IT-Security (ITS) B1

DIKU, E2020

Lecture plan

36	31 Aug	10-12	TL	Introduction, security concepts and the threat of hacking
	04 Sep	10-12	TL	Buffer overflow
37	07 Sep	10-12	CJ	Software security, Operating system security
	11 Sep	10-12	CJ	User authentication and access control
38	14 Sep	10-12	TL	Malicious software
	18 Sep	10-12	CJ	Firewalls and denial-of-service attacks
39	21 Sep	10-12	CJ	Cloud and IoT
	25 Sep	10-12	TL	Cryptography
40	28 Sep	10-12	TL	Internet security protocols
	02 Oct	10-12	TL	Intrusion detection
41	05 Oct	10-12	TL	Forensics
	09 Oct	10-12	CJ	IT security management
42				Fall Vacation - No lectures
43	19 Oct	10-12	CJ	Privacy 1
	23 Oct	10-12	CJ	Privacy 2
44	26 Oct	10-11	Guest	Final guest lecture
		11-12	All	Recap and Q/A
45	xx Nov			Exam

Today’s agenda

Memory forensics

Disk forensics

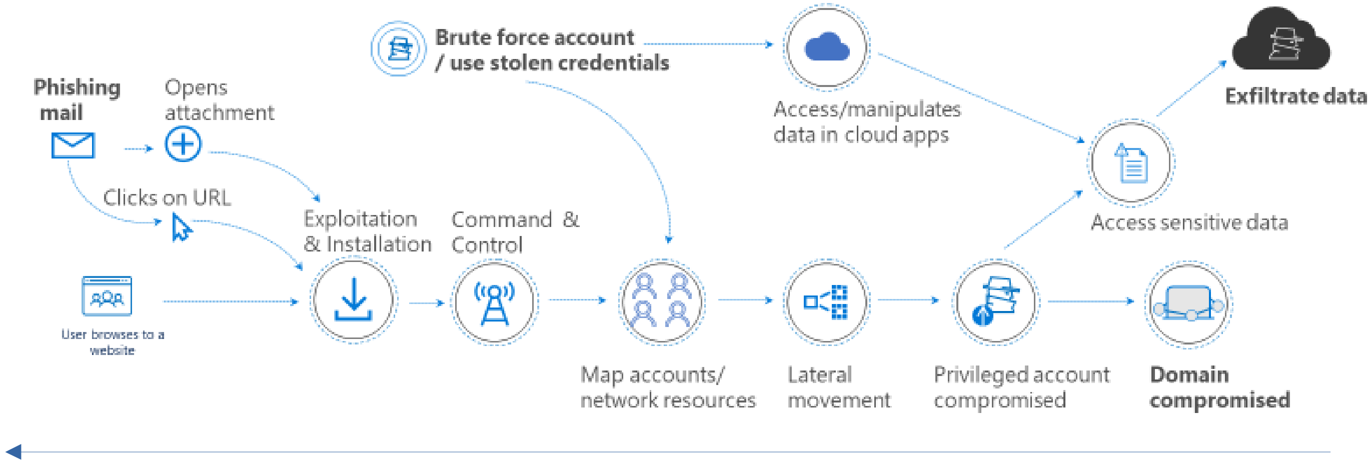
Log analysis

Malware analysis

Recap – Intrusion Detection

Host and network analysis

IOCs and anomalies



Brute force account / use stolen credentials
Phishing mail
Opens attachment
Clicks on URL
User browses to a website
Exploitation & Installation
Command & Control
Map accounts/ network resources
Lateral movement
Privileged account compromised
Domain compromised
Access/manipulates data in cloud apps
Access sensitive data
Exfiltrate data

Forensics vs Incident Response

Formally, **digital forensics** is a branch of forensic science encompassing the recovery and investigation of material found on digital devices, often in relation to crimes

Incident response involves the execution of proper responses to computer intrusions

In practice, when responding to computer intrusions, they are used interchangeably

Add: **Malware analysis**

DFIRMA

Sidebar: Digital forensics

Digital forensics =

Computer forensics

Memory forensics

Network forensics

Mobile forensics

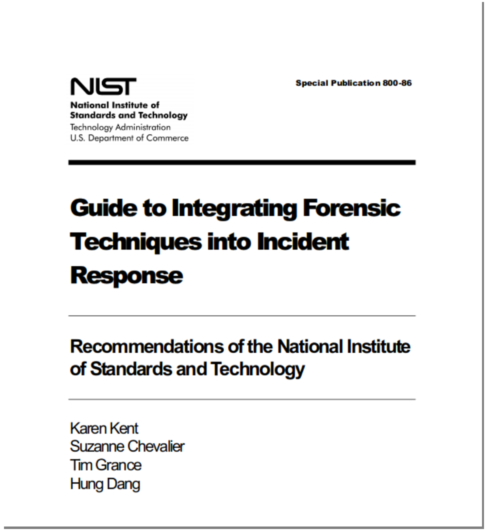
Etc. forensics

In practice, they coexist

```
while true:
    intrusion analysis

    if intrusion suspected:
        preliminary analysis (triage)

        if intrusion verified:
            repeat until incident fully grasped:
                incident analysis
                forensic analysis
                malware anaysis

            incident response

    update plans
```

A note on today’s reading material

National Institute of Standards and Technology (NIST) SP 800-56



NIST
National Institute of
Standards and Technology
Technology Administration
U.S. Department of Commerce

Special Publication 800-86

Guide to Integrating Forensic Techniques into Incident Response

Recommendations of the National Institute of Standards and Technology

Karen Kent
Suzanne Chevalier
Tim Grance
Hung Dang

Spearphishing, revisited

Memory forensics

Situation: Evil code is running

Out job: Find it in memory

Memory forensics

From Wikipedia:

“Memory forensics is forensic analysis of a computer's memory dump.

Its primary application is investigation of advanced computer attacks which are stealthy enough to avoid leaving data on the computer's hard drive.”

First, get a copy

- Live acquisition
    - Different techiniques
- Live analysis
    - Direct analysis of the running kernel
- Dead acquisition
    - Hibernation files, page files
- Virtualization - thank you

What to find in memory?

Running processes

Listening sockets

Open connections

Loaded modules

Encryption keys

Credentials

Memory only malware

Closed connections

Terminated processes

Open file handles

Memory forensic process

1: Find rogue processes

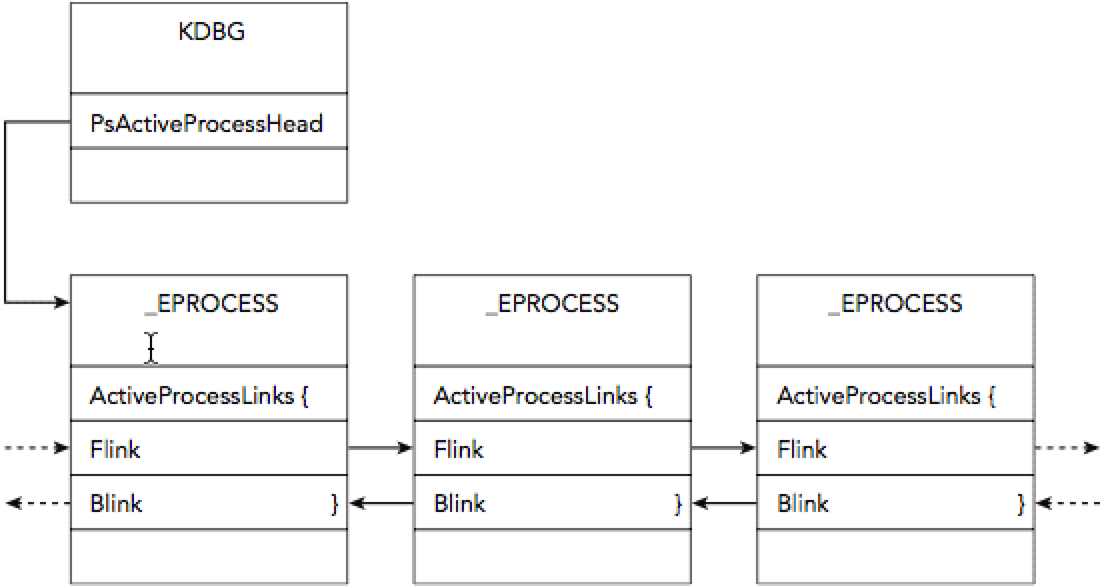
2: Analyse DLLs

3: Review network artefacts

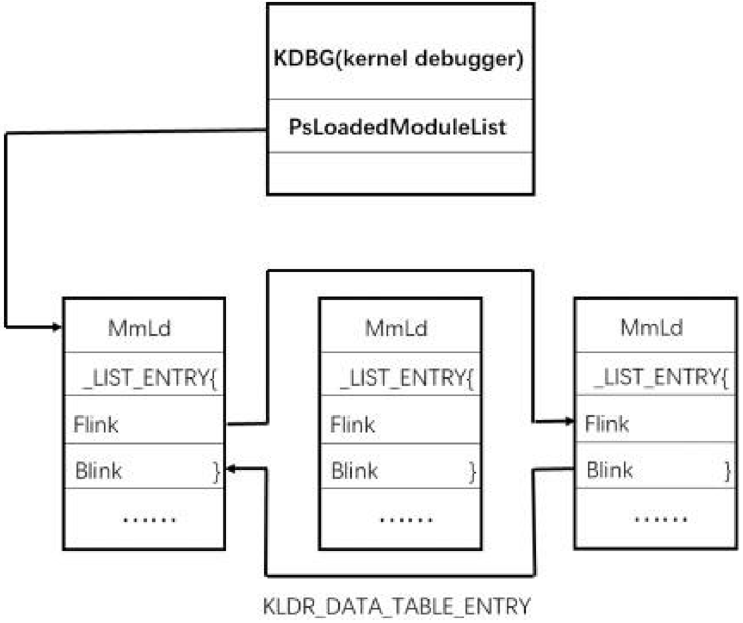
4: Look for evidence of code injections

5: Dump suspicious processes → further analysis

How to find it – process enumeration



Direct kernel objection manipulation (DKOM)

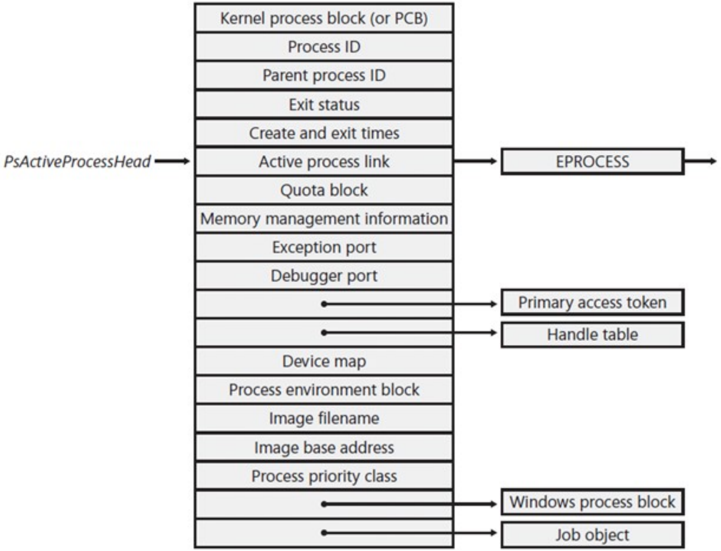


How to find it – scanning for processes

Key concepts in memory forensics

Walking a list, or

Scanning for objects



PsActiveProcessHead

Kernel process block (or PCB)

Process ID

Parent process ID

Exit status

Create and exit times

Active process link

Quota block

Memory management information

Exception port

Debugger port

Device map

Process environment block

Image filename

Image base address

Process priority class

EPROCESS

Primary access token

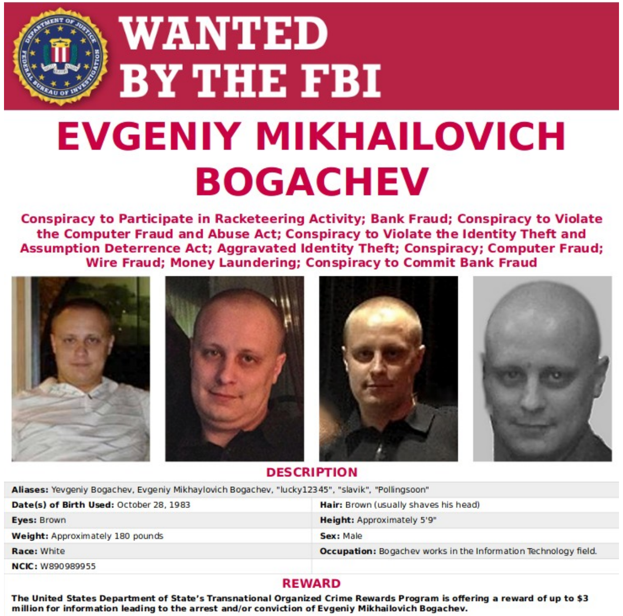
Handle table

Windows process block

Job object

Walk a list vs scanning

Zeus infection



WANTED BY THE FBI

EVGENIY MIKHAILOVICH BOGACHEV

Conspiracy to Participate in Racketeering Activity; Bank Fraud; Conspiracy to Violate the Computer Fraud and Abuse Act; Conspiracy to Violate the Identity Theft and Assumption Deterrence Act; Aggravated Identity Theft; Conspiracy; Computer Fraud; Wire Fraud; Money Laundering; Conspiracy to Commit Bank Fraud

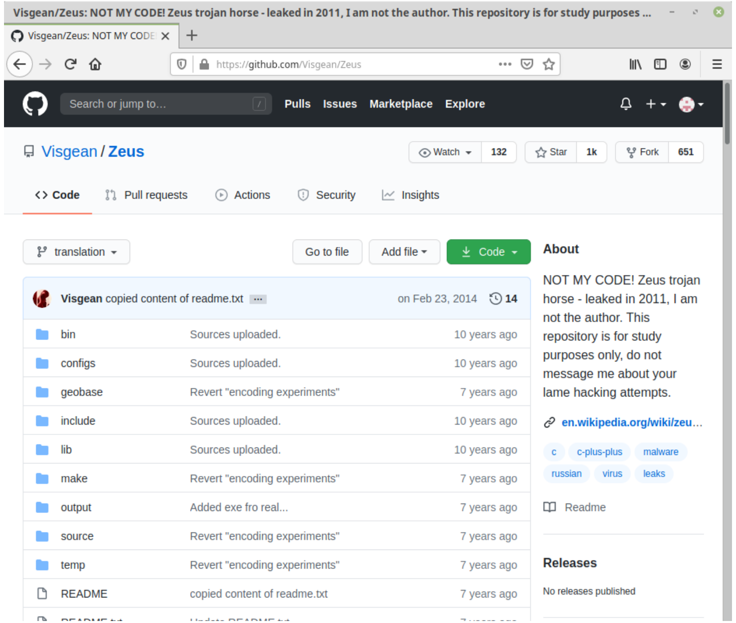
DESCRIPTION

Aliases: Yevgeniy Bogachev, Evgeniy Mikhaylovich Bogachev, "lucky12345", "slavik", "Pollingsoon"

Date(s) of Birth Used: October 28, 1983	**Hair:** Brown (usually shaves his head)
Eyes: Brown	**Height:** Approximately 5'9"
Weight: Approximately 180 pounds	**Sex:** Male
Race: White	**Occupation:** Bogachev works in the Information Technology field.
NCIC: W890989955	

REWARD

The United States Department of State's Transnational Organized Crime Rewards Program is offering a reward of up to $3 million for information leading to the arrest and/or conviction of Evgeniy Mikhailovich Bogachev.



Visgean/Zeus: NOT MY CODE! Zeus trojan horse - leaked in 2011, I am not the author. This repository is for study purposes ...

https://github.com/Visgean/Zeus

Visgean copied content of readme.txt … on Feb 23, 2014 14

bin	Sources uploaded.	10 years ago
configs	Sources uploaded.	10 years ago
geobase	Revert "encoding experiments"	7 years ago
include	Sources uploaded.	10 years ago
lib	Sources uploaded.	10 years ago
make	Revert "encoding experiments"	7 years ago
output	Added exe fro real...	7 years ago
source	Revert "encoding experiments"	7 years ago
temp	Revert "encoding experiments"	7 years ago
README	copied content of readme.txt	7 years ago
README.txt	Update README.txt	7 years ago

About

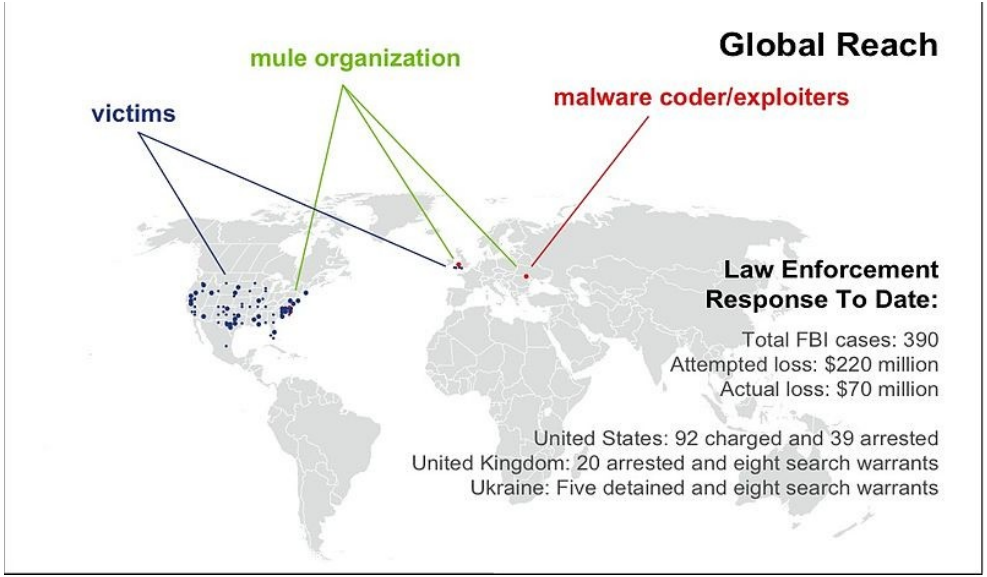
NOT MY CODE! Zeus trojan horse - leaked in 2011, I am not the author. This repository is for study purposes only, do not message me about your lame hacking attempts.

en.wikipedia.org/wiki/zeu...

Releases

No releases published

Zeus infection

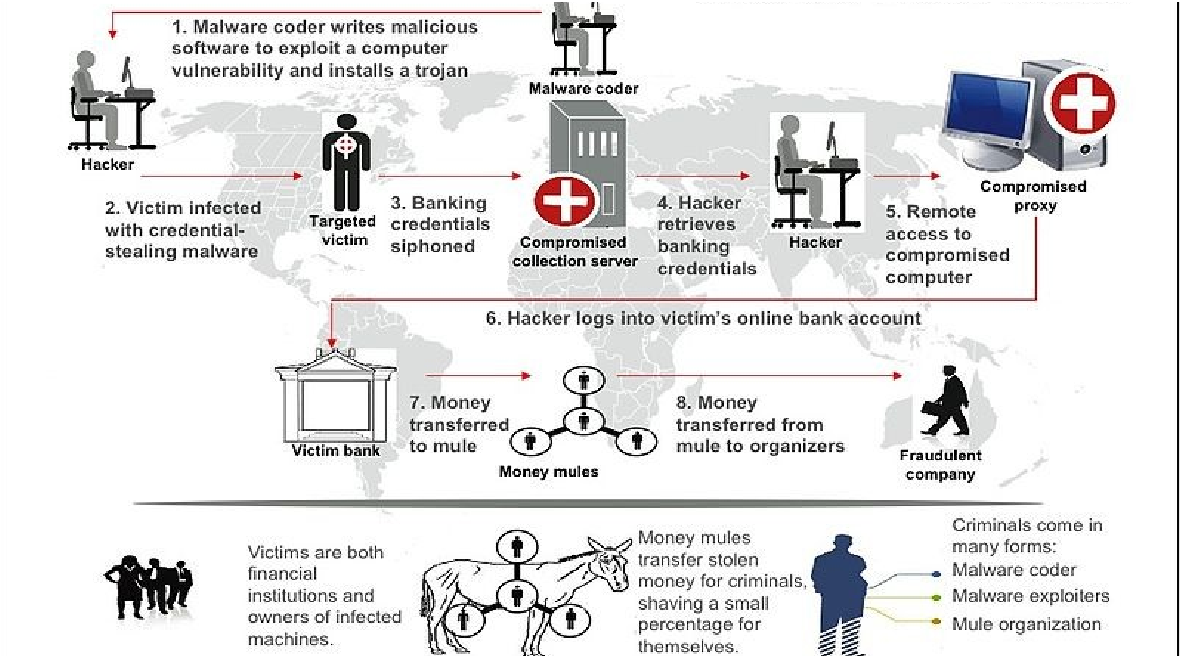


Law Enforcement Response To Date:

Total FBI cases: 390
Attempted loss: $220 million
Actual loss: $70 million

United States: 92 charged and 39 arrested
United Kingdom: 20 arrested and eight search warrants
Ukraine: Five detained and eight search warrants

Zeus infection


1. Malware coder writes malicious software to exploit a computer vulnerability and installs a trojan
Malware coder
Hacker
2. Victim infected with credential-stealing malware
Targeted victim
3. Banking credentials siphoned
Compromised collection server
4. Hacker retrieves banking credentials
Hacker
5. Remote access to compromised computer
Compromised proxy
6. Hacker logs into victim's online bank account
Victim bank
7. Money transferred to mule
Money mules
8. Money transferred from mule to organizers
Fraudulent company
Victims are both financial institutions and owners of infected machines.
Money mules transfer stolen money for criminals, shaving a small percentage for themselves.
Criminals come in many forms:
Malware coder
Malware exploiters
Mule organization

The life of a network connections struct

Socket()

Bind()

Listen()

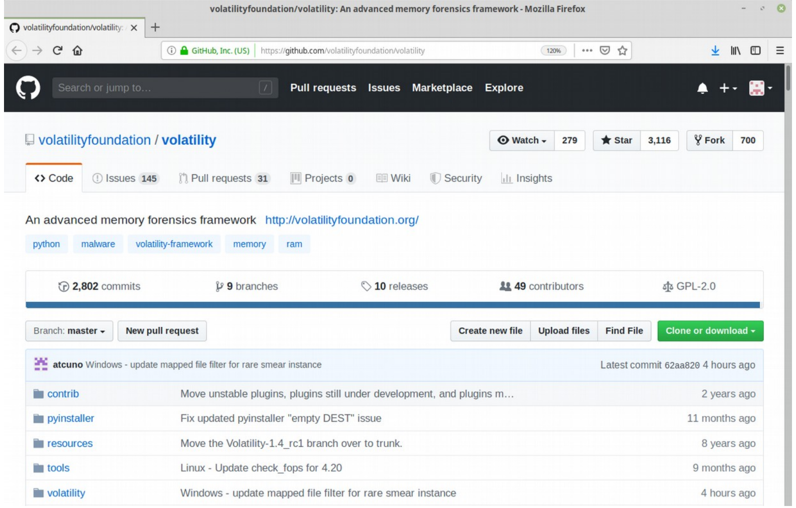
Closesocket()

Deallocate

Example memory analyses

Volatility

Volatility is an open source memory analysis framework writtin in Python



volatilityfoundation/volatility: An advanced memory forensics framework - Mozilla Firefox

https://github.com/volatilityfoundation/volatility

volatilityfoundation / volatility

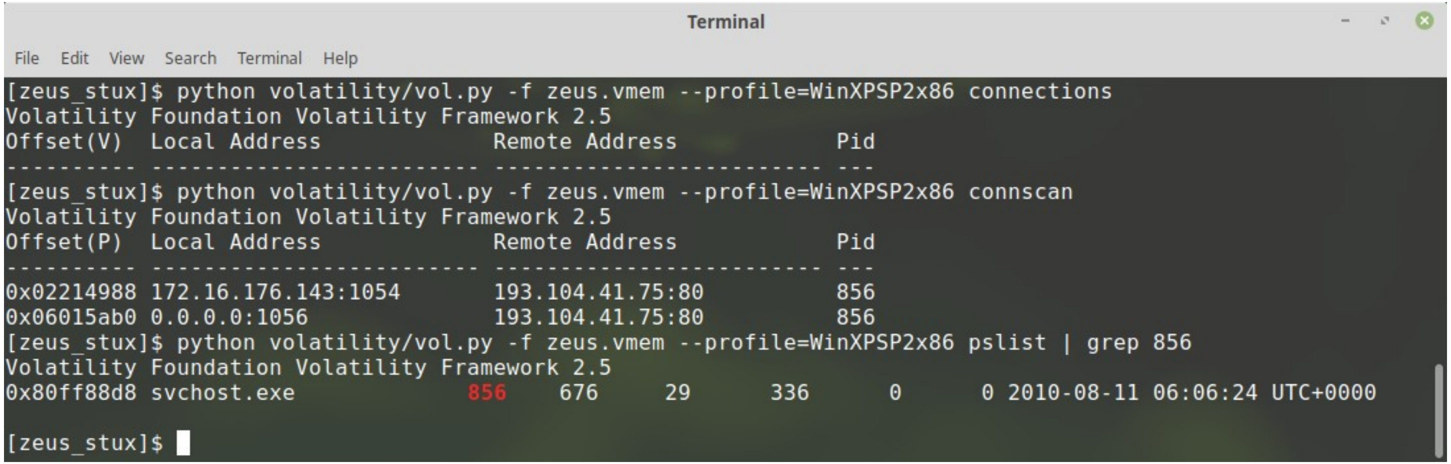
An advanced memory forensics framework http://volatilityfoundation.org/

2,802 commits · 9 branches · 10 releases · 49 contributors · GPL-2.0

atcuno Windows - update mapped file filter for rare smear instance · Latest commit 62aa820 4 hours ago

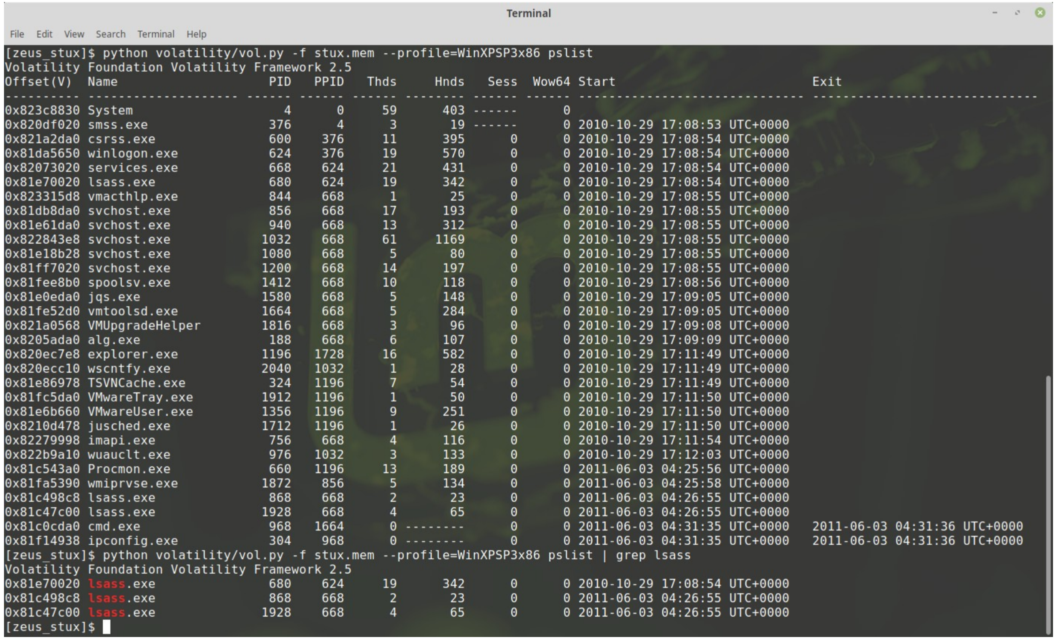
contrib	Move unstable plugins, plugins still under development, and plugins m…	2 years ago
pyinstaller	Fix updated pyinstaller "empty DEST" issue	11 months ago
resources	Move the Volatility-1.4_rc1 branch over to trunk.	8 years ago
tools	Linux - Update check_fops for 4.20	9 months ago
volatility	Windows - update mapped file filter for rare smear instance	4 hours ago

Volatility and Zeus



```
[zeus_stux]$ python volatility/vol.py -f zeus.vmem --profile=WinXPSP2x86 connections
Volatility Foundation Volatility Framework 2.5
Offset(V)  Local Address             Remote Address            Pid
---------- ------------------------- ------------------------- ---
[zeus_stux]$ python volatility/vol.py -f zeus.vmem --profile=WinXPSP2x86 connscan
Volatility Foundation Volatility Framework 2.5
Offset(P)  Local Address             Remote Address            Pid
---------- ------------------------- ------------------------- ---
0x02214988 172.16.176.143:1054       193.104.41.75:80          856
0x06015ab0 0.0.0.0:1056              193.104.41.75:80          856
[zeus_stux]$ python volatility/vol.py -f zeus.vmem --profile=WinXPSP2x86 pslist | grep 856
Volatility Foundation Volatility Framework 2.5
0x80ff88d8 svchost.exe             856    676     29      336      0      0 2010-08-11 06:06:24 UTC+0000

[zeus_stux]$ 
```

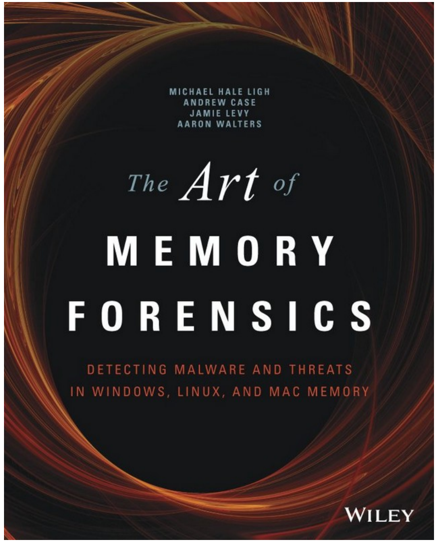
Volatility and Stuxnet

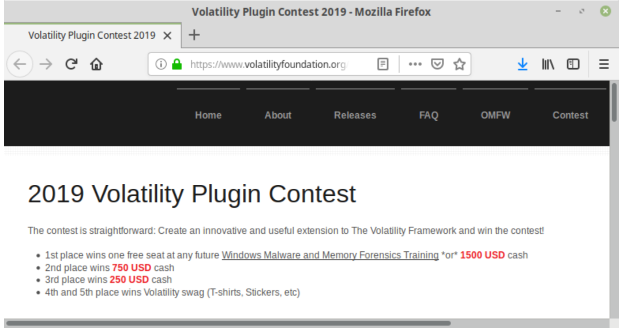


```
[zeus_stux]$ python volatility/vol.py -f stux.mem --profile=WinXPSP3x86 pslist
Volatility Foundation Volatility Framework 2.5
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ------------------------------
0x823c8830 System                    4      0     59      403 ------      0
0x820df020 smss.exe                376      4      3       19 ------      0 2010-10-29 17:08:53 UTC+0000
0x821a2da0 csrss.exe               600    376     11      395      0      0 2010-10-29 17:08:54 UTC+0000
0x81da5650 winlogon.exe            624    376     19      570      0      0 2010-10-29 17:08:54 UTC+0000
0x82073020 services.exe            668    624     21      431      0      0 2010-10-29 17:08:54 UTC+0000
0x81e70020 lsass.exe               680    624     19      342      0      0 2010-10-29 17:08:54 UTC+0000
0x823315d8 vmacthlp.exe            844    668      1       25      0      0 2010-10-29 17:08:55 UTC+0000
0x81db8da0 svchost.exe             856    668     17      193      0      0 2010-10-29 17:08:55 UTC+0000
0x81e61da0 svchost.exe             940    668     13      312      0      0 2010-10-29 17:08:55 UTC+0000
0x822843e8 svchost.exe            1032    668     61     1169      0      0 2010-10-29 17:08:55 UTC+0000
0x81e18b28 svchost.exe            1080    668      5       80      0      0 2010-10-29 17:08:55 UTC+0000
0x81ff7020 svchost.exe            1200    668     14      197      0      0 2010-10-29 17:08:55 UTC+0000
0x81fee8b0 spoolsv.exe            1412    668     10      118      0      0 2010-10-29 17:08:56 UTC+0000
0x81e0eda0 jqs.exe                1580    668      5      148      0      0 2010-10-29 17:09:05 UTC+0000
0x81fe52d0 vmtoolsd.exe           1664    668      5      284      0      0 2010-10-29 17:09:05 UTC+0000
0x821a0568 VMUpgradeHelper        1816    668      3       96      0      0 2010-10-29 17:09:08 UTC+0000
0x8205ada0 alg.exe                 188    668      6      107      0      0 2010-10-29 17:09:09 UTC+0000
0x820ec7e8 explorer.exe           1196   1728     16      582      0      0 2010-10-29 17:11:49 UTC+0000
0x820ecc10 wscntfy.exe            2040   1032      1       28      0      0 2010-10-29 17:11:49 UTC+0000
0x81e86978 TSVNCache.exe           324   1196      7       54      0      0 2010-10-29 17:11:49 UTC+0000
0x81fc5da0 VMwareTray.exe         1912   1196      1       50      0      0 2010-10-29 17:11:50 UTC+0000
0x81e6b660 VMwareUser.exe         1356   1196      9      251      0      0 2010-10-29 17:11:50 UTC+0000
0x8210d478 jusched.exe            1712   1196      1       26      0      0 2010-10-29 17:11:50 UTC+0000
0x82279998 imapi.exe               756    668      4      116      0      0 2010-10-29 17:11:54 UTC+0000
0x822b9a10 wuauclt.exe             976   1032      3      133      0      0 2010-10-29 17:12:03 UTC+0000
0x81c543a0 Procmon.exe             660   1196     13      189      0      0 2011-06-03 04:25:56 UTC+0000
0x81fa5390 wmiprvse.exe           1872    856      5      134      0      0 2011-06-03 04:25:58 UTC+0000
0x81c498c8 lsass.exe               868    668      2       23      0      0 2011-06-03 04:26:55 UTC+0000
0x81c47c00 lsass.exe              1928    668      4       65      0      0 2011-06-03 04:26:55 UTC+0000
0x81c0cda0 cmd.exe                 968   1664      0 --------      0      0 2011-06-03 04:31:35 UTC+0000   2011-06-03 04:31:36 UTC+0000
0x81f14938 ipconfig.exe            304    968      0 --------      0      0 2011-06-03 04:31:35 UTC+0000   2011-06-03 04:31:36 UTC+0000
[zeus_stux]$ python volatility/vol.py -f stux.mem --profile=WinXPSP3x86 pslist | grep lsass
Volatility Foundation Volatility Framework 2.5
0x81e70020 lsass.exe               680    624     19      342      0      0 2010-10-29 17:08:54 UTC+0000
0x81c498c8 lsass.exe               868    668      2       23      0      0 2011-06-03 04:26:55 UTC+0000
0x81c47c00 lsass.exe              1928    668      4       65      0      0 2011-06-03 04:26:55 UTC+0000
[zeus_stux]$
```

Don’t pull the plug

Further reading

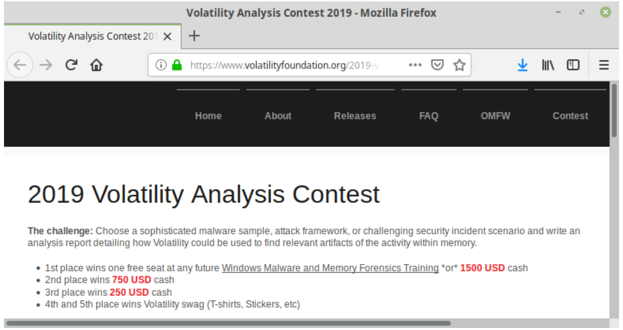



Volatility Plugin Contest 2019 - Mozilla Firefox
Volatility Plugin Contest 2019
https://www.volatilityfoundation.org
Home
About
Releases
FAQ
OMFW
Contest

2019 Volatility Plugin Contest

The contest is straightforward: Create an innovative and useful extension to The Volatility Framework and win the contest!

- 1st place wins one free seat at any future Windows Malware and Memory Forensics Training *or* **1500 USD** cash
- 2nd place wins **750 USD** cash
- 3rd place wins **250 USD** cash
- 4th and 5th place wins Volatility swag (T-shirts, Stickers, etc)


Volatility Analysis Contest 2019 - Mozilla Firefox
Volatility Analysis Contest 20
https://www.volatilityfoundation.org/2019-
Home
About
Releases
FAQ
OMFW
Contest

2019 Volatility Analysis Contest

The challenge: Choose a sophisticated malware sample, attack framework, or challenging security incident scenario and write an analysis report detailing how Volatility could be used to find relevant artifacts of the activity within memory.

- 1st place wins one free seat at any future Windows Malware and Memory Forensics Training *or* **1500 USD** cash
- 2nd place wins **750 USD** cash
- 3rd place wins **250 USD** cash
- 4th and 5th place wins Volatility swag (T-shirts, Stickers, etc)

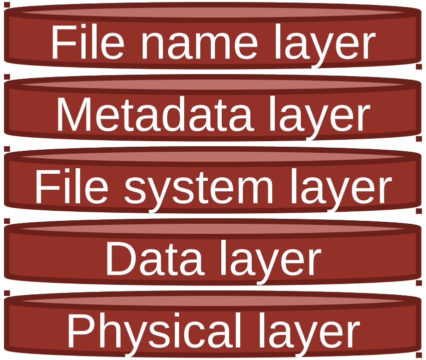
Disk (or, file system) forensics

Situation

- Evil file has reached disk
- Persistence is achieved

- Our job: Find the malware

A closer look at files



File name layer

Metadata layer

File system layer

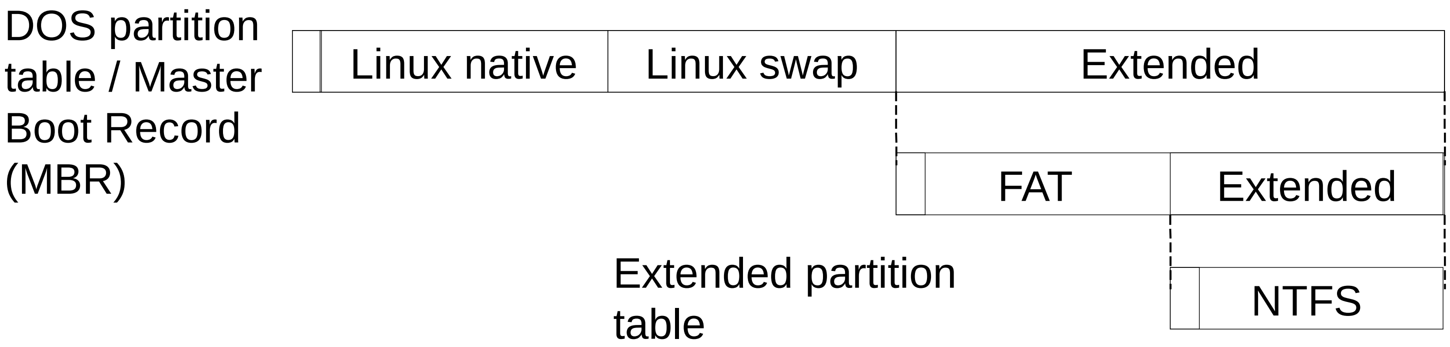
Data layer

Physical layer

- File names, directories
- Structure information about files/directories
- Partition information
- Sectors, blocks, clusters
- The drive itself, and partitions

Physical layer

- DOS-based partitions, primary partition table, extended partitions



DOS partition table / Master Boot Record (MBR)

Linux native

Linux swap

Extended

FAT

Extended

NTFS

Extended partition table

MBR and EBR

- Primary = Master Boot Record (MBR)
- Extended = Extended Boot Record (EBR)
- Same layout, 512 bytes or 1 sector

Bytes	Content
0-445	Upstart code, disk signature
446-461	Partition entry 1
462-477	Partition entry 2
478-493	Partition entry 3
494-509	Partition entry 4
510-511	MBR/EBR signature (0xAA55)

Partitions

Bytes	Content
0	0x00 not boot, 0x80 boot
1-3	Cylinder-head-sector (CHS) of start sector
4	Partition type
5-7	Cylinder-head-sector (CHS) of end sector
8-11	Logical block addressing (LBA) of start sector
12-15	Number of sectors in partition

Type	FAT12	FAT16	FAT32	Linux native	Linux swap	Exten-ded	NFTS
Hex value	0x01	0x0E	0x0C	0x83	0x82	0x05	0x07

Data layer

- 512-byte sectors
- 1 or more sectors = clusters (Windows) or blocks (Unix)

- Blocks either **allocated**
  - Actively being used by a file
- Or **unallocated**
  - Not being used by a file
  - May contain deleted or unused data

Deleted != destroyed

- When a file is deleted, data exists on disk until overwritten
- If overwritten, remnants may still exist in
  - page/swap/hibernation file, or
  - elsewhere on the disk due to (de)fragmentation
  - extra copies
- If disk wiped, only just once, recovery infeasible

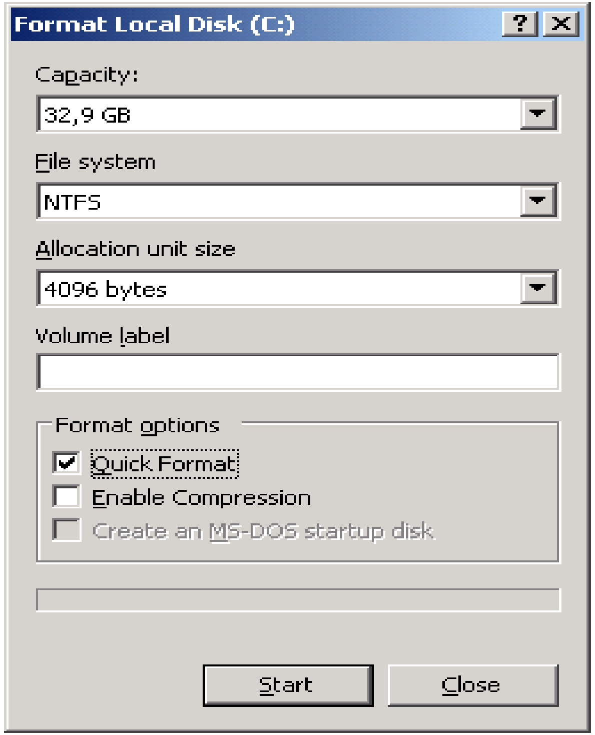
Think libraries

For NTFS

- An entry in the Master File Table describes a file
- Each entry contains the filename and metadata like permissions, timestamps
- Entries are 1024 bytes
- For files > 1024, socalled non-resident files, entry contains an allocation map of clusters allocated to the file

Format is not wiping

- Formats create and replace file system structures
- Files are not overwritten
- Regular formats take more time as the disk is scanned for bad sectors
- Use wiping software for wiping



Format Local Disk (C:)

Capacity:
32,9 GB

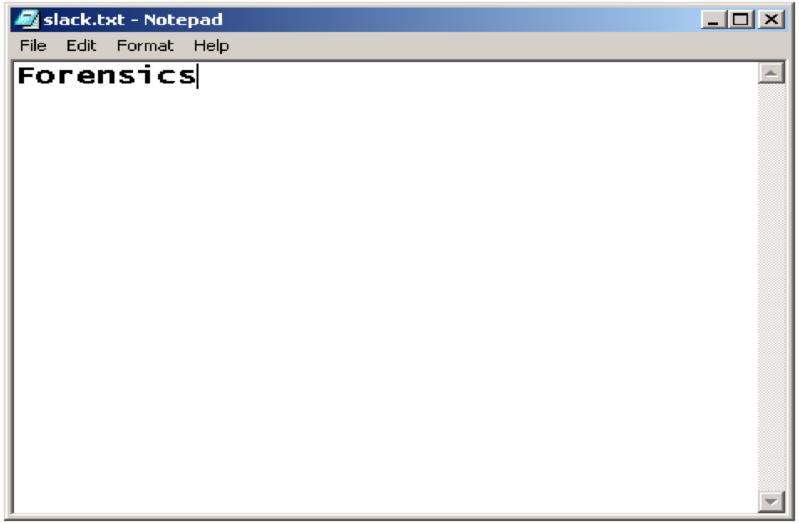
File system
NTFS

Allocation unit size
4096 bytes

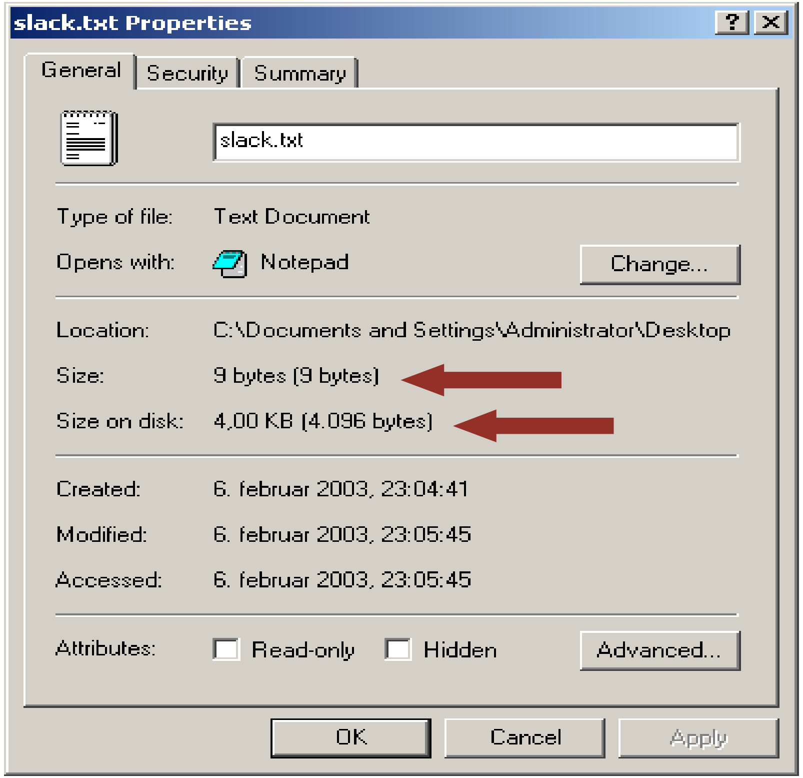
Volume label

Format options
☑ Quick Format
☐ Enable Compression
☐ Create an MS-DOS startup disk

Slack space



```
Forensics
```



slack.txt

Type of file: Text Document

Opens with: Notepad

Location: C:\Documents and Settings\Administrator\Desktop

Size: 9 bytes (9 bytes)

Size on disk: 4,00 KB (4.096 bytes)

Created: 6. februar 2003, 23:04:41

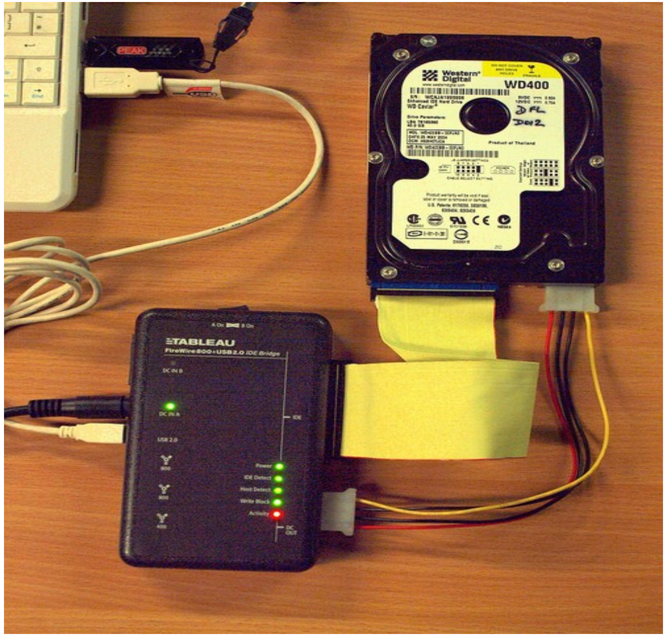
Modified: 6. februar 2003, 23:05:45

Accessed: 6. februar 2003, 23:05:45

Attributes: □ Read-only □ Hidden

Create bit-by-bit copy

Forensic workstation

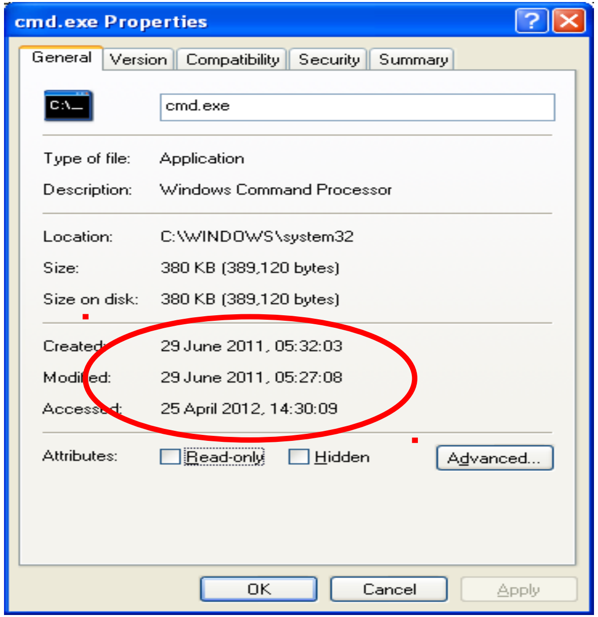


Seized harddrive

Write blocker

Timelines

- MAC = Modified+Accessed+Changed



cmd.exe Properties

cmd.exe

Type of file: Application

Description: Windows Command Processor

Location: C:\WINDOWS\system32

Size: 380 KB (389,120 bytes)

Size on disk: 380 KB (389,120 bytes)

Created: 29 June 2011, 05:32:03

Modified: 29 June 2011, 05:27:08

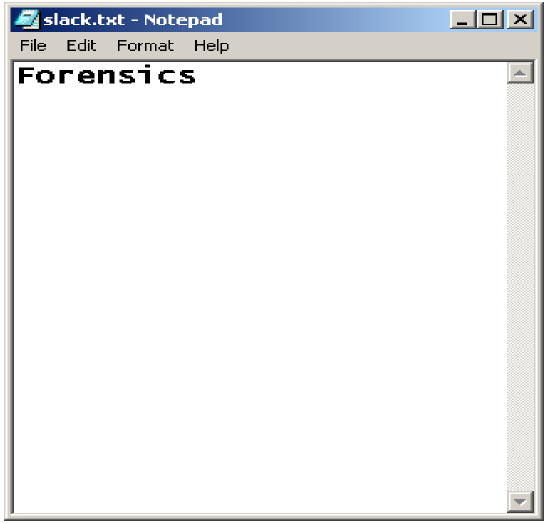
Accessed: 25 April 2012, 14:30:09

Attributes: □ Read-only □ Hidden

File types

- Certain file types may be of interest

Slack.txt



Forensics

Slack.exe

Slack.pdf

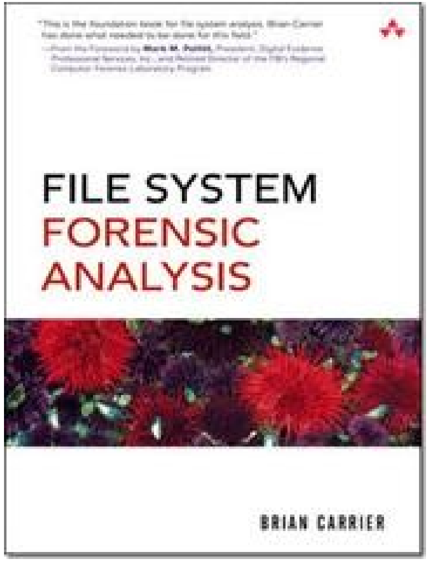
Slack.zip

Slack.dat

Slack.mp3

Slack.dll

Further reading



"This is the foundation book for file system analysis. Brian Carrier has done what needed to be done for this field."

—From the Foreword by **Mark M. Pollitt**, President, Digital Evidence Professional Services, Inc., and Retired Director of the FBI's Regional Computer Forensic Laboratory Program

FILE SYSTEM
FORENSIC
ANALYSIS

BRIAN CARRIER

Log analysis

Wrap-up

Lecture plan

36	31 Aug	10-12	TL	Introduction, security concepts and the threat of hacking
	04 Sep	10-12	TL	Buffer overflow
37	07 Sep	10-12	CJ	Software security, Operating system security
	11 Sep	10-12	CJ	User authentication and access control
38	14 Sep	10-12	TL	Malicious software
	18 Sep	10-12	CJ	Firewalls and denial-of-service attacks
39	21 Sep	10-12	CJ	Cloud and IoT
	25 Sep	10-12	TL	Cryptography
40	28 Sep	10-12	TL	Internet security protocols
	02 Oct	10-12	TL	Intrusion detection
41	05 Oct	10-12	TL	Forensics
	09 Oct	10-12	CJ	IT security management
42				Fall Vacation - No lectures
43	19 Oct	10-12	CJ	Privacy 1
	23 Oct	10-12	CJ	Privacy 2
44	26 Oct	10-11	Guest	Final guest lecture
		11-12	All	Recap and Q/A
45	xx Nov			Exam