Human Hunting

PRESENTED BY **Sean Gillespie** | April 20, 2015

Summary

- General Automation & Tools
- About the Adversary
- Preparations for Hunting
- Hunting Examples
- Q&A

Automation is Great…

- ID badge access using scanned & visual validation
- Complex logistics and battlefield management
- EMR patient validation for blood and meds
- Critical factors include skilled humans making final decisions

...Except When It Isn't

- Air France 447
- Asiana 214
- 2003 Northeast Blackout
- F-22 dateline failure
- Critical factors included over reliance on automation

A Tale of Two Systems

- Airbus tends to engineer towards a pilot serving a machine
- Boeing tends to engineer towards a machine serving a pilot
- This fundamentally alters behavior of plane and pilot

Shall We Play A Game

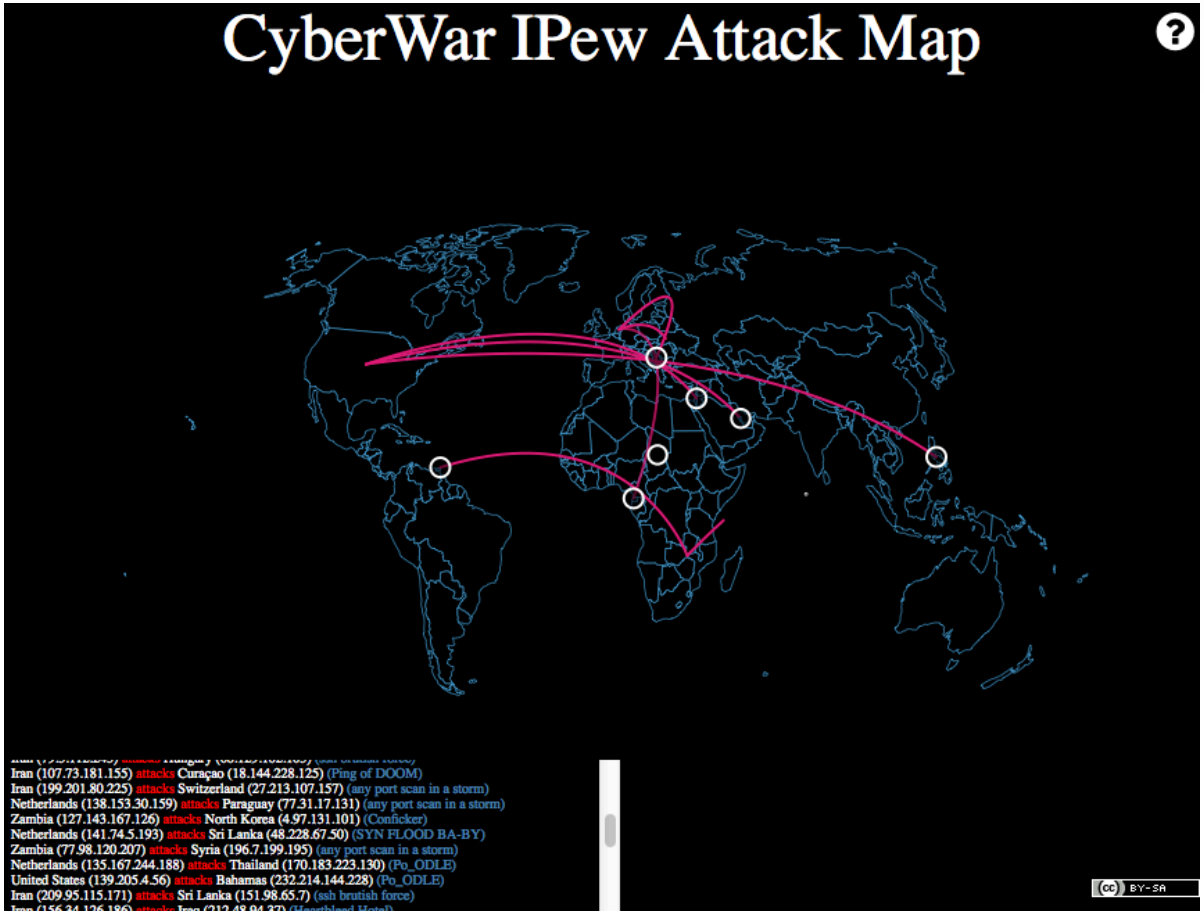
- We can build computers to compete with highly skilled Chess players
  - At substantial cost for the most successful systems
- We can’t build computers to compete with highly skilled Go players
- The more variables/branching the more humans outperform machines

And the Moral Is…

- Automation is good at doing the heavy lifting to enhance capabilities
  - Repetitive tasks
  - Exact sorting/categorization
  - Accurate calculations
  - Large data sets

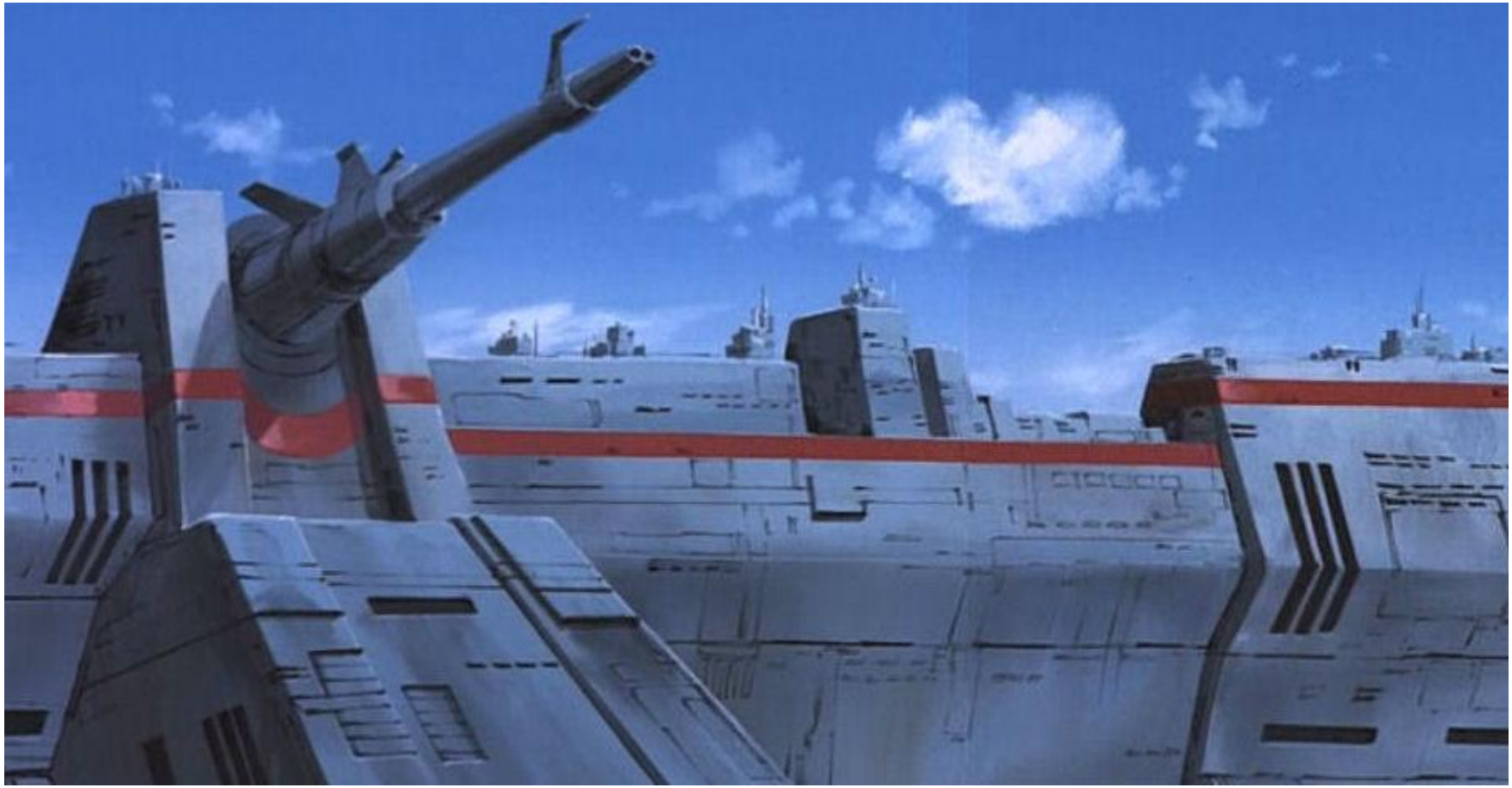
- Automation is bad at replacing highly skilled humans
  - Pattern recognition
  - Data interpretation
  - Fuzzy sorting/categorization
  - Bluffing & misdirection

CyberWar



YAHOO!

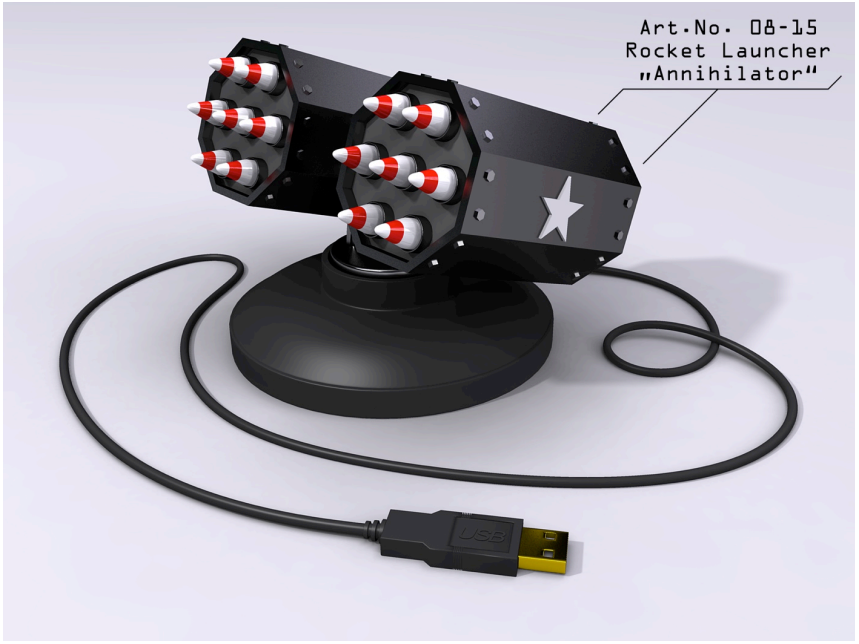
CyberBase



YAHOO!

CyberMissiles

Remote Attacks



Local Attacks

YAHOO!

CyberEnemies

APT1

YAHOO!

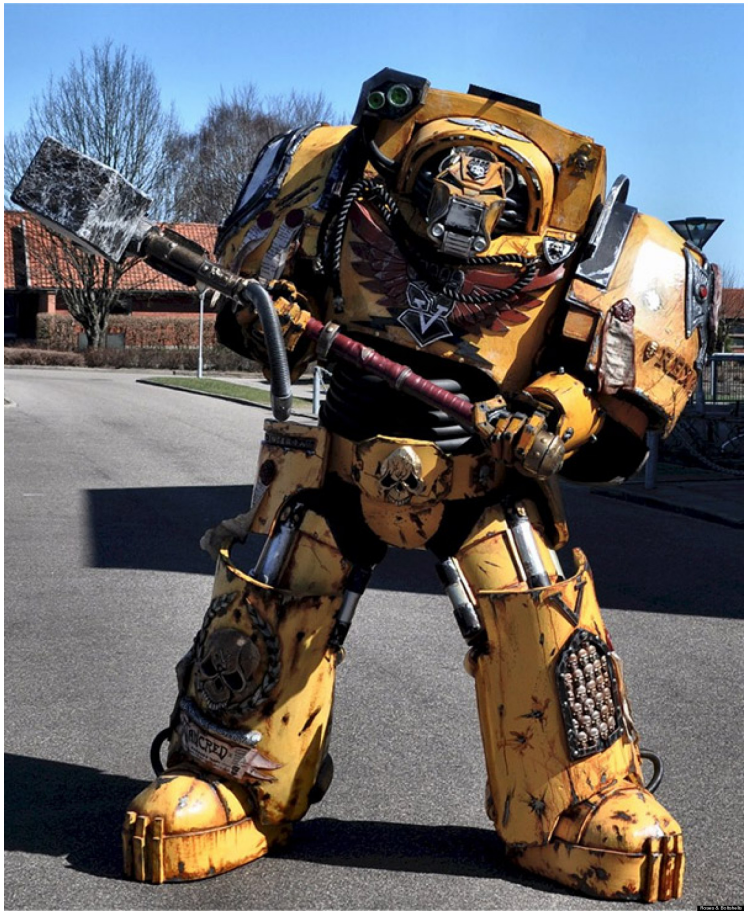
CyberWarriors

Data Marines

YAHOO!

CyberWeapons

Ban Hammer



YAHOO!

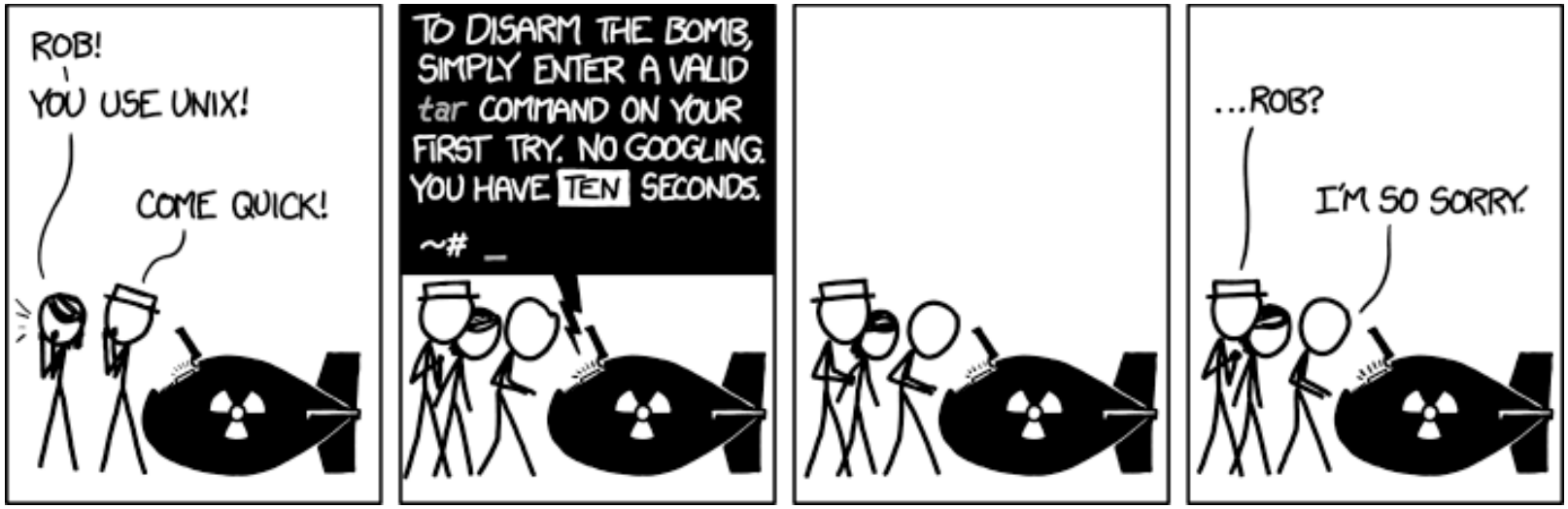
Automatic Real-time Detection Using Cloud Based Big Data Threat Intel Driven Machine Learning for Rapid Containment of Malicious Anomalies at Scale in Enterprise Networks

- Heavy vendor focus on magic black boxes that replace humans
- Very few with interfaces for analysts
- Over reliance on automation to find evil

YAHOO!

Do --You --Have --A --Flag?

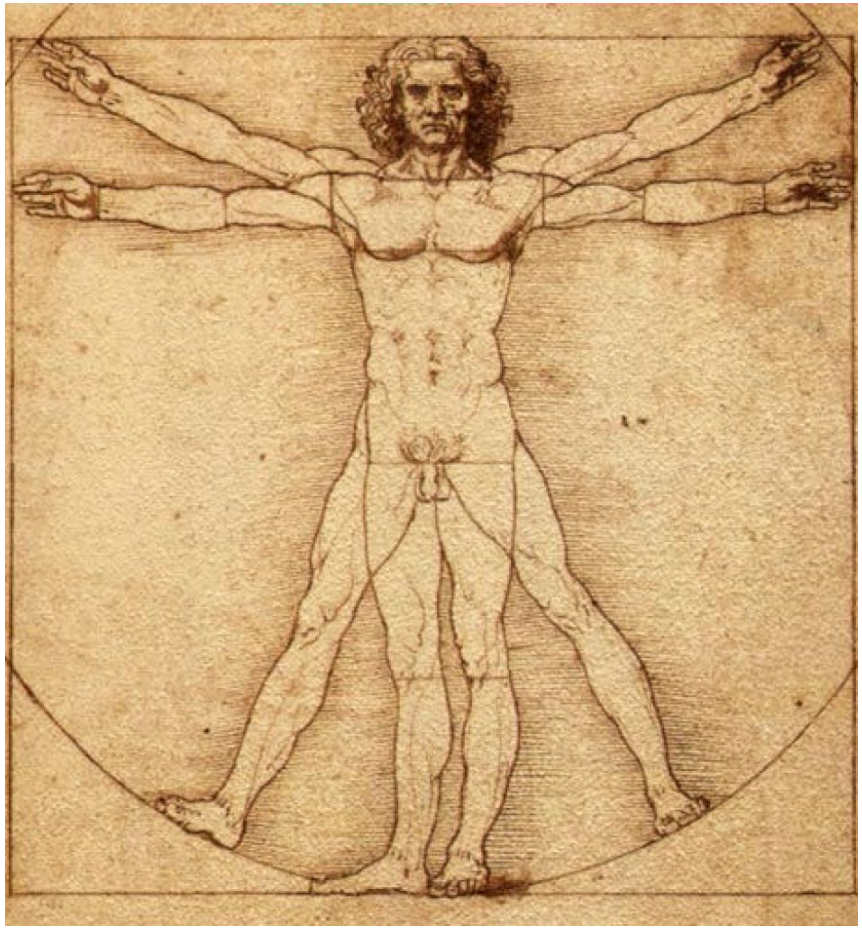
- Community builds a lot of excellent tools
- Often overly complex usage with minimal documentation
- Tools rarely scale well



YAHOO!

The Adversary – The Most Dangerous Game

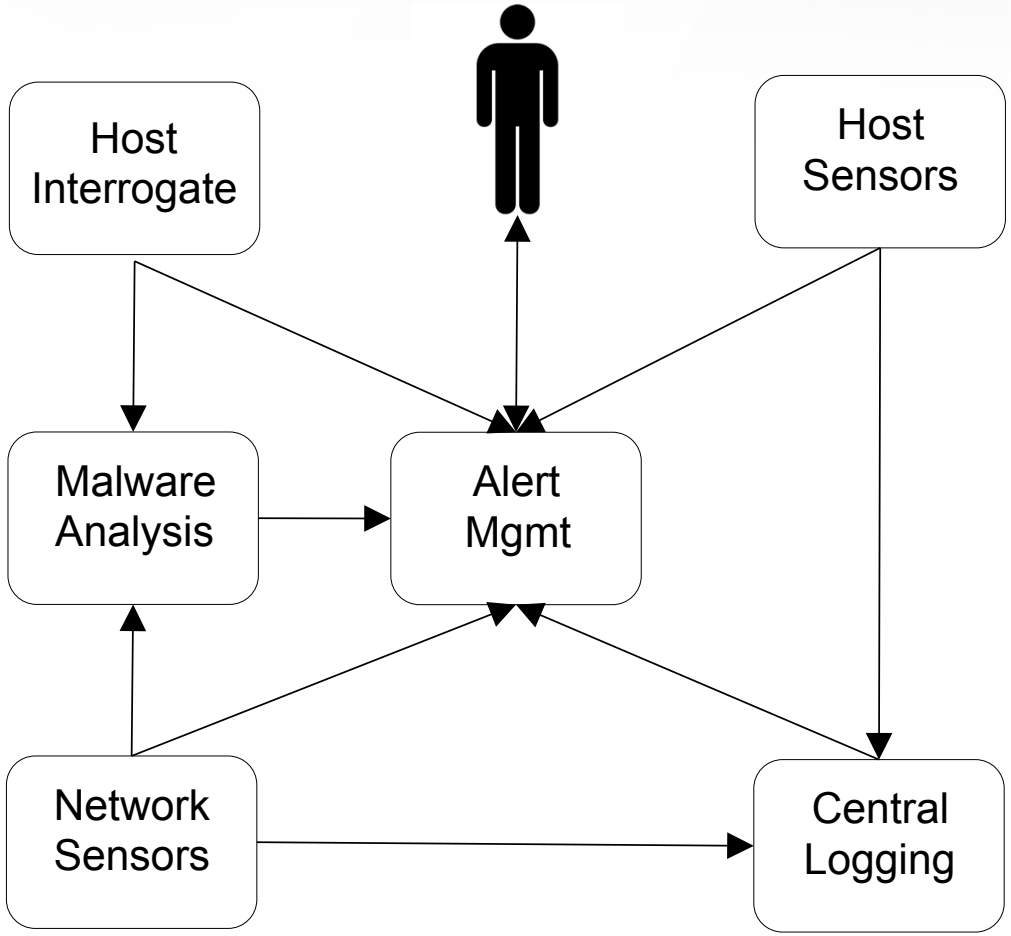
- Very creative
- Very adaptive
- Highly skilled
- Highly focused



YAHOO!

What You Need to Hunt

- Collection Systems
  - Event logging
  - Host sensors & interrogation
  - Network sensors
- Alert Management
  - Severity classification
  - Resolution status
  - Notes
- Threat Research
  - Books & reports
  - Sandbox analysis
  - Threat Intel data



YAHOO!

Human Adversary – Human Defender

- Alert validation
- Effective response
- Predict emerging threats

YAHOO!

Collection Systems

Stream Capture

- Captures state changes
- Ongoing timeline of events
- Good for investigations
- Good for signatures/searches
- Large volumes of data to handle

State Capture

- Captures present state
- Further historical reach of data
- Good for discovery of new threats
- Good for forensics
- Gaps in timeline

Alert Management

- Object Oriented Alerting
- UIs should be tailored for data type
  - Relatively simple and intuitive
  - Allow searching & pivoting
  - Stacking similar data/alerts
  - Data tagging & reporting
- Examples: Snorby & ePO (With proper dashboards)

Threat Research

- Books
  - System operations
  - Forensics & malware
- Analysis lab
  - Malware analysis
  - Protocols
  - Techniques
- Threat Intelligence
  - Reports
  - Malware databases
  - Intel sharing/feeds
- Red vs Blue practice

Host Interrogation With GRR & ForensicArtifacts

- GRR Rapid Response is an incident response framework focused on remote live forensics.

- https://github.com/google/grr
- https://github.com/google/rekall
- https://github.com/sleuthkit/sleuthkit
- https://github.com/ForensicArtifacts/artifacts

Hunting with GRR

- State capture type collection system
- State machine based operation
- Cross-platform
- Scriptable
- Server side parsing
- Limited client side filtering

GRR Overview

- Flow – The unit of work for GRR. Flows can call a sequence of client actions, processes results, perform server maintenance, or reporting tasks. Flows are written in python and stored on the server.

- Hunts – Mechanism for running a Flow across a fleet of clients. When a client checks it will be evaluated against the criteria of the Hunt. Scheduling is determined by rules, client rate, client limit, and hunt expiration.

- Artifact – Yaml defined “point of interest” for forensics. Examples include crontab files or Windows RunKeys.

Interesting or Common Flows

- Interrogate
- ArtifactCollectorFlow
- FileFinder
- RegistryFinder
- AnalyzeClientMemory
- MemoryCollector
- Netstat
- ListProcesses
- LaunchBinary

Preparations & Process

- Create an Artifact to collect your Live Response data from a host
- Create Hunt with that Flow that only applies to an “LR” label
- Create Hunts with hunting Flows
- Validate Hunt results and apply “LR” label to any suspected hosts

What to Hunt

- Process
  - Bad parents
  - Unique processes
  - Open connections
- Registry
  - Persistence
  - Authentication manipulation
- Services & Daemons
  - Wrong configuration
  - Unique services
- Files
  - Obfuscated files
  - Suspicious placement
- Scheduled Tasks & Cronjobs

What to Hunt

- Threat Research
- Think about evasion
- Think about evidence
- Generalize specific attacks into solid indicators for hunting

How to Hunt

- Use generalized Artifacts
  - Bulk collection where possible
  - Filtered collection where impractical
- Use Artifacts with parsers when available
- Use API or console to extract results
- Stacking & Filtering

The Hunt Begins

- WindowsCommonFilePlacementAttacks
- WindowsRunKeys
- OSXLaunchDaemons

File Placement

```
name: WindowsCommonFilePlacementAttacks
doc: Common files associated with search order hijacking and other file placement attacks
collectors:
- collector_type: FILE
  args:
    path_list:
      - '%%environ_systemdrive%%\program.exe’
      - '%%environ_systemroot%%\System32\oci.dll’
      - '%%environ_systemroot%%\ntshrui.dll'
      - '%%environ_systemroot%%\System32\sysprep\cryptbase.dll’
      … list truncated to use a visible font size
supported_os: [Windows]
```

Stacked Results

```
   1 /C:/Windows/System32/oci.dll,1015808
   1 /C:/Windows/System32/sysprep/CRYPTBASE.dll,81408
   2 /C:/Windows/System32/Utilman.exe,785920
   2 /C:/Windows/System32/sethc.exe,776192
   7 /C:/Windows/System32/Utilman.exe,82432
   7 /C:/Windows/System32/sethc.exe,270336
  37 /C:/Windows/System32/Utilman.exe,81920
  37 /C:/Windows/System32/sethc.exe,268288
5018 /C:/Windows/System32/Utilman.exe,1402880
5018 /C:/Windows/System32/sethc.exe,279040
```

Registry Keys

```
name: WindowsRunKeys
collectors:
-   collector_type: REGISTRY_KEY
-   args:
    path_list:
      - 'HKEY_USERS\%%users.sid%%\Software\Microsoft\Windows\CurrentVersion\Run\*'
      - 'HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run\*'
      … list truncated to use a visible font size
labels: [Software]
supported_os: [Windows]
```

Stacked Results

1 C:\Users\aaaa\AppData\Local\Vidyo\Vidyo Desktop\VidyoDesktop.exe
1 C:\Users\bbbb\AppData\Local\Vidyo\Vidyo Desktop\VidyoDesktop.exe
1 C:\Users\cccc\AppData\Local\Vidyo\Vidyo Desktop\VidyoDesktop.exe
1 C:\Users\dddd\AppData\Local\Vidyo\Vidyo Desktop\VidyoDesktop.exe
1 C:\Users\eeee\AppData\Local\Vidyo\Vidyo Desktop\VidyoDesktop.exe
1 C:\Users\ffff\AppData\Local\Vidyo\Vidyo Desktop\VidyoDesktop.exe
1 C:\Users\gggg\AppData\Local\Vidyo\Vidyo Desktop\VidyoDesktop.exe
1 C:\Users\hhhh\AppData\Local\Vidyo\Vidyo Desktop\VidyoDesktop.exe
…

egrep "[A-Z]:\\\\[A-Za-z0-9]+:[A-Za-z0-9\.]+"

regsvr32 /s "C:\Temp:1A2A61B8.dat"

regsvr32 /s "C:\Temp:02357D83.dat"

aff4:/<CLIENT_ID>/registry/HKEY_USERS/<SID>/Software/Microsoft/Windows/CurrentVersion/Run/svchost

egrep “vb[se]”

wscript.exe //B "C:\Users\aaaa\AppData\Local\Temp\dopgglqblq..vbe”
wscript.exe //B "C:\Users\bbbb\AppData\Local\Temp\myphyoljsc..vbs”
wscript.exe //B "C:\Users\cccc\AppData\Local\Temp\fklxtwwwbv..vbs”
wscript.exe //B "C:\Users\dddd\AppData\Local\Temp\ozkvgpcbtp..vbs”
wscript.exe //B "C:\Users\eeee\AppData\Local\Temp\krqvsfjxld.vbs”
wscript.exe //B "C:\Users\ffff\AppData\Local\Temp\fyzbnaksvu..vbs”
wscript.exe //B "C:\Users\gggg\AppData\Local\Temp\gsiiwnbhjk.vbs”

aff4:/<CLIENT_ID>/registry/HKEY_USERS/<SID>/Software/Microsoft/Windows/CurrentVersion/Run/gsiiwnbhjk

File Content

```
name: OSXLaunchDaemons
doc: Mac OS X Launch Daemons files.
collectors:
- collector_type: FILE
  args:
    path_list:
      - '/Library/LaunchDaemons/*'
      - '/System/Library/LaunchDaemons/*'
      - '%%users.homedir%%/Library/LaunchDaemons/*'
labels: [System]
supported_os: [Darwin]
urls: ['http://www.forensicswiki.org/wiki/Mac_OS_X']
```

Stacking & grep “apple”

1 /Library/LaunchDaemons/com.apple.globalupdate.plist

Program: /usr/bin/globalupdate

1 /Library/LaunchDaemons/com.apple.machook_damon.plist

Program: /usr/local/machook/machook

WatchPaths: /usr/local/machook/watch.sh

Stacking & grep “apple”

/usr/bin/globalupdate – Missing File

/usr/local/machook/watch.sh –

```
jobid=`ps -ef |grep machook |grep -v "grep" |wc -l`
if [ $jobid -eq 0 ];then
        /usr/local/machook/machook  &
fi
updateid=`ps -ef |grep globalupdate |grep -v "grep" |wc -l`
if [ $jobid -eq 0 ];then
        /usr/bin/globalupdate  &
fi
```

/usr/local/machook/machook – Wirelurker MD5

Coming Soon…ish

- More parsers
- Better searching based on file type/contents
- Better filtering for Artifacts/Flows
- Better tools for working with GRR data
  - Tagging
  - Filtering
  - Comments
  - History

Advanced Hunting

- Process execution/AV logs fueling binary or LR capture
- Beacon detection or network alerts trigger investigative Flows
- Automatic delivery of acquired files to malware analysis

FIN

- We need more automation that enhances human capabilities
- We need more focus on hunting/alert validation skills
- We need higher quality threat data and sharing
- We need to stop looking for magic unicorns

YAHOO!

Q&A
@pidydx