Learning Normal with the Kansa PowerShell IR Framework

Jason Simsay
April 2017
GIAC GSEC GCIH GCIA GCUX GCPM

Objective

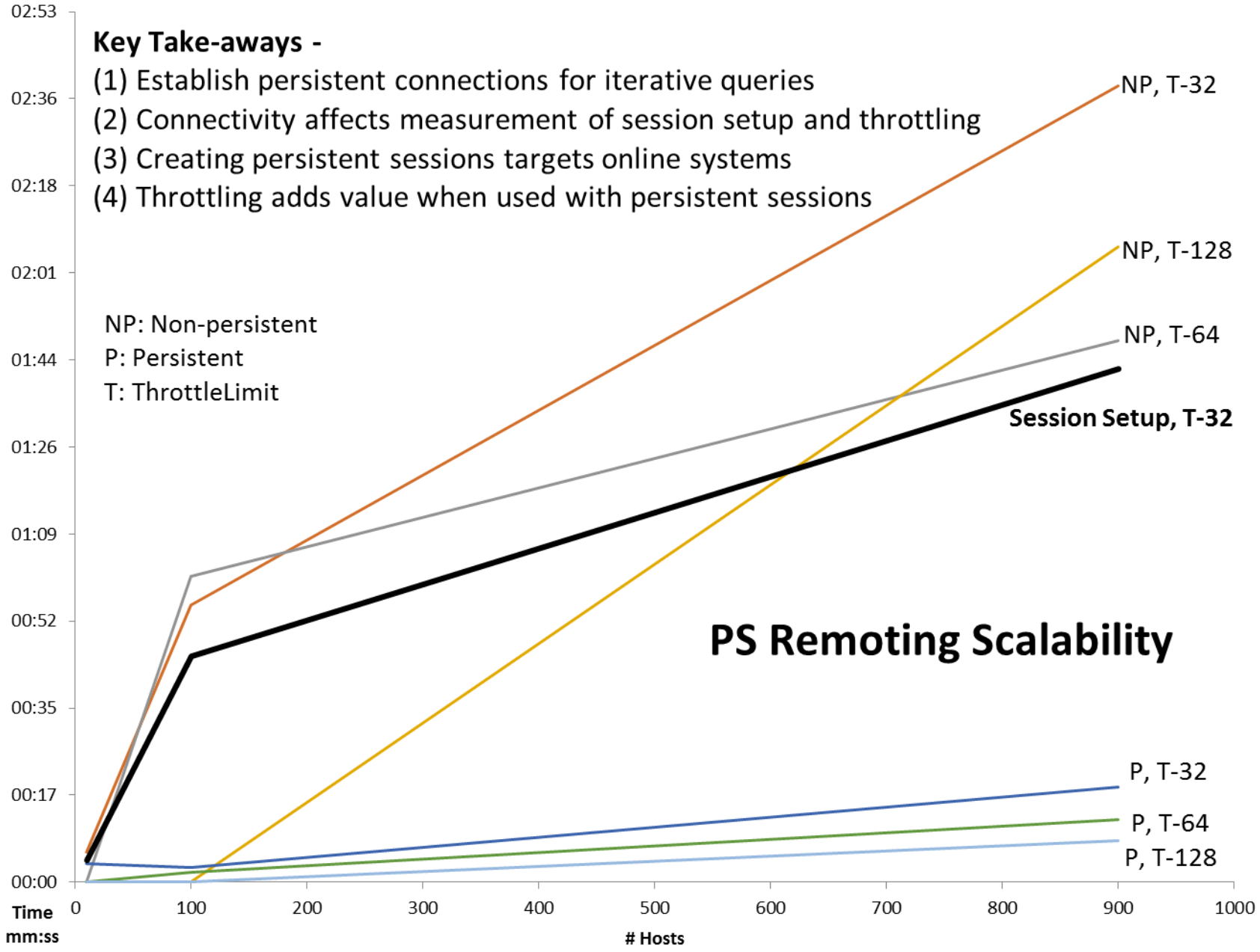
- Prepare now to respond effectively
  - Arm yourself with capability (tools)
  - Know your environment (baseline)
  - Patrol your network (unknowns)
- Frequency analysis, what is normal?
  - Kansa: rapid data acquisition and analysis
  - Kansa-Profiler: profile systems, establish known baseline, and identify deltas

Incident Response

- How do we respond to incidents?
  - When do we not reimage?
  - Take the system(s) offline?
- Preparation imperative, ready to go
  - Skills – hire professional services for breach
  - Resources – organization needs to equip
- Key capability – immediate response to ad-hoc searches across the enterprise

Kansa PowerShell IR

- Scalable, parallel data collection across a Windows environment with PSRemoting
- Collect data, conduct frequency analysis
  - Low frequency - unknown
    - "malware artifacts tend to be relatively unique"
    - "over 30 Windows system artifacts to search for IOCs"
  - High frequency – normal
    - Consistent presence amongst homogeneous systems
    - This is part of the known good system baseline


Key Take-aways -
(1) Establish persistent connections for iterative queries
(2) Connectivity affects measurement of session setup and throttling
(3) Creating persistent sessions targets online systems
(4) Throttling adds value when used with persistent sessions
NP: Non-persistent
P: Persistent
T: ThrottleLimit
NP, T-32
NP, T-128
NP, T-64
Session Setup, T-32
PS Remoting Scalability
P, T-32
P, T-64
P, T-128
02:53
02:36
02:18
02:01
01:44
01:26
01:09
00:52
00:35
00:17
00:00
Time
mm:ss
0
100
200
300
400
500
600
700
800
900
1000
Hosts

WinRM

- Web Services Management (WS-Man)
- Windows Remote Management
  - Event log forwarding/collection
  - Remote shell
- Powerful tools for good and bad
  - Secure service: firewall, Kerberos only
  - PowerShell logging: WMF version 5

All this data…

- Incident response – low frequency
- System baselining – high frequency
  - System profiles vary based on function
  - Analyze subsets of similar systems
  - Isolate similar systems using system and organizational properties

Kansa-Profiler (1)

- https://github.com/dry-fly/Kansa-Profiler
- A Kansa fork for system baselining
  - Modules to collect system properties
  - Build system properties database
  - Group and filter and isolate systems based on the characteristics of the system profile
  - Establish a profile directory linked to Kansa data files – scope analysis here

Kansa-Profiler (2)

- Optional reuse of Kansa Get-Analysis
- Directory structure mirrors that of Kansa
- Theoretically: consistent, fewer outliers
- Actuality: it depends on the data point and the filtering and grouping decisions
  - not baseline, but baselines
  - analyst's job is harder and takes longer than expected [insert look of total surprise]

createProfileDirectory.ps1

- Import records from profilingDatabase.csv
- Acquire user input for grouping and filtering
- Get filtered hosts from group property
- Identify hosts that meet criteria/user elect to proceed
- Create profile directory and identify Kansa data dirs
- For each data directory
  - Enumerate files, if match filtered host
  - { Create symbolic link to Kansa data file for host }
- If Analysis parameter specified { Call Get-Analysis }

Get-TasklistV Uniqueness

Grouping Property	Profile Group	Hosts	Unique Tasks	Intersection	Group Unique	Common Tasks	Common/ Unique
All Hosts		823	459			10	2.20%
Architecture	32-bit	337	217	180	37	14	6.50%
	64-bit	486	422	180	242	10	2.40%
Manufacturer	Dell	735	390	42	242	12	3.10%
	Microsoft	72	192	42	41	16	8.30%
	OEMC	6	84	42	0	47	56.00%
	VMWare	10	69	42	16	25	36.20%
Windows Version	Windows 7	740	407	95	254	11	2.70%
	Windows 8.1	68	150	95	13	17	11.30%
	Windows 10	15	160	95	29	13	8.10%
All Dell Systems		733	385			13	
Model	Latitude	61	254	170	80	29	11.40%
	OptiPlex	672	308	170	135	18	5.80%
All Dell Optiplex7010 Win7		219	232			17	
Model/Win7/ Org Unit	BackOffice	117	217	91	105	18	8.30%
	Platforms	46	119	91	9	24	20.20%
	Tellers	55	100	91	5	17	17.00%

Important Baselines

- SANS Intrusion Discovery Cheat Sheet
- Command Line -> Kansa module
  - Sc.exe query state=all -> Get-SvcAll
  - Net localgroup administrators -> Get-LocalAdmins
  - Netstat –nao -> Get-Netstat
  - Schtasks -> Get-AutorunscSchtasks
  - Tasklist /svc -> Get-Tasklistv
  - Wmic process list full -> Get-ProcsWMI
  - Dir c:\ -> Get-WMILogicalDisk

Finding New Unknowns

- Compare the current state of systems to the last known good baseline
  - Transition from IR preparation to detection
  - Operationalize continuous monitoring
- Security analyst interested in:
  - New low frequency data items
  - Which host(s) account for new item

findUnknowns.ps1

- List output directories, prompt user for baseline and comparison directories
- Identify matching profile directories, prompt user for profile to be analyzed
- Identify matching analysis results files, prompt user to pick module analysis to compare
- Import the analysis result of interest for each data set to a PowerShell object
- Compare-object on the NoteProperty property of the PowerShell object that is not named 'CNT' and output those not equal

Summary

- Incidents are unique, you don't know what you will be looking for
- Baselining will establish a familiarity with normal and provide a point of reference
- Establishing a secure infrastructure for rapid data collection is worthwhile prep
- Operationalize the analysis/disseminate the normal state/identify the unknown

References (1)

- Adams, Robert. (2015, December 7). The power and implications of enterprise incident response with PowerShell. Retrieved from https://www.sans.org/reading-room/whitepapers/incident/power-implications-enabling-powershell-remoting-enterprise-36542
- Cichonski, P., Millar, T., Grance, T., & Scarfone, K. (2012, August). National Institute of Standards and Technology. Special Publication 800-61 Revision 2. Computer security incident handling guide. Retrieved from http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf
- Hallenbeck, Chris. (2016, March 29). Avoiding incident response groundhog day. Retrieved from https://blog.tanium.com/avoiding-incident-response-groundhog-day/
- Hull, Dave. (2014, April 14). [Web blog comment]. Retrieved from http://trustedsignal.blogspot.com/search/label/Kansa
- Hull, Dave. (n.d.) Kansa. Readme.MD Retrieved from https://GitHub.com/davehull/Kansa
- Hull, Dave. (2014, July 18). Kansa: A PowerShell-based incident response framework. Retrieved from http://www.powershellmagazine.com/2014/07/18/kansa-a-powershell-based-incident-response-framework/
- Hull, Dave. (2011, April 23). [Web blog comment]. Retrieved from http://digital-forensics.sans.org/blog/2011/04/23/digital-forensics-least-freq-strings

References (2)

- Nair, Sajeev. (2013, August 7). Live response using PowerShell. Retrieved from https://www.sans.org/reading-room/whitepapers/forensics/live-response-powershell-34302
- National Security Agency. (2013, December 16). Spotting the adversary with Windows event log monitoring. Retrieved from https://www.iad.gov/iad/customcf/openAttachment.cfm?FilePath=/iad/library/ia-guidance/security-configuration/applications/assets/public/upload/Spotting-the-Adversary-with-Windows-Event-Log-Monitoring.pdf&WpKes=aF6woL7fQp3dJiwesfNwhEgT5nbAuQVRBwKBfN
- Skoudis, E., Strand, J., & SANS. (2015). SANS hacker tools, techniques, exploits & incident handling. (Vol. 1).
- Tilbury, Chad. (2010, November 08). Digital forensics how-to: memory analysis with Mandiant Memoryze. Retrieved from https://digital-forensics.sans.org/blog/2010/11/08/digital-forensics-howto-memory-analysis-mandiant-memoryze/
- Verizon. (2016). 2016 Data breach investigations report. Retrieved from http://www.verizonenterprise.com/resources/reports/rp_DBIR_2016_Report_en_xg.pdf