RASTREA2R

Collecting & Hunting for IOCs with *gusto* and *style!*

Author: Ismael Valenzuela (@aboutsecurity)

Contributors: Ryan O'Connor (@_remixed) and Rob Gresham (@rwgresham)

https://github.com/aboutsecurity/rastrea2r

{ THE CHALLENGE};

- Working as threat researcher for a large healthcare agency
- +50,000 endpoints centrally managed by McAfee ePO
- 700 events/second (17 million events/day)
- Little or no forensics in place. Triage, evidence collection, and analysis was manual and reactive
- Results of the triage process were difficult to validate and process
- Initial vector of compromise was difficult to determine
- Attacks were difficult to classify based on impact to the business
- Containment was the same on all cases (AV scan and reimage)
- No lessons learned

{ Prototyping a solution: Quick & Dirty };

1. Collect host artifacts required for triage (event analysis) from all endpoints
2. Without deploying additional agents to the endpoint
3. Cross-platform
4. Must support automated, remote response
5. With minimal footprint, adhering to forensically sound procedures
6. Triage using both native and third party tools
7. Support proactive hunting for any given IoCs
8. Extensible to integrate with well known IR and forensics open source frameworks

{ AUTOMATION == PYTHON };

1. Tons of libraries created related to security and computer forensics
2. Ideal to create quick prototypes of applications or algorithms
3. Supports the design of RESTful interfaces (client-server services)
4. Python lets me focus on concepts rather than code
5. With Pyinstaller I can freeze (package) Python programs into small stand-alone executables
6. Also… it was named after the great Monty Python!

{ What do we need to automate? };

- Malware doesn't exist in a vacuum:
  - They need to run
  - They need to communicate
  - They need to be persistent
- How do we find "evidence"?
- Traditional forensics techniques are too time consuming
- Triaging can be used to identify relevant evidence quickly and guide the IR process
- Live forensic analysis
  - Volatile data
    - Processes
    - Network connections, etc.
  - Non volatile data
    - Program execution (prefetch)
    - Autorun locations
    - Master File Table (MFT), etc.
- Dump and examine memory
- Scan with Indicators of Compromise (IOCs)

{ HUNTING & SMART INCIDENT RESPONSE };

{ HUNTING WITH YARA };

- "The pattern matching Swiss knife for malware researchers (and everyone else!)"
  - http://plusvic.github.io/yara
- Pattern matching:
  - Strings, regular expressions and binary patterns (hex strings)
- Classification:
  - On input: combination of strings & logic, stored in a YARA rule
  - On output: tags, metadata
- Can be integrated in your Python projects (bindings)
- Great repository: http://yararules.com/

@YaraRules

{ SO HOW DO WE DO ALL THIS? – **RASTREA2R** };

- **RASTREA2R** (PRONOUNCED *RASTREADOR*):
  - HTTPS://GITHUB.COM/ABOUTSECURITY/RASTREA2R (OPENSOURCE!)
  - COMMAND LINE TOOL (COZ COMMAND LINE IS SEXY!)
  - PYTHON / MULTIPLATFORM (WIN32/64, LINUX AND OSX)
  - USES A REST API TO REPORT **YARA** SCANS
  - WRAPPER TO SYSINTERNAL, SYSTEM COMMAND AND 3RD PARTY TOOLS
  - EASY TO INTEGRATE WITH MCAFEE EPO (BUT ALSO DISTRIBUTABLE VIA SSCM, ETC.)
  - PACKAGED BINARIES AVAILABLE ON GITHUB

{ CURRENT FUNCTIONALITY IN RASTREA2R V0.7 };

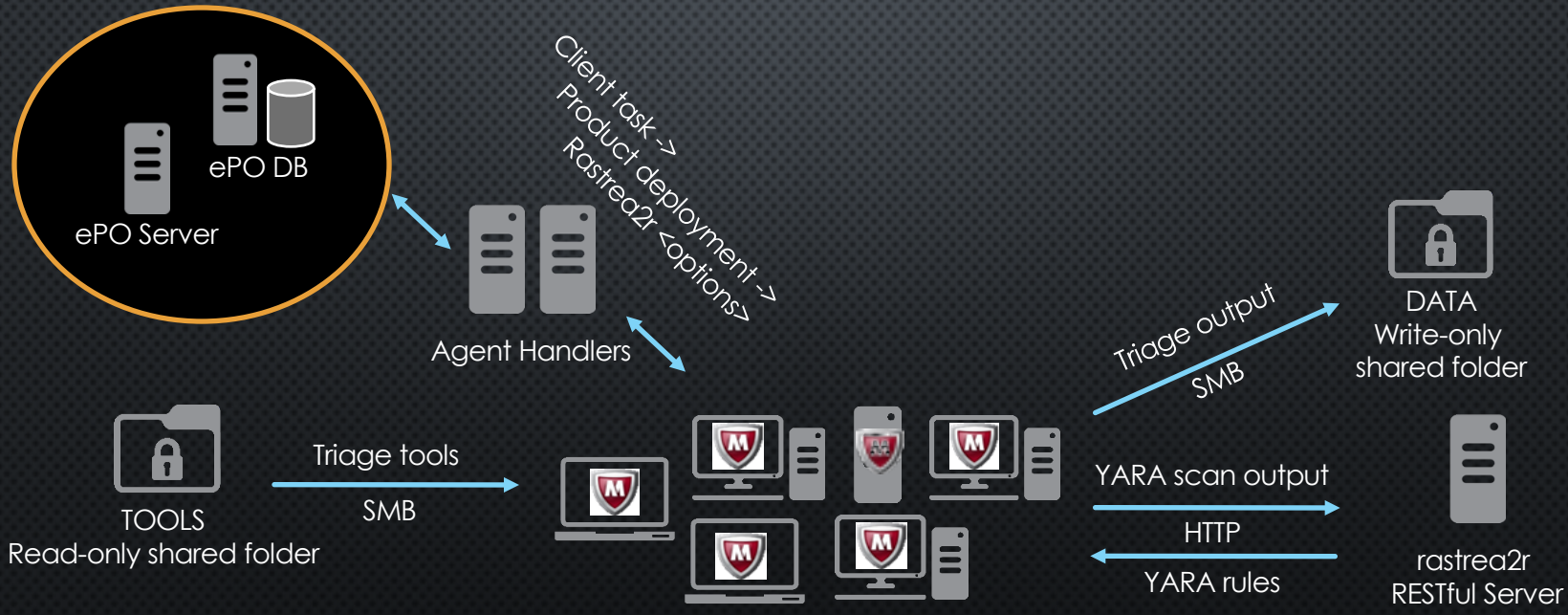
- **YARA-DISK**: YARA SCAN FOR FILE/DIRECTORY OBJECTS ON DISK
- **YARA-MEM**: YARA SCAN FOR RUNNING PROCESSES IN MEMORY
- **MEMDUMP**: ACQUIRES A MEMORY DUMP FROM THE ENDPOINT ** WIN ONLY
- **TRIAGE**: COLLECTS TRIAGE INFORMATION FROM THE ENDPOINT ** WIN ONLY

OBTAINING HELP:

RASTREA2R_WIN32_V0.7.EXE –H

{ TRIAGE + HUNTING ON STEROIDS = RASTREA2R};

FORENSICALLY SOUND ARCHITECTURE AND COMMUNICATION FLOWS



{ TRIAGING WITH RASTREA2R };

- EXAMPLE:
  - RASTREA2R_WIN32_V0.7.EXE TRIAGE *TOOLS.MYSERVER.COM DATA.MYSERVER.COM*

    *** TOOLS.MYSERVER.COM -> HAS A READ ONLY SHARED-FOLDER CALLED **TOOLS**

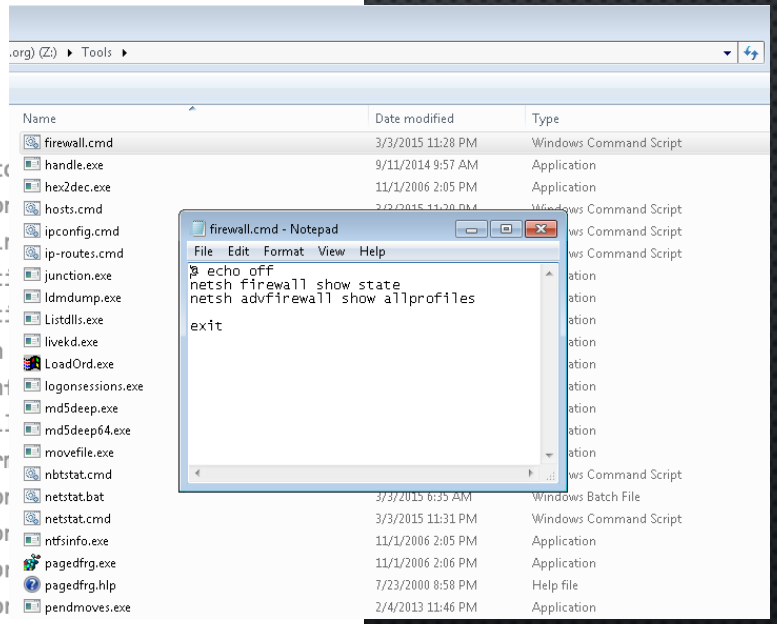
    *** DATA.MYSERVER.COM -> HAS A WRITE ONLY SHARED-FOLDER CALLED **DATA**

```
C:\Users\user\Desktop\rastrea2r client\rastrea2r_win32_v0.6>rastrea2r_win32_v0.6.exe triage -h
usage: rastrea2r_win32_v0.6.exe triage [-h] [-s] BIN_server DATA_server

positional arguments:
  BIN_server    Binary tool server (SMB share)
  DATA_server   Data output server (SMB share)

optional arguments:
  -h, --help    show this help message and exit
  -s, --silent  Suppresses standard output
```

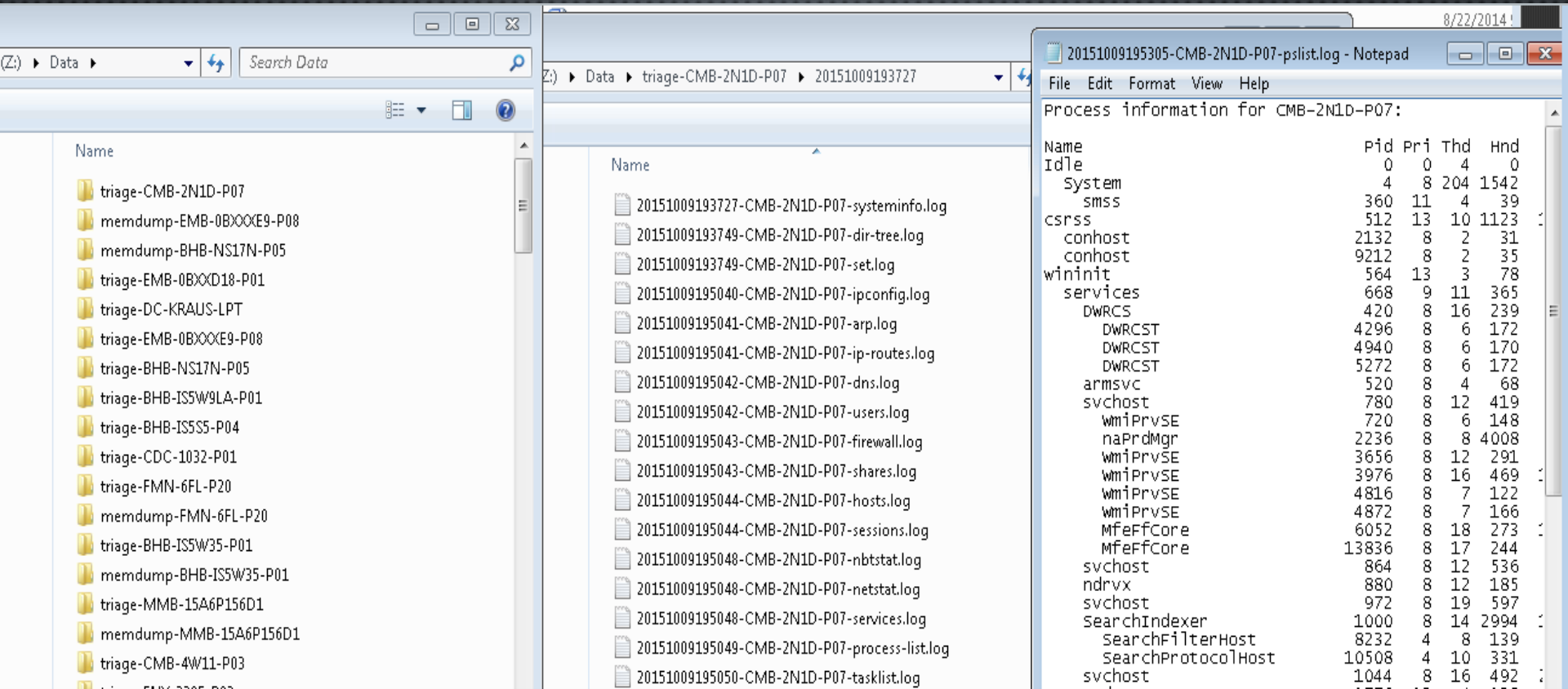
{ 3RD PARTY TOOLS & NATIVE WIN COMMANDS };

COPY THE TOOLSET TO THE READ-ONLY 'TOOLS' SHARE

```
""" Add your list of Sysinternal / third-party / BATCH files here """

tool=(
     'systeminfo.cmd', # Gathers systeminfo
     'set.cmd', # Gathers Set variables
     'dir-tree.cmd', # Enumerates C:\ directo
     'ipconfig.cmd', # Gathers IP information
     'ip-routes.cmd', # Gathers IP routing in
     'arp.cmd', # Gathers ARP table informati
     'dns.cmd', # Gathers DNS Cache informati
     'users.cmd', # Gathers User/local Admin
     'shares.cmd', # Gathers local shares inf
     'firewall.cmd', # Gathers local firewall
     'hosts.cmd', # Captures Host file inform
     'sessions.cmd', # Gathers Active Session
     'nbtstat.cmd', # Gathers NetBios Session
     'netstat.cmd', # Gathers Netstat with pr
     'services.cmd', # Gathers services infor
```



{ 3RD PARTY TOOLS & NATIVE WIN COMMANDS };



{ MEMORY DUMPS WITH RASTREA2R };

- EXAMPLE:
  - RASTREA2R_WIN32_V0.7.EXE MEMDUMP *TOOLS.MYSERVER.COM DATA.MYSERVER.COM*

    *** TOOLS.MYSERVER.COM -> HAS A READ ONLY SHARED-FOLDER CALLED **TOOLS**

    *** DATA.MYSERVER.COM -> HAS A WRITE ONLY SHARED-FOLDER CALLED **DATA**

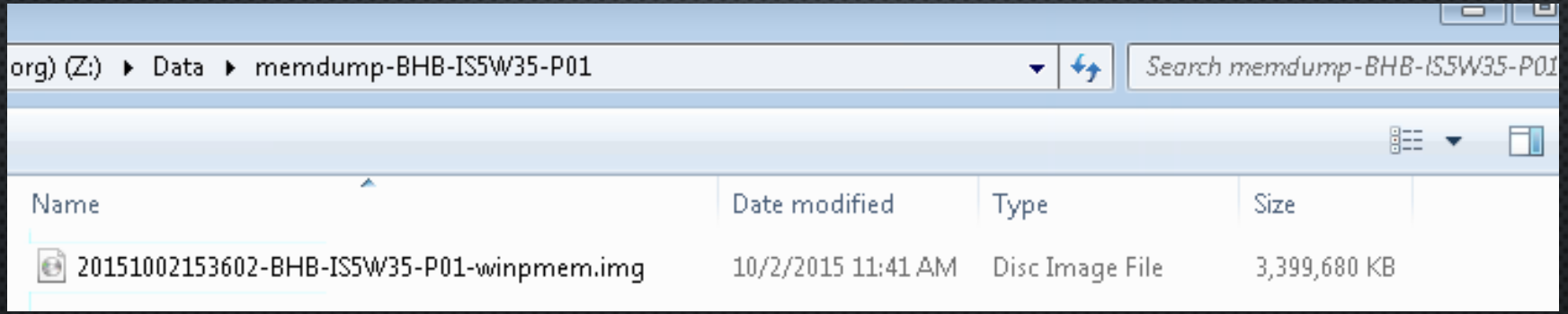
```
C:\Users\user\Desktop\rastrea2r client\rastrea2r_win32_v0.6>rastrea2r_win32_v0.6.exe memdump -h
usage: rastrea2r_win32_v0.6.exe memdump [-h] [-s] BIN_server DATA_server

positional arguments:
  BIN_server    Binary tool server (SMB share)
  DATA_server   Data output server (SMB share)

optional arguments:
  -h, --help    show this help message and exit
  -s, --silent  Suppresses standard output
```

{ MEMORY DUMPS WITH RASTREA2R };

MEMORY DUMPS OF ANY MANAGED HOST PIPED OVER SMB USING *WINPMEM*

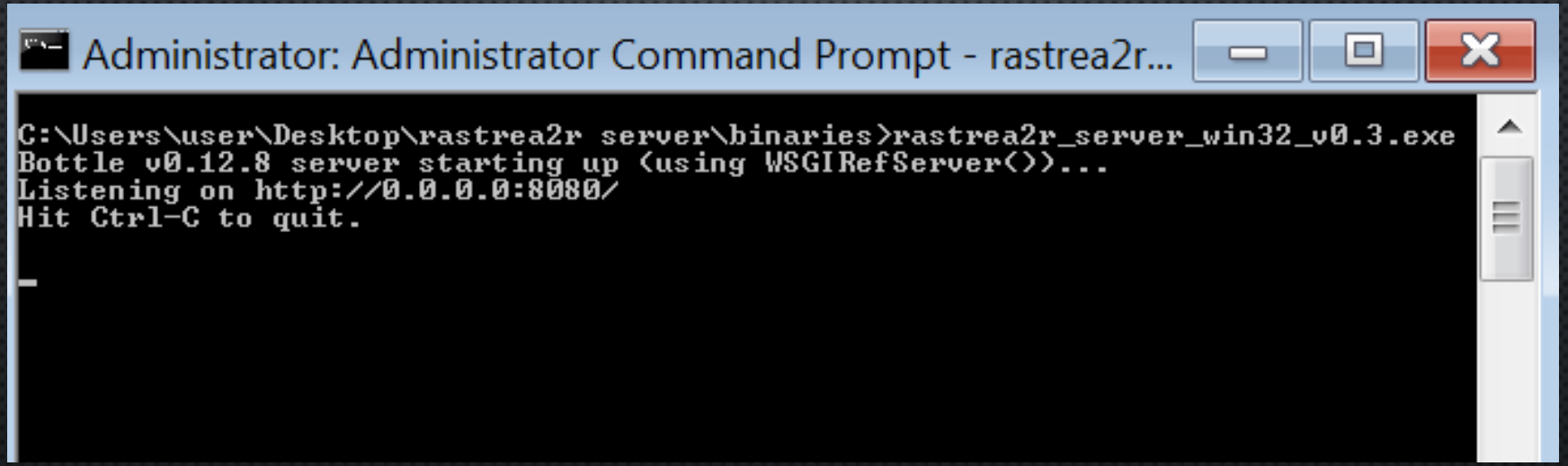
Full memory dump on RAW format:

- Ready to be parsed with memory analysis tools like volatility



{ STARTING THE RASTREA2R SERVER };

LISTENING ON ALL INTERFACES ON PORT 8080



{ HUNTING WITH RASTREA2R };

*** CLIENT / SERVER ARCHITECTURE USING A RESTFUL API

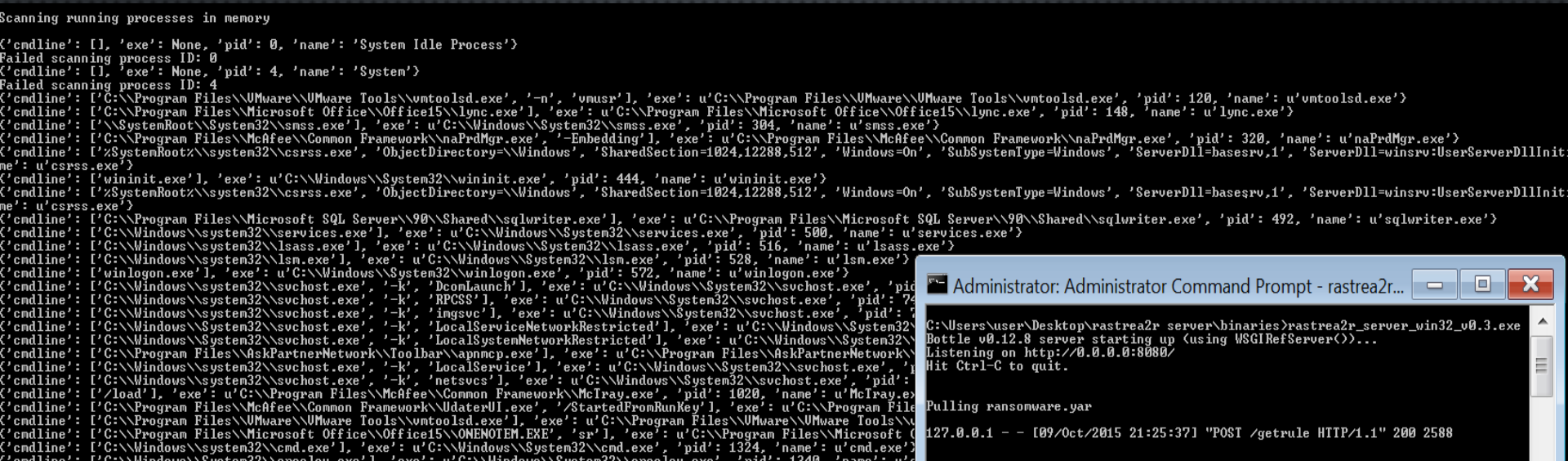
*** YARA RULES MUST BE STORED ON THE SAME DIRECTORY AS THE SERVER

```
C:\Users\user\Desktop\rastrea2r client\rastrea2r_win32_v0.6>rastrea2r_win32_v0.6.exe yara-mem -h
usage: rastrea2r_win32_v0.6.exe yara-mem [-h] [-s] server rule

positional arguments:
  server        rastrea2r REST server
  rule          Yara rule on REST server

optional arguments:
  -h, --help    show this help message and exit
  -s, --silent  Suppresses standard output

C:\Users\user\Desktop\rastrea2r client\rastrea2r_win32_v0.6>rastrea2r_win32_v0.6.exe yara-disk -h
usage: rastrea2r_win32_v0.6.exe yara-disk [-h] [-s] path server rule

positional arguments:
  path          File or directory path to scan
  server        rastrea2r REST server
  rule          Yara rule on REST server

optional arguments:
  -h, --help    show this help message and exit
  -s, --silent  Suppresses standard output
```

{ HUNTING FOR IOCS IN MEMORY WITH YARA };

Example:

- rastrea2r_win32_v0.7.exe yara-mem *localhost ransomware.yar*



{ HUNTING FOR IOCS ON DISK WITH YARA };

Example:

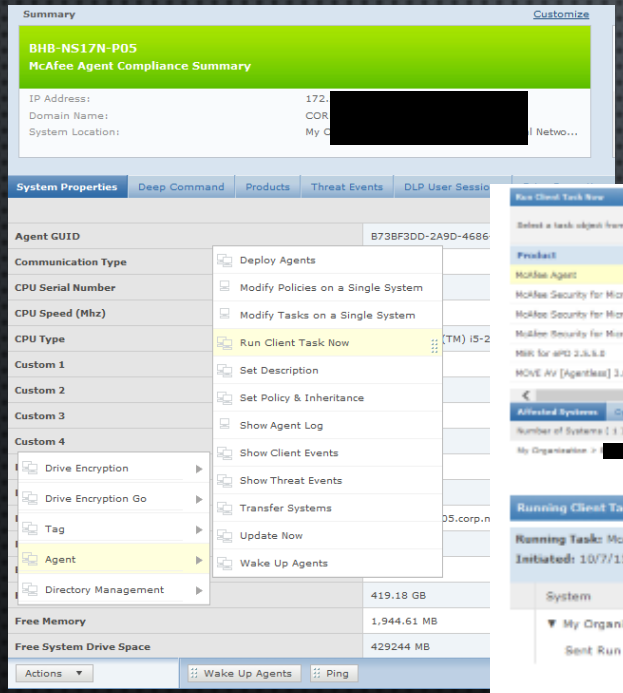
- rastrea2r_win32_v0.7.exe yara-disk c:\users\user *localhost ransomware.yar*

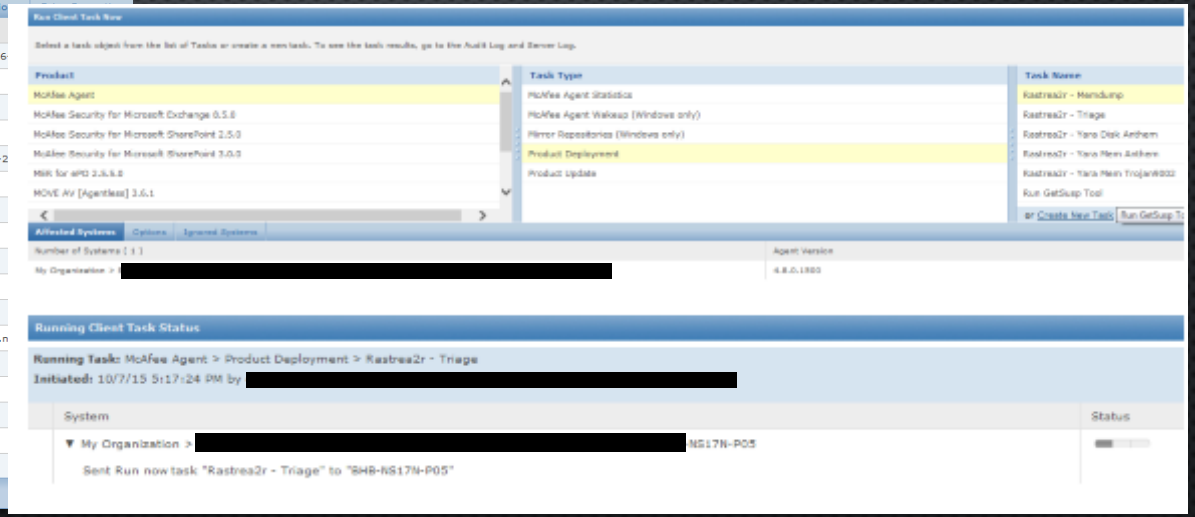
```
            all of ($s*) and filesize < 600
}
rule BackdoorFCKG: CTB_Locker_Ransomware
{
meta:
author = "ISG"
date = "2015-01-20"
reference = "https://blogs.mcafee.com/mcafee-labs/rise-backdoor-fckq-ctb-locker"
description = "CTB_Locker"

strings:
$string0 = "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
$string1 = "RNDBAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
$string2 = "keme132.DLL"
$string3 = "klospad.pdb"
condition:
3 of them
}



Scanning c:\users\user

{'filename': 'c:\\users\\user\\Desktop\\rastrea2r server\\binaries\\ransomware.yar', 'hostname': 'WIN-M4BNKTI076D', 'module': 'yaradisk', 'rulename': BackdoorFCKG}
```

```
C:\Users\user\Desktop\rastrea2r server\binaries>rastrea2r_server_win32_v0.3.exe
Bottle v0.12.8 server starting up (using WSGIRefServer())...
Listening on http://0.0.0.0:8080/
Hit Ctrl-C to quit.


Pulling ransomware.yar

127.0.0.1 - - [09/Oct/2015 21:25:37] "POST /getrule HTTP/1.1" 200 2588
127.0.0.1 - - [09/Oct/2015 21:26:40] "POST /putpid HTTP/1.1" 200 0
127.0.0.1 - - [09/Oct/2015 21:26:46] "POST /putpid HTTP/1.1" 200 0

Pulling ransomware.yar

127.0.0.1 - - [09/Oct/2015 21:31:35] "POST /getrule HTTP/1.1" 200 2588
127.0.0.1 - - [09/Oct/2015 21:33:28] "POST /putfile HTTP/1.1" 200 0
_
```

{ TRIAGING WITH 'RASTREA2R' FROM MCAFEE EPO CONSOLE };

CLIENT TASKS -> PRODUCT DEPLOYMENT -> RASTREA2R TRIAGE/MEMDUMP



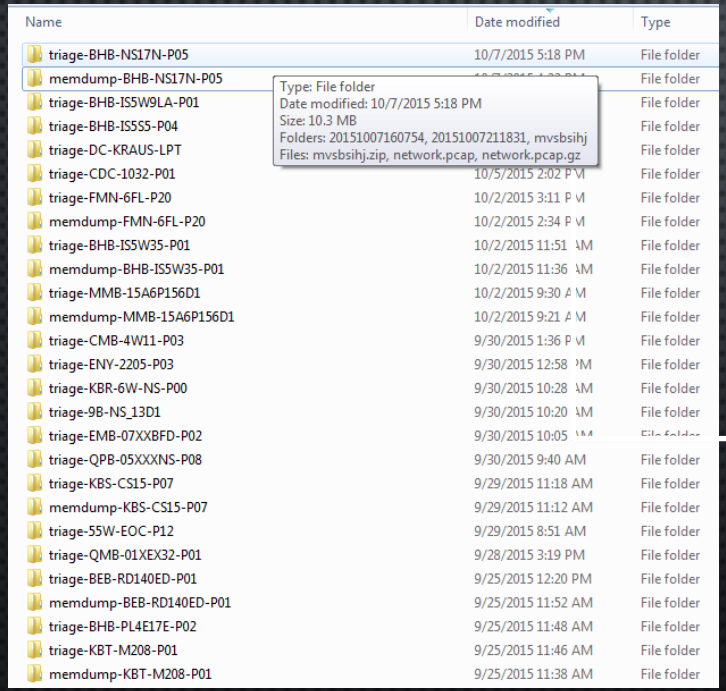


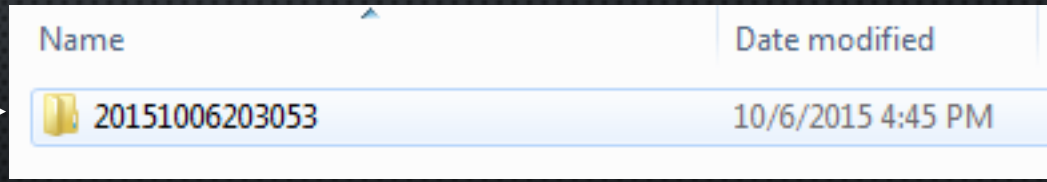
Modularity:

- A specific task is created for each combination of command line switches

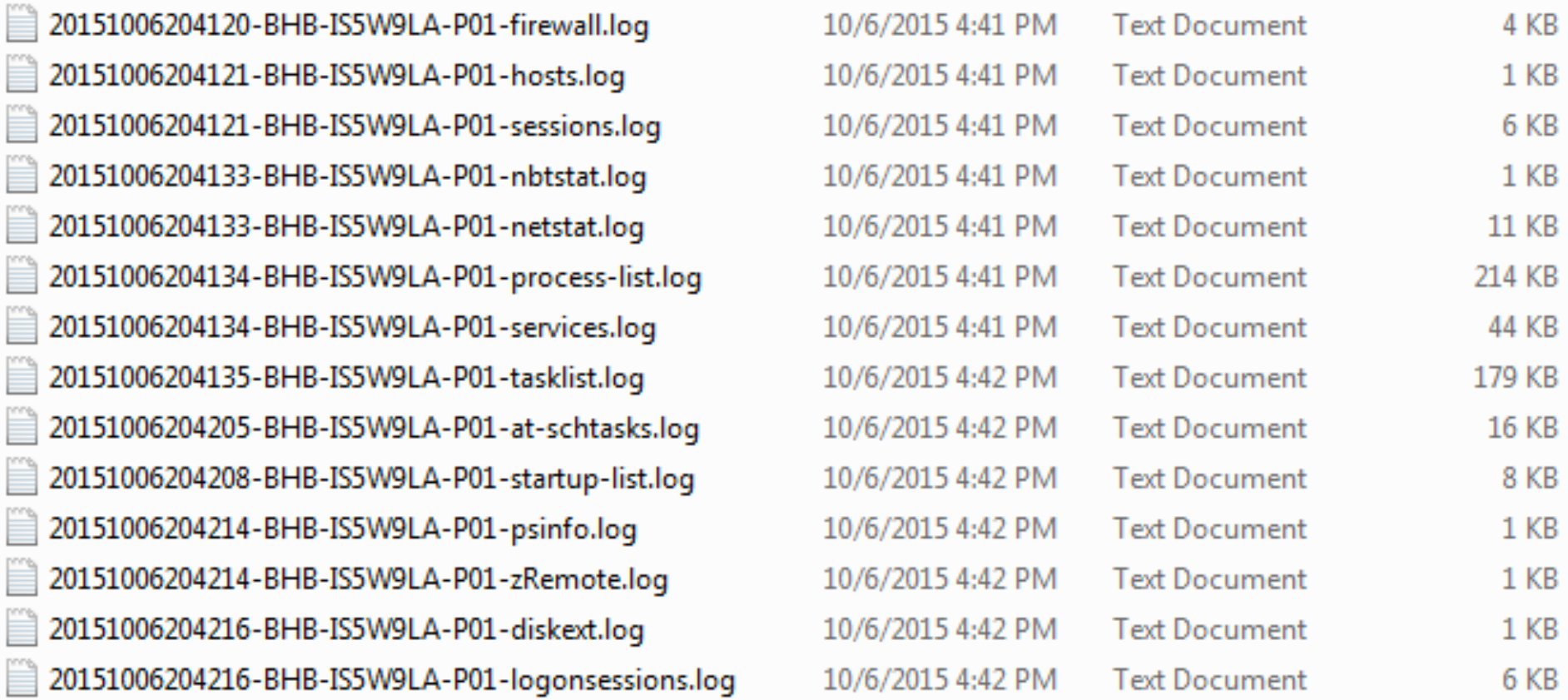
{ ANALYZING THE RESULTS (~5 MINUTES) };

A FOLDER IS CREATED PER SYSTEM, THEN PER COLLECTION SET (TIMESTAMP)





{ ANALYZING THE RESULTS };

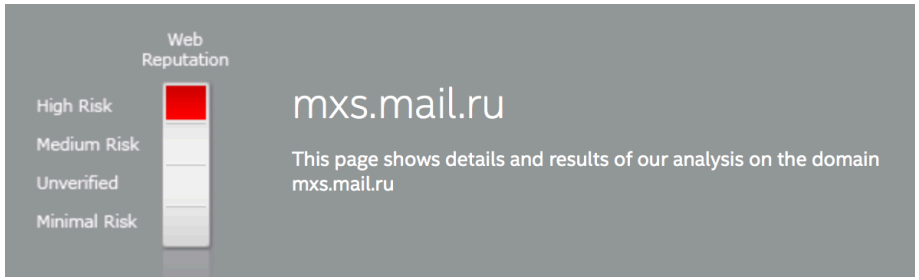


20151006204120-BHB-IS5W9LA-P01-firewall.log	10/6/2015 4:41 PM	Text Document	4 KB
20151006204121-BHB-IS5W9LA-P01-hosts.log	10/6/2015 4:41 PM	Text Document	1 KB
20151006204121-BHB-IS5W9LA-P01-sessions.log	10/6/2015 4:41 PM	Text Document	6 KB
20151006204133-BHB-IS5W9LA-P01-nbtstat.log	10/6/2015 4:41 PM	Text Document	1 KB
20151006204133-BHB-IS5W9LA-P01-netstat.log	10/6/2015 4:41 PM	Text Document	11 KB
20151006204134-BHB-IS5W9LA-P01-process-list.log	10/6/2015 4:41 PM	Text Document	214 KB
20151006204134-BHB-IS5W9LA-P01-services.log	10/6/2015 4:41 PM	Text Document	44 KB
20151006204135-BHB-IS5W9LA-P01-tasklist.log	10/6/2015 4:42 PM	Text Document	179 KB
20151006204205-BHB-IS5W9LA-P01-at-schtasks.log	10/6/2015 4:42 PM	Text Document	16 KB
20151006204208-BHB-IS5W9LA-P01-startup-list.log	10/6/2015 4:42 PM	Text Document	8 KB
20151006204214-BHB-IS5W9LA-P01-psinfo.log	10/6/2015 4:42 PM	Text Document	1 KB
20151006204214-BHB-IS5W9LA-P01-zRemote.log	10/6/2015 4:42 PM	Text Document	1 KB
20151006204216-BHB-IS5W9LA-P01-diskext.log	10/6/2015 4:42 PM	Text Document	1 KB
20151006204216-BHB-IS5W9LA-P01-logonsessions.log	10/6/2015 4:42 PM	Text Document	6 KB

{ THAT SOUNDS HIGHLY SUSPICIOUS… };

```
Caption=ConnectionCenter
Command="C:\Users\BHB-NS17N-P05\AppData\Local\Citrix\ICA Client\concentr.exe" /startup
Description=ConnectionCenter
Location=HKU\S-1-5-21-2250110424-2442967196-2465209428-110119\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
SettingID=
User=CORP\BHB-NS17N-P05


Caption=MSConfig
Command="C:\Users\BHB-NS17N-P05\mvsbsihj.exe"
Description=MSConfig
Location=HKU\S-1-5-21-2250110424-2442967196-2465209428-110119\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
SettingID=
User=CORP\BHB-NS17N-P05


Caption=RTHDVCPL
Command=C:\Program Files\Realtek\Audio\HDA\RtHDVCpl.exe -s
Description=RTHDVCPL
Location=HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
SettingID=
User=Public


Caption=IMSS
Command="C:\Program Files\Intel\Intel(R) Management Engine Components\IMSS\PIconStartup.exe"
Description=IMSS
Location=HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
SettingID=
User=Public
```

```
mail.ru
----------------------------------------
Record Name . . . . . : mail.ru
Record Type . . . . . : 15
Time To Live  . . . . : 4
Data Length . . . . . : 8
Section . . . . . . . : Answer
MX Record . . . . . . : mxs.mail.ru
                        10
                        0


Record Name . . . . . : mxs.mail.ru
Record Type . . . . . : 1
Time To Live  . . . . : 4
Data Length . . . . . : 4
Section . . . . . . . : Additional
A (Host) Record . . . : 217.69.139.150


Record Name . . . . . : mxs.mail.ru
Record Type . . . . . : 1
Time To Live  . . . . : 4
Data Length . . . . . : 4
Section . . . . . . . : Additional
A (Host) Record . . . : 94.100.180.150


b-0.19-43000408.9851081.1644.1f0a.2f4a.410.0.9ape7qnbhhzejna5sl525sn3wb.avts.mcafee.com
----------------------------------------
Record Name . . . . . : b-0.19-43000408.9851081.1644.1f0a.2f4a.410.0.9ape7qnbhhzejna5sl525sn3wb.avts.mcafee.com
Record Type . . . . . : 1
Time To Live  . . . . : 3061
Data Length . . . . . : 4
Section . . . . . . . : Answer
A (Host) Record . . . : 127.129.0.128
```



{ WHAT ABOUT THAT FUNKY BINARY? };

ALWAYS CHECK HASHES. BAD GUYS USE VIRUSTOTAL TOO!

MVSBSIHJ.EXE

- LOW AV DETECTION AT THE TIME OF SUBMISSION (CHECKED WITH PESTUDIO)
- INJECTS ITSELF INTO SVCHOST.EXE
- CHECKS NETWORK STATUS
- DOWNLOADS A SECOND PAYLOAD WITH SPAMBOT CAPABILITIES
- IN ADDITION TO SENDING SPAM IT CAN DOWNLOAD ADDITIONAL PLUGIN COMPONENTS FROM C&C SERVERS:
  - DDOS ATTACKS
  - SNIFF TRAFFIC AND STEAL CREDENTIALS
  - READ MESSAGES ON TWITTER, SKYPE, FACEBOOK, ETC.
  - BITCOIN MINING, ETC.

Type	Size	Blacklisted (5)	Value
ascii	14	x	VirtualAllocEx
ascii	18	x	CreateRemoteThread
ascii	11	x	SendMessage
ascii	8	x	VBA6.DLL
ascii	15	x	DllFunctionCall
ascii	40	-	!This program cannot be run in DOS mode.
ascii	5	-	.text
ascii	6	-	`.data
ascii	5	-	.rsrc
ascii	12	-	MSVBVM60.DLL
ascii	5	-	'2@x;
ascii	24	-	= Filmwissenschaften

{ WHAT NEXT? };

1. CREATE A NEW YARA RULE BASED ON THIS SAMPLE
2. SCAN THE ENVIRONMENT AND TRIAGE INFECTED SYSTEMS
3. TUNE THE YARA RULE BASED ON FINDINGS
4. REPEAT, RINSE AND STOP ONCE THE FULL SCOPE IS DETERMINED
5. CONTAIN & ERADICATE
6. CONDUCT MEMORY ANALYSIS AND FULL DISK FORENSICS TO DETERMINE ROOT CAUSE

```
RULE SYSTEMXYZ-SPAMBOT
{
STRINGS:
    $STRING0 = "FFFFF."
    $STRING1 = "AVVWSH"
    $STRING2 = "AWAVAUATVWSH"
    $STRING3 = "FFFFFF."
    $STRING4 = ".RELOC"
CONDITION:
    4 OF THEM }
```

{ THE RESULTS };

Where are we now?

- Ability to triage and collect evidence from thousands of endpoints centrally managed by McAfee ePO ***IN MINUTES***
- Triage and evidence collection is **AUTOMATED**, with **PROACTIVE** hunting of IOCs based on FBI TLPs, vendor reports, internal IR investigations and other threat intelligence feeds
- Results of the triage process can be **VALIDATED** by trained analysts
- Working on automating evidence/artifacts processing and analysis
- Initial vector of compromise can be determined on most cases

Thank You!

@aboutsecurity

http://blog.ismaelvalenzuela.com

https://github.com/aboutsecurity/