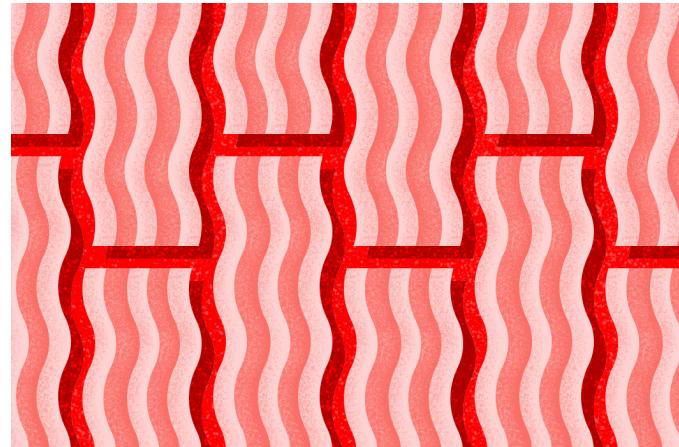


Getting the Bacon from the Beacon

September 29, 2020 | Kareem Hamdan and Lucas Miller | From The Front Lines



In recent months, [CrowdStrike® Services](#) has observed a continued increase in the use of Cobalt Strike by eCrime and [nation-state adversaries](#) to conduct their operations following the initial access to victims' environments.

Cobalt Strike is a commercially available post-exploitation framework developed for

[Featured](#)

[Recent](#)

[Video](#)

[Category](#)

[Start Free Trial](#)

the memory space of a compromised system, typically leaving minimal on-disk footprints. This blog discusses CrowdStrike's research and testing of Cobalt Strike's Beacon in an isolated Active Directory domain to identify host-based indicators generated from the



environments, aiding in the positive identification of remnants of Beacon execution.

Beacon Behavior Summary

Adversaries often execute a variety of Beacon commands once they establish a foothold within an environment. Beacon commands can be used to spawn other Beacons on additional systems accessible to the initial Beacon, effectively furthering persistence in the target environment. Beacons can also be leveraged for remote access and execution.

- The execution of the commands highlighted in this blog will generate a variety of Windows security events depending on the context of the command: The Beacon commands `jump psexec` and `jump psexec_psh` will generate an EID 7045 (Service Installation) from `System.evtx`.
- The additional commands will generate an EID 400 event log

(PowerShell Engine Startup) from Windows `PowerShell.evtx`.

The majority of PowerShell Engine Startup events generated by Cobalt Strike will have the `HostApplication` field begin with a command prefix. With the default configuration

Featured

Recent

Video

Category

Start Free Trial

As part of our research, CrowdStrike Services evaluated the following Beacon commands,

which are encountered frequently in incident response engagements:

- powershell and powershell-import
- powerpick



- jump winrm
- remote-exec wmi
- remote-exec powershell

In the following sections we'll review the purpose behind each of these commands, and the artifacts generated that may be useful for security analysts and threat hunters.

The `powershell` and `powershell-import` Commands

Both of these commands have a similar aim: to allow the user to execute PowerShell scripts on the target system. The `powershell` Beacon command executes commands written in PowerShell within the Cobalt Strike framework. When a red teamer or an adversary executes a command within a Beacon session, the operating system will generate an EID 400 event log (PowerShell Engine Startup) on the system that the command is executed on. The `powershell-import` Beacon command imports a PowerShell script into the Beacon session. In several WastedLocker ransomware attacks, CrowdStrike Services<1> observed evidence of the network discovery tool PowerView imported by adversaries shortly after establishing a Beacon on a compromised system. The file system artifacts that are generated will vary depending on whether the `powershell` command is executed before or after the `powershell-import` command.

Featured

Recent

Video

Category

Start Free Trial

command before a script has been imported with `powershell-import`. The base64 encoded command decodes to ls, the command that was executed via the `powershell`



Observations of `powershell` before `powershell-import`:

- The `HostApplication` field is set to `powershell -nop -exec -bypass -EncodedCommand <base64-encoded-command>`
- The Base64 encoded command decodes to the `<command>` executed

Event 400, PowerShell (PowerShell)

General Details

Engine state is changed from None to Available.

Details:

```
NewEngineState=Available  
PreviousEngineState=None  
  
SequenceNumber=13  
  
HostName=ConsoleHost  
HostVersion=5.1.14393.1884  
HostId=9730d99f-2d12-4d0f-8cdb-aad76bf5dbb2  
HostApplication=powershell -nop -exec bypass -EncodedCommand bABzAA==  
EngineVersion=5.1.14393.1884  
RunspaceId=e72a56b3-b855-45ea-8e31-635f07f9bacd  
PipelineId=  
CommandName=  
CommandType=  
ScriptName=  
CommandPath=  
CommandLine=
```

Log Name: Windows PowerShell
Source: PowerShell (PowerShell) Logged: 7/24/2020 11:51:40 AM
Event ID: 400 Task Category: Engine Lifecycle
Level: Information Keywords: Classic
User: N/A Computer: WIN-M46E489VVUE.testinglab.corp
OpCode:
More Information: [Event Log Online Help](#)

Featured

Recent

Video

Category

Start Free Trial

Artifacts generated after `powershell-import`

Figure 2, shows an example of the EID 400 generated on the compromised system after



Net.Webclient).DownloadString('http://127.0.0.1:22426/'); Is . The IEX (New-Object Net.Webclient).DownloadString('http://127.0.0.1:22426/') component of the base64 encoded command is how Cobalt Strike manages imported PowerShell scripts within a Beacon session. The rest of the command, after the DownloadString component, is the PowerShell command run by the adversary.

Observations from `powershell` after `powershell-import`:

- The `HostApplication` field is set to `powershell -nop -exec -bypass -EncodedCommand <base64-encoded-command>`
- The base64 encoded command decodes to IEX (`New-Object Net.Webclient).DownloadString('http://127.0.0.1:<ephemeral-port-number>/'); <command>`)

Featured

Recent

Video

Category

Start Free Trial



Engine state is changed from None to Available.

Details:

```
NewEngineState=Available  
PreviousEngineState=None  
  
SequenceNumber=13  
  
HostName=ConsoleHost  
HostVersion=5.1.14393.1884  
HostId=4b2fa080-2dc6-4ef0-9e95-74b46750c8de  
HostApplication=powershell -nop -exec bypass -EncodedCommand  
SQBFAFgAIAAoAE4AZQB3AC0ATwBiAGOAZQBjAHQAIABOAGUAdAAuAFcAZQBjAGMAbABpAGUAbgB0ACKALgBEAG8AdwBuAGwAbwBhAGQAUwB0AHIAaQBuAGcAKAAnAGgAdAB0  
AHAAOgAvAC8AMQAyAdcALgAwAC4AMAAuADEAOgAyADQAMQA5ADIALwAnACKAOwAgAGwAcwA=
```

EngineVersion=5.1.14393.1884
Runspaceld=3f604b37-58cc-4957-a30f-6eeee5475ec8
PipelineId=
CommandName=
CommandType=
ScriptName=
CommandPath=
CommandLine=

Log Name: Windows PowerShell
Source: PowerShell (PowerShell) Logged: 7/24/2020 11:52:51 AM
Event ID: 400 Task Category: Engine Lifecycle
Level: Information Keywords: Classic
User: N/A Computer: WIN-M46E489VUE.testinglab.corp
OpCode:
More Information: [Event Log Online Help](#)

Figure 2. Artifact generated by the **powershell** command after **powershell-import** is executed
(click image to enlarge)

An example of the observed artifact as shown in Figure 2:

```
HostApplication=powershell -nop -exec Bypass -EncodedCommand  
SQBFAFgAIAAoAE4AZQB3AC0ATwBiAGOAZQBjAHQAIABOAGUAdAAuAFcAZQBjAGMAbABp  
AGUAbgB0ACKALgBEAG8AdwBuAGwAbwBhAGQAUwB0AHIAaQBuAGcAKAAnAGgAdAB0AHAA
```

Featured

Recent

Video

Category

Start Free Trial

The **powerpick** Beacon command executes unmanaged PowerShell on a compromised system. It provides a way to execute a PowerShell command without invoking **powershell.exe**. When a red teamer or adversary executes the **powerpick** command



event log generated by executing the **powerpick** command will contain a mismatch between the version number in the **HostVersion** and **EngineVersion** event log fields. The event generated will also have the path to the **rundll32.exe** executable in the **HostApplication** field, as it is the default program that a Beacon will use to create a new process.

Observations of **powerpick**:

- **HostName** field is set to **ConsoleHost**
- **HostApplication** field is set to the file path of **rundll32.exe**
- The **HostVersion** and **EngineVersion** fields are set to different values

Event 400, PowerShell (PowerShell)

General Details

Engine state is changed from None to Available.

Details:

NewEngineState=Available
PreviousEngineState=None

SequenceNumber=17

HostName=ConsoleHost
HostVersion=1.0
HostId=bfb9cc49-31b3-4805-af69-ff07f8e9b1fc
HostApplication=C:\Windows\sysnative\rundll32.exe
EngineVersion=5.1.17763.1
Source: Microsoft-Windows-SysNative-EventLog
Time: 2023-10-31 16:01:29Z

Featured

Recent

Video

Category

Start Free Trial

OpCode:

More Information: [Event Log Online Help](#)

Figure 3. Artifact generated by the **powerpick** Beacon command when executed (click image to enlarge)



EngineVersion=5.1.17763.1

The `jump psexec`

Command

The `jump psexec` Beacon command establishes an additional Beacon on a remote system. When an adversary executes the `jump psexec` command through a Beacon session, the filesystem will generate an EID 7045 event log (Service Installation) on the remote system.

Observations of `jump psexec`:

- The Service Name field is set to `<7-alphanumeric-characters>`
- The Service File Name field is set to `\\"127.0.0.1\ADMIN$\<7-alphanumeric-characters>.exe`

[Featured](#)

[Recent](#)

[Video](#)

[Category](#)

[Start Free Trial](#)



A screenshot of a Windows Event Log window. At the top, it says "A service was installed in the system." Below that, there is detailed service information:

Service Name:	af5ce43
Service File Name:	\\127.0.0.1\ADMIN\$\af5ce43.exe
Service Type:	user mode service
Service Start Type:	demand start
Service Account:	LocalSystem

At the bottom of the event details, there is a table of event properties:

Log Name:	System
Source:	Service Control Manager
Event ID:	7045
Level:	Information
User:	TESTINGLAB\Administrator
OpCode:	Info
Logged:	7/6/2020 8:39:10 AM
Task Category:	None
Keywords:	Classic
Computer:	WIN-M46E489VVUE.testinglab.corp

Below the event properties, there is a link: [More Information: Event Log Online Help](#).

Figure 4. Artifact generated by the `jump psexec` Beacon command when executed on the remote system prior to version 4.1 of Cobalt Strike (click image to enlarge)

An example of the observed artifact as shown in Figure 4:

Service Name: af5ce43 **Service File Name:**

`\\\127.0.0.1\ADMIN$\af5ce43.exe`

By default, events generated by the `jump psexec` Beacon command using versions of Cobalt Strike prior to version 4.1 will have the `127.0.0.1` localhost string in the value of the "Service File Name," an example of this is `\\\127.0.0.1\ADMIN$\7f5747a.exe`.

Featured

Recent

Video

Category

Start Free Trial

- The Service Name field is set to `<7-alphanumeric-characters>`
- The Service File Name field is set to `\\\<System-IPAddress>\ADMIN$\<7-alphanumeric-characters>.exe`



Service Name: 850c1a1
Service File Name: \\10.0.0.16\ADMIN\$\850c1a1.exe
Service Type: user mode service
Service Start Type: demand start
Service Account: LocalSystem

Log Name: System
Source: Service Control Manager Logged: 8/4/2020 10:12:51 PM
Event ID: 7045 Task Category: None
Level: Information Keywords: Classic
User: TESTINGLAB\Administrator Computer: WIN-M46E489VVUE.testinglab.corp
OpCode: Info
More Information: [Event Log Online Help](#)

Figure 5. Artifact generated by the `jump psexec` Beacon command when executed on the remote system created by version 4.1+ of Cobalt Strike (click image to enlarge)

The `jump psexec_psh` Command

The `jump psexec_psh` command

establishes an additional Beacon on a remote system via the Windows Service Control

Featured

Recent

Video

Category

Start Free Trial

`-nop -w hidden -encodedCommand`.

Observations of `jump psexec_psh`:



```
powershell -nop -w hidden -encodedcommand <base64-encoded-command>
```

- The base64 encoded command decodes to a PowerShell stager for a Cobalt Strike Beacon

Event 7045, Service Control Manager

General Details

A service was installed in the system.

Service Name: 9df3724

Service File Name: %COMSPEC% /b /c start /b /min powershell -nop -w hidden -encodedcommand

[Redacted long base64 string]

Log Name: System

Source: Service Control Manager

Event ID: 7045

Level: Information

User: TESTINGLAB\Administrator

OpCode: Info

More Information: [Event Log Online Help](#)

Featured

Recent

Video

Category

Start Free Trial

The jump winrm Command

The **jump winrm** Beacon command establishes a Beacon on a remote system utilizing the Windows Remote Management (WinRM) interface (native on all Windows devices).



command prefix in the **HostApplication** field. The generated event is not affected by the usage of any of the PowerShell-related Beacon commands.

Observations of **jump winrm** on the compromised system:

- The **HostApplication** field is set to `powershell -nop -exec -bypass -EncodedCommand <base64-encoded-command>`
- The base64 encoded command decodes to `IEX (New-Object Net.WebClient).DownloadString('http://127.0.0.1:<ephemeral-port-number>')`

The screenshot shows a Windows Event Viewer window titled "Event 400, PowerShell (PowerShell)". The "Details" tab is selected. The event details indicate that the engine state has changed from None to Available. The "HostApplication" field contains the command: `powershell -nop -exec bypass -EncodedCommand`. The "EncodedCommand" value is a long base64 string: `SQBFAFgIAAAoAE4AZQB3AC0ATwbIAgoAZQBjAHQAIABOAGUAdAuuFcAZQBiAGMAbABpAGUAbgB0ACKALgBEAG8AdwBuAGwAbwBhAGQAUwB0AHIAaQBuAGcAKAAAnAGgAdAB0AHAAOgAvAC8AMQAyAdcALgAwAC4AMAAuADEAOgAyADgAMwA0ADUALwAnACKA`. Other fields include HostName=ConsoleHost, HostVersion=5.1.14393.1884, HostId=3bc0e472-7ee0-4f21-a03f-6224a3da623f, SequenceNumber=13, and EngineVersion=5.1.14393.1884.

Featured

Recent

Video

Category

Start Free Trial

Figure 7. Artifact generated by the **jump winrm** Beacon command when executed, on the compromised system (click image to enlarge)

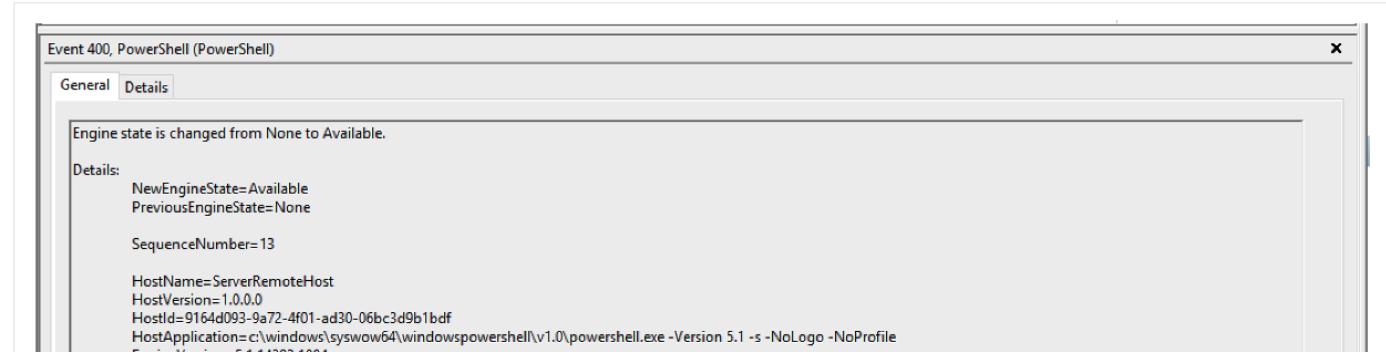


SQBFAFgAIAAoAE4AZQB3AC0ATwBiAGoAZQBjAHQAIABOAGUAdAAuAFcAZQBiAGMAbABp
AGUAbgB0ACKALgBEAG8AdwBuAGwAbwBhAGQAUwB0AHIAaQBuAGcAKAAnAGgAdAB0AHAA
OgAvAC8AMQAYADcALgAwAC4AMAAuADEAOgAyADgAMwA0ADUALwAnACkADecoded
Base64 Command: IEX (New-Object
Net.Webclient).DownloadString('http://127.0.0.1:28345/')

If a WinRM listener is not present on the remote system when the `jump winrm` command is executed, Cobalt Strike will create an EID 400 event log on the remote system, as shown in Figure 7.

Observations of an event created by `jump winrm` on the remote system:

- The `HostApplication` field is set to `<path-to-PS-executable> -Version <PS-Version> -s -NoLogo -NoProfile`



Featured

Recent

Video

Category

Start Free Trial

Figure 8. Artifact generated by the `jump winrm` Beacon command when executed on the remote system
(click image to enlarge)



```
HostApplication=c:\windows\syswow64\windowspowershell\v1.0\powershell.exe -Version 5.1 -s -NoLogo -NoProfile
```

The `remote-exec wmi` Command

The `remote-exec wmi` Beacon command executes a command on a remote system via WMI. When the `remote-exec wmi` command is executed, the filesystem will generate an EID 400 event log (PowerShell Engine Startup) on the compromised system with the standard Cobalt Strike PowerShell command prefix in the `HostApplication` field.

Observations of `remote-exec wmi`:

- The `HostApplication` field is set to `powershell -nop -exec Bypass -EncodedCommand <base64-encoded-command>`
- The base64 encoded command decodes to `Invoke-WMIMethod win32_process -name create -argumentlist '<command>' -ComputerName <target>`

[Featured](#)

[Recent](#)

[Video](#)

[Category](#)

[Start Free Trial](#)



Engine state is changed from None to Available.	
Details:	
NewEngineState=Available	
PreviousEngineState=None	
SequenceNumber=13	
HostName=ConsoleHost	
HostVersion=5.1.14393.1884	
HostId=4dbf8a7d-cbf5-432b-be7a-37cf5d7f3574	
HostApplication=powershell -nop -exec bypass -EncodedCommand	
SQBuAHYAbwBrAGUALQBAE0ASQBNAGUAdABoAG8AZAAgAHCAaQBuADMAMgBfAHAACgBvAGMAZQBzAHMAIAAtAG4AYQBtAGUAIAbjAHIAZQBhAHQAZQAgAC0AYQBByAGcAdQ	
BtAGUAbgB0AgwAaBzAHOAIAnAHcAaAbvAGEAbQbPcCAlAAAtEMAAbwTbAHAAdQb0AGUAcgBOAGEAbQbIACAAVwBJAE4AMQAwAA==	
EngineVersion=5.1.14393.1884	
RunspaceId=5ce753d6-9a79-4c92-aba1-8daff982a649	
PipelineId=	
CommandName=	
CommandType=	
ScriptName=	
CommandPath=	
CommandLine=	
Log Name: Windows PowerShell	
Source: PowerShell (PowerShell)	Logged: 7/9/2020 9:28:03 AM
Event ID: 400	Task Category: Engine Lifecycle
Level: Information	Keywords: Classic
User: N/A	Computer: WIN-M46E489VVUE.testinglab.corp
OpCode:	
More Information: Event Log Online Help	

Figure 9. Artifact generated by the **remote-exec wmi** Beacon command when executed on the compromised system (click image to enlarge)

An example of the observed artifact as shown in Figure 9:

```
HostApplication=powershell -nop -exec Bypass -EncodedCommand  
SQBuAHYAbwBrAGUALQBXAE0ASQBNAGUAdABoAG8AZAAgAHcAaQBuADMAMgBfAHAACgBv  
AGMAZQBzAHMAIAAtAG4AYQBtAGUAIABjAHIAZQBhAHQAZQAgAC0AYQByAGcAdQBtAGUA  
h-0QAC-AcQB-AuGATAAa-AuIcAc-ABr-AcGEAl-QB-Ac-GATAAa-AfEMAh-BfAUAa-JQBcAChA-acQB
```

Featured

Recent

Video

Category

Start Free Trial

The `remote-exec powershell` Beacon command executes a command on a remote system via PowerShell remoting from a compromised system. When the `remote-exec powershell` command is executed, the filesystem will generate an EID 400 event log

Observations of `remote-exec powershell`:

- The `HostApplication` field is set to `powershell -nop -exec Bypass -EncodedCommand <base64-encoded-command>`
- The Base64 encoded command decodes to `Invoke-Command -ComputerName <target> -ScriptBlock { <command> }`

Event 400, PowerShell (PowerShell)

General Details

Engine state is changed from None to Available.

Details:

```
NewEngineState=Available  
PreviousEngineState=None  
  
SequenceNumber=13  
  
HostName=ConsoleHost  
HostVersion=5.1.14393.1884  
HostId=be722e9f-15a5-4dd0-93fd-541bf8a84151  
HostApplication=powershell -nop -exec bypass -EncodedCommand  
SQBuAHYAbwBrAGUALQBDAG8AbQbtAGEAbgBkACAALQBDAG8AbQBwAHUAdABIAHIAgBhAG0AZQAgADEAMAAuADAALgAwAC4AMQAwACAALQBTAGMAcgBpAHAAdABCAGwAbwBjAGsAIA87ACAAdwB0oAG8AYQbtAgkAIAB9AA==  
EngineVersion=5.1.14393.1884  
RunspaceId=224994f8-cd08-412b-9b3b-8c8795ff9808  
PipelineId=  
CommandName=  
CommandType=  
ScriptName=  
CommandPath=  
CommandLine=  
  
Log Name: Windows PowerShell  
Source: PowerShell (PowerShell) Logged: 7/8/2020 5:03:48 PM  
Event ID: 400 Task Category: Engine Lifecycle
```

Featured

Recent

Video

Category

Start Free Trial

`HOSTApplication=powershell -nop -exec Bypass -EncodedCommand
SQBuAHYAbwBrAGUALQBDAG8AbQbtAGEAbgBkACAALQBDAG8AbQBwAHUAdABIAHIAgBhAG0AZQAgADEAMAAuADAALgAwAC4AMQAwACAALQBTAGMAcgBpAHAAdABCAGwAbwBjAGsA`

Conclusions

Although Cobalt Strike provides the operator a degree of freedom to configure some of the previously mentioned commands, those features are not always leveraged by adversaries. Due to the high prevalence of Cobalt Strike in contemporary intrusions, CrowdStrike recommends collecting EID 400 (PowerShell Engine Startup) and EID 7045 event logs (Service Installation) for monitoring and alerting in a centralized [security information and event management \(SIEM\) platform](#).

CrowdStrike also recommends upgrading to the most recent version of PowerShell and disabling previous versions, as PowerShell is backward compatible. While these additional security measures do not provide full visibility into Cobalt Strike activity, they can aid in its detection. <1> CrowdStrike has previously reported on adversaries that use Cobalt Strike, such as [COBALT SPIDER](#).

Additional Resources

- Learn more about the [CrowdStrike Services team](#) and how it can help your organization improve your cybersecurity readiness.

Featured

Recent

Video

Category

[Start Free Trial](#)



DIRECTED BY COBALT STRIKE

PROTECT AGAINST MALWARE, RANSOMWARE AND FILELESS ATTACKS

Related Content



CrowdStrike Named a Leader with “Bold Vision” in 2024 Forrester Wave for Cybersecurity Incident Response Services



How to Defend Employees and Data as Social Engineering Evolves



The Anatomy of an ALPHA SPIDER Ransomware Attack

CATEGORIES

[Featured](#)

[Recent](#)

[Video](#)

[Category](#)

[Start Free Trial](#)

 Exposure Management

84

 From The Front Lines

190



 Public Sector	37
 Small Business	8

CONNECT WITH US



Featured

Recent

Video

Category

Start Free Trial



CROWDSTRIKE

Get started with CrowdStrike for free.

[Featured](#)

[Recent](#)

[Video](#)

[Category](#)

[Start Free Trial](#)





Start Free Trial

FEATURED ARTICLES

October 01, 2024

CrowdStrike Named a Leader in 2024 Gartner® Magic Quadrant™ for Endpoint Protection Platforms

September 25, 2024

Featured

Recent

Video

Category

Start Free Trial

SUBSCRIBE



See CrowdStrike Falcon® in Action

Detect, prevent, and respond to attacks— even malware-free intrusions—at any stage, with next-generation endpoint protection.

[See Demo](#)

« [New Report: Falcon OverWatch Threat Hunting Leaves Adversaries with Nowhere to Hide](#)

[Duck Hunting with Falcon Complete: Analyzing a Fowl Banking Trojan, Part 1»](#)

Featured

Recent

Video

Category

Start Free Trial



Copyright © 2024 CrowdStrike | Privacy | Request Info | Blog | Contact Us | 1.888.512.8906 | Accessibility