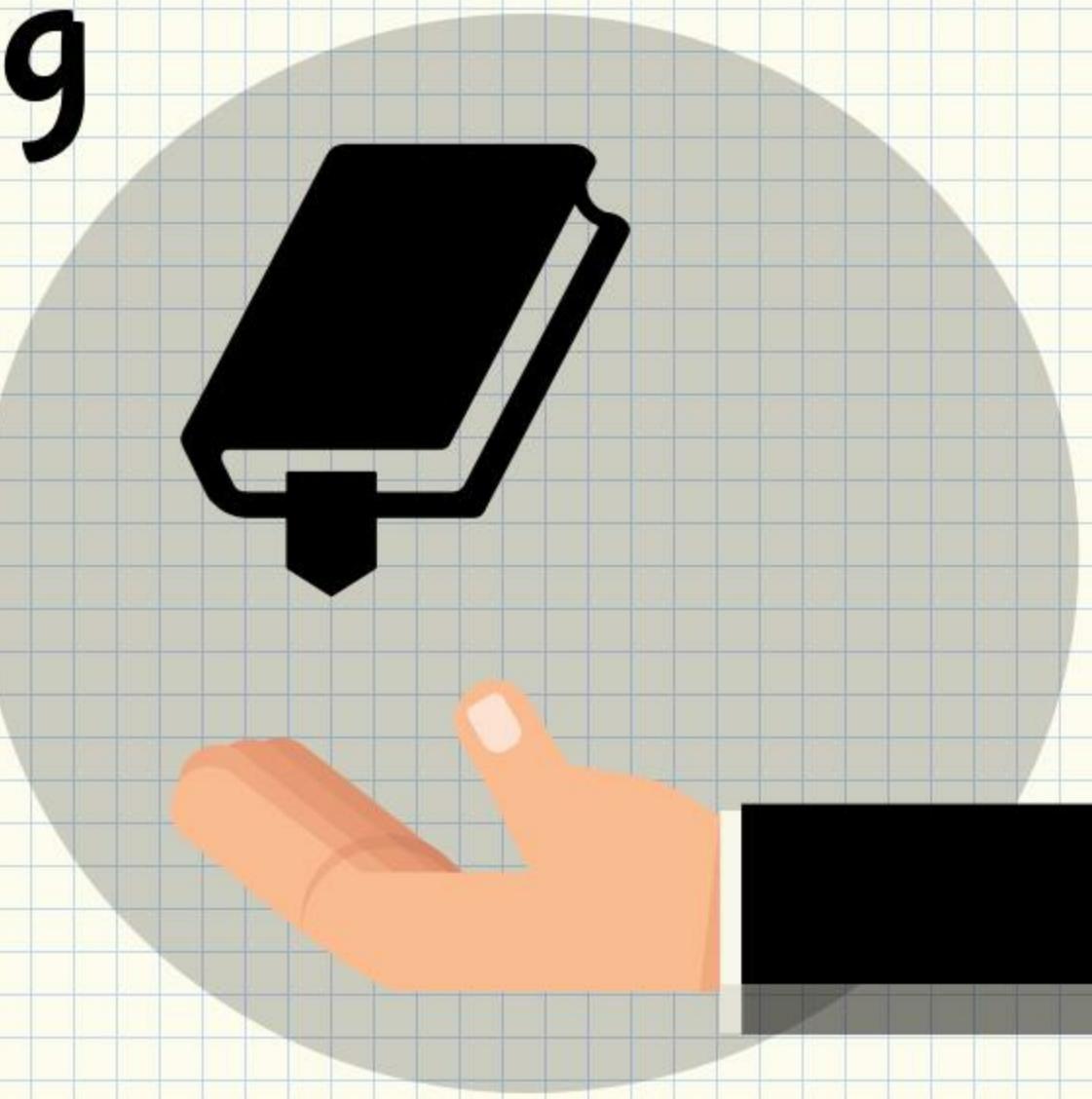


Limon – Sandbox for Analyzing Linux Malwares

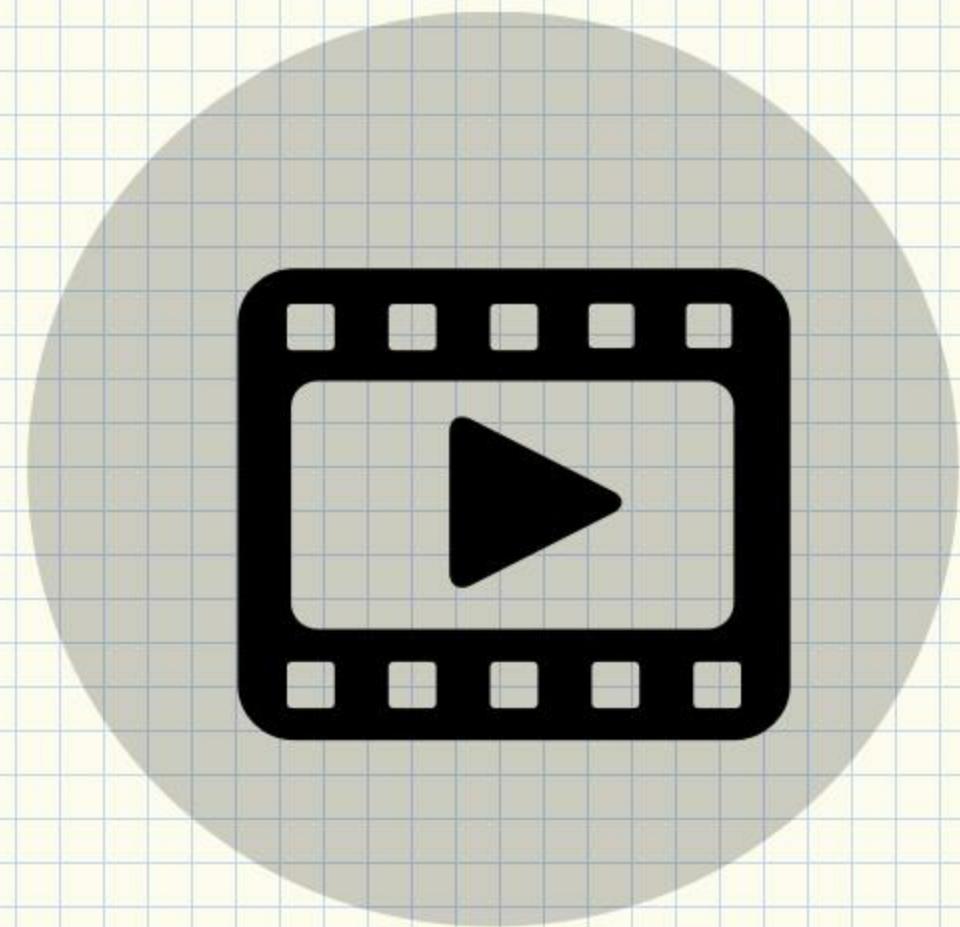
by Monnappa K A



Who Am I

Monnappa K A

- ⚙️ Info Security Investigator – Cisco CSIRT
- ⚙️ Member of SecurityXploded
- ⚙️ Reverse Engineering, Malware Analysis, Memory Forensics
- ⚙️ Author of Limon Sandbox
- ⚙️ Conferences – Black Hat, FIRST, 4SICS etc
- ⚙️ Articles – EForensics, Hakin9, Hack Insight
- ⚙️ Email: monnappa22@gmail.com



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

Types of Analysis

**Static
Analysis**

Analysis without
Executing the
malware

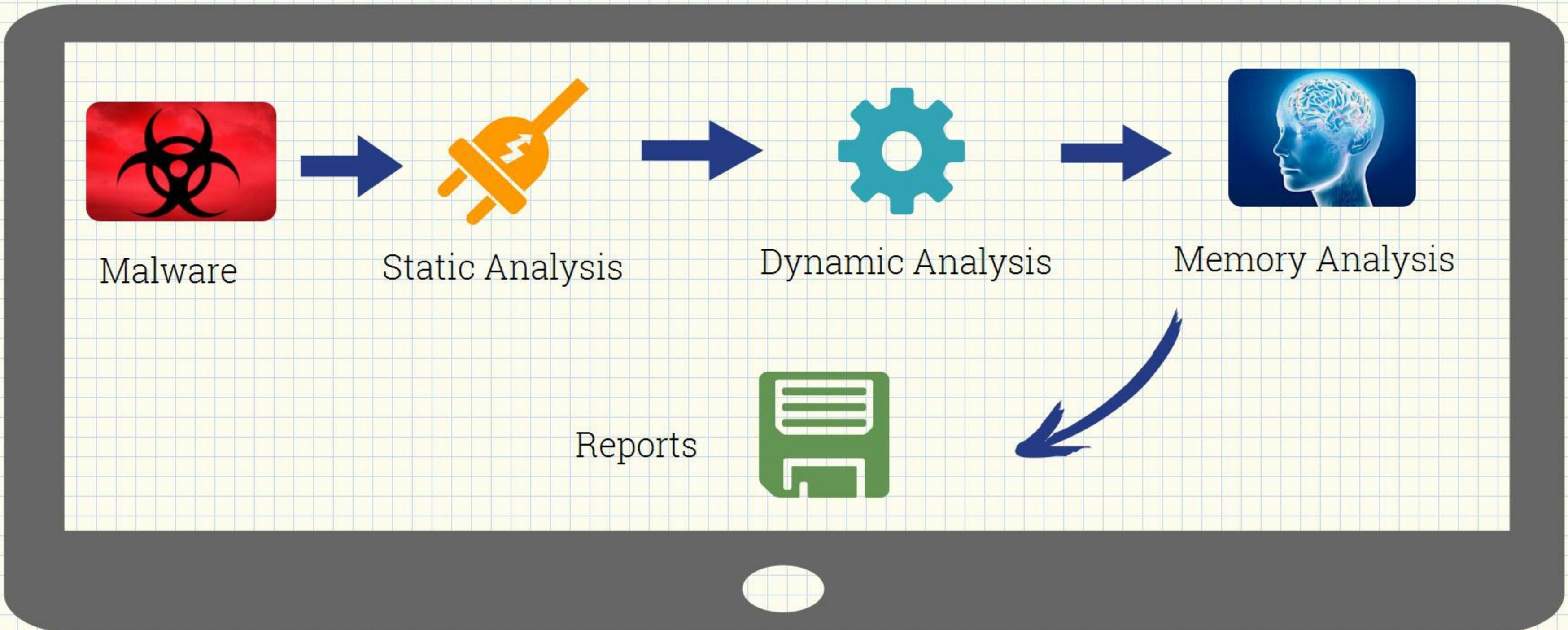
**Dynamic
Analysis**

Analysis by
executing the
malware

**Memory
Analysis**

Analysis of RAM (main
memory) after executing
the malware

Working of Limon

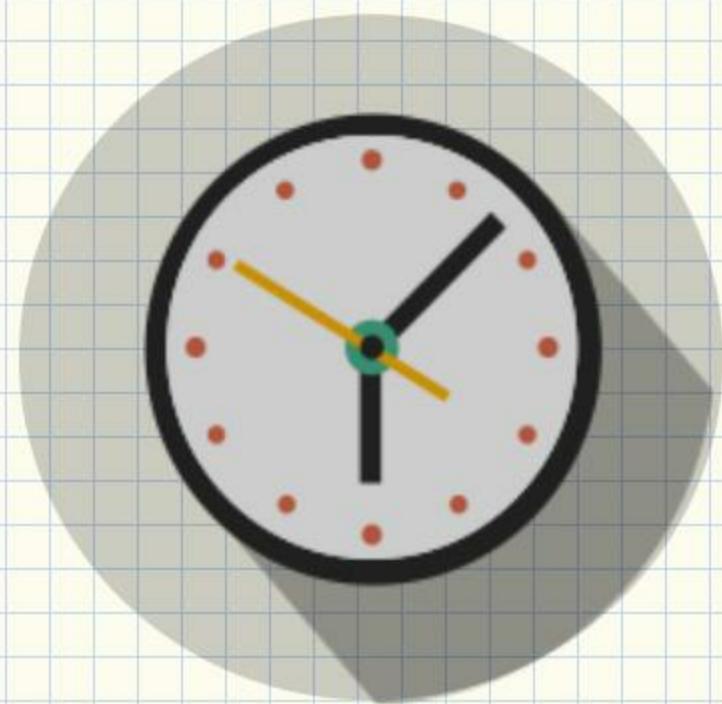


General Features

- ◆ 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

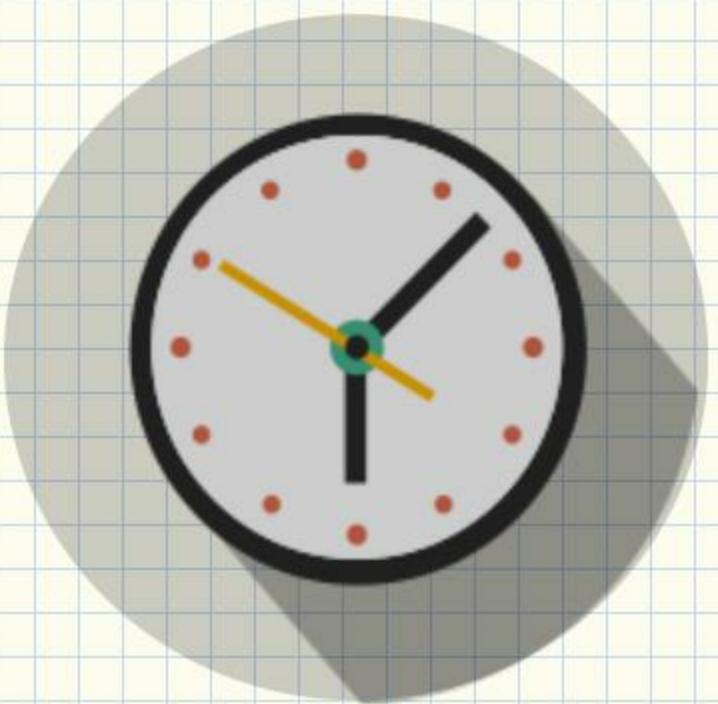
Static Analysis Features

- ⚙ 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

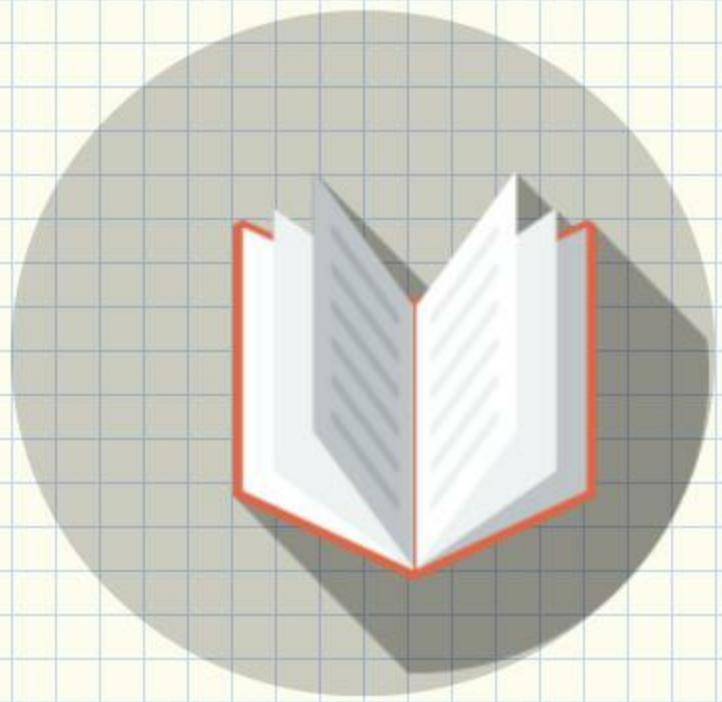
- ⚙️ Determines malware capability using YARA rules (ability to run custom YARA rules will be added soon)
- ⚙️ Performs md5 search on VirusTotal (does not submit samples)
- ⚙️ Displays dependencies of the malware (shared objects)
- ⚙️ Displays program header structures
- ⚙️ Displays section header information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



Dynamic Analysis Features

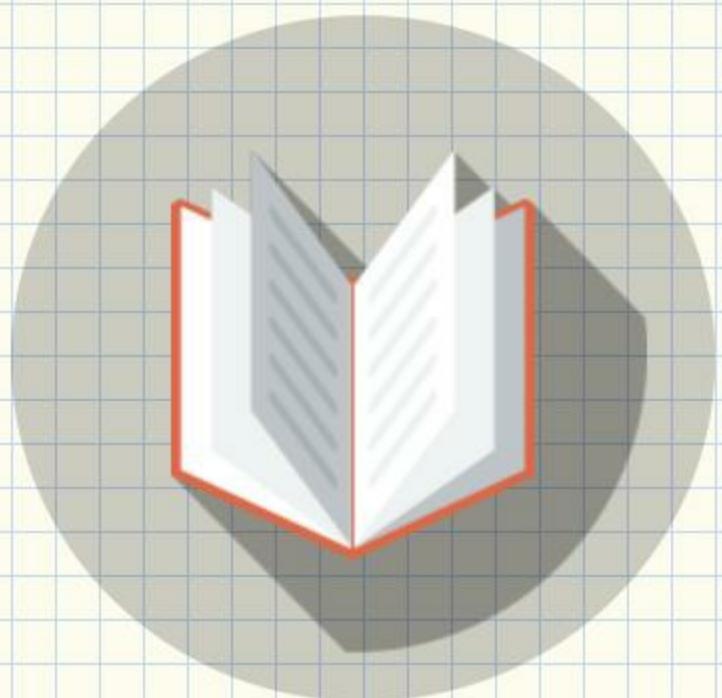
Limon gives different options for performing dynamic analysis to track activity of the malware(during execution), below are the different options

- 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 monitoring 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

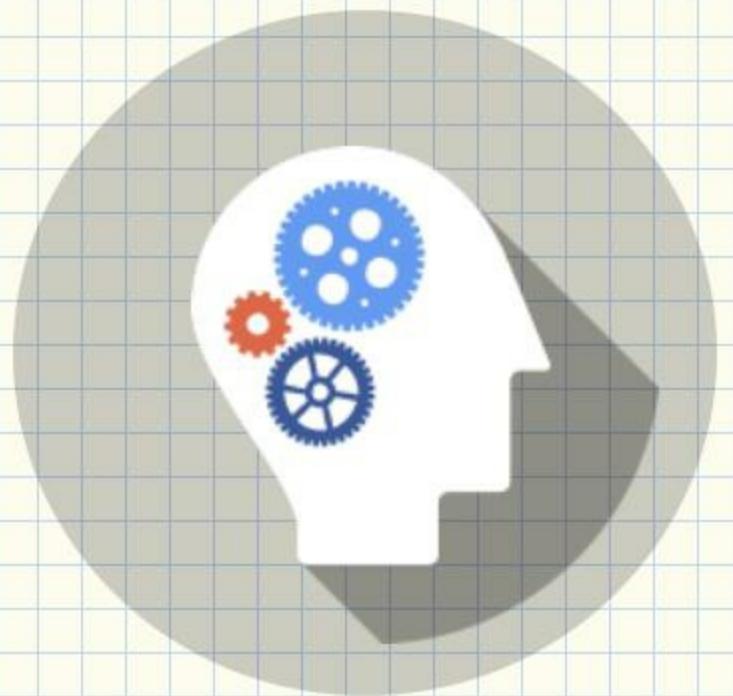
- Shows DNS summary
- Shows TCP conversations
- Stores packet captures
- Stores event trace dump



Memory Analysis Features

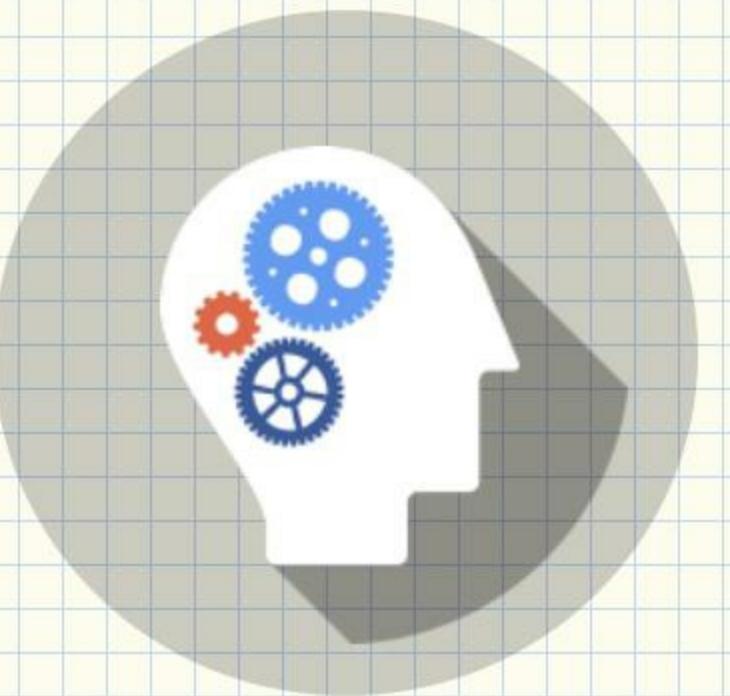
Limon performs post-mortem analysis by performing memory analysis using Volatility framework. This feature should help in detecting stealthy rootkits and malwares performing Anti-Forensic tricks. Below are the memory analysis features:

- 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
- Displays Network connections (TCP and UDP)



Memory Analysis Features

- Displays Interface Information
- Displays processes running with RAW sockets
- Displays shared libraries associated with the processes (using different methods)
- Displays kernel modules
- Displays 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)

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:

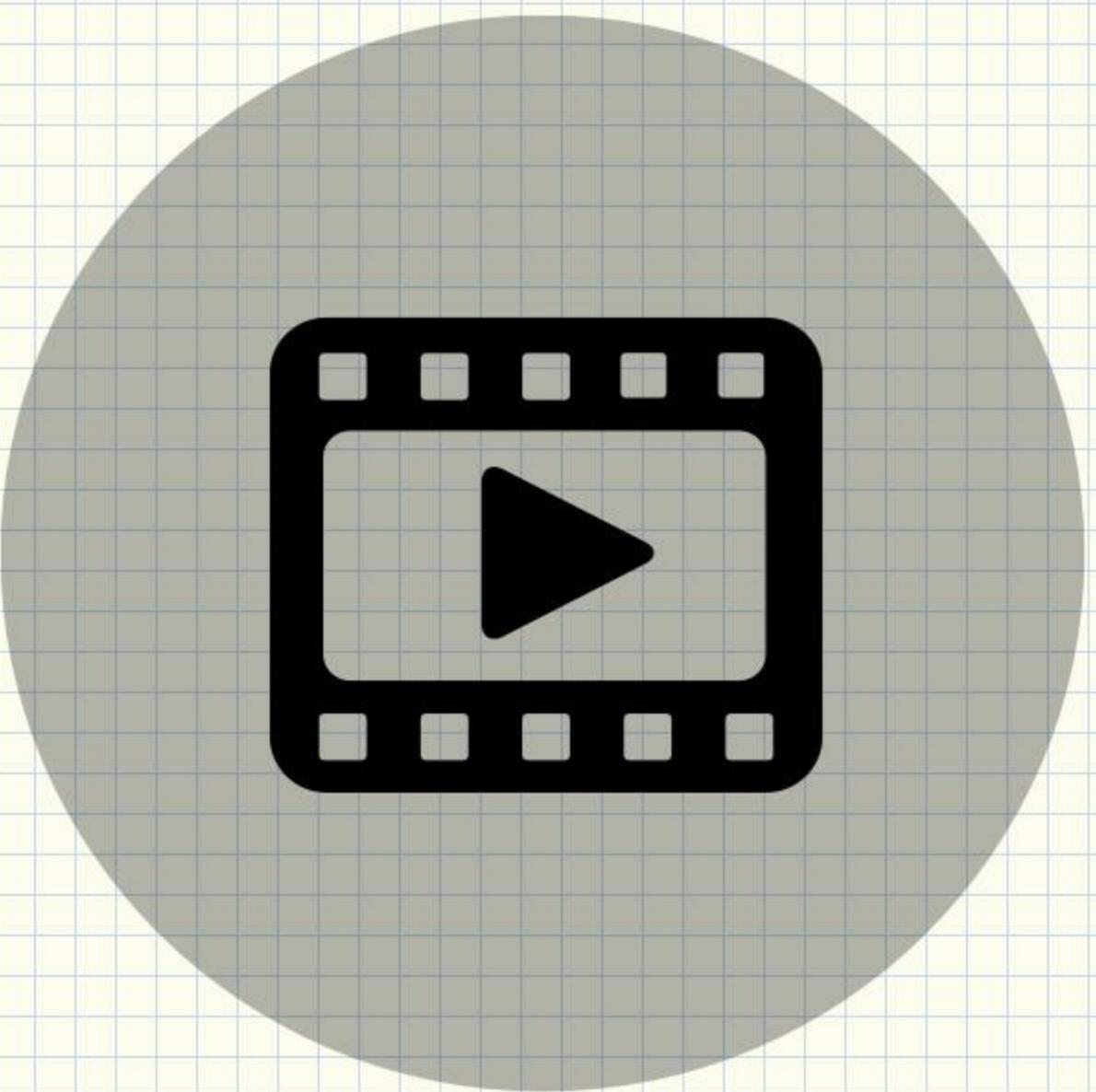
Limon can analyse below file types (both with and without parameters)

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



Demo

Analysis of Tsunami Using Limon



Running Tsunami malware

```
root@helios:~/limon_sandbox# python limon.py -h
Usage: limon.py [Options] <file> [args] ←

Options:
  -h, --help            show this help message and exit
  -t TIMEOUT, --timeout=TIMEOUT
                        timeout in seconds, default is 60 seconds
  -i, --internet        connects to internet
  -p, --perl             perl script (.pl)
  -P, --python           python script (.py)
  -z, --php              php script ←
  -s, --shell            shell script
  -b, --bash             BASH script
  -k, --lkm              load kernel module
  -C, --ufctrace         unfiltered call trace(full trace)
  -e, --femonitor        filtered system event monitoring
  -E, --ufemonitor       unfiltered system event monitoring
  -m, --memfor           memory forensics
  -M, --vmemfor          verbose memory forensics(slow)
  -x, --printhexdump     print hex dump in call trace (both filtered and
                        unfiltered call trace)

root@helios:~/limon_sandbox# python limon.py /root/linux_malwares/tsuna -t 40 -x -m
```

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+C3P0CtZ8ax0h/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 | htonl@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 | tclose@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>  
most firewalls  
NOTICE %s :PAN <target> <port> <secs>  
most network drivers  
NOTICE %s :UDP <target> <port> <secs>  
NOTICE %s :UNKNOWN <target> <secs>  
NOTICE %s :NICK <nick>  
NOTICE %s :SERVER <server>  
NOTICE %s :GETSPOOFS  
NOTICE %s :SPOOFS <subnet>  
NOTICE %s :DISABLE  
NOTICE %s :ENABLE  
NOTICE %s :KILL  
NOTICE %s :GET <http address> <save as>  
it onto the hd  
NOTICE %s :VERSION  
NOTICE %s :KILLALL  
NOTICE %s :HELP  
NOTICE %s :IRC <command>  
NOTICE %s :SH <command>  
NOTICE %s :Killing pid %d.
```

= Special packeter that wont be blocked by
most firewalls
= An advanced syn flooder that will kill
most network drivers
= A udp flooder
= Another non-spoof udp flooder
= Changes the nick of the client
= Changes servers
= Gets the current spoofing
= Changes spoofing to a subnet
= Disables all packeting from this client
= Enables all packeting from this client
= Kills the client
= Downloads a file off the web and saves
it onto the hd
= Requests version of client
= Kills all current packeting
= Displays this
= Sends this command to the server
= Executes a command

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

The screenshot shows the Tsunami Network Communication interface. The main window displays a list of network packets captured over time, filtered by 'tcp.stream eq 0'. The first few packets show the initial TCP handshake between the source (192.168.1.150) and destination (80.243.54.131). Subsequent packets represent an IRC connection, with frames containing commands like 'NICK YXXES' and 'USER OAQL localhost localhost :VKHLC'. A detailed view of the first packet is shown in a modal dialog, revealing its structure as an Ethernet II frame with Src: Tp-LinkT_27:3e:42 (14:cc:20:22:00:00), IP Version 4 (Src: 192.168.1.150), and TCP Port 37002. The raw bytes of the frame are also displayed at the bottom.

No.	Time	Source	Destination	Protocol	Length	Info
1	2015-10-06 14:26:59.821070	192.168.1.150	80.243.54.131	TCP	74	37002->5566 [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=4294
4	2015-10-06 14:26:59.821156	80.243.54.131	192.168.1.150	TCP	74	5566->37002 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1
5	2015-10-06 14:26:59.821232	192.168.1.150	80.243.54.131	TCP	66	37002->5566 [ACK] Seq=1 Ack=1 Win=14600 Len=0 TSval=4294912246 TSecr=16
6	2015-10-06 14:26:59.822661	80.243.54.131	192.168.1.150	TCP	74	[TCP Spurious Retransmission] 5566->37002 [SYN, ACK] Seq=0 Ack=1 Win=28
7	2015-10-06 14:26:59.822670	80.243.54.131	192.168.1.150	TCP	74	[TCP Spurious Retransmission] 5566->37002 [SYN, ACK] Seq=0 Ack=1 Win=28
8	2015-10-06 14:26:59.822740	192.168.1.150	80.243.54.131	TCP	002->5566 [ACK] Seq=1 Ack=1 Win=14600 Len=0 TSval=4	
9	2015-10-06 14:26:59.823803	80.243.54.131	192.168.1.150	TCP	002->5566 [SYN] 5566->37002 [SYN, ACK] Seq=0 Ack=1 Win=28	
10	2015-10-06 14:26:59.823812	80.243.54.131	192.168.1.150	TCP	002->5566 [SYN, ACK] Seq=0 Ack=1 Win=28	
11	2015-10-06 14:26:59.823877	192.168.1.150	80.243.54.131	TCP	002->5566 [ACK] Seq=1 Ack=1 Win=14600 Len=0 TSval=4	
12	2015-10-06 14:26:59.824830	80.243.54.131	192.168.1.150	TCP	002->5566 [SYN] 5566->37002 [SYN, ACK] Seq=0 Ack=1 Win=28	
13	2015-10-06 14:26:59.824837	80.243.54.131	192.168.1.150	TCP	002->5566 [SYN, ACK] Seq=0 Ack=1 Win=28	
14	2015-10-06 14:26:59.824883	192.168.1.150	80.243.54.131	TCP	002->5566 [ACK] Seq=1 Ack=1 Win=14600 Len=0 TSval=4	
15	2015-10-06 14:26:59.825872	80.243.54.131	192.168.1.150	TCP	002->5566 [SYN] 5566->37002 [SYN, ACK] Seq=0 Ack=1 Win=28	
16	2015-10-06 14:26:59.825875	80.243.54.131	192.168.1.150	TCP	002->5566 [SYN, ACK] Seq=0 Ack=1 Win=28	
17	2015-10-06 14:26:59.825909	192.168.1.150	80.243.54.131	TCP	002->5566 [ACK] Seq=1 Ack=1 Win=14600 Len=0 TSval=4	
18	2015-10-06 14:26:59.826957	80.243.54.131	192.168.1.150	TCP	002->5566 [SYN] 5566->37002 [SYN, ACK] Seq=0 Ack=1 Win=28	
19	2015-10-06 14:26:59.826960	80.243.54.131	192.168.1.150	TCP	002->5566 [SYN, ACK] Seq=0 Ack=1 Win=28	
20	2015-10-06 14:26:59.826993	192.168.1.150	80.243.54.131	TCP	002->5566 [ACK] Seq=1 Ack=1 Win=14600 Len=0 TSval=4	
21	2015-10-06 14:26:59.827996	80.243.54.131	192.168.1.150	TCP	002->5566 [SYN] 5566->37002 [SYN, ACK] Seq=0 Ack=1 Win=28	
22	2015-10-06 14:26:59.828000	80.243.54.131	192.168.1.150	TCP	002->5566 [SYN, ACK] Seq=0 Ack=1 Win=28	
23	2015-10-06 14:26:59.828030	192.168.1.150	80.243.54.131	TCP	002->5566 [ACK] Seq=1 Ack=1 Win=14600 Len=0 TSval=4	
24	2015-10-06 14:26:59.829043	80.243.54.131	192.168.1.150	TCP	002->5566 [SYN] 5566->37002 [SYN, ACK] Seq=0 Ack=1 Win=28	
25	2015-10-06 14:26:59.829049	80.243.54.131	192.168.1.150	TCP	002->5566 [SYN, ACK] Seq=0 Ack=1 Win=28	
26	2015-10-06 14:26:59.829114	192.168.1.150	80.243.54.131	TCP	002->5566 [ACK] Seq=1 Ack=1 Win=14600 Len=0 TSval=4	

Frame 1: 74 bytes on wire (592 bits), 74 bytes
Ethernet II, Src: Tp-LinkT_27:3e:42 (14:cc:20:22:00:00), Dst: 00:0c:29:14:cc:00 (00:0c:29:14:cc:00)
Internet Protocol Version 4, Src: 192.168.1.150, Dst: 80.243.54.131
Transmission Control Protocol, Src Port: 37002, Dst Port: 5566

Stream Content:

```
NICK YXXES
USER OAQL localhost localhost :VKHLC
```

Entire conversation (48 bytes)

Find Save As Print ASCII EBCDIC Hex Dump C Arrays Raw Help Filter Out This Stream Close

0000 14 cc 20 27 3e 42 14 cc 20 27 3e 42 08 00 45 00 .. '>B.. '>B..E.
0010 00 3c 44 48 40 00 40 06 ac bf c0 a8 01 96 50 f3 .<DH@. @.P.
0020 36 83 90 8a 15 be 87 6f 6c be 00 00 00 00 a0 02 6.....o l.....
0030 39 08 01 db 00 00 02 04 05 b4 04 02 08 0a ff ff 9.....

Tsunami – Memory Process Listing

shows malicious process "tsuna" running on the system

DATE/TIME	PROCESS NAME	PPID	PPID2	MEMORY ADDRESS
2015-10-06 08:56:06 UTC+0000	0xfffff88001c332de0 goa-daemon	2603	0	0x00000000006cb8000
2015-10-06 08:56:06 UTC+0000	0xfffff88001a250000 gnome-screensav	2608	0	0x0000000009126000
2015-10-06 08:56:11 UTC+0000	0xfffff88001a35dbc0 aptd	2662	0	0x0000000000c8cf000
2015-10-06 08:56:52 UTC+0000	0xfffff8800020b8000 vmtoolsd	2671	0	0x0000000001a5c7000
2015-10-06 08:56:59 UTC+0000	0xfffff88001a35ade0 strace	2672	0	0x0000000001a3d8000
2015-10-06 08:56:59 UTC+0000	0xfffff88001b992de0 tsuna	2674	0	0x000000000005aa000
2015-10-06 08:56:59 UTC+0000	0xfffff88001a555bc0 dnsmasq	2691	65534	30 0x000000001a1a8000
2015-10-06 08:57:44 UTC+0000	0xfffff88001efedbc0 dbus-daemon	2698	102	105 0x000000001cc07000
2015-10-06 08:57:44 UTC+0000	0xfffff88001a5544d0 dbus-daemon-lau	2700	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
```

Getting Started with Limon?

Download Link

🔗 <https://github.com/monnappa22/Limon>

Setting Up Limon Sandbox

🔗 <http://malware-unplugged.blogspot.in/2015/11/setting-up-limon-sandbox-for-analyzing.html>

Limon Sandbox for analyzing Linux Malwares

🔗 <http://malware-unplugged.blogspot.in/2015/11/limon-sandbox-for-analyzing-linux.html>

Black Hat Europe 2015 presentation

🔗 <https://youtu.be/fSCKyF--tRs>

THANK YOU



@monnappa22



<http://malware-unplugged.blogspot.in>



monnappa22@gmail.com

