Data Analytics Layer for High-Interaction Honeypots

Iqra Khan

Agenda

- Motivation
- Virtualization & cloud security
- VMI
- Honeypots
- Malware analysis
- Methodology
- STIX

Motivation

- Cloud computing – today’s most exciting & important technology
- Relocation of systems and services into cloud environments is on the rise
- Users loose direct access / control over their systems
- Memory investigations and forensic processes for attacks/malwares are limited in cloud

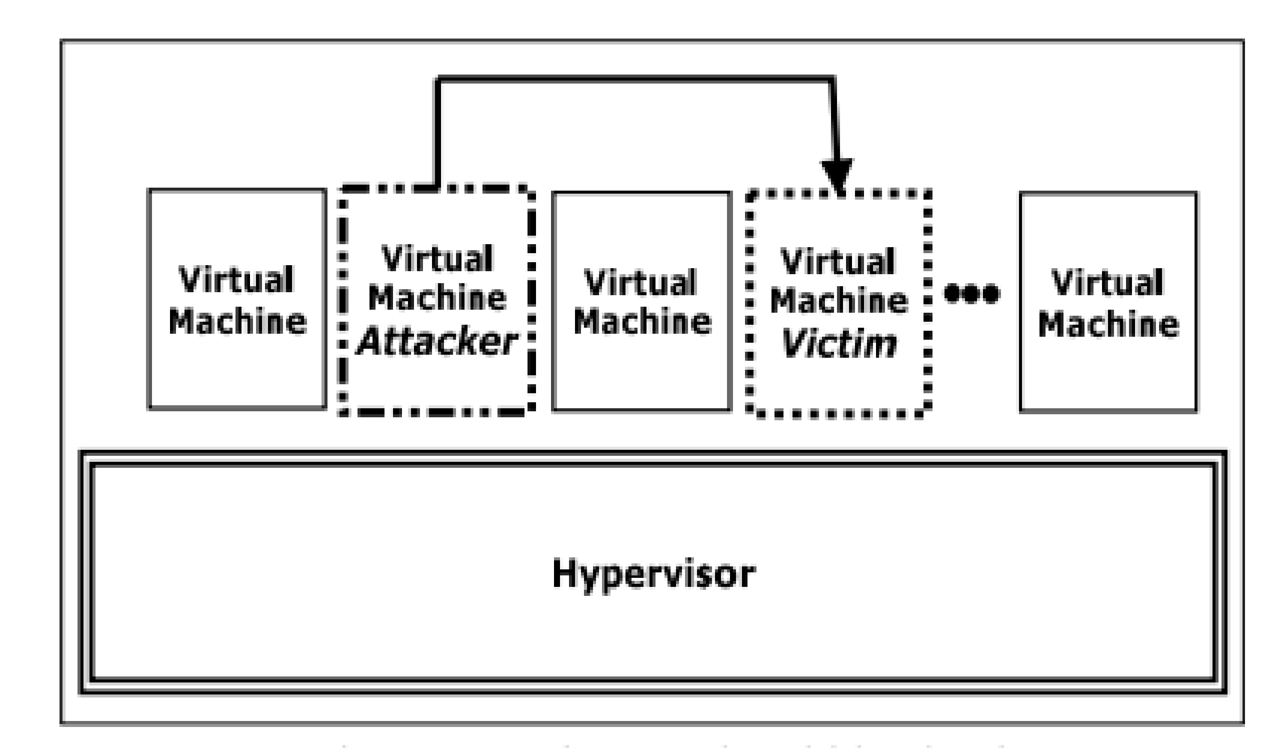
Leveraging virtualization in cloud computing

- Deployment of Clouds are all about pooling resources to increase efficiency
- Reduces cost
- Server virtualization, storage virtualization etc.
- High availability
- **Virtualization is an excellent foundation for building clouds**

Cloud Security

- Security of VMs is a hot topic due to their outsourcing in cloud computing
- Large number of VMs
- Network of VMs
- As Scope of virtualization in cloud computing is increased so does the sophistication of attacks on it

Attack Scenario in Cloud



Traditional approaches for VMs security

- In-guest antiviruses or Host based IDSs
  - Provides no isolation
- Network Intrusion detection systems
  - Limited or no context
- Scan VM disk and memory
  - No interposition

Cloud Security

- Move protection out from the VM
  - Hypervisor based isolation
- Full view of the VM state
  - Interpret virtual hardware to see processes, users, connections, files..
- Actively monitor & control
  - Interposition

Virtual Machine Introspection (VMI)

Virtual Machine Introspection (VMI) is the act of observing the state of VMs from an external entity that can be either another VM or VMM\hypervisor.

VMI (cont.)

- VMI leverages virtualization in three ways:
  - **Isolation**
    - prevents a guest code from reading and writing outside of a VM.
  - **Inspection**
    - VMM can examine the entire state of the guest system (memory, devices, etc.).
  - **Interposition**
    - VMM can interrupt guest code at any time

Why VMI ?

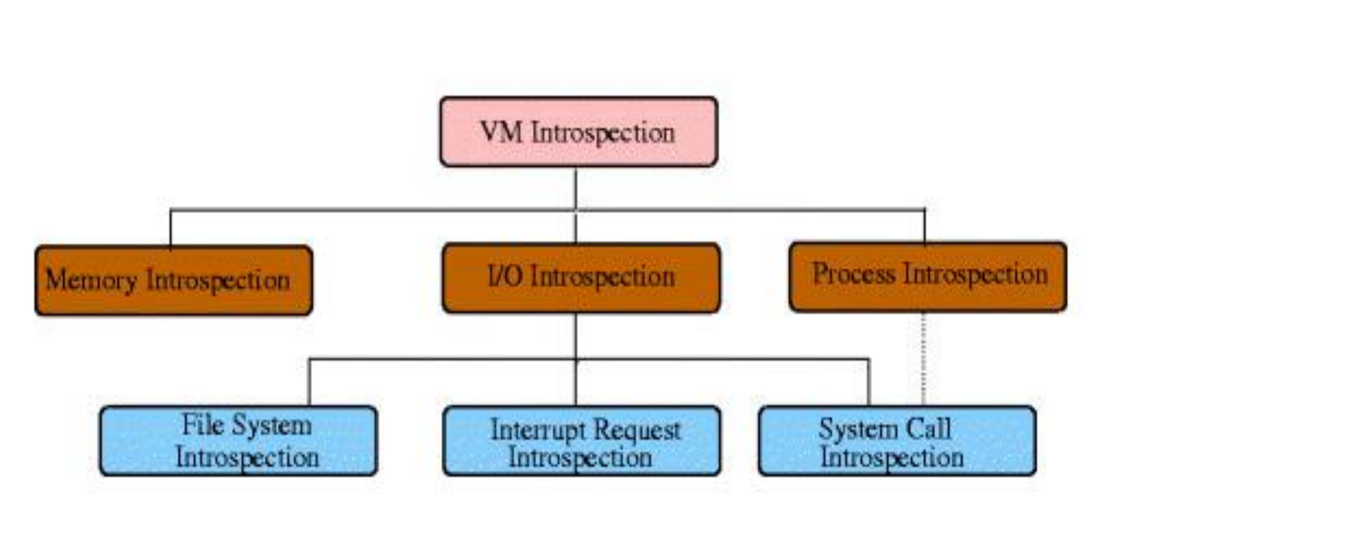
- VMI introspection offers greater credibility of malware analysis than traditional antivirus software running on VMs
- VMI technique inspects and monitor state of VM in an isolated environment separate from VM
- This isolation separates the VMI software from tampering by any application or malware inside the monitored VM.

VMI Advantages

- No altering of the target system
- Very hard to detect the monitoring
- Live analysis of memory content
- Detection of advanced memory resident malware
- More reliable data
  - No data corruption through malware

VMI deployment levels

- Process Introspection
- I/O Introspection
- Memory Introspection


VM Introspection
Memory Introspection
I/O Introspection
Process Introspection
File System Introspection
Interrupt Request Introspection
System Call Introspection

Classification of VMI techniques [2]

Memory Introspection

- Memory introspection deals with live analysis of VM memory.
- Memory contains information like:
  - Running processes
  - Kernel Data Structures
  - Page Tables
  - Registry Entries

Memory Introspection (cont.)

- Majority of malware analysis tools inspect the program behaviour by examining main memory contents of the given program
- These contents helps in intrusion detection or process analysis of the guest VM

How can memory of a VM be accessed from outside?

- LibVMI

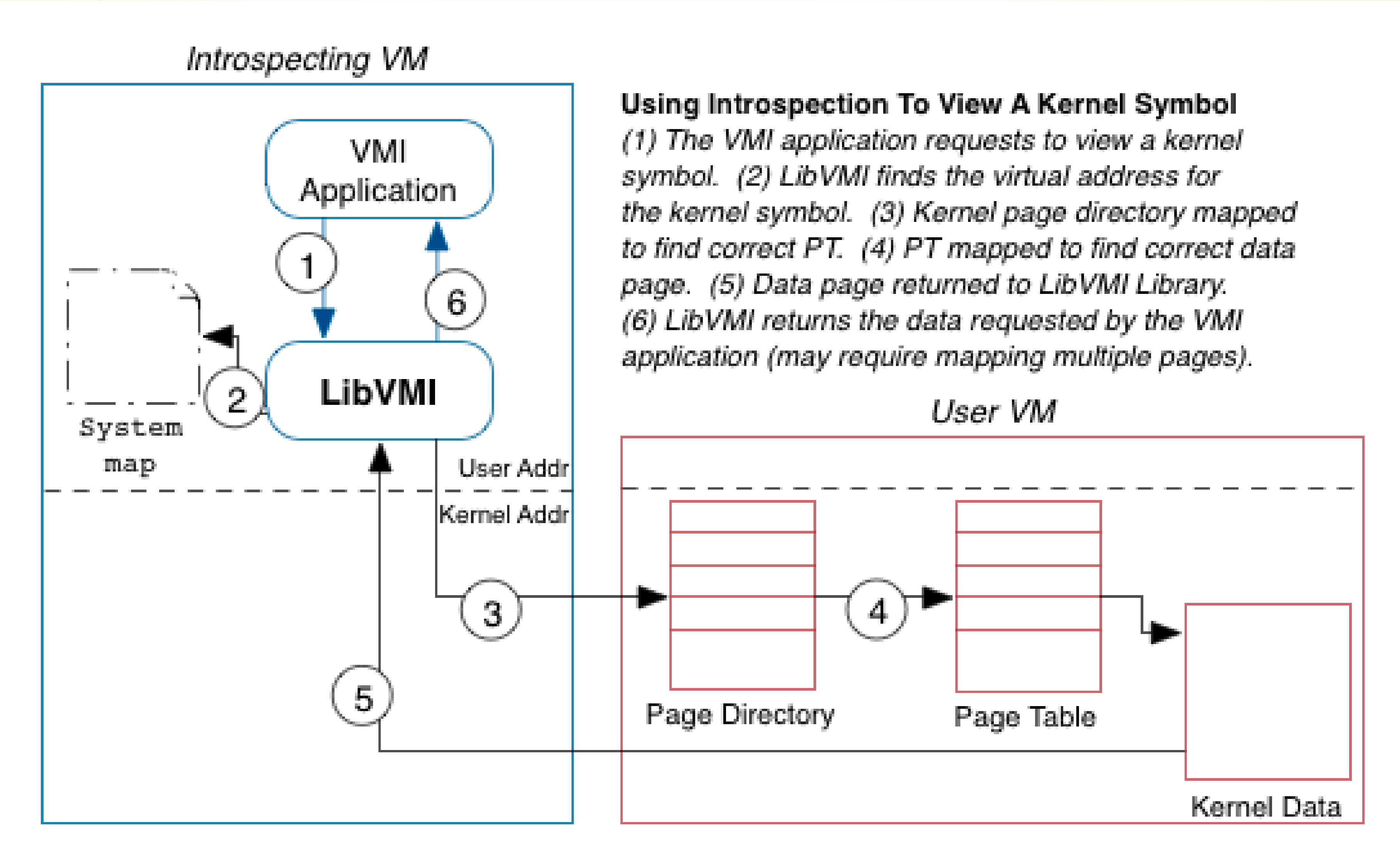
LibVMI

- LibVMI is an open source library for VMI. It is based on XenAccess library used for VMI.
- XenAccess provides a useful application programming interface (API) for reading to and writing from a virtual machine’s memory.
- Modified to support KVM hypervisor
- That’s why named as LibVMI
- Xen provides built-in functionality to support VMI whereas KVM doesn’t provide any

Features

- Read and write arbitrary data from and to memory
- Access memory using physical addresses, virtual addresses, or kernel symbols
- Parse kernel symbols dynamically from running Windows kernel
- Load Linux kernel symbols from system map file
- Expose useful address translation functions through API functions to resolve kernel symbols to a virtual address or translate a kernel or user virtual address into a physical address
- Pause/unpause the VM through an API function
- Write your introspection code once and have it work across multiple virtualization platforms

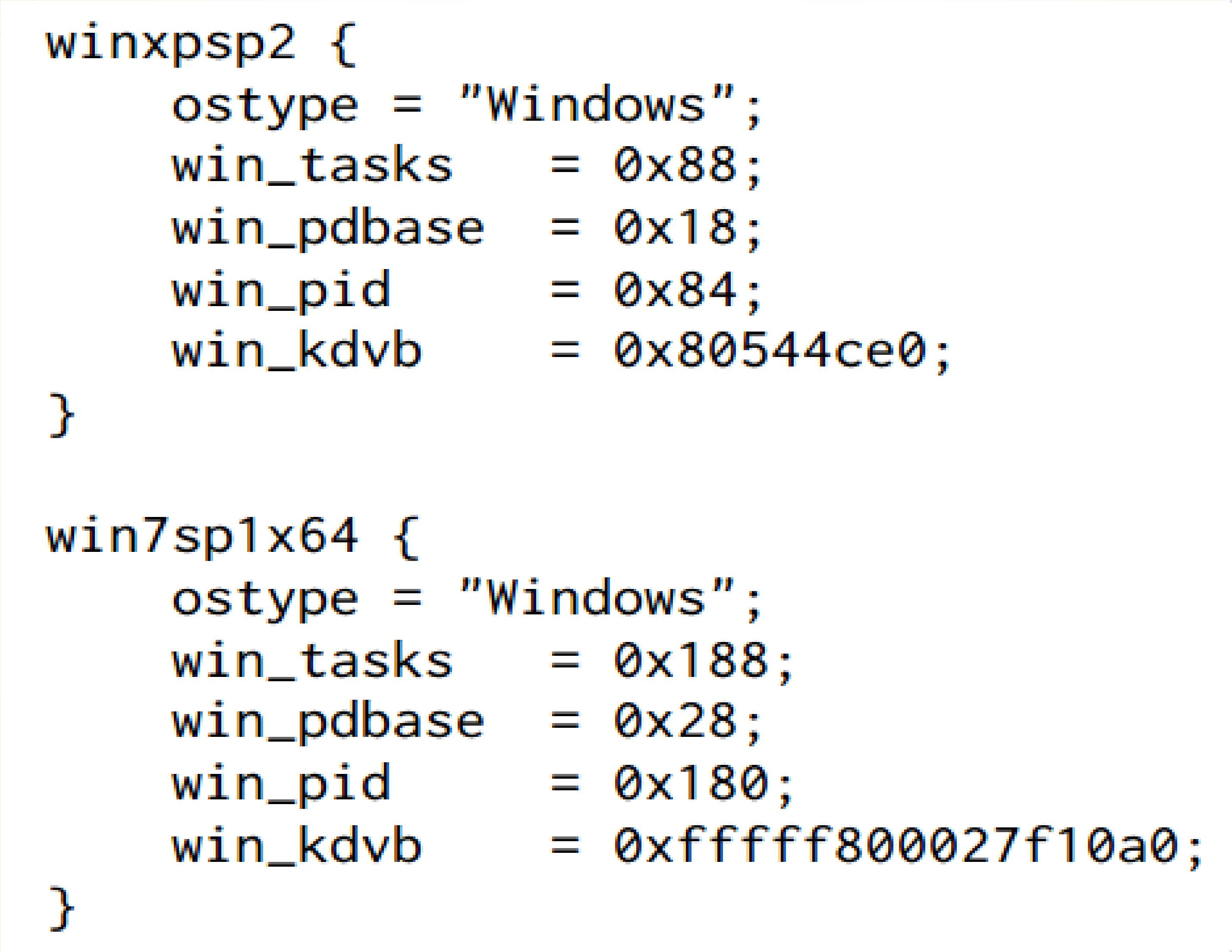
Features (cont.)



Using Introspection To View A Kernel Symbol

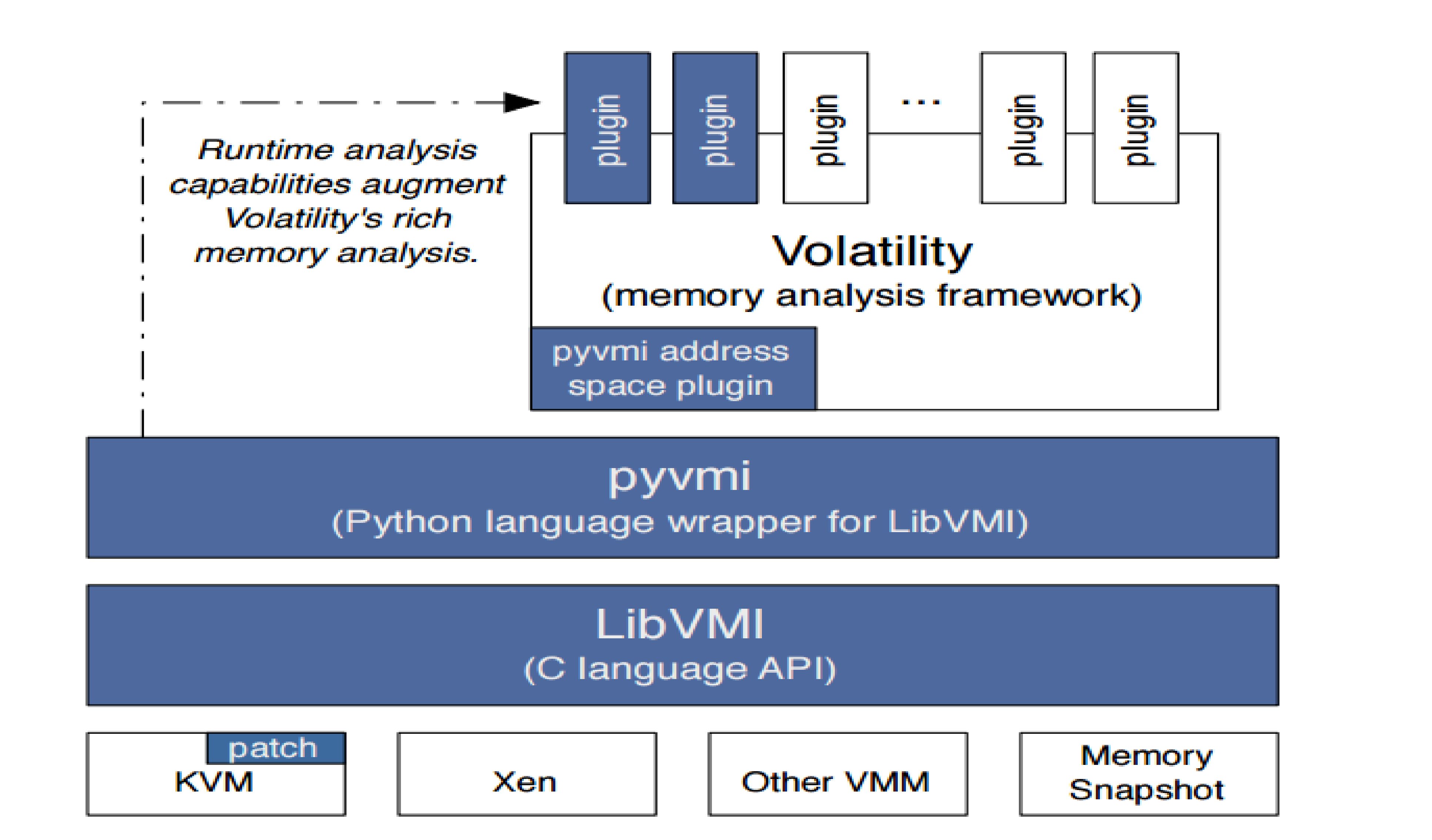
(1) The VMI application requests to view a kernel symbol. (2) LibVMI finds the virtual address for the kernel symbol. (3) Kernel page directory mapped to find correct PT. (4) PT mapped to find correct data page. (5) Data page returned to LibVMI Library. (6) LibVMI returns the data requested by the VMI application (may require mapping multiple pages).

libvmi.conf example



```
winxpsp2 {
    ostype = "Windows";
    win_tasks   = 0x88;
    win_pdbase  = 0x18;
    win_pid     = 0x84;
    win_kdvb    = 0x80544ce0;
}

win7sp1x64 {
    ostype = "Windows";
    win_tasks   = 0x188;
    win_pdbase  = 0x28;
    win_pid     = 0x180;
    win_kdvb    = 0xfffff800027f10a0;
}
```

Features (cont.)


plugin
plugin
plugin
...
plugin
plugin
Runtime analysis capabilities augment Volatility's rich memory analysis.
Volatility
(memory analysis framework)
pyvmi address space plugin
pyvmi
(Python language wrapper for LibVMI)
LibVMI
(C language API)
patch
KVM
Xen
Other VMM
Memory Snapshot

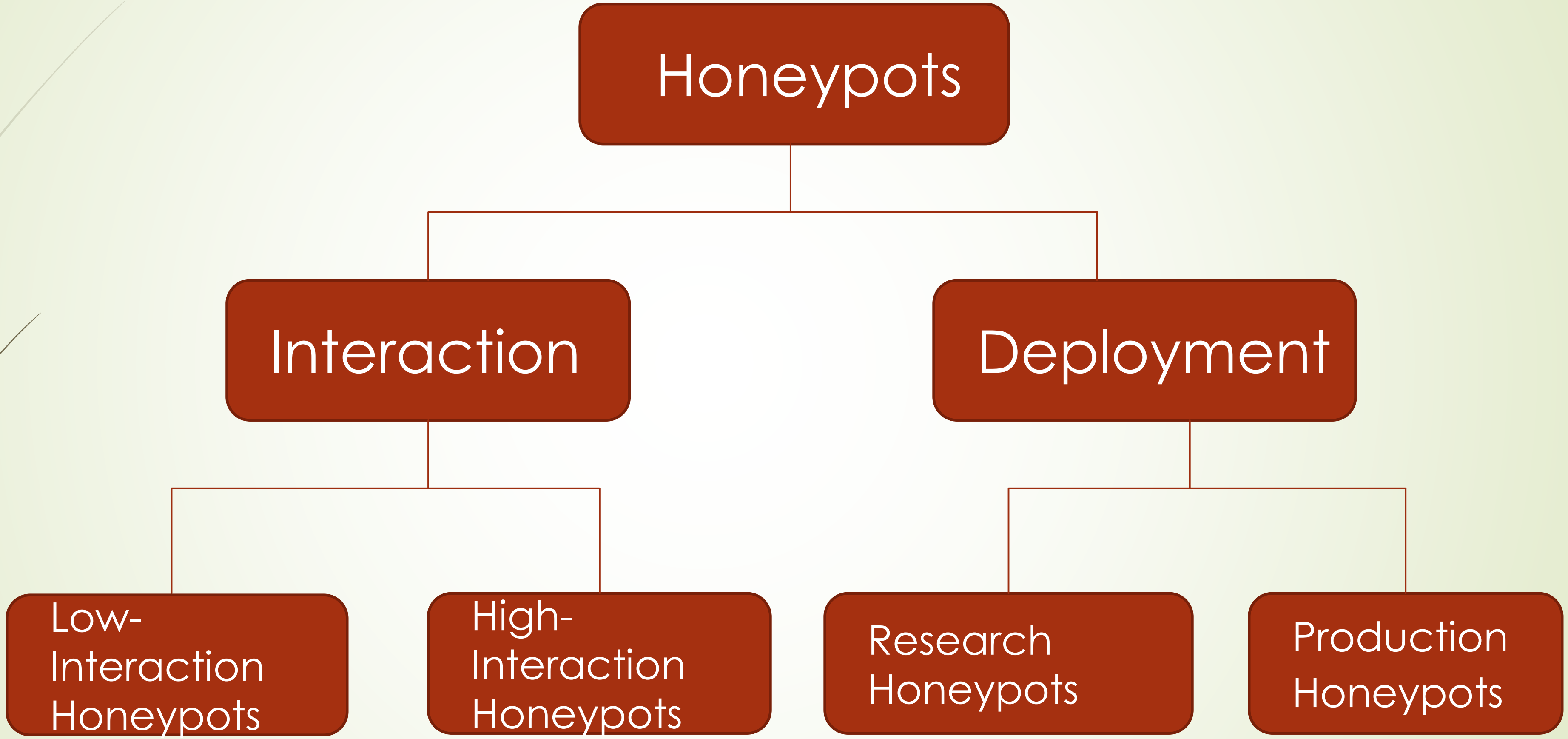
Virtual Honeypots

- Virtual honeypots exist as a virtual resource instead of dedicated physical system with the purpose of attracting and logging cyber-attacks in real time
  - Often emulate or are exposed to live security vulnerabilities in order to capture and monitor both malware and cyber-attackers
  - Can be used to monitor various protocols, applications, or operating system attacks
  - Malware execution behaviors can be logged and can be used in malware research

Virtual honeypots (cont.)

- Detection & Response not prevention
  - Collects evidence information and detects attack patterns
  - Defenders can respond to these evidences by building better defenses and countermeasures against future security threats

Honeypots Categorization



Related work

- *CloudVMI* - VMI offered as a service in public clouds
- *VMI-Honeymon* - high-interaction honeypot monitor which uses virtual machine memory introspection on Xen
- *Livewire*
- *Collapsar*
- *VMScope*

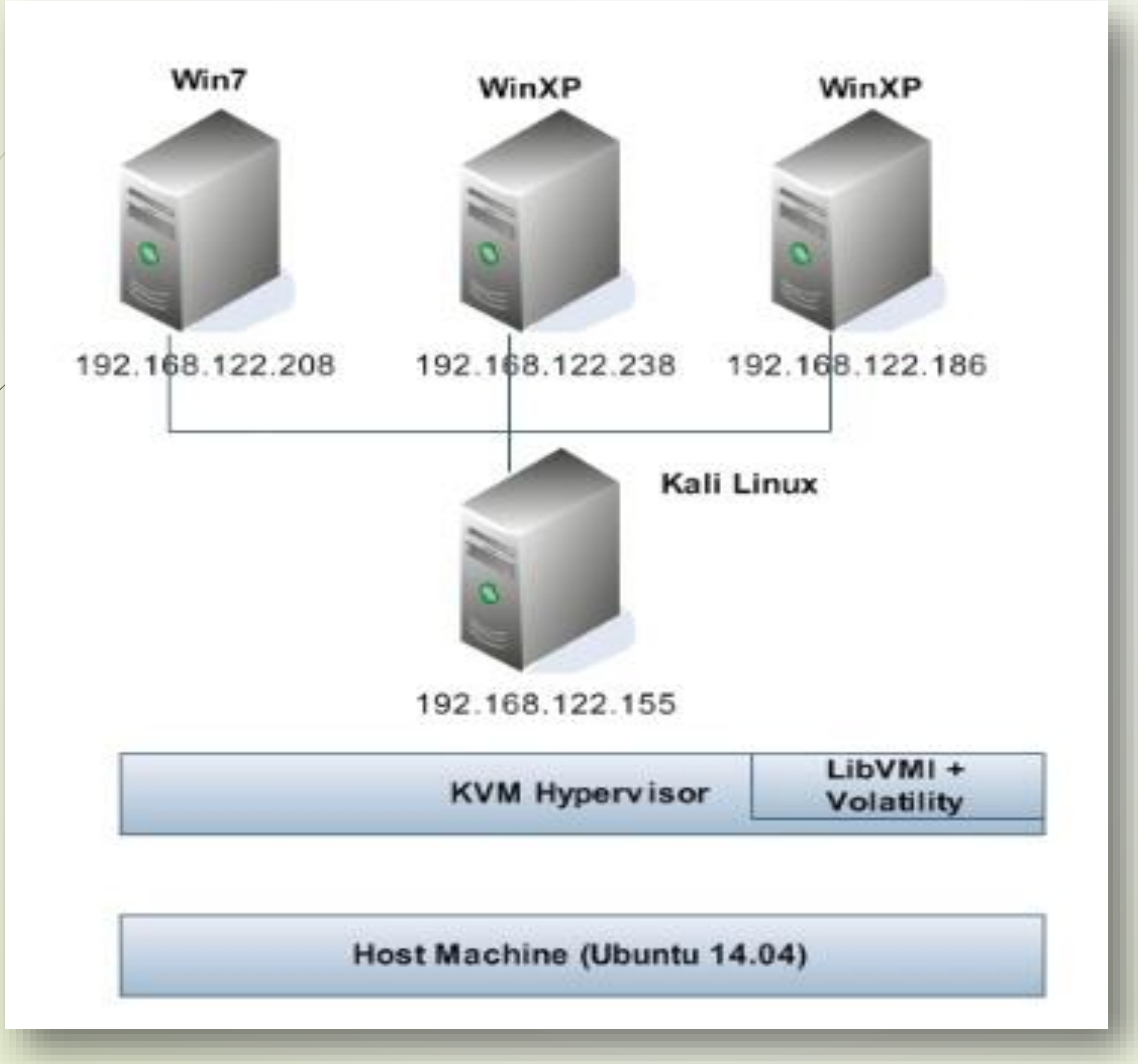
Methodology

- VMI capability is combined with malware analysis and virtual honeypots to achieve the objective
- Extracted IOCs are then converted in STIX programming language

Architecture Design

- KVM hypervisor
- Server Virtualization
- Host-only networking
- LibVMI and Volatility
- Virtual Honeypots

Architecture Design (cont.)



LibVMI KVM support

- For KVM there are two approaches to access VM memory

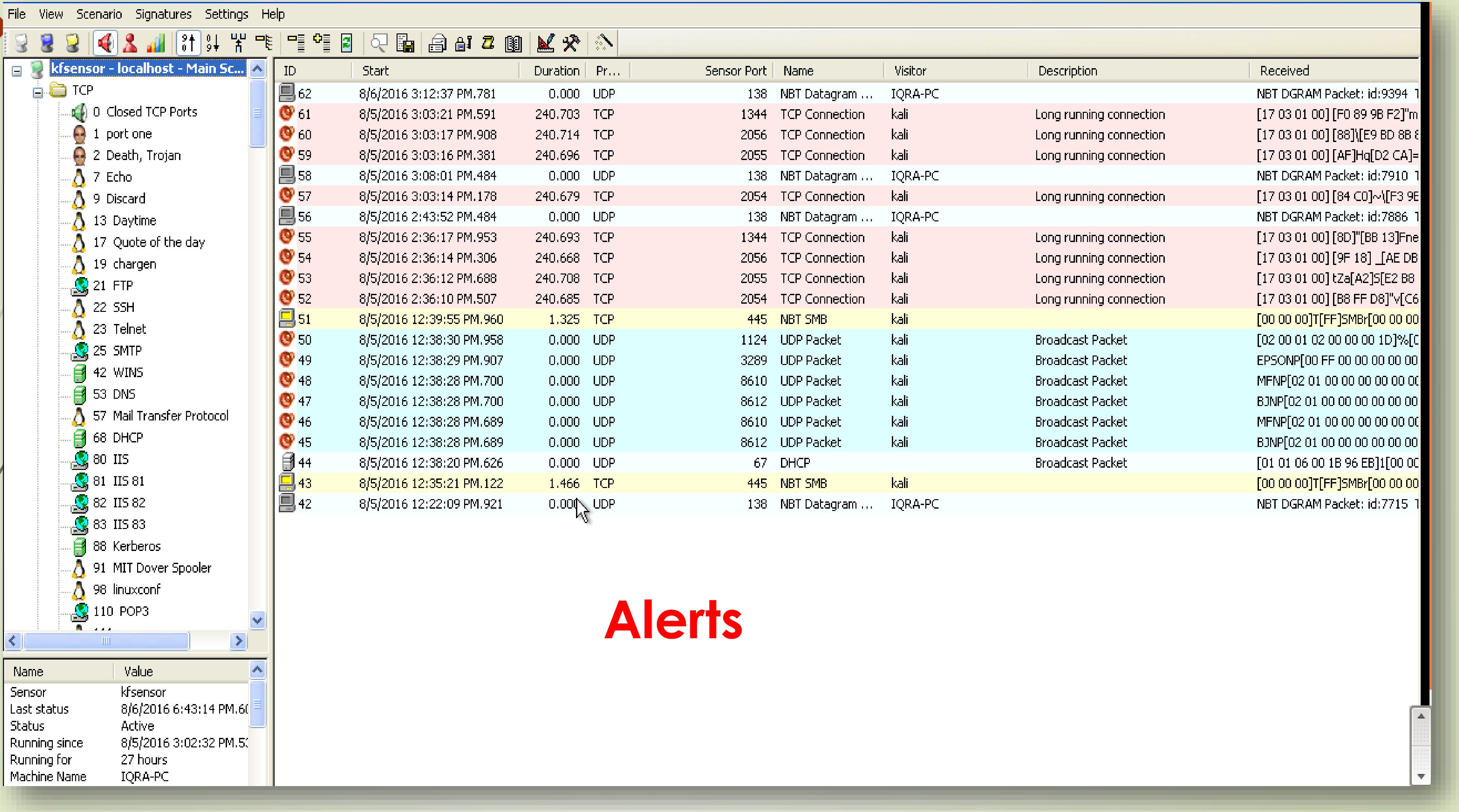
1. GDB (GNU Debugger)
2. A patch created for KVM that enabled memory access through a UNIX domain socket

KVM (kernel-based VM) hypervisor

- Hypervisor of choice for open source clouds
- Low cost
- High scalability
- Ease of deployment
- Openstack
- IBM SmartCloud Enterprise
- Intel IT

Deployed Honeypots

- Kfsensor
- Valhala
- HoneyBOT



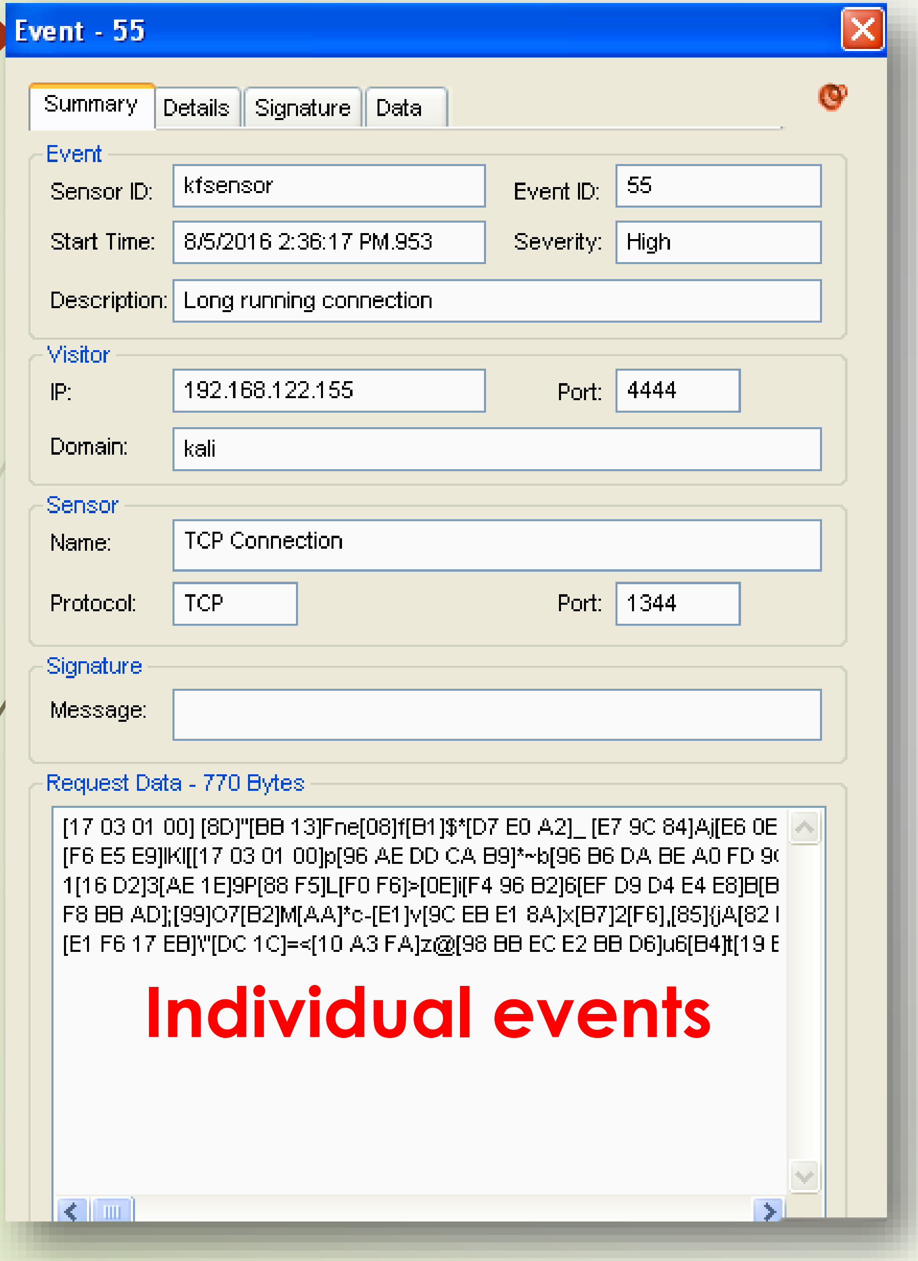
kfsensor - localhost - Main Sc...

- TCP
  - 0 Closed TCP Ports
  - 1 port one
  - 2 Death, Trojan
  - 7 Echo
  - 9 Discard
  - 13 Daytime
  - 17 Quote of the day
  - 19 chargen
  - 21 FTP
  - 22 SSH
  - 23 Telnet
  - 25 SMTP
  - 42 WINS
  - 53 DNS
  - 57 Mail Transfer Protocol
  - 68 DHCP
  - 80 IIS
  - 81 IIS 81
  - 82 IIS 82
  - 83 IIS 83
  - 88 Kerberos
  - 91 MIT Dover Spooler
  - 98 linuxconf
  - 110 POP3

Name	Value
Sensor	kfsensor
Last status	8/6/2016 6:43:14 PM.6(
Status	Active
Running since	8/5/2016 3:02:32 PM.5:
Running for	27 hours
Machine Name	IQRA-PC

ID	Start	Duration	Pr...	Sensor Port	Name	Visitor	Description	Received
62	8/6/2016 3:12:37 PM.781	0.000	UDP	138	NBT Datagram ...	IQRA-PC		NBT DGRAM Packet: id:9394 T
61	8/5/2016 3:03:21 PM.591	240.703	TCP	1344	TCP Connection	kali	Long running connection	[17 03 01 00] [F0 89 9B F2]"m
60	8/5/2016 3:03:17 PM.908	240.714	TCP	2056	TCP Connection	kali	Long running connection	[17 03 01 00] [88]\[E9 BD 8B 8
59	8/5/2016 3:03:16 PM.381	240.696	TCP	2055	TCP Connection	kali	Long running connection	[17 03 01 00] [AF]Hq[D2 CA]=
58	8/5/2016 3:08:01 PM.484	0.000	UDP	138	NBT Datagram ...	IQRA-PC		NBT DGRAM Packet: id:7910 T
57	8/5/2016 3:03:14 PM.178	240.679	TCP	2054	TCP Connection	kali	Long running connection	[17 03 01 00] [84 C0]~\[F3 9E
56	8/5/2016 2:43:52 PM.484	0.000	UDP	138	NBT Datagram ...	IQRA-PC		NBT DGRAM Packet: id:7886 T
55	8/5/2016 2:36:17 PM.953	240.693	TCP	1344	TCP Connection	kali	Long running connection	[17 03 01 00] [8D]"[BB 13]Fne
54	8/5/2016 2:36:14 PM.306	240.668	TCP	2056	TCP Connection	kali	Long running connection	[17 03 01 00] [9F 18] _[AE DB
53	8/5/2016 2:36:12 PM.688	240.708	TCP	2055	TCP Connection	kali	Long running connection	[17 03 01 00] tZa[A2]S[E2 B8
52	8/5/2016 2:36:10 PM.507	240.685	TCP	2054	TCP Connection	kali	Long running connection	[17 03 01 00] [B8 FF D8]"v[C6
51	8/5/2016 12:39:55 PM.960	1.325	TCP	445	NBT SMB	kali		[00 00 00]T[FF]SMBr[00 00 00
50	8/5/2016 12:38:30 PM.958	0.000	UDP	1124	UDP Packet	kali	Broadcast Packet	[02 00 01 02 00 00 00 1D]%[C
49	8/5/2016 12:38:29 PM.907	0.000	UDP	3289	UDP Packet	kali	Broadcast Packet	EPSONP[00 FF 00 00 00 00 00
48	8/5/2016 12:38:28 PM.700	0.000	UDP	8610	UDP Packet	kali	Broadcast Packet	MFNP[02 01 00 00 00 00 00 00
47	8/5/2016 12:38:28 PM.700	0.000	UDP	8612	UDP Packet	kali	Broadcast Packet	BJNP[02 01 00 00 00 00 00 00
46	8/5/2016 12:38:28 PM.689	0.000	UDP	8610	UDP Packet	kali	Broadcast Packet	MFNP[02 01 00 00 00 00 00 00
45	8/5/2016 12:38:28 PM.689	0.000	UDP	8612	UDP Packet	kali	Broadcast Packet	BJNP[02 01 00 00 00 00 00 00
44	8/5/2016 12:38:20 PM.626	0.000	UDP	67	DHCP		Broadcast Packet	[01 01 06 00 1B 96 EB]1[00 00
43	8/5/2016 12:35:21 PM.122	1.466	TCP	445	NBT SMB	kali		[00 00 00]T[FF]SMBr[00 00 00
42	8/5/2016 12:22:09 PM.921	0.000	UDP	138	NBT Datagram ...	IQRA-PC		NBT DGRAM Packet: id:7715 T

Alerts



Event - 55

Event

Sensor ID:	kfsensor	Event ID:	55
Start Time:	8/5/2016 2:36:17 PM.953	Severity:	High
Description:	Long running connection		

Visitor

IP:	192.168.122.155	Port:	4444
Domain:	kali		

Sensor

Name:	TCP Connection		
Protocol:	TCP	Port:	1344

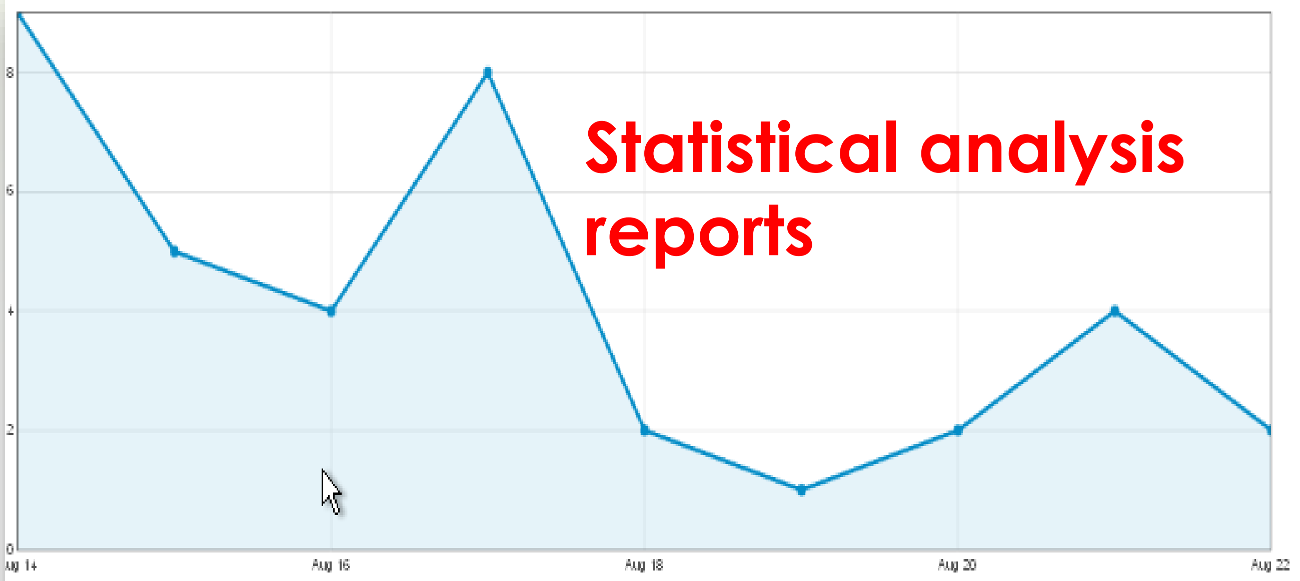
Signature

Message:

Request Data - 770 Bytes

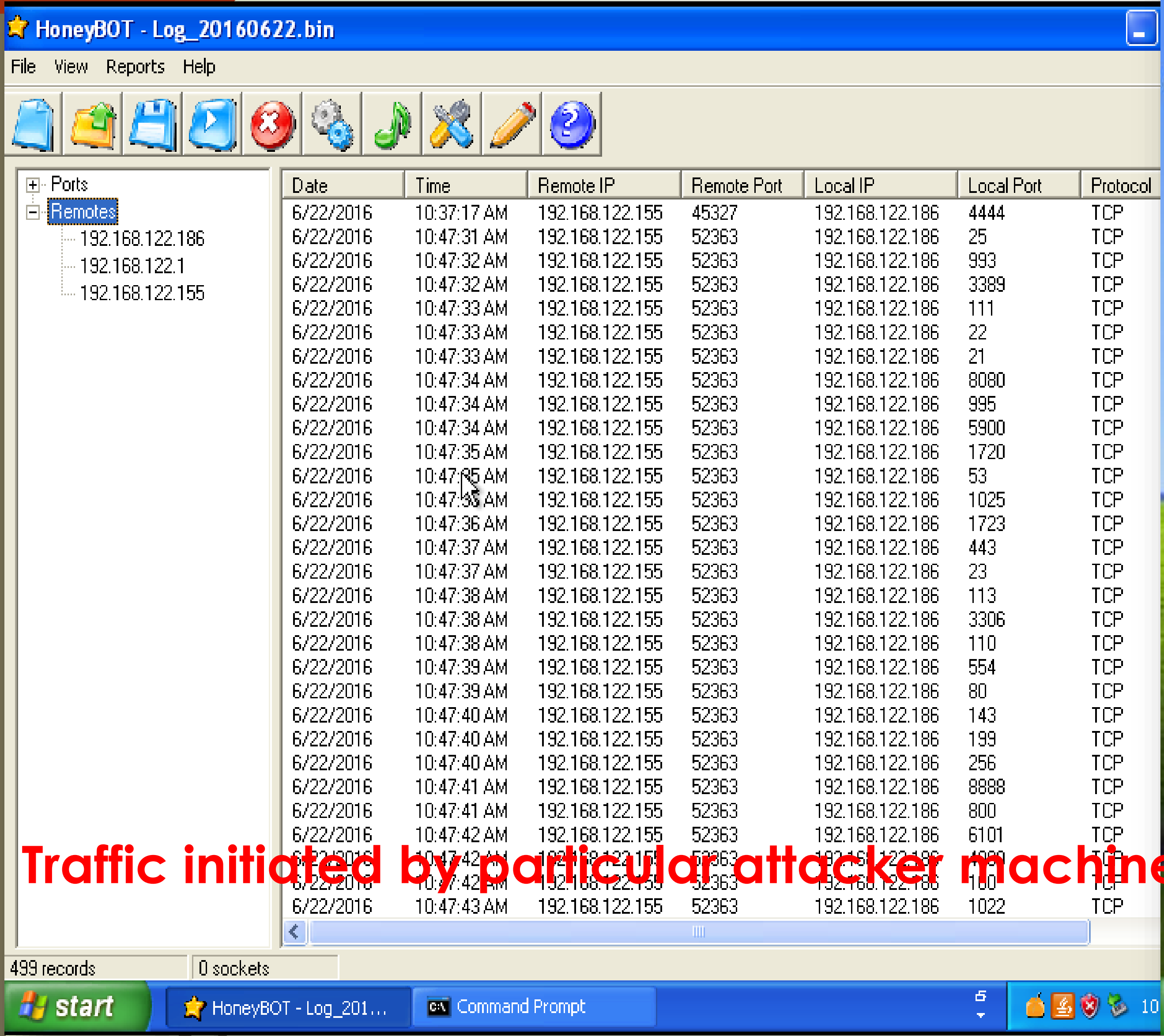
```
[17 03 01 00] [8D]"[BB 13]Fne[08]f[B1]$*[D7 E0 A2]_ [E7 9C 84]Aj[E6 0E
[F6 E5 E9]lKl[[17 03 01 00]p[96 AE DD CA B9]*~b[96 B6 DA BE A0 FD 9(
1[16 D2]3[AE 1E]9P[88 F5]L[F0 F6]>[0E]i[F4 96 B2]6[EF D9 D4 E4 E8]B[B
F8 BB AD];[99]O7[B2]M[AA]*c-[E1]v[9C EB E1 8A]x[B7]2[F6],[85]{jA[82 I
[E1 F6 17 EB]\"[DC 1C]=<[10 A3 FA]z@[98 BB EC E2 BB D6]u6[B4]t[19 E
```

Individual events



Statistical analysis reports

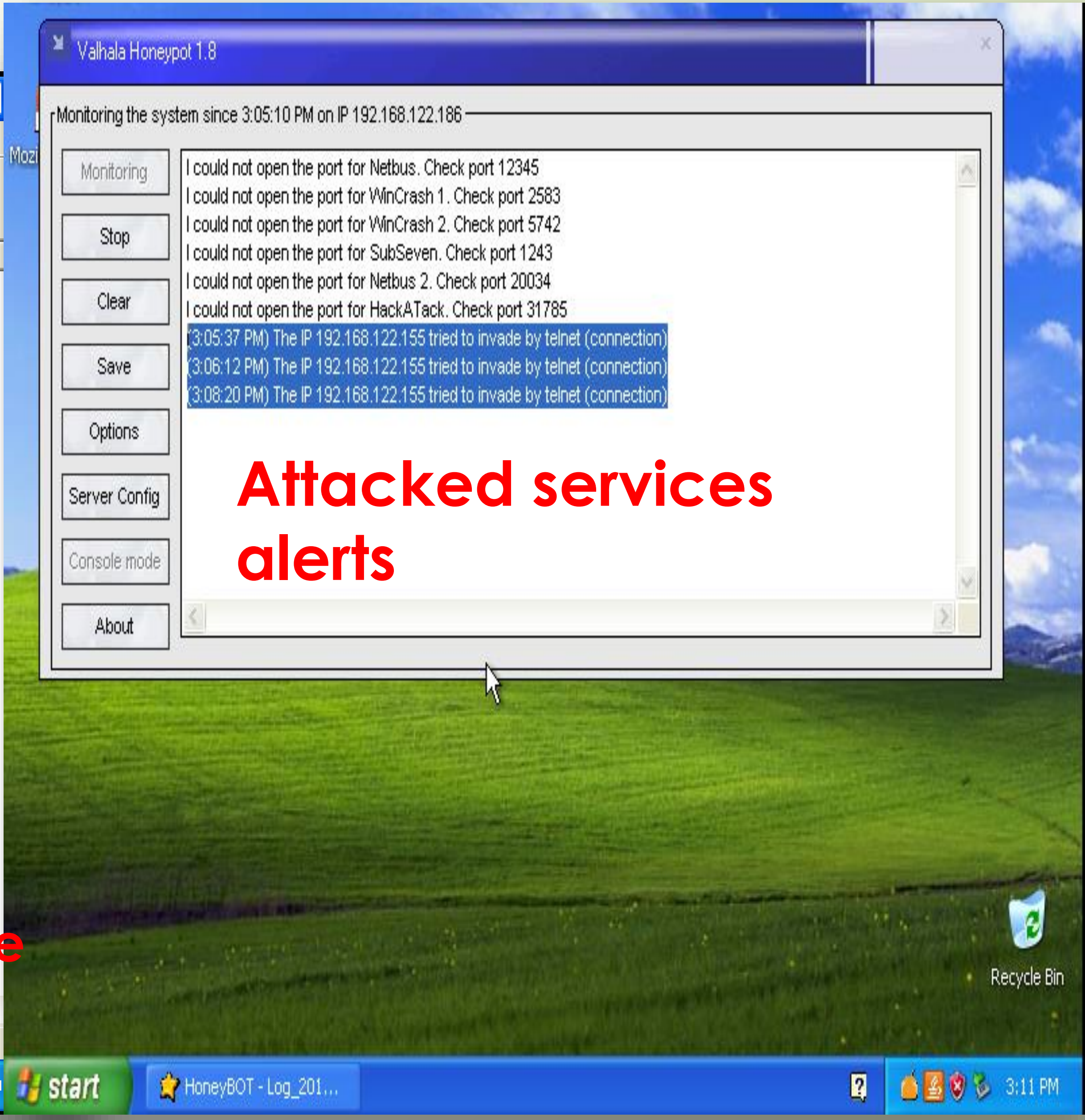
Date	Count
August 14, 2016	9
August 15, 2016	5
August 16, 2016	4
August 17, 2016	8
August 18, 2016	2
August 19, 2016	1
August 20, 2016	2
August 21, 2016	4
August 22, 2016	2



Date	Time	Remote IP	Remote Port	Local IP	Local Port	Protocol
6/22/2016	10:37:17 AM	192.168.122.155	45327	192.168.122.186	4444	TCP
6/22/2016	10:47:31 AM	192.168.122.155	52363	192.168.122.186	25	TCP
6/22/2016	10:47:32 AM	192.168.122.155	52363	192.168.122.186	993	TCP
6/22/2016	10:47:32 AM	192.168.122.155	52363	192.168.122.186	3389	TCP
6/22/2016	10:47:33 AM	192.168.122.155	52363	192.168.122.186	111	TCP
6/22/2016	10:47:33 AM	192.168.122.155	52363	192.168.122.186	22	TCP
6/22/2016	10:47:33 AM	192.168.122.155	52363	192.168.122.186	21	TCP
6/22/2016	10:47:34 AM	192.168.122.155	52363	192.168.122.186	8080	TCP
6/22/2016	10:47:34 AM	192.168.122.155	52363	192.168.122.186	995	TCP
6/22/2016	10:47:34 AM	192.168.122.155	52363	192.168.122.186	5900	TCP
6/22/2016	10:47:35 AM	192.168.122.155	52363	192.168.122.186	1720	TCP
6/22/2016	10:47:35 AM	192.168.122.155	52363	192.168.122.186	53	TCP
6/22/2016	10:47:36 AM	192.168.122.155	52363	192.168.122.186	1025	TCP
6/22/2016	10:47:36 AM	192.168.122.155	52363	192.168.122.186	1723	TCP
6/22/2016	10:47:37 AM	192.168.122.155	52363	192.168.122.186	443	TCP
6/22/2016	10:47:37 AM	192.168.122.155	52363	192.168.122.186	23	TCP
6/22/2016	10:47:38 AM	192.168.122.155	52363	192.168.122.186	113	TCP
6/22/2016	10:47:38 AM	192.168.122.155	52363	192.168.122.186	3306	TCP
6/22/2016	10:47:38 AM	192.168.122.155	52363	192.168.122.186	110	TCP
6/22/2016	10:47:39 AM	192.168.122.155	52363	192.168.122.186	554	TCP
6/22/2016	10:47:39 AM	192.168.122.155	52363	192.168.122.186	80	TCP
6/22/2016	10:47:40 AM	192.168.122.155	52363	192.168.122.186	143	TCP
6/22/2016	10:47:40 AM	192.168.122.155	52363	192.168.122.186	199	TCP
6/22/2016	10:47:40 AM	192.168.122.155	52363	192.168.122.186	256	TCP
6/22/2016	10:47:41 AM	192.168.122.155	52363	192.168.122.186	8888	TCP
6/22/2016	10:47:41 AM	192.168.122.155	52363	192.168.122.186	800	TCP
6/22/2016	10:47:42 AM	192.168.122.155	52363	192.168.122.186	6101	TCP
6/22/2016	10:47:42 AM	192.168.122.155	52363	192.168.122.186	[illegible]	TCP
6/22/2016	10:47:42 AM	192.168.122.155	52363	192.168.122.186	100	TCP
6/22/2016	10:47:43 AM	192.168.122.155	52363	192.168.122.186	1022	TCP

499 records | 0 sockets

Traffic initiated by particular attacker machine



Monitoring the system since 3:05:10 PM on IP 192.168.122.186

```
I could not open the port for Netbus. Check port 12345
I could not open the port for WinCrash 1. Check port 2583
I could not open the port for WinCrash 2. Check port 5742
I could not open the port for SubSeven. Check port 1243
I could not open the port for Netbus 2. Check port 20034
I could not open the port for HackATack. Check port 31785
(3:05:37 PM) The IP 192.168.122.155 tried to invade by telnet (connection)
(3:06:12 PM) The IP 192.168.122.155 tried to invade by telnet (connection)
(3:08:20 PM) The IP 192.168.122.155 tried to invade by telnet (connection)
```

Attacked services alerts

Volatility

- Volatility is an open source memory forensic tool helping incident response and memory forensics.

Used Volatility plug-ins for memory introspection

- pslist
- pstree
- connections
- connscan
- malfind
- handles
- dlllist
- svscan
- getsids
- strings etc.

IOCs to look for?

- Suspicious processes are spawned out of right path?
- Suspicious process is running under its legitimate parent process, or some

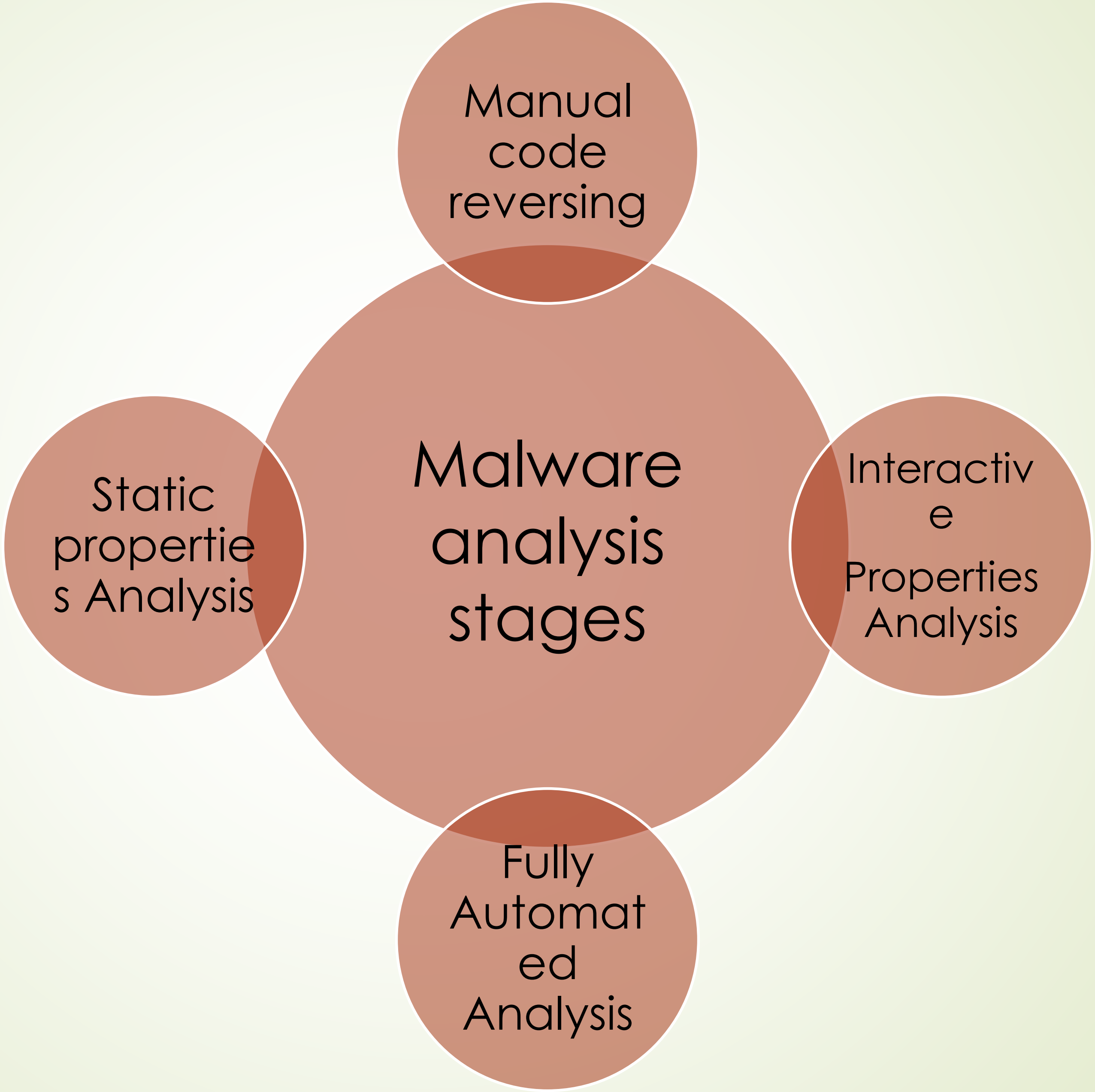
  other process spawned it?
- At what time process started and exited?
- What privileges process under consideration has? Whether this process

  should have these privileges?

IOCs to look for? (cont.)

- Another important point is process name. See is it matching to some legitimate Windows process and malware attacker change it a bit to match a legitimate Windows process to avoid detection.
- See for the associated process objects like threads, mutexes, DLL, process to file mappings, memory Sections, associated sockets and ports open by that process.
- Connections initiated by the process and the connection initiated it

Performed Analysis stages



Malware analysis stages

Manual code reversing

Static properties Analysis

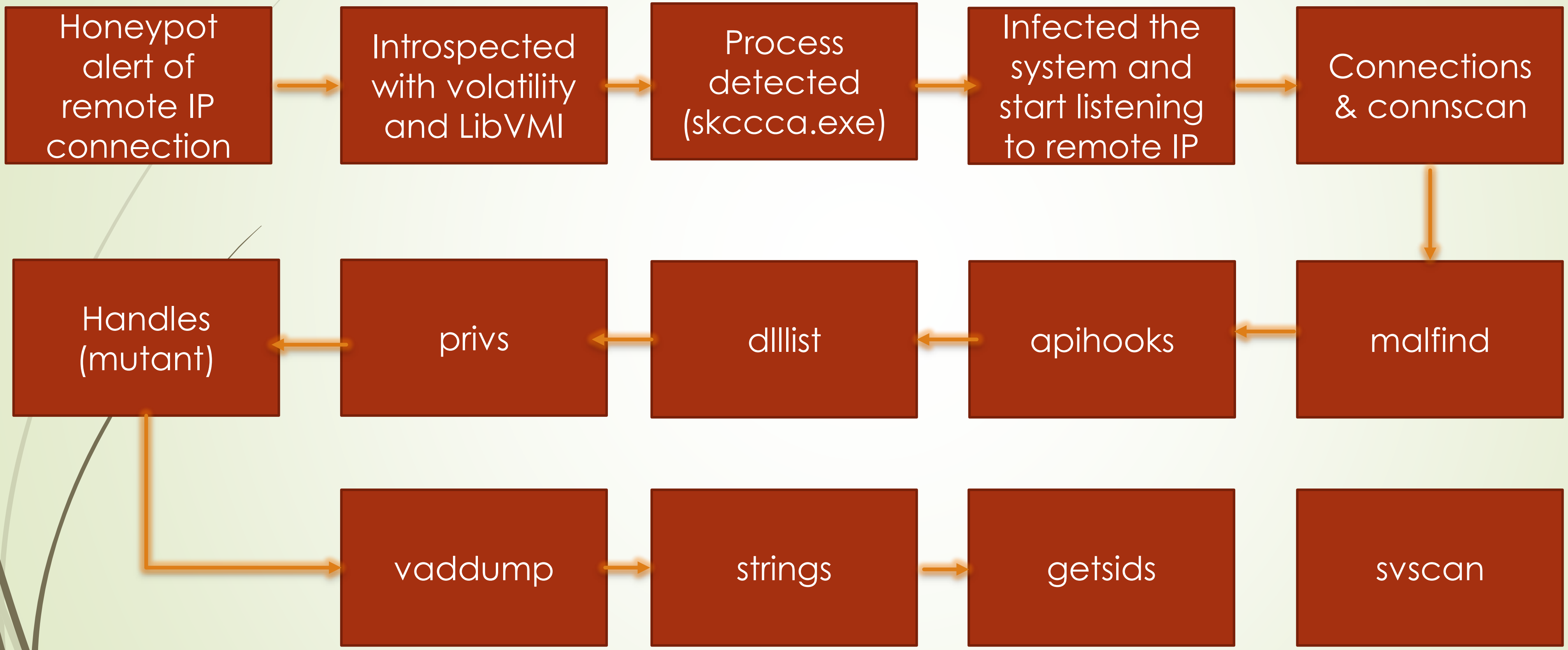
Interactive Properties Analysis

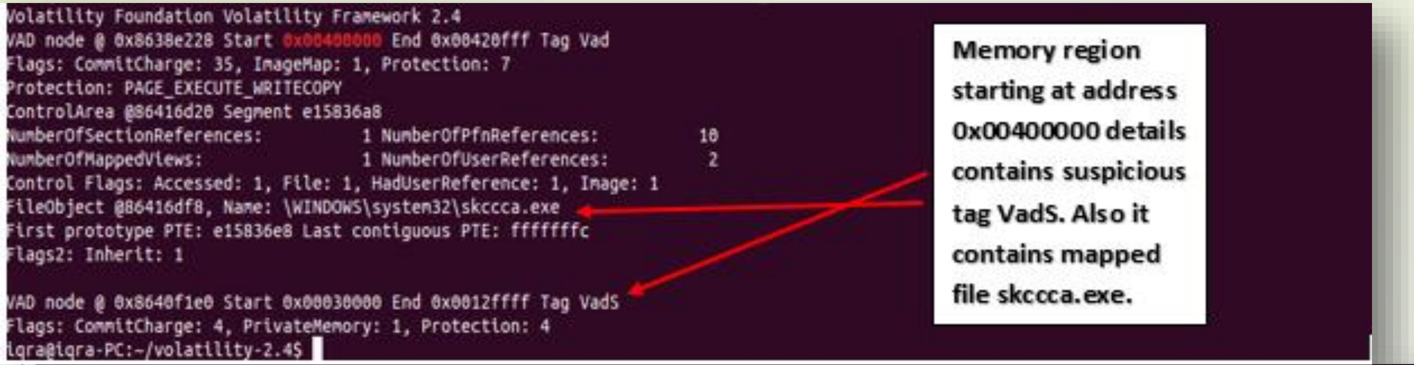
Fully Automated Analysis

Studied attacks

- Reflective Injection
- Trojans
- Attacks on specific vulnerable ports used by most attackers

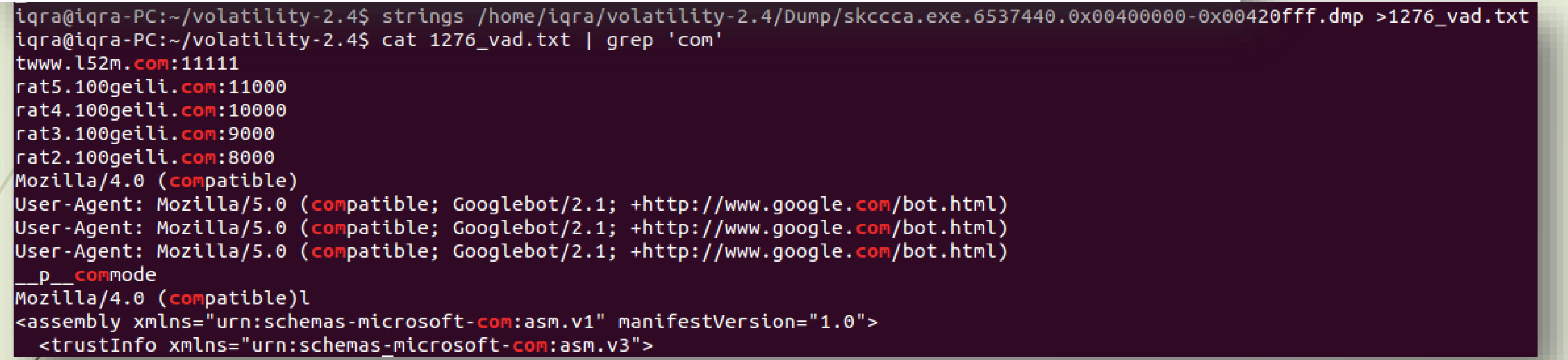
Flow chart


Honeypot alert of remote IP connection
Introspected with volatility and LibVMI
Process detected (skccca.exe)
Infected the system and start listening to remote IP
Connections & connscan
malfind
apihooks
dlllist
privs
Handles (mutant)
vaddump
strings
getsids
svscan

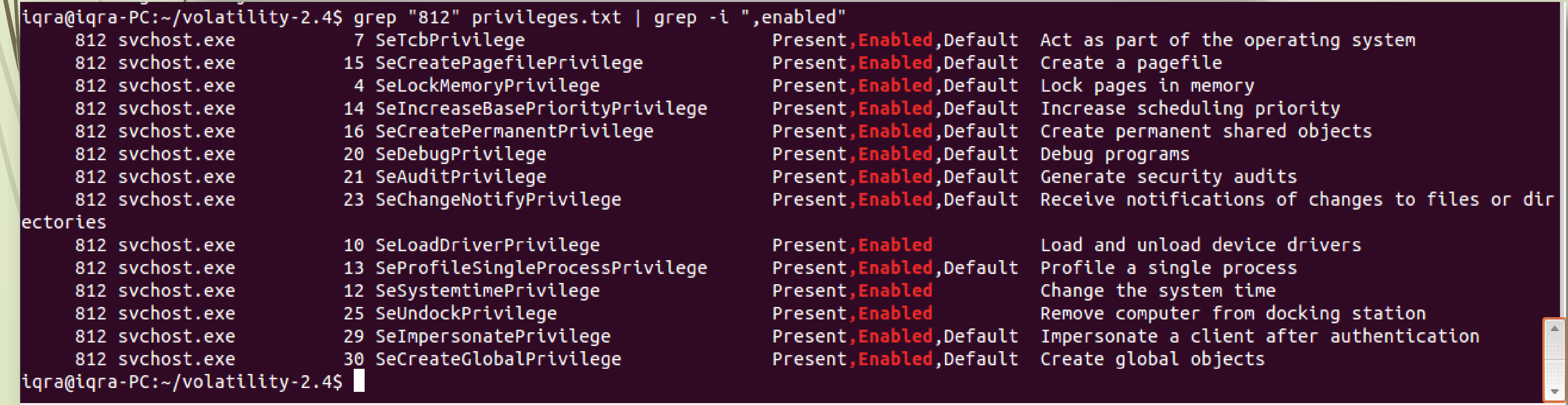


```
Volatility Foundation Volatility Framework 2.4
VAD node @ 0x8638e228 Start 0x00400000 End 0x00420fff Tag Vad
Flags: CommitCharge: 35, ImageMap: 1, Protection: 7
Protection: PAGE_EXECUTE_WRITECOPY
ControlArea @86416d20 Segment e15836a8
NumberOfSectionReferences:          1 NumberOfPfnReferences:         10
NumberOfMappedViews:                1 NumberOfUserReferences:         2
Control Flags: Accessed: 1, File: 1, HadUserReference: 1, Image: 1
FileObject @86416df8, Name: \WINDOWS\system32\skccca.exe
First prototype PTE: e15836e8 Last contiguous PTE: fffffffc
Flags2: Inherit: 1

VAD node @ 0x8640f1e0 Start 0x00030000 End 0x0012ffff Tag VadS
Flags: CommitCharge: 4, PrivateMemory: 1, Protection: 4
iqra@iqra-PC:~/volatility-2.4$
```

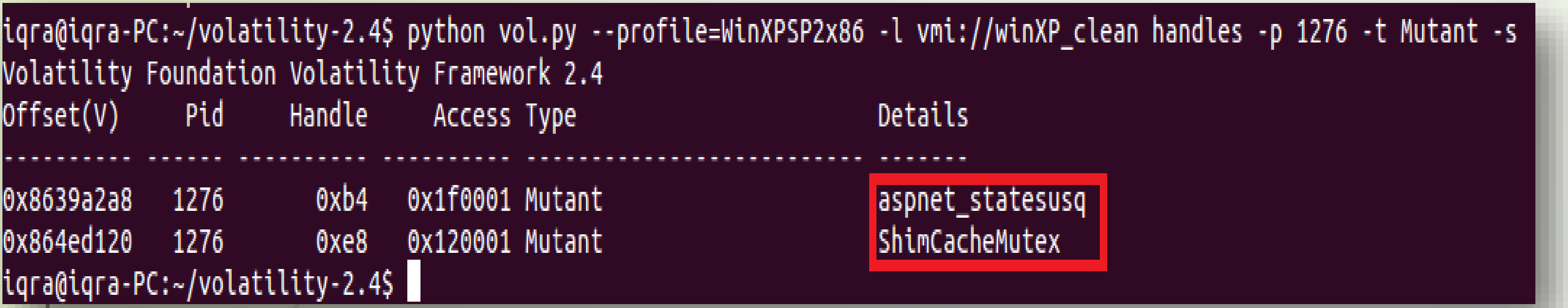
Memory region starting at address 0x00400000 details contains suspicious tag VadS. Also it contains mapped file skccca.exe.



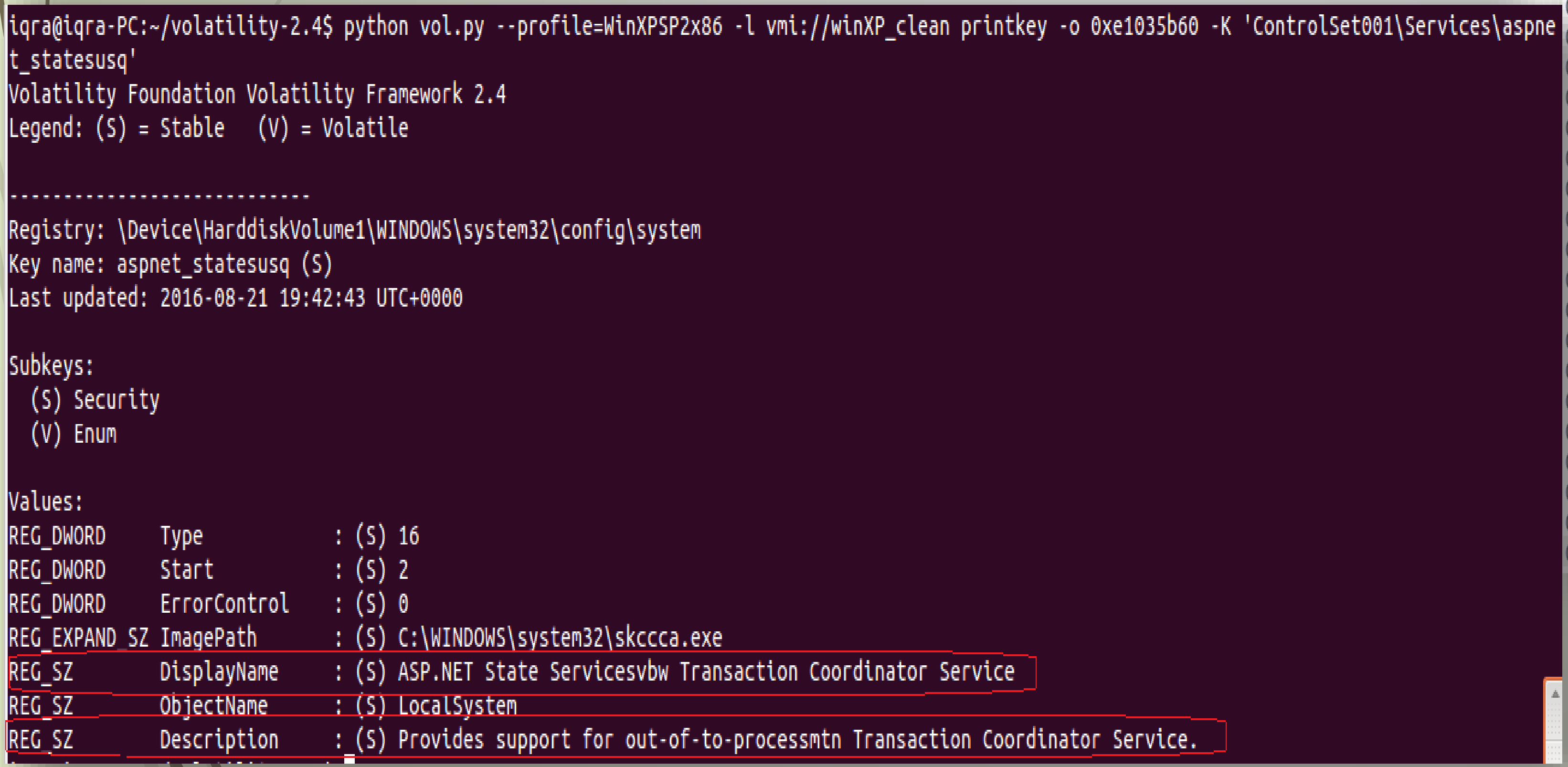
```
iqra@iqra-PC:~/volatility-2.4$ strings /home/iqra/volatility-2.4/Dump/skccca.exe.6537440.0x00400000-0x00420fff.dmp >1276_vad.txt
iqra@iqra-PC:~/volatility-2.4$ cat 1276_vad.txt | grep 'com'
twww.l52m.com:11111
rat5.100geili.com:11000
rat4.100geili.com:10000
rat3.100geili.com:9000
rat2.100geili.com:8000
Mozilla/4.0 (compatible)
User-Agent: Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)
User-Agent: Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)
User-Agent: Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)
__p__commode
Mozilla/4.0 (compatible)l
<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">
  <trustInfo xmlns="urn:schemas-microsoft-com:asm.v3">
```



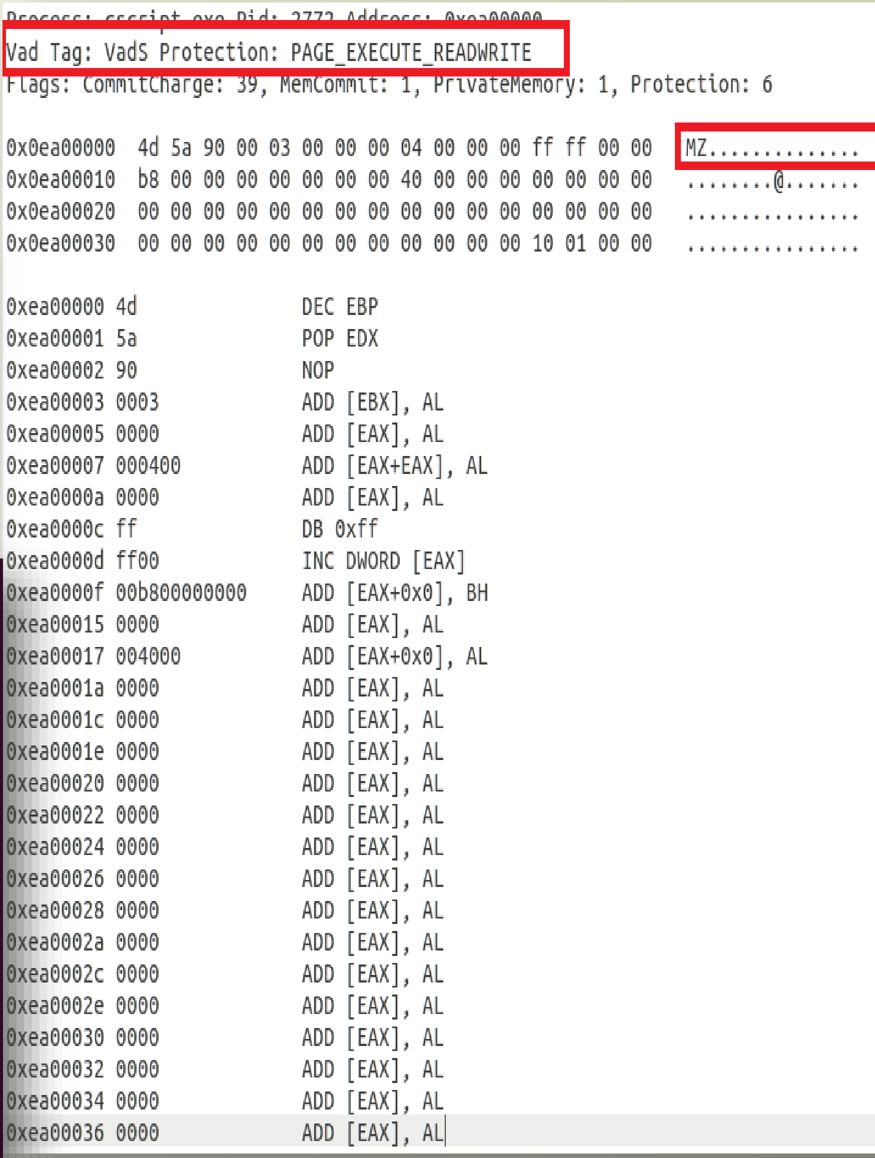
```
iqra@iqra-PC:~/volatility-2.4$ grep "812" privileges.txt | grep -i ",enabled"
     812 svchost.exe             7 SeTcbPrivilege                  Present,Enabled,Default  Act as part of the operating system
     812 svchost.exe            15 SeCreatePagefilePrivilege       Present,Enabled,Default  Create a pagefile
     812 svchost.exe             4 SeLockMemoryPrivilege           Present,Enabled,Default  Lock pages in memory
     812 svchost.exe            14 SeIncreaseBasePriorityPrivilege Present,Enabled,Default  Increase scheduling priority
     812 svchost.exe            16 SeCreatePermanentPrivilege      Present,Enabled,Default  Create permanent shared objects
     812 svchost.exe            20 SeDebugPrivilege                Present,Enabled,Default  Debug programs
     812 svchost.exe            21 SeAuditPrivilege                Present,Enabled,Default  Generate security audits
     812 svchost.exe            23 SeChangeNotifyPrivilege         Present,Enabled,Default  Receive notifications of changes to files or dir
ectories
     812 svchost.exe            10 SeLoadDriverPrivilege           Present,Enabled          Load and unload device drivers
     812 svchost.exe            13 SeProfileSingleProcessPrivilege Present,Enabled,Default  Profile a single process
     812 svchost.exe            12 SeSystemtimePrivilege           Present,Enabled          Change the system time
     812 svchost.exe            25 SeUndockPrivilege               Present,Enabled          Remove computer from docking station
     812 svchost.exe            29 SeImpersonatePrivilege          Present,Enabled,Default  Impersonate a client after authentication
     812 svchost.exe            30 SeCreateGlobalPrivilege         Present,Enabled,Default  Create global objects
iqra@iqra-PC:~/volatility-2.4$
```



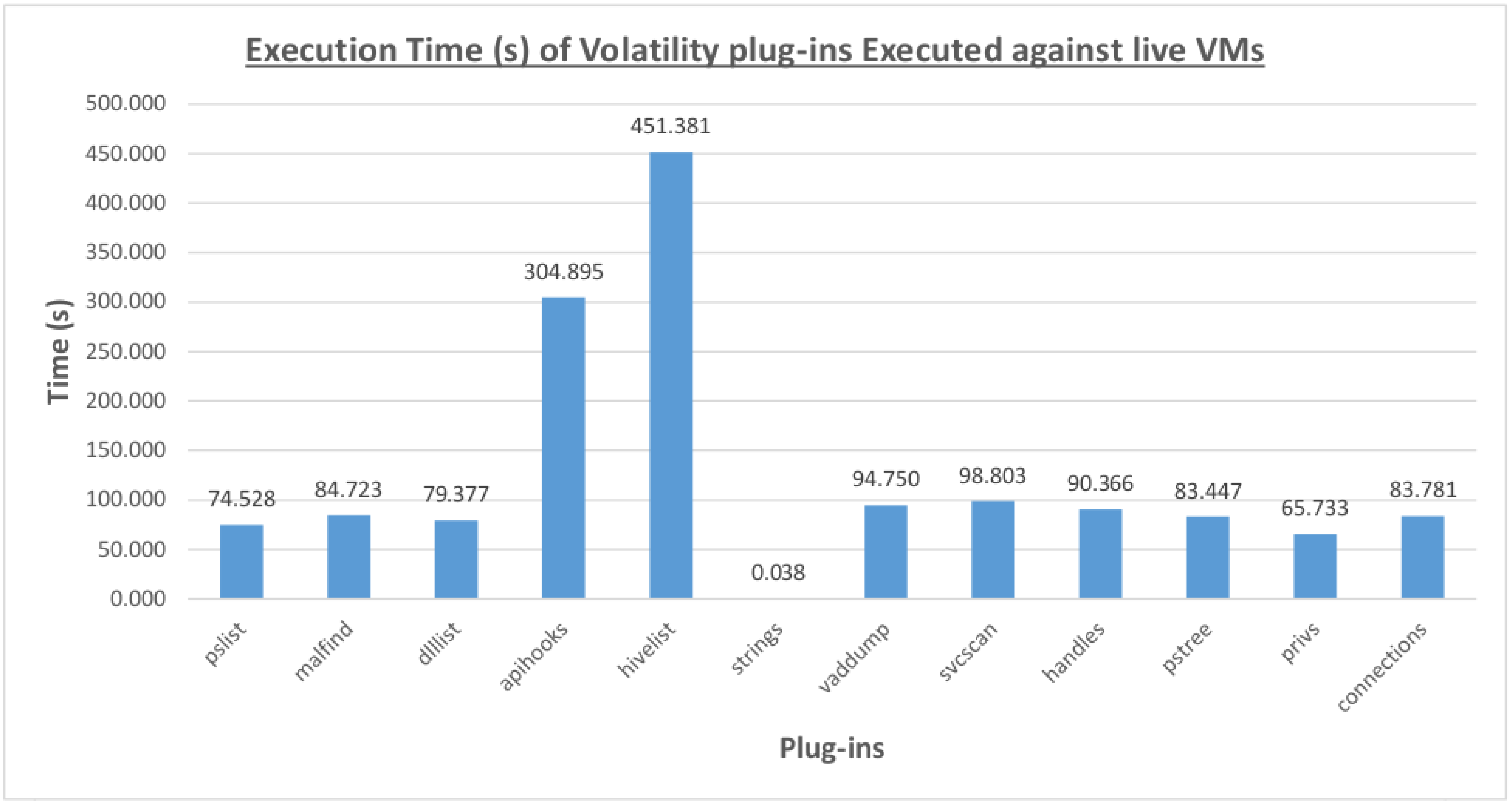
```
iqra@iqra-PC:~/volatility-2.4$ python vol.py --profile=WinXPSP2x86 -l vmi://winXP_clean handles -p 1276 -t Mutant -s
Volatility Foundation Volatility Framework 2.4
Offset(V)     Pid     Handle     Access Type                     Details
---------- ------ ---------- ---------- ------------------------ -------
0x8639a2a8   1276       0xb4   0x1f0001 Mutant                   aspnet_statesusq
0x864ed120   1276       0xe8   0x120001 Mutant                   ShimCacheMutex
iqra@iqra-PC:~/volatility-2.4$
```



```
iqra@iqra-PC:~/volatility-2.4$ python vol.py --profile=WinXPSP2x86 -l vmi://winXP_clean printkey -o 0xe1035b60 -K 'ControlSet001\Services\aspne
t_statesusq'
Volatility Foundation Volatility Framework 2.4
Legend: (S) = Stable   (V) = Volatile

----------------------------
Registry: \Device\HarddiskVolume1\WINDOWS\system32\config\system
Key name: aspnet_statesusq (S)
Last updated: 2016-08-21 19:42:43 UTC+0000

Subkeys:
  (S) Security
  (V) Enum

Values:
REG_DWORD     Type            : (S) 16
REG_DWORD     Start           : (S) 2
REG_DWORD     ErrorControl    : (S) 0
REG_EXPAND_SZ ImagePath       : (S) C:\WINDOWS\system32\skccca.exe
REG_SZ        DisplayName     : (S) ASP.NET State Servicesvbw Transaction Coordinator Service
REG_SZ        ObjectName      : (S) LocalSystem
REG_SZ        Description     : (S) Provides support for out-of-to-processmtn Transaction Coordinator Service.
```



```
Process: cscript.exe Pid: 2772 Address: 0xea00000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 39, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x0ea00000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00   MZ..............
0x0ea00010  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00   ........@.......
0x0ea00020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x0ea00030  00 00 00 00 00 00 00 00 00 00 00 00 10 01 00 00   ................

0xea00000 4d               DEC EBP
0xea00001 5a               POP EDX
0xea00002 90               NOP
0xea00003 0003             ADD [EBX], AL
0xea00005 0000             ADD [EAX], AL
0xea00007 000400           ADD [EAX+EAX], AL
0xea0000a 0000             ADD [EAX], AL
0xea0000c ff               DB 0xff
0xea0000d ff00             INC DWORD [EAX]
0xea0000f 00b800000000     ADD [EAX+0x0], BH
0xea00015 0000             ADD [EAX], AL
0xea00017 004000           ADD [EAX+0x0], AL
0xea0001a 0000             ADD [EAX], AL
0xea0001c 0000             ADD [EAX], AL
0xea0001e 0000             ADD [EAX], AL
0xea00020 0000             ADD [EAX], AL
0xea00022 0000             ADD [EAX], AL
0xea00024 0000             ADD [EAX], AL
0xea00026 0000             ADD [EAX], AL
0xea00028 0000             ADD [EAX], AL
0xea0002a 0000             ADD [EAX], AL
0xea0002c 0000             ADD [EAX], AL
0xea0002e 0000             ADD [EAX], AL
0xea00030 0000             ADD [EAX], AL
0xea00032 0000             ADD [EAX], AL
0xea00034 0000             ADD [EAX], AL
0xea00036 0000             ADD [EAX], AL
```


Execution Time (s) of Volatility plug-ins Executed against live VMs
Time (s)
500.000
450.000
400.000
350.000
300.000
250.000
200.000
150.000
100.000
50.000
0.000
74.528
84.723
79.377
304.895
451.381
0.038
94.750
98.803
90.366
83.447
65.733
83.781
pslist
malfind
dlllist
apihooks
hivelist
strings
vaddump
svcscan
handles
pstree
privs
connections
Plug-ins

Structured Threat Information Expression (STIX)

- A programming language for conveying data about cybersecurity threats in a common language that can be easily understood by humans and security technologies.
- A variety of high-level cyber security use cases rely on such information including:
  - Analyzing cyber threats
  - Specifying indicator patterns for cyber threat
  - Managing cyber threat response activities
  - Sharing cyber threat information
- Consistency, efficiency, interoperability, and overall situational awareness.
- CybOX: Cyber Observable eXpression

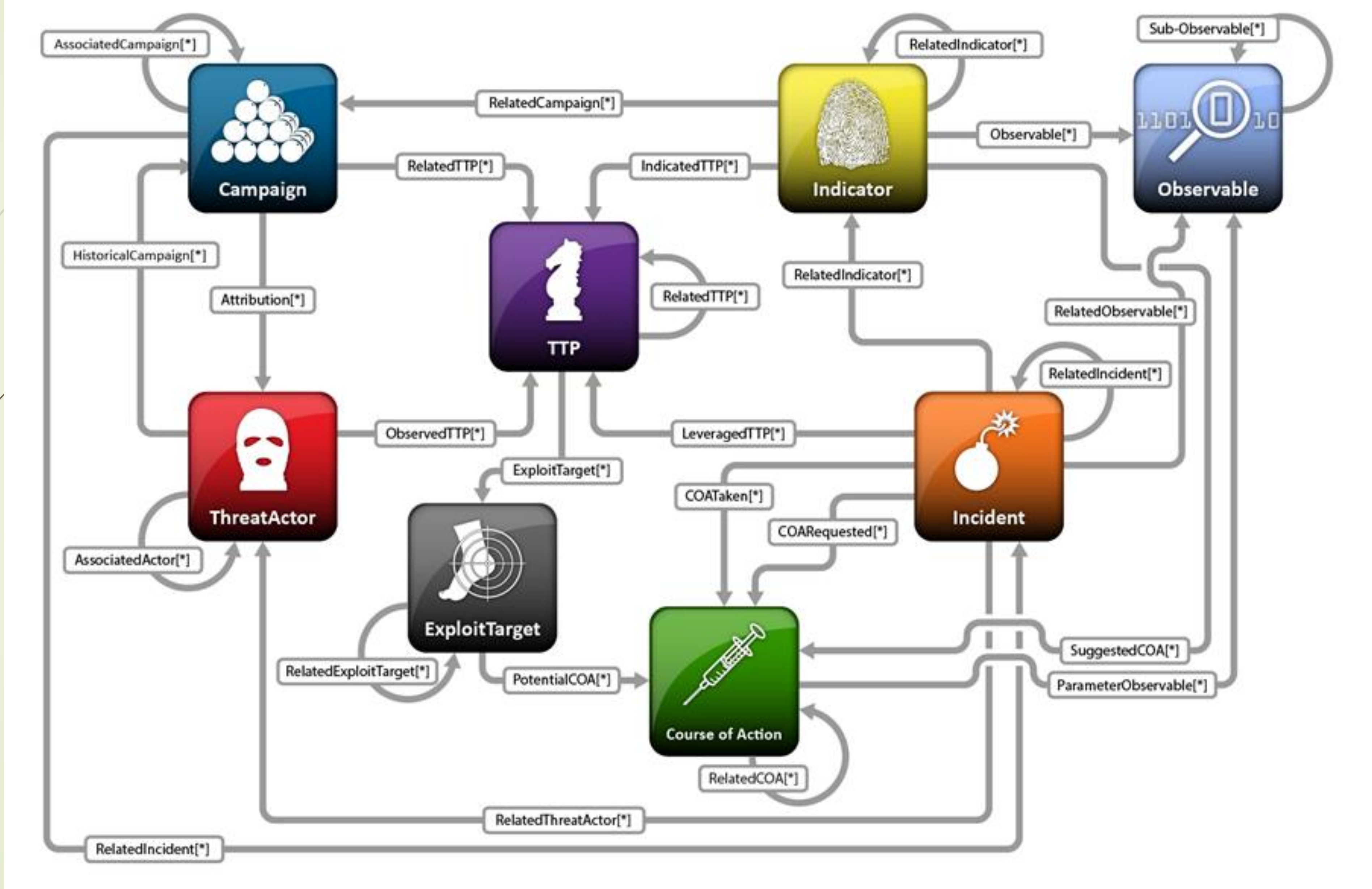
STIX Architecture

- ***Cyber Observables*** - ***what activities we are observing on our networks or systems***
- ***Indicators*** - ***What threats should I look for on my networks and systems and why?***
- ***Incidents*** - ***Where has this threat been seen?***
- ***Adversary Tactics, Techniques, and Procedures*** (including attack patterns, malware, exploits, kill chains, tools, infrastructure, victim targeting, etc.) - ***What does it do?***

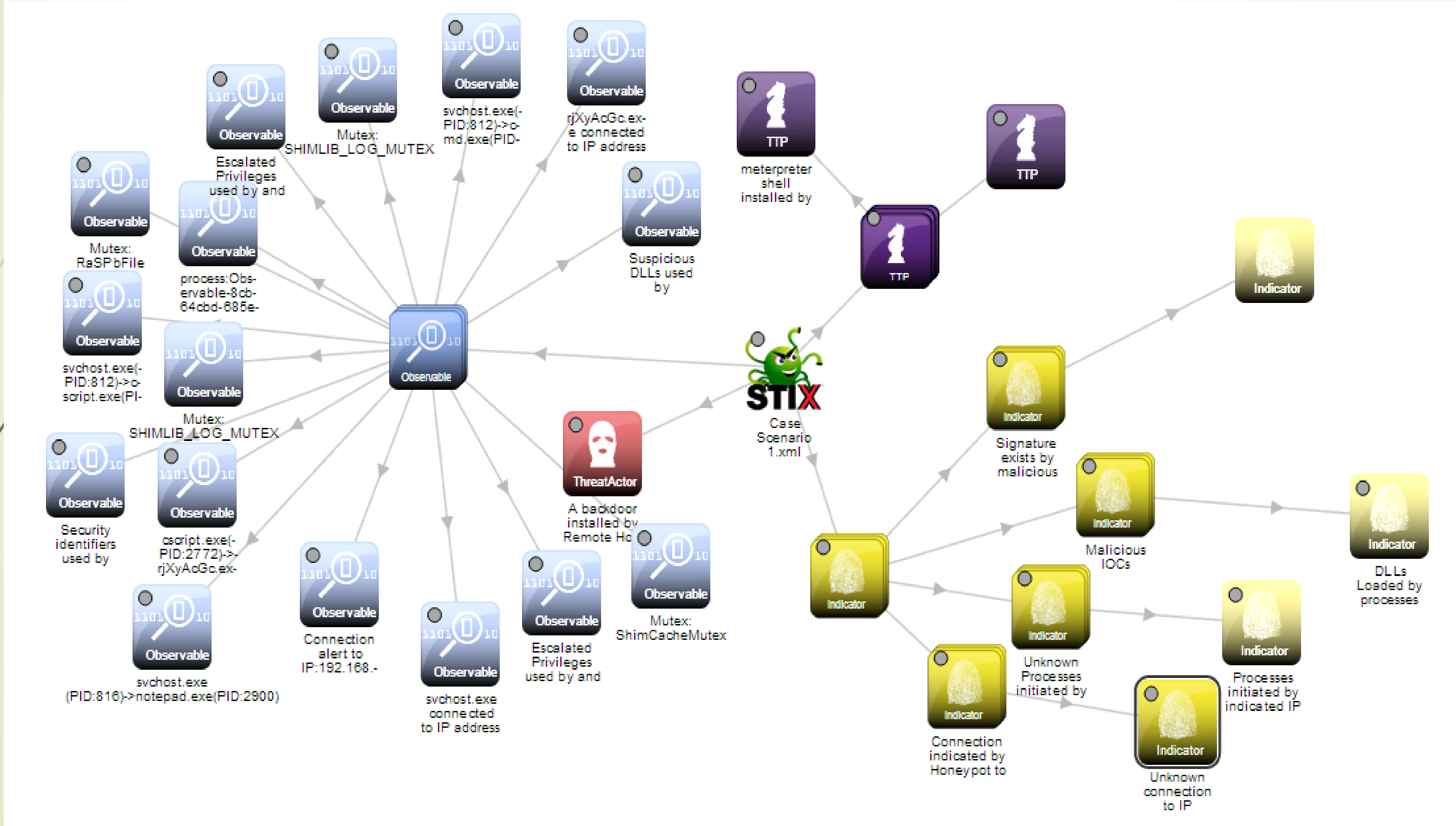
STIX Architecture (cont.)

- ***Exploit Targets*** (e.g., vulnerabilities, weaknesses or configurations) - ***What weaknesses does this threat exploit?***
- ***Courses of Action*** (e.g., incident response or vulnerability/weakness remedies or mitigations) - ***What can we do about it?***
- ***Cyber Attack Campaigns*** - ***Why does it do this?***
- ***Cyber Threat Actors*** - ***Who is responsible for this threat?***

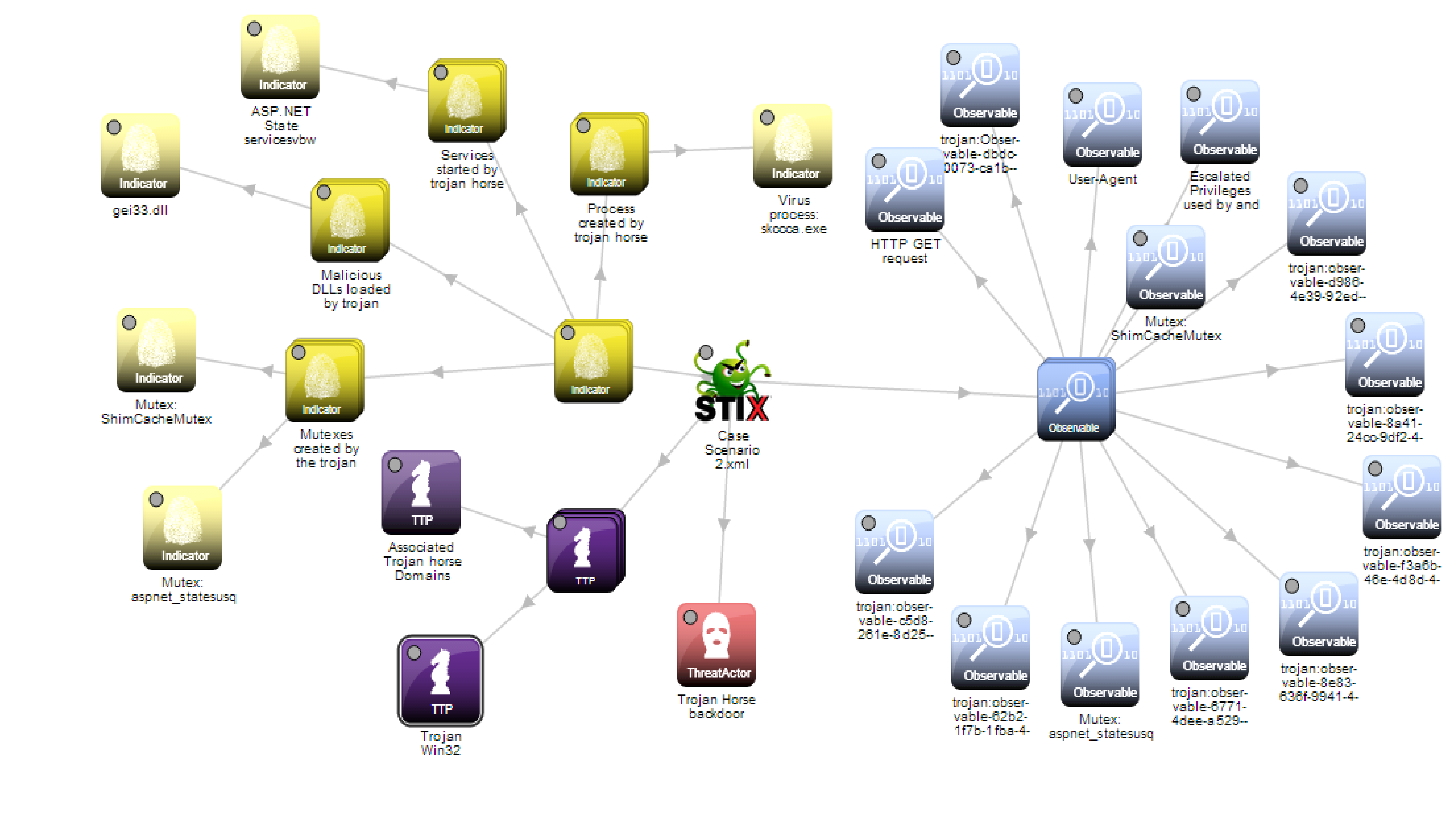
STIX Architecture (cont.)



Converted STIX IOCs


Observable
Escalated Privileges used by and
Observable
Mutex: SHIMLIB_LOG_MUTEX
Observable
svchost.exe(-PID:812)->c-md.exe(PID-
Observable
rjXyAcGc.ex-e connected to IP address
TTP
meterpreter shell installed by
TTP
TTP
Observable
Mutex: RaSPbFile
Observable
process:Obs-ervable-8cb-64cbd-685e-
Observable
Suspicious DLLs used by
Indicator
Observable
svchost.exe(-PID:812)->c-script.exe(PI-
Observable
Mutex: SHIMLIB_LOG_MUTEX
Observable
STIX
Case Scenario 1.xml
Indicator
Signature exists by malicious
Observable
Security identifiers used by
Observable
cscript.exe(-PID:2772)->-rjXyAcGc.ex-
ThreatActor
A backdoor installed by Remote Ho
Indicator
Malicious IOCs
Indicator
DLLs Loaded by processes
Observable
svchost.exe (PID:816)->notepad.exe(PID:2900)
Observable
Connection alert to IP:192.168.-
Observable
svchost.exe connected to IP address
Observable
Escalated Privileges used by and
Observable
Mutex: ShimCacheMutex
Indicator
Indicator
Unknown Processes initiated by
Indicator
Processes initiated by indicated IP
Indicator
Connection indicated by Honeypot to
Indicator
Unknown connection to IP

Converted STIX IOCs (cont.)


Indicator
ASP.NET State servicesvbw
Indicator
Services started by trojan horse
Indicator
gei33.dll
Indicator
Process created by trojan horse
Indicator
Virus process: skccca.exe
Indicator
Malicious DLLs loaded by trojan
Indicator
Mutex: ShimCacheMutex
Indicator
Mutexes created by the trojan
Indicator
Indicator
Mutex: aspnet_statesusq
TTP
Associated Trojan horse Domains
TTP
TTP
Trojan Win32
STIX
Case Scenario 2.xml
ThreatActor
Trojan Horse backdoor
Observable
trojan:Observable-dbdc-0073-ca1b--
Observable
HTTP GET request
Observable
User-Agent
Observable
Escalated Privileges used by and
Observable
Mutex: ShimCacheMutex
Observable
trojan:observable-d986-4e39-92ed--
Observable
Observable
trojan:observable-8a41-24co-9df2-4-
Observable
trojan:observable-f3a6b-46e-4d8d-4-
Observable
trojan:observable-c5d8-281e-8d25--
Observable
trojan:observable-62b2-1f7b-1fba-4-
Observable
Mutex: aspnet_statesusq
Observable
trojan:observable-6771-4dee-a529--
Observable
trojan:observable-8e83-636f-9941-4-

Future Work

- Extract low-level information programmatically through LibVMI
- Using a network of honeypots

References

- https://www.usenix.org/conference/cset12/workshopprogram/presentation/Lengyel
- http://libvmi.com/docs/gcode-intro.html
- https://www.blackhat.com/docs/us-16/materials/us-16-Zillner-Memory-Forensics-Using-VMI-For-Cloud-Computing.pdf
- http://www.ijser.org/paper/Cloud-Security-using-Honeypot-Systems.html
- http://www.esecurityplanet.com/network-security/how-vmi-can-improve-cloud-security.html
- https://publish.illinois.edu/assured-cloudcomputing/files/2015/05/041915-Virtual-Machine-Instrospection-Overview.pdf

Thanks !