Detection of attack on Windows password: PupyRAT

[1]Pratheesh Francis, [2]Reshma Sarah Rony, [3]Sandra Sebastian, [4]Dr.Juby Mathew

[1, 2, 3] P G Scholar, Amal Jyothi College of Engineering, Kanjirappally, 686518

[4]Associate Professor, Amal Jyothi College of Engineering, Kanjirappally, 686518

[1]pradeeshkurianfrancis@mca.ajce.in, [2]reshmasarahrony@mca.ajce.in, [3]sandrasebastian@mca.ajce.in, [4]jubymathew@amaljyothi.ac.in

***Abstract*—Password cracking has become one of the popular means of breaking into one's privacy. Various methods have been adopted to store passwords securely from intruders. One such method is to keep the passwords hashed. The user passwords in Windows are stored as hashed codes in a registry hive. To obtain these codes secretly and remotely from the target system, the Remote Administration Tool (RAT) like PupyRAT is used. It can connect to a target system through a backdoor attack. Once a session is created, the attacker can bypass the User Account Control (UAC) to gain the admin privilege. Only then can the attacker download the registry hive. To be able to bypass the UAC, PupyRAT injects multiple PowerShell commands into the target system. A system that detects these PowerShell windows is developed to alert and terminate possible RAT attacks.**

Key Words: Password cracking, Remote Administration Tool, Hashing, Backdoor attack

I. INTRODUCTION

Backdoors are a type of malware used by both authorized and unauthorized users to access system data. They function by appending the payload with the actual transmitted data. This payload performs intended actions on the target once it is run in the target machine.

REMOTE ADMINISTRATION TOOL

A remote administration tool is a program used by hackers or other people to connect to a computer via the Internet or across a local network remotely. It is based on the server and client technology such that the server part runs on a controlled computer and receives commands from the client, which is installed on the remote host. PupyRAT is a multi-function RAT (Remote Administration Tool) and exploitation tool mainly written in python which can exploit a targeted system and retrieve confidential data stored in the system.

SECURITY ACCOUNT MANAGER

The Security Account Manager (SAM) is a database file in Windows to store user's passwords. It can be used to authenticate users using cryptographic measures to prevent unauthenticated users from accessing the system. This file is found in %SystemRoot%/system32/config/SAM. The "System" account is the only account that can access the SAM file during operation.

LAN MANAGER (LM) HASH

The LAN Manager hash was one of the first password hashing algorithms used by Windows operating systems. The LM hash of a password is computed through six-steps:

1. The whole user's password is converted into uppercase letters
2. The null characters are added to the password until it equals to 14 characters
3. The new password is split into two halves of 7 character each
4. 64 bit DES encryption keys, one from each half are created with a parity bit added to each.
5. Each of these DES keys is used to encrypt a pre-set ASCII string (KGS!@#$%) which thereafter produces two 8-byte ciphertext values.
6. These two 8-byte ciphertext values are then combined to form a 16-byte value, which is the completed LM hash

NEW TECHNOLOGY LAN MANAGER (NTLM) HASH

NT LAN Manager (NTLM) is the Microsoft authentication protocol created as the successor of LAN Manager (LM). NTLM hash creation is a simpler process and relies on the MD4 hashing algorithm to create the hash-based upon a series of mathematical calculations. After the password is converted to Unicode, the MD4 algorithm is used to produce the NT hash A user can generate what are called rainbow tables containing every single hash value for every possible password possibility up to a certain number of characters. Using a rainbow table, one can simply take

the hash value extracted from the target computer and search for it. Once it is found from the table, you have the password.

II. LITERATURE REVIEW

Anis Ismail[1], Mohammad Hajjar[1], Haissam Hajjar[1] had done some comparative study on different Remote Administration Tools (RAT). RAT provides fast secure access to remote PC's on Windows platforms so, that we see the remote computer's screen on our monitor and all the mouse movements and keystrokes are transferred directly to the remote machine. Remote administration of computers is increasingly common because it is low cost, easy-to-use and reliable and all tasks are automated. There exist many remote administrator tools in the market and it is difficult to choose what we need. Some of the Tools are GoToMyPC, PCAnywhere, and RemotelyAnywhere. Remote-control solutions such as GoToMyPC and PCAnywhere are one way to provide cost-effective network access to their remote and mobile employees. It provides a highly secure and cost-effective method for employees to access their network resources remotely and work on their computers using any Web browsers. GoToMyPC is a cost-effective, easy-to-implement, fast and secure way to enable employees to remotely access corporate network resources. It is convenient because it works from almost anywhere and requires no configuration So, GoToMyPC has significant advantages over PCAnywhere. RemotelyAnywhere is a more secure, cost-effective, and powerful remote administration solution that provides tools for complete administration of workstations and servers on and off the LAN. It provides fast and secure remote access to our corporate network. RemotelyAnywhere does not require special client software to be installed on the local machine. A Remote administrator tool is an affordable tool that has no special hardware requirements. This evaluation lets customers choose their need of remote administrator tools carefully.

Sam Martin[2] and Mark Tokutomi[2] describe how passwords can be cracked and the techniques for password cracking. A password is a string of characters designed by the system to provide authentication. There are many different ways to authenticate users in a system it includes physical objects like a key card, fingerprint, or something that only the user knows. When the user inputs the password, the system can simply check the hash of the input with the stored hash value. Security Accounts Manager (SAM) file is a Password file for Windows, is located in C:\windows\system32\config\sam. Storing the hashed values for passwords is certainly much more secure than storing the plain text. Some additional measures that can be adopted to enhance the security for a hashed password by a process called password salting. It is a technique that adds some random data to the input before hashing. Even though there exist several attacks against hashed passwords. A brute force attack is a cryptographic hack that guesses possible combinations of a targeted password until the correct password is discovered. It doesn't work for sufficiently long passwords so, creating a dictionary that contains commonly occurring passwords helps an attacker to guess correct passwords more often than brute-forcing. Rainbow tables or hash tables are free online tables that store password hashes of common passwords. LM Hash was a password hashing protocol used for Windows systems. This scheme cannot generate a hash for a password longer than fourteen characters. Password cracking is a major threat that affects one's privacy. So, storing only salted hashes of passwords reduces the effectiveness of most computational password attacks.

Navjyotsinh Jadejaa[3] and Viral Parmarb[3] focused on novel approaches of testing various tools that can be used to measure the potential helplessness of a digital system from assaults of particular sorts that uses lateral movement and privileged heightening, such as Pass the Hash. It's an attack that can gain access to the hash of the password, there won't be any need to get a password as the authentication process is a comparison between hashes in the windows system. These password hashes can be dumped by the attacker, using hash dump tools. This attack is less time consuming than other attacks like password guessing, password cracking. The various tools used they used were Pwdump7, Windows credentials Editor, Fgdump. These tools were tested to the system that had anti-virus like Bit Defender (Paid), Microsoft Essential Security, AVG Antivirus, installed. fgdump was considered as a better version of pwdump. The difference between pwdump and fgdump is that pwdump crashes when Anti-Virus is active where fgdump first shut the Anti-Virus and then runs its script. They found out that one of the reasons why that attack could be implementable was because of the vulnerability of windows. They came to an understanding that the attacker already has the administrative rights in the victim's system. If that could be avoided or secured then multiple attacks would be mitigated. Several ideas and methods were introduced by them. Systems which are not trustworthy and less secure, should not be allowed to manipulate data was the first solution they came up with. The next solution was to use the Least User Access approach given by Microsoft as a study showed that 92% of threats can be solved by implementing this approach. In this approach, admin rights can be revoked from these users. Even though this approach cannot be implemented in all circumstances and systems, but it reduces the risks. Decreasing the limit of Cached Credentials was the next solution they came up with since some organizations use the same passwords. Because of this, if one system gets compromised then the attacker can gain the password credentials from that system and get to the main domain in no time. To keep the system safe, organizations should keep their security updating regularly.

Yu Tsuda[4] and teammates proposed a lightweight host-based intrusion detection system by using process generation patterns. Their system periodically collected lists of active processes from each host, then the system constructed process trees from those lists. Plus the system detected anomaly processes from the process trees considering parent-child relationships, execution sequences and lifetime of processes. The reason behind the making of this paper was due to the rise in Advanced Persistent

Threat (APT). This threat, at first, tries to penetrate targeting organizations by using a backdoor. After intruding, they usually execute OS built-in commands, management tools published by OS vendors, etc. to gain more privileged access in that system or network. PowerShell is one such tool so extremely powerful that attackers are increasingly using PowerShell in their attack methods (mainly to gain privilege). After creating their system, they deployed it to their organizations for testing. From 498 hosts, 2,403,230 process paths in total and 4,120 unique process paths were obtained. Then they could detect 38 anomaly processes by using the system. Among the anomaly processes, there was a PowerShell process created by a macro in Microsoft Excel. It was also detected by using an antivirus software running on our organization.

Sanjay B N[5], Rakshith D C[5], Akash R B[5] and Dr.Vinay V Hegde[5] in their paper discusses the technical details of Fileless malware and their related attacks in depth. This paper also includes various Fileless malware detection and mitigation techniques in detail.

Malware, or malicious program, refer to any malicious program or code written to disk in one form or another that requires execution in order to carry out their malicious activities. They are designed to alter, damage or gain access over the victim's computing systems. On the other hand, Fileless malware is purposed to be memory resident rather than writing artifacts to the file system, ideally leaving no trace after its execution making it difficult to identify. After writing malicious content to memory, hackers sometimes try to gain persistence on the system and seek out control over legitimate user applications and system administration tools such as PowerShell and Windows Management Instrumentation. These are software applications developed to install themselves in the background on a host computer and affect the users in some way.

Fileless malware has been developed by the attackers in a way that it becomes very hard for antivirus software to detect one. Antivirus software begins by scanning files from the specified location and comparing them to specific bits of code that is usually a hash key, against information in its database. If any match is found, it is considered a virus and deletes that particular file. Because fileless malware does not require a file to be downloaded, it is difficult to prevent, detect, and remove. The only optimal scenario is to reboot your machine. Since RAM is a volatile memory, it only keeps its data when your computer is on. Once the power is off, the malicious malware is no longer live.

With the release of .NET framework, it became easy for malware coders to spread malware and achieve the goals of the malware. The popular host platform of attacking is by using PowerShell. PowerShell is tightly integrated into the Microsoft Windows environment and it is hard and impractical for many system administrators to turn it off. PowerShell scripts can be loaded into the memory of the operating system which can execute commands without writing files to the hard drive. By dynamically loading the PowerShell scripts, the malware is able to attack the system without leaving any file-based evidence. This ability also provides the ability to hide from file-based antivirus protection.

III. PROBLEM DEFINITION

Step 1: Configure PupyRat in Kali Linux

Fire up Kali Linux Machine, open up the terminal, change the directory to your wish and clone the tool from https://github.com/n1nj4sec/pupy.git using the git clone command.

Step 2: Now change the directory to the pupy

Use the commands git submodule init and git submodule update, to pull the code from the submodule and update it. Before launching the tool, the system needs to satisfy some requirements to make the tool run properly. Use the command:

pip install -r pupy/requirements.txt

Step 3: Run the PupyRat

The last command will install all the necessary packages required to run the tool without error. Now it's time to launch the tool. Run it by the command:

./pupygen.py

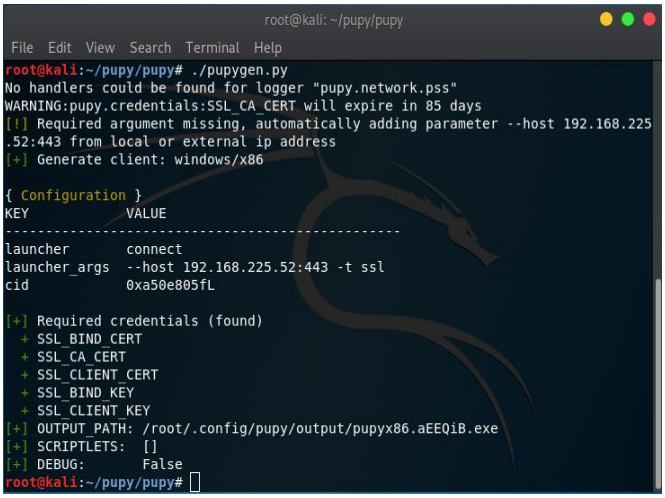

Fig 4.1

It creates a payload and saves it as .exe file(default) in the path root/.config/pupy/output.

Step 4: Run the PupyRat listener

The payload can be integrated into exe files and it is then run in the target system so that pupyshell can establish a session acting as a listener. To start the server, start pupysh.py on the correct port with the correct transport:

./pupysh.py

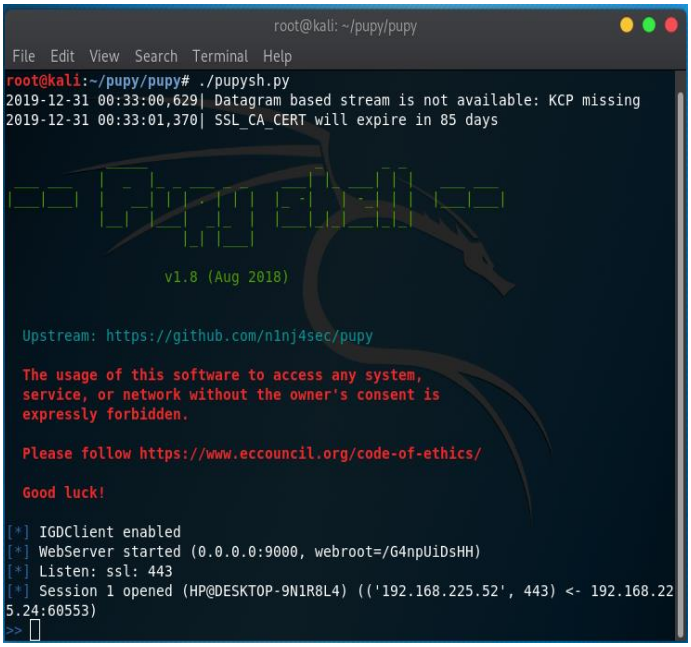

Fig 4.2

Step 5: Explore the modules

Pupy has several capabilities in the form of modules that can be initiated. To explore and understand these modules type:

help –M

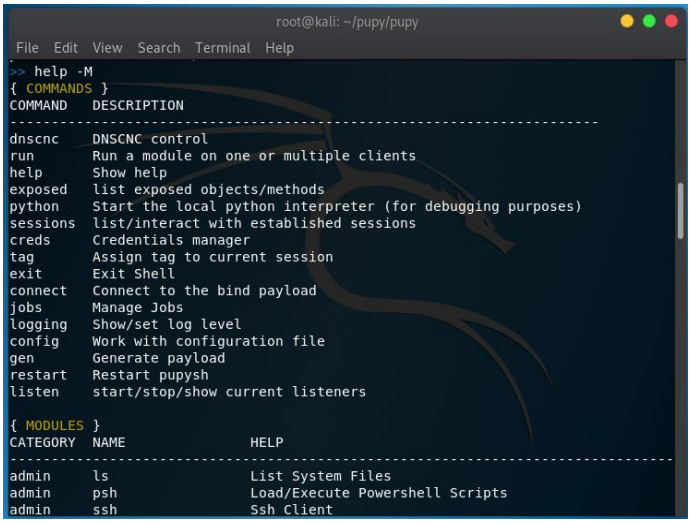

Fig 4.3

Step 6: Bypassing the UAC

A normal or a middle-level user can't obtain the credentials of a system. To gain access to those hashed credentials the attacker has to bypass the UAC and recreate the current program as an admin. First, a PowerShell payload is generated to target system disabling file system redirection, creating delegateExecute key using the command: **bypassuac**

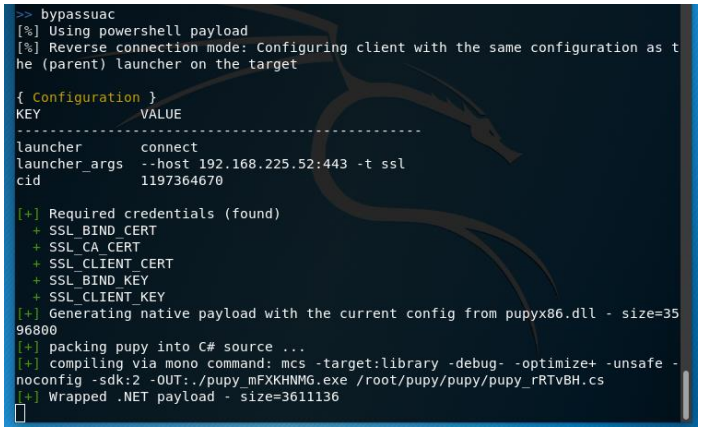

Fig 4.4

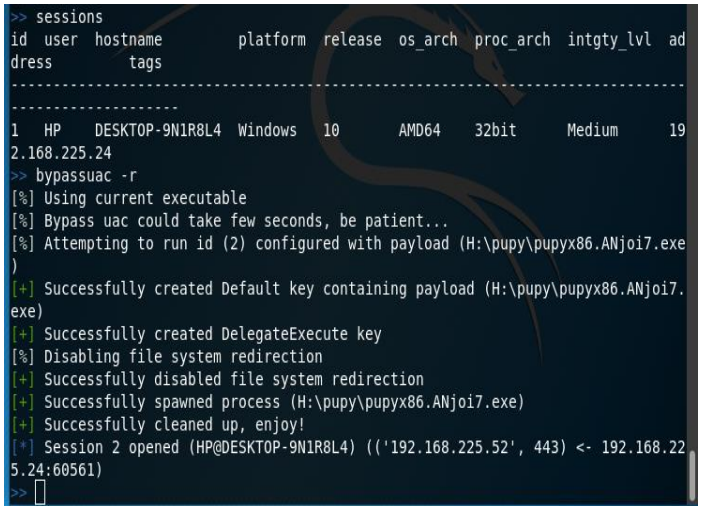

Fig 4.5

When a delegateExecute key is created for the current executable we use the command: **bypassuac –r** to restart the current executable. A new session is created with high integrity.

Step 7: Obtain system credentials

Once the admin privilege is gained using the bypassuac module, the attacker can then use the command: **run creddump**.

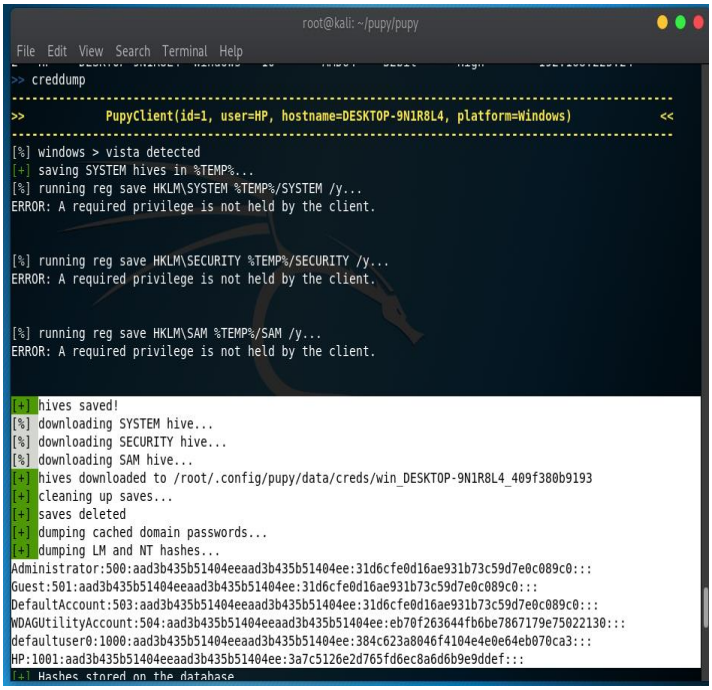

Fig 4.6

Creddump is a module that retrieves system hashes of windows. It downloads the SAM hive where the windows credentials are stored as hashed.

Step 8: Cracking windows password

It's possible to crack the windows password from its hashed form. Technically reversing a hashing is impossible but some tools and sites like crackstation.net use wordlist that acts as a candidate password. Passwords that are common and in the wordlist can be easily cracked.

Fig 4.7

IV. IMPLEMENTATION

It's is noticed from the target system that a PowerShell window opens and closes in the blink of an eye. A user with no computer background may not understand and think of it as a glitch. Thus the user will not be able to take preventive measures to ensure the safety of the system. This might also be a similar case with an experienced user. A detection system is developed to alert the users about a possible RAT attack. The users are made aware of the situation so that they can prevent the chances of being compromised. Fig 5.1 is a flowchart that shows the working of ratdetection system developed in python.

Algorithm of detection system:

1. start
2. set timer=15
3. while (timer>0)
4. Search for the process called PowerShell.
5. if PowerShell found then
6. return process_id, process_name, creation time
7. print alert of a possible attack.
8. If process like 'pupy%'
9. Eliminate process
10. Print 'possible threat eliminated'
11. End if
12. else timer=timer-1
13. end if
14. end while
15. Stop

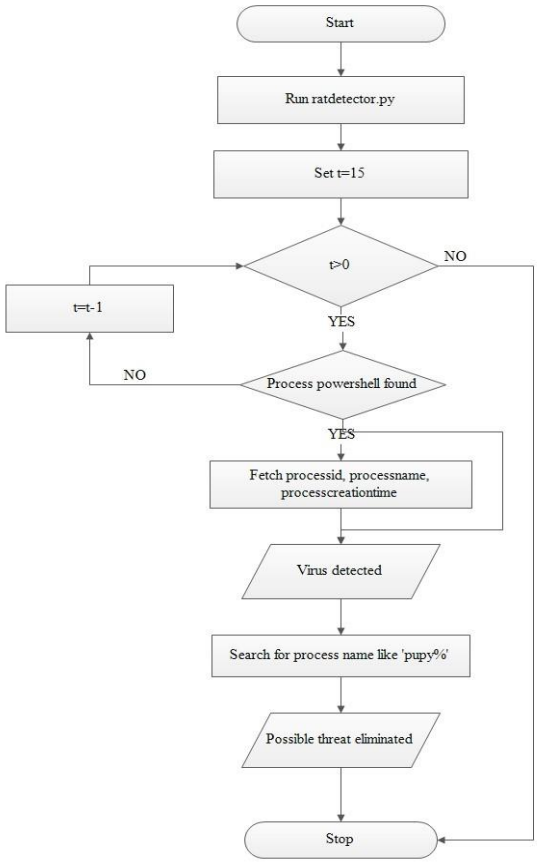

Fig 5.1 Flowchart of detection system

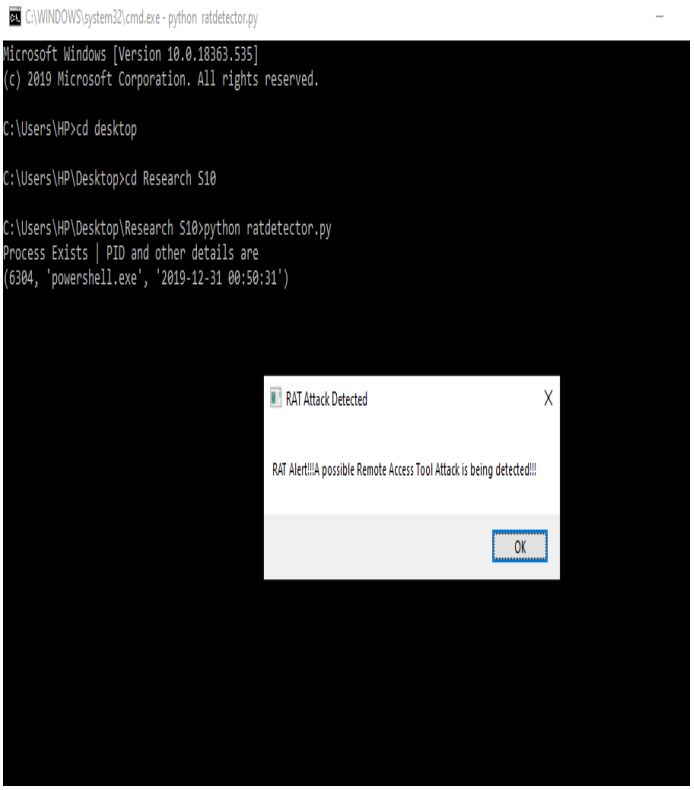

Fig 5.2 an example of alert

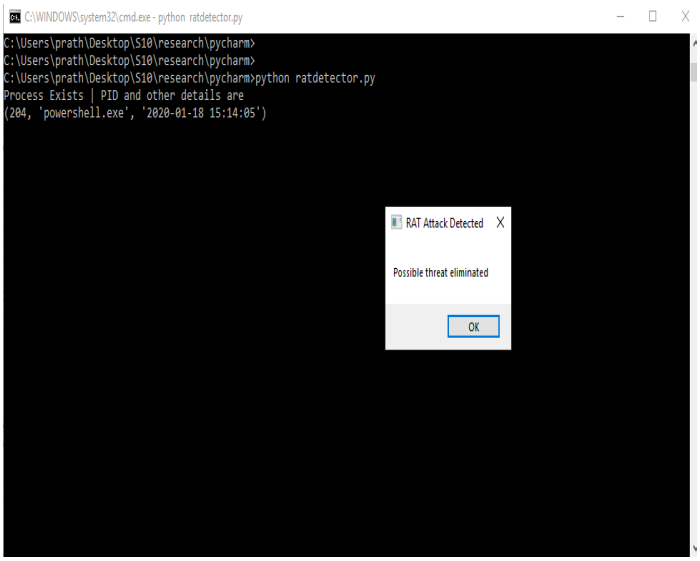

Fig 5.3 process termination

V. RESULT

Here PupyRat exploits remote systems by retrieving its credentials. The detection system developed in Python detects a PowerShell attack, alerts the user about the background process and terminates it.

VI. CONCLUSION

It becomes easier for attackers to get credentials of a remote system if the user is unaware of the process running in the background. Attacks by loading PowerShell scripts has increased as it is able to attack the system without leaving any evidence. PupyRat is such a tool used by attackers to crawl into the target system using the PowerShell. While running the tool, the PowerShell was sighted multiple times on the target machine as it opened and closed continuously. The detection system developed here detects the attack when the attacker tries to use PowerShell for gaining admin privilege. This is done by monitoring the presence of PowerShell, alerting the user about the background process and then terminating it is found as the hunted anomaly. As a result, the target system is protected from the attacker's clutch. In addition to the developed system, tracking back the attacker would prove an important area for future research.

REFERENCES

[1] Anis Ismail, Mohammad Hajjar, Haissam Hajjar: Remote Administration Tools:A comparative study
Journal of Theoretical and Applied Information Technology.
http://www.jatit.org/volumes/research-papers/Vol4No2/Remote%20Administration%20Tool%20(RAT),%20LAN,%20WAN,%20control,%20WMI,%20RPC,%20Web.pdf

[2] Sam Martin and Mark Tokutomi: Password Cracking.
https://www2.cs.arizona.edu/~collberg/Teaching/466-566/2012/Resources/presentations/topic7-final/report.pdf

[3] Navjyotsinh Jadejaa, Viral Parmarb: Implementation and mitigation of various tools for pass the hash attack
https://www.sciencedirect.com/science/article/pii/S1877050916002301

[4] Yu Tsuda, Junji Nakazato, Yaichiro Takagi, Daisuke Inoue, Koji Nakao and Kenjiro Terada: A Lightweight Host-Based Intrusion Detection based on Process Generation Patterns
2018 13th Asia Joint Conference on Information Security.
https://ieeexplore.ieee.org/document/8453769

[5] Sanjay B N, Rakshith D C, Akash R B, Dr.Vinay V Hegde: An Approach to Detect Fileless Malware and Defend its Evasive mechanisms
3rd IEEE International Conference on Computational Systems and Information Technology for Sustainable Solutions 2018.
https://ieeexplore.ieee.org/abstract/document/8768769