Foundstone®

Hunting for Indicators of Compromise with Free/Open Source Tools

SANS @Night, McLean VA - February 18, 2016

Ismael Valenzuela, Lead IR/Forensics Technical Practice Manager
(Foundstone Professional Services)

intel Security

About me

Twitter: @aboutsecurity

- +15 years of experience in InfoSec
- Leading the Incident Response and Forensics team at Intel Security
- SANS Instructor
- Author of security articles for Hakin9, INSECURE, SANS Forensic Blog & OpenSecurityResearch
  - http://blog.ismaelvalenzuela.com
- GSE #132 + 10 other GIAC certificates
- Love eating packets for breakfast (specially non-RFC compliant – yummy!!)

So What Is Continuous Monitoring?

Taking your head out of the sand

- What **is not** CM:
  - It's not COMPLIANCE
    - Checking boxes
    - A point in time vulnerability audit
- Key CM principles
  - Presumption of Compromise
  - Detection Oriented - Response Driven
  - Post Exploitation Focus
  - Proactive Detection: **HUNTING**!

Successful Techniques for Hunting / CM

What we will cover today

- Spotting abnormal in your network
  - Network flows with SiLK
  - Behavioral analysis with Bro
  - Detecting entropy with freq.py
- Monitoring changes in endpoints
  - Sysmon
- Centralizing, Analyzing and Alerting on Logs
  - ELK & Elastalert
- Deceiving the attacker
  - Honeytokens
- Collecting & Hunting for Indicators of Compromise
  - rastrea2r

All of this and much more in the new SANS SEC511 Bootcamp Edition

Northern Virginia - McLean 2016

McLean, VA | Mon, Feb 15 - Sat, Feb 20, 2016

Notice: SEC511 has extended Bootcamp hours, 5pm-7pm (Day 1 - Day 5) designed to enhance your skills with even more hands-on labs. (Excludes Mentor)

SEC511: Continuous Monitoring and Security Operations

- Contents | Schedule | Additional Info
- **Instructor:** Ismael Valenzuela
- 5,620 USD
- GMON Certification
- Affiliate Pricing
- 46 CPEs
- Laptop Required

Event starts in 3 Days

Register **Now**

Free Excerpt

Event Home

Spotting abnormal in your network with SiLK

Network Situational Awareness

- What's on my network?
- What happened before / after event X?
- Was there exfiltration at a any given time?
- Are users browsing to known infected websites?
- Is there a spam bot in my network?
- What are the top high volume connections in my network?
- What systems are talking to IPs in foreign countries?

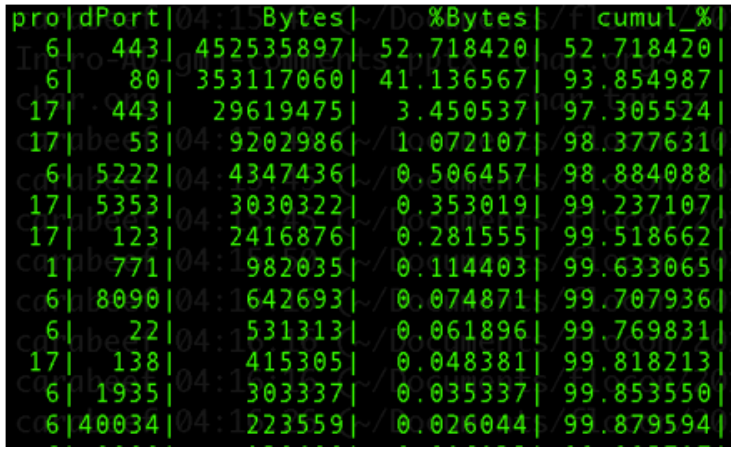
- System for Internet Level Knowledge (SiLK)
- Developed and maintained by the CERT Network Security Situational Awareness Team (CERTNetSA)
- Designed to facilitate security analysis of large networks, supporting the notion of network behavioral analysis
- Supplements signature based detection (IDS / IPS) and Full Packet Captures
- SiLK flow (opensource) ≠ Netflow (commercial)

https://tools.netsa.cert.org/silk/

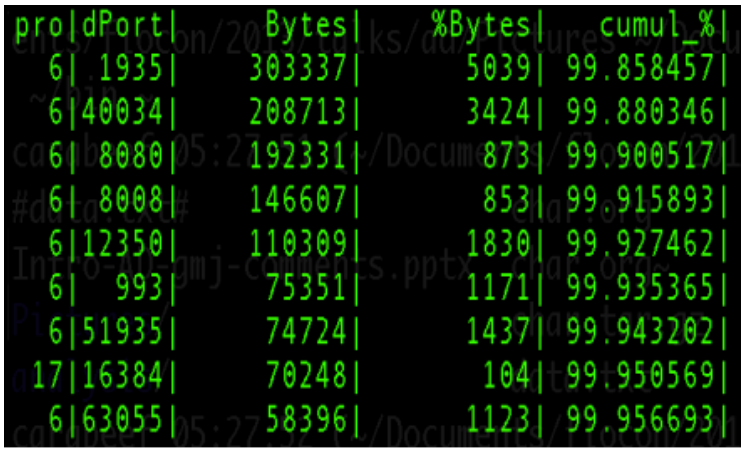
Example: profiling potential exfiltration with SiLK

Rwp2yaf2silk and rwfilter

- Converting a pcap to flow with rwp2yaf2silk:
  - rwp2yaf2silk –in=capture.pcap –out=capture.pcap
  - rwfilter usage: http://tools.netsa.cert.org/silk/analysis-handbook.pdf



pro	dPort	Bytes	%Bytes	cumul_%
6	443	452535897	52.718420	52.718420
6	80	353117060	41.136567	93.854987
17	443	29619475	3.450537	97.305524
17	53	9202986	1.072107	98.377631
6	5222	4347436	0.506457	98.884088
17	5353	3030322	0.353019	99.237107
17	123	2416876	0.281555	99.518662
1	771	982035	0.114403	99.633065
6	8090	642693	0.074871	99.707936
6	22	531313	0.061896	99.769831
17	138	415305	0.048381	99.818213
6	1935	303337	0.035337	99.853550
6	40034	223559	0.026044	99.879594



pro	dPort	Bytes	%Bytes	cumul_%
6	1935	303337	5039	99.858457
6	40034	208713	3424	99.880346
6	8080	192331	873	99.900517
6	8008	146607	853	99.915893
6	12350	110309	1830	99.927462
6	993	75351	1171	99.935365
6	51935	74724	1437	99.943202
17	16384	70248	104	99.950569
6	63055	58396	1123	99.956693

http://www.cert.org/flocon/2013/presentations/sample-char-intro-anomaly-detection.pdf

Behavioral Analysis with Bro: Beyond the IDS

Show me what you've seen Bro!

- Bro (created in 1994) enhances network visibility beyond traditional signature-based detection, enabling true analysis through protocol decoding.
  - More than an IDS: a **Network Programming Language**

- Provides full context of all activity related to network events:
  - What domains a host queries
  - What SSL certificates are used
  - What files are downloaded
  - Any FTP/SMTP/IRC/SQL activity, etc
  - What User Agents are used

- Provides a flexible framework that facilitates customized, in-depth monitoring beyond traditional IDS

Snort vs Bro: Signature vs Behavior based Analysis


```
Terminal
File  Edit  View  Terminal  Go  Help
[/tmp/bro]$ bro -r /pcaps/fraudpack.pcap
[/tmp/bro]$ ls
conn.log  dns.log  files.log  http.log  packet_filter.log  weird.log
[/tmp/bro]$ cat http.log | bro-cut user_agent uri
Downloader MLR 1.0.0    /get_xml?stb=1&did=566628426&file_id=167110456
Downloader MLR 1.0.0    /download/252948
Downloader MLR 1.0.0    /music/7/07/e-type_-_russian_lullaby_(zvukoff.ru).mp3
Downloader MLR 1.0.0    /Internet.exe
Downloader MLR 1.0.0    /mailrusputnik.exe
Downloader MLR 1.0.0    /download/252948
Downloader MLR 1.0.0    /mailrusputnik.exe
FULLSTUFF       /update/2/version.txt?type=install&GUID={99CFA828-D430-4DFE-A391
-215BF62C53BC}&rfr=newcustom2&standalone=1&bgn=1
Downloader MLR 1.0.0    /music/7/07/e-type_-_russian_lullaby_(zvukoff.ru).mp3
FULLSTUFF       /update/2/version.txt?type=install&GUID={99CFA828-D430-4DFE-A391
-215BF62C53BC}&rfr=newcustom2&success=1&ieovr=0&ffovr=0&br=ie&brver=6.00&bfr=0&a
ftr=1&bfr2=aHR0cDovL3d3dy5taWNyb3NvZnQuY29tL2lzYXBpL3JlZGlyLmRsbD9wcmQ9aWUmcHZlc
j02JmFyPW1zbmhvbWU[&aftr2=aHR0cDovL3d3dy5tYWlsLnJ1L2NudC85NTE2
FULLSTUFF       /update/2/version.txt?type=install&GUID={99CFA828-D430-4DFE-A391
-215BF62C53BC}&rfr=newcustom2&standalone=1&uacenabled=0&uacpass=1
GuardMailRu     /guard_settings.xml
GaurdMailRu     /guard/verinfo.xml?hash=wN3bydmXOdiJwd7Owt6Um52I2M3Il5mOyNvEyJbR
kJHs6OyUmZKE7JudnYGf7u/tgu+elZqHm5ma7Ouamemcm+3t0Irb2MjN28HfxcrElJiJ38TczN/I2suT
nYrdz9uVnoCdgpuEn5qc
GuardMailRu     /update_guard/version.xml?ver=1.0.0.623&guid={99CFA828-D430-4DFE
[/tmp/bro]$
[/tmp/bro]$
```

Some of the Great Functionalities in Bro

BroAWESOMENESS

- One of the most powerful is the ability to extract files:
  - Geo Spacial -> per country of origin
  - Signature based
  - Destination based
- Can be combined with actions to do powerful automated analysis:
  - Look up in Malware Hash Registry or other Threat Intel resources
  - Upload to VT, Cuckoo, you name it!
- Intelligence framework to integrate TI feeds (see http://intel.criticalstack.com)
- New functionalities in 2.4:
  - Support for external plugins
  - Support for new protocols:
    - MySQL
    - RDP
    - Kerberos
    - DTLS
    - SIP
    - New rewritten SSH analyzer
  - A new file analyzer for Portable Executables
  - Access to ICMP payloads
- Play and learn:
  - http://www.bro.org/community/index.html

Ask Bro!

Suspicious SSL & DNS traffic: a shameless plug

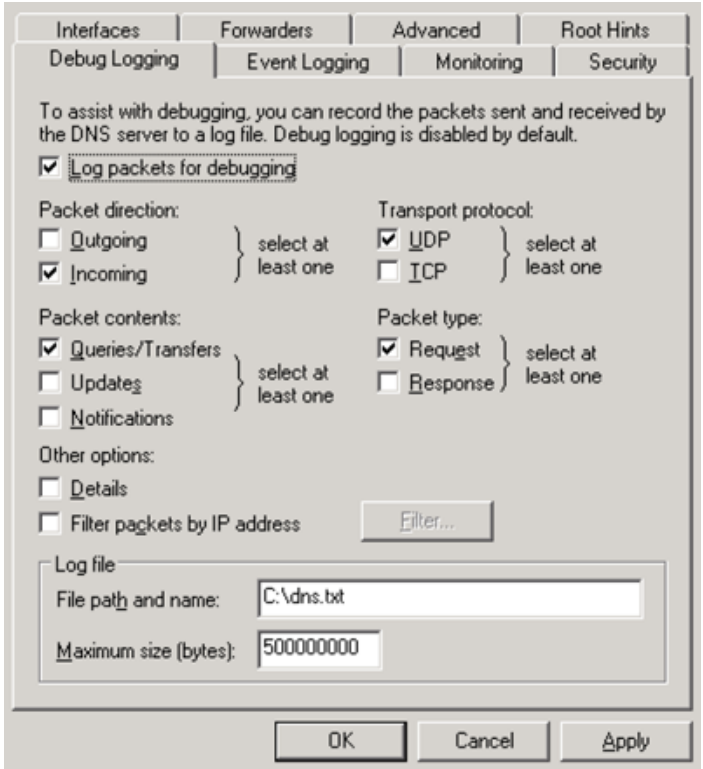
- Blog post at OpenSecurityResearch, now also included in SANS SEC 503 material
  - http://blog.opensecurityresearch.com/2014/03/identifying-malware-traffic-with-bro.html

```
$ cat ssl.log | bro-cut server_name, subject, issuer_subject

www.seu4oxkf6.com CN=www.tl6ou6ap7fjroh2o.net CN=www.tbajutyf.com
www.fjpv.com CN=www.vklxa6kz.net CN=www.ohqnkijzzo5vt.com
www.pdpqsu.com CN=www.5rthkzelyecfpir56.net CN=www.qbboo7mcwzv7.com
www.vkojgy6imcvg.com CN=www.dctpbbpif6zy54mspih.net CN=www.m6hoayo5cga.com
www.dbyryztrr7sui3rskjvikes.com CN=www.getvdkk6ibned7k3krkc.net CN=www.7pz4gaio6
www.xqwf7xs6nycmciil3t5e4fy5v.com CN=www.hstk2emyai4yqa5.net CN=www.wc62pgaaorhc
www.rix56ao4hxldum4zbyim.com CN=www.icab4ctxldy.net CN=www.wmylm3gln.com
www.uabjbwhkanlomodm5xst.com CN=www.bnbhckfytu.net CN=www.w4rlc25peis46haafa.com
www.dl2eypxu3.com CN=www.e6nbbzucq2zrhzqzf.net CN=www.cbj5ajz4qgeieshx32n.com
www.ebd7caljnsax.com CN=www.cvapjjtbfd6yohbarw5q.net CN=www.brbqn4rqhscp4rdq.com
```

```
$ cat dns.log | bro-cut query | sort –u

a37fwf32k17gsgylqb58oylzgvlsi35b58m19bt.com
a47d20ayd10nvkshqn50lrltgqcxb68n20gup62.com
a47dxn60c59pziulsozaxm59dqj26dynvfsnw.com
a67gwktaykulxczeueqf52mvcue61e11jrc59.com
axgql48mql28h34k67fvnylwo51csetj16gzcx.ru
ayp52m49msmwmthxoslwpxg43evg63esmreq.info
azg63j36dyhro61p32brgyo21k37fqh14d10k37fx.com
cvlslworouardudtcxato51hscupunua57.org
cyh44jud50g33iuarlzgqbup22fqisixf62kr.org
d10h34othyp62b18lyfwnzazj26p42fud50gzc49.biz
d20iwe51ftitg53lvl18a27hvlqjyjtd20gue61.com
dqhzhtbto21h14lvp12iqhtlrnxasarcte61.biz
drp42i25ati55m69pvgza57nyh34hwk57i55m19n60.ru
iqcqmrn30iuoubuo11crfydvkylrbtmtev.info
iqo11c69mud20krk57j16fqnrfwgva67oraql48.com
isjqn30a27hwgqbxnxksi65hrnsgyc49mylt.biz
iupqhxfwpylxm29jsexovj16cqfybwb68aw.org
iwpslvesj26i65oynxhtoyc39o41asdvnqc59.com
j36lxf52hsj56itc49lqayoveymwfzosi15jw.org
```

DNS logs FTW!

- Outbound DNS traffic is generally ignored
  - Debugging tools? Seriously? Thanks Microsoft!
- Quickwin:
  - Log and monitor:
    - Large DNS queries with high entropy
    - Large TXT record responses
    - High volume of NXDOMAIN responses (typically a sign of compromise)
  - BRO, wireshark/tshark are your friends!
  - i.e. **tshark –r sample.pcap -T fields -e dns.qry.name -e ip.dst -R "dns.flags.rcode==3" | sort |uniq –c**



DGAs and How to Detect Entropy

Mark Baggett's way!

- Zeus Gameover, Skybot, Styx Exploit Kit and many others leverage Domain Generation Algorithms to create random looking hostnames for their C&C servers
- They are challenging to 'detect' in an automated fashion
- Quickwin:
  - Extract all DNS queries with BRO or from DNS Server's logs
  - Use linux's comand "ent" to score the degree of entropy
    - *$ head –c 1000000 /dev/urandom | ent* → Entropy = 7.999982 bits per byte
    - *$ python –c "print 'A' * 1000000" | ent* → Entropy = 0.000021 bits per byte

- ISC Handler Mark Baggett has provided a Python script to measure badness using 'ent' and frequency tables: https://github.com/MarkBaggett/MarkBaggett/tree/master/freq

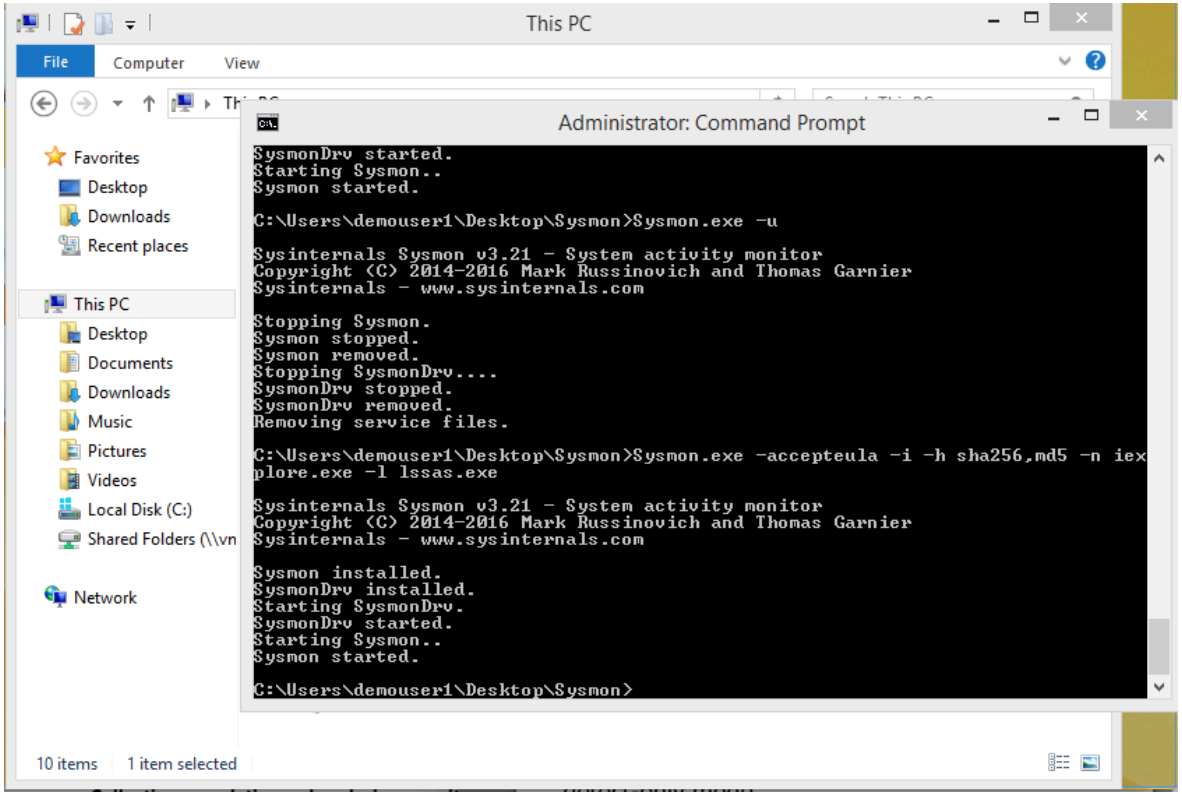
https://isc.sans.edu/diary/Detecting+Random+-+Finding+Algorithmically+chosen+DNS+names+(DGA)/19893

Endpoint Monitoring with Sysmon (Sysinternals)

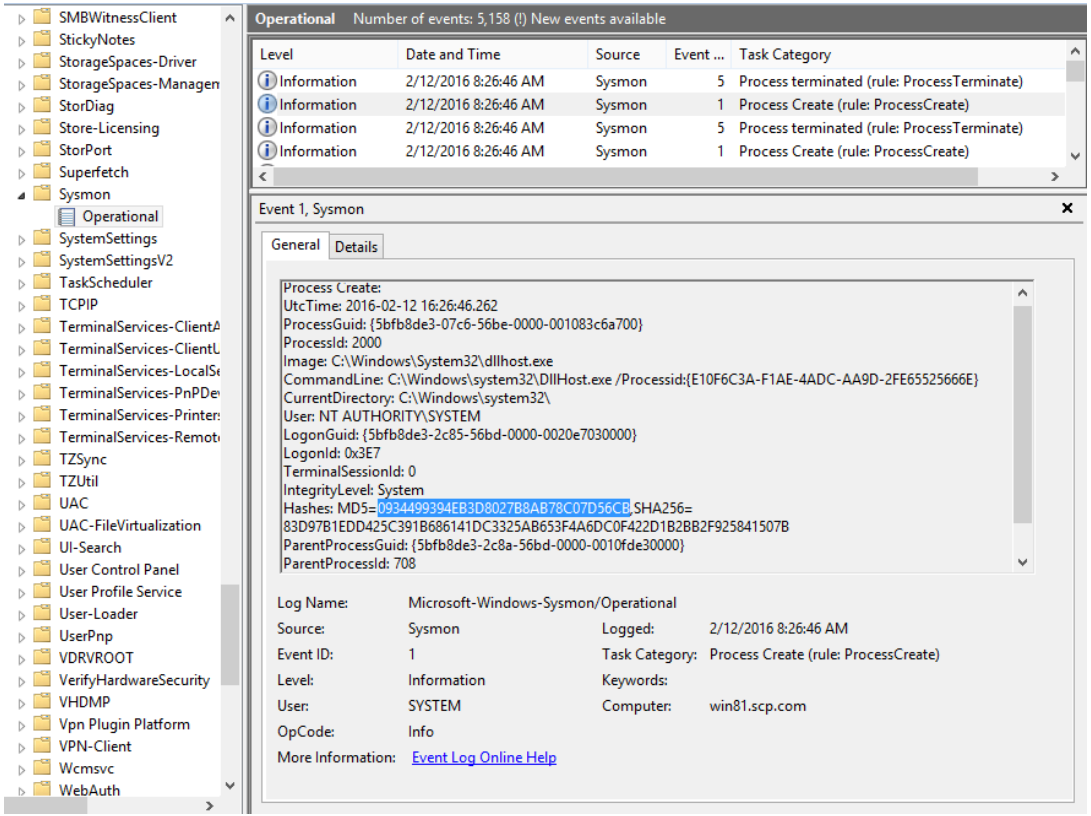
It's FREE!

- Microsoft Sysinternals Sysmon (latest release 3.2 – January 2016)
- A background service that monitors and logs security relevant processes and network activity to the Windows Event log
  - Process creation with full command line
  - Records the hash of process image using SHA1, MD5, SHA256 or IMPHASH
  - Can automatically submit to services as VirusTotal
  - Logs loading of drivers or DLLs with their signatures of hashes
  - Logs opens for raw read access of disks and volumes (used by malware to bypass security!)
  - Optionally, it can log network connections too (source process, IP address, port numbers, hostnames and port names)
- A must have to monitor your Crown Jewels!
- Download and more info on usage at https://technet.microsoft.com/en-us/sysinternals/sysmon

Endpoint Monitoring with Sysmon (Sysinternals)



Endpoint Monitoring with Sysmon (Sysinternals)



Collecting, Searching, Analyzing and Visualizing Logs

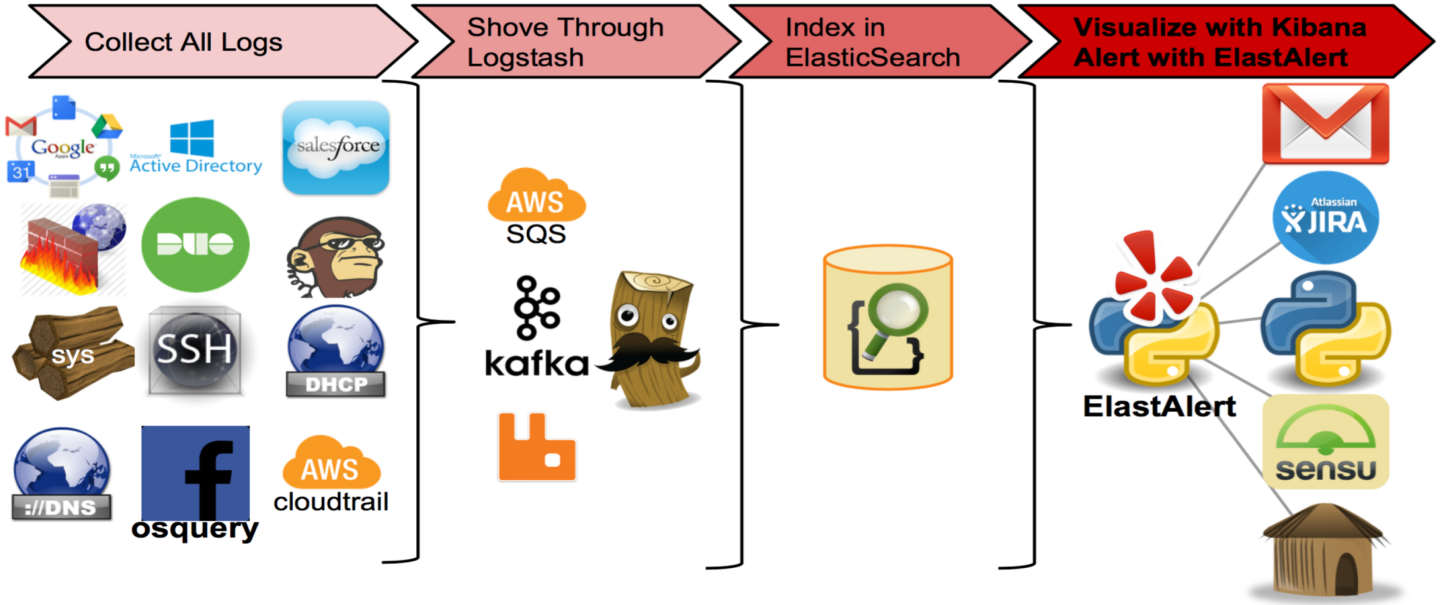
ELK

- Index as much as you want - no limit on volume, speed or position of the moon
- Great for **long tail analysis** and querying
- Open Source, free to use, commercial support
  - **Logstash**: https://www.elastic.co/downloads/logstash
  - **Elasticsearch:** https://www.elastic.co/downloads/elasticsearch
  - **Kibana:** https://www.elastic.co/downloads/kibana

ELK customization

There's no such thing as 'enough' logs!

- Each step can be customized using different plugins:

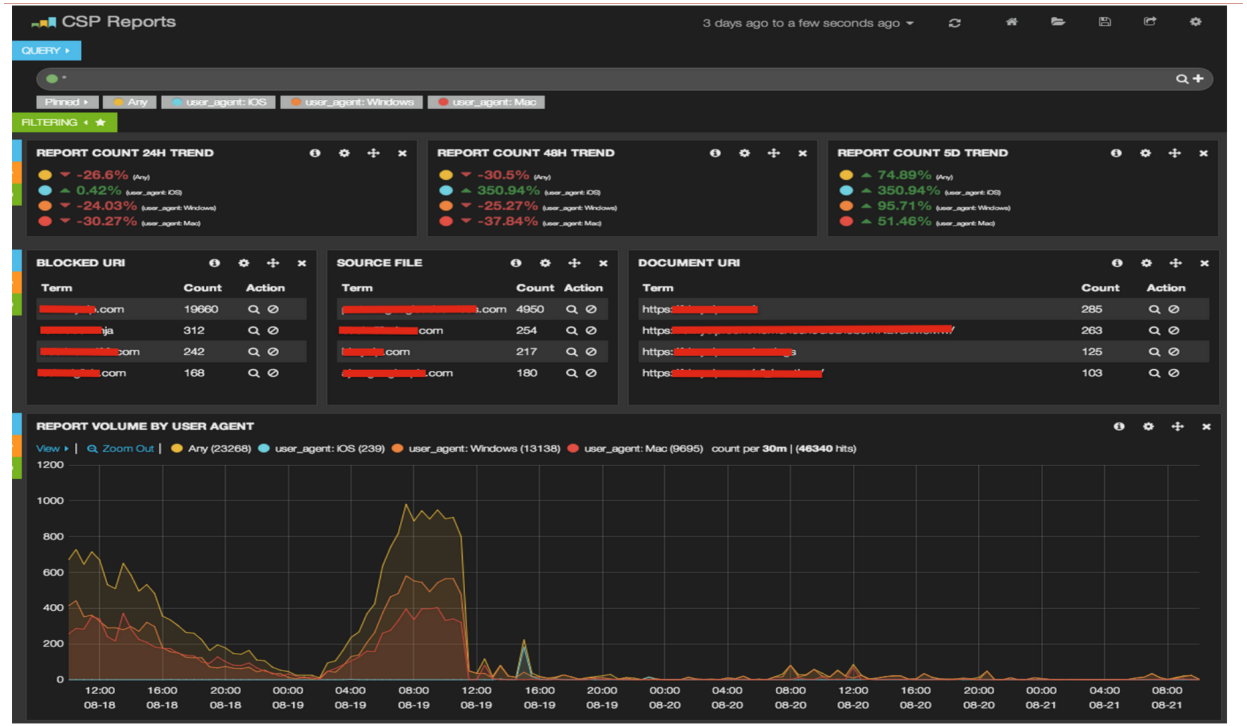


Elastalert: Easy & Flexible Alerting with ElasticSearch

CM on steroids

- ElastAlert is a simple framework for alerting on anomalies, spikes, or other patterns of interest from data in Elasticsearch
- YELP project that leverages Elasticsearch, Logstash and Kibana to create alerts when data matches certain patterns, leveraging the ELK stack in near real time
- Support fo these types of alert:
  - Command
  - Email
  - JIRA
  - SNS
  - Slack
  - Debug…
- More on https://github.com/Yelp/elastalert & https://elastalert.readthedocs.org/en/latest/

Elastalert: Easy & Flexible Alerting with ElasticSearch



Detecting ‘pivoting’ with Honeytokens

Or how to deceive the attacker

- Honeytokens or non-computer based honeypots
  - Also, honeyhashes, credential canaries, password phonies, fake SATs, etc.

- They can be extremely useful to increase your capabilities to detect and react faster to credential theft and lateral movement
  - *echo “superpassword”* | *runas /user:mydomain.com\superadmin /netonly ipconfig*
  - Create a scheduled task that checks for Event ID 4625 in the Security event log (logon failed) and a script that sends an alert whenever the ‘superadmin’ account is found on that log
  - Also add the HASH of “superpassword” to a rule in your IDS and alert anytime it’s seen in the internal network

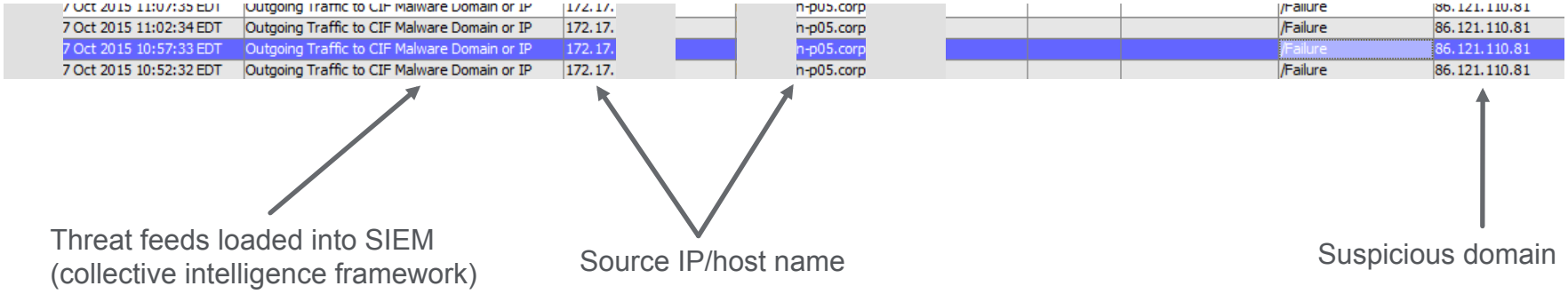
https://isc.sans.edu/forums/diary/Detecting+Mimikatz+Use+On+Your+Network/19311/

Sometimes you need to dig deeper though…

Typical Day in the Life of a SOC Analyst

And why do you need more context

- Traffic blocked to a suspicious IP
  - What process is generating this traffic on the endpoint? And why?
  - Is there any other malicious activity on this host that is not being detected?
  - How do I respond to this? What should I do next?



Some of the Questions in My Head…

And probably in yours too

- When a system is reported as infected or is acting suspiciously:
  - Do I have enough context to determine what to do next?
  - Should I simply disconnect the system and re-image?
  - What if the attacker can detect my response and change tactics?
  - Is this ransomware, a non-targeted campaign or an APT like attack?
  - Is it possible that the system holds other malware that hasn't been detected yet?
  - How can I collect enough information from these systems (quickly enough) to determine the best containment and eradication strategy?
  - Can they help me to profile the attacker's techniques?
  - How can I proactively search for Indicators of Compromise (IoC) across my endpoint?

Prototyping a Solution: Quick & Dirty

Requirements

1. I should be able collect host artifacts required for triage (event analysis) from all endpoints whenever a system is infected or exhibiting some suspicious behavior
2. Without deploying additional agents to the endpoint
3. Cross-platform
4. Must support automated, remote response
5. With minimal footprint
6. Adhering to forensically sound procedures
7. Able to triage using both native and third party tools
8. Able to support proactive hunting for any given IoCs
9. Extensible to integrate with well known IR and forensics open source frameworks

Automation=Python

And why you should love Flying Circus too

- Why did I choose 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 under Windows, Linux and Mac OSX, and uses the OS support to load dynamic libraries, ensuring full compatibility
6. Also… it was named after the great Monty Python!

What Do We Need to Automate?

Automated smart “triage”

- Malware doesn’t exist in a vacuum:
  - They need to run
  - They need to communicate
  - They need to be persistent

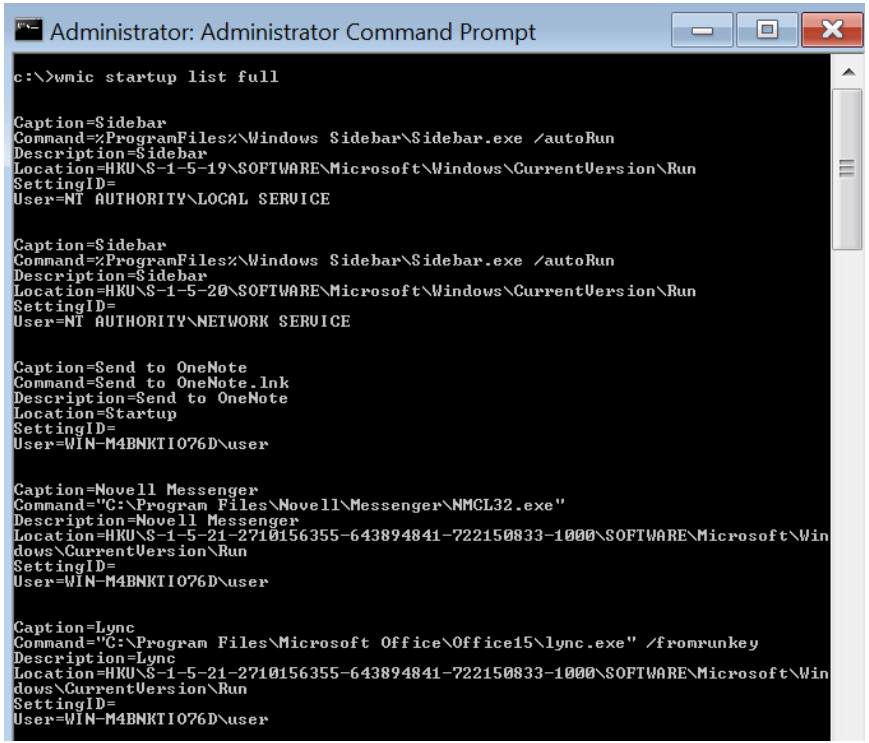
- How do we find “evidence”?
- Traditional forensics techniques are too time consuming
- Triaging can be used to identify relevant evidence quickly and guide the IR process

- Triage techniques:
- 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)

Remote Live Forensics

Artifacts of interest

- Typical windows artifacts:
- Binaries executing from temporary or cache folders
- Binaries executing from user profiles (AppData, Roaming, Local, etc.)
- Binaries executing from C:\RECYCLER
- Binaries executing as alternate data streams
- Binaries with high entropy (random, unusual file names)
- Other suspicious activity



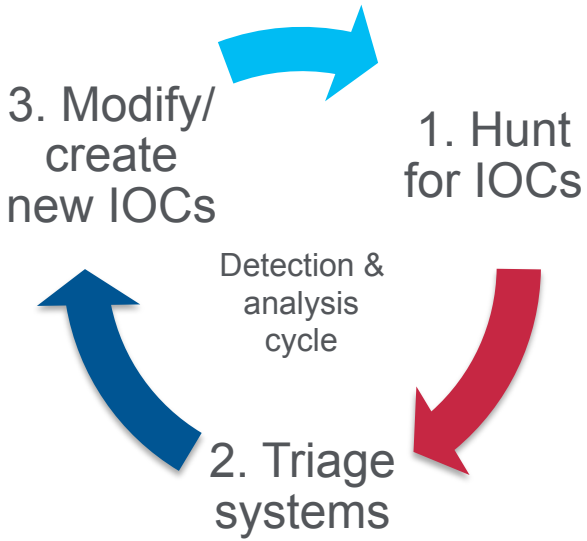
Hunting: From Passive Detection to Proactive Defense

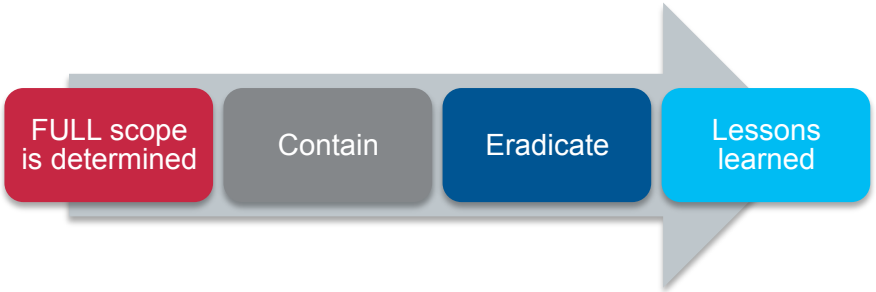
Like turning the lights of the stadium on!

- Understand the Cyber Attack Chain:
  - Map detection/reaction mechanisms with each phase
  - Look for indicators in each phase
  - Proactively hunt for those in your organization
- Can leverage third party IOCs (hashes, IP addresses, URLs, etc.)
- Requires multi-disciplinary skills and understanding of attackers TTPs
- An additional benefit is that it gets you familiar with what is normal vs. what is abnormal in your environment

Hunting & Smart Incident Response

And why you shouldn't hunt on a Friday





Hunting with YARA

Simply a better GREP!

- “The pattern matching Swiss knife for malware researchers (and everyone else!)”
  - http://plusvic.github.io/yara

- Pattern matching:
  - Strings
  - Regular expressions
  - Binary patterns (hex strings)

- Classification:
  - On input: combination of strings & logic, stored in a YARA rule
  - On output: tags, metadata

- YARA is not an AV Scanner, a correlation engine nor does it use any IA

- Can be integrated in your Python projects (bindings)

Show Me a YARA Rule Please

NEW.NEXT.
SECURITY IN FOCUS.

Yara rule to detect the malware:

```
rule EmiratesStatement :
{
meta:
author = “Christiaan Beek”
date = “2013-06-30”
description = “Credentials Stealing Attack”
hash0 = “0e37b6efe5de1cc9236017e003b1fc37”
hash1 = “a28b22acf2358e6aced43a6260af9170”
hash2 = “6f506d7adfcc2288631ed2da37b0db04”
hash3 = “8aebade47dc1aa9ac4b5625acf5ade8f “

strings:
$string0 = “msn.klm”
$string1 = “wmsn.klm”
$string2 = “bms.klm”
condition:
all of them
}
```

A Yara rule to detect Shifu:

```
rule Shifu : Shifu
{
strings:

$a = “CryptCreateHash”
$b = “RegCreateKeyA”
$c = {2F 00 63 00 20 00 73 00 74 00 61 00 72 00 74 00 20 00 22 00 22 00 20 00 22 00 25
00 73 00 22 00 20 00 25 00 73 00 00 00 00 00 63 00 6D 00 64 00 2E 00 65 00 78 00 65 00
00 00 72 00 75 00 6E}
$d = {53 00 6E 00 64 00 56 00 6F 00 6C 00 2E 00 65 00 78 00 65}
$e = {52 00 65 00 64 00 69 00 72 00 65 00 63 00 74 00 45 00 58 00 45}

condition:
all of them
}
```

- Sources:
  https://blogs.mcafee.com/mcafee-labs/targeted-campaign-steals-credentials-in-gulf-states-and-caribbean/:https://blogs.mcafee.com/mcafee-labs/japanese-banking-trojan-shifu-combines-malware-tools/

When you are told that you need to deploy an additional agent!!

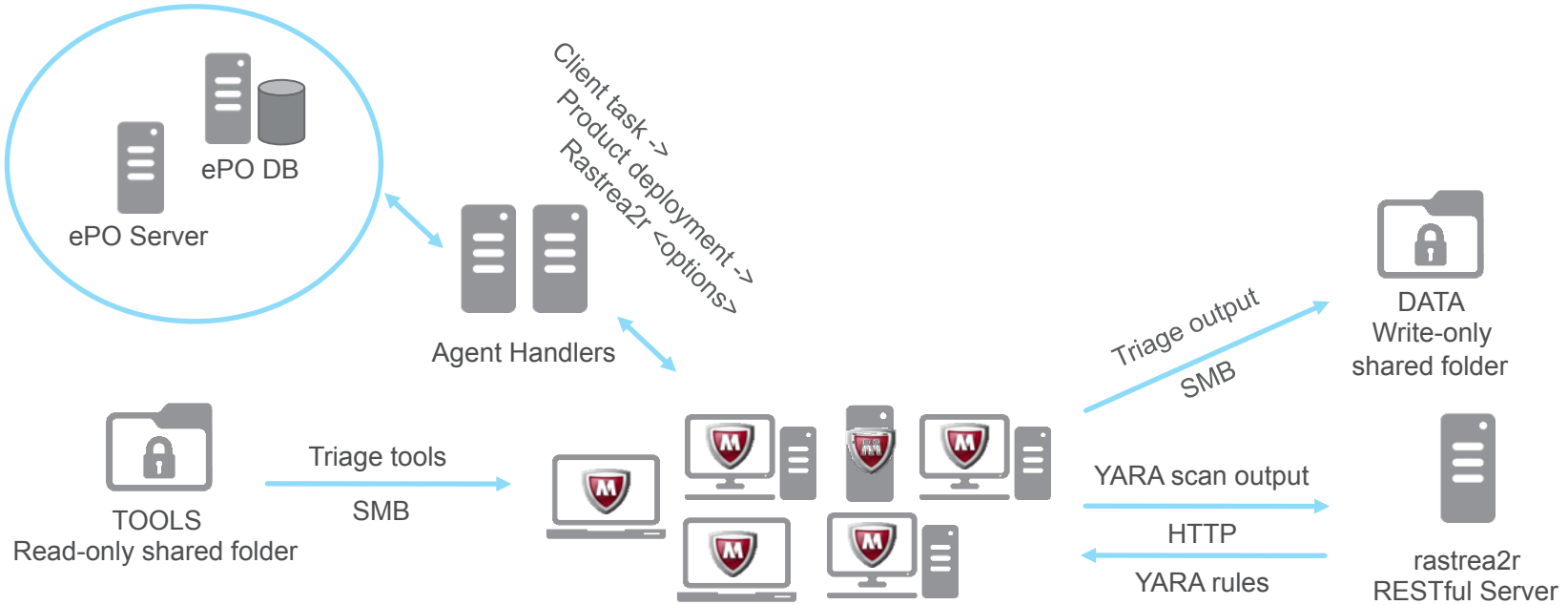
So How Do We Do All This?

Triage and Hunting for IOCs with 'gusto' and style ☺

- **Rastrea2r** (pronounced *rastreador*):
  - https://github.com/aboutsecurity/rastrea2r (opensource!)
  - Multiplatform (win32/64, linux and osx)
  - Uses a REST API to report **YARA** scans
  - Can run sysinternal, system command and other 3rd party tools remotely on endpoints
  - Easy to integrate with McAfee ePO and other consoles
  - Built in python (compiled binaries available)
- Current functionality in v0.6:
  - 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

Triage + Hunting on Steroids = rastrea2r

Forensically sound architecture and communication flows



Rastrea2r Command Line & Arguments

Coz command line is sexy!

rastrea2r_win32_v0.6.exe -h

```
C:\Users\user\Desktop\rastrea2r client\rastrea2r_win32_v0.6>rastrea2r_win32_v0.6.exe -h
usage: rastrea2r_win32_v0.6.exe [-h] [-v]
                                {yara-disk,yara-mem,memdump,triage} ...

::Rastrea2r RESTful remote Yara/Triage tool for Incident Responders by Ismael
Valenzuela @aboutsecurity / Foundstone (Intel Security)::

positional arguments:
  {yara-disk,yara-mem,memdump,triage}
                        modes of operation
    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
    triage              Collects triage information from the endpoint

optional arguments:
  -h, --help            show this help message and exit
  -v, --version         show program's version number and exit
```

Triaging with rastrea2r

Wrapper for 3rd party tools and native Windows commands

- Example:
  - rastrea2r_win32_v0.6.exe triage *tools.myserver.com data.myserver.com*

*** tools.myserver.com -> has a read only shared-folder called **TOOLS**

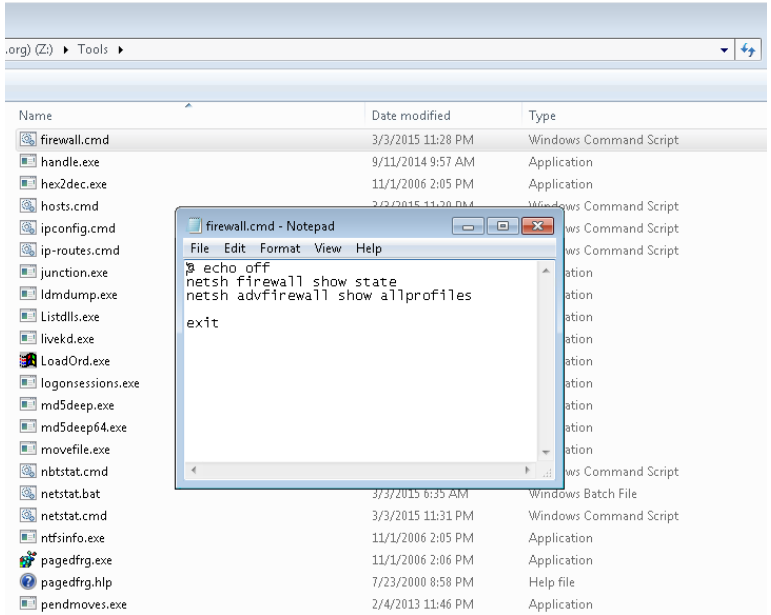
*** data.myserver.com -> has a write only shared-folder called **DATA**

```
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
```

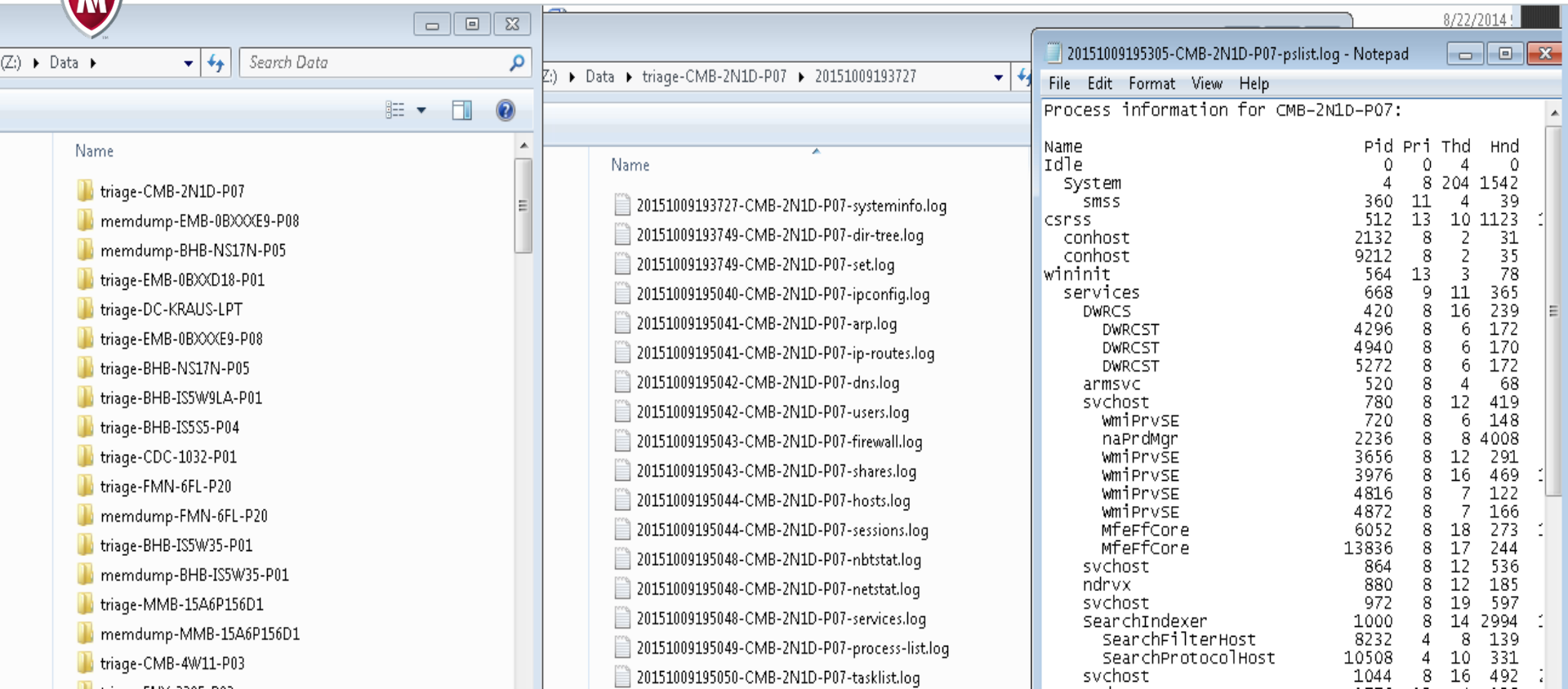
3rd Party Tools & Native Win commands

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:\ directory tree
     'ipconfig.cmd', # Gathers IP information
     'ip-routes.cmd', # Gathers IP routing information
     'arp.cmd', # Gathers ARP table information
     'dns.cmd', # Gathers DNS Cache information
     'users.cmd', # Gathers User/local Admin accounts
     'shares.cmd', # Gathers local shares information
     'firewall.cmd', # Gathers local firewall information
     'hosts.cmd', # Captures Host file information
     'sessions.cmd', # Gathers Active Session information
     'nbtstat.cmd', # Gathers NetBios Sessions information
     'netstat.cmd', # Gathers Netstat with process IDs
     'services.cmd', # Gathers services information
```



3rd Party Tools & Native Win commands



Memory Dumps with Rastrea2r

Only a click away from your McAfee ePO console

- Example:
  - rastrea2r_win32_v0.6.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**

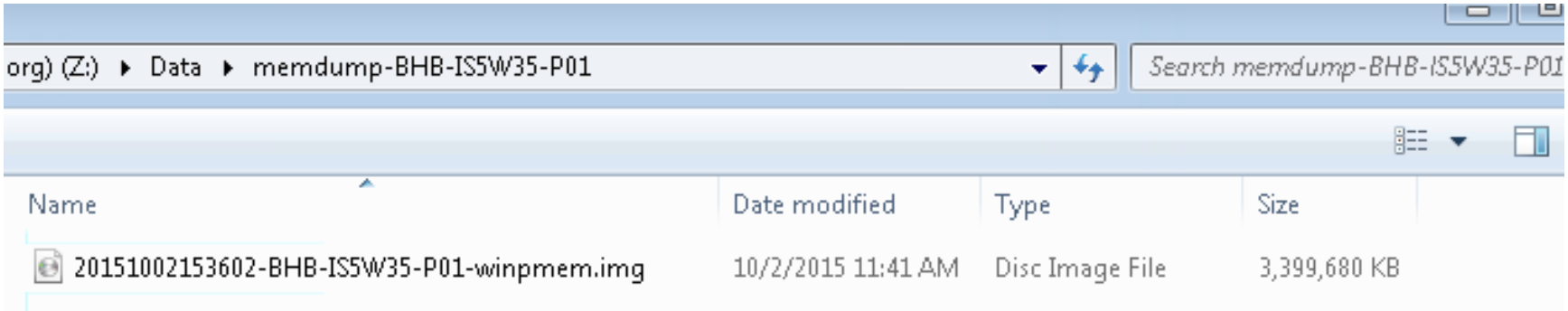
```
C:\Users\user\Desktop\rastrea2r client\rastrea2r_win32_v0.6>rastrea2r_win32_v0.6.exe memdump -h
usage: rastrea2r_win32_v0.6.exe memdump [-h] [-s] BIN_server DATA_server

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*

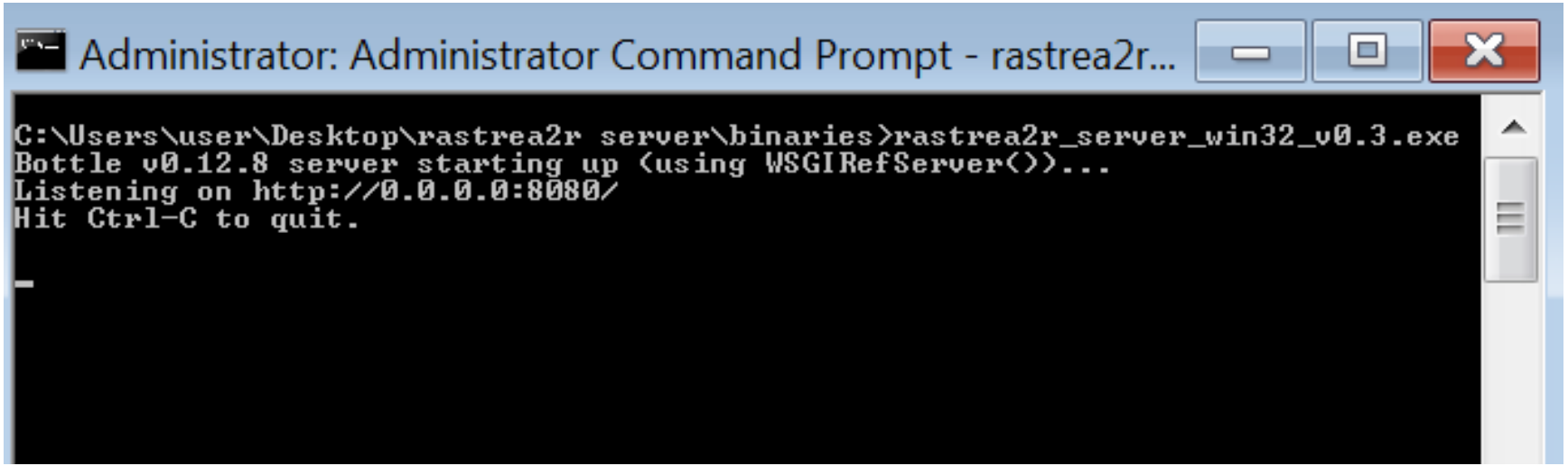
Full memory dump on RAW format:

- Ready to be parsed with memory analysis tools like volatility



Starting the rastrea2r server

Listening on all interfaces on port 8080



Hunting with Rastrea2r

IOC hunting in memory and disk a click away from your McAfee ePO console

*** Client / server architecture using a RESTful API

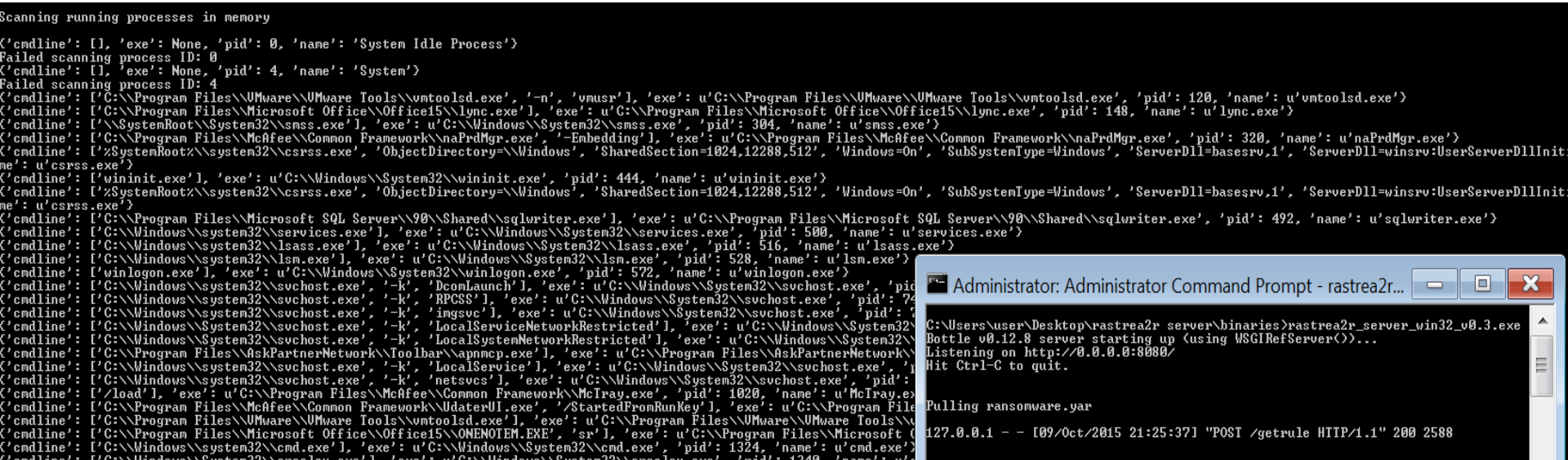
*** YARA rules must be stored on the same directory as the server

```
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:

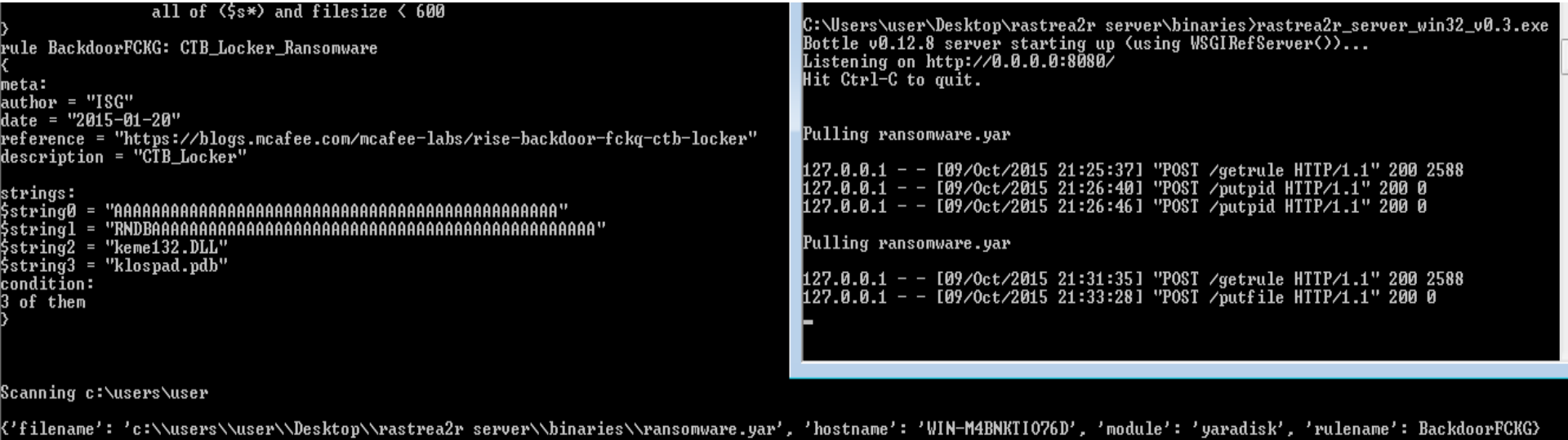
- rastrea2r_win32_v0.6.exe yara-mem *localhost ransomware.yar*



Hunting for IOCs on Disk with YARA

Example:

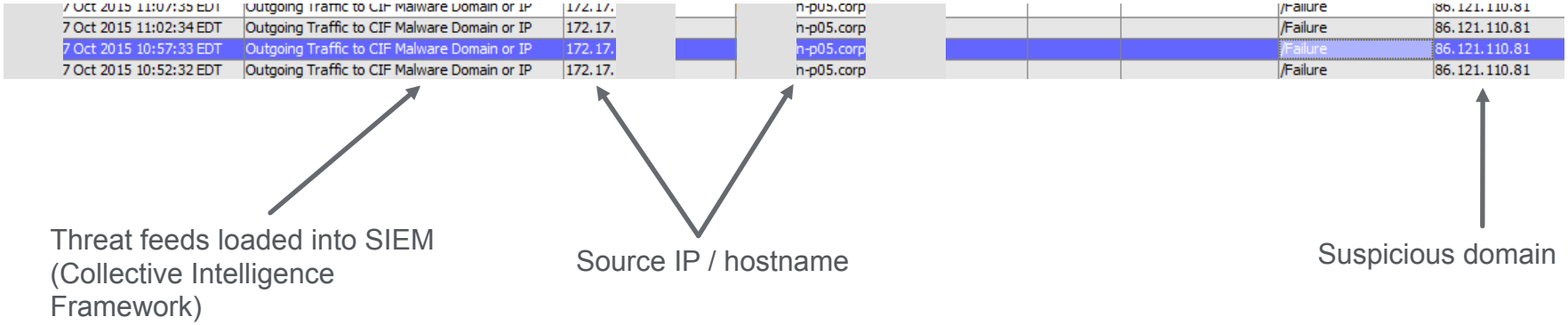
- rastrea2r_win32_v0.6.exe yara-disk c:\users\user *localhost ransomware.yar*



Rastrea2r in Action

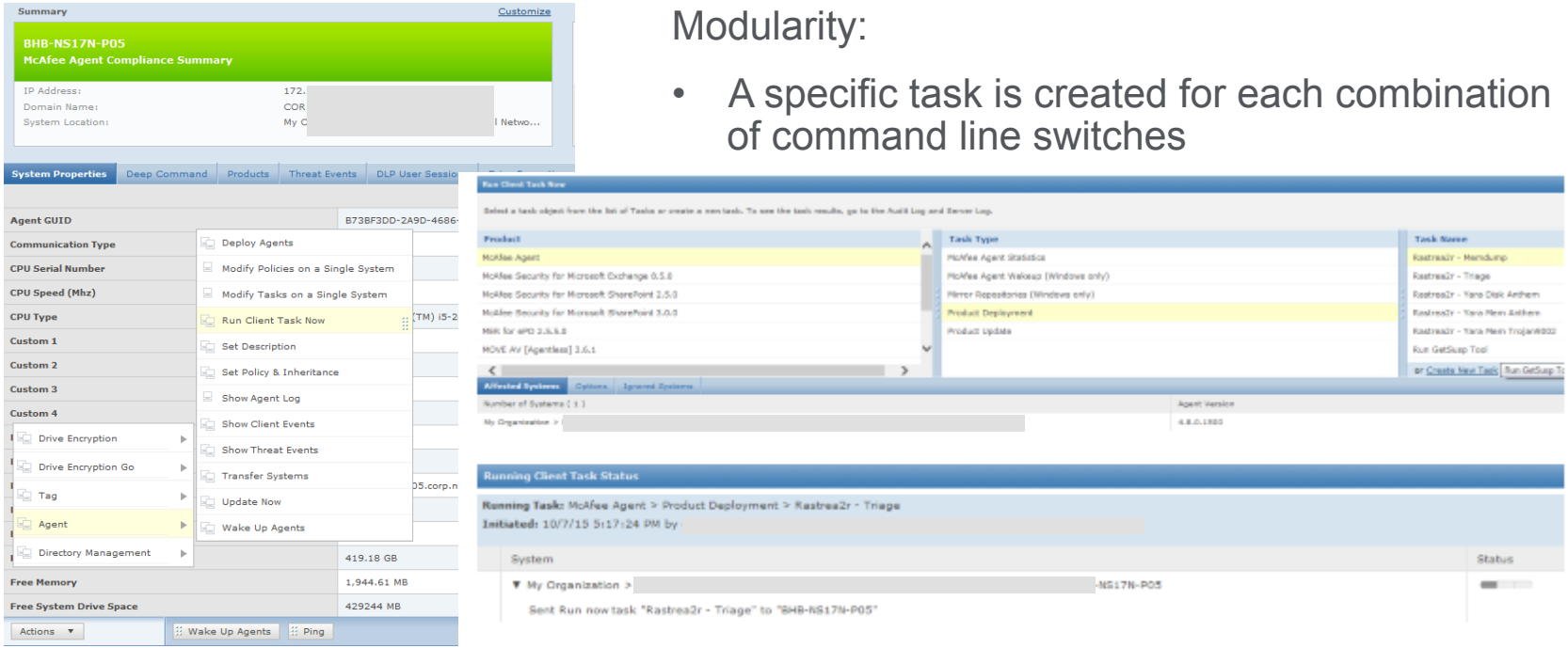
Let's revisit our event…

- Traffic blocked to a suspicious IP
  - What process is generating this traffic on the endpoint? And why?
  - Is there any other malicious activity on this host that is not being detected?
  - How do I respond to this? What should I do next?



Triaging with ‘rastrea2r’ from McAfee ePO Console

Client tasks -> product deployment -> rastrea2r triage/memdump

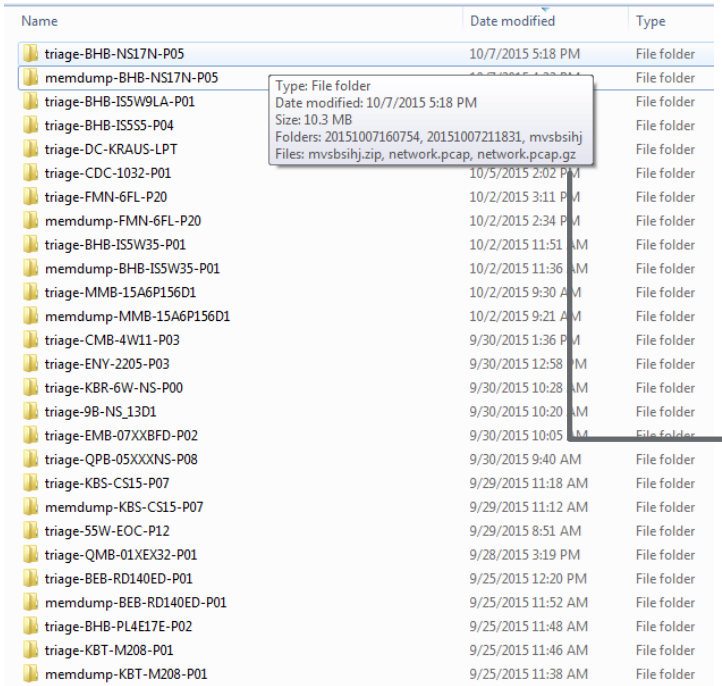


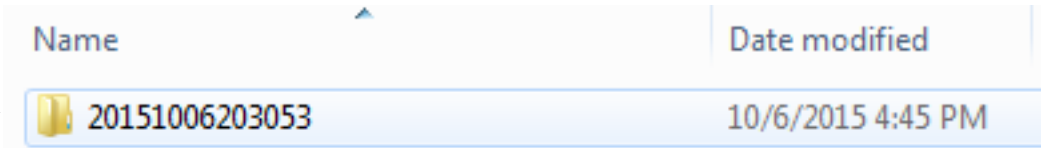
Modularity:

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

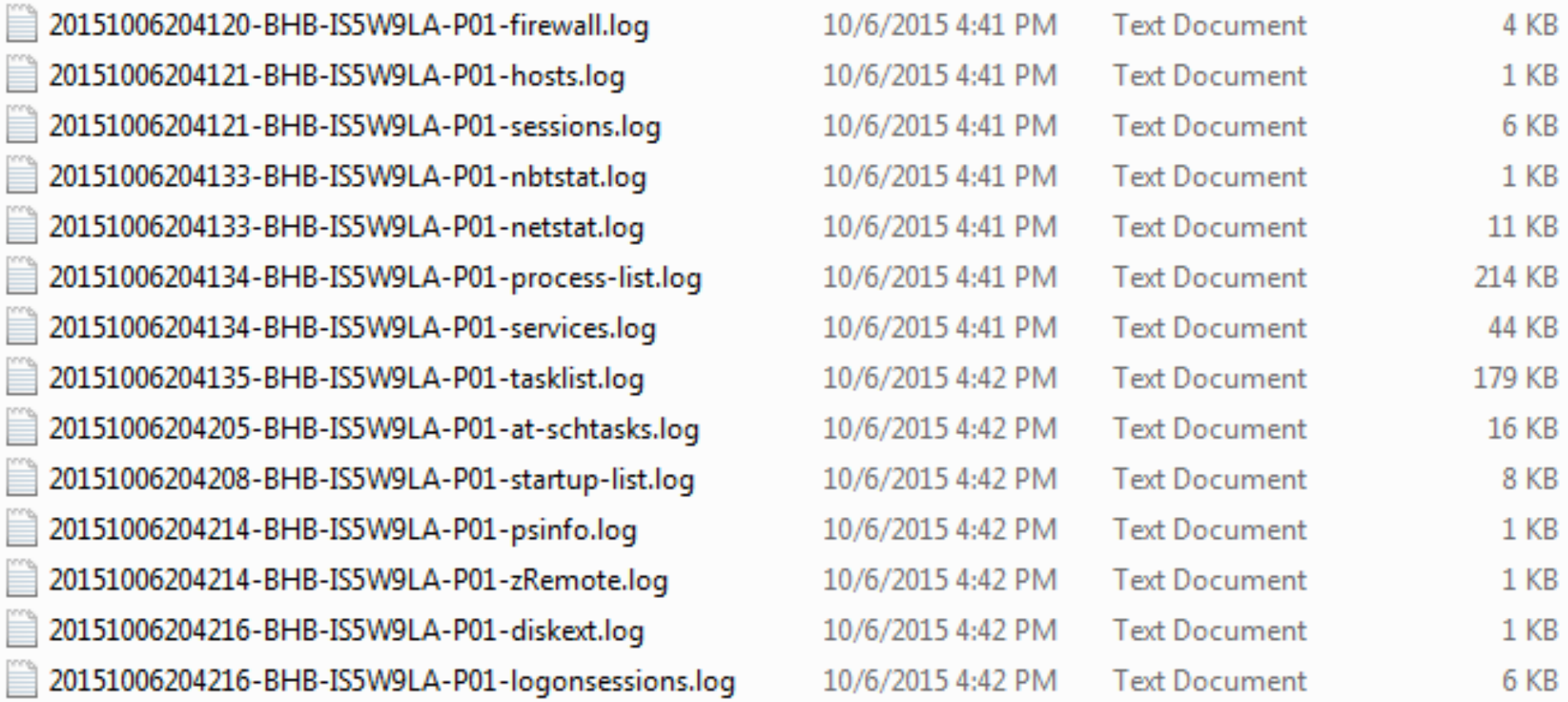
Analyzing the Results (~5 Minutes)

A folder is created per system, then per collection set (timestamp)





Analyzing the Results



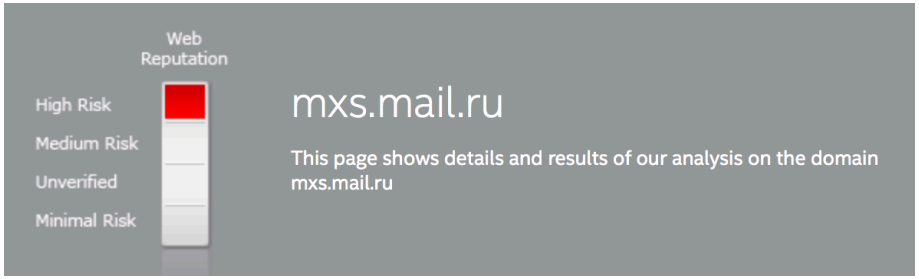
20151006204120-BHB-IS5W9LA-P01-firewall.log	10/6/2015 4:41 PM	Text Document	4 KB
20151006204121-BHB-IS5W9LA-P01-hosts.log	10/6/2015 4:41 PM	Text Document	1 KB
20151006204121-BHB-IS5W9LA-P01-sessions.log	10/6/2015 4:41 PM	Text Document	6 KB
20151006204133-BHB-IS5W9LA-P01-nbtstat.log	10/6/2015 4:41 PM	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
```

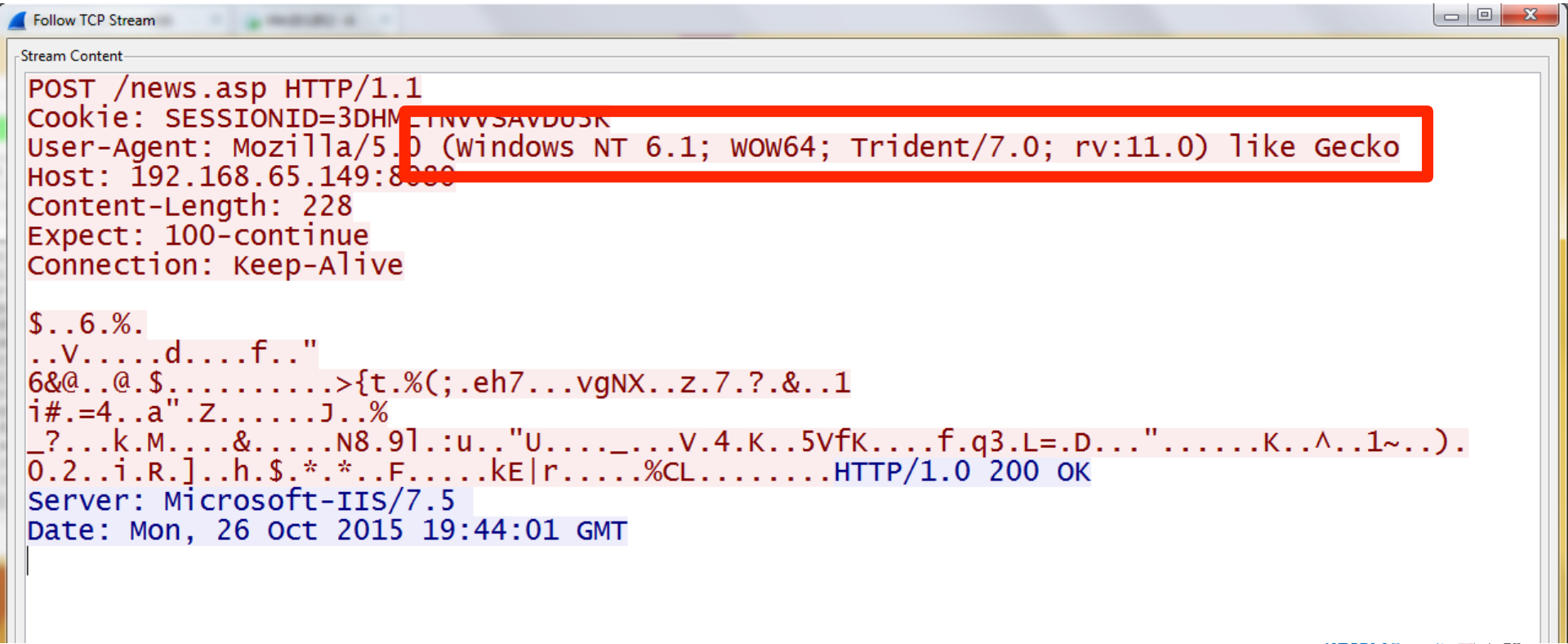
Someone Wants to Send Mail to Russia!

```
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
```



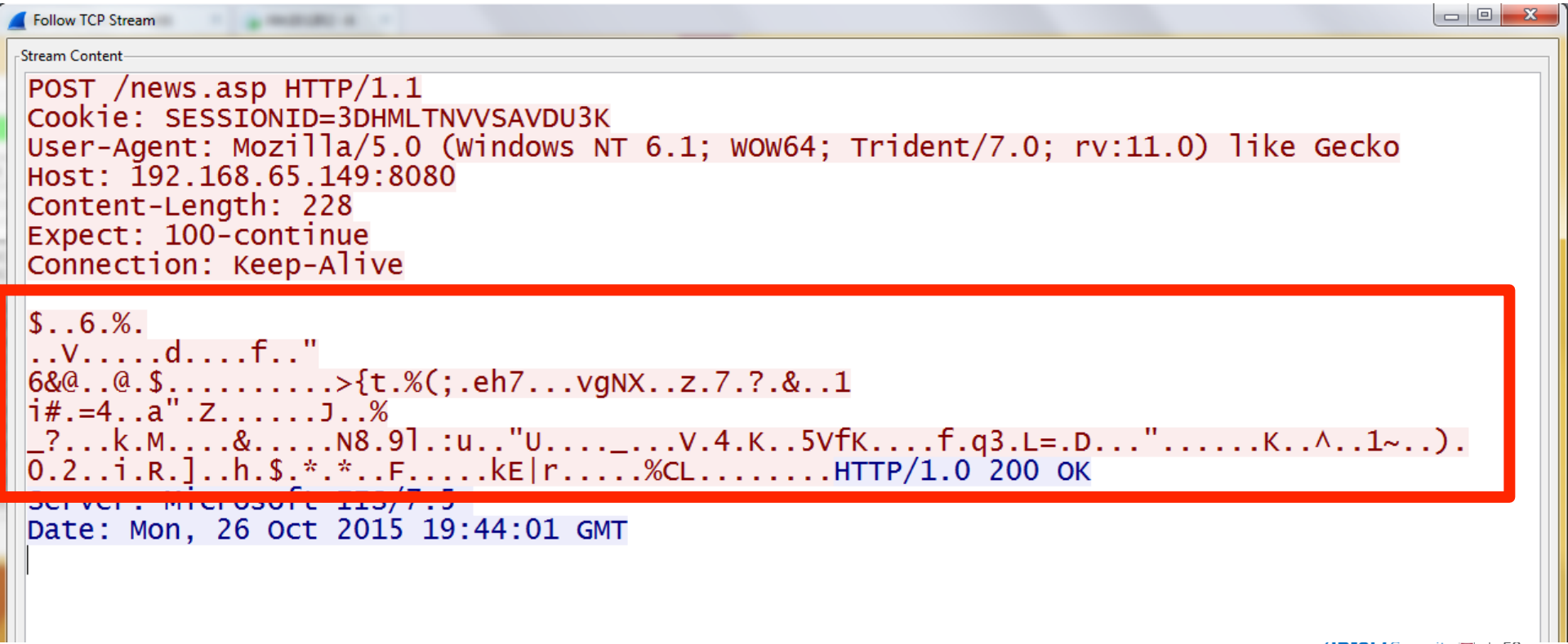
Network Behavior Analysis

This is a Windows 8.1 Host, user agent shows Windows 7!



Network Behavior Analysis

Encrypted Data over HTTP?



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.
  - Coin 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?

Hunting with YARA and rastrea2r

1. Create a new YARA rule based on this sample
2. Scan your environment using McAfee ePO + rastrea2r and triage infected systems
3. Tune your YARA rule based on your findings
4. Repeat, rinse and stop once the FULL scope is determined
5. Contain & eradicate

- 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, 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!

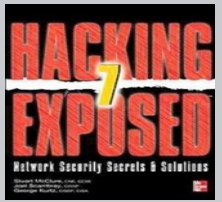
@aboutsecurity

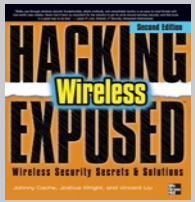
http://blog.ismaelvalenzuela.com

ismael.valenzuela@intel.com

Thought Leadership

Contributing authors to all editions of Hacking Exposed





Professors and Lecturers

Competition Judges/Mentors


Foundstone®


intel®
Security