



Careers



855-525-2489



Contact Us

REQUEST A DEMO



WHY CARBON BLACK ▾

PRODUCTS ▾

SOLUTIONS ▾

PARTNERS ▾

RESOURCES ▾

COMPANY ▾

BLOG



Investigating Bitcoin Malware Infections Using Carbon Black

24
JUL

Investigating Bitcoin Malware Infections Using Carbon Black

July 24, 2014 / Ryan Nolette / Advanced Threat Protection, Detection and Response, Tech Toolbox

In my [previous post](#), I demonstrated a few simple ways a user can take advantage of the built-in utilities in Windows to perform a high-level-malware investigation. In this post, we will examine a specific sample of this malware using Carbon Black.

Sample Used:

<http://ow.ly/zy0Gc>

SHA-256:

94FE198E4614BEC6233585D518ADDE34A01DC0A35C7115C79532564B9E0E4080

MD5:

8BDF872A5D2253F0D1DFFD4E5C4FB2A1

For this analysis I executed the sample above on a Windows 7 host. The Windows system was fully up-to-date on patches (as of 05/30/2014) and I intentionally ran only Carbon Black on this host.

Where do we start?

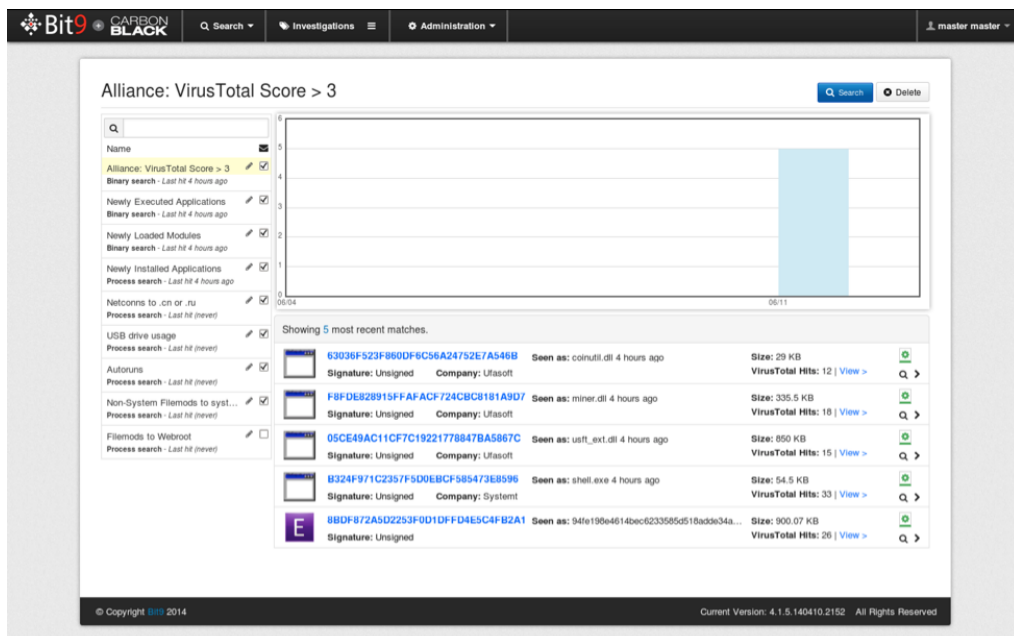
In this scenario we have the advantage of knowing when the malware was detonated and on what system. In a real-world scenario we might not have that information and would need to find malware. To make this more realistic, I will not leverage my prior knowledge of this malware to investigate it with Carbon Black.

So, where do we start and how do we find this malware on this system?

Watchlists

The first thing I did after installing the Carbon Black server was enable watchlists under the alliance feeds administrative option. I won't delve into the setup of this application too deeply because this is an investigation, not a product demo. With watchlists enabled and the sensor installed on the win7 client, I detonate the Bitcoin malware.

Next, I switch to the Carbon Black console tab "watchlists" and look for any hits.



Right away I can see five matches for files in my environment that have a VirusTotal rating of four or more. These files are:

- 63036F523F860DF6C56A24752E7A546B

Seen as: coinutil.dll 19 minutes ago

Signature: Unsigned Company: Ufasoft

Size: 29 KB

VirusTotal Hits: 12

F8FDE828915FFAFACF724CBC8181A9D7

Seen as: miner.dll 19 minutes ago

Signature: Unsigned Company: Ufasoft

Size: 335.5 KB

VirusTotal Hits: 18

05CE49AC11CF7C19221778847BA5867C

Seen as: usft_ext.dll 19 minutes ago

Signature: Unsigned Company: Ufasoft

Size: 850 KB

VirusTotal Hits: 15

B324F971C2357F5D0EBCF585473E8596

Seen as: shell.exe 19 minutes ago

Signature: Unsigned Company: Systemt

Size: 54.5 KB

VirusTotal Hits: 33

<http://ow.ly/zy1f6>

8BDF872A5D2253F0D1DFFD4E5C4FB2A1

Seen as:

94fe198e4614bec6233585d518adde34a01dc0a35c7115c79532564b9e0e4080.bin 19 minutes ago

Signature: Unsigned

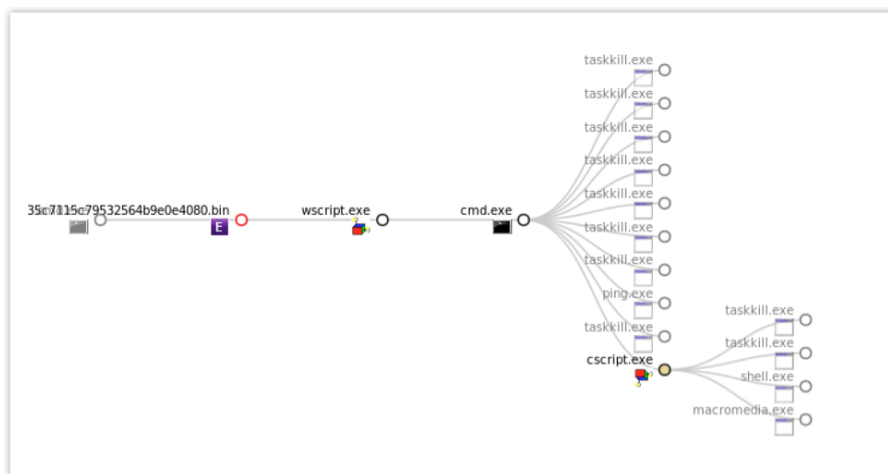
Size: 900.07 KB

VirusTotal Hits: 26

<http://ow.ly/zy1ra>

Drilling into process execution chains

Immediately, my eyes are drawn to the last two files in this view because they have more than 25 hits from VT. Let's click on this bottom one and see what we learn about it.

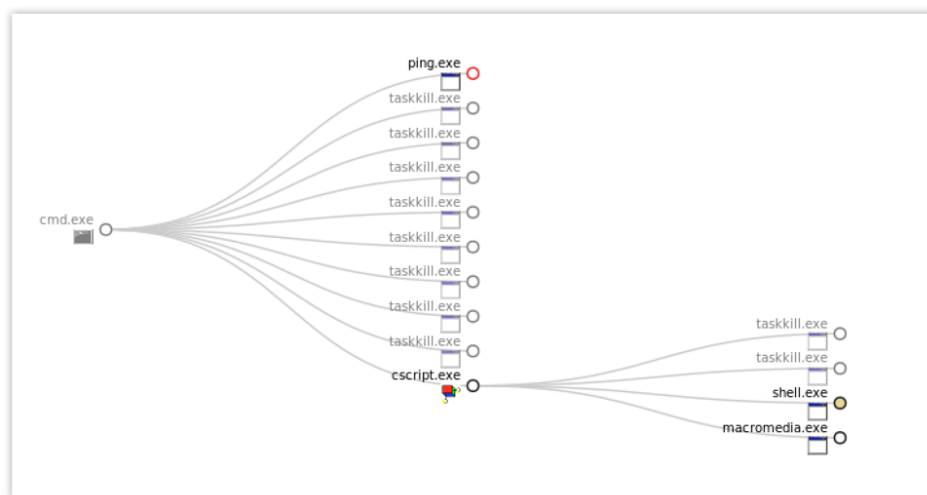


From the Carbon Black process analysis of the file

"94fe198e4614bec6233585d518adde34a01dc0a35c7115c79532564b9e0e4080.bin" we are able to see the spawn of :

- Wscript.exe
- Cmd.exe
- Taskkill.exe
- Cscript.exe
- Ping.exe

If we then drill into each of these child processes we can see that "cscript.exe" spawned three processes.




Cscript.exe shows us that it spawned:

- Taskkill.exe
- Shell.exe
- Macromedia.exe

This visualization of the execution chain gives us a very quick and easy view of this binary we are investigating spawning another of the processes that came up in the VT watchlist search, "shell.exe". This leads me to believe from the evidence shown so far that "94fe198e4614bec6233585d518adde34a01dc0a35c7115c79532564b9e0e4080.bin" and "shell.exe" are at least related, if not also malicious, and that the bin file is the parent process that created "shell.exe."

Drilling into process analysis

Since this bin file seems to have started the trouble, let's look a bit closer at it. Since the page that drills down into this process has a bunch of information on it, I have broken it up below for easy digestion in this blog post.

8BDF872A5D2253F0D1DFFD4E5C4FB2A1
Seen as: 94fe198e4614bec6233585d518adde34a01dc0a35c7115c79532564b9e0e4080.bin
First seen at: 2014-06-11T16:52:52.961Z (22 minutes ago)
Status: **Unsigned**
Publisher Name:
 Watchlist(s): 2 | [View >](#)
Q File writer(s): 2 | [Find writers >](#)
Q Related process(es): 5 | [Find related >](#)
Search the web: [Google >](#)

Frequency Data
1 computers have seen this md5 in 3 processes.

Partner Information
VirusTotal Hits: 26 [View on VirusTotal >](#)

General Info

Architecture	32 bit
Binary Type	Shared Resource
Size	900.07 KB Download

Digital Signature Metadata

Result	Unsigned
Result Code	0x800b0100

File Version Metadata

File Description	(unknown)
File Version	(unknown)
Original Filename	(unknown)
Internal Name	(unknown)
Company Name	(unknown)
Product Name	(unknown)
Product Version	(unknown)

Observed Paths

Observed Path	c:\users\win7\desktop\94fe198e4614bec6233585d518adde34a01dc0a35c7115c79532564b9e0e4080.bin
---------------	--

Observed Hosts (1) @

Hostname	win7-pc
----------	---------

Below, we can see a bunch of useful expected file information about the process we are analyzing. We have its hash; we have the first seen event, status, and a few other odds and ends about it. I mostly used this section to compare against timestamps of the other associated files to confirm this file came first and is most likely the catalyst binary.

8BDF872A5D2253F0D1DFFD4E5C4FB2A1
Seen as: 94fe198e4614bec6233585d518adde34a01dc0a35c7115c79532564b9e0e4080.bin
First seen at: 2014-06-11T16:52:52.961Z (22 minutes ago)
Status: **Unsigned**
Publisher Name:
 Watchlist(s): 2 | [View >](#)
Q File writer(s): 2 | [Find writers >](#)
Q Related process(es): 5 | [Find related >](#)
Search the web: [Google >](#)

Next, we can see something very important for any incident—scope. We can see that only one computer in our environment has this file on it and that VT has 26 hits for it.

Frequency Data

1 computers have seen this md5 in 3 processes.

Partner Information

VirusTotal Hits: **26**  [View on VirusTotal »](#)

Next we see that the file is a 32-bit binary with shared resources, the size of the file, and that observed path of the file. This path in this case shows the username of the user that executed the binary.

General Info

Architecture	32 bit
Binary Type	Shared Resource
Size	900.07 KB Download

Observed Paths

Observed Path	c:\users\win7\desktop\94fe198e4614bec6233585d518adde34a01dc0a35c7115c79532564b9e0e4080.bin
----------------------	--

Observed Hosts (1)

Hostname	win7-pc
-----------------	---------

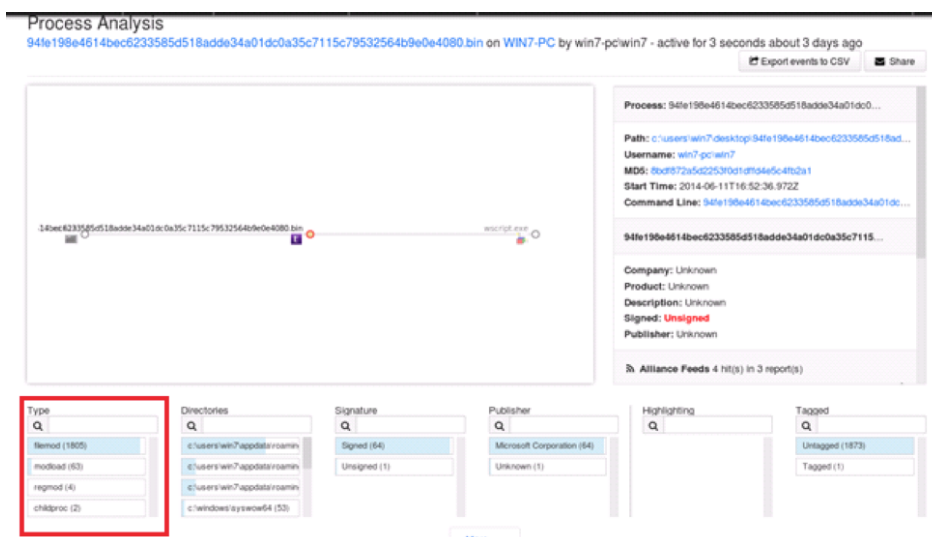
Last, but not least, we can see that this file is only observed on one host. This tells us that this incident is not part of a large-scale attack and is most likely either a targeted attack or just a one-off infection.

Now that we've seen what the file is, what did it do? This bin file's behavior can be narrowed down to eight file creations, five registry modifications, and a handful of child processes spawned that will require further investigation.

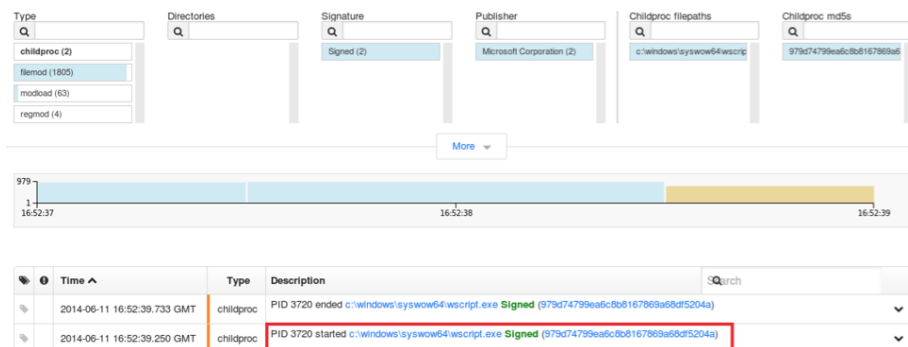
Time	Type	Description
Wed Jun 11 2014 12:52:39 GMT-0400 (EDT)	regmod	First wrote to 'registry\user-s-1-5-21-1581004485-488565272-3498079395-1000\software\microsoft\windows\currentversion\internet settings\zonemap\autodetect'
Wed Jun 11 2014 12:52:39 GMT-0400 (EDT)	regmod	First wrote to 'registry\user-s-1-5-21-1581004485-488565272-3498079395-1000\software\microsoft\windows\currentversion\internet settings\zonemap\uncasintranet'
Wed Jun 11 2014 12:52:37 GMT-0400 (EDT)	regmod	First wrote to 'registry\user-s-1-5-21-1581004485-488565272-3498079395-1000\software\winrar\stx\c%\users%win7%appdata%\roaming%\windowspid'
Wed Jun 11 2014 12:52:37 GMT-0400 (EDT)	regmod	Created 'registry\user-s-1-5-21-1581004485-488565272-3498079395-1000\software\winrar\stx'
Wed Jun 11 2014 12:52:37 GMT-0400 (EDT)	filemod	Created c:\users\win7\appdata\roaming\windowspid\shell\shell.exe_part100

Time ^	Type	Description	Search
2014-06-11 16:52:37.581 GMT	filemod	Created c:\users\win7\appdata\roaming\windowspid\usft_ext.exe.vbs	
2014-06-11 16:52:37.565 GMT	filemod	Created c:\users\win7\appdata\roaming\windowspid\usft_ext.dll	
2014-06-11 16:52:37.565 GMT	filemod	Created c:\users\win7\appdata\roaming\windowspid\put.vbs	
2014-06-11 16:52:37.565 GMT	filemod	Created c:\users\win7\appdata\roaming\windowspid\phatk.ptx	
2014-06-11 16:52:37.565 GMT	filemod	Created c:\users\win7\appdata\roaming\windowspid\phatk.cl	
2014-06-11 16:52:37.549 GMT	filemod	Created c:\users\win7\appdata\roaming\windowspid\kill.bat	
2014-06-11 16:52:37.549 GMT	filemod	Created c:\users\win7\appdata\roaming\windowspid\colnutil.dll	
2014-06-11 16:52:37.159 GMT	filemod	Created c:\users\win7\appdata\roaming\windowspid_tmp_rar_sfx_access_check_1827738	

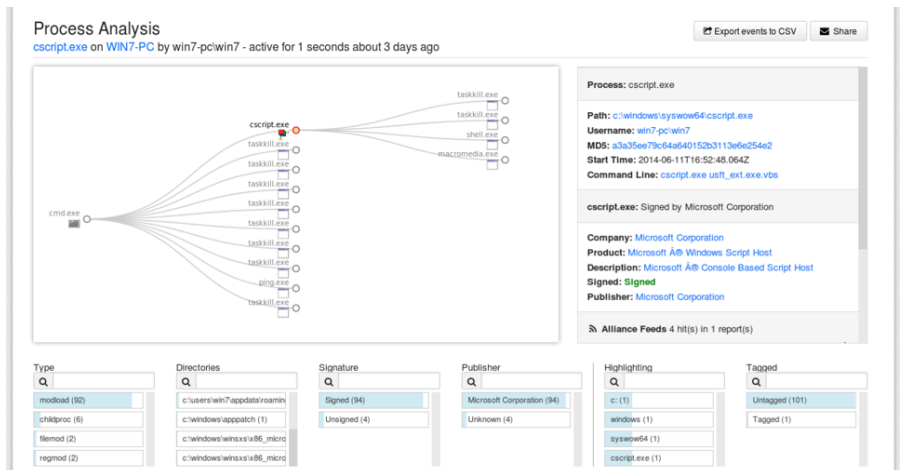
According to the analysis page the above snippet is just the tip of the iceberg. There were 1805 file modifications, 63 module loads, four registry modifications, and two child processes spawned just from this file alone. These counts do not include everything done by the child processes spawned.



To find out what the child processes did, you simply have to choose that option under type and drill into the results.



If you continue to drill down into each child process, you will eventually get to the first diagram I displayed with the csript.exe file running.



This is the process analysis of the script and you can see it showing the spawning of six child processes, two of which are on the VT watch list. Two file modifications are the file creation events for the fake skype.ink file.

Time	Type	Description	Search
2014-06-11 16:52:48.345 GMT	childproc	PID 1348 started c:\windows\system64\taskkill.exe Signed (94bdcfb584c979b385adee14b08ab4)	
2014-06-11 16:52:48.501 GMT	childproc	PID 1348 ended c:\windows\system64\taskkill.exe Signed (94bdcfb584c979b385adee14b08ab4)	
2014-06-11 16:52:48.516 GMT	childproc	PID 920 started c:\windows\system64\taskkill.exe Signed (94bdcfb584c979b385adee14b08ab4)	
2014-06-11 16:52:48.610 GMT	childproc	PID 920 ended c:\windows\system64\taskkill.exe Signed (94bdcfb584c979b385adee14b08ab4)	
2014-06-11 16:52:48.625 GMT	childproc	PID 1124 started c:\users\win7\appdata\roaming\windows\pid\shell.exe Unsigned (b324f971c2357f5dbecf585473e8596)	
2014-06-11 16:52:48.703 GMT	childproc	PID 2624 started c:\users\win7\appdata\roaming\windows\pid\macromedia.exe Unsigned (b324f971c2357f5dbecf585473e8596)	
2014-06-11 16:52:48.282 GMT	filemod	Created c:\users\win7\appdata\roaming\microsoft\windows\start menu\programs\startup\skype.ink	
2014-06-11 16:52:48.282 GMT	filemod	First wrote to c:\users\win7\appdata\roaming\microsoft\windows\start menu\programs\startup\skype.ink	

The more we drill into each of these unique events, the more we can learn about the malware. This is a great feature of Carbon Black and extremely useful when trying to track down the flow of each execution and child processes.

Did it phone home? Was data exfiltrated?

Now that we have the binary that started all the fuss, let's do some research on it. I click on the VirusTotal link from inside the Carbon Black console and see the output below.

Antivirus	Result	Update
AVG	BAT/Agent.BP	20131020
Agnitum	Trojan.XPACK!KCiDxv8Zl8c	20131019
AhnLab-V3	Trojan/Win32.BitMiner	20131020
AntiVir	VBS/CoinMiner.I	20131020
Avast	VBS:Malware-gen	20131020
Baidu-International	Trojan.VBS.CoinMiner.AK	20131020
BitDefender	Application.BitCoinMiner.AT	20131012
CAT-QuickHeal	BAT/Lashtorm.B	20131019
CommTouch	W32/Trojan.LEBU-8331	20131020
Comodo	UnclassifiedMalware	20131020
DrWeb	Trojan.KillProc.28742	20131020
ESET-NOD32	VBS/CoinMiner.N	20131020
Fortinet	W32/BitCoinMiner.Z	20131020
GData	Application.BitCoinMiner.AT	20131020
Ikarus	VBS.Malware	20131020
Kaspersky	Trojan.VBS.Bitmin.d	20131020
Kingsoft	Win32.Troj.Bitmin.d.(kcloud)	20130829
Malwarebytes	PUP.BitcoinMiner	20131020
McAfee	RDN/Generic.dxlq3	20131020
McAfee-GW-Edition	RDN/Generic.dxlq3	20131020
MicroWorld-eScan	Application.BitCoinMiner.AT	20131020
Panda	Trj/CI.A	20131020
Sophos	Bitcoin Miner	20131020
Symantec	Trojan.ADH	20131020
TrendMicro-HouseCall	TROJ_GEN.F47V0522	20131020
VIPRE	Trojan.Win32.Generic!BT	20131020

Above you can see that 26 virus scanners confirm that this binary is malware and that most of them name it as something containing the word "Bitcoin," "miner," or "coin" in variations. This now leads us to what is a Bitcoin miner and a quick Google search will return most of what we posted in [my first blog in this series](#).

Almost every description of a Bitcoin mining malware contains some discussion on how the malware talks to home. So did it talk to home? Did it just send information about the coins it mined or did it send back all of your browser's saved passwords?

How do we know?

Easy. We use Carbon Black to look for network connections from the host in question, "Win7," and any of the associated processes we found in the previous steps.

Contains text... Search

+ Add Criteria ▾

Count of network connections ⊕ ▾

☒ Greater than or Equal

☐ Less than or Equal

☐ Between and

☐ Equals

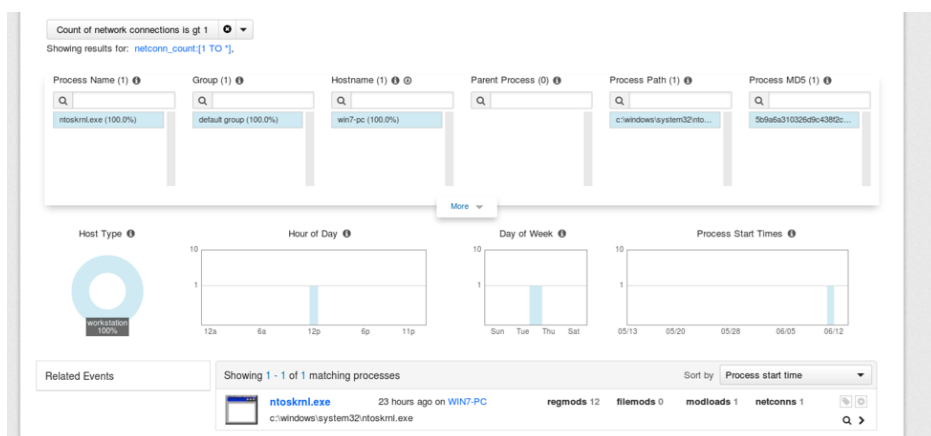
Cancel Update

Hostname (1) ⓘ ⓘ

win7-pc (100.0%)

M

This narrow search returned zero results so I broadened the search to any network connections from the host "Win7."



This wider search returned only one connection attempt. It is from ntoskrnl.exe and as you can see below, was from 172.16.233.1, which is the gateway address for my virtual network. This tells us that the malware did not phone home and no data was exfiltrated.

Time ^	Type	Description
2014-06-11 16:39:28.891 GMT	netconn	Connection from 172.16.233.1 on tcp/57139

Repetition

Now that we have completed the analysis into one binary, we can repeat this process for the other associated files found. Repeating this process for new files will help fill in any analysis gaps and solidify our incident scope as it both expands and contracts with each repetition.

Wrap up

Using Carbon Black really sped up that process, and that was just for one host. If I had 100 or more hosts, it is not only impractical to do the searching manually but next to impossible with time constraints. I would have to script the majority of that searching and basically create a poor imitation of Carbon Black. This would not give us the same level of visibility of Carbon Black nor would we have the drill-down effect of Carbon Black to lead us to associated processes easily.

The last thing I want to point out is that while we provide this enterprise intelligence to our users to make responding to incidents much faster, there's a whole new realm of detection that is possible. By creating watchlists, we can detect and be alerted when these or similar behaviors occur. These watchlists can be customized for each environment.

For a few examples of watch lists, check out Ben Johnson's blog "[Screenshot Demo: Hunt 'Evil' Faster than Ever with Carbon Black.](#)"

The final installment of the series will be a high-level walkthrough of stopping this malware leveraging the Bitg Security Platform. Check back in to the Bitg blog soon so you don't miss it!



Tags:

[alliance feeds](#)[Carbon Black](#)[malware](#)[VirusTotal](#)[watchlists](#)

More Posts

