Sule Lamido University Journal of Science and Technology (SLUJST) Vol. 2 No. 1 [January, 2021], pp. 65-72

# Performance Comparison of PyRAT and Phantom Antivirus Software Evasion Tools

Aminu Shawwal Adam[1], Zahraddeen Sufyanu[2]
[1,2]Department of computer sciences, Faculty of Computing
Federal University Dutse (F.U.D) Jigawa State
[1,2] [aminushawwal, sufyanzzzzzi] @gmail.com, [1][Amshal2005]@yahoo.com

**Abstract**

*Nowadays, Malware becomes a new way of cybercrime, and hackers are finding various ways to generate it in all available platforms. Information security breach is one of the challenging issues affecting most of the organizations. This paper employed window platform to implement tools (pyRAT and Phantom) that automate the generation of Metasploit payload executable. The exploitation process generates a meterpreter session between malicious user and the target system. An experimental research design was adopted, using Virtual lab setup with VM Oracle virtualbox, which consisted of two machines (attacking and target machine), so as to test the Evasion tools against AV software products. The development of pyRAT and Phantom is strictly for educational purposes; therefore, any other malicious action using these tools is not recommendable. The study proved pyRAT with the best evading capability having bypassed most of the chosen antivirus by 67% while Phantom evasion tool acquired 50% evading stand.*

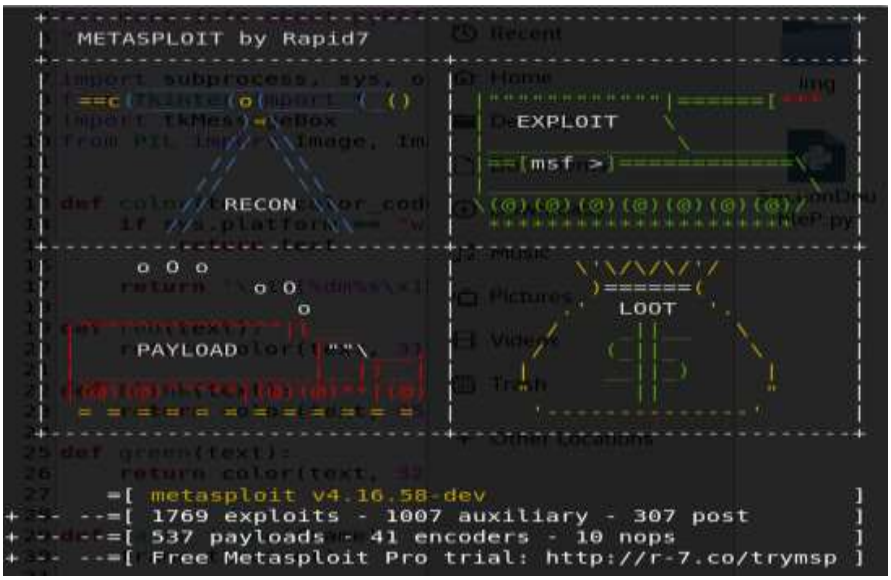
Keywords: ***Antivirus, Evasion tool, Malware, Payload, Hacker.***

## 1. Introduction

Use of Malware recently, has become a method used to have access and compromise an important data or information stored in organizations' or individual's computers. The malware incorporates a payload that enables backdoors access to computers and stealing of sensitive information. Exploits application such as Core Impact, Canvas and Metasploit Framework are used to generate such malicious payload (Joel, 2014). Henceforth, the malware is simple to be generated given the number of freely available tools on the Internet. According to Barriga and Yoo (2017), organizations need to protect their information resources, since network security breaches might lead to loss of business reliability and efficiency. Meanwhile, time and labour involved in reorganizing infected systems pose a big expense (Nishant, 2012).

Python programming language is used to develop an interactive Phantom antivirus evasion tool. The tool is aimed to make antivirus evasion software easy for penetration testers, through the use of modules that focuses on antivirus sandbox detection techniques and polymorphic code (Cornacchini, 2018). Phantom-Evasion Tool has the ability to generate fully undetectable (FUD) payload, even with the foremost common thirty-two (32) bit and lower detection quantitative relation with sixty-four (64) bit payloads. However, The Phantom also incorporated with a section performing post exploitation, which is dedicated to auxiliary modules and endurance. According to the github.com website, Phantom evasion tool can run through manual setup, on the following Operating Systems (OS): BlackArch Linux (64 bit), Elementary (64 bit), Arch Linux (64 bit), Ubuntu 15.10+ (64 bit), Windows 7/8/10 (64 bit) and Linux Mint (64 bit). Consequently, Parrot Security (64 bit) and Kali Linux Rolling 2018.1 plus (64 bit) are the Operating System supporting automatic setup.
On the other hand, pyRAT Antivirus Evasion Tool also called Python Rat which is written in Python programming language. Python is developed by Guido van Rossum and originally released in 1991, and is among the Object Oriented Programming language that contains a dynamic linguistics for general purpose programming (Dave, 2012). pyRAT have become a very attractive, and easy to learn and use, as it comes with built-in high level data structure and automated memory management. Besides, it consists dynamic binding, typing system, and supports a large amount of packages and modules.

Metasploit project provides detail information on security weakness and helps in penetration testing activities for Intrusion Detection System (IDS) signature development and exploitation. It is an open source Metasploit framework, a tool for developing and executing exploit code against a remote target machine (Nikolaos, 2018). Metasploit framework can run on Linux, Mac OS X and Windows operating system platforms. It can also be extended to use add-ons in multiple languages. The Metasploit project is very popular in anti-forensic and evasion tools, which constituted in Metasploit framework. Figure 1 shows the Metasploit framework start-up interface.




(Source: Nikolaos, 2018)
Figure 1: Metasploit framework interface

As the information security breach became one of the complex challenging issues. In this paper, open source security software Metasploit framework is used, as it offers a complete development environment to come-up with a new software evasion tools. This automates many aspects of Pentest with reliable library of constantly updated exploits. Thus, the AV Evasion tools efficiency of pyRAT and Phantom are compared on window platform to determined their evading capabilities.

## 2. Related Works

Themelis (2018) explained that pyRAT can generate payload from Metasploit for a variety of Window machines, which can bypass AV Products. According to Barriga *et al.* (2017), among the AV software protections, there are many techniques that can be used to detect malicious activities, but the main techniques include: Behaviour-Based, Heuristic Based, and Signature-Based techniques. In cyber-security, the AV software products are considered as one of the first line of defence malicious users face when trying to make an attack. On the other hand, computer hackers find a way to avoid detection by these AV products, using AV Evasion techniques such as code reuse attacks encryption, oligomorphism, obfuscation, metamorphism, and polymorphism.

According to Bei-Tseng *et al*. (2011), penetration testing (also called Pentest) is a practice conducted by professional in cyber-security to assess the security strength of a system or networks. The purpose is to discover vulnerabilities that can be exploited by hackers. Also, the Pentest assessment results are important bases to build up a network and security solution (Garcia & Johnston, 2002).

According to Christos (2018), metasploit framework is built-up with a lot of potential, this includes libraries, interfaces, mixings, plug-ins and modules that are suitable for penetration testing activities. Meterpreter is a payload that can be generated from Metasploit framework. The Metasploit Project was created by H. D. Moore in 2003, as a computer security project. It is originally written in Perl programming language, but completely modified to Ruby language in 2007.

This framework dynamically works in memory DLL injection stager, and is extensible over the network at run time. The stager socket is used to channel the meterpreter and provide a comprehensive Ruby API at client-side computers. Meterpreter conceals completely in memory which writes nothing to disk, but uses encrypted communication on the victim machine, leaving limited forensic evidence and impact (Nikolaos, et al., 2018). The window reverse tcp meterpreter is the commonly used meterpreter payload. There exist many studies in the literature that evaluate the Antivirus Evasion tools with various strengths and weaknesses. Summary of related studies is presented in Table 1.

Table 1: Summary of related studies

| S/N | Author(s) | Research | Strength | Weakness |
|---|---|---|---|---|
| 1. | Faisal A. G., et al. (2019) | Evaluates The State of the Art Antivirus Evasion Tools On Windows And Android Platform. | Generated the meterpreter payload using popular Metasploit framework. | The study need to be extended to other platform s such as Linux and Mac. |
| 2. | Christos, K. (2018) | AV Software Evasion: Evaluation of the Antivirus Evasion Tools. | Used popular Metasploit reverse tcp meterpreter payload, and some sample files custom payload. | Small number of evasion tools, and no encoding |
| 3. | Themelis (2018) | Tool for AV Evasion: pyRAT . | The study uses Metasploit Framework along with its features to automate the payload. | Employed predefined payload |
| 4. | Jameel Haffejee (2015) | An Analysis of Malware Evasion Techniques Against Modern AV Engines. | The study employed a number of binaries that uses virus like techniques to exploit the area of testing that could evade an AV engine online. | Delay in time testing: costing in terms of time and resources when using an offline laboratory |
| 5. | Kim, H. S. and Sukwong, O. (2011) | Evaluate the Effectiveness of Commercial AV Software. | Used variety of detection techniques. | The test conducted on random collection of malware sample that are not necessary obfuscated. |
| 6. | Balanchandra and chua (2018) | Evaluate the Effective ness of Android Obfuscation on Evading Malwares | Used large number of malware sample and 57 Anti-malware Tools (AMTs) on VirusTotal | Command- Line version was used that might be performing static analysis with certain degree of the signature database. |
| 7. | Present Study | Performance Comparison of pyRAT and Phantom AV Software Evasion tools | Used popular Metasploit Framework. | Limited Numbers of AV Evasion Tools are Used in the Study. |

This section concludes with review on how to select the AV software products in the market. Different sources were reviewed and points are assigned to each AV product that made an appearance in the review. AV software products with highest points of appearance are chosen for this study, as depicted in Table 2.

Table 2: Selection of AV software products

| **Antivirus Score** | J. N. Rubenking (2019) | J. N. Rubenking (2020) | T. Fisher (2019) | Mike Williams (2020) | Desire Athow (2020) | R. Batema (2020) | Paul W. (2020 | Paul W. (2019 | J. Allen (2019 | K. G. Orphan (2019) | **rating Scores** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Avast AV** | **1** | **0** | **1** | **0** | **1** | **0** | **0** | **1** | **1** | **1** | **6** |
| **Kaspersky** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **0** | **1** | **9** |
| **AVG AV** | **1** | **0** | **1** | **0** | **0** | **0** | **0** | **1** | **1** | **1** | **5** |
| **Bitdefender** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **1** | **10** |
| CheckPoint | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| ZoneAlarm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| G-Data AV | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2 |
| F-Secure | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2 |
| Sophos AV | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| Intego AV | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| **Avira AV** | **1** | **0** | **1** | **0** | **0** | **1** | **0** | **1** | **1** | **1** | **6** |
| Adaware | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| Comodo | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 4 |
| ESET NOD32 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 3 |
| **Panda AV** | **1** | **0** | **1** | **0** | **0** | **1** | **0** | **1** | **0** | **1** | **5** |
| Total AV | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| Norton AV | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 4 |
| BullGuard | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| McAfee | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 3 |
| Intrusta | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CYLANCE | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| SmartAV | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| Heimdal | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Webroot | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 3 |
| FortiClient | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Immunet | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Malware bytes | | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Windows Defender | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |

## 3. Methodology

An experiment was conducted in Network Address Translation (NAT) Virtualbox lab setup, which constituted 2 virtual machines: Kali Linux (malware generation machine) and Window 10 (target or victim machine). This is to test the generated evasion tools from Metasploit framework against the selected AV software products. The host computer in the experiment contained Windows 8 OS, 8 gigabytes memory, and 500 gigabyte disk space HDD. The isolated laboratory environment is illustrated in Figure 2. Note that, throughout this paper the two virtual machines are referred to as Attacking machine and Target machine, respectively.

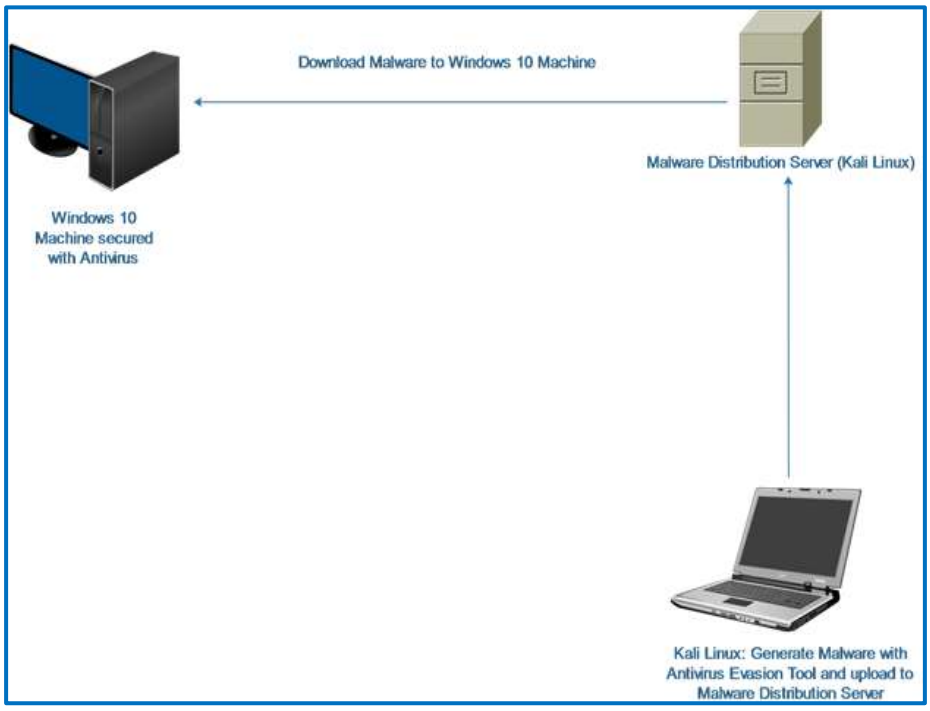



Figure 2: Laboratory setup system architecture (Source: Aminu *et al.*, 2020)

Figure 2 shows how the target machine downloaded the malware generated from the attacking machine and run for executions. The selected AV software products used in this study were consecutively installed on target machine, and tested by the malware generated from the evasion tools (pyRAT and Phantom). During the evaluation, a score of 1 point was awarded to the AV software product that detected and stop the malware, otherwise 0 point was assigned to it.

Furthermore, when the generated AV Evasion tool bypassed the AV software product, a score of 1 point also assigned to it, otherwise 0 point. This process continued until all the selected AV software products were tested over the evasion malware. The results were organised to determine the strengths and weaknesses of the evaluations. Subsequently, the AV Evasion tools chosen for this study follow different series of phases for their generation in the Metasploit framework:

### 3.1 Phantom Av Evasion Tools Workflow

When the Phantom evasion tool was fully installed on Kali Linux Attacking Machine, it indicates how Windows Module was selected in the home page, and then the following options and choices were made: Index Stager Modules, Window Stager Module x64, pop up of Module Description, Entering of Parameter. Finally, Phantom portable executable file generated.

The general procedure is listed here:

- ✓ Select option [1] for Windows modules : in the main menu of Phantom Evasion Home page.
- ✓ Select option [2] for stager: in the Windows module indexs
- ✓ Insert option: 2 in the msfconsole CLI, for x64 bit system window stager modules
- ✓ Select the kind off reverse meterpreter payload: in the Windows x64 stager modules
  The meterpreter payload setup descriptions appear in detail, such as (window/x64/meterpreter/reverse_tcp)
- ✓ Press enter to continue
- ✓ Insert the following parameters to generate the payload:
  LHOST: 10.0.2.15
  LPORT: 440

OUTPUT FILENAME: aminu
- ✓ Compiling to generate C meterpreter stager
- ✓ Select y{yes} to strip executable
- ✓ Select y{yes} to sign executable
  If succeeded the payload executable file aminu.exe should be saved in the Phantom – Evasion folder
- ✓ Finally, press enter to continue

### 3.2 pyRAT Evasion Tools Workflow

When the pyRAT AV evader start-up, it indicate "Show exploits button" which allows pyRAT evasion software to begin the first stage. The first Phase displays various Metasploit exploits for window OS. By pressing "show compatible payload button" it moved to the next stage. And in order to achieve a successful exploitation, before running the application the user has to make reconnaissance or intelligent findings so as to gain information on the target machine for vulnerabilities. The next stage displayed different payloads (meterpreter types) depends on the system compatible. By chosen the desired meterpreter payload next is to click on the button called 'Choose Payload button', next phase a form contained four different blank fields appeared, therein the user filled all the information required to create the payload such as: Local host, Local port, Remote host (optional) and Payload name. Afterward the user clicks on "Generate Payload" button, to create the payload.
Note; for the user to check if the generated payload in the previous step can be caught by AVs then, ClamAV can be used for the verifications. In addition, the user can manually scan the malware online, for instance using VirusTotal online scanner to get the level at which the AVs can detect the malware scanned. In the final stage, once the payload portable executable file opened successfully in the target system and the meterpreter session become active. Then, the attacker can take full control of the victim's machine.

The general procedure is listed here:

- ✓ Press 'show exploits' button in the pyRAT AV Evader Home page
- ✓ Select the desired Windows portable executable PE file
- ✓ Press 'show compatible payload' button
- ✓ Select payload type
- ✓ Press 'choose payload' button
- ✓ Enter the payload details, to generate the payload such as
  Local IP: 192.168.2.87
  Local Port: 4444
  Payload's name: aminu
- ✓ Scanning file for virus
- ✓ Press OK to continue
- ✓ Press 'hide payload' button for the next step
- ✓ The payload is now ready for the attack
- ✓ Finally, press 'Quit' button to close

Finally, as it has been depicted in Table 2 of literature reviewed that six Antiviruses were selected as candidates for this study; these include Avira Prime AV, Bitdefender AV Plus, Avast Pro, Kaspersky internet Security, and Panda Antivirus.

## 4. Results and Discussion

The results generated during the experimentation and the points scored, are displayed in Table 3 through Table 5.The findings in Table 5 from the previous sections of results reveal that, pyRAT evasion tool reported the best performance in terms of evasion capability which evaded 4 out of 6 AVs, while Phantom evasion tool bypassed three AV products from the six chosen AV products in this study. Therefore, the evasion ratios between pyRAT and Phantom evasion tools are compared and displayed in Figure 3. Thus, the ratios indicated that, the AV evasion tools bypassed the selected Antivirus software products counts from 50% -

67% respectively. In addition, pyRAT evasion was observed with 67% evasion ratio and Phantom acquired 50%.

Table 3: pyRat AV Evasion Tool Test Results

| Antivirus Solution | pyRat Score | Antivirus Score |
|---|---|---|
| Avira | 1 | 0 |
| Bitdefender | 1 | 0 |
| Avast | 1 | 0 |
| Kaspersky | 0 | 1 |
| AVG | 1 | 0 |
| Panda | 0 | 1 |

Table 4: Phantom AV Evasion Tool Test Results

| Antivirus Solution | Phantom Evasion Score | Antivirus Score |
|---|---|---|
| Avira | 0 | 1 |
| Bitdefender | 1 | 0 |
| Avast | 0 | 1 |
| Kaspersky | 0 | 1 |
| AVG | 1 | 0 |
| Panda | 1 | 0 |

Table 5: pyRAT and Phantom AV Evasion Tools result summary

| S/N | AV Evasion Tools | Antivirus Free | | | | | | AV Evasion Tools Total Scores |
|---|---|---|---|---|---|---|---|---|
| | | Avira | Bitdefender | Avast | Kaspersky | AVG | Panda | |
| 1. | Phantom | 0 | 1 | 0 | 0 | 1 | 1 | **3** |
| 2. | pyRAT | 1 | 1 | 1 | 0 | 1 | 0 | **4** |

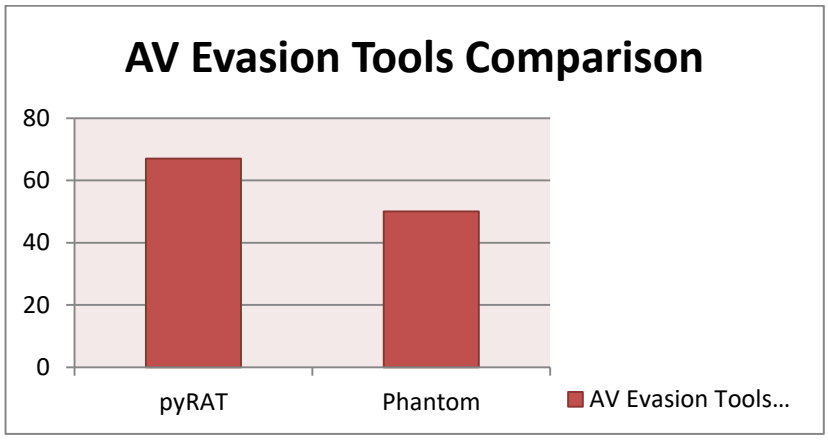



Figure 3: Comparison of antivirus evasion tools

## 5. Conclusions

pyRAT and Phantom Evasion tools were investigated and compared to find the better effective evasion capability. The work has presented a concise and well-defined way on how to attack systems or networks

effectively. There are many other AV evasion tools available in public circulations, which can be tested. The results of the study proved that pyRAT evasion tool outperformed Phantom in terms of evading capability by 17%. The study can serve as a guide to those that want to practice and learn penetration testing and Metsploit exploit.

## Reference


Aminu, S. A., Zahraddeen S. and Tajudden S. (2020) Evaluating the Effectiveness of AV Software Evasion Tools against a Window Platform. *Fudma Journal of Sciences (FJS), Vol. 4 No. 1, March, 2020,* pp. 89 – 92

Bei-Tseng, B., Chu, Aileen, G. B., Xiaohong, Y. and Monique, J. (2011). Overview of Penetration Testing. *International Journal of Network Security & Its Applications (IJNSA), 3(6),*

Balachandran and Chua (2018). Evaluation of the Effectiveness of Android Obfuscation on Evading Antimalware.Athttps://www.researchgate.net/publication/323786257_Effectiveness_of_Android_Obfuscation_on_Evading_Anti-malware (Accessed 28 December, 2019).

Barriga, J. J. and Yoo, S. G. (2017) Malware Detection and Evasion with Machine Learning Techniques: *International Journal of Applied Engineering Research (IJAER),* Vol 12, pp. 7207-7214.

Christos, K. (2018). Antivirus Software Evasion: An Evaluation of the AV Evasion Tools. Published by Piraeus University, 2018.

Desire Athow (2020) The best Antivirus Software Products of 2020. Retrieved from: https://www.ITproportal.com/us/best-freeantivirus, accessed on 10th August, 2020

Dave, K. (2012). A Python Book: Beginning Python, Advanced Python, and Python Exercises. https://www.archived/learnpython.com (Accessed on 20 January, 2020).

Faisal A. Garba, Kabiru I. Kunya, Shazali A. Ibrahim, Abubakar B. Isah, Khadija M. Muhammad, Nasir N. Wali (2019). Evaluation the State of the Art Antivirus Evasion Tools on Window and Android Platform. The 2019 2nd International Conference of IEEE (NigeriaComputConf)

Fisher, T. (2019). The 10 Best Free Antivirus Software of 2019. Retrieved from: https://www.lifewire.com/best-free-antivirus-software- 4151895. Accessed on 02, Feb., 2020.

Garcia, G., A. R., Johnston, R., (2002). Vulnerability Assessment of Security Seals. A Technical report LA- UR-96-3672. Published by Alamos National Lab.

J., Allen (2019). Eight (8) of the Best Free Antivirus Software. Retrieved from: https://mashable.com/roundup/best-free-antivirus/. Accessed: 5th Jan., 2020.

Joel, K. A. (2014). Network and Systems Security Assessment using penetration testing. University Of Science and Technology Kwame Nkrumah.

Kim, H. S. and Sukwong, O. (2011). Commercial Antivirus Software Effectiveness: An Empirical Study, *IEEE Computer Society*, pp. 63-70.

Moore, H. D., (2003) Metasploits Project. Retrieved from: https://www.metasploitproect.com. Accessed on 2nd March, 2020.

Mike Williams (2020) Top Five Antiviruses for 2020. Retrieved from: https://www.safetyDetection.com/us/best-freeantivirus, accessed on 10th August, 2020.

Nikolaos, Themelis (2018). A Tool for Antivirus Evasion: pyRAT. The University of Piraeus, Available: https://github.com/govolution/avet.net. Accessed on 10 Feb., 2019.

Nishant, S. (2012). Security Assessment via Penetration Testing: A Network and System Administrator's Approach. University Of Oslo, accessed on Feb. 14, 2020.

Neil, J., Rubenking (2019). The 2019-2020 Best Free Antivirus Products. Retrieved from: https://www.pcmag.com/roundup/267984/the-bestfree- antivirus-protection. Accessed on 9th Feb., 2020.

Orphanides, K. G. (2019). Best Free Antivirus 2019: 6 tried and tested ways to stay safe. Retrieved from: https://www.trustedreviews.com/best/best-freeantivirus-net. Accessed on 28, Jan., 2020.

Paul W. (2020). Best Free Antivirus Software 2019-2020. Retrieved from: https://www.tomsguide.com/us/best-freeantivirus, review-6003.html Accessed: Feb., 2020.

Robert Beteman (2020). The 10 Best Antiviruses in 2020| Windows, Android iOS and Mac. Retrieved from: https://www.safetyDetection.com/us/best-freeantivirus, accessed on 10th August, 2020

Techopedia website (2019). Malware – Payload Behavior. Available at www.technopedia.com Accessed: on January 2020.