

# BATTLEFIELD MALWARE AND THE FIGHT AGAINST CYBER CRIME

Marwan Omar

## BATTLEFIELD MALWARE AND THE FIGHT AGAINST CYBER CRIME

Report presented to Universidade Fernando Pessoa as requisite to fulfill the post-doctoral program in Information Science under the supervision of Prof. Dr. Luis Borges Gouveia.

Universidade Fernando Pessoa  
Porto 2021

## ABSTRACT

Our cyber space is quickly becoming over-whelmed with ever-evolving malware that breaches all security defenses, works viciously in the background without user awareness or interaction, and secretly leaks of confidential business data. One of the most pressing challenges faced by business organizations when they experience a cyber-attack is that, more often than not, those organizations do not have the knowledge nor readiness of how to analyze malware once it has been discovered on their production computer networks. The objective of this six months post-doctoral project is to present the fundamentals of malware reverse-engineering, the tools and techniques needed to properly analyze malicious programs to determine their characteristics which can prove extremely helpful when investigating data breaches. Those tools and techniques will provide insights to incident response teams and digital investigation professionals. In order to stop hackers in their tracks and beat cyber criminals in their own game, we need to equip cyber security professionals with the knowledge and skills necessary to detect and respond to malware attacks. Learning and mastering the inner workings of malware will help in the fight against the ever-changing malware landscape.

**Keywords:** *Reverse-engineering malware, malware analysis, static analysis, behavioral analysis, code-level analysis, incident response, criminal hackers, cybercrime, cryptocurrency, ransomware.*

# BATTLEFIELD MALWARE AND THE FIGHT AGAINST CYBER CRIME

## **DEDICATION**

This submission is dedicated to my wonderful wife Maha Nawaf for her unwavering support and for inspiring me to pursue this degree. And to my children Tala and Adam, you are the sole motivation for pursuing and conquering the impossible.

**ACKNOWLEDGEMENTS**

This submission would not have been possible without my supervisor's guidance, Professor Luis Borges Gouveia of University Fernando Pessoa. I am grateful for the opportunity to work under his research supervision and appreciate his support throughout this journey.

.

**TABLE OF CONTENTS**

INTRODUCTION .....6

CHAPTER I – Introduction to Malware Analysis .....8

1.1. What is Malware Analysis .....9

1.2. Malware Analysis Techniques..... 10

1.3. Static Analysis ..... 11

1.4. Case Study: E-mail Scam Investigation..... 17

1.5. Detailed Analysis and Findings .....21

CHAPTER II – Ransomware.....30

2.1. Ryuk Ransomware .....32

2.2. Prevention .....33

2.3. Recovery .....35

2.4. Recent Prominent Variants .....36

2.5. Ransomware Cases .....38

2.6. Demonstration Methods.....40

CONCLUSION.....49

BIBLIOGRAPHY .....52

**LITERATURE PUBLISHED UNDER THE PROGRAMME.....55**

## INTRODUCTION

Security breaches due to attacks by malicious software (malware) continue to rise exponentially posing a major security threat to everyone in this digital age. With many computer users, corporations, and governments affected due to an unprecedented growth in malware attacks, malware detection and analysis has emerged as a hot research topic.

The extent of the damage caused by malicious software will often depend on whether the malware has infected a home computer or a corporate network. The consequences of the damage may also vary according to the specific type of malware and the type of device that is infected – plus the nature of the data that is stored on or accessed by the device.

Whereas, in some cases the results of a malware infection may be imperceptible to the user, in other cases the damage can have serious consequences: In the crimeware world, financial botnets are a global threat to banking organizations. Such malware purposely performs financial fraud and steals critical information from clients' computers. In this thesis report we aim to highlight the criticality of malware attacks and the severe consequences of becoming a victim to malware data breaches. We observe that at present, computer viruses statistics show that malware attacks in 2021 have been recorded as costing the average US company an average of \$2.4 million per year. Furthermore, Analysts predict that malware security spending forecasts for US businesses in 2022 will break the trillion-dollar mark easily. On a global scale, Reports incorporating malware statistics by operating system have received data showing that the average global cost of cybercrime increased by over 27% in 2017 and went up by a further 32% in 2020. Additionally, we show in figure 1 that malware facts show that attacks ran wild in 2020 and resulted in nearly

## BATTLEFIELD MALWARE AND THE FIGHT AGAINST CYBER CRIME

80% of all large corporations taking a hit from cybercriminals. In addition to corporations, millions of users across the US were targeted and fell victim to malware attacks.

Contributions of this research project:

1. We conduct an in-depth analysis of malware samples through static analysis to understand their dynamics, by analyzing various artifacts, such as strings, disassembly, Control Flow Graph (CFG), IP addresses, ports, and functions;
2. Exploring the analysis of the malicious software further and to circumvent obfuscation, we perform dynamic analysis to understand the behavior of malware. To mitigate the threats posed by the numerous malware variants, analysts group malware samples based on their behavior and intent;
3. We identify gaps in malware detection. Our developed tools can be leveraged to understand the unique characteristics and behaviors of the malware families, and the impact of software weaknesses on the victim organization.

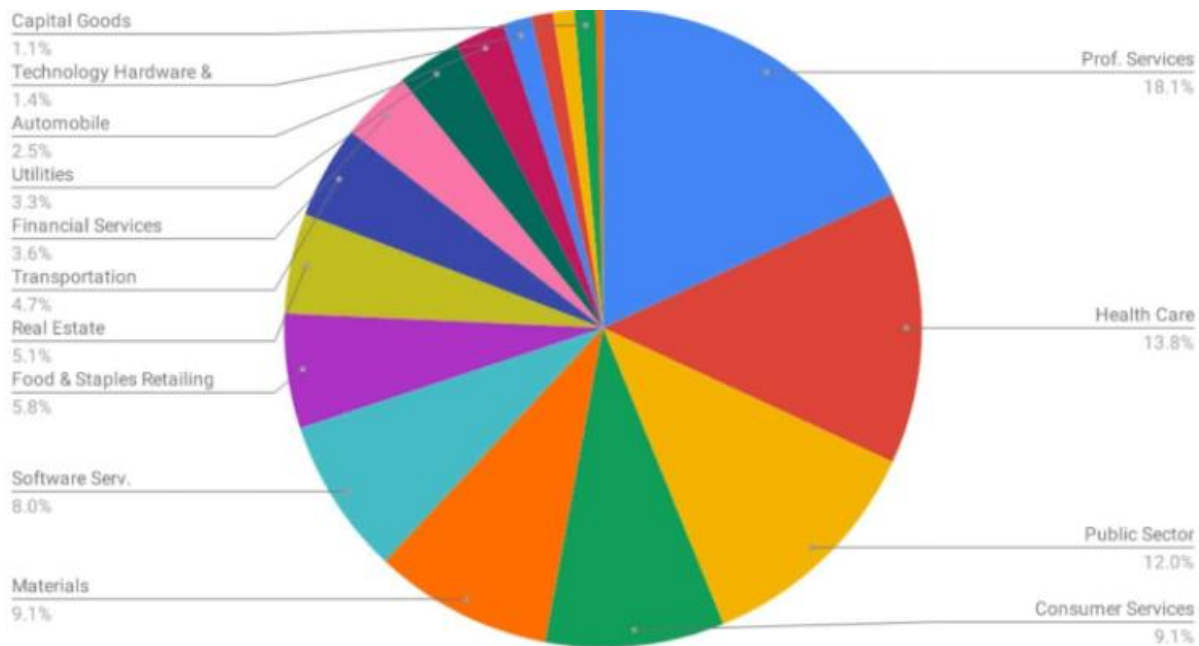


Figure 1: Common Industries Targeted by Malware

## **CHAPTER 1 – Introduction to the Fascinating World of Malware Analysis**

Information technology has forever changed the way we live and work and there is no doubt about the fact that the world has benefited from technological advancements in ways that are immeasurable and never imagined before. However, these technological advancements are not risk-free and there is a flip-side to this: cybercriminal activities have skyrocketed in the recent years to the point where in some cases hackers have been able to take business organizations as hostage using malware.

Most cyber-attacks involve deploying some type of malware. Malware that viciously targets every industry, every sector, every enterprise and even individuals has shown its capabilities to take entire business organizations offline and cause significant financial damage in billions of dollars annually. Malware authors are constantly evolving in their attack strategies and sophistication and are developing malware that is difficult to detect and can lay dormant in the background for quite some time in order to evade security controls (Carrillo-Mondejar et al. 2020).

Our cyber space is quickly becoming over-whelmed with ever-evolving malware that breaches all security defenses, works viciously in the background without user awareness or interaction, and secretly leaks confidential business data. In order to stop hackers in their tracks and beat cyber criminals in their own game, we need to equip cyber security professionals with the knowledge and skills necessary to detect and respond to malware attacks. Learning and mastering the inner workings of malware will help in the fight against the ever-changing malware landscape. This learning is pursued via malware analysis techniques which could be performed statically, automatically, dynamically, or on the code level.

### **1. 1. What is Malware Analysis?**



## BATTLEFIELD MALWARE AND THE FIGHT AGAINST CYBER CRIME

Before we try to understand what is malware analysis and why malware analysis is important in the context of cyber security, let's try to define malware. Malware is code that is utilized to perform malicious actions with the intent of causing harm and destruction on computer system and networks. Malware is typically designed to take advantage of some type of security flaw or backdoor and benefit at the victim's expense. Moreover, malware is often written by people or organizations to use its capabilities for malicious intentions and purpose.

Malware analysis aims to examine malware's behavior. The objective of malware analysis is to gain an understanding of the inner workings of malware and how to detect and remove it. To reliably analyze malware, we analyze the malware specimen in a safe environment to identify its characteristics and functionalities so security defenses can be developed to secure and protect a business organization's digital assets (Monnappa, 2018).

Many of the cyber incidents and data breaches that we see and hear about in the news are typically carried out using some sort of malware, which might be designed to enable the attacker to gain remote control of a compromised computing system, steal business sensitive data, spy on the victim's online activities, spread within the victim/target organization, and so on. That's where the importance of knowing how to examine and analyze malicious program comes into play as it's critical to be able to control the situation and minimize the damage and disruption to business operations and the organization at large (Afianian et al. 2020).

One of the most pressing challenges faced by business organizations when they experience a cyber-attack is that, more often than not, those organizations do not have the knowledge nor readiness of how to analyze malware once it has been discovered on their production computer networks. This is where this paper can help, this paper will help shed light on the tools and techniques needed to properly analyze malicious programs to determine their characteristics which

## BATTLEFIELD MALWARE AND THE FIGHT AGAINST CYBER CRIME

can prove extremely helpful when investigating data breaches as those tools and techniques will provide insights to incident response teams and digital investigation professionals. Some of the key things that cyber professionals can learn when analyzing malware are questions related to the nature of threat posed by malware, the objective of the adversary using the malware, how to contain, eradicate, and recover from an incident and perhaps more importantly, how to strengthen cyber defenses so that the cyber-attack does not reoccur in the future (MONNAPPA, 2018)

### 1.2. Malware Analysis Techniques

The process of analyzing malware should be a methodical one and generally involves several stages, which can be viewed in the order of increasing complexity. The pyramid below has been used to illustrate such one ranking:

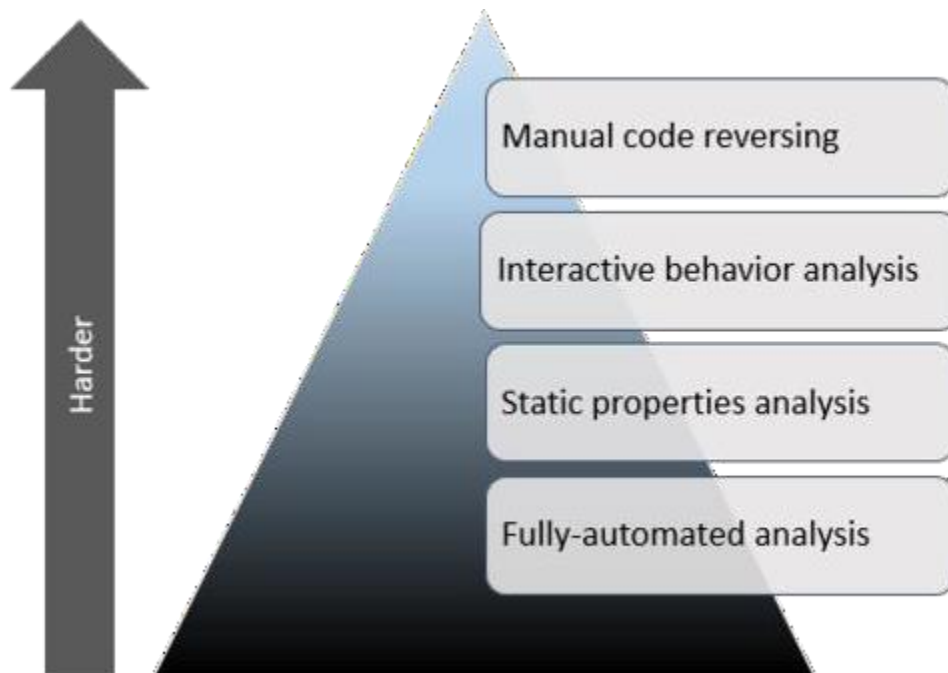


Figure 2: Malware Analysis Techniques

Fully automated tools can offer an easy and simple way to examine and analyze a particular malware specimen (as shown in the figure above), some of which are available as commercial

## BATTLEFIELD MALWARE AND THE FIGHT AGAINST CYBER CRIME

products and some as free ones. These tools are designed as a triage method to quickly assess the behavior of the specimen if it ran on a system. They typically produce reports with details such as the registry keys used by the malicious program, network traffic, and so on. The down side of those fully-automated utilities is that they may not provide as much insight as human analysts would obtain when examining the specimen in a more manual manner.

On the flip side, and as a benefit, they can contribute to the incident response by rapidly handling vast amounts of malware, allowing malware examiners to focus on the malicious binaries that truly require their time and attention (Han et al, 2015).

### **1.3. Static Analysis**

Static analysis is one of the malware analysis techniques used by malware analysts to quickly triage suspect programs/files without executing them. During this initial assessment phase, the goal is to be able to extract valuable insights from the suspect binary which would help inform the subsequent steps so that we can determine how to analyze or categorize the suspect file and where to focus our analysis efforts (Kirubavathi & Anitha, 2018).

This section covers various tools and techniques to extract valuable information from the suspect binary. we will learn the following: identifying the malware's target architecture, fingerprinting the malware, scanning the suspect binary with anti-virus engines extracting strings, functions, and metadata associated with the file identifying the obfuscation techniques used to thwart analysis classifying and comparing the malware samples. (Ul Haq et al., 2018).

These techniques can reveal different information about the file. It is not required to follow all these techniques, and they need not be followed in the order presented. The choice of techniques to use depends on your goal and the context surrounding the suspect file (Shekhawat et al., 2019).

Static properties analysis examines the static properties of suspicious files such as file hashes, embedded resources, digital certificates, and interesting strings (Zhang & Song, 2020). A

## BATTLEFIELD MALWARE AND THE FIGHT AGAINST CYBER CRIME

good starting point to analyzing potential malware files is the static properties analysis. The objective of the static properties analysis is to quickly assess the nature of a potential malware file and develop plans for taking a closer look in the subsequent phases of malware analysis (Chakkaravarthy et al., 2019).

The best way to demonstrate how to apply the static analysis technique is to use a real- world malware sample that exhibits static properties (Ni; Qian & Zhang, 2018). We will consider a fictional enterprise security incident scenario whereby a system is misbehaving and showing indicators of compromise (IOC). The said system is suddenly rebooting and is experiencing slow performance, in addition, there is a foreign process called `brbbot.exe` running from `%AppData%`.

The `%AppData%` is an environment variable on Windows that typically points to a directory in the user's profile, such as `C:\Users\REM\AppData\Roaming`. This location is designed for storing *“user-specific files that applications install”* according to Microsoft (Technet.microsoft.com).

# BATTLEFIELD MALWARE AND THE FIGHT AGAINST CYBER CRIME

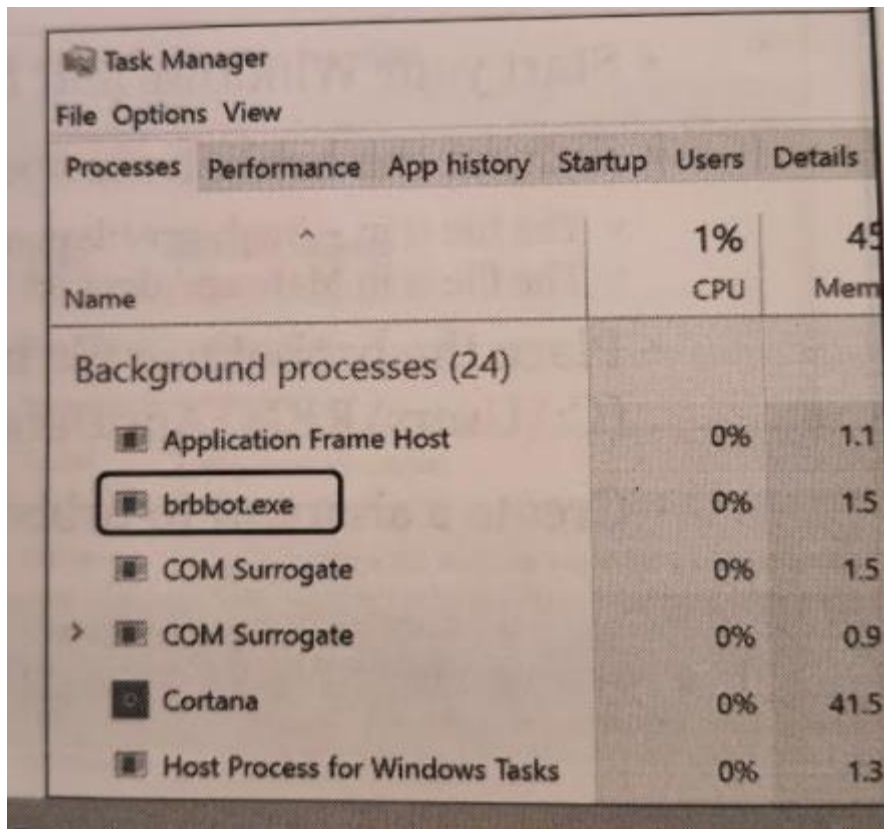


Figure 3: Foreign process running on the system

## BATTLEFIELD MALWARE AND THE FIGHT AGAINST CYBER CRIME

As a “rule of thumb”, when we run malware in our lab, we want to determine what is the worst that the malware specimen can do, allowing it to have opportunity to reach its full malicious potential. That said, we will make the *BRBbot* specimen run in the virtual lab system and infect our virtual host with this potential malware, all of this is done with full administrator privileges to allow the potential malware sample to exhibit its full malicious potential, as mentioned earlier (Monnappa, 2018).

This approach will help us in making sure that the specimen is able to interact with all aspects of the infected host, allowing it to achieve its full malicious potential. However, sometimes it’s also useful to run the specimen with normal, non-administrative privileges to observe its behavior-for instance, if mimicking a particular scenario of a production system in a business setting (Stiborek et al., 2018).

## BATTLEFIELD MALWARE AND THE FIGHT AGAINST CYBER CRIME

### **Initial Assessment of a Potential Malware Specimen – BRBbot.exe**

Prior to deciding whether to examine the malicious (or suspicious) program using behavioral or code analysis techniques, we should consider performing an initial assessment of the malicious binary by examining its static properties (Kara, I. (2019)). In some cases, the binary may turn out to be not malicious because it possesses a hash that belongs to a trusted program. Performing static properties analysis is important as it can help us determine where to focus our subsequent analysis efforts. Static properties analysis is a useful first step as part of the triage effort (Yadav, R. M. (2019)).

Given our brbbot.exe specimen, which is a Windows executable, static properties of a Windows executable include asking the following questions:

1. Is it malware?
2. What type of file is it (e.g. .exe, .dll, .com, .sys, and so on)?
3. What is the target architecture (32-bit or 64-bit platform)?
4. How bad is it?
5. How to detect it?
6. How to analyze it?

Some of those properties will be examined using the brbbot.exe sample.

### Extracting String

A string can be defined as a sequence of ASCII and Unicode-printable characters embedded within a file. Extracting and examining strings can give malware analysts insights about the program functionality and capabilities. Therefore, a common first step in examining a suspicious file involves looking at the strings imbedded in it. This examination may reveal filenames, domain names, URLs, IP addresses, hostnames, or registry keys that the program may attempt to access and can help focussubsequent steps of the investigation. It must be noted, however, that we cannot trust all the strings embedded into the program because they might have been embedded there to mislead themalware analyst. Not to mention the fact that relying on strings alone does not allow a clear or compelling picture of the purpose and functionality of a malware binary (Carrillo-Mondejar et al., 2020).

A convenient way to accomplish this is to use the *pestr* tool on *REMnux*, this utility designed for extracting strings from Windows executable files, automatically obtains both ASCIIand Uni-code-encoded strings in one shot. Alternatively, we could use the well-known *Strings* tool, present on most Linux distributions. By default, *Strings* extracts only ASCII-encoded strings. We can tell the tool to extract Unicode strings by specifying the *-encoding=l* (lower caseL) parameter as shown on the screen capture below.

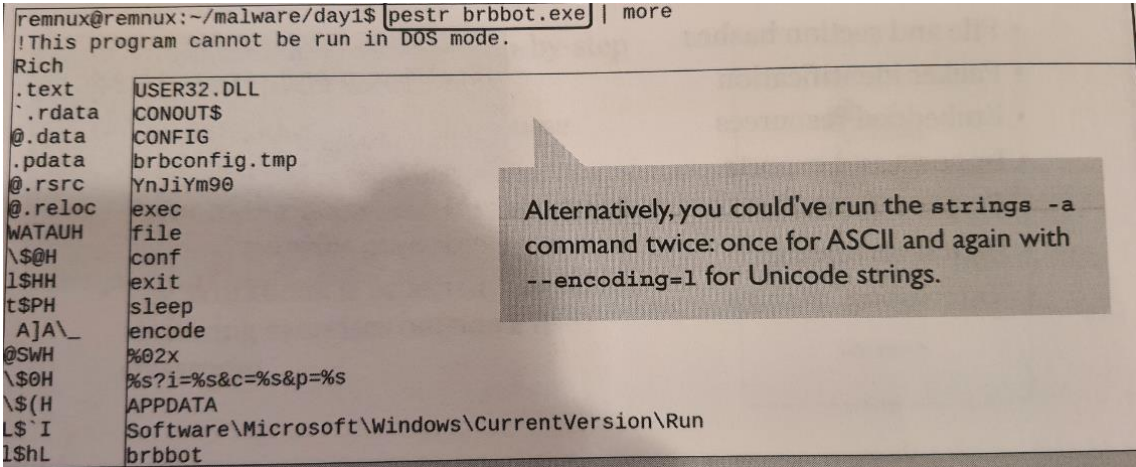


Figure 4: Pestr string helps extract strings from a malicious executable.



# BATTLEFIELD MALWARE AND THE FIGHT AGAINST CYBER CRIME

Another way to examine ASCII and Unicode-encoded strings on a file is to use the GUI tool *BinText*, available free from <http://www.mcafee.com/us/downloads/free-tools/bintext.aspx>. *BinText*'s Filter tab enables you to configure parameters such as what characters the tool considers as belonging to a string, as well as the minimum number of characters the tool considers as belonging to a string. The default minimum text length value is 5. You can decrease it to find shorter strings.

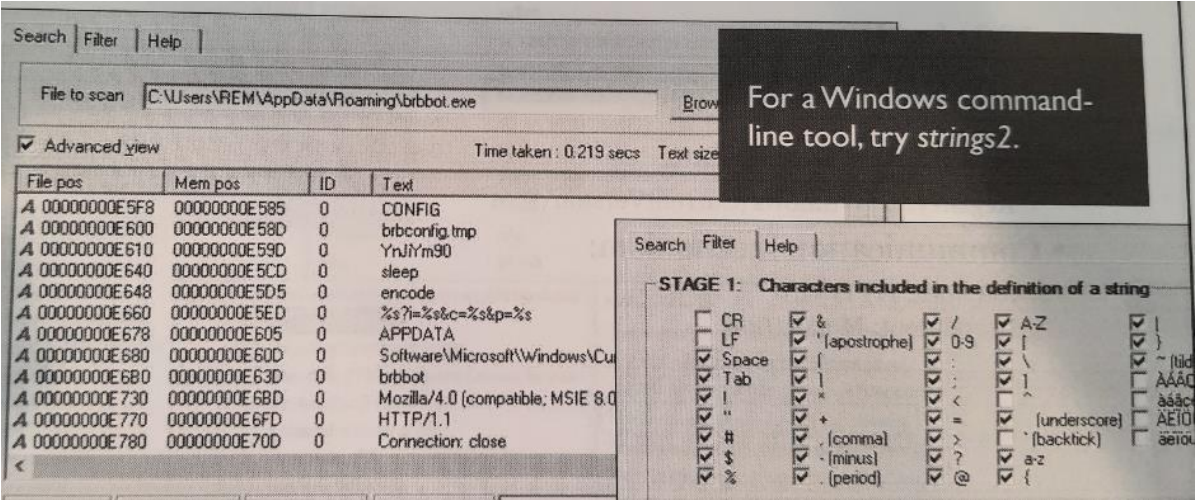


Figure 5: Bin Text as a GUI to examine embedded strings on Windows.

### **Strings embedded in brbbot.exe suggest a few potential characteristics**

By examining the strings embedded inside brbbot.exe, we can probably begin developing some hypothesis about its characteristics. For instance, the string *HTTP/1.1* might imply that this program can communicate over HTTP. The string referring to the *Run registry key* suggests that this program might use this key as a persistence mechanism.

Strings might also enable us to formulate indicators of compromise (IOCs), which can help in identifying systems throughout our organization infected with this specimen. Potential IOCs based on strings are the *brbbconfig.tmp* file and the string *%s?i=%s&c=%s&p=%s* as part of an HTTP request. Along these lines, the string that began with *Mozilla 1/4.0* could be a specific User-Agent header sent as part of the specimen's HTTP request; this could help us spot the associated network traffic in our environment (Feizollah et al., 2017).

Thus far, we have assumptions and theories, which have not been validated yet. To verify and validate the above theories, we will need to take our efforts to the next level and perform behavioral and code-analysis. In most cases, conducting static analysis on a suspect binary is a necessary and important first step to inform the next steps (behavioral and code-analysis) of the malware analysis process. We'll use a very methodical approach along with many handy tools throughout this paper to examine malicious software. They can be used for performing tasks related to static properties analysis, behavioral analysis, and code analysis of malware.

### **1.4. Case Study: E-mail Scam Investigation**

This case study is a real-world security incident which was given to the researcher as part of his freelancing work in the cyber security industry. A company employee had received a fake email pretending to be from Amazon, this email had a digital gift card embedded in order to lure the employee to click on it and claim a 50\$ gift card from Amazon. This was a highly convincing e-mail scam (Kirubavathi & Anitha, 2018).

Card scam has emerged that seeks to capitalize on the rise in online spending during the holiday season to infect victims with a banking trojan.

According to a report from security firm Cybereason, scammers are distributing a highly convincing phishing email that contains a document “weaponized with malicious macros”. (Stiborek; Pevný & Reháč, 2018).

Dressed up with Amazon branding, the email claims to offer the recipient a free \$100 voucher that they must download to activate. Once the victim has downloaded the file, they are redirected to a legitimate Amazon webpage, adding to the sense of legitimacy cultivated by the scammers.

- We've built a list of the best email clients out there;
- Check out our list of the best endpoint protection services right now;
- Here's our list of the best email services available.

The malware installed on the victim's device is a fearsome banking trojan known as Dridex, designed to steal e-banking credentials and other sensitive information. Operated by notorious cybercrime syndicate Evil Corp, the trojan has been active in various different forms since 2012 (Carrillo-Mondejar et al., 2020).

In this particular instance, the operators use three distinct delivery methods to infect users with the Dridex trojan: infected Word documents, self-extractive screensaver files

and VBScript files. This level of variety maximizes the opportunity to bypass email security tools that might filter for certain file extensions.

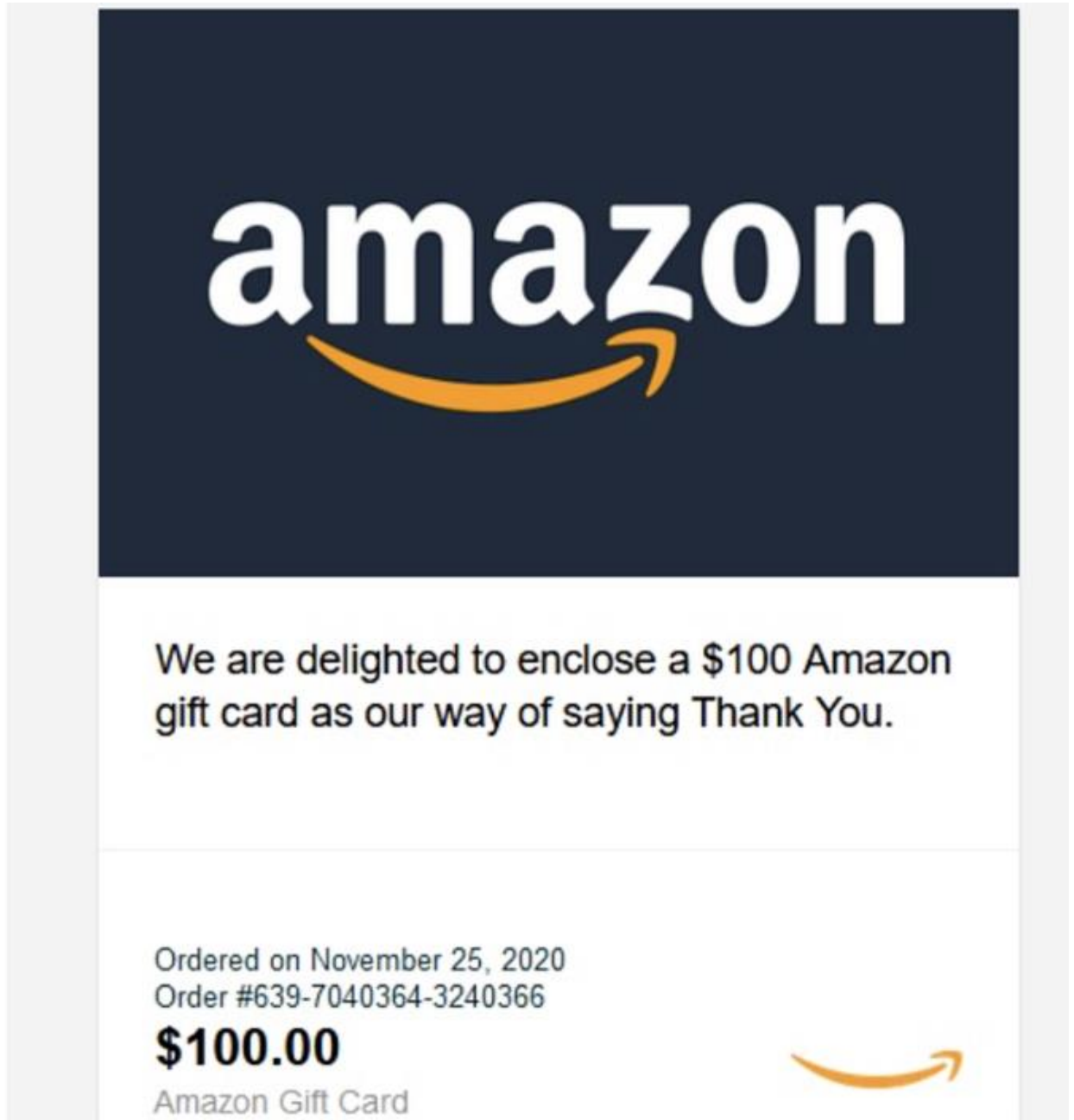


Figure 6: Amazon digital gift card embedded in an e-mail

### Handling of the security incident

The researcher used a Web browser from a lab system in order to allow the malicious *javascript* to exhibit its true behavior. The researcher activated behavioral monitoring tools to observe the behavior of the malicious *javascript* and captured network activities using a sniffer and a proxy called Fiddler (Yadav, 2019).

Initial examination of network traffic using Wireshark shows that the malicious *javascript* specimen was communicating with domains and encrypting all network traffic to complicate the analysis and investigation process. Therefore, the researcher turned to Fiddler which is a specialized network traffic analysis tool that has the ability to decrypt network traffic (Feizollah et al., 2017).

Here are the key points to keep in-mind about this malicious script:

1. Attackers employ scripts as part of web-based attacks, send them as e-mails and embed them in pictures within e-mails;
2. This malicious script implemented redirection techniques to redirect users from Amazon Web site to several malicious domains and hosts;
3. The malicious script was communicating with several malicious domains and obfuscating network traffic to complicate the analysis process;
4. The security analyst was able to de-obfuscate and decoded network traffic using a specialized tool called *Fiddler*;
5. When a user clicks on the “*Amazon Gift Card*” image within the spam e-mail, it redirects him to a malicious Web site (Host: *bh.contextweb.com*) which in turn communicates with 72 different malicious domains and hosts;
6. The malicious script included the “*iframe*” element which is a tag used to embed another Web site or document within the current HTML document. Attackers use this technique to redirect users from a legitimate web site to a malicious web site unbeknown to them;
7. If users clicked on “*login to your account*” link which was presented to them on Amazon Web page, they would be unwittingly providing their login credentials to a malicious domain;

8. The entire malicious script is attached to this report as word document for your reference.

## 1.5. Detailed Analysis and Findings

When a user receives the spam email, they are “lured” to click on the “Amazon Gift Card” image which was embedded in the email. Clicking on that image is equivalent to visiting the malicious web host which was programmed in the JavaScript (Xue et al., 2019).

Once a user clicks on the image, they are presented with the following Web page.

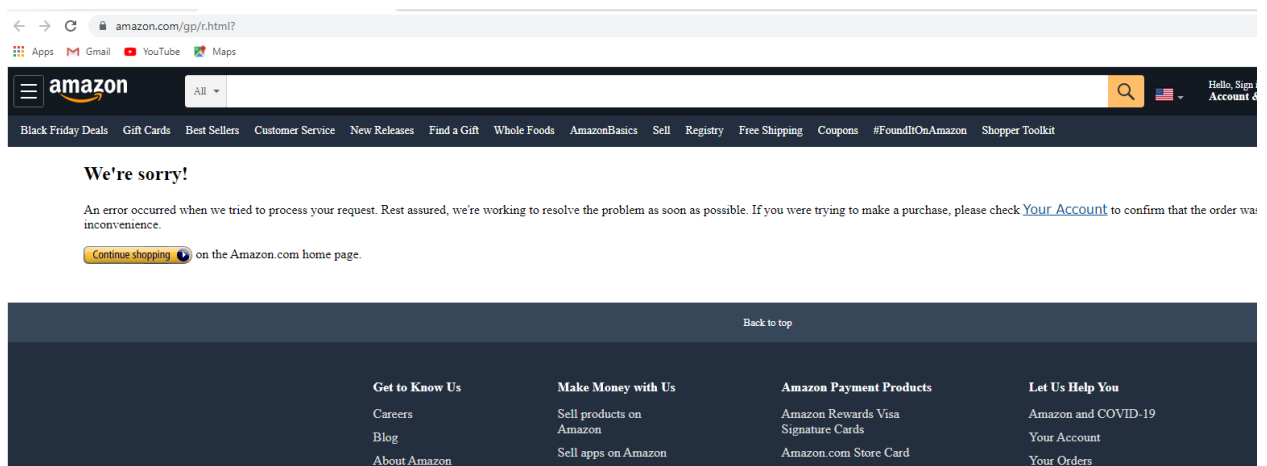


Figure 7: Redirect Web page after user clicks on the e-card from figure 6 above.

This is intelligently designed to trick users and make them believe that they visited an Amazon web page that is experiencing a technical issue. As shown in the screen shot, it even offers unwitting users to check “your account” to further make them believe that this is an actual Amazon Web page. Of course, it’s not and the moment a user clicks on “your account” they will actually be redirected again to yet another malicious domain

where users would have to type-in their login credentials. Once a user divulges their login credentials, those credentials would be transmitted to the malicious domain and users would be redirected to another error page (Lim & Yi, 2016).

It's typical for cyber criminals to use this redirection technique because unwitting users believe that they have landed on a real, legitimate page when in fact they are being redirected to a malicious host. Of course, users do not know nor do they see the malicious Web domains /hosts because the malicious *javascript* does its magic in the background and establishes the connection to the malicious domains (Carlin; O'Kane & Sezer, 2019).

Initial examination of network traffic using Wireshark did not yield any good results because this malicious *javascript* specimen was obfuscating all its network traffic when communicating with the adversary (see screenshot below). This is typical because attackers want to protect their network communications to complicate analysis and mislead security analysts (Ul Haq et al., 2018). To de-obfuscate network traffic, the researcher turned to a specialized tool called "*Fiddler*" which is a network monitoring and packet capturing tool that is capable of decoding network traffic. This tool came-in very handy and allowed the security analyst to decode the traffic and make sense of the inner working of this malicious JavaScript specimen.

## BATTLEFIELD MALWARE AND THE FIGHT AGAINST CYBER CRIME

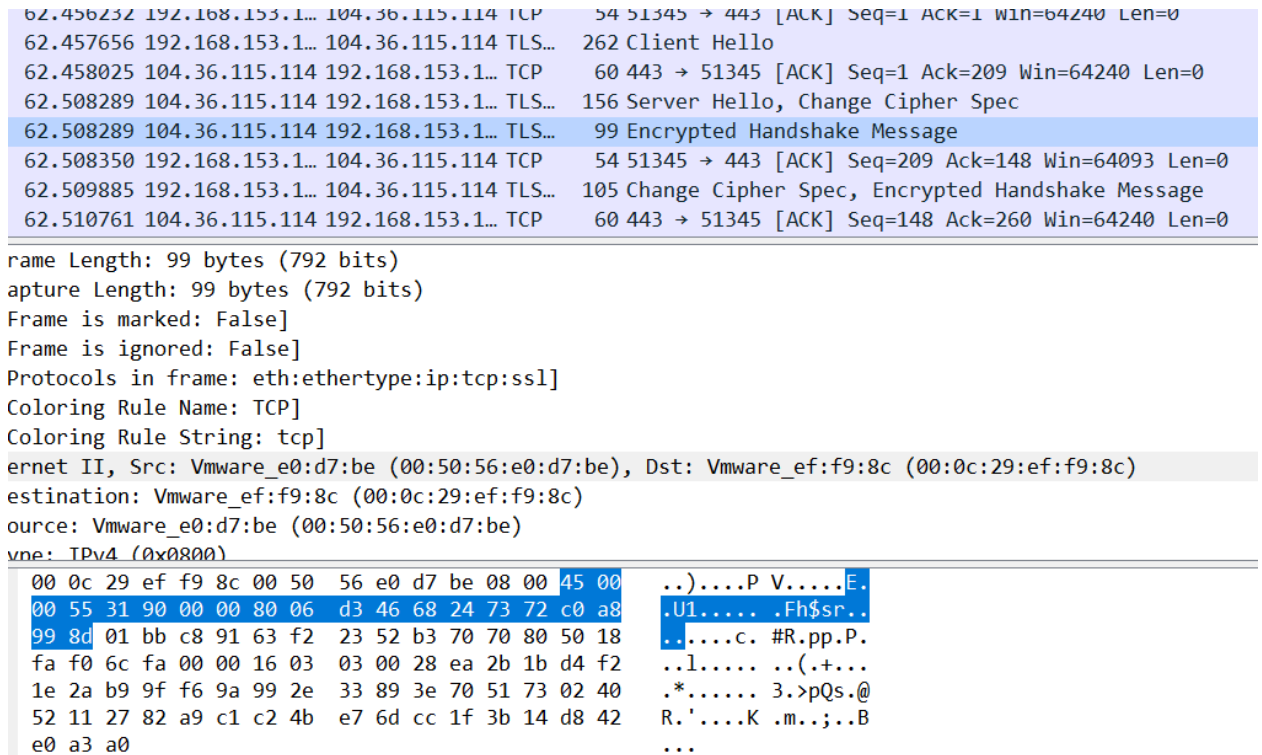


Figure 8: Screen shot above shows obfuscated network traffic from the malicious

*javascript*

Upon deploying *Fiddler* to capture and decode obfuscated network traffic, the researcher determined that the malicious script is redirecting users to a malicious host named *bh.contextweb.com*. After conducting “*open source threat intelligence*” (OSINT) and running the Web host through automated malware analysis engines, it’s evident that *bh.contexthost.web*’ is a malicious artifact and that it is contacting about 72 domains and about 75 hosts (Zhang & Song, 2020). See screenshots below”



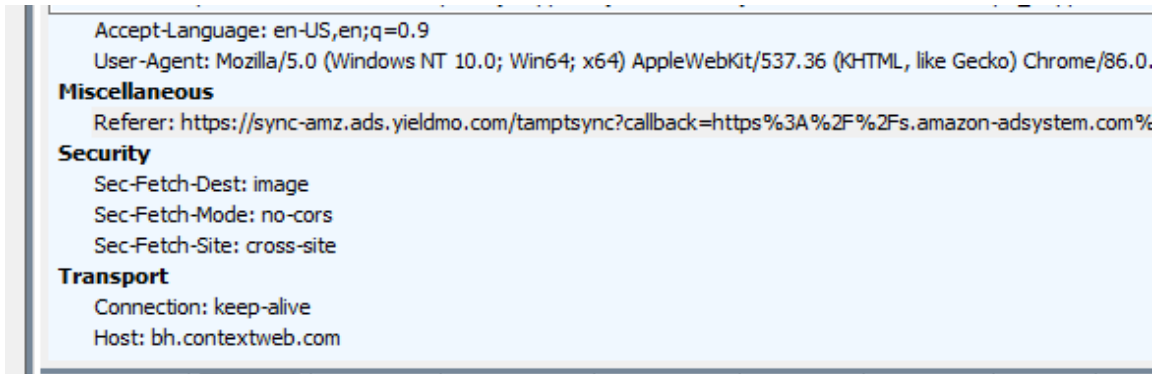


Figure 9: Screen shot show *Fiddler* was able to decode network traffic and an Amazon link redirects user to the “*bh.contextweb.com*” which was determined to be malicious.

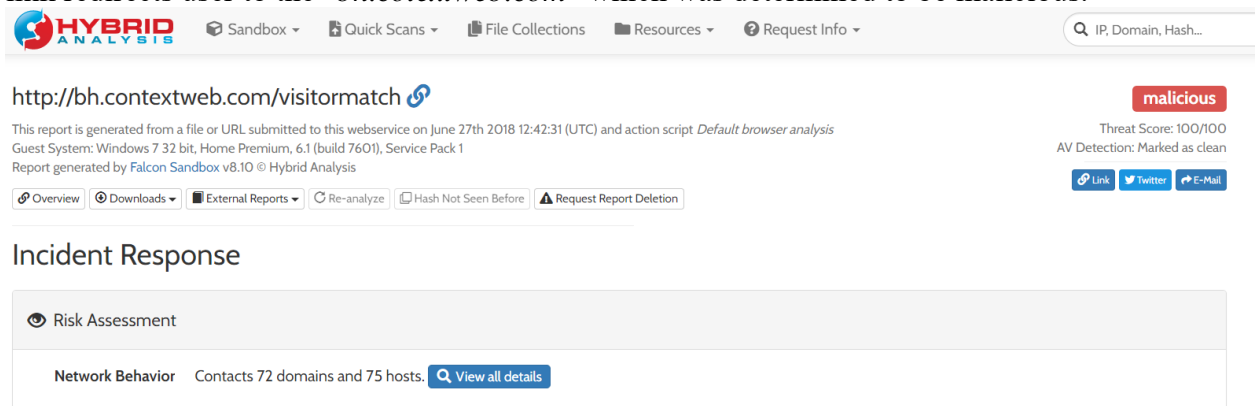


Figure 10: Screen shot above shows that “*bh.contextweb.com*” is malicious according to Automated Malware Analysis tools.

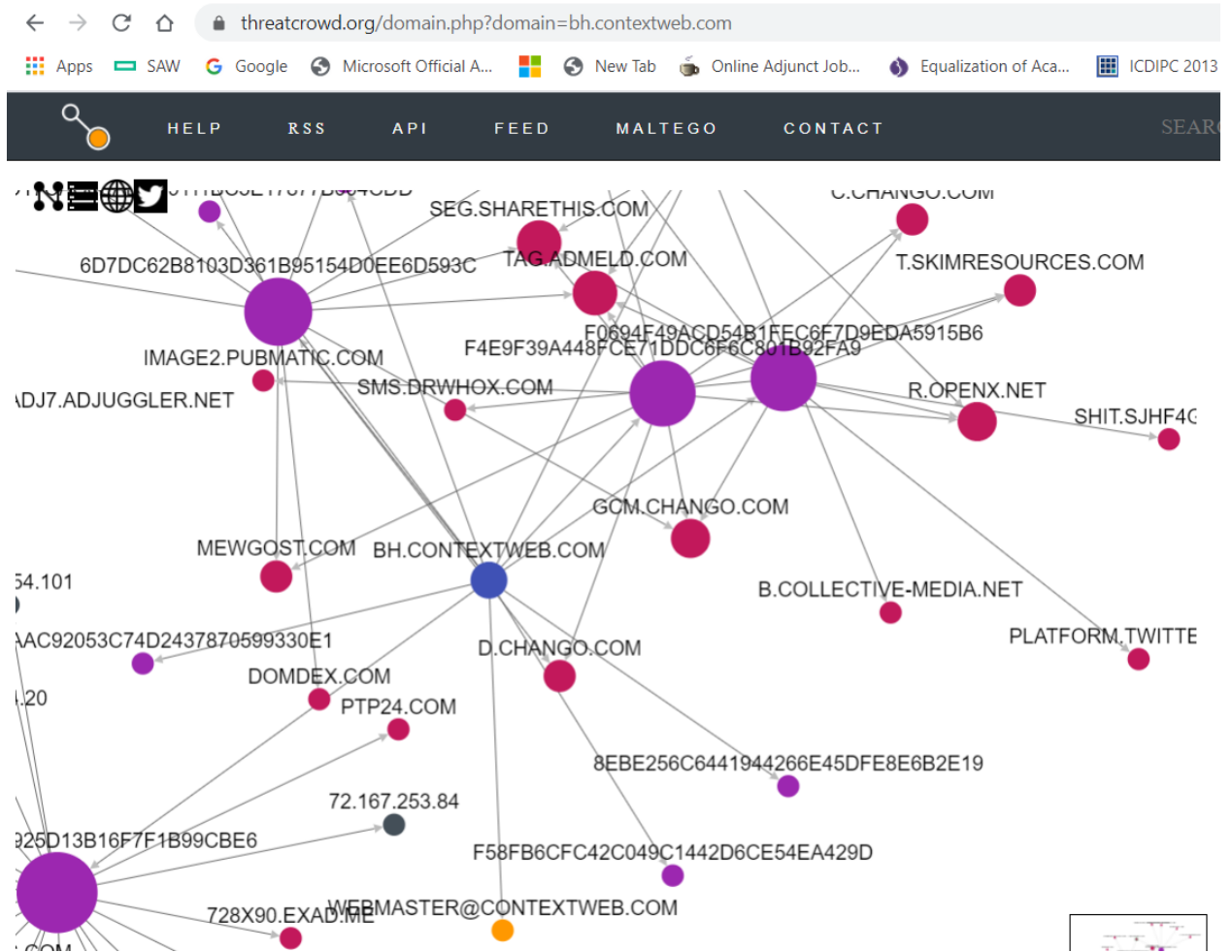


Figure 11: Screen capture above shows a graph of how “*bh.contextweb.com*” would contact other malicious domains and hosts according to [https://www.threatcrowd.org/domain.php? domain=bh.contextweb.com](https://www.threatcrowd.org/domain.php?domain=bh.contextweb.com).

## Host: odr.mookie1.com

This is another host that the malicious JavaScript specimen established contact with when it ran under controlled conditions in the lab system. Screenshot below shows how Fiddler decoded the host name (Shekhawat; Troia & Stamp, 2019).

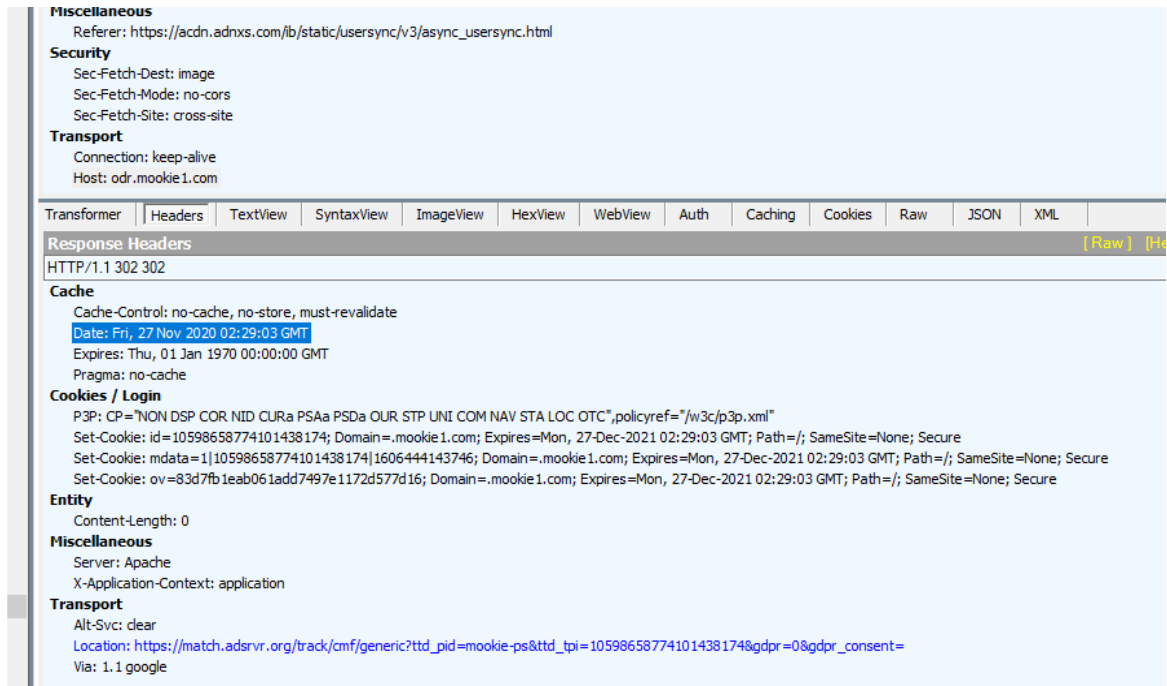


Figure 12: *Fiddler* decoded contents

The screen shot below show the analysis report from a malware assessment sand box. Its flagged as a malicious domain. This is the domain that hosts the malicious host “odr.mookie1.com”.

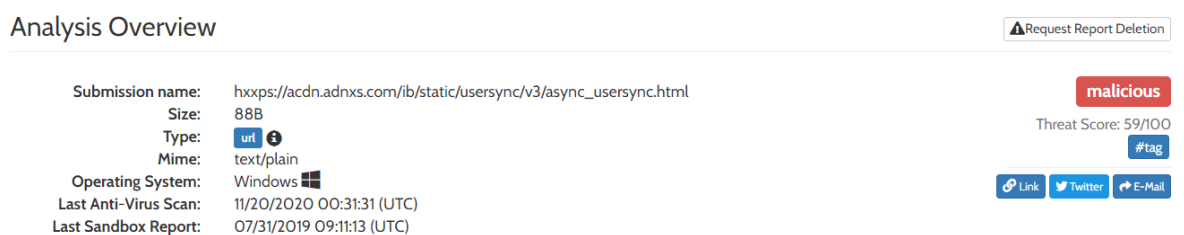


Figure 13: domain flagged as malicious from *Fiddler* decoded contents

Domain: <http://idsync.rlcdn.com/>

According to OSINT and automated malware analysis tools, the above domain is communicating with the host “idsync.rlcdn.com”. The screen shot below shows that the domain has the ability to contact other malicious artifacts. This is typical in malware infrastructure because malware has to communicate with other malicious command and

control servers and services in order to obtain instructions on what damage to inflict on victim computers (Chakkaravarthy; Sangeetha & Vaidehi, 2019).

## Network Related

### Malicious artifacts seen in the context of a contacted host

**details** Found malicious artifacts related to "34.95.92.78": ...

URL: <http://ei.rlcdn.com/384946.gif?m=Od179845fb55827ba6973d2b47658065&amp> (AV positives: 1/70 scanned on 07/02/2019 12:36:20)  
URL: [https://idsync.rlcdn.com/455709.gif?partner\\_uid=459723377660%27%3Bp%3Dnew](https://idsync.rlcdn.com/455709.gif?partner_uid=459723377660%27%3Bp%3Dnew) (AV positives: 1/70 scanned on 07/02/2019 10:41:28)  
URL: [https://idsync.rlcdn.com/394499.gif?partner\\_uid=3011255410852](https://idsync.rlcdn.com/394499.gif?partner_uid=3011255410852) (AV positives: 1/70 scanned on 07/02/2019 00:21:54)  
URL: <http://ei.rlcdn.com/384946.gif?m=4c65c979095988a12d6f806ef0f89560&n=1> (AV positives: 1/70 scanned on 07/01/2019 14:15:01)  
URL: <http://ei.rlcdn.com/384946.gif?m=3f11d67e4a3cea06f43bf1eala063bfd&n=1> (AV positives: 1/70 scanned on 07/01/2019 12:25:57)

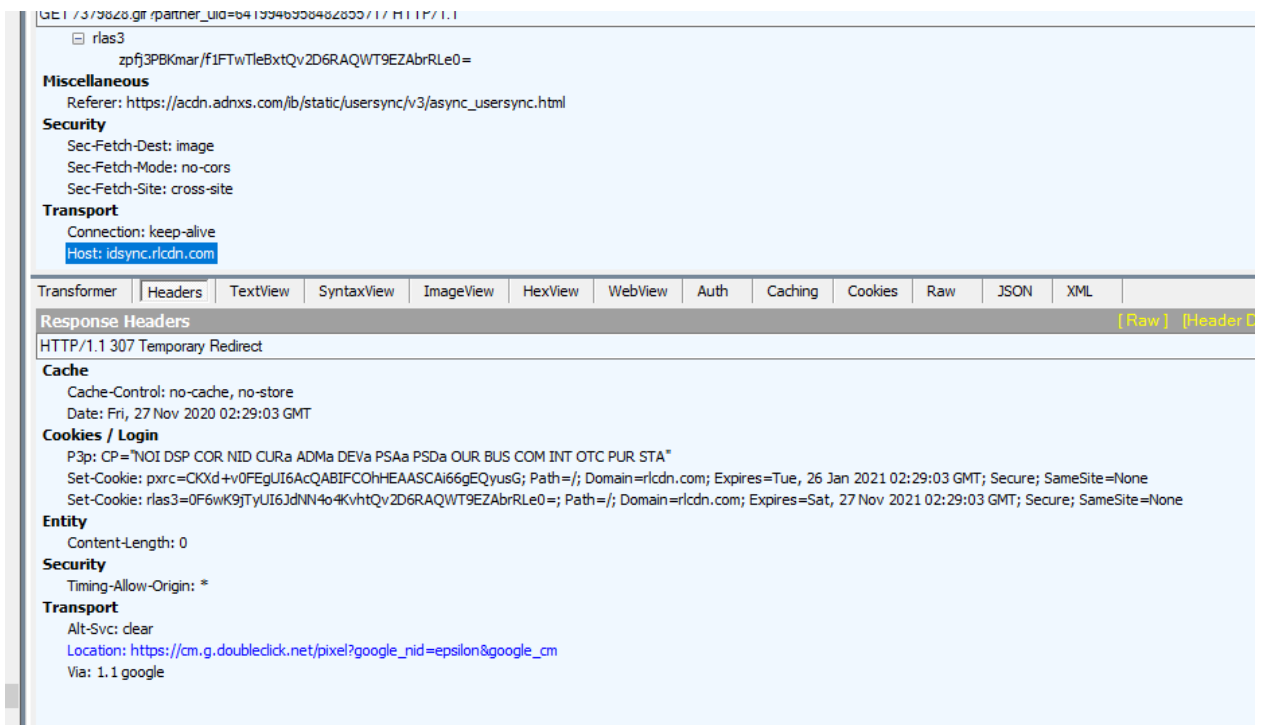


Figure 14: Screen shot shows that *Fiddler* was able to carve out the malicious domain and decode it from obfuscated network traffic.

# BATTLEFIELD MALWARE AND THE FIGHT AGAINST CYBER CRIME

The screenshot shows a malware analysis interface. At the top, the URL <http://idsync.rlcdn.com/> is displayed with a lock icon. Below it, a report summary states: "This report is generated from a file or URL submitted to this webservice on July 2nd 2019 13:43:43 (UTC) and action script *Default browser analysis*. Guest System: Windows 7 32 bit, Professional, 6.1 (build 7601), Service Pack 1. Report generated by Falcon Sandbox v8.30 © Hybrid Analysis".

On the right side, a red badge indicates the file is "malicious". Below this, the "Threat Score" is 86/100, "AV Detection" is "Marked as clean", and it is "Labeled as: Malicious site". There is also a "#tag" button.

A navigation bar includes buttons for "Overview", "Downloads", "External Reports", "Re-analyze", "Hash Not Seen Before", and "Request Report Deletion".

The main content area is titled "Incident Response" and contains two expandable sections:

- Risk Assessment**: Shows "Network Behavior" with the text "Contacts 1 domain and 1 host." and a "View all details" button.
- MITRE ATT&CK™ Techniques Detection**: Shows "This report has 2 indicators that were mapped to 4 attack techniques and 4 tactics." and a "View all details" button.

At the bottom right, there are social media sharing buttons for "Link", "Twitter", and "E-Mail".

Figure 15: Screen shot above shows the <https://dsync.rlcdn.com> has been flagged as malicious by malware analysis tools.

## Host: X.bidswitch.net

X.bidswitch.net is a domain that is associated with a variety of dubious activities on users' web browsers and computers. The URL has been found on a list of ad campaigns that focus on malvertising. Users can be exposed to pop-ups, in-text links, auto-play, JavaScript, and other types of malicious ads during their web browsing (Ni; Qian & Zhang, 2018).

The screenshot shows a network traffic capture from Fiddler. It displays the following details:

- tuuid\_lu=1606332070
- tuuid=8218ee1d-98e8-4f21-97c0-059bcbf345ce
- Miscellaneous**: Referer: <https://sync-amz.ads.yieldmo.com/tamptsync?callback=https%3A%2F%2Fs.amazon-adssystem.com%2Fecm3%3Fex%3Dym.com%26id%3D%24UID>
- Security**:
  - Sec-Fetch-Dest: image
  - Sec-Fetch-Mode: no-cors
  - Sec-Fetch-Site: cross-site
- Transport**:
  - Connection: keep-alive
  - Host: x.bidswitch.net

Figure 16: See screen shot shows the said domain as captured and decoded by *Fiddler*:

While the so-called X.bidswitch.net virus is usually related to ads and redirects that might be encountered on numerous sites, it might also indicate that some malicious

files or programs could have infiltrated the computer. Users reported that the anti-virus repeatedly attempts to block website's (*x.bidswitch.net/sync*, particularly) access to the PC (Monnappa, 2018).

One of the dangers of *X.bidswitch.net* is that it redirects to malicious websites can result malware infections or disclosure of personal information, monetary losses, installation of other potentially unwanted programs/malware (Yadav, 2019).

### **Case study conclusion: e-mail scam investigation**

The forgone demonstration shows that a malicious *JavaScript* specimen was embedded in an e-mail which was sent to users to lure them into clicking on what appears to be a fake Amazon digital gift card. If a user clicked on the gift card image, they would be redirected to an Amazon Web page which displays an error message and prompts users to log in to their Amazon account.

This is a redirection technique whereby cyber criminals clone the real Amazon login page to make users believe it's the real web page. Once a user clicks on the account login link and provides their Amazon login credentials, those credentials would be transmitted to a malicious domain and the user would get another error message telling them that their login credentials were incorrect.

This malicious JavaScript used obfuscation techniques to further complicate the analysis process, however, the researcher was able to deploy Fiddler as a specialized network monitoring and packet de-obfuscation tool and as a result was able to determine the malicious domains and hosts. The malicious JavaScript imbedded code that would allow it to contact about 72 domains and hosts.

## **CHAPTER II – Ransomware**

Ransomware is rapidly emerging as one of the top threat vectors for Cyberattacks. As perfect storm of Malware attacks, it is relatively easily available as a widely distributable attack toolkit, it can be controlled using difficult to trace methods on the Dark Web and the ransom demand is generally payable in nearly untraceable funding Bitcoin digital currency.

Ransomware is a growing threat faced by both businesses and individuals. This section will discuss what ransomware is, how it infects systems, and how to prevent infection. It will also examine some different variations of ransomware, as well as explore some cases of ransomware. Finally, a downloaded sample of *WannaCry* ransomware will be executed and examined for demonstration.

### **Introduction**

In this age of technology, almost everything is stored digitally. Data can be quickly accessed remotely, and physical degradation is no longer a concern. It requires less space and allows much more information to be kept for much longer. However, among all these positive elements, there is risk associated with storing data digitally.

What happens if suddenly all that data stored digitally is no longer accessible? Documents, family photos, intellectual property – none of it is immune to ransomware, should it strike. Unless this data is backed up, the ease and accessibility of digitally storing data could be its downfall. Users must be prepared and cautious to protect their data.

### **What is Ransomware**

Ransomware is a type of malicious software, or malware that infects a system and demands a ransom for system restoration (Oberly, 2019). It comes in three varieties: those

that lock the screen, those that make threats but do not actually do anything, and those that encrypt the files, which is the most common (Allen, 2017).

In each case, the user is presented with the option of paying a ransom in order to regain access to their system and data or risk losing it. If a user does not have backups of their data, they may be forced to pay whatever fee is demanded, which can range from hundreds of dollars for a personal computer to millions of dollars for a large-scale company network (Oberly, 2019). Not only do companies suffer monetary loss, but their business is also affected when it becomes public knowledge that their sensitive data has been compromised and their security breached (Oberly, 2019). Although it is less common, some forms of ransomware mimic encrypting ransomware, but merely change file names to make them seem encrypted (Hachman, 2017).

### **Infection**

The most common technique used by attackers to compromise systems is phishing (Oberly, 2019). Phishing is the fraudulent practice of sending emails claiming to be from reputable companies to induce individuals to reveal personal information or download malicious content.

An example of a phishing email can be seen in Figure 17, along with indicators of suspicious content. Ransomware may also infect a system through the downloading of seemingly benign files that contain embedded malware (Oberly, 2019). Attackers may embed malware in the metadata of an otherwise harmless file. This does not change the file size or raise any other flags but can have a catastrophic effect. Unauthorized access to a computer is another way that malicious code could infect a machine. For example,



leaving an unprotected computer around strangers could lead to infection, as it may not be obvious who has malicious intentions (Allen, 2017).

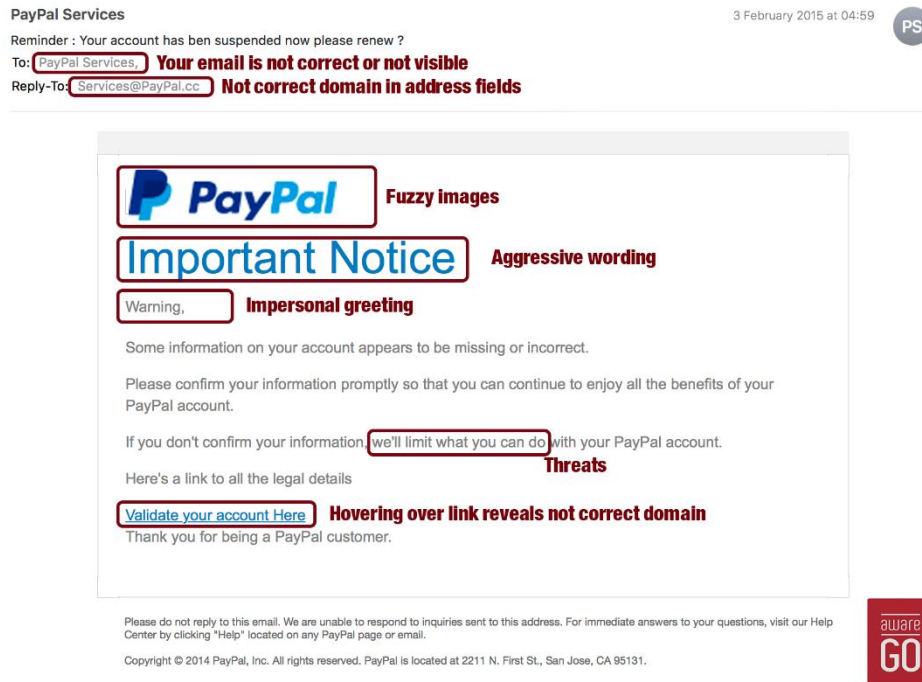


Figure 17: Example of a phishing email, retrieved from: <https://cdn2.hubspot.net/hubfs/3950430/email-1.png>

## 2.1. Ryuk Ransomware

One particular ransomware took the state of Florida host in 2019. Ryuk was the ransomware that took Florida hostage and was successful in getting about a million dollars from some cities in Florida. The Ryuk ransomware is deployed by a phishing email or visiting a sketchy website and clicking the link where bots are then able to access the network. Once the bots have access to the network they then use lateral movement to start encrypting machines and locking users out.

The attackers caused so much havoc that it took 911 offline in Riveria beach, some police officers were writing paper tickets, healthcare servers were compromised, criminal evidence was erased and about a year and half of digital evidence was lost letting six narcotics dealers walk free.

Preventing ransomware can be just as easy as preventing any other attack. Understanding that clicking links that you are not expecting is probably not the best thing that our end users could do. Also, Norton has some tips for keeping safe from ransomware. One thing that Norton suggests is do not pay the ransom. They suggest this because if you are doing proper backups of your data then you can easily recover from these attacks and, it *“encourages and funds these attackers and even if the ransom is paid, there is no guarantee that you will be able to regain access to your files”* (Norton, 2018).

Another good strategy in keeping systems safe from malware is to ensure that proper patching is happening. Keeping your antivirus definition up to date as well as applying the latest operating system patches will go a long way in keeping your systems secure. It is also important to note that *“exploit kits hosted on compromised websites are commonly used to spread malware”* (Norton, 2018). The IT department should also talk with leadership about working to get an email scanner installed to ensure that any links or attachments that are in inbound emails are safe and malicious ones are dropped.

### **2.2. Prevention**

One of the most important elements of preventing ransomware infection is user training and awareness. Humans are often the weakest link of security, so it is important that network users are educated on the dangers of ransomware and on common infection methods, such as following unknown links (Oberly, 2019). It is always a good practice to type out links instead of following them, as this can help avoid subtle link differences that

may otherwise have gone unnoticed (Mohurle and Patil, 2017). For example, typing out [www.google.com](http://www.google.com) could help prevent infection from a link such as [www.google.com](http://www.google.com), which may lead to a malicious site or download malware. Resources exist that examine links or files for malicious content, such as <https://www.virustotal.com/gui/home/upload>. Educating users on potential risks and corresponding safeguards is an important part of ransomware prevention.

There are several policies that can be established by companies for their employees to safeguard against ransomware attacks. Companies should draft policies that define expectations for users concerning areas such as personal email and file-sharing programs on the company network (Oberly, 2019). All users should also have strong passwords for their devices and keep these passwords secret; posting passwords on monitors to remember them is even worse than having no password, as it lulls the user into a false sense of security (Allen, 2017). A strong password should have at least eight characters and be a mix of uppercase and lowercase letters, numbers, and symbols. Another basic but very important policy that should be employed to minimize risk is the principle of least privilege. It entails only giving users access to what they require to do their job; administrative access should only be given when necessary and revoked when no longer needed (Oberly, 2019).

This minimizes the risk of system damage if a user is compromised, as they will have limited power. Along similar lines of restricted privileges, policies should be employed that limit user access to common grounds of infection, such as Facebook, Instagram, and other social media sites (Oberly, 2019). Users should also disable wireless connections such as Bluetooth and Wi-Fi when not needed, and ensure that the network is secure before connecting (Mohurle and Patil, 2017). However, these policies will have

no effect if they are not enforced, so companies should regularly have external audits done to test their users. By crafting policies such as these for their employees, companies can minimize their risk of ransomware infection.

In addition to policies for the employees, general company policies can be established to protect against ransomware attacks. As ransomware also takes advantage of known security vulnerabilities in systems, it is very important to regularly update and patch systems as soon as possible. If systems and applications are not patched, they have a much higher risk of infection, as the vulnerabilities are known to attackers and provide them with an obvious opportunity (Oberly, 2019). According to Oberly (2019), *“If a program or device is too old to update, retire it, as the limited range of tasks that it can perform is vastly outweighed by the risk that the outdated technology presents to the company’s computer networks and systems, which can provide attackers with vulnerable entry points to launch a ransomware attack”*.

Another method of prevention that can be employed by a company is whitelisting software. Whitelisting creates a list of acceptable and trusted applications, executable files, and connections. The system allows these to run, while blocking any software or port not on the list. Antivirus software and firewalls are other basic requirements that every company should include in their systems (Hachman, 2017). Company-wide policies such as these are essential to mitigate possible infections.

### **2.3. Recovery**

The best option for combating ransomware is to ensure that the company policy is set up to avoid having to pay the ransom, as even if the ransom is paid there is no guarantee that data will be restored (Allen, 2017).

One way to do this is to have off-network backups of data. All data should be backed up regularly, with essential data being backed up daily if necessary. Several copies of the entire system should be stored at different locations to allow for a complete reloading of the system (Allen, 2017).

Copies may be stored at different physical locations on a variety of media, including servers, hard drives, or solid-state drives (SSDs). Additionally, companies should have a backup that is a few weeks old. This may be necessary if ransomware remains undetected for a period of time, propagating throughout the network before launching all at once.

A backup that is a few weeks old will not contain the most up-to-date data, but it should predate any infection. A prudent approach is to keep both recent and older backups. Even with regular back-ups, however, any data created since the last backup will be lost (Allen, 2017).

#### **2.4. Recent Prominent Variants**

The following are a few examples of notorious strings of ransomware. Ransomware first appeared in 1989 with the **AIDS** Trojan. This ransomware attack failed overall because the ransom could be bypassed by extracting the encryption key from the trojan code. Its creator, Joseph Popp, was caught and ended up donating the money he had managed to capture to *AIDS* research, hence the attack's name. Since this time, attacks have become much more sophisticated and successful. The percentage of organizations affected by ransomware has dropped significantly, but the gains in revenue for attackers have remained largely the same.

**SimpleLocker** was a ransomware attack that surfaced in 2015. This attack was the first android platform attack to actually encrypt file, as opposed to blocking the user from

the user interface like previous attacks. *SimpleLocker*'s payload was in the trojan downloader, but it affected a relatively small portion of Android users due to the fact that the infection came from downloading sketchy apps from outside of the Google Play store. This attack originated in Eastern Europe, but most of the targets were in the United States.

**SamSam** was another attack that had Eastern Europe origins but targeted the US. This attack took place in 2016 and offered ransomware-as-a-service, where controllers carefully probed targets for weaknesses and then exploited the vulnerabilities that they found. Once this attack made its way into the system, it quickly escalated privileges and encrypt files.

Just a year later, the **WannaCry** ransomware attack appeared. This attack was a cryptoworm, encrypting data and spreading to other machines via port 445, or the *Server Message Block* (SMB) port. *WannaCry* utilized *EternalBlue*, an exploit discovered by the NSA that took advantage of a defect in Microsoft's implementation of SMB protocol. Although a patch was released from Microsoft on March 14 (2017), many people had not yet installed it and had the SMB port 445 open. This allowed the ransomware to spread easily, as it didn't need user interaction but traveled across servers via 445 port. This attack ended after four days, as Marcus Hutchins discovered a kill switch within the malware that greatly reduced further compromise.

**NotPetya** was a ransomware that exploited the same *EternalBlue* vulnerability as *WannaCry*, taking place only weeks after. This attack was primarily focused on Ukraine, targeting energy companies, power grids, bus stations, gas stations, airports, and banks. A Ukrainian tax preparation program, M.E. Doc, was compromised to spread the malware, and investigation afterward revealed that software updates had not been applied on these

servers since 2013. The backdoor utilized in this attack was present nearly six weeks before, showing this attack was well planned. A “vaccine” for this threat was developed, as Amit Serper found that before encryption, *NotPetya* would search for its own filename. Users could create the read-only file *perf.c*, and the ransomware would think the machine was already infected and not execute. This attack was further stopped when the attacker’s email service provider shut down their email account, preventing them from receiving payment confirmation.

One of the more recent ransomware attacks is **Ryuk**, which was common from 2018 to 2019. It was derived from *Hermes* ransomware and specifically targeted organizations with little tolerance for downtime. This attack worked by disabling the *Windows System Restore* to minimize chances of users getting around the ransom, and higher ransoms were demanded from higher-value targets. This attack did not run on computers with Russian, Belarusian, or Ukrainian set as their language, suggesting that it was of Russian origin. These are only a few examples of the multifarious variations of ransomware that have targeted users and companies.

### 2.5. Ransomware Cases

There have been countless cases of ransomware, with targets ranging from personal computers to government institutions. Medical institutions such as hospitals are a common target, as they have sensitive and vital information that they normally cannot afford to lose. Hollywood Presbyterian Medical Center (HPMC) was one such example. HPMC was stricken with ransomware and had to pay \$17,000 to restore critical data and communication capabilities. HPMC paid the ransom before reaching out to the authorities, wanting to restore systems as quickly as possible. Conversely, Ottawa

Hospital experienced a ransomware attack that affected over 9,800 machines, but they did not pay the demanded ransom. This hospital had extensive back-ups, so they were able to wipe all the hard drives and restore the system from their back-ups. Kentucky Methodist Hospital, Chino Valley Medical Center, and Desert Valley Hospital were similar cases. These hospitals had their files, images, and documents encrypted and renamed with the *.locky* extension by the ransomware, but they were able to successfully restore their systems from their backups.

In addition to medical institutions, entire cities may be targeted by ransomware attacks. In 2016, the *San Francisco Municipal Transportation Agency* (SFMTA) experienced a ransomware attack that disrupted their train ticketing and bus management systems, with a \$73,000 ransom demand for restoration of their system. SFMTA did not end up paying the ransom, thanks to a speedy response and comprehensive backup process that allowed SFMTA to restore their systems in two days. However, SFMTA did not escape unharmed, as during the two-day restoration process, all the passengers were able to ride for free. Atlanta is another city that experienced a ransomware attack. A January 2018 audit found 1,500-2,000 vulnerabilities in the city's infrastructure, and in March, it fell to a brute force *SamSam* attack. One third of all software programs in the city were affected by this attack. Atlanta is an economic and transportation hub, and utilities, parking, court services, and police services were all affected. Many police and court documents were permanently deleted, but the police department was able to restore all of its investigation files. The city took many services offline to recover from the attack, and for a short time, all payments and bills had to be in paper format until systems could be restored.

### **2.6. Demonstration Methods**



To investigate the behavior of ransomware, this project used a virtual machine, *VMware*, to download and run a sample of ransomware. In the virtual machine, a *WannaCry* ransomware sample was downloaded from *GitHub* and uploaded to the *any.run* website.

This site is an interactive malware analysis tool that allows the analysis of Windows programs, scripts, and other files in a full sandbox environment. *Any.run* has a free community version, which was used in this project, and paid plans with extended features. The community version supports a Windows 7 32-bit virtual environment and includes detailed process information, as well as network information, files, and debug functions.

When the ransomware sample is uploaded to this site, it is virtually executed, showing how the file would affect a regular system. The process information pane tracks any processes that are executed or created and offers a threat analysis on each. The file was executed and allowed to run for 60 seconds to collect information, which was examined in detail.

### Demonstration

The ransomware sample chosen was a *WannaCry* sample downloaded from <https://github.com/fadyosman/WannaCrySample/network/members>, in a file called *wannacry.exe*. The file was downloaded inside the *SIFT distro VMware* and uploaded to the *any.run* website.

After executing for 60 seconds, the file completely encrypted the virtual desktop and displayed a ransom note. Below are the four stages of the desktop encryption. As can

be seen, the ransomware was able to fully execute and take over the host system within 60 seconds.



Figure 18: Initial state of desktop

Figure 18 provides an initial state of the Windows desktop, while Figure 19, presents the same desktop after 60 seconds. Several more files (14) are start to be visible now. Additionally, several more actions will take place, as documented in the following figures.



Figure 19: Additional files added to desktop

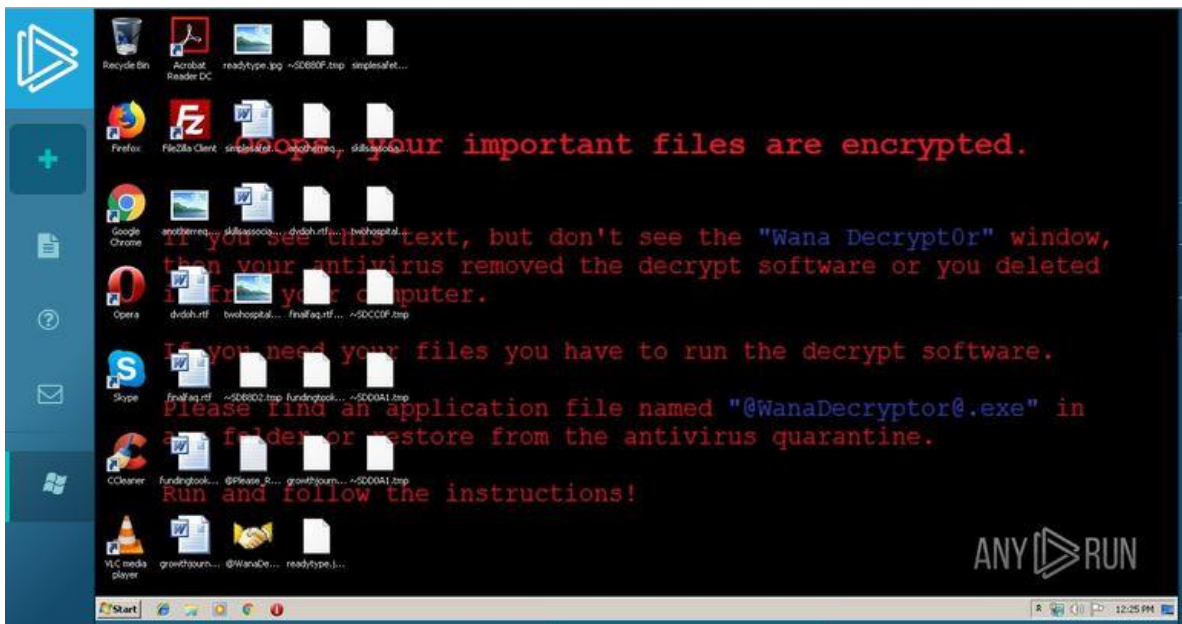


Figure 20: Encryption message set as background



Figure 21: Ransom note displayed on screen with payment instructions

Following the 60 seconds of execution, *any.run* offers a variety of information regarding the process just executed. As can be seen below, information regarding network connection, threats, files, processes executed, and more are readily available.

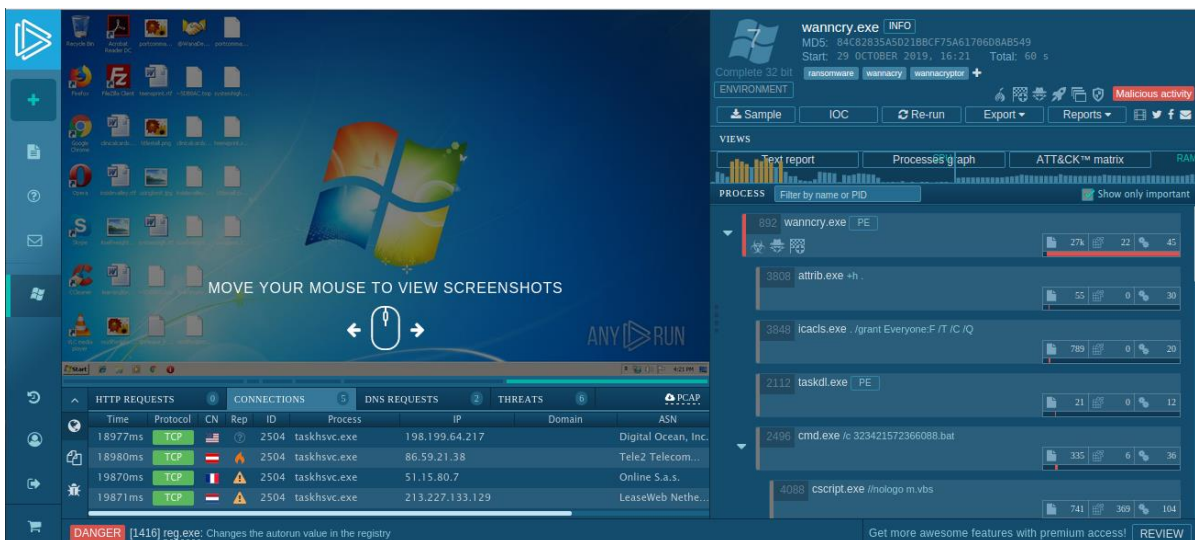
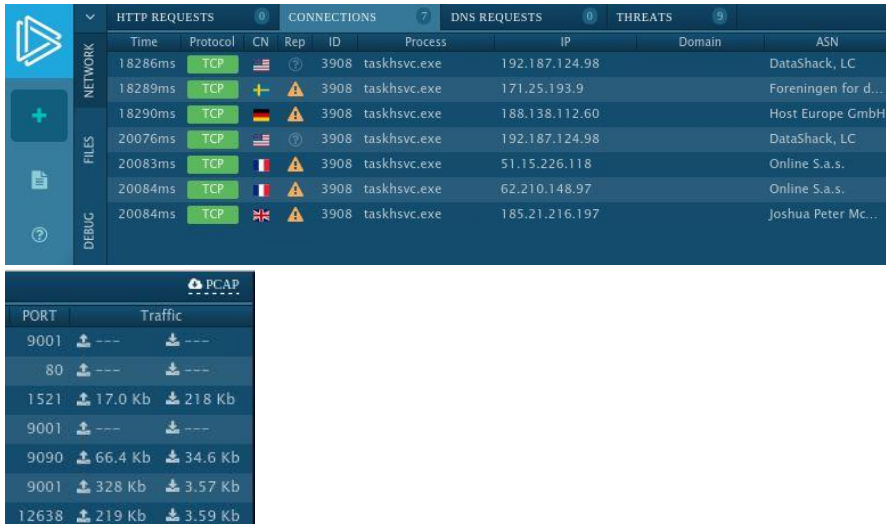


Figure 22: Overview of *any.run* analysis



## BATTLEFIELD MALWARE AND THE FIGHT AGAINST CYBER CRIME

The network tab beneath the desktop offers information regarding connections made by this process. Ports 9001, 9090, 1512, and 12638 are all TCP/UDP port, and port 80 allows Internet connection.



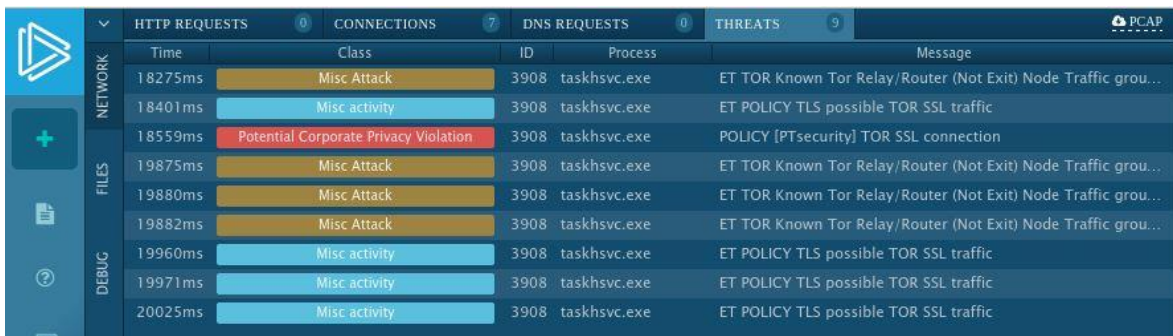
CONNECTIONS (7)									
Time	Protocol	CN	Rep	ID	Process	IP	Domain	ASN	
18286ms	TCP			3908	taskhsvc.exe	192.187.124.98	DataShack, LC		
18289ms	TCP			3908	taskhsvc.exe	171.25.193.9	Foreningen for d...		
18290ms	TCP			3908	taskhsvc.exe	188.138.112.60	Host Europe GmbH		
20076ms	TCP			3908	taskhsvc.exe	192.187.124.98	DataShack, LC		
20083ms	TCP			3908	taskhsvc.exe	51.15.226.118	Online S.a.s.		
20084ms	TCP			3908	taskhsvc.exe	62.210.148.97	Online S.a.s.		
20084ms	TCP			3908	taskhsvc.exe	185.21.216.197	Joshua Peter Mc...		

PORT	Traffic
9001	--- / ---
80	--- / ---
1521	17.0 Kb / 218 Kb
9001	--- / ---
9090	66.4 Kb / 34.6 Kb
9001	328 Kb / 3.57 kb
12638	219 Kb / 3.59 Kb

Figure 23: Connections tab

*Any.run* also contains a threat tab that displays all the threats related to *wannacry.exe*.



THREATS (9)					
Time	Class	ID	Process	Message	
18275ms	Misc Attack	3908	taskhsvc.exe	ET TOR Known Tor Relay/Router (Not Exit) Node Traffic grou...	
18401ms	Misc activity	3908	taskhsvc.exe	ET POLICY TLS possible TOR SSL traffic	
18559ms	Potential Corporate Privacy Violation	3908	taskhsvc.exe	POLICY [PTsecurity] TOR SSL connection	
19875ms	Misc Attack	3908	taskhsvc.exe	ET TOR Known Tor Relay/Router (Not Exit) Node Traffic grou...	
19880ms	Misc Attack	3908	taskhsvc.exe	ET TOR Known Tor Relay/Router (Not Exit) Node Traffic grou...	
19882ms	Misc Attack	3908	taskhsvc.exe	ET TOR Known Tor Relay/Router (Not Exit) Node Traffic grou...	
19960ms	Misc activity	3908	taskhsvc.exe	ET POLICY TLS possible TOR SSL traffic	
19971ms	Misc activity	3908	taskhsvc.exe	ET POLICY TLS possible TOR SSL traffic	
20025ms	Misc activity	3908	taskhsvc.exe	ET POLICY TLS possible TOR SSL traffic	

Figure 24: Threats tab

The Process tab on the left displays a list of all the processes executed by the file, a total of 24 in this sample.

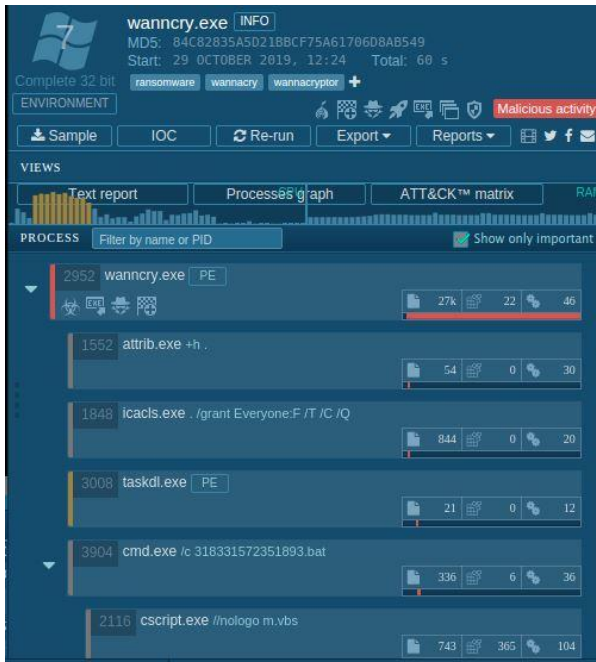


Figure 25: Process Tab

By clicking on a particular process, more information can be seen. The first process, *wannacry.exe*, was flagged by *any.run* as strongly malicious. It correctly flagged this as *WannaCry* ransomware, and displayed a long list of all the files modified by this process. Some files were newly created, and others were encrypted. This section also displayed the children processes of *wannacry.exe*, which offered a good place to look for other malicious behavior.

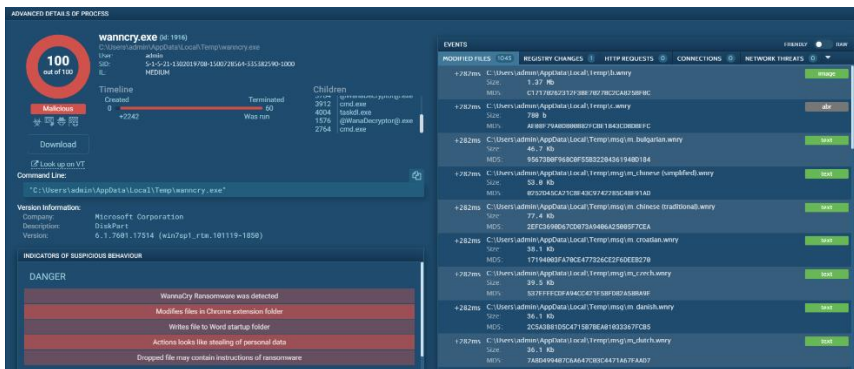


Figure 26: *wannacry.exe* advanced details pane

An important section to examine here is the “*Indicators of Suspicious Behavior*” (ISB) section. It contains Danger, Warning, and Info sectors for the varying levels of concern. Below are the suspicious behavior sections of *wannacry.exe*.

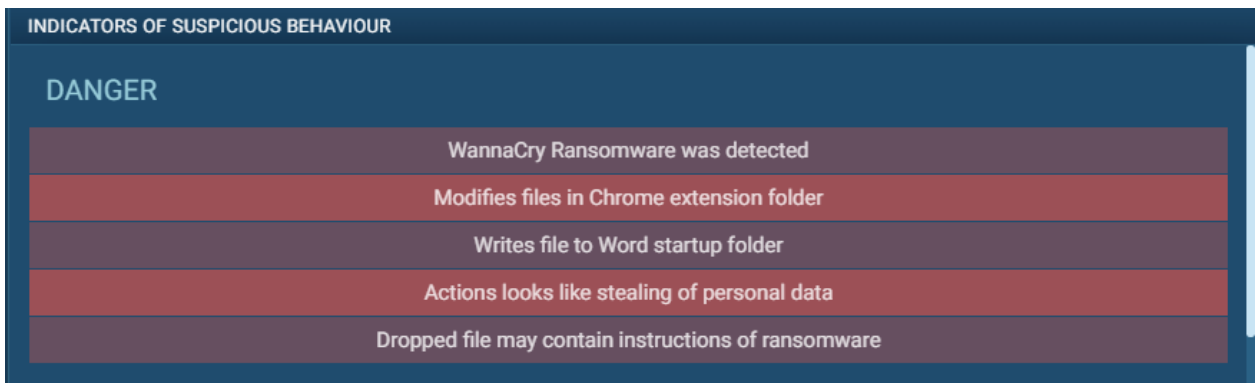


Figure 27: *wannacry.exe* suspicious activities pane ISB Danger

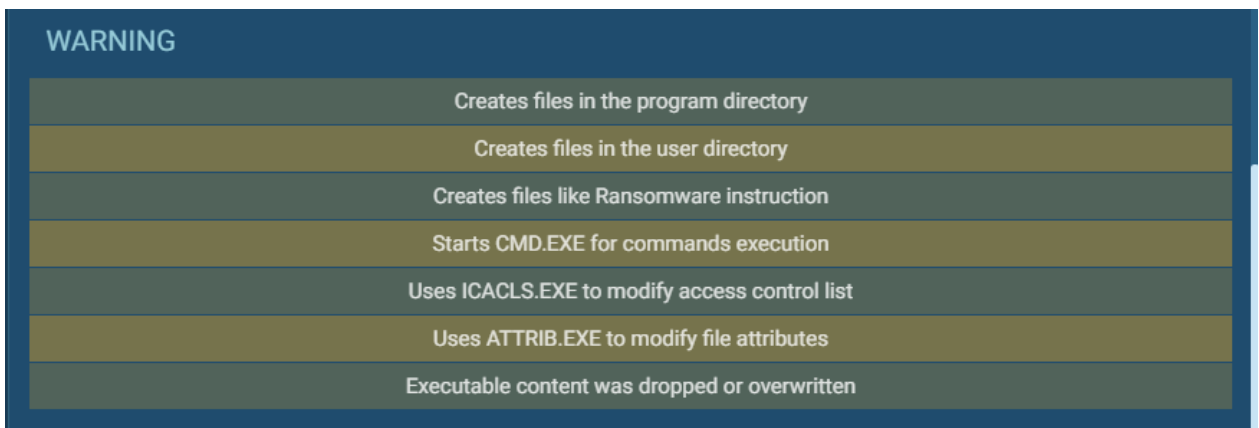


Figure 28: *wannacry.exe* suspicious activities pane ISB Warning

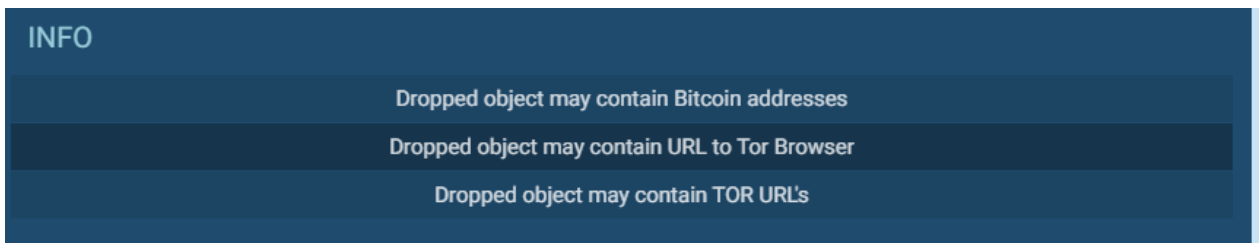


Figure 29: *wannacry.exe* suspicious activities pane ISB Info

Examining the children processes of wannacry.exe gives more insight into its behavior. Not every process is malicious, but upon examination, many were found to be malicious. Below is the layout of processes and their children, with child processes indented.

- A. 1916 wannacry.exe
  - 1. 3752 attrib.exe
  - 2. 3992 icacls.exe
  - 3. 912 taskdl.exe
  - 4. 2524 cmd.exe
    - a. 4072 cscript.exe
  - 5. 3784 @WanaDecryptor@.exe
    - a. 1520 taskhsvc.exe – modifies 13 tor files
  - 6. 3912 cmd.exe
    - a. 2144 @WanaDecryptor@.exe
      - i. 2040 cmd.exe\*
        - 1. 2712 vssadmin.exe – deletes shadow copies
        - 2. 2884 WMIC.exe – deletes shadow copies
        - 3. 2384 bcdedit.exe – disables recovery
        - 4. 2716 bcdedit.exe – disables recovery
        - 5. 2700 wbadmin.exe – deletes catalog
  - 7. 4004 taskdl.exe
  - 8. 1576 @WanaDecryptor@.exe – modifies bitmap (ransom note)
  - 9. 2764 cmd.exe
    - a. 3856 reg.exe – changes the autorun value in the registry

By examining the children processes of wannacry.exe, cmd.exe was another malicious process. This process was also flagged as highly malicious, upon examination was found to delete shadow copies of files to prevent the recovery of data.



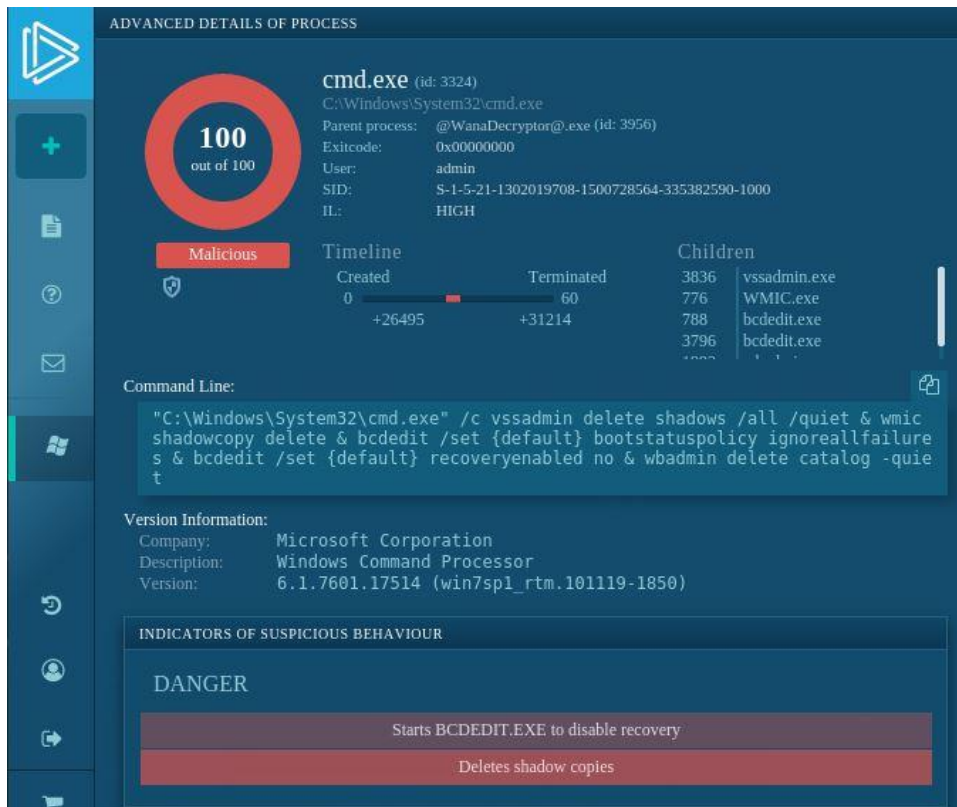


Figure 14. *cmd.exe* advanced details panel

As can be seen, once ransomware has been downloaded to a machine and executed, it works very quickly. In a span of only 60 seconds, this *WannaCry* sample was able to completely lock down the virtual desktop. It did so by creating processes to encrypt files and to delete shadow copies of files to prevent recovery.

It is also able to change registry items to edit what happens on boot-up, so rebooting the system does not bypass the ransomware. Additionally, it is interesting to note that these abilities require administrator-level access, so the ransomware sample was able to escalate its privileges.

## CONCLUSION

Information technology has forever changed the way we live and work and there is no doubt about the fact that the world has benefited from technological advancements in ways that are immeasurable and never imagined before. However, these technological advancements are not risk-free and there is a flip-side to this: cybercriminal activities have skyrocketed in the recent years to the point where in some cases hackers have been able to take business organization as hostage using malware.

Cyber-crimes have become a multi-billion-dollar industry in the recent years. Most cybercrimes/attacks involve deploying some type of malware. Malware that viciously targets every industry, every sector, every enterprise and even individuals has shown its capabilities to take entire business organizations offline and cause significant financial damage in billions of dollars annually. Malware authors are constantly evolving in their attack strategies and sophistication and are developing malware that is difficult to detect and can lay dormant in the background for quite some time in order evade security controls].

The number of malware samples detected in the wild has been consistently growing over the past couple of years. According to research by McAfee labs, more than 1,224,628 malware threats were detected in Q4 2020, this includes a total of 7,899 unique new hashes. This necessitates stepping up the game in the fight against malware detection and prevention especially because cyber-hackers are developing new variants of malware every-day.

As a result of this, malware classification and detection has become a major area of concern for both academia and the industry alike. Traditional malware detection techniques are based on a data-base of known signatures thus are unable to detect new variants of malware. Additionally, reverse-engineering malware and performing static or dynamic analysis is a task that takes a significant time and effort and it can't scale nor keep pace with the surge in malware variants that we see running in the wild every-day. To make matters worse, malware authors use creative detection -evasion techniques such as packers and obfuscation to by-pass traditional security controls. Consequently, an effective approach to tackle this mountainous task is necessary. One such approach would be to apply deep learning models to efficiently classify and detect malware.

The results of our malware analysis demonstrate that malware authors are smart and creative and that they develop new malware every day. We also noticed from our results that many malware authors actually utilize encryption in order to protect their communication and make it harder for malware analysts to analyze malware samples. It's clear that malware is becoming more advanced, more destructive, and is capable of evading traditional anti-virus programs. Even code-analysis techniques of malware can't keep pace with the scale and sophistication of today's malware threat landscape therefore classifying and visualizing malware using deep learning models will provide significant performance gains as well as the advantage of accuracy because deep learning algorithms generally provide 95-99% accuracy of classifying and detecting malware. We therefore believe that we should harness the power of artificial intelligence and machine learning in order to build intelligent and proactive malware detection systems. Additionally, we recommend that businesses start new initiatives such as providing cyber security as a

service in order to utilize the power of machine learning to detect emerging malware threats and stop hackers in their own tracks.

The analysis and results of this report have been published as a book chapter in the upcoming IGI Global book entitled “*Cybersecurity Capabilities in Developing Nations and Its Impact on Global Security*”.

## BIBLIOGRAPHY

Afianian, A.; Niksefat, S.; Sadeghiyan, B. & Baptiste, D. (2020). *Malware dynamic analysis evasion techniques: a survey*. *ACM Computing Surveys*, 52(6), 1–28. DOI: 10.1145/3365001.

Allen, J. (2017). Surviving Ransomware. *American Journal of Family Law*, 31(2), 65–68. Retrieved from <http://search.ebscohost.com.saintleo.idm.oclc.org/login.aspx?direct=true&db=a9h&AN=123206569&site=ehost-live&scope=site>, accessed in [01/11/2021].

Bat-Erdene, M.; Park, H.; Li, H.; Lee, H. & Choi, M. S. (2017). Entropy analysis to classify unknown packing algorithms for malware detection. *International Journal of Information Security*, 16(3), 227–248. DOI: 10.1007/s10207-016-0330-4.

Carlin, D.; O’Kane, P. & Sezer, S. (2019). A cost analysis of machine learning using dynamic runtime opcodes for malware detection. *Computers & Security*, 85, 138–155. DOI: 10.1016/j.cose.2019.04.018.

Carrillo-Mondejar, J.; Castelo, G. J. M.; Nunez-Gomez, C.; Roldan, G. J. & Martinez, J. L. (2020). Automatic analysis architecture of IoT malware samples. *Security and Communication Networks*, 2020. DOI: 10.1155/2020/8810708.

CISA (2019) *Ransomware Outbreak*. CISA INSIGHTS. Retrieved from [https://www.us-cert.gov/sites/default/files/2019-08/CISA\\_Insights-Ransomware\\_Outbreak\\_S508C.pdf](https://www.us-cert.gov/sites/default/files/2019-08/CISA_Insights-Ransomware_Outbreak_S508C.pdf), accessed in [01/11/2021].

Feizollah, A.; Anuar, N. B.; Salleh, R.; Suarez-Tangil, G. & Furnell, S. (2017). Androdialysis: analysis of android intent effectiveness in malware detection. *Computers & Security*, 65, 121–134. DOI: 10.1016/j.cose.2016.11.007.

Hachman, M. (2017). *How to remove ransomware: Use this battle plan to fight back*. *PCWorld*, 35(4), 129–135. Retrieved from <http://search.ebscohost.com.saintleo.idm.oclc.org/login.aspx?direct=true&db=a9h&AN=122359473&site=ehost-live&scope=site>, accessed in [01/11/2021].

Kirubavathi, G., & Anitha, R. (2018). Structural analysis and detection of android botnets using machine learning techniques. *International Journal of Information Security*, 17(2), 153–167. DOI: 10.1007/s10207-017-0363-3.

Lim, J., & Yi, J. H. (2016). Structural analysis of packing schemes for extracting hidden codes in mobile malware. *Eurasip Journal on Wireless Communications and Networking*, 2016(1), 1–12. DOI: 10.1186/s13638-016-0720-3.

Mohurle, S. & Patil, M. (2017). A brief study of wannacry threat: Ransomware attack 2017. *International Journal of Advanced Research in Computer Science*, 8(5) Retrieved from <https://saintleo.idm.oclc.org/login?url=https://search.proquest.com/docview/1912631307?accountid=4870>, accessed in [01/11/2021].

Monnappa, K. A. (2018). *Learning Malware Analysis: explore the concepts, tools, and techniques to analyze and investigate Windows malware*. Packt Publishing.

No Author. (2018, January 18). *7 tips to prevent ransomware*. Retrieved December 08, 2020, from <https://us.norton.com/internetsecurity-malware-7-tips-to-prevent-ransomware.html>, accessed in [01/11/2021].

Oberly, D. J. (2019). Best Practices for Effectively Defending Against Ransomware Cyber Attacks. *Intellectual Property & Technology Law Journal*, 31(7), 17–20. Retrieved from <http://search.ebscohost.com.saintleo.idm.oclc.org/login.aspx?direct=true&db=iih&AN=137351116&site=ehost-live&scope=site>, accessed in [01/11/2021].

Oza, S. (2020, January 24). *Ryuk Ransomware – Malware of the Month*, January 2020. Retrieved December 08, 2020, from <https://securityboulevard.com/2020/01/ryuk-ransomware-malware-of-the-month-january-2020/>, accessed in [01/11/2021].

Stiborek, J.; Pevný, T. & Reháč, M. (2018). Probabilistic analysis of dynamic malware traces. *Computers & Security*, 74, 221–239. DOI: 10.1016/j.cose.2018.01.012.

Sussman, B. (2020, February 27). *Ryuk Ransomware Attack Sets Accused Criminals Free*. Retrieved December 09, 2020, from <https://www.secureworldexpo.com/industry-news/ryuk-ransomware-attack-sets-criminals-free>, accessed in [01/11/2021].

Ul Haq, I.; Chica, S.; Caballero, J. & Jha, S. (2018). Malware lineage in the wild. *Computers & Security*, 78, 347–363. DOI: 10.1016/j.cose.2018.07.012.

Xue, D.; Li, J.; Wu, W.; Tian, Q. & Wang, J. (2019). Homology analysis of malware based on ensemble learning and multifeatures. *Plos One*, 14(8), 0211373. DOI: 10.1371/journal.pone.0211373.

Yadav, R. M. (2019). Effective analysis of malware detection in cloud computing. *Computers & Security*, 83, 14–21. DOI: 10.1016/j.cose.2018.12.005.

Yadav, R. M. (2019). Effective analysis of malware detection in cloud computing. *Computers & Security*, 83, 14–21. DOI: 10.1016/j.cose.2018.12.005.

Zhang, X. & Song, X. (2020). Stability analysis of a dynamical model for malware propagation with generic nonlinear countermeasure and infection probabilities. *Security and Communication Networks*, 2020, 1–7. DOI: 10.1155/2020/8859883.

**LITERATURE PRODUCTION WITHIN THE PROGRAMME**

Marwan Omar & Prof. Luis Borges Gouveia (2021) “Reverse-engineering Malware”.  
Book chapter accepted for publication in IGI-Global Book titled: Cybersecurity  
Capabilities in Developing Nations and Its Impact on Global Security. Release date:  
December, 2021