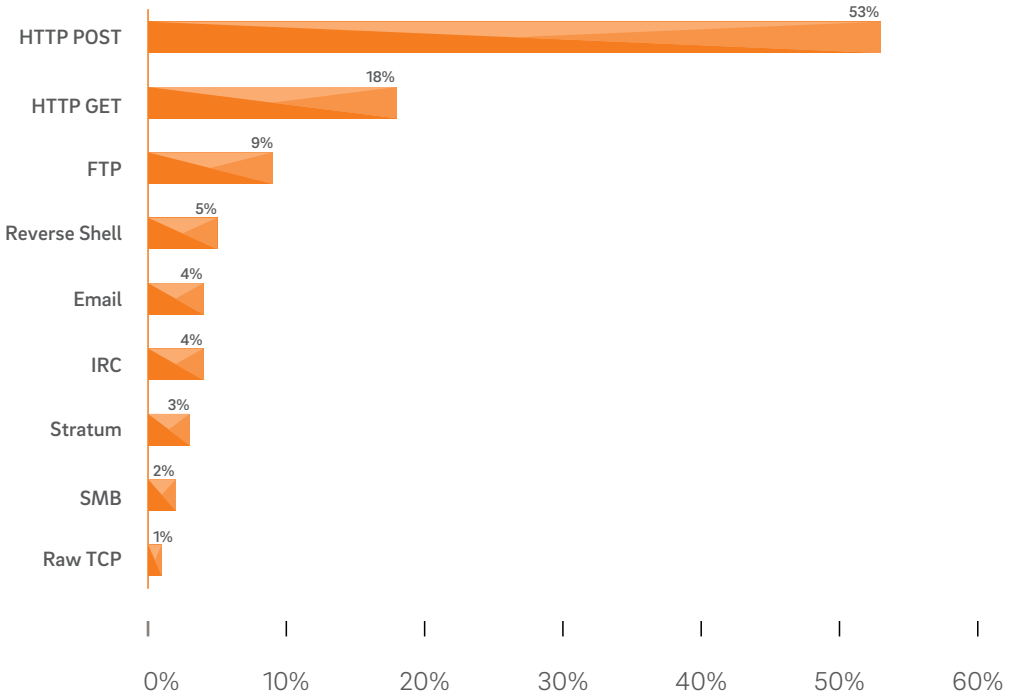
More than half of the malware encountered did not use any exfiltration method. Not all malware authors choose to implement exfiltration because it can provide a trail that might help investigators identify the source of the malware. In these cases, the attacker typically connects to the computer remotely to exfiltrate the data. In other cases, exfiltration is handled by another malware component.

PENETRATION TESTING TOOLS BECOME PENETRATION TOOLS

One interesting observation in 2015 was the malicious use of ethical hacking tools – the kind that penetration testers use to detect vulnerabilities while securing systems. In particular, the use of malicious PowerShell scripts as computers and devices running Windows XP are replaced by machines running Windows 7 or later, which ship with PowerShell installed by default. Some of the examples we observed in 2015 include:

- FrameworkPoS is a well-documented family of malware that targets POS systems and has been attributed to at least one high-profile retail breach. A new FrameworkPoS variant discovered by Trustwave SpiderLabs researchers in late 2015 includes two PowerShell scripts for execution, one of which comes from an open-source penetration testing framework called PowerSploit. The new FrameworkPoS variant uses a slightly modified version of the Invoke-ReflectivePEInjection cmdlet from PowerSploit to base64 encode the FrameworkPoS DLLs and inject them into memory.
- Another POS environment breach used Meterpreter, a generic malware payload from the Metasploit penetration testing framework, to open a reverse shell for interactive access into the compromised system.
- In a breach of a hotel chain, the attackers used psexec, a remote process execution utility from the Windows Sysinternals tool suite, to execute malware on remote hosts.
- A breach of a restaurant chain used nircmdc, a command-line utility from Nirsoft, and Sysinternals psexec and pskill for interactive remote access into a compromised system.
- In a breach of a supermarket chain, the attackers used the FGdump tools pwdump and cachedump64 to dump passwords from compromised Windows systems.

MALWARE EXFILTRATION METHODS



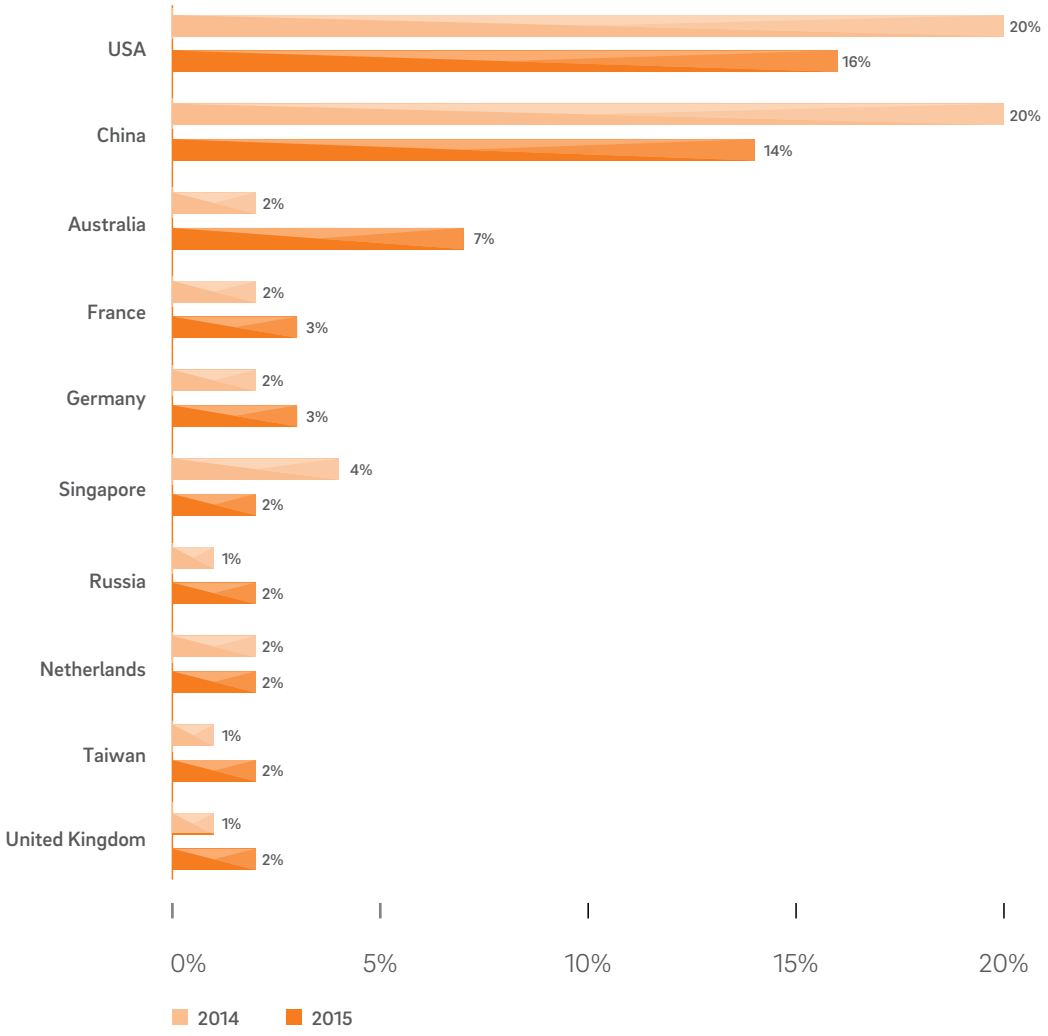
More than half of all malware we encountered that included exfiltration capability extracted data over HTTP (including HTTPS), the same protocol used to view web pages. Malware authors prefer to use this protocol for data heists so they can hide stolen assets in plain sight among normal web traffic. Most of these use the POST request method to transmit information to the attacker, but a significant subset of the samples use the GET method, which requires appending the stolen data to the end of the URL being requested as a (potentially long) query string.

SUSPICIOUS TRAFFIC AND ALERTS

Monitoring suspicious activity into, out of, and within a network provides an important first line of defense against attack. For the data in this section, we've inspected a sample of firewall, network and host IDS alerts from business customers in the Asia-Pacific region. Most of these alerts involve suspicious inbound or outbound traffic, although a small percentage involve suspicious behavior on a local computer.

Sources of Suspicious Traffic

TOP 10 SOURCES OF SUSPICIOUS TRAFFIC



IP addresses in the United States and China were the most common sources of suspicious traffic analyzed at the Trustwave Singapore SOC in both 2014 and 2015 by a large margin. This does not necessarily suggest that the attackers themselves were located in these countries. Criminals typically attack from botnets and other compromised computers to hide their real locations and take advantage of large amounts of bandwidth. The prominence of the United States and China is most likely due to the large numbers of computers there. Australia, France, and Germany were also significant sources of traffic, probably for related reasons. Although cybercrime tends to ignore geographic borders, the prominence of Singapore as a source of suspicious traffic (third in 2014, sixth in 2015) shows that there are still a large number of attacks that keep close to home. The increase in traffic from Australia, from 2 percent in 2014 to 7 percent in 2015, may also be at least partially influenced by geographic proximity to Singapore.

Types of Suspicious Traffic

MOST COMMON ALERTS ANALYZED BY THE SINGAPORE SOC IN 2015

Description	Category	Percent of whole
Automated tools used to crawl websites and obtain server information	Other reconnaissance	18.84%
Request for active scripting (pl, cgi, php extension)	Other reconnaissance	5.51%
Inbound traffic allowed from known bad address	Potential exploitation	5.05%
Potential UDP DDoS	Potential DDoS	3.85%
Host-based alert	Potential exploitation	3.52%
Outbound traffic allowed to malicious web pages	Potential exploitation	2.40%
Outbound traffic allowed to known bad address	Potential exploitation	2.23%
Too many blocked alerts targeting the same hosts	Potential exploitation	1.87%
Privileged database operation triggered	Potential database injection	1.71%
Excessive login attempts to a database	Bruteforce	1.66%
Web page access that returned directory listing or verbose debug page	Other reconnaissance	1.36%
Excessive login attempts	Bruteforce	1.22%
Overly long SQL query (>2MB)	Potential database injection	1.10%
SQL response returns more than 10,000 records	Potential database injection	0.76%
OpenSSL Heartbeat exploit	Potential exploitation	0.70%
Potential SQL injection targeting a website	Potential database injection	0.47%
PHP include() function exploited for code execution	Potential exploitation	0.45%
Admin activities on authentication server	Malicious administrative actions	0.39%
Email with executable attachment	Potential exploitation	0.29%
Total		**79.94%**

MOST COMMON ALERTS ANALYZED BY THE SINGAPORE SOC IN 2014

Description	Category	Percent of whole
Numerous dropped/deny firewall hits from a single source	Potential port scan	24.82%
Automated tools used to crawl websites and obtain server information	Other reconnaissance	18.84%
Request for active scripting (pl, cgi, php extension)	Other reconnaissance	6.70%
Potential detected port scan dropped at firewall	Potential port scan	5.51%
Host-based alert	Potential exploitation	3.85%
Potential malicious user administration (e.g., enabling a Guest account or promoting a regular user to Admin status)	Malicious administrative actions	3.52%
Potential exploit targeting OpenSSL on firewall	Potential exploitation	2.40%
Remote user login to a Windows system	Suspicious remote access	2.23%
Outbound traffic allowed to malicious web pages	Potential exploitation	1.87%
Inbound traffic allowed from known bad address	Potential exploitation	1.71%
Attacker trying to execute shell code on an internal web server	Potential exploitation	1.36%
PHP include() function exploited for code execution	Potential exploitation	1.22%
XPath injection attempt (very similar to SQL injection but targeting a different database type)	Potential database injection	1.10%
Potentially malicious administrative actions	Malicious administrative actions	0.76%
Potential breach due to single IP logging into multiple systems	Suspicious remote access	0.70%
UDP port scan	Potential port scan	0.47%
OpenSSL Heartbeat exploit (potential exploitation)	Potential exploitation	0.45%
Potential bruteforce attack on SQL server	Bruteforce	0.39%
Potential attempt to hijack a VPN session	Potential exploitation	0.29%
Total		**78.18%**

Analyzing the top alerts received in 2014 and 2015 and grouping them into categories shows that most suspicious traffic involved port scans and other reconnaissance activities, such as probes of web servers seeking information about scripting environments and back-end infrastructure. Other reconnaissance activity is widespread and common and has been for some time. It is typically conducted with automated tools and requires little effort on the part of the attacker. However, most targeted attacks start with reconnaissance, so although these activities may be commonplace on the network, it's important not to ignore such traffic, even if it's considered low priority.

Potential exploitation efforts accounted for the third-most common category of suspicious traffic in both years. This includes traffic to and from known malicious IP addresses and URLs, suspicious file changes, attempts to exploit known vulnerabilities in software and hardware, and a range of other traffic patterns suggestive of exploitation.

Attacks on data using SQL and XPath injection attempts accounted for a small percentage of suspicious traffic during both years. Injection flaws have been some of the most dangerous vulnerabilities affecting websites for more than a decade, according to the Open Web Application Security Project (OWASP). SQL injection in particular can pose a significant threat to sensitive data like personally identifiable information (PII) and credit card data. It can be hard to prevent and relatively easy to exploit. XPath injection, though not as well known, can also be used by an attacker to steal sensitive information stored in XML files and not normally exposed to the public. Installing a web application firewall (WAF) in front of web servers can help block injection attacks and other threats.

Other categories each accounted for a small percentage of traffic each year, including brute force password cracking attempts, potential distributed denial-of-service (DDoS) attempts, and malicious administrative actions.

SECTION 3

THE STATE OF SECURITY

In this section, we take stock of the state of affairs in 2015 for databases, networks and web applications – three of the most important components of the worldwide data infrastructure, and three of the biggest targets for cybercriminals. Where are they vulnerable? What's the risk? How good have software vendors, businesses, and end-users been at staying safe from attack?

In "Network Security," we discuss what we learned in 2015 from our network vulnerability scanning systems, which probe client systems both within their own networks and from outside, looking for insecure configurations and other vulnerabilities. "Database Security" looks at the security patches issued by major database vendors in 2015, and what problems they were designed to solve. "Application Security" highlights the results of our security testing services, which use penetration testing to circumvent security controls – with permission from our clients – and gain access to target systems. We believe that understanding these findings can help businesses get a better understanding not only of where their weak points are, but where to look for them in the future.

NETWORK SECURITY

Our 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. The figures given for each vulnerability indicate the percentage of all vulnerability detections by our scanner that could be attributed to that vulnerability. For example, 10.17 percent of the vulnerability detections we recorded in 2015 could be attributed to the "SSL Vulnerable to CBC Attacks" vulnerability.

TOP FIVE VULNERABILITIES BY OCCURRENCE

Occurrence	Name
10.17%	SSL Vulnerable to CBC Attacks (CVE-2011-3389)
9.23%	SSL/TLS Weak Encryption Algorithms (CVE-2013-2566; CVE-2015-2808)
8.45%	No X-FRAME-OPTIONS Header
5.35%	TLSv1.0 Supported
4.72%	SSLv3 Supported (CVE-2014-3566)

"BEAST Mode" Still the Champ

The most common configuration problem we observed in 2015, accounting for 10.17 percent of occurrences, involved servers that accept block-based ciphers with SSL versions 2.0 or 3.0 or TLS version 1.0. This could lead to attacks such as "Browser Exploit Against SSL/TLS" (BEAST) that takes advantage of these weaknesses to allow an attacker to decrypt the SSL/TLS connections. (By definition, all of the servers reporting this vulnerability were also counted separately for supporting at least one of the older, less secure SSL/TLS protocols, discussed below.)

That this remains the top issue four years after the BEAST attack was initially disclosed in 2011 is worrisome, although new industry standards implemented in 2015 are likely to force many businesses to drop support for the protocols that give rise to this vulnerability in the near future, as explained below.

"Four of the five network vulnerabilities detected most often resulted from insecure server configurations for Secure Socket Layer (SSL) and Transport Layer Security (TLS)."

Taking Out the POODLE

One of the more significant developments around SSL/TLS security in 2015 had more to do with policy than with technology. In April, the Payment Card Industry Security Standards Council (PCI SSC) published version 3.1 of the PCI Data Security Standard (PCI DSS), which restricted the definition of strong encryption to exclude all versions of SSL and version 1.0 of TLS. Henceforth, businesses and other organizations that handle payment cards would be required to use TLS version 1.1 or later for secure communications to remain in compliance with the standard. The change came in the wake of the October 2014 disclosure of CVE-2014-3566, a significant weakness in SSL 3.0 that researchers dubbed "Padding Oracle on Downgraded Legacy Encryption," or POODLE. A successful exploit takes advantage of a padding oracle attack against the cipher block chaining (CBC) encryption mode in SSL to capture a session cookie and hijack the encrypted SSL session. A variant of the POODLE attack that affects TLS 1.0 was announced a few months later. The POODLE disclosure was widely perceived as sounding the death knell for SSL 3.0, as all the major browser makers announced that they would drop support for the protocol. (Earlier versions of SSL have been deprecated for several years.)

In response to the change in PCI DSS, we updated our scanner engine to report support for SSL 3.0 and TLS 1.0 as an automatic compliance failure, and these configuration problems accounted for the fourth and fifth most common vulnerability scan findings in 2015, with TLS 1.0 support accounting for 5.35 percent of occurrences and SSL 3.0 support accounting for 4.72 percent.

Other Top Vulnerabilities

This year we updated our "SSL/TLS Weak Encryption Algorithms" detection logic by adding RC4 and MD5 based ciphers. As a result, this vulnerability accounted for the second highest number of occurrences, at 9.23 percent.

Web servers that don't supply X-FRAME-OPTIONS headers when sending web pages accounted for the third-largest number of occurrences we observed, at 8.45 percent. This HTTP response header is used to indicate whether the browser should be allowed to render the requested page in an HTML frame or inline frame. Servers that don't send this header are vulnerable to clickjacking attacks, in which an attacker fools a victim into clicking in a partially hidden or obscured frame for various purposes.

TOP FIVE VULNERABILITIES RELEASED IN 2015 BY OCCURRENCE

Occurrence	Name
8.18%	SSL RC4-based Ciphers Supported (“Bar Mitzvah” attack)
4.93%	SSL/TLS Weak Encryption Algorithms (“Bar Mitzvah” attack)
0.41%	Diffie-Hellman key exchange (“Logjam” vulnerability)
0.05%	Vulnerability in HTTP.sys Could Allow Remote Code Execution (CVE-2015-1635; MS15-034)
0.05%	Apache HTTP Server Request Smuggling Vulnerability via Invalid Chunk-Extension Characters (CVE-2015-3183)

The most common new vulnerabilities detected by the Trustwave scanner relate to an attack nicknamed “Bar Mitzvah” by researchers, which exploits a vulnerability in the RC4 cipher that was first described 13 years prior to the disclosure of the attack (hence “bar mitzvah,” which Jewish boys undergo at the age of 13). An eavesdropper can exploit the vulnerability against SSL/TLS connections using RC4 encryption to recover up to 100 bytes at the start of an encrypted stream. This can include passwords and other sensitive information.

The third most common new vulnerability in 2015 involved servers that support so-called export-level encryption keys, a legacy of U.S. government restrictions on the export of strong cryptography in the mid-1990s. A man-in-the-middle attacker can exploit a vulnerability dubbed “Logjam” to force an encryption session to use these weak, insecure keys. (See the “High-Profile Threats” section for more.)

Two web server vulnerabilities, one that affects the Apache HTTP server and one that affects Windows Internet Information Services (IIS), round out the list of the top five new vulnerabilities detected by our scanner, at 0.05 percent each.

Addressing Network Vulnerabilities

Given the attention that SSL/TLS weaknesses drew in 2015, it’s unfortunate that they still accounted for as many detected vulnerabilities as they did. Nevertheless, the media interest in vulnerabilities like BEAST, POODLE, and Logjam have helped raise awareness of the issue and spurred businesses to take the steps necessary to protect themselves and their customers. We expect this class of vulnerabilities to be lower next year as businesses continue to implement the new PCI DSS rules regarding TLS, but with full compliance not mandated until 2018, it may be a few years before insecure SSL/TLS configurations drop out of the top five.

DATABASE SECURITY

Most of the 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. Databases hold a treasure trove of assets that is only getting larger as digital information grows at record rates. Examining the vulnerabilities that were patched in several of the more widely used database systems provides insight into the state of database security in 2015.

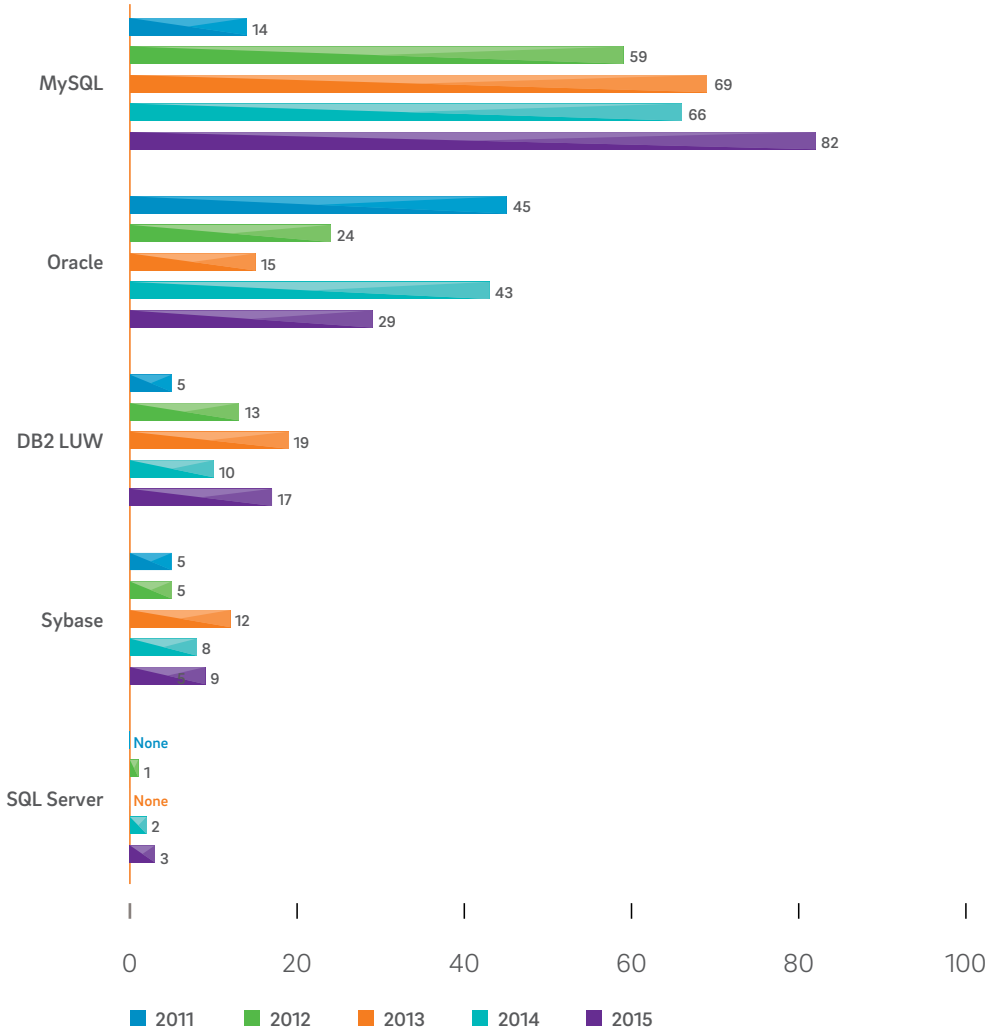
"Databases hold a treasure trove of assets that is only getting larger as digital information grows at record rates."

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 and cause a denial-of-service condition.
- 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 database management system (DBMS) service itself, or any other essential components.
- Default credentials still present an opportunity for abuse by attackers. In our penetration testing engagements, we often find administrator-level accounts with default passwords.

Database Patching, 2011-2015

DATABASE VULNERABILITIES PATCHED, 2011-2015

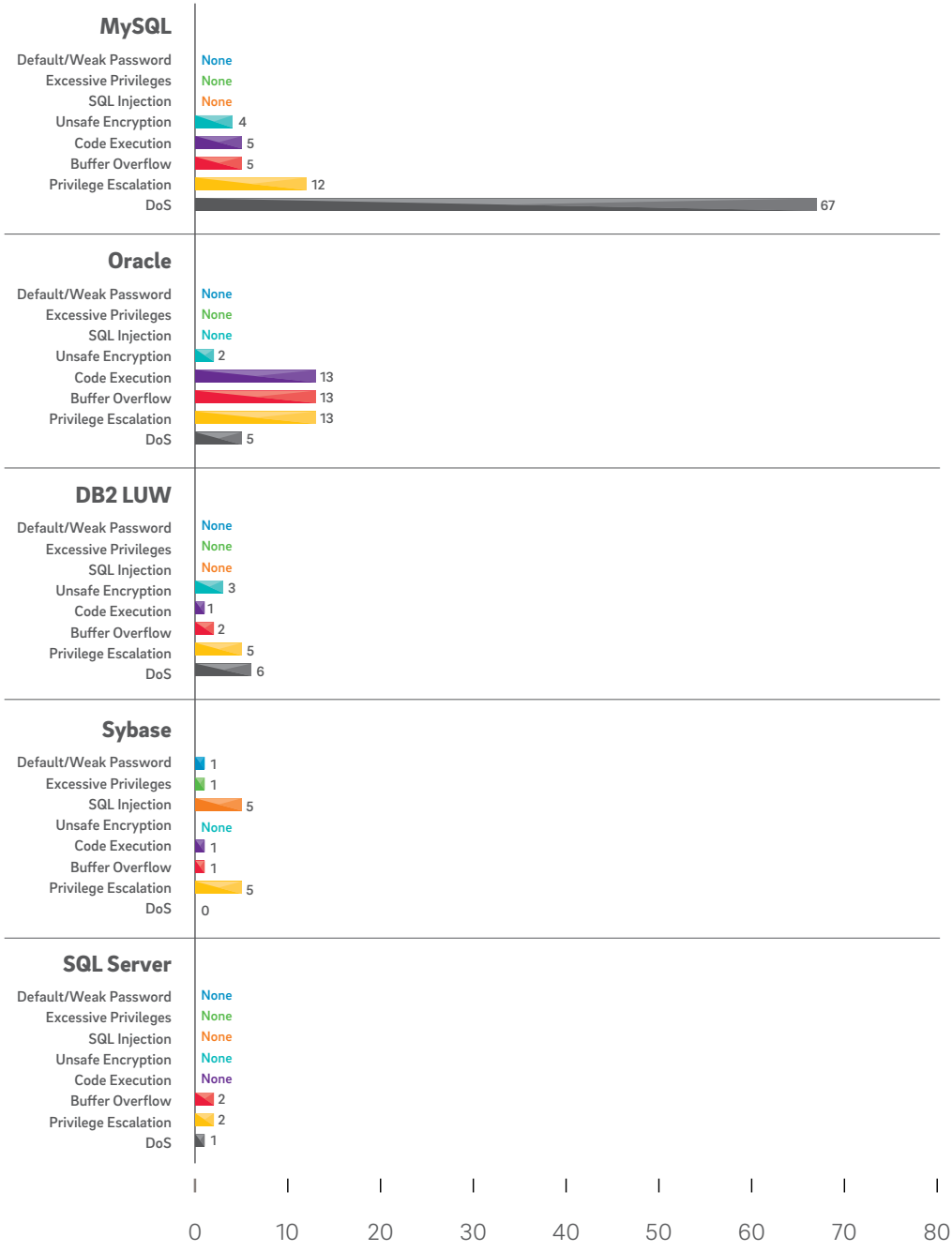


The five database products we analyzed all ranked the same relative to one another as they did in 2014, though the numbers changed significantly for some. In first place was MySQL, which received security patches for 82 vulnerabilities in 2015, more than all the other listed database products combined. MySQL has had more vulnerabilities patched each year since 2012, when MySQL developer Oracle first included the product in its Critical Patch Update cycle. 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 the number is usually more heavily influenced by how much time and effort researchers and other experts expend trying to find vulnerabilities in each product. 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. Generally, every vulnerability that is found and disclosed receives its own CVE identifier, which contributes to the number of distinct vulnerabilities attributed to the product. By contrast, the process for discovering and fixing bugs in proprietary software is much different. Whereas independent researchers can find bugs in open-source software by simply reading the code, they must use techniques like fuzz testing to locate vulnerabilities in closed-source software, which makes them harder to find. 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 the normal testing process, with the fix then being rolled out as part of a routine maintenance release.

Oracle Database received the second largest number of patches in 2015 with 29, down from 43 in 2014. IBM's DB2 and SAP's Sybase were third and fourth with 17 and eight patches, respectively, while Microsoft SQL Server brought up the rear with three vulnerabilities fixed.

Database Patching by Vulnerability

In 2015, all of the major database vendors fixed and released patches for various vulnerabilities. We analyzed the issues fixed in 2015 and categorized them into one or more multiple classes of vulnerabilities per product. (In most cases, the totals for each database exceed the overall 2015 count for that database because some vulnerabilities fall into more than one category and are therefore counted multiple times.)



Denial-of-service (DoS) vulnerabilities accounted for the largest number of vulnerabilities patched, with MySQL responsible for all but a handful of them. 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.

All five databases received patches for privilege escalation and buffer overflow vulnerabilities. Privilege escalation vulnerabilities in databases often involve built-in stored procedures that don't properly validate input, making them vulnerable to a form of SQL injection. Because stored procedures often execute in a different security context than the account that calls the procedure, an attacker who is familiar with the database can sometimes supply SQL statements as parameters to a procedure, allowing the attacker to access data and procedures that would not otherwise be available. Likewise, buffer overflow vulnerabilities often involve supplying invalid data as a parameter to a vulnerable stored procedure, causing memory corruption.

Vulnerabilities that could allow an attacker to execute malicious code, often through buffer overflows, received patches in all five databases, with Oracle Database being the most affected. A handful of bugs involved weaknesses in SSL/TLS encryption that could be exploited to force a session to fall back to a weaker encryption protocol. These are related to the POODLE and FREAK vulnerabilities disclosed in 2014 and 2015. Several other categories accounted for a small number of patches each, all for Sybase.

Keeping Databases Safe

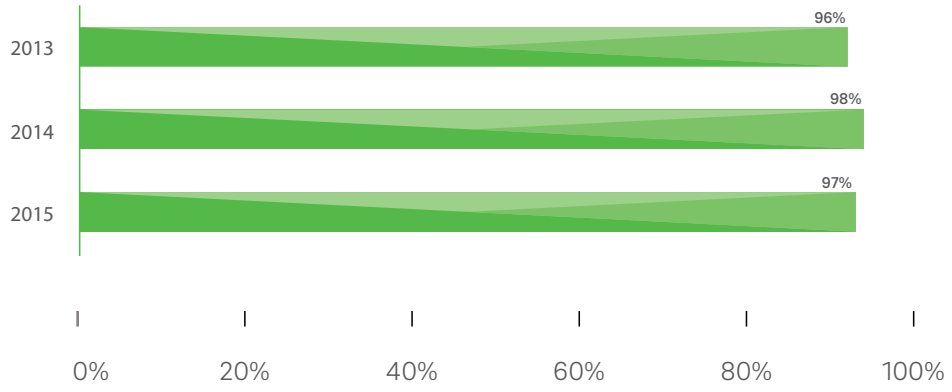
Database security is often taken less seriously than security for high-visibility installations like networks, web servers and applications. Techniques like partitioning are inadequate defenses against attackers who know how to exploit databases to jump between partitions. But protecting databases doesn't have to be difficult. At its core, a database is just another system, and many of the best practices for preventative security apply to databases just as they apply to any other system. Organizations should keep databases up to date with all the latest patches, watch out for insecure configurations, disable or remove any unused add-ons, and pay attention to important lifecycle dates. Premier Support for Oracle 11gR2 ended on January 31, 2015, and Microsoft ended extended support for SQL Server 2005 on April 12, 2016. Above all, recognize that anything that holds an organization's most valuable electronic and intellectual assets deserves the highest level of protection.

"Protecting databases doesn't have to be difficult. At its core, a database is just another system, and many of the best practices for preventative security apply to databases just as they apply to any other system."

APPLICATION SECURITY

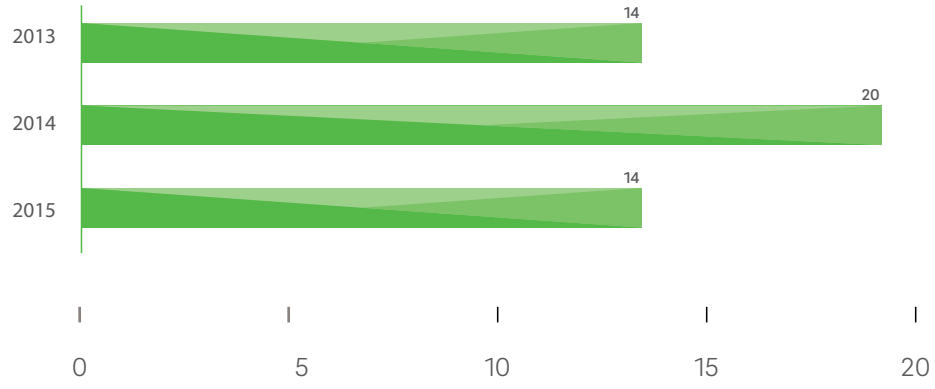
We identified more than 14,000 vulnerabilities in applications tested with Trustwave application scanning and testing services in 2015. Ninety-seven percent of the applications we tested had one or more security vulnerabilities.

VULNERABLE APPLICATIONS



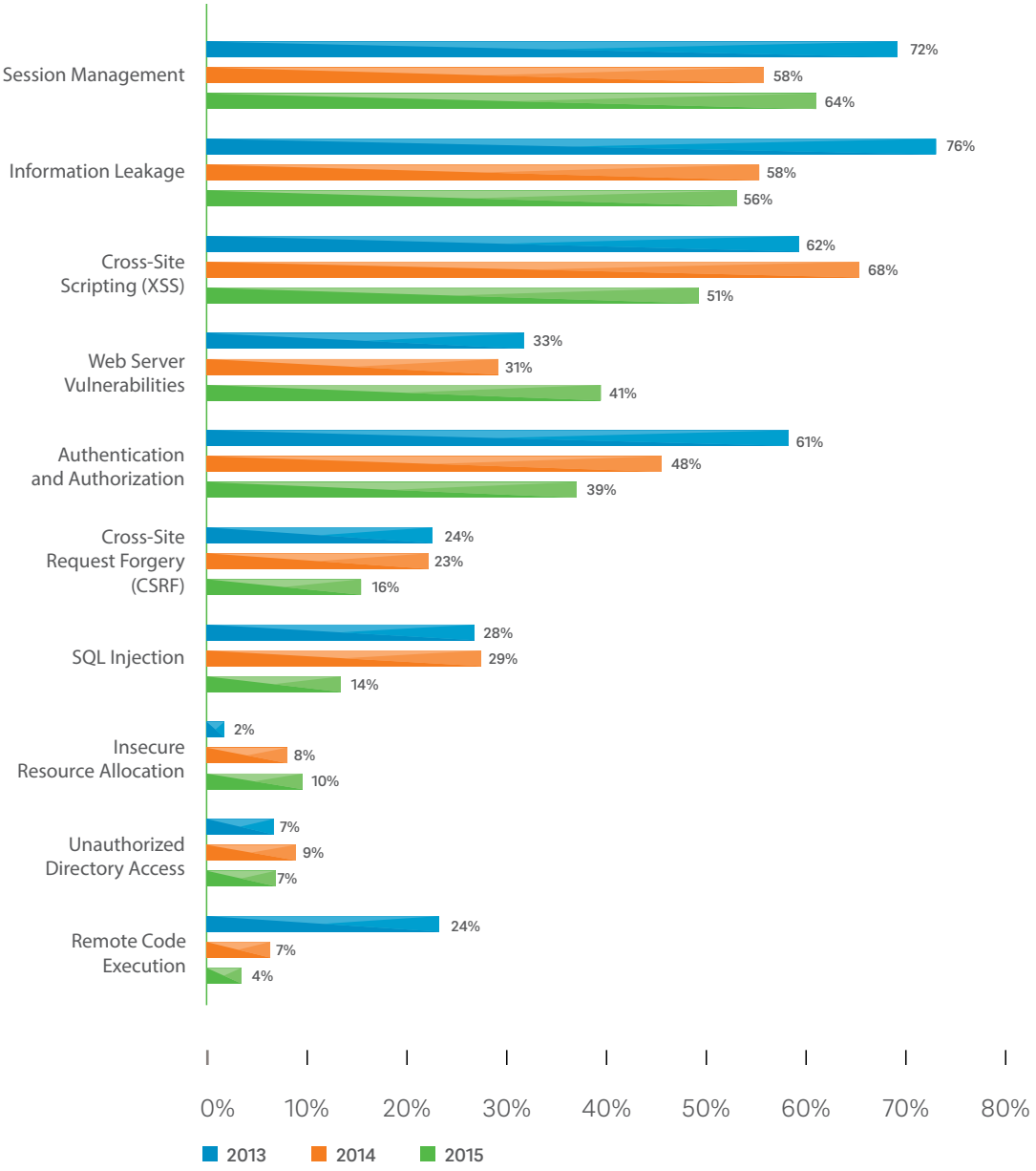
The median number of vulnerabilities per application decreased 30 percent in 2015 compared to the prior year, from 20 to 14. The maximum number of vulnerabilities we found in a single application was 667.

MEDIAN VULNERABILITIES PER APPLICATION



Application Vulnerability Categories

PERCENT OF APPLICATIONS WITH VULNERABILITIES



Session management vulnerabilities made up the most common class of vulnerability affecting the applications that Trustwave tested in 2015. Sixty-four percent of the applications had session management vulnerabilities, up from 58 percent in 2014. Session management vulnerabilities can allow an attacker to take over or eavesdrop on a user session, which can place sensitive information at risk. The vast majority of the session management vulnerabilities identified involved improper handling of HTTP cookies, which are used to preserve state across inherently stateless web connections. Cookies can contain session tokens, authentication information, or other sensitive information that can facilitate session hijacking if compromised, making proper cookie protection vital for ensuring the security of an application.

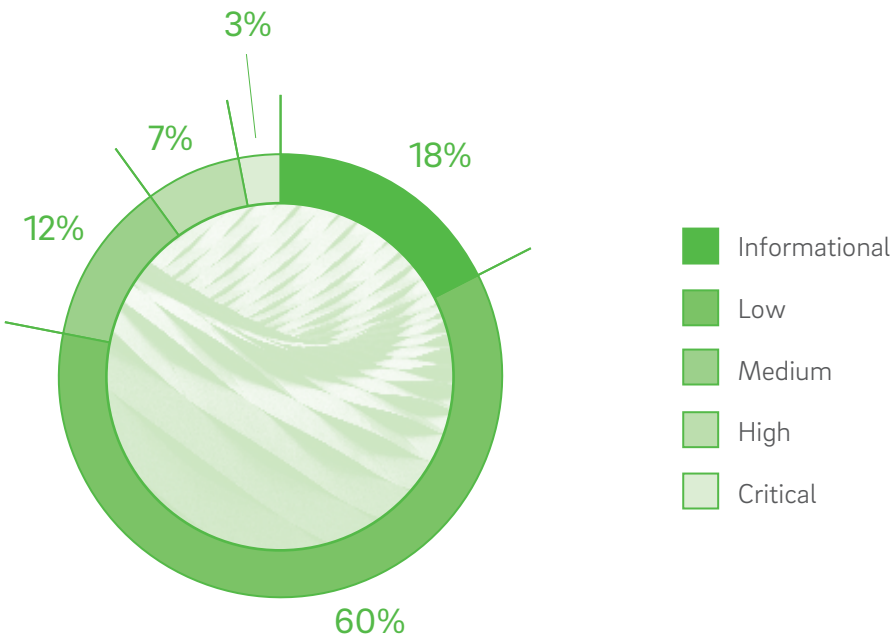
Fifty-six percent of the applications Trustwave tested had information leakage vulnerabilities, a slight decrease from 58 percent in 2014. Information leakage flaws involve the inappropriate disclosure of sensitive data, such as technical details of the application, environment or user data. Examples include application exception and form-caching vulnerabilities. These flaws can provide a cybercriminal with actionable information about how an application operates and potential weaknesses within it. However, these flaws alone do not provide an entry point for compromise of the application. So while such vulnerabilities are cause for concern, they're only a small part of an attack.

Cross-site scripting (XSS) vulnerabilities, which were present in 51 percent of applications tested, enable attacks on an application's users. A cross-site scripting attack allows an attacker to relay malicious scripts from an otherwise trusted URL to compromise information maintained within the victim's browser. Such flaws don't typically place a company's tangible assets at risk. A vulnerability that results in the compromise of a visitor's browser might tarnish a company's brand, but not to the extent a large-scale compromise of their customers' personally identifiable or payment card information might.

One significant bright spot in these results involves the decrease in the percentage of tested applications that were affected by SQL injection vulnerabilities, which dropped from 29 percent in 2014 to 14 percent in 2015. One of the most dangerous classes of vulnerability, SQL injection flaws allow a cybercriminal to compromise—or outright destroy—a database and plunder system administration credentials, intellectual property, sensitive customer information, or payment card data. While any number of SQL injection vulnerabilities is too many, the decrease is a welcome indication that application developers may be making prevention a higher priority.

Application Vulnerability Risk Levels

FREQUENCY OF VULNERABILITIES IDENTIFIED BY RISK LEVEL



Trustwave uncovered more than 24,000 vulnerabilities in web applications in 2015 through its on-demand security scanning and testing service. Of these, more than three-fourths were classified as informational, or low-risk. Medium-risk vulnerabilities accounted for 12 percent of the vulnerabilities identified. High-risk vulnerabilities accounted for 7 percent, and 3 percent of identified vulnerabilities were classified as critical, the most severe category we use.

TOP 10 CRITICAL VULNERABILITIES IDENTIFIED THROUGH PENETRATION TESTING

Vulnerability	Percent of all vulnerabilities	Percent of critical vulnerabilities
Heartbleed OpenSSL Memory Leakage	0.66%	17.6%
Authentication Bypass	0.31%	8.2%
Weak Administrator Password	0.27%	7.2%
JBoss Administrative Console Access	0.27%	7.1%
SQL Injection	0.18%	4.9%
Sensitive Data Stored Unencrypted	0.18%	4.8%
NetBIOS Name Service Poisoning	0.14%	3.6%
LLMNR Name Service Poisoning	0.11%	2.9%
Phishing Site Captures Employee Usernames and Passwords	0.11%	2.9%
Vertical Privilege Escalation	0.11%	2.8%

The most common critical vulnerability Trustwave identified in 2015 was a serious vulnerability in the OpenSSL library designated “Heartbleed.” First reported in 2014, Heartbleed is a flaw in the way OpenSSL implements the “heartbeat” function of Transport Layer Security (TLS), which allows for longer sessions without renegotiating the encryption channel. Exploiting the vulnerability can allow an attacker to access up to 64KB of data dumped directly from memory, which can include sensitive information like usernames, passwords, payment card details, and even the server’s private encryption key.

The second most common critical vulnerability identified in 2015 involved web pages intended for authenticated users that could nevertheless be accessed without a valid session identifier. In some cases, these pages exposed sensitive information such as user data and credentials, source code, or public and private encryption keys.

Weak administrator passwords uncovered through pen testing accounted for a notable fraction of critical vulnerabilities identified, as did improperly secured instances of the administrative console for JBoss, a Java-based web application server platform. Other critical vulnerabilities encountered included SQL injection flaws, sensitive data stored unencrypted, and name services that were vulnerable to spoofing.

TOP 10 HIGH-RISK VULNERABILITIES IDENTIFIED THROUGH PENETRATION TESTING

Vulnerability	Percent of all vulnerabilities	Percent of critical vulnerabilities
LLMNR Name Service Poisoning	1.02%	10.7%
Cross-Site Scripting (XSS), Persistent	0.89%	9.3%
NetBIOS Name Service Poisoning	0.48%	5.1%
SQL Injection	0.45%	4.7%
Shared Password for Local Administrator with Remote Logon	0.44%	4.6%
Sensitive Data Stored Unencrypted	0.36%	3.8%
Web Proxy Autodiscovery Protocol Man-in-the-Middle	0.27%	2.9%
Horizontal Privilege Escalation	0.25%	2.6%
Vertical Privilege Escalation	0.24%	2.5%
Path Traversal	0.24%	2.5%

Link-Local Multicast Name Resolution (LLMNR) implementations accounted for the largest percentage of high-risk vulnerabilities Trustwave identified in 2015. LLMNR is a name-resolution protocol used by Microsoft Windows machines, similar to DNS. When a DNS name lookup fails, the host making the request can use LLMNR to broadcast the request to other hosts on the network. A malicious host on the network can respond with the IP address of a machine under the control of an attacker, which can then be used to capture user credentials and other sensitive information. Depending on the circumstances, this can be either a critical or a high-risk vulnerability. NetBIOS name service poisoning, the third most common entry on the list, is a similar vulnerability.

Applications that were vulnerable to cross-site scripting accounted for the second largest percent of vulnerabilities. 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 may then be executed by other users. This type of attack can be used to steal information such as usernames and passwords, remotely control or monitor the victim's browser, or impersonate a web page used to gather order information, including payment card numbers.

Other high-risk vulnerabilities uncovered through our penetration testing included SQL injection flaws, administrator passwords shared between machines, sensitive data stored unencrypted, and various man-in-the-middle and privilege escalation vulnerabilities.

2016 TRUSTWAVE GLOBAL SECURITY REPORT CONTRIBUTORS

Brett Andrews

Assi Barak

Daniel Chechik

Anirban Chowdhuri

Chua Chee Kwang Edmund

Anat Davidi

Christophe De La Fuente

Dixie Fisher

Phil Hay

Paul Henry

Brian Hussey

Dan Kaplan

Rami Kogan

Arseny Levin

Ziv Mador

Dan Meged

Ryan Merritt

Lawrence Munro

Ooi Lih Shyue Alex

Cas Purdy

John Randall

Alex Rothacker

Chaim Sanders

Karl Sigler

Todd Wilson

Aaron Wooten

WORLDWIDE HEADQUARTERS

70 West Madison St.
Suite 1050
Chicago, IL 60602

P: +1 (312) 873-7500
F: +1 (312) 443-8028

EMEA HEADQUARTERS

Westminster Tower
3 Albert Embankment
London SE1 7SP

P: +44 (0) 845 456 9611
F: +44 (0) 845 456 9612

APAC HEADQUARTERS

Level 2, 48 Hunter St.
Sydney, NSW 2000
Australia

P: +61 (0) 2 9236 4200
F: +61 (0) 2 9236 4299

LAC HEADQUARTERS

Rua Cincinato Braga, 340
n 71 - Edificio Delta Plaza
São Paulo - SP
CEP: 01333-010 - BRASIL

P: +55 (11) 4064-6101

Trustwave®