2020 IEEE International Conference for Innovation in Technology (INOCON)
Bengaluru, India. Nov 6-8, 2020

Developing Malware and Analyzing it Afore & After Steganography with OSINTs

Mrinal Kaushik
Cyber Forensics Department
National Institute of Criminology and Forensic Science
Delhi, India
mrinalkaushik77@gmail.com

Monica Malik
IT Department
Indira Gandhi Delhi Technical University for Women (IGDTUW)
Delhi, India
monicamalik17@gmail.com

Bhawna Narwal
IT Department
Indira Gandhi Delhi Technical University for Women (IGDTUW)
Delhi, India
bhawnanarwal@igdtuw.ac.in

***Abstract*— Malicious software aka Malware is one of the stern threats that has been around ever since the computers were developed. Malware is designed by malicious attackers whose intension is to bring down the system or to take over the system. It is very difficult to detect unknown malware as compared to known ones. Steganography tools allow users for embedding the hidden data inside the carrier file like video, audio, or an image, and later data is extracted. Here, malware is hidden inside an image by attackers to manipulate the victims and it can be done via phishing technique which is one of the famous and well-known practices done by the attackers. This paper provides an overview of creating malware using Metasploit in the Parrot OS (Operating System) which is a secure OS, later the Steganography tool is used to manipulate the victim and the victim's system is taken over by the attacker. Also, a summarized analysis is given on the Open Source Intelligence Tools (OSINT) and which one gives the best results.**

***Keywords*— OSINT, cyberattacks, malicious software, malware, metasploit, security.**

I. Introduction

Malware has come from malicious software, which is used to disrupt the system, gain access to the system or gain sensitive information. Over the past few years, there is a significant increase in the number of cases related to malware. Malware is one of the biggest threats to the IT industry from the past few years and it is going to be around. Even after using malware detection techniques like antiviruses, still, there is a consequential increase in malware. Malware is untraceable as attackers use obfuscation techniques to build malware to beat the signature-based malware detection systems. This era is full of cyber threats and there is no escape from them but still, if preventive measures are taken then one can save a lot from the loss.

Many researches have been done related to malware detection and many presented a review of many tools and techniques for detecting and then analysing the executable malware file [1, 18-20]. Diffusion of malware has infected almost everyone's life, whether it is social media platform [2], politics [3], healthcare [4], industry [5], institutes [6], etc. Many researchers have researched on the analysis of the malware where they used deep learning [7-8], malware metadata [9], machine learning [10], reverse engineering [11], and guessing signatures [12], etc.

Self-malware is developed using Metasploit [13] in Parrot OS. That malware is bound with an image, that image can be of the victim's interest which will help to manipulate the victim easily, this is known as steganography technique [14] which allows to hide the secret data inside a common file such as an image. Then, via mail or external drive it can be sent as an image to the victim, this is known as phishing technique [15] which is a very well-known way to harm people, to obtain data via unethical way and to take over the devices, or to obtain the information about the device or the victim. When the victim clicks on the malware it is downloaded on the victim's machine. This is the simplest way to get into the victim's machine and the attacker takes control, gain information, and misuse the obtained information.

Spear phishing [16] is also one of the kinds where a person or a person with a high position in an organization is targeted to obtain crucial information about an organization. That is why there are security teams nowadays in big companies and they take a lot of preventive measures and defensive as well [21-24].

Here in this work, malware is developed, and using that malware the victim's machine is exploited. Later, an analysis is done via freely available OSINTs [17] to find out which is the best OSINT that gives the best results. OSINTs have been very helpful in identifying whether the URLs and files are malicious or not. If before running those files on the system are checked then it can help many from losing important information. But how binding an image with the malware makes it difficult for OSINTs to identify they are malicious or not is being analysed in this work.

II. Creating and Validating the Malware

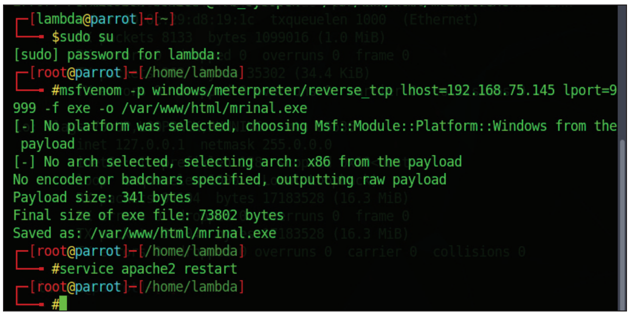
In this section, a brief overview is given on how malware is created and how it is used to exploit the victim's machine. Later the same malware is analysed on different freely available OSINTs.

A. *Developing the Malware*

Here, malware is created, and to do so the requirements are depicted in Table 1. Parrot OS is very user friendly, the user interface is very simple, it is easy to access, it requires lower specification hardware as compared to Kali OS and it comes with a bunch of system tools.

Malware is created in Parrot OS, keeping security and privacy in mind this OS is built for security purpose. Using the command as shown in Fig 1., the malware is created.

978-1-7281-9744-9/20/$31.00 ©2020 IEEE



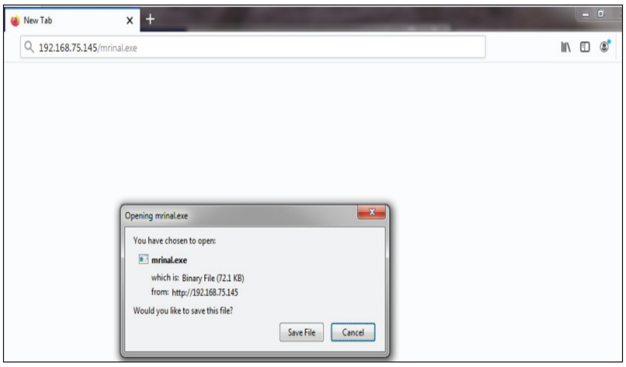
```
┌─[lambda@parrot]─[~]
└──╼ $sudo su
[sudo] password for lambda:
┌─[root@parrot]─[/home/lambda]
└──╼ #msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.75.145 lport=9
999 -f exe -o /var/www/html/mrinal.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the
 payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 341 bytes
Final size of exe file: 73802 bytes
Saved as: /var/www/html/mrinal.exe
┌─[root@parrot]─[/home/lambda]
└──╼ #service apache2 restart
┌─[root@parrot]─[/home/lambda]
└──╼ #
```

Fig 1. Creating Malware.

TABLE I. SYSTEM CONFIGURATION.

Host Machine	
Model	Assembled
Processor	AMD Ryzen 7 3700X, 8 Core @ 4.4 GHZ 36MB Cache
RAM	16.0 GB DDR4 3000MHz
System Type	64-bit Operating System
Operating System	Windows 10 Pro
Tools/Packages	OSINTs Metasploit
Hypervisor	
Version	VMware® Workstation Pro15.5.0 build14665864
Processor	64-bit Operating System
Virtual Machine	
Parrot Security KDE	v64-bit,2GB RAM, Hard Disk:20 GB
Windows 7	v64-bit,1GB RAM, Hard Disk:40 GB

Then that malware is downloaded on the Windows machine for steganography and to show how easy it is to create and send the file for downloading as shown in Fig 2.



Opening mrinal.exe

You have chosen to open:

mrinal.exe

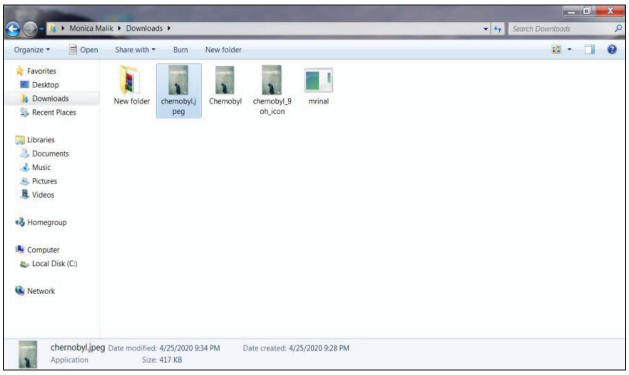
which is: Binary File (72.1 KB)

from: http://192.168.75.145

Would you like to save this file?

Fig 2. Victim downloading the Malware.

For Steganography here the malware mrinal.exe is embedded with the image of any known TV Series Chernobyl (or any other picture can also be taken of the victim's interest to attract him/her) as demonstrated in Fig 3.



chernobyl.jpeg Date modified: 4/25/2020 9:34 PM Date created: 4/25/2020 9:28 PM

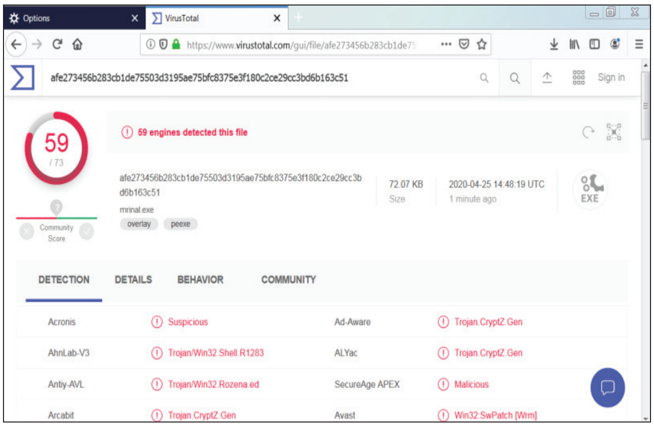
Application Size: 417 KB

Fig 3. Malware afore binding and after binding with and image.

As it can be seen the name of the selected image is Chernobyl.jpeg but if its properties are seen it is showing as the application which means it is an executable file.

B. *Analysis using OSINT: VirusTotal*

As demonstrated in Fig 4 and Fig 5 the results of the mrinal.exe and Chernobyl.jpeg are shown. And it can be seen that there is a huge difference between the results afore & after binding malware with an image. As afore embedding malware with an image 59 out of 73 engines were able to scan the file using VirusTotal as demonstrated in Fig .4, whereas after embedding malware with an image only 36 out of 71 engines were able to scan the file as shown in Fig 5.



afe273456b283cb1de75503d3195ae75bfc8375e3f180c2ce29cc3bd6b163c51

59 / 73

59 engines detected this file

afe273456b283cb1de75503d3195ae75bfc8375e3f180c2ce29cc3bd6b163c51

mrinal.exe

overlay peexe

72.07 KB Size

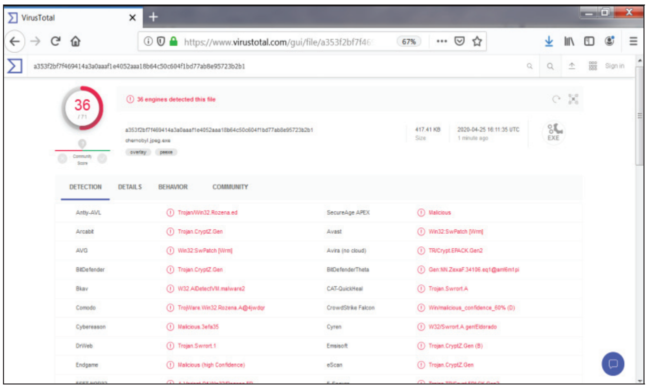
2020-04-25 14:48:19 UTC 1 minute ago

EXE

Community Score

Acronis	Suspicious	Ad-Aware	Trojan.CryptZ.Gen
AhnLab-V3	Trojan/Win32.Shell.R1283	ALYac	Trojan.CryptZ.Gen
Antiy-AVL	Trojan/Win32.Rozena.ed	SecureAge APEX	Malicious
Arcabit	Trojan.CryptZ.Gen	Avast	Win32:SwPatch [Wrm]

Fig 4. VirusTotal results afore binding the malware with an image.



a353f2bf7f469414a3a0aaf1e4052aaa18b64c50c604f1bd77ab8e95723b2b1

36 / 71

36 engines detected this file

a353f2bf7f469414a3a0aaf1e4052aaa18b64c50c604f1bd77ab8e95723b2b1

chernobyl.jpeg.exe

overlay peexe

417.41 KB Size

2020-04-25 16:11:35 UTC 1 minute ago

EXE

Community Score

Antiy-AVL	Trojan/Win32.Rozena.ed	SecureAge APEX	Malicious
Arcabit	Trojan.CryptZ.Gen	Avast	Win32:SwPatch [Wrm]
AVG	Win32:SwPatch [Wrm]	Avira (no cloud)	TR/Crypt.EPACK.Gen2
BitDefender	Trojan.CryptZ.Gen	BitDefenderTheta	Gen:NN.ZexaF.34106.eq1@am6m1pi
Bkav	W32.AIDetectVM.malware2	CAT-QuickHeal	Trojan.Swrort.A
Comodo	TrojWare.Win32.Rozena.A@4jwdqr	CrowdStrike Falcon	Win/malicious_confidence_60% (D)
Cybereason	Malicious.3efa35	Cyren	W32/Swrort.A.gen!Eldorado
DrWeb	Trojan.Swrort.1	Emsisoft	Trojan.CryptZ.Gen (B)
Endgame	Malicious (high Confidence)	eScan	Trojan.CryptZ.Gen

Fig 5. VirusTotal results after binding the malware with an image.

The results show how the engines were able to scan afore & after steganography. Similarly, the anti-malware works. They go into the file and check whether the file is malicious or not but if the anti-malware doesn't have any information about the malware's signature then the anti-malware won't be able to detect the malicious content and will tell the user that the file is non-malicious and safe to use and after that the machine gets infected. So, it's very important to have good defenders or anti-malware.

C. Analysis Table

In this section, similar to VirusTotal other OSINTs are taken to identify how many engines can detect the malware as one should never rely on one source's results. As the analysis shown in Table 2, after steganography, the malware is not detectable as afore steganography it is. This means steganography is very effective and strong as there is a huge difference in the results.

TABLE II. OSINTS ANALYSIS.

S. No.	OSINTs	Afore binding the Malware	After binding the Malware
1.	VirusTotal	59/73	36/71
2.	OPSWAT (Meta Defender)	29/40	6/40
3.	VirSCAN	27/49	9/49
4.	Jotti	13/15	11/15
5.	Bitbaan MaLab	11/21	4/20
6.	PolySwarm	11/15	6/11

Along with the scanning results, the VirusTotal gives a lot of other information also asMD5, SHA-1, SHA-256, Vhash values, etc. Similarly, other OSINTs also provide the same or more information about the files.

III. LIMITATIONS AND FUTURE WORK

Only freely available OSINTs are considered which doesn't have many known engines to detect the malware. In the future, paid tools also can be used for enhancing the results and to provide more information on the malware.

IV. CONCLUSION

Malware is a rapidly growing and never-ending concern, but effective and defensive measures can be taken to overcome the damage that they can cause if properly not taken care of. This paper gives a thorough overview of the phishing technique, which is widely used to manipulate victims, and how the malware can exploit the system to gain access and information. In this work, malware is created, and then the same malware is being analyzed on different freely available OSINTs. A light has been thrown on how binding a malware makes it difficult for OSINTs to identify and detect them. The analysis shows that the best OSINT is VirusTotal which has a greater number of engines that could detect the malware whereas others don't have a variety of engines to detect the malware. Also, when it comes to malware afore binding it with an image is easier to detect whereas for an OSINT it was difficult to identify and detect the malware after binding with an image.

REFERENCES

[1] Talukder, S., & Talukder, Z. A SURVEY ON MALWARE DETECTION AND ANALYSIS TOOLS.

[2] Mansour, R. F. (2016). Understanding how big data leads to social networking vulnerability. Computers in Human Behavior, 57, 348-351.

[3] Stevens, C. (2020). Assembling cybersecurity: The politics and materiality of technical malware reports and the case of Stuxnet. Contemporary Security Policy, 41(1), 129-152.

[4] Patel, N. (2020). SOCIAL ENGINEERING AS AN EVOLUTIONARY THREAT TO INFORMATION SECURITY IN HEALTHCARE ORGANIZATIONS. Jurnal Administrasi Kesehatan Indonesia, 8(1).

[5] Kumar, S. (2020). An emerging threat Fileless malware: a survey and research challenges. Cybersecurity, 3(1), 1-12.

[6] Tariq, N. (2018). IMPACT OF CYBERATTACKS ON FINANCIAL INSTITUTIONS. Journal of Internet Banking and Commerce, 23(2), 1-11.

[7] Alzaylaee, M. K., Yerima, S. Y., & Sezer, S. (2020). DL-Droid: Deep learning based android malware detection using real devices. Computers & Security, 89, 101663.

[8] Wang, W., Zhao, M., & Wang, J. (2019). Effective android malware detection with a hybrid model based on deep autoencoder and convolutional neural network. Journal of Ambient Intelligence and Humanized Computing, 10(8), 3035-3043.

[9] Asquith, M. (2016). Extremely scalable storage and clustering of malware metadata. Journal of Computer Virology and Hacking Techniques, 12(2), 49-58.

[10] Jiang, X., Mao, B., Guan, J., & Huang, X. (2020). Android malware detection using fine-grained features. Scientific Programming, 2020.

[11] Polino, M., Scorti, A., Maggi, F., & Zanero, S. (2015, July). Jackdaw: Towards automatic reverse engineering of large datasets of binaries. In International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (pp. 121-143). Springer, Cham.

[12] Howard, M., Pfeffer, A., Dalai, M., & Reposa, M. (2017, October). Predicting signatures of future malware variants. In 2017 12th International Conference on Malicious and Unwanted Software (MALWARE) (pp. 126-132). IEEE.

[13] Saha, S., Das, A., Kumar, A., Biswas, D., & Saha, S. (2019, August). Ethical Hacking: Redefining Security in Information System. In International Ethical Hacking Conference (pp. 203-218). Springer, Singapore.

[14] Denemark, T. D., Boroumand, M., & Fridrich, J. (2016). Steganalysis features for content-adaptive JPEG steganography. IEEE Transactions on Information Forensics and Security, 11(8), 1736-1746.

[15] Prem, S. P., & Reddy, B. I. (2019). PHISHING AND ANTI-PHISHING TECHNIQUES.

[16] Mesdaq, A., Singh, A., & Jain, V. (2020). U.S. Patent No. 10,601,865. Washington, DC: U.S. Patent and Trademark Office.

[17] Ramirez, J. B. Q. (2019). U.S. Patent Application No. 16/175,648.

[18] Narwal, B., Mohapatra, A. K., & Usmani, K. A. (2019). Towards a taxonomy of cyber threats against target applications. Journal of Statistics and Management Systems, 22(2), 301-325.

[19] Dhawan, S., & Narwal, B. (2019). Unfolding the Mystery of Ransomware. In International Conference on Innovative Computing and Communications (pp. 25-32). Springer, Singapore.

[20] Narwal, B. (2019, November). Security analysis and verification of authenticated mobile payment protocols. In 2019 4th International Conference on Information Systems and Computer Networks (ISCON) (pp. 202-207). IEEE.

[21] Bhawna Narwal* and Amar Kumar Mohapatra, "SALMAKA: Secured, Anonymity Preserving and Lightweight Mutual Authentication and Key Agreement Scheme for WBAN", International Journal of Sensors, Wireless Communications and Control (2020) 10: 1. https://doi.org/10.2174/2210327910999200507124851

[22] Dhawan S., Shah S., Narwal B. (2020) A Walkthrough of Blockchain Technology and Its Potential Applications. In: Batra U., Roy N., Panda B. (eds) Data Science and Analytics. REDSET 2019. Communications in Computer and Information Science, vol 1230. Springer, Singapore.

[23] Narwal, B., & Mohapatra, A. K. (2020). SEEMAKA: Secured Energy-Efficient Mutual Authentication and Key Agreement Scheme for Wireless Body Area Networks. Wireless Personal Communications, 1-24.

[24] Malik, M., & Narwal, B. Automated Malware Identifier and Analyzer. In Next Generation Information Processing System (pp. 82-90). Springer, Singapore.