

Spring 21

## Malware Analysis using Rekall

Stuxnet  
Zeus  
SpyEye

# Table Of Contents

<b>Acknowledgement</b>	3
What is Rekal and why do we use it?	3
<b>Downloading Rekall</b>	4
Installing Python 3.6	4
Setting up a Virtual Environment	5
Installing Rekall From Github	6
<b>Using Rekall for Memory Analysis</b>	8
List of Definitions	8
<b>Analyzing Stuxnet using Rekall</b>	9
Connecting stuxnet to Rekal	9
<b>Analyzing Zeus using Rekall</b>	15
Running Zeus on Rekal	15
<b>Analyzing SpyEye Malware using Rekal</b>	21
<b>References</b>	24

# Acknowledgement

Before we start with the report, I just want to make a note that the Rekall Project has not been maintained since 2017 and does not provide full support for python 3.x, so before we start the process of downloading rekall I would definitely recommend downloading python 3.6 for this.

Rekall has many dependencies and the developers made it easier for us to know what they are, and so , if for some reason the download fails, if you look at the error log they will provide a link to the file you need to continue with the installation process.

I would also recommend using a virtual environment to download rekall as that is the easiest way to keep the integrity of your main python dir.

## What is Rekal and why do we use it?

Rekall is an advanced forensic and incident response framework, while it began as a memory forensic framework, it has now evolved into a complete platform, although not currently managed like it was a couple years ago. Rekall implements the most advanced analysis techniques in the field, while still being developed in the open, with a free and open source license.

# Downloading Rekall

Before we start with downloading rekall, I will show how to download python 3.6 for windows, for MacOS and linux, the process will be the same but the only difference will be the file we download.

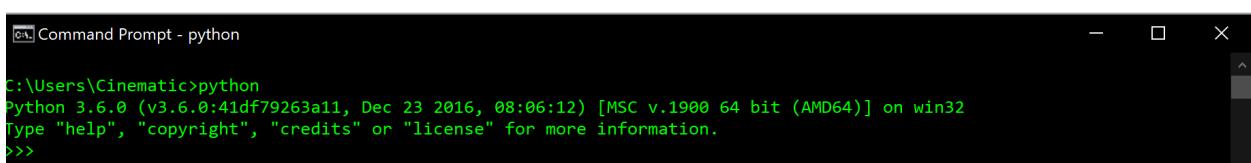
## Installing Python 3.6

Steps to install python on your machine

1. Go to this link : <https://www.python.org/downloads/release/python-360/>
2. Scroll all the way down until you see this:

Files						
Version	Operating System	Description	MD5 Sum	File Size	GPG	
Gzipped source tarball	Source release			22256403	SIG	
XZ compressed source tarball	Source release			16805836	SIG	
Mac OS X 64-bit/32-bit installer	macOS	for Mac OS X 10.6 and later	72acb0175e7622dec7e1b160a43b8c42	27442222	SIG	
Windows help file	Windows			7940890	SIG	
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	0ec0caeeaa75bae5d2771cf619917c71f	6925798	SIG	
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	71c9d30c1110abf7f80a428970ab8ec2	31505640	SIG	
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	25b8b6c93a098dfade3b014630f9508e	1312376	SIG	
Windows x86 embeddable zip file	Windows			6315855	SIG	
Windows x86 executable installer	Windows			30566536	SIG	
Windows x86 web-based installer	Windows			1286984	SIG	

3. Select the file you need, for windows it's usually “**Windows x86-64 executable installer**”.
4. After Downloading it, On the first window you see, there should be an “ADD TO PATH” option, make sure to click it.
5. After finishing the installation process, open command prompt and type “python”.
6. If everything went smoothly this should be the output

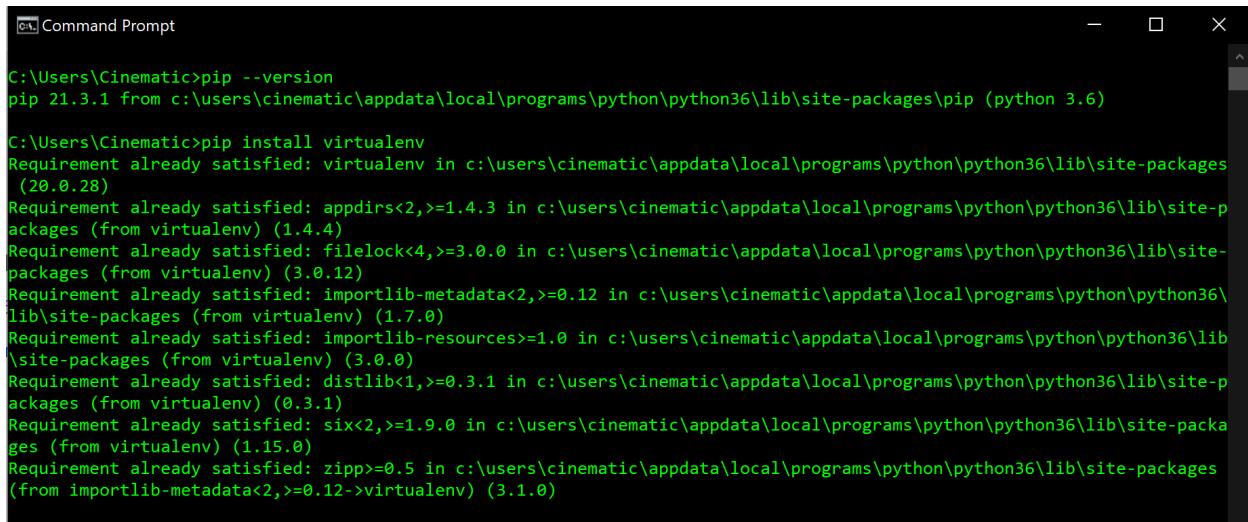


```
C:\Users\Cinematic>python
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Congrats you now have python 3.6 on your machine.

## Setting up a Virtual Environment

After downloading python, we now move on to installing rekall. But before installing rekall we need to set up a virtual environment to download rekall in. To do this open command prompt then “pip install virtualenv”



```
Command Prompt

C:\Users\Cinematic>pip --version
pip 21.3.1 from c:\users\cinematic\appdata\local\programs\python\python36\lib\site-packages\pip (python 3.6)

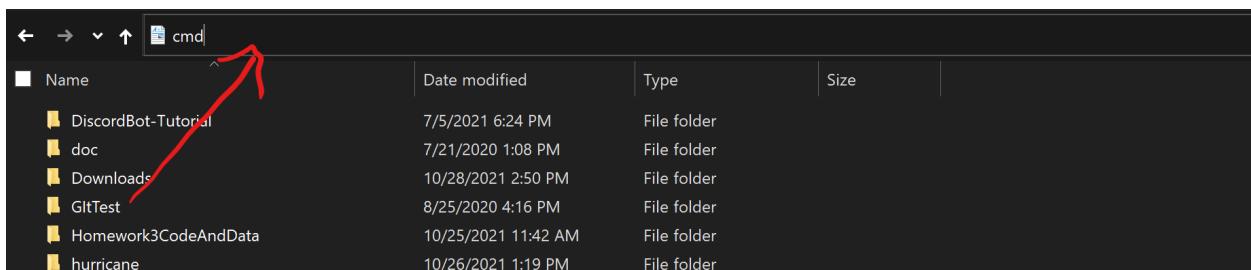
C:\Users\Cinematic>pip install virtualenv
Requirement already satisfied: virtualenv in c:\users\cinematic\appdata\local\programs\python\python36\lib\site-packages
(20.0.28)
Requirement already satisfied: appdirs<2,>=1.4.3 in c:\users\cinematic\appdata\local\programs\python\python36\lib\site-p
ackages (from virtualenv) (1.4.4)
Requirement already satisfied: filelock<4,>=3.0.0 in c:\users\cinematic\appdata\local\programs\python\python36\lib\site-p
ackages (from virtualenv) (3.0.12)
Requirement already satisfied: importlib-metadata<2,>=0.12 in c:\users\cinematic\appdata\local\programs\python\python36\lib
\site-packages (from virtualenv) (1.7.0)
Requirement already satisfied: importlib-resources>=1.0 in c:\users\cinematic\appdata\local\programs\python\python36\lib
\site-packages (from virtualenv) (3.0.0)
Requirement already satisfied: distlib<1,>=0.3.1 in c:\users\cinematic\appdata\local\programs\python\python36\lib\site-p
ackages (from virtualenv) (0.3.1)
Requirement already satisfied: six<2,>=1.9.0 in c:\users\cinematic\appdata\local\programs\python\python36\lib\site-p
ackages (from virtualenv) (1.15.0)
Requirement already satisfied: zipp>=0.5 in c:\users\cinematic\appdata\local\programs\python\python36\lib\site-packages
(from importlib-metadata<2,>=0.12->virtualenv) (3.1.0)
```

Since I already have virtualenv downloaded on my machine the requirements for the pip request have already been satisfied.

Onto the next step, now we will create our project where we will install rekall and setup the virtualenv.

Go to the directory where you would want to create the project.

Click on the address bar and type cmd and hit enter to open the command prompt to the same dir.



Create a new folder here using the mkdir command

Once created, cd into that folder and run the command “virtualenv Dev” , to create a virtualenv called Dev. (This can be called whatever you want)

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Cinematic\Documents>mkdir RekallTutorial
C:\Users\Cinematic\Documents>cd RekallTutorial

C:\Users\Cinematic\Documents\RekallTutorial>virtualenv Dev
created virtual environment CPython3.6.0.final.0-64 in 9445ms
  creator CPython3Windows(dest=C:\Users\Cinematic\Documents\RekallTutorial\Dev, clear=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=C:\Users\Cinematic\AppData\Local\pypa\virtualenv)
    added seed packages: pip==21.2.4, setuptools==58.2.0, wheel==0.37.0
  activators BashActivator,BatchActivator,FishActivator,PowerShellActivator,PythonActivator,XonshActivator

C:\Users\Cinematic\Documents\RekallTutorial>dir
 Volume in drive C is Local Disk
 Volume Serial Number is 6825-5B8A

 Directory of C:\Users\Cinematic\Documents\RekallTutorial

11/09/2021  11:15 AM    <DIR>      .
11/09/2021  11:15 AM    <DIR>      ..
11/09/2021  11:15 AM    <DIR>      Dev
          0 File(s)   0 bytes
          3 Dir(s)  121,454,800,896 bytes free
```

Now cd into Dev and run Scripts\activate to activate the virtualenv

```
C:\Users\Cinematic\Documents\RekallTutorial>cd Dev

C:\Users\Cinematic\Documents\RekallTutorial\Dev>Scripts\activate

(Dev) C:\Users\Cinematic\Documents\RekallTutorial\Dev>
```

Now we are ready to install Rekall

## Installing Rekall From Github

We will now need the rekall api from github, for everything to go like mint in this process you will need to have git. You can download the files [here](#). Git clone it to the project dir.

Once downloaded, cd into Rekall / Recall-core, there should be a setup.py file in there, to execute it just run “python setup.py install”.

If you encounter any errors, pip install whatever module is missing. If there's an executable file missing they will let you know where to download it.

Once Successfully downloaded you should see this

```
Using c:\python27\lib\site-packages\artifacts-20170909-py2.7.egg
Searching for arrow==0.10.0
Best match: arrow 0.10.0
Processing arrow-0.10.0-py2.7.egg
arrow 0.10.0 is already the active version in easy-install.pth

Using c:\python27\lib\site-packages\arrow-0.10.0-py2.7.egg
Searching for acora==2.1
Best match: acora 2.1
Processing acora-2.1-py2.7-win-amd64.egg
acora 2.1 is already the active version in easy-install.pth

Using c:\python27\lib\site-packages\acora-2.1-py2.7-win-amd64.egg
Searching for six==1.16.0
Best match: six 1.16.0
Adding six 1.16.0 to easy-install.pth file

Using c:\python27\lib\site-packages
Finished processing dependencies for rekall-core==1.7.3.dev67

(Dev) C:\Users\Cinematic\Documents\RekallTutorial\rekall\rekall-core>
```

# Using Rekall for Memory Analysis

To run rekall in live mode run “ rekall -v live”, -v is the flag for verbosity.

```
[env] c:\Users\Cinematic\Documents\Rekall>rekall -v live
Webconsole disabled: cannot import name 'webconsole_plugin'
2021-11-11 12:21:39,177:DEBUG:rekall.1:Logging level set to 10
2021-11-11 12:21:39,192:DEBUG:rekall.1:Running plugin (live) with args ()() kwargs ({'profile': None, 'mode': 'Memory', 'driver': None, '1})
Message
-----
Launching live memory analysis
2021-11-11 12:21:39,236:DEBUG:rekall.1:Unable to open \\.\pmem: (2, 'CreateFile', 'The system cannot find the file specified.')
2021-11-11 12:21:39,237:DEBUG:rekall.1:Loading driver from c:\users\cinematic\documents\rekall\rekall-core\resources\WinPmem\winpm
2021-11-11 12:21:39,239:DEBUG:rekall.1:Removing service pmem
2021-11-11 12:21:39,240:DEBUG:rekall.1:Stopping service: pmem
2021-11-11 12:21:39,289:DEBUG:rekall.1:Error stopping service: (1062, 'ControlService', 'The service has not been started.')
2021-11-11 12:21:39,289:DEBUG:rekall.1:Deleting service: pmem
2021-11-11 12:21:39,292:DEBUG:rekall.1:Created service pmem
2021-11-11 12:21:39,338:DEBUG:rekall.1:(31, 'StartService', 'A device attached to the system is not functioning.'): will try to continue
2021-11-11 12:21:39,339:ERROR:rekall.1:Unable to load driver: Unable to open \\.\pmem: (2, 'CreateFile', 'The system cannot find the file
-----
The Rekall Digital Forensic/Incident Response framework 1.7.3.dev67 (Hurricane Ridge).

"We can remember it for you wholesale!"

This program is free software; you can redistribute it and/or modify it under
the terms of the GNU General Public License.

See http://www.rekall-forensic.com/docs/Manual/tutorial.html to get started.
-----
>>> -
```

## List of Definitions

There are a few functions that we need to be familiar with before moving forward.

1. Pslist - The pslist plugin lists all the processes on windows using a variety of methods. Since it is required by all plugins which has process selectors, this plugin will, by default, list processes using all methods
2. Pstree - This plugin displays all known processes in a tree form. This is useful to see which processes launched another process
3. Procinfo - This plugin displays basic info about a process. It takes all the usual process selectors and prints information about the PE file (using peinfo) as well as the process environment strings.
4. Threads - This plugin iterates over all processes and lists all threads in all processes. This is the list walking version of the third scan plugin.
5. Malfind - This command helps find hidden or injected code/DLLs in user mode memory, based on characteristics such as VAD tag and page permissions.
6. Netstat - Shows current network connections.
7. Netscan - Shows current and past connections.

8. Moddump - To extract a kernel module from memory and dump it to disk for analysis we use the moddump command. A regular expression can be specified for the module name to dump.

## Analyzing Stuxnet using Rekall

After installing rekall, we now know some of the functions the library provides. For our first malware we will be using stuxnet, in this first part we will be using functions calls like pslist, pstree, and we will also be using WMI queries to get the list of programs.

What is Stuxnet?

Stuxnet is a computer worm that was originally aimed at Iran's nuclear facilities and has since mutated and spread to other industrial and energy-producing facilities, the original stuxnet malware attack targeted the programmable logic controllers used to automate machine processes. The most recent stuxnet attack struck an unspecified network infrastructure in Iran in October 2018.

## Connecting stuxnet to Rekal

To start stuxnet in rekall we run `rekall --filename stuxnet.vmem`

```
(env) c:\Users\Cinematic\Documents\Rekall>rekall --filename stuxnet.vmem
Webconsole disabled: cannot import name 'webconsole_plugin'

-----
The Rekall Digital Forensic/Incident Response framework 1.7.3.dev67 (Hurricane Ridge).
"We can remember it for you wholesale!"

This program is free software; you can redistribute it and/or modify it under
the terms of the GNU General Public License.

See http://www.rekall-forensic.com/docs/Manual/tutorial.html to get started.

-----
>>>
```

To look at the processes which are running we use pstree

```

>>> pstree()
2021-12-01 12:39:37,162:WARNING:rekall.1:Inventory for repository "http://profiles.rekall-forensics.com" you forgot to create an inventory? You must use the tools/profiles/build_profile_repo.py tool w
2021-12-01 12:39:37,163:WARNING:rekall.1:Repository http://profiles.rekall-forensic.com will be deleted
_EPROCESS
          ppid thd_count hnd_count      create_time
-----
0x823c8830 System (4)           0      59      403 -
. 0x820df020 smss.exe (376)     4      3      19 2010-10-29 17:08:53Z
.. 0x821a2da0 csrss.exe (600)   376     11     395 2010-10-29 17:08:54Z
.. 0x81da5650 winlogon.exe (624) 376     19     570 2010-10-29 17:08:54Z
... 0x82073020 services.exe (668) 624     21     431 2010-10-29 17:08:54Z
.... 0x8205ada0 alg.exe (188)    668     6     107 2010-10-29 17:09:09Z
.... 0x82279998 imapi.exe (756)   668     4     116 2010-10-29 17:11:54Z
.... 0x823315d8 vmauthl.exe (844) 668     1      25 2010-10-29 17:08:55Z
.... 0x81db8da0 svchost.exe (856) 668     17     193 2010-10-29 17:08:55Z
.... 0x81fa5390 wmpiprvse.exe (1872) 856     5     134 2011-06-03 04:25:58Z
.... 0x81c498c8 lsass.exe (868)    668     2      23 2011-06-03 04:26:55Z
.... 0x81e61da0 svchost.exe (940)   668    13     312 2010-10-29 17:08:55Z
.... 0x822843e8 svchost.exe (1032) 668    61    1169 2010-10-29 17:08:55Z
.... 0x822b9a10 wuauctl.exe (976)  1032    3     133 2010-10-29 17:12:03Z
.... 0x820ecc10 wsctnfy.exe (2040) 1032    1      28 2010-10-29 17:11:49Z
.... 0x81e18b28 svchost.exe (1080)  668     5      80 2010-10-29 17:08:55Z
.... 0x81ff7020 svchost.exe (1200)  668    14     197 2010-10-29 17:08:55Z
.... 0x81fee8b0 spoolsv.exe (1412) 668    10     118 2010-10-29 17:08:56Z
.... 0x81e0eda0 jqs.exe (1580)    668     5     148 2010-10-29 17:09:05Z
.... 0x81fe52d0 vmtoolsd.exe (1664) 668     5     284 2010-10-29 17:09:05Z
.... 0x81c0cda0 cmd.exe (968)     1664    0      - 2011-06-03 04:31:35Z
.... 0x81f14938 ipconfig.exe (304)  968     0      - 2011-06-03 04:31:35Z
.... 0x821a0568 VMUpgradeHelper (1816) 668     3      96 2010-10-29 17:09:08Z
.... 0x81c47c00 lsass.exe (1928)   668     4      65 2011-06-03 04:26:55Z
.... 0x81e70020 lsass.exe (680)    624    19     342 2010-10-29 17:08:54Z
0x820ec7e8 explorer.exe (1196)   1728    16     582 2010-10-29 17:11:49Z
. 0x81e86978 TSVNCache.exe (324) 1196    7      54 2010-10-29 17:11:49Z
. 0x81c543a0 Procmon.exe (660)   1196   13     189 2011-06-03 04:25:56Z
. 0x81e6b660 VMwareUser.exe (1356) 1196    9     251 2010-10-29 17:11:50Z
. 0x8210d478 jusched.exe (1712)  1196    1      26 2010-10-29 17:11:50Z
. 0x81fc5da0 VMwareTray.exe (1912) 1196    1      50 2010-10-29 17:11:50Z
Plugin: pstree (PSTree)

```

In the picture above we see 3 lsass.exe processes. One has a parent of “Winlogon.exe” and the other two have a parent of “services.exe”. So now we can go ahead and take a closer look at these 3 processes.

```

>>> processes = tokens(proc_regex="("lsass.exe")")
      Process                         Sid           Comment
-----
0x81e70020 lsass.exe               680 S-1-5-18      Local System
0x81e70020 lsass.exe               680 S-1-5-32-544 Administrators
0x81e70020 lsass.exe               680 S-1-1-0      Everyone
0x81e70020 lsass.exe               680 S-1-5-11      Authenticated Users
0x81c498c8 lsass.exe               868 S-1-5-18      Local System
0x81c498c8 lsass.exe               868 S-1-5-32-544 Administrators
0x81c498c8 lsass.exe               868 S-1-1-0      Everyone
0x81c498c8 lsass.exe               868 S-1-5-11      Authenticated Users
0x81c47c00 lsass.exe               1928 S-1-5-18     Local System
0x81c47c00 lsass.exe               1928 S-1-5-32-544 Administrators
0x81c47c00 lsass.exe               1928 S-1-1-0      Everyone
0x81c47c00 lsass.exe               1928 S-1-5-11      Authenticated Users
>>>

```

As we can see, all the “lsass.exe” processes have the same SID’s

We can now identify the priority of the 3 processes, we know one of them is legit and so that process needs to have a higher priority and the other 2 that we think are suspicious need to have the same priority

```

>>> dlllist([680, 688, 1928])
2021-12-01 12:55:45,689:WARNING:rekall.1:Inventory for repository "http://profiles.rekall-forensic.com"
d you forget to create an inventory? You must use the tools/profiles/build_profile_repo.py tool
2021-12-01 12:55:45,690:WARNING:rekall.1:Repository http://profiles.rekall-forensic.com will be used
    base      size      reason      dll_path
    -----  -----  -----
lsass.exe pid: 680
Command line : C:\WINDOWS\system32\lsass.exe
Service Pack 3
-----
0x1000000 0x6000 65535 C:\WINDOWS\system32\lsass.exe
0x7c90000 0xaf000 65535 C:\WINDOWS\system32\ntdll.dll
0x7c80000 0xf6000 65535 C:\WINDOWS\system32\kernel32.dll
0x77dd000 0xb0000 65535 C:\WINDOWS\system32\ADVAPI32.dll
0x77e7000 0x92000 65535 C:\WINDOWS\system32\RPCRT4.dll
0x77fe000 0x11000 65535 C:\WINDOWS\system32\Secur32.dll
0x7573000 0xb5000 65535 C:\WINDOWS\system32\LSASRV.dll
0x71b2000 0x12000 65535 C:\WINDOWS\system32\MPR.dll
0x7e41000 0x91000 65535 C:\WINDOWS\system32\USER32.dll
0x77f1000 0x49000 65535 C:\WINDOWS\system32\GDI32.dll
0x77b2000 0x12000 65535 C:\WINDOWS\system32\MSASN1.dll
0x77c1000 0x58000 65535 C:\WINDOWS\system32\msvcrt.dll
0x5b86000 0x55000 65535 C:\WINDOWS\system32\NETAPI32.dll
0x767a000 0x13000 65535 C:\WINDOWS\system32\NTDSAPI.dll
0x76f2000 0x27000 65535 C:\WINDOWS\system32\DNSAPI.dll
0x71ab000 0x17000 65535 C:\WINDOWS\system32\WS2_32.dll
0x71aa000 0x8000 65535 C:\WINDOWS\system32\WS2HELP.dll
0x76f6000 0x2c000 65535 C:\WINDOWS\system32\WLDAP32.dll
0x71bf000 0x13000 65535 C:\WINDOWS\system32\SAMLIB.dll
0x7444000 0x6a000 65535 C:\WINDOWS\system32\SAMSRV.dll
0x7679000 0xc000 65535 C:\WINDOWS\system32\cryptdll.dll
0x5cb7000 0x26000 1 C:\WINDOWS\system32\ShimEng.dll
0x6f88000 0x1ca000 1 C:\WINDOWS\AppPatch\AcGeneral.DLL
0x76b4000 0x2d000 2 C:\WINDOWS\system32\WINMM.dll
0x774e000 0x13d000 4 C:\WINDOWS\system32\ole32.dll
0x7712000 0x8b000 2 C:\WINDOWS\system32\OLEAUT32.dll
0x77be000 0x15000 1 C:\WINDOWS\system32\MSACM32.dll
0x77c0000 0x8000 1 C:\WINDOWS\system32\VERSION.dll
0x7c9c000 0x817000 2 C:\WINDOWS\system32\SHELL32.dll
0x77f6000 0x76000 4 C:\WINDOWS\system32\SHLWAPI.dll
0x769c000 0xb4000 15 C:\WINDOWS\system32\USERENV.dll
0x5ad7000 0x38000 3 C:\WINDOWS\system32\UxTheme.dll
0x773d000 0x103000 1 C:\WINDOWS\WinSxS\x86_Microsoft.Windows
0x5d09000 0x9a000 1 C:\WINDOWS\system32\comctl32.dll
0x4d20000 0xe000 1 C:\WINDOWS\system32\msprivs.dll
0x71cf000 0x4c000 2 C:\WINDOWS\system32\kerberos.dll
0x77c7000 0x24000 5 C:\WINDOWS\system32\msv1_0.dll
0x76d6000 0x19000 8 C:\WINDOWS\system32\iphlpapi.dll
0x744b000 0x65000 2 C:\WINDOWS\system32\netlogon.dll
0x767c000 0x2c000 2 C:\WINDOWS\system32\w32time.dll
0x7608000 0x65000 2 C:\WINDOWS\system32\MSVCP60.dll
0x767f000 0x27000 7 C:\WINDOWS\system32\schannel.dll
0x77a8000 0x95000 9 C:\WINDOWS\system32\CRYPT32.dll
0x7438000 0xf000 1 C:\WINDOWS\system32\wdigest.dll
0x68000000 0x36000 1 C:\WINDOWS\system32\rsaenh.dll
0x7441000 0x2f000 1 C:\WINDOWS\system32\scecli.dll
0x7792000 0xf3000 1 C:\WINDOWS\system32\SETUPAPI.dll

```

```

lsass.exe pid: 868
Command line : "C:\WINDOWS\system32\lsass.exe"
Service Pack 3

-----  

0x1000000 0x6000 65535 C:\WINDOWS\system32\lsass.exe  

0x7c900000 0xaf000 65535 C:\WINDOWS\system32\ntdll.dll  

0x7c800000 0xf6000 65535 C:\WINDOWS\system32\kernel32.dll  

0x77dd0000 0x9b000 65535 C:\WINDOWS\system32\ADVAPI32.dll  

0x77e70000 0x92000 65535 C:\WINDOWS\system32\RPCRT4.dll  

0x77fe0000 0x11000 65535 C:\WINDOWS\system32\Secur32.dll  

0x7e410000 0x91000 65535 C:\WINDOWS\system32\USER32.dll  

0x77f10000 0x49000 65535 C:\WINDOWS\system32\GDI32.dll  

-----  

lsass.exe pid: 1928
Command line : "C:\WINDOWS\system32\lsass.exe"
Service Pack 3

-----  

0x1000000 0x6000 65535 C:\WINDOWS\system32\lsass.exe  

0x7c900000 0xaf000 65535 C:\WINDOWS\system32\ntdll.dll  

0x7c800000 0xf6000 65535 C:\WINDOWS\system32\kernel32.dll  

0x77dd0000 0x9b000 65535 C:\WINDOWS\system32\ADVAPI32.dll  

0x77e70000 0x92000 65535 C:\WINDOWS\system32\RPCRT4.dll  

0x77fe0000 0x11000 65535 C:\WINDOWS\system32\Secur32.dll  

0x7e410000 0x91000 65535 C:\WINDOWS\system32\USER32.dll  

0x77f10000 0x49000 65535 C:\WINDOWS\system32\GDI32.dll  

0x8700000 0x138000 1 C:\WINDOWS\system32\KERNEL32.DLL.A  

0x76f20000 0x27000 2 C:\WINDOWS\system32\DNSAPI.dll  

0x77c10000 0x58000 39 C:\WINDOWS\system32\msvcrt.dll  

0x71ab0000 0x17000 10 C:\WINDOWS\system32\WS2_32.dll  

0x71aa0000 0x8000 8 C:\WINDOWS\system32\WS2HELP.dll  

0x76d60000 0x19000 2 C:\WINDOWS\system32\IPHLPIAPI.DLL  

0x5b860000 0x55000 2 C:\WINDOWS\system32\NETAPI32.dll  

0x774e0000 0x13d000 5 C:\WINDOWS\system32\ole32.dll  

0x77120000 0x8b000 4 C:\WINDOWS\system32\OLEAUT32.dll  

0x76bf0000 0xb000 2 C:\WINDOWS\system32\PSAPI.DLL  

0x7c9c0000 0x817000 2 C:\WINDOWS\system32\SHELL32.dll  

0x77f60000 0x76000 8 C:\WINDOWS\system32\SHLWAPI.dll  

0x769c0000 0xb4000 2 C:\WINDOWS\system32\USERENV.dll  

0x77c00000 0x8000 2 C:\WINDOWS\system32\VERSION.dll  

0x771b0000 0xaa000 2 C:\WINDOWS\system32\WININET.dll  

0x77a80000 0x95000 2 C:\WINDOWS\system32\CRYPT32.dll  

0x77b20000 0x12000 2 C:\WINDOWS\system32\MSASN1.dll  

0x71ad0000 0x9000 2 C:\WINDOWS\system32\WSOCK32.dll  

0x773d0000 0x103000 2 C:\WINDOWS\WinSxS\x86_Microsoft.Wi  

0x5d090000 0x9a000 1 C:\WINDOWS\system32\comctl32.dll  

-----  

Plugin: dlllist (WinDllList)

```

Looking at the output we can see that the suspicious processes have a fewer amount of DLLs associated with them. We also see that for both these processes the command line arguments, the backslash is being escaped (\\" instead of \). Now lets look at the handles associated with the 3 PIDs using the handles plugin.

>>> handles(([680, 868, 1928])						
_OBJECT_HEADER _EPROCESS	name	pid	handle	access	obj_type	details
0xe10096c8	0x81e70020 lsass.exe	680	0x4	0xf0003 KeyedEvent	CritSecOutOfMemoryEvent	
0xe16008e0	0x81e70020 lsass.exe	680	0x8	0x3 Directory	KnownDlIs	
0x8225bce0	0x81e70020 lsass.exe	680	0xc	0x100020 File	\Device\HarddiskVolume1\WINDOWS\system32	
0x8205ac18	0x81e70020 lsass.exe	680	0x10	0x10003 Semaphore		
0xe1613960	0x81e70020 lsass.exe	680	0x14	0xf000f Directory	Windows	
0xe1a5a340	0x81e70020 lsass.exe	680	0x18	0x21f0001 Port		
0x81d9ee78	0x81e70020 lsass.exe	680	0x1c	0x10003 Semaphore		
0xe1623520	0x81e70020 lsass.exe	680	0x20	0x2000f Directory	BaseNamedObjects	
0x81ee3968	0x81e70020 lsass.exe	680	0x24	0xf0001 Mutant	SHMLIB_LOG_MUTEX	
0xe19d7b28	0x81e70020 lsass.exe	680	0x28	0x20f003f Key	MACHINE	
0xe1ed81f0	0x81e70020 lsass.exe	680	0x2c	0xf0001 Port		
0x81d9c658	0x81e70020 lsass.exe	680	0x30	0xf016e WindowStation	Service-0x0-3e7\$	
0x822563d8	0x81e70020 lsass.exe	680	0x34	0xf00cf Desktop	Default	
0x81d9c658	0x81e70020 lsass.exe	680	0x38	0xf016e WindowStation	Service-0x0-3e7\$	
0x81db9d38	0x81e70020 lsass.exe	680	0x3c	0xf0003 Semaphore		
0x820e0110	0x81e70020 lsass.exe	680	0x40	0xf0003 Event		
0xe19b5758	0x81e70020 lsass.exe	680	0x44	0x20019 Key	MACHINE\SYSTEM\CONTROLSET001\CONTROL\NETWORKPROVIDER\HORDER	
0x821052d8	0x81e70020 lsass.exe	680	0x48	0x10003 Semaphore		
0x820651d8	0x81e70020 lsass.exe	680	0x4c	0x10003 Semaphore		
0x81ea5b50	0x81e70020 lsass.exe	680	0x50	0xf0003 Event		
0x81db9d08	0x81e70020 lsass.exe	680	0x54	0xf0003 Event		
0x81eb4f48	0x81e70020 lsass.exe	680	0x58	0x10003 Semaphore		
0x82339e10	0x81e70020 lsass.exe	680	0x5c	0x10003 Semaphore		
0xe1a66240	0x81e70020 lsass.exe	680	0x60	0x20019 Key	MACHINE\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\DRIVERS32	
0x81ef5d08	0x81e70020 lsass.exe	680	0x64	0xf0003 Event	DINPUTWINMM	
0x8225bae8	0x81e70020 lsass.exe	680	0x68	0x10001 File	\Device\ksecdd	
0x821052a8	0x81e70020 lsass.exe	680	0x6c	0xf0003 Event		
0x820651a8	0x81e70020 lsass.exe	680	0x70	0xf0003 Event		
0xe1a3b1c8	0x81e70020 lsass.exe	680	0x74	0x20019 Key	MACHINE\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\DRIVERS32	
0x82082cf30	0x81e70020 lsass.exe	680	0x78	0xf0003 Semaphore	shell.(A48F1A32-A340-11D1-BC6B-00A0C90312E1)	
0x8209a588	0x81e70020 lsass.exe	680	0x7c	0xf0003 Event	userenv: User Profile setup event	
0x8225b8f0	0x81e70020 lsass.exe	680	0x80	0x100020 File	\Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windows.Commo	
0x81eddc00	0x81c498c8 lsass.exe	868	0x744	0x1f03ff Thread	TID 592 PID 940	
0x82083a48	0x81c498c8 lsass.exe	868	0x7ac	0x1f0003 IoCompletion		
0x81c42790	0x81c498c8 lsass.exe	868	0x7b0	0x1f0003 IoCompletion		
0x82083a48	0x81c498c8 lsass.exe	868	0x7b4	0x1f0003 IoCompletion		
0x822bbd90	0x81c498c8 lsass.exe	868	0x7b8	0x1f03ff Thread	TID 1884 PID 868	
0x81f9ead0	0x81c498c8 lsass.exe	868	0x7bc	0x1f0003 Event		
0x81c36ee0	0x81c498c8 lsass.exe	868	0x7c0	0x1f0003 Event		
0x81c8ede8	0x81c498c8 lsass.exe	868	0x7c4	0x1f0003 Event		
0x81f6cfd8	0x81c498c8 lsass.exe	868	0x7c8	0x1f0003 Event		
0x81e61d88	0x81c498c8 lsass.exe	868	0x7cc	0x1f0fff Process	svchost.exe(940)	
0x822bbd90	0x81c498c8 lsass.exe	868	0x7d0	0x1f03ff Thread	TID 1884 PID 868	
0x81d9c658	0x81c498c8 lsass.exe	868	0x7d4	0xf016e WindowStation	Service-0x0-3e7\$	
0x822563d8	0x81c498c8 lsass.exe	868	0x7d8	0xf00cf Desktop	Default	
0x81d9c658	0x81c498c8 lsass.exe	868	0x7dc	0xf016e WindowStation	Service-0x0-3e7\$	
0x821a4660	0x81c498c8 lsass.exe	868	0x7e0	0x21f0003 Event		
0xe2a6e818	0x81c498c8 lsass.exe	868	0x7e4	0x20f003f Key	MACHINE	
0x81c68440	0x81c498c8 lsass.exe	868	0x7e8	0x10003 Semaphore		
0xe2b19ac8	0x81c498c8 lsass.exe	868	0x7ec	0x21f0001 Port		
0xe1613960	0x81c498c8 lsass.exe	868	0x7f0	0xf000f Directory	Windows	
0x81fb0a70	0x81c498c8 lsass.exe	868	0x7f4	0x10003 Semaphore		
0xe16008e0	0x81c498c8 lsass.exe	868	0x7f8	0x3 Directory	KnownDlIs	
0xe10096c8	0x81c498c8 lsass.exe	868	0x7fc	0xf0003 KeyedEvent	CritSecOutOfMemoryEvent	
0x8225b6f8	0x81c47c00 lsass.exe	1928	0xc	0x100020 File	\Device\HarddiskVolume1\WINDOWS\system32	
0x81be6378	0x81c47c00 lsass.exe	1928	0x700	0x1f0003 Event	WkssvcShutdownEvent2	
0x81fed2a8	0x81c47c00 lsass.exe	1928	0x704	0x1f03ff Thread	TID 780 PID 1928	
0xe2995c10	0x81c47c00 lsass.exe	1928	0x708	0x1f0001 Port		
0x81f5f380	0x81c47c00 lsass.exe	1928	0x70c	0x1f0003 Event		
0x81e22d90	0x81c47c00 lsass.exe	1928	0x710	0x1f03ff Thread	TID 404 PID 1928	
0x81f60260	0x81c47c00 lsass.exe	1928	0x714	0x1f0003 Event		
0x829fa838	0x81c47c00 lsass.exe	1928	0x718	0x1f0007 Section	{4A9A9FA4-5292-4607-B3CB-EE6A87A008A3}	
0x81fc3c00	0x81c47c00 lsass.exe	1928	0x71c	0x1f0001 Mutant	{5EC171BB-F130-4a19-B782-B6E655E091B2}	
0x81c73940	0x81c47c00 lsass.exe	1928	0x720	0x1f0003 Event		
0x81f402a0	0x81c47c00 lsass.exe	1928	0x724	0x1f0003 Event	{CAA6BD26-6C7B-4af0-95E2-53DE46FDDF26}	
0x81f64378	0x81c47c00 lsass.exe	1928	0x728	0x1f0001 Mutant		
0x81f63b60	0x81c47c00 lsass.exe	1928	0x72c	0x1f0001 Mutant		
0x81e22d90	0x81c47c00 lsass.exe	1928	0x730	0x1f03ff Thread	TID 404 PID 1928	
0x81f8cdf0	0x81c47c00 lsass.exe	1928	0x734	0x1f0001 Mutant		
0x81f8ce30	0x81c47c00 lsass.exe	1928	0x738	0x1f0001 Mutant		
0x81f9d298	0x81c47c00 lsass.exe	1928	0x73c	0x1f0001 Mutant		
0x81f90478	0x81c47c00 lsass.exe	1928	0x740	0x1f0001 Mutant		
0x81f9cbe8	0x81c47c00 lsass.exe	1928	0x744	0x1f0001 Mutant		
0x81c685c0	0x81c47c00 lsass.exe	1928	0x748	0x1f0001 Mutant		
0x821318b8	0x81c47c00 lsass.exe	1928	0x74c	0x1f0001 Mutant		
0x81fa2b90	0x81c47c00 lsass.exe	1928	0x750	0x1f0001 Mutant		
0x81ea9328	0x81c47c00 lsass.exe	1928	0x754	0x1f0001 Mutant		
0xe247ecd0	0x81c47c00 lsass.exe	1928	0x758	0x2001f Key	USER\.DEFAULT\SOFTWARE\MICROSOFT\WINDOWS\	
0x82062728	0x81c47c00 lsass.exe	1928	0x75c	0x100020 File	\Device\HarddiskVolume1\WINDOWS\WinSxS\x8	

When we look at the handles we see that PID 680 has a significant amount of handles while 868 and 1928 does not have much, 680 has been truncated to fit this page.

Now lets run malfind against these PIDs to see if there is anything suspicious with the processes.

```
>>> malfind([680, 868, 1928])
*****
f pid 680
Process: lsass.exe Pid: 868 Address: 0x80000
Vad Tag: Vad Protection: EXECUTE_READWRITE
Protection: 6

0x80000 4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00 MZ..... vad_0x80000
0x80010 b8 00 00 00 00 00 00 40 00 00 00 00 00 00 00 .....@.....
0x80020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x80030 00 00 00 00 00 00 00 00 00 00 00 00 00 08 01 00 00 .....
```

```
Process: lsass.exe Pid: 1928 Address: 0x80000
Vad Tag: Vad Protection: EXECUTE_READWRITE
Protection: 6

0x80000 4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00 MZ..... vad_0x80000
0x80010 b8 00 00 00 00 00 00 40 00 00 00 00 00 00 00 .....@.....
0x80020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x80030 00 00 00 00 00 00 00 00 00 00 00 00 00 08 01 00 00 .....
```

We we ran the command we can see malfind return results for the 2 suspicious processes and both of these offsets start with 0x80000.

Using the ldrmodules plugin we can gain a bit more insight into the 2 suspicious processes

```
>>> ldrmodules(1928)
    base   in_load in_init in_mem mapped
-----
-----
0x81c47c00 lsass.exe 1928
-----
    0x80000 False  False  False
0x7c900000 True   True   True   C:\WINDOWS\system32\ntdll.dll
0x10000000 True   False  True
0x6f0000 False  False  False
0x680000 False  False  False
0x870000 True   True   True
0x7c800000 True   True   True   C:\WINDOWS\system32\kernel32.dll
0x77dd0000 True   True   True   C:\WINDOWS\system32\advapi32.dll
0x76f20000 True   True   True   C:\WINDOWS\system32\dnsapi.dll
0x71ab0000 True   True   True   C:\WINDOWS\system32\ws2_32.dll
0x71aa0000 True   True   True   C:\WINDOWS\system32\ws2help.dll
```

From here we can see, there are no paths information for the 3 entries for module 1928, also the offset matches the one we saw in malfind and that is why the dll is hidden as it may be unlinked from one or more lists within the PEB

Using the plugins given we can see how malwares react with rekal

# Analyzing Zeus using Rekall

What is the Zeus malware?

Zeus, ZeuS, or Zbot is a Trojan horse malware package that runs on version of Microsoft Windows. While it can be used to carry out many malicious attacks it is often used to steal banking info by man-in-the-browser keystroke logging and form grabbing. It is spread through drive-by downloads and phishing schemes.

## Running Zeus on Rekal

Running Zeus on Rekal is the same as stuxnet we run `rekal -filename zeus.vmem`

```
(env) c:\Users\Cinematic\Documents\Rekall>rekall --filename zeus2x4.vmem
Webconsole disabled: cannot import name 'webconsole_plugin'

-----
The Rekall Digital Forensic/Incident Response framework 1.7.3.dev67 (Hurricane Ridge).

"We can remember it for you wholesale!"

This program is free software; you can redistribute it and/or modify it under
the terms of the GNU General Public License.

See http://www.rekall-forensic.com/docs/Manual/tutorial.html to get started.

-----
>>> pstree()
2021-12-09 20:04:22,949:WARNING:rekall.1:Inventory for repository "http://profiles.rekall-fore
d you forget to create an inventory? You must use the tools/profiles/build_profile_repo.py too
2021-12-09 20:04:22,949:WARNING:rekall.1:Repository http://profiles.rekall-forensic.com will b
_EPROCESS
          ppid   thd_count  hnd_count      create_time
-----
0x823c8a00 System (4)           0       57      671 -
. 0x82292da0 smss.exe (596)     4       3       19 2010-09-02 12:25:18Z
.. 0x821f2978 csrss.exe (668)   596      14      471 2010-09-02 12:25:21Z
... 0x822c09f8 winlogon.exe (692) 596      21      588 2010-09-02 12:25:22Z
.... 0x821a5da0 services.exe (744) 692      15      279 2010-09-02 12:25:22Z
..... 0x82129370 svchost.exe (364) 744      4       88 2010-09-02 12:25:33Z
..... 0x82189530 prl_tools_servi (436) 744      3       78 2010-09-02 12:25:36Z
..... 0x82086798 prl_tools.exe (632) 436      9       107 2010-09-02 12:25:36Z
..... 0x82089558 jqs.exe (472)     744      5       146 2010-09-02 12:25:33Z
..... 0x8208abf0 sqlservr.exe (488) 744      25      306 2010-09-02 12:25:33Z
..... 0x82077da0 coherence.exe (572) 744      4       51 2010-09-02 12:25:36Z
..... 0x821aa7e8 sqlwriter.exe (660) 744      4       84 2010-09-02 12:25:36Z
..... 0x82150b90 svchost.exe (912)   744      20      202 2010-09-02 12:25:22Z
..... 0x822c8bf8 svchost.exe (992)   744      10      277 2010-09-02 12:25:22Z
..... 0x82151da0 svchost.exe (1084) 744      58      1327 2010-09-02 12:25:22Z
..... 0x8205dda0 wuauctl.exe (940) 1084      4       126 2010-09-02 12:26:40Z
..... 0x8213dda0 wsctfy.exe (2180) 1084      3       48 2010-09-02 12:25:41Z
..... 0x82311648 rundll32.exe (3768) 1084      1       53 2010-09-09 19:56:33Z
..... 0x81eab2f8 wuauctl.exe (3984) 1084      8       325 2010-09-09 19:52:45Z
..... 0x821521b0 svchost.exe (1140) 744      6       81 2010-09-02 12:25:22Z
..... 0x8214f488 svchost.exe (1192) 744      13      175 2010-09-02 12:25:23Z
..... 0x8221e278 iscsie.exe (1436) 744      6       78 2010-09-02 12:25:24Z
..... 0x82095500 spoolsv.exe (1616) 744      13      140 2010-09-02 12:25:24Z
..... 0x81e8a368 alg.exe (2588)    744      6       107 2010-09-02 12:25:44Z
.... 0x822c8798 lsass.exe (756)   692      24      437 2010-09-02 12:25:22Z
  0x821b2020 explorer.exe (1752) 1720      22      520 2010-09-02 12:25:25Z
. 0x822b96c0 SharedIntApp.ex (1900) 1752      3       75 2010-09-02 12:25:25Z
. 0x820ee580 prl_cc.exe (1908)   1752      14      133 2010-09-02 12:25:25Z
. 0x82282380 ImmunityDebugge (1932) 1752      2       86 2010-09-08 19:23:02Z
.. 0x8223c020 vaelh.exe (952)    1932      2       40 2010-09-08 19:23:02Z
.. 0x8212ada0 jusched.exe (1936) 1752      1       43 2010-09-02 12:25:26Z
.. 0x82066478 ImmunityDebugge (2404) 1752      2       85 2010-09-09 19:56:19Z
.. 0x81f4bb28 b98679df6defbb3 (3772) 2404      1       46 2010-09-09 19:56:19Z
... 0x81e87da0 ihah.exe (3276)   3772      1       45 2010-09-09 19:56:32Z
. 0x82001ad0 ImmunityDebugge (2972) 1752      2       87 2010-09-08 19:14:36Z
.. 0x8207bda0 nifek_locked.ex (2204) 2972      2       38 2010-09-08 19:14:36Z
. 0x81ffb6d8 ImmunityDebugge (3788) 1752      2       103 2010-09-08 22:39:40Z
```

As you can see when we were dealing with stuxnet we got 3 instances of the lsass.exe file, now we do not. But when dealing with Zeus we need to look at the connections that it is making so to look at the connections my files are making I simply run the `connscan()` plugin

```
>>> connscan()
tcpip/GUID/F43034C3D7D24BB5A88C36C1672CE1541 matched offset 0x31553+0xb2ef3000= offset_p
-----
0x20f5410 10.211.55.5:1427      65.54.81.89:80          1084
0x2125008 10.211.55.5:1423      207.46.21.123:80      1084
0x22ace08 10.211.55.5:1432      193.43.134.14:80      1752
Plugin: connscan (ConnScan)
```

Looking at the pids and comparing them to the pstree output above we can see that the ip 192.43.134.14 is connected to the internet through port 80 and had the 1752 pid which does not correspond to a web browser so that is a red flag, when I look up the ip address on [ipvoid.com](http://ipvoid.com) I get the following result.

Analysis Date	2021-11-03 21:12:24
Elapsed Time	3 seconds
Blacklist Status	<b>BLACKLISTED 1/115</b>
IP Address	<b>193.43.134.14</b> <a href="#">Find Sites</a>   <a href="#">IP Whois</a>
Reverse DNS	Unknown
ASN	Unknown
ASN Owner	Unknown
ISP	Unknown
Continent	Unknown
Country Code	Unknown
Latitude / Longitude	Unknown
City	Unknown
Region	Unknown

As you can see the ip is blacklisted. Sometimes trojans adds a registry key to make sure that it is running every time the computer is restarted.

To check if the malware has added itself to the register we run

```
printkey("Microsoft\Windows NT\CurrentVersion\Winlogon")
```

```

values.
0xcd5521fc REG_DWORD AutoRestartShell : (S) 1
0xcd552224 REG_SZ DefaultDomainName : (S) JASONRESACC69
0xcd552274 REG_SZ DefaultUserName : (S) Administrator
0xcd5522bc REG_SZ LegalNoticeCaption : (S)
0xcd552304 REG_SZ LegalNoticeText : (S)
0xcd55232c REG_SZ PowerdownAfterShutdown : (S) 0
0xcd55237c REG_SZ ReportBootOk : (S) 1
0xcd5523a4 REG_SZ Shell : (S) Explorer.exe
0xcd55240c REG_SZ ShutdownWithoutLogon : (S) 0
0xcd55235c REG_SZ System : (S)
0xcd5523e4 REG_SZ Userinit : (S) C:\WINDOWS\system32\userinit.exe,
0xcd5524b4 REG_SZ VmApplet : (S) rundll32 shell32,Control_RunDLL "sysdm.cpl"
0xcd55243c REG_DWORD SfcQuota : (S) 4294967295
0xcd55256c REG_SZ allocatedcdroms : (S) 0
0xcd552534 REG_SZ allocatedasd : (S) 0
0xcd5525d4 REG_SZ allocatefloppies : (S) 0
0xcd552594 REG_SZ cachedlogonscount : (S) 10
0xcd552644 REG_DWORD forceunlocklogon : (S) 0
0xcd5525fc REG_DWORD passwordexpirywarning : (S) 14
0xcd5526bc REG_SZ scremoveoption : (S) 0
0xcd55266c REG_DWORD AllowMultipleTSSessions : (S) 1
0xcd55269c REG_EXPAND_SZ UIHost : (S) logonui.exe
0xcd5526e4 REG_DWORD LogonType : (S) 1
0xcd55270c REG_SZ Background : (S) 0 0 0
0xcd55275c REG_SZ AutoAdminLogon : (S) 1
0xcd5527ac REG_SZ DebugServerCommand : (S) no
0xcd5527dc REG_DWORD SFCDisable : (S) 0
0xcd56991c REG_SZ WinStationsDisabled : (S) 0
0xcd56b144 REG_DWORD HibernationPreviouslyEnabled : (S) 1
0xcd56a2dc REG_DWORD ShowLogonOptions : (S) 0
0xcd56a304 REG_SZ AltDefaultUserName : (S) Administrator
0xcd56b1cc REG_SZ AltDefaultDomainName : (S) JASONRESACC69
Plugin: printkey (PrintKey)

```

Looking at the output, if you notice when we only have one value to print out it doesn't add a comma at the end but for REG\_SZ Userinit there is a comma after C:\Windows\system32\userinit.exe. It seems like there is something wrong with the userinit register. Lets dive deeper

We can now look for hidden or injected code in the user mode memory dump and see if the firewall was shut down.

To do this we run

```
printkey("ControlSet001\Services\SharedAccess\Parameters\FirewallPolicy\StandardProfile")
```

After running this we can see that the firewall has no value meaning it was disabled. This is evident that the computer was infected with the Zeus Trojan

```
>>> printkey("ControlSet001\Services\SharedAccess\Parameters\FirewallPolicy\StandardProfile")
Legend: (S) = Stable   (V) = Volatile

Printing $$$PROTO.HIV/ControlSet001/Services/SharedAccess/Parameters/FirewallPo-----
Registry: C:\WINDOWS\system32\config\system @ 0xe1035b60
Key name: StandardProfile (S) @ 0xd74a27ac
Last updated: 2009-04-04 02:24:53Z

Subkeys:
(S) AuthorizedApplications

Values:
```

# Analyzing SpyEye Malware using Rekal

What is the SpyEye Malware?

SpyEye is a malware program that attacks user running Google Chrome, Opera Firefox and Internet Explorer on Microsoft Windows operating Systems. This malware uses keystroke logging and form grabbing to steal user credentials for malicious use.

As usual we start with running the SpyEye malware with rekall and to do that we run  
Rekal -filename spyeye.vmem

Once We have spyeye running we can use pstree to list the current services running

_EPROCESS	ppid	thd_count	hnd_count	create_time
0x825c8830 System (4)	0	58	387	-
. 0x823fe020 smss.exe (572)	4	3	19	2010-11-11 22:02:08Z
.. 0x82503220 csrss.exe (636)	572	13	399	2010-11-11 22:02:13Z
... 0x81f4c550 winlogon.exe (660)	572	21	596	2010-11-11 22:02:14Z
... . 0x8207d5f0 services.exe (704)	660	17	285	2010-11-11 22:02:15Z
.... . 0x8230c5f8 vmacthlp.exe (872)	704	2	26	2010-11-11 22:02:16Z
.... . 0x8226cda0 svchost.exe (904)	704	16	191	2010-11-11 22:02:16Z
.... . 0x823f2020 svchost.exe (972)	704	9	264	2010-11-11 22:02:17Z
.... . 0x824578b0 imapi.exe (1040)	704	5	114	2010-11-11 22:03:54Z
.... . 0x822a0758 svchost.exe (1068)	704	58	1256	2010-11-11 22:02:17Z
..... . 0x8236d7a0 wuauctl.exe (536)	1068	4	107	2010-11-11 22:03:33Z
..... . 0x82389020 wscntfy.exe (2772)	1068	2	29	2010-11-11 22:03:56Z
.... . 0x81f4b020 svchost.exe (1108)	704	7	82	2010-11-11 22:02:17Z
.... . 0x82406da0 svchost.exe (1232)	704	13	169	2010-11-11 22:02:18Z
.... . 0x82436a48 spoolsv.exe (1456)	704	12	121	2010-11-11 22:02:19Z
.... . 0x82067858 svchost.exe (1540)	704	6	95	2010-11-11 22:02:26Z
.... . 0x82072660 jqs.exe (1612)	704	6	149	2010-11-11 22:02:27Z
.... . 0x82284b80 vmtoolsd.exe (1816)	704	6	268	2010-11-11 22:02:30Z
.... . 0x822e69f8 VMUpgradeHelper (1872)	704	4	100	2010-11-11 22:02:30Z
.... . 0x82458020 alg.exe (2108)	704	7	107	2010-11-11 22:03:54Z
.... . 0x81f7a708 WPFFontCache_v0 (3084)	704	7	70	2010-11-11 22:05:04Z
.... . 0x824264c0 lsass.exe (716)	660	20	356	2010-11-11 22:02:15Z
0x823e32f8 explorer.exe (1008)	680	15	468	2010-11-11 22:02:55Z
. 0x81ec2020 TSVNCache.exe (1252)	1008	9	58	2010-11-11 22:02:58Z
. 0x81ebd300 VMwareTray.exe (1484)	1008	2	51	2010-11-11 22:03:00Z
. 0x82159958 VMwareUser.exe (1588)	1008	7	230	2010-11-11 22:03:00Z
. 0x8214ba18 jusched.exe (1672)	1008	2	97	2010-11-11 22:03:00Z
.. 0x81f5e020 jucheck.exe (3892)	1672	3	105	2010-11-11 22:08:01Z
. 0x82226b48 cleansweep.exe (2268)	1008	0	-	2011-01-06 14:36:52Z
. 0x820bd760 gmer.exe (2728)	1008	2	33	2011-01-06 14:37:41Z

Nothing out of the ordinary except Explorer.exe is a separate service running on its own, we can now look at what it is doing, since we are expecting the spyeye to latch itself to the web browser and not send any requests back when I run connscan, it should return 2 ip addresses.

```

>>> connscan()
  offset_p      local_net_address      remote_net_address      pid
-----
0x1eacc00 192.168.16.129:1039      65.55.185.26:443      1068
0x1fd3170 192.168.16.129:1040      207.46.21.58:80      1068
Plugin: connscan (ConnScan)
>>>

```

As expected no malicious calls are being made from the computer, since it is a keylogger when I run malfind for the explorer it should return null results if there is no malware

```

>>> malfind(1008)
*****
Process: explorer.exe Pid: 1008 Address: 0x26b0000
Vad Tag: VadS Protection: EXECUTE_READWRITE
CommitCharge: 1, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x26b0000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... vad_0x26b0000
0x26b0010 00 00 6b 02 00 00 00 00 00 00 00 00 00 00 00 00 ..k.....
0x26b0020 10 00 6b 02 00 00 00 00 00 00 00 00 00 00 00 00 ..k.....
0x26b0030 20 00 6b 02 00 00 00 00 00 00 00 00 00 00 00 00 ..k.....
----- vad_0x26b0000 -----: 0x26b0000
0x26b0000 0x0 0000      add byte ptr [eax], al
0x26b0002 0x2 0000      add byte ptr [eax], al
0x26b0004 0x4 0000      add byte ptr [eax], al
0x26b0006 0x6 0000      add byte ptr [eax], al
0x26b0008 0x8 0000      add byte ptr [eax], al
0x26b000a 0xa 0000      add byte ptr [eax], al
0x26b000c 0xc 0000      add byte ptr [eax], al
0x26b000e 0xe 0000      add byte ptr [eax], al
0x26b0010 0x10 0000     add byte ptr [eax], al
0xea000000 4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00 MZ..... vad_0xea00000
0xea00010 b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 .....@.....
0xea00020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... .
0xea00030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 f8 00 00 00 ..... .

----- vad_0xea00000 -----
0xea00000 0x0 4d      dec ebp
0xea00001 0x1 5a      pop edx
0xea00002 0x2 90      nop
0xea00003 0x3 0003     add byte ptr [ebx], al
0xea00005 0x5 0000     add byte ptr [eax], al
0xea00007 0x7 000400   add byte ptr [eax + eax], al
0xea0000a 0xa 0000     add byte ptr [eax], al
0xea0000c 0xc ff       db 0xff

```

```
Process: explorer.exe Pid: 1008 Address: 0xea50000
Vad Tag: VadS Protection: EXECUTE_READWRITE
CommitCharge: 58, MemCommit: 1, PrivateMemory: 1, Protection: 6

0xea50000 4d 5a 90 00 03 00 00 00 00 04 00 00 00 ff ff 00 00 MZ..... vad_0xea50000
0xea50010 b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 .....@.....
0xea50020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0xea50030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 ......

----- vad_0xea50000 -----: 0xea50000
0xea50000 0x0 4d dec ebp
0xea50001 0x1 5a pop edx
0xea50002 0x2 90 nop
0xea50003 0x3 0003 add byte ptr [ebx], al
0xea50005 0x5 0000 add byte ptr [eax], al
0xea50007 0x7 000400 add byte ptr [eax + eax], al
0xea5000a 0xa 0000 add byte ptr [eax], al
0xea5000c 0xc ff db 0xff
0xea5000d 0xd ff00 inc dword ptr [eax]
```

```
Process: explorer.exe Pid: 1008 Address: 0xeab0000
Vad Tag: VadS Protection: EXECUTE_READWRITE
CommitCharge: 54, MemCommit: 1, PrivateMemory: 1, Protection: 6

0xeab0000 4d 5a 90 00 03 00 00 00 00 04 00 00 00 ff ff 00 00 MZ..... vad_0xeab0000
0xeab0010 b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 .....@.....
0xeab0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0xeab0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 ......

----- vad_0xeab0000 -----: 0xeab0000
```

```
0xeab0000 0x0 4d dec ebp
0xeab0001 0x1 5a pop edx
0xeab0002 0x2 90 nop
0xeab0003 0x3 0003 add byte ptr [ebx], al
0xeab0005 0x5 0000 add byte ptr [eax], al
0xeab0007 0x7 000400 add byte ptr [eax + eax], al
0xeab000a 0xa 0000 add byte ptr [eax], al
0xeab000c 0xc ff db 0xff
0xeab000d 0xd ff00 inc dword ptr [eax]
0xeab00f 0xf 00b800000000 add byte ptr [eax], bh
0xeab0015 0x15 0000 add byte ptr [eax], al
0xeab0017 0x17 004000 add byte ptr [eax], al
0xeab001a 0x1a 0000 add byte ptr [eax], al
```

If we look closely all the processes start with 0xeba we saw something similar when we were dealing with stuxnet. This means the malware has attached itself to the Web Browser.

## References

- <http://www.rekall-forensic.com/#:~:text=Rekall%20is%20an%20advanced%20forensic,free%20and%20open%20source%20license>.
- <https://www.sans.org/posters/rekall-cheat-sheet/>
- <https://holdmybeersecurity.com/2017/07/29/rekall-memory-analysis-framework-for-windows-linux-and-mac-osx/>
- <https://isc.sans.edu/forums/diary/Live+memory+analysis+using+Rekall/24454/>
- <https://www.oreilly.com/learning-paths/learning-path-forensic/9781491998496/9781492029144-video317695/>