June 21, 2023

Table of Contents

Executive Summary

Synopsis

Conducted a lifelike pentest on the relevant penetration testing challenge on the TryHackme website. This challenge was meant to test my skills and prepare myself to take the eJPT certification exam. This test was conducted on June 16, 2023, and the goal of this simulated pentest is to put my skills and everything I have learned so far to practical use. I will treat this as if it were a real-life pentesting engagement and write a detailed report about my vulnerability findings.

Findings Overview

While conducting the internal penetration test there were several critical vulnerabilities discovered in the Relevant Penetration Testing Challenge server. I was able to gain full administrative access to the relevant server, this was possible due to a vulnerable web application and an SMB share that did not require user credentials to login.

- Target: **10.10.31.246** - A low privilege shell was obtained by performing an upload vulnerability attack against the web server by uploading a malicious payload to the "nt4wrksv" SMB share and navigating to the following url: http//10.10.31.246:49663/nt4wrksv/payload.aspx, to execute the payload. Once the payload was executed. I was able to get a shell on the system and escalate my Privileges to the local administrator.

Critical	9.0 - 10.0

Recommendations

To increase the security posture of Relevant, I recommend the following mitigations and remediations.

- Require a valid set of user credentials to access the "nt4wrksv" SMB share.
- Reconfigure the web server so that the "nt4wrksv" SMB share cannot be accessed through the browser and possibly allow attackers to execute arbitrary files.
- Disable SeImpersonatePrivilege to prevent local privilege escalation.

Severity Scale

Rating	Score
Low	0.0 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0

Severity Level: Low

Vulnerabilities in the low range typically have very little impact on an organization's business. Exploitation of such vulnerabilities usually requires local or physical system access.

Severity Level: Medium

Vulnerabilities in the medium range usually have some of the following characteristics:

- Vulnerabilities that require an attacker to manipulate victims into disclosing sensitive information or gaining access to systems also known as social engineering
- Denial of service attacks
- Exploits that require the attacker to be on the same local area network (LAN) as the target.
- Vulnerabilities where exploitation provides limited access.
- Vulnerabilities that require user privileges for successful exploitation.

Severity Level: High

Vulnerabilities in the high score range usually have these characteristics:

- Vulnerabilities that are difficult to exploit.
- Exploitation could result in elevated privileges.
- Exploitation could result in a significant data loss or downtime.

Severity Level: Critical

Vulnerabilities in the critical score range have most of the following characteristics:

- Exploitation will likely result in root level compromise of servers or infrastructure devices.
- Exploitation is usually straightforward, in the sense that the attacker does not need any Special Authentication credentials or knowledge about individual victims, and does not need to persuade a target using social engineering.

Final Report

Methodology

In this penetration testing challenge, I employed testing methods that are widely adopted in the cyber security assessment industry. This includes 5 phases: Information Gathering, Enumeration, Vulnerability Assessment, Exploitation and Reporting/Mitigation. During these phases, both automated and manual auditing techniques were used to ensure the best possible results.

Information Gathering

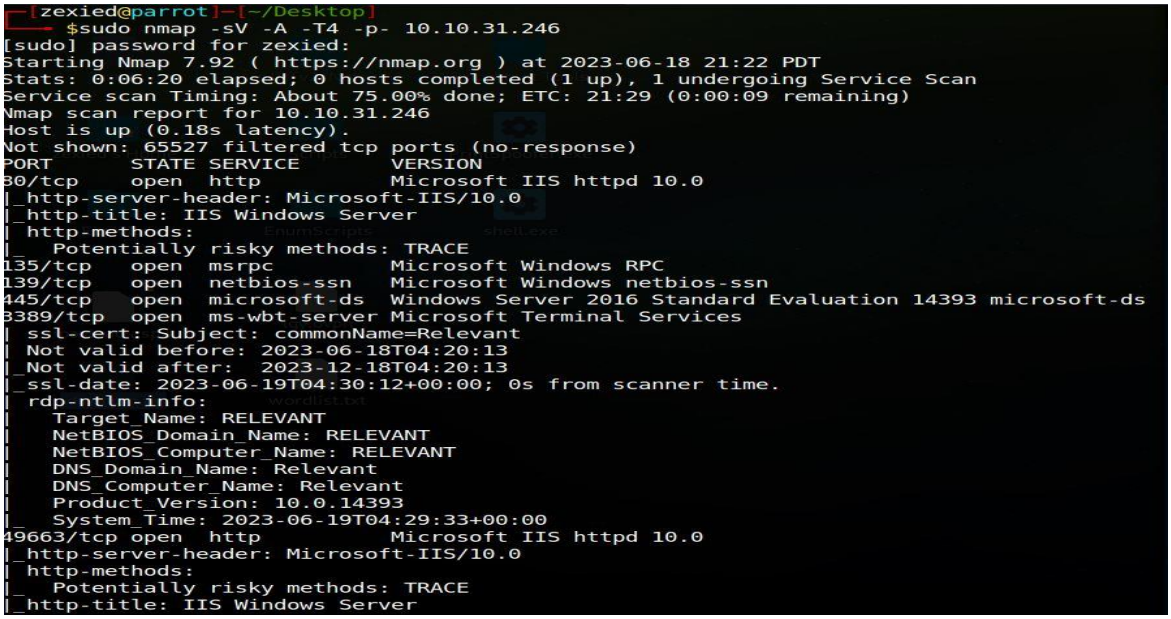
You can see the details of the network device listed below.

- Hostname: RELEVANT
- IP Address: 10.10.31.246

I was able to verify the IP address of the hostname by performing a basic nmap scan on the target as listed below.

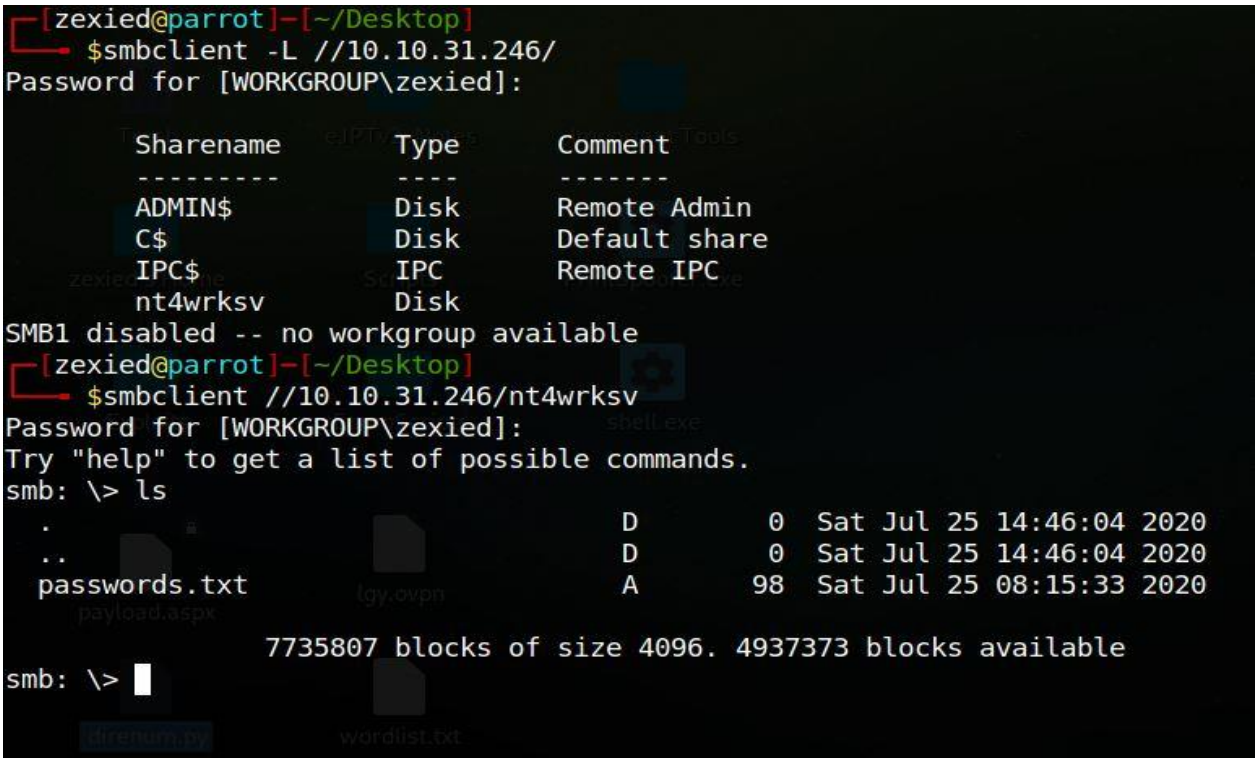
Enumeration

Used nmap to perform service enumeration on the target ip and discover information about the server that may reveal critical details that could be leveraged to bypass security and gain an initial foothold on the system.



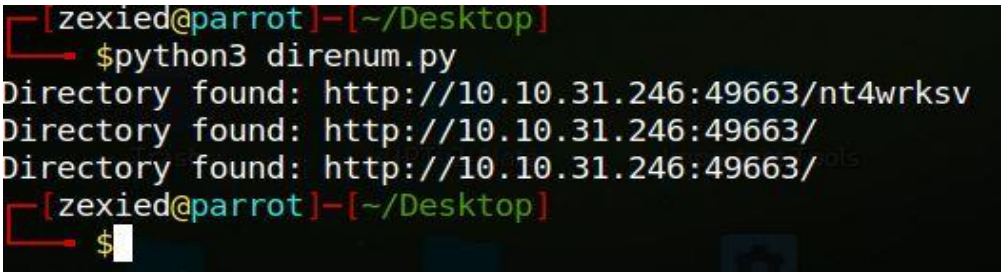
As you can see from the scan above, I started by scanning all ports with the -p- nmap command argument to reveal all ports running on the host. The initial nmap scan revealed many active services running on the target including two Microsoft IIS httpd 10.0 web services on port 80 and port 49663, SMB on port 139 and port 445 as well as RDP on port 3389.

The first thing I decided to look at after scanning the server was to enumerate the SMB service using the tool smbclient. My initial findings were significant in that I managed to find a null session on the “nt4wrksv” SMB share listed below.



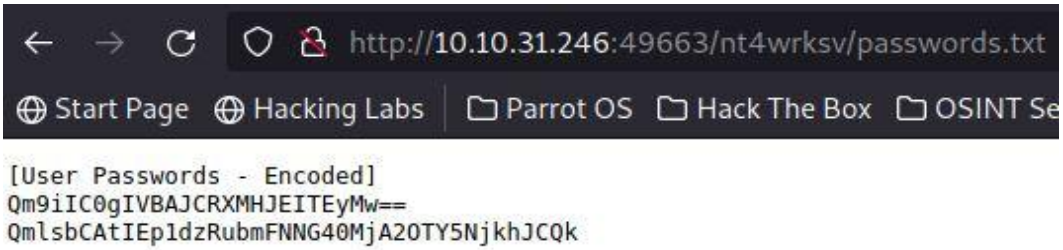
As you can see there is a file called “passwords.txt” that had some credentials inside but luckily I was not able to log into the server with them using RDP or Psexec.

The next step for me was to enumerate the web server for hidden web directories. I scanned the web server on port 80 and found nothing of interest until I scanned the other service running on port 49663. I found a web directory called “nt4wrksv” which was coincidentally the same name as the SMB share I previously enumerated.



I thought it was strange that the hidden web directory on port 49663 was also called “nt4wrksv” so I figured the SMB share was somehow linked to the web server. My suspicions were confirmed when I navigated to the passwords.txt file from the “nt4wrksv” SMB share through my browser and found found the contents of “passwords.txt” in my browser as you can see from the screenshot below.

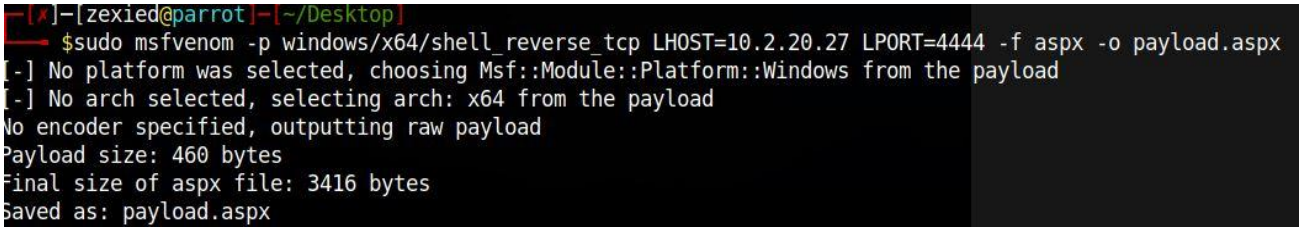
.



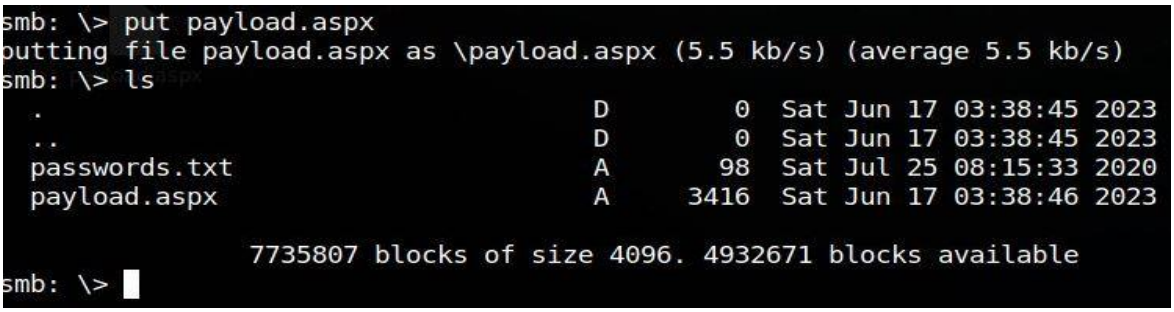
Exploitation

Gaining a Low-Privilege Shell

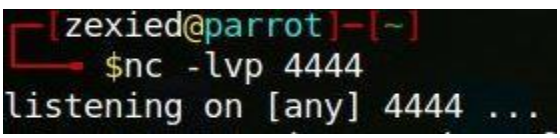
I was successful in gaining Remote Code Execution (RCE) by means of the following method.



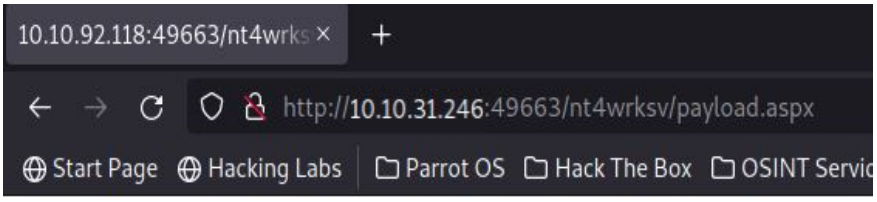
I generated a reverse shell payload using msfvenom.



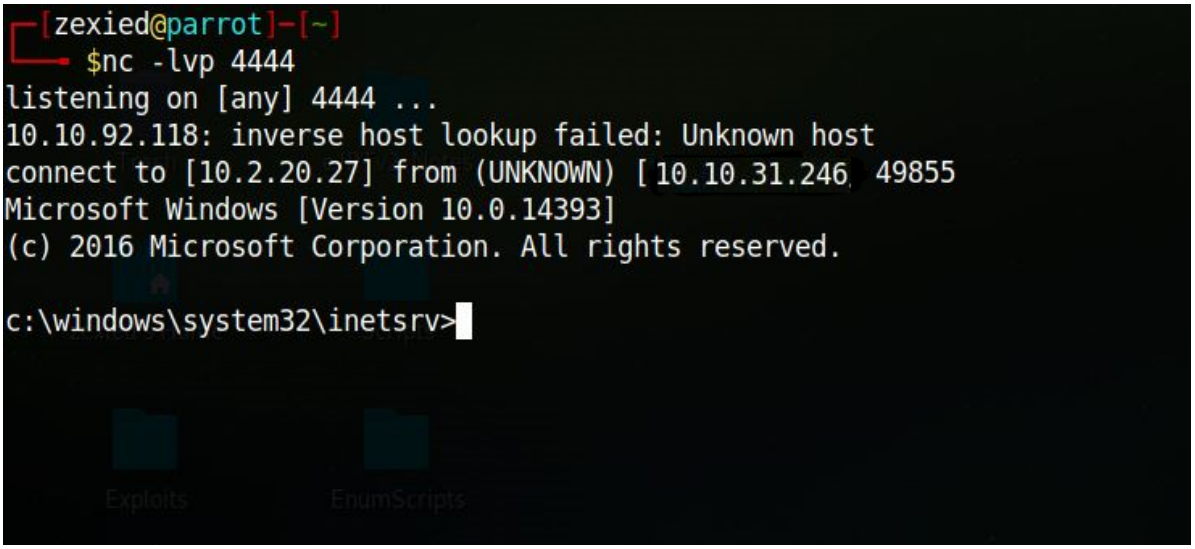
After that I uploaded the reverse shell to the SMB share using the “put” command.



Once the shell was uploaded I started a Netcat listener to listen for connections on port 4444 To catch the initial shell.



After navigating to the following url you can see in the screenshot above, the web application executed my payload and I was able to get a low privilege shell as seen in the screenshot below.



Post Exploitation

Escalating Privileges

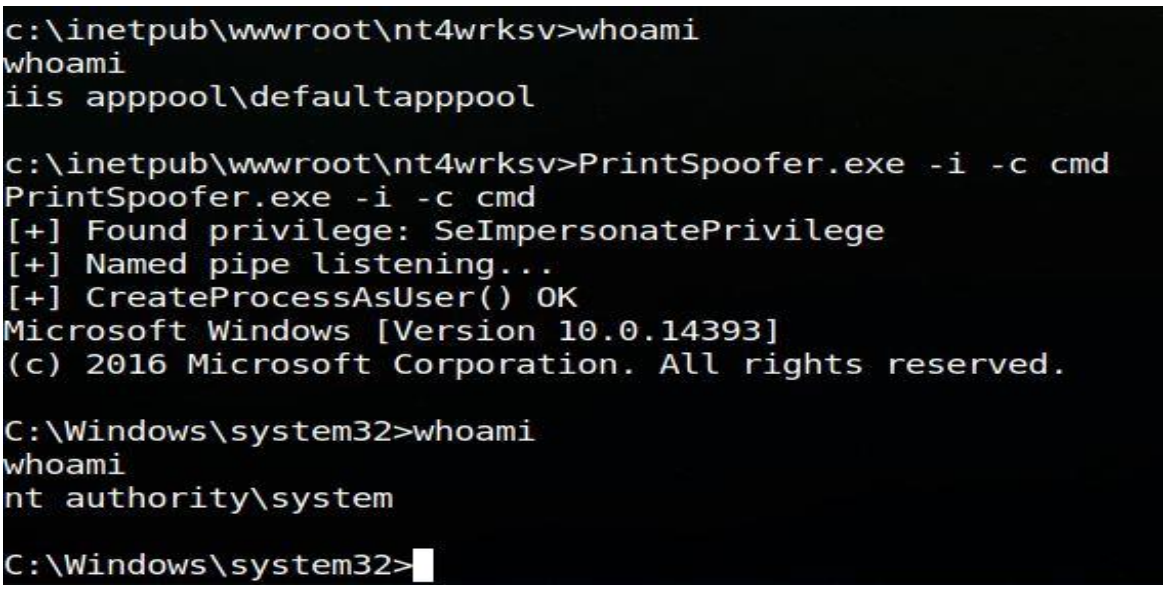
After the initial exploitation was complete, I then attempted to escalate my privileges.
The first thing I did after getting the shell was run the command whoami /priv to display the Security privileges of the current user.

```
c:\windows\system32\inetsrv>whoami /priv
whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeAuditPrivilege              Generate security audits                  Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled

c:\windows\system32\inetsrv>whoami
whoami
iis apppool\defaultapppool

c:\windows\system32\inetsrv>
```

In this screenshot you can see that the SeImpersonsatePrivilege is set to enabled. This means that the user has the privilege to impersonate other users on the system. This is potentially dangerous because we could abuse this privilege to gain administrative access. One fairly recent exploit that takes advantage of this weakness is called PrintSpoofer. https://github.com/itm4n/PrintSpoofer

```
smb: \> put PrintSpoofer.exe
putting file PrintSpoofer.exe as \PrintSpoofer.exe (31.7 kb/s) (average 20.7 kb/s)
smb: \>
```

I uploaded the PrintSpoofer exploit to the “nt4wrksv” SMB share.



After uploading the PrintSpoofer exploit to the “nt4wrksv” SMB share I then navigated to the “nt4wrksv” file path in my shell where the share was being hosted from. As you can see in the screenshot above the command “whoami” returned that I was running as the iis apppool\defualtapppool account when After I ran the exploit and ran the “whoami” command again, it returned the nt authority\system account indicating that I successfully elevated my privileges to administrator.

House Cleaning

During a penetration testing engagement, tools, files, user accounts, etc, are created on the client’s systems which would compromise the client’s security. That’s why I was diligent to ensure that no potential security issues to the Relevant system through remnants left on the client’s systems after the completion of the engagement. Relevant has had all tools, files, user accounts, etc that were used during the engagement removed accordingly.