How I Quit Freaking Out Over Strange Network Traffic

06 JUL

How I Quit Freaking Out Over Strange Network Traffic

July 6, 2015 / Ryan Nolette / Advanced Threat Protection, Detection and Response, Endpoint and Server Security, Mobile Security, Prevention, Response, Tech Toolbox

This past weekend I received one of the most dreaded messages a SecOps member can get: "I think someone is trying to get into my account."

I immediately ran over to my laptop and logged in. Once I logged into my system I was able to get some more context than a 50-character text. I think I looked something like this:

What I gathered from the frantically written email:

- A user received 3 requests from a 2-factor authentication application for access to their cloud account.
- The user denied these requested codes.
- This user was not working at the time when they received these messages.
- They had not logged into their work laptop since the previous workday.

At this point, a few facts could be established:

1. The user did not manually request these codes for access.
2. The user reported that they denied access to the requests.
3. The source of the requests is unknown.
4. The number of user accounts getting these requests is unknown.
5. My coffee is getting cold.

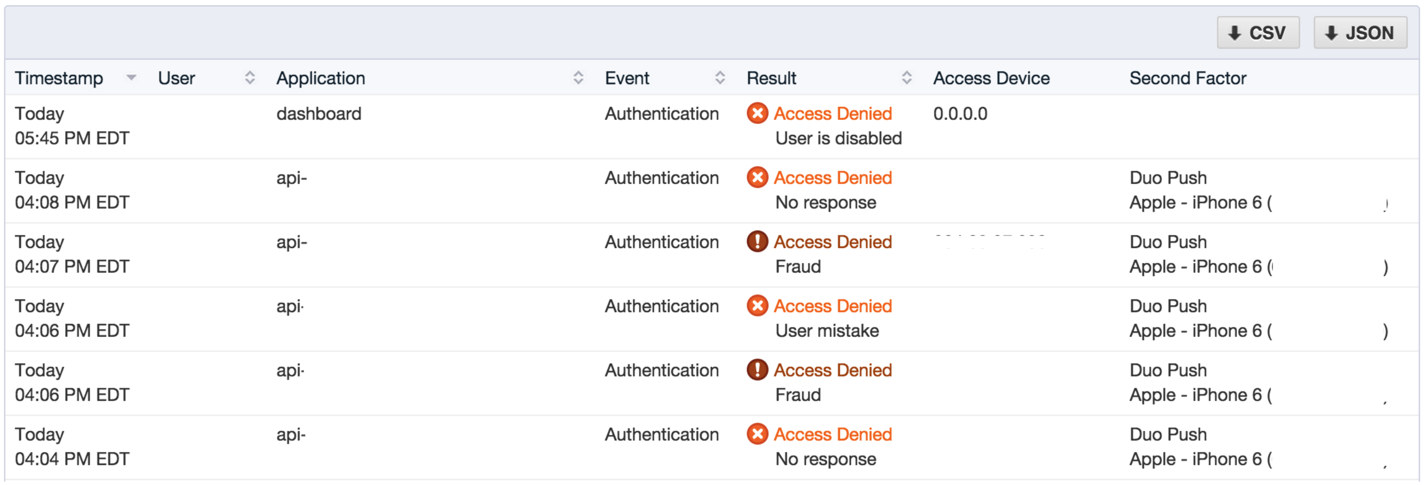
But we still had many questions to answer:

1. Who/what requested access?
2. Were they successful at gaining access to the cloud account?
3. Were they able to breach the corporate network?
4. Was that really coffee that I just drank?

After notifying the appropriate internal contacts and getting a fresh cup of coffee, I started to look into what the logs could tell me.

The first things I look at are the two-factor authentication logs. I do this for a few reasons.

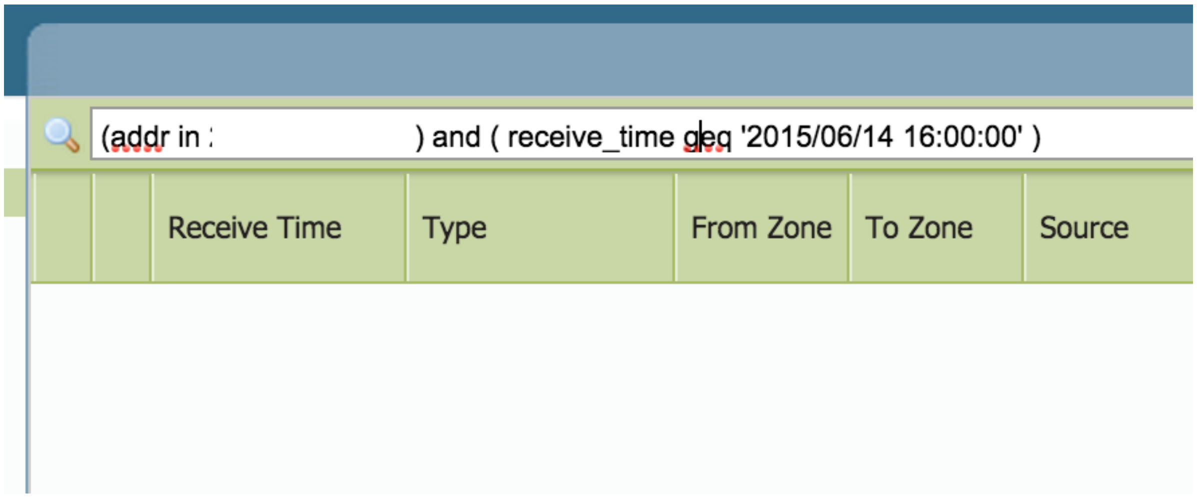
First, I can see the scope of this event pretty easily. If I only see one account getting these prompts, I can shrink my immediate search to just that user for now. Second, I can see the source IP of the access-requesting device. Dependent on the IP, I can rule out legitimate events., this user was the same for all the events, as was the access device IP address, the application that was trying to be accessed and the second factor destination. Also, the timestamps on the attempts have outlined a rough window for me to search for other anomalous behavior in. Luckily for me, this window is fairly small.



Timestamp	User	Application	Event	Result	Access Device	Second Factor
Today 05:45 PM EDT		dashboard	Authentication	Access Denied User is disabled	0.0.0.0	
Today 04:08 PM EDT		api-	Authentication	Access Denied No response		Duo Push Apple - iPhone 6 (
Today 04:07 PM EDT		api-	Authentication	Access Denied Fraud		Duo Push Apple - iPhone 6 ( )
Today 04:06 PM EDT		api-	Authentication	Access Denied User mistake		Duo Push Apple - iPhone 6 ( )
Today 04:06 PM EDT		api-	Authentication	Access Denied Fraud		Duo Push Apple - iPhone 6 (
Today 04:04 PM EDT		api-	Authentication	Access Denied No response		Duo Push Apple - iPhone 6 (

This is awesome because this means I can shrink my search down to just this destination and just this user.

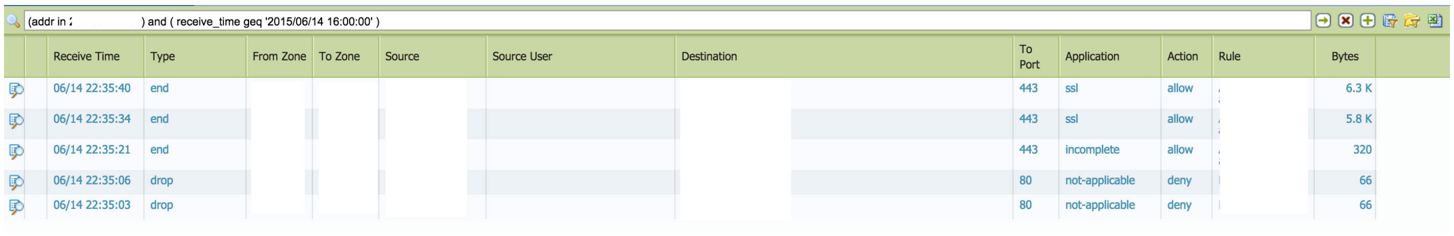
Next, I need to confirm this IP is associated with the user in question. For that I refer to my firewall logs to see if I have had any previous connections to or from that IP address within the time window referenced by the two-factor logs.



(addr in [redacted]) and (receive_time geq '2015/06/14 16:00:00')

		Receive Time	Type	From Zone	To Zone	Source

The above screen shot shows my first query to the firewall. I am asking it to show me all the connections (source or destination) for the IP address that I found in the two-factor logs, beginning a few minutes before the first suspected access attempt.

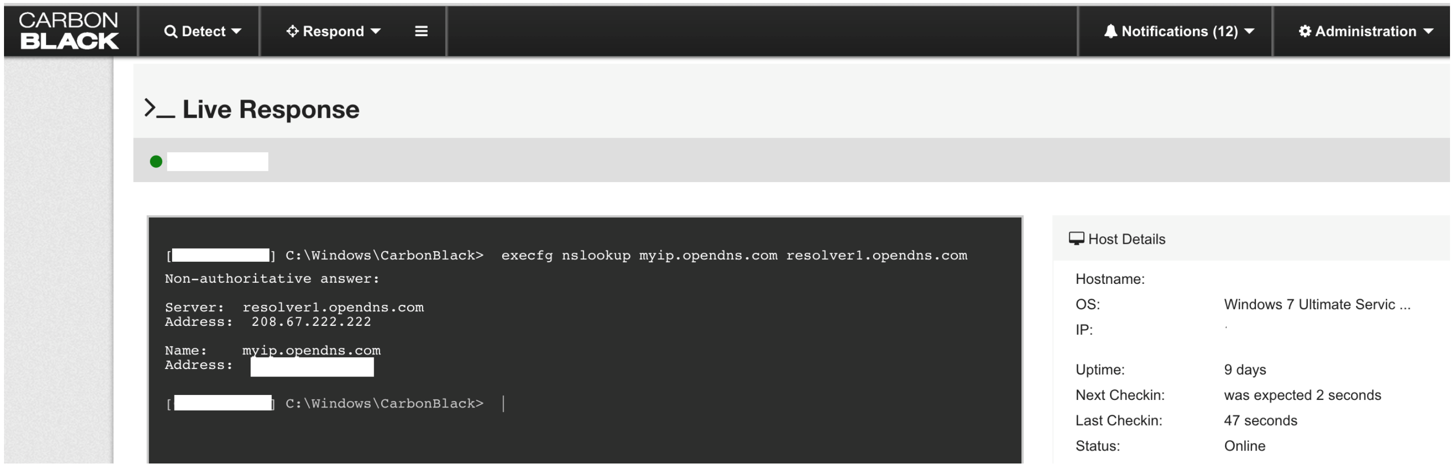


(addr in [redacted]) and (receive_time geq '2015/06/14 16:00:00')

		Receive Time	Type	From Zone	To Zone	Source	Source User	Destination	To Port	Application	Action	Rule	Bytes
		06/14 22:35:40	end	[redacted]	[redacted]	[redacted]		[redacted]	443	ssl	allow	[redacted]	6.3 K
		06/14 22:35:34	end	[redacted]	[redacted]	[redacted]		[redacted]	443	ssl	allow	[redacted]	5.8 K
		06/14 22:35:21	end	[redacted]	[redacted]	[redacted]		[redacted]	443	incomplete	allow	[redacted]	320
		06/14 22:35:06	drop	[redacted]	[redacted]	[redacted]		[redacted]	80	not-applicable	deny	[redacted]	66
		06/14 22:35:03	drop	[redacted]	[redacted]	[redacted]		[redacted]	80	not-applicable	deny	[redacted]	66

Using the two-factor authentication logs and the firewall logs, I confirmed that only one account was receiving these strange lockouts, that all the authentication attempts were coming from one IP, and that that IP had previously been seen conducting legitimate traffic with our corporate network—so this was probably something on an employee's computer instead of a totally unknown machine.

For this step, I used Carbon Black Live Response (CBLR). I logged into their computer via CBLR and ran "execfg nslookup myip.opendns.com resolver1.opendns.com" which queries opendns for this hosts current external IP address. The returned results confirmed that this IP was the same as the IP in the two-factor authentication logs.



[redacted]

```
[[redacted]] C:\Windows\CarbonBlack>  execfg nslookup myip.opendns.com resolver1.opendns.com
Non-authoritative answer:

Server:  resolver1.opendns.com
Address:  208.67.222.222

Name:    myip.opendns.com
Address:  [redacted]

[[redacted]] C:\Windows\CarbonBlack>
```

Host Details

Hostname:	
OS:	Windows 7 Ultimate Servic ...
IP:	
Uptime:	9 days
Next Checkin:	was expected 2 seconds
Last Checkin:	47 seconds
Status:	Online

Now that I have answered the majority of my questions, it is time to figure out what made these requests.

Quick recap. At this point a few facts could be determined:

1. The user did not manually request these codes for access.
   1. Still unconfirmed at this point.
2. The user reported that they denied access to the requests.

    1. Confirmed via the two-factor authentication logs.
3. The source of the requests is unknown.
    1. Found a single source IP generating these requests.
4. The number of user accounts getting these requests is unknown.
    1. Only a single user account is generating these requests.
5. My coffee is getting cold
    1. And that makes me a sad panda.

But we still had many questions to answer:

1. Who/what requested access?
    1. A single user from a single location made these requests.
    2. Still unknown is what from that location made these requests.
2. Were they able to breach the corporate network?
    1. If they have the ability to generate requests from this user's system, they could be attempting other access as well.
3. Was that really coffee that I just drank?
    1. I really hope so.

Let's review the facts:

1. I have a source IP address and a cloud destination address.
2. I have a user name.
3. I have a time window in which to look.
4. I have confirmed via the two-factor authentication logs that all requests were denied.
5. I finish off my coffee because I know this is going to be a short search.

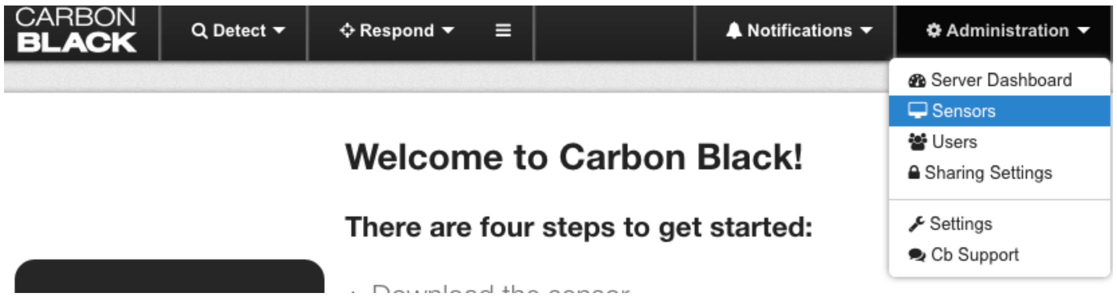
Time to find the culprit.

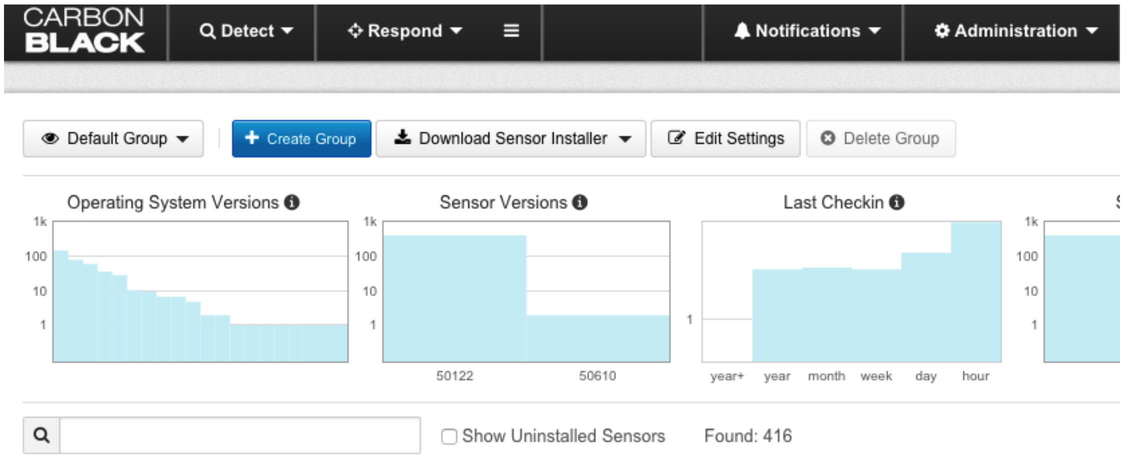
With the facts I have gathered, the standard method of finding the source of the request would require forensics, luck and a bit of creative profanity.

Luckily for me I have a full corporate license for both Bit9 and Carbon Black. Since the majority of my information is network-based, I am going to lean on Carbon Black for the initial analysis and confirm my findings via Bit9.

Searches:

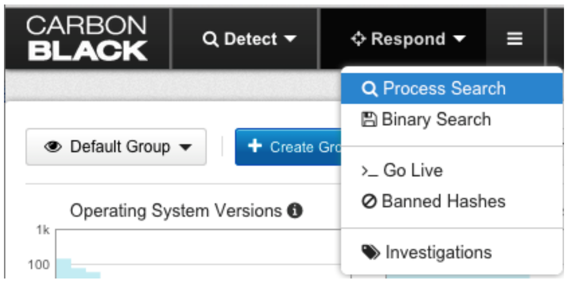
My first search in CB is to find what systems the user has. I do this with a quick search on the sensor page.





Since I know our company's naming scheme, I am able to search for specific strings to pull up a user's systems. These searches result in two hosts for that user, MacBook Pro and a Windows laptop.

Now that I know what sensors to search I need to start checking the systems for network connections.



I use a pretty basic search of "*sensor_id:54341 netconn_count:[1 TO *] start:[NOW-10MINUTES TO NOW]*" to see every process the sensor has spawned in the past 10 minutes that had a network connection.

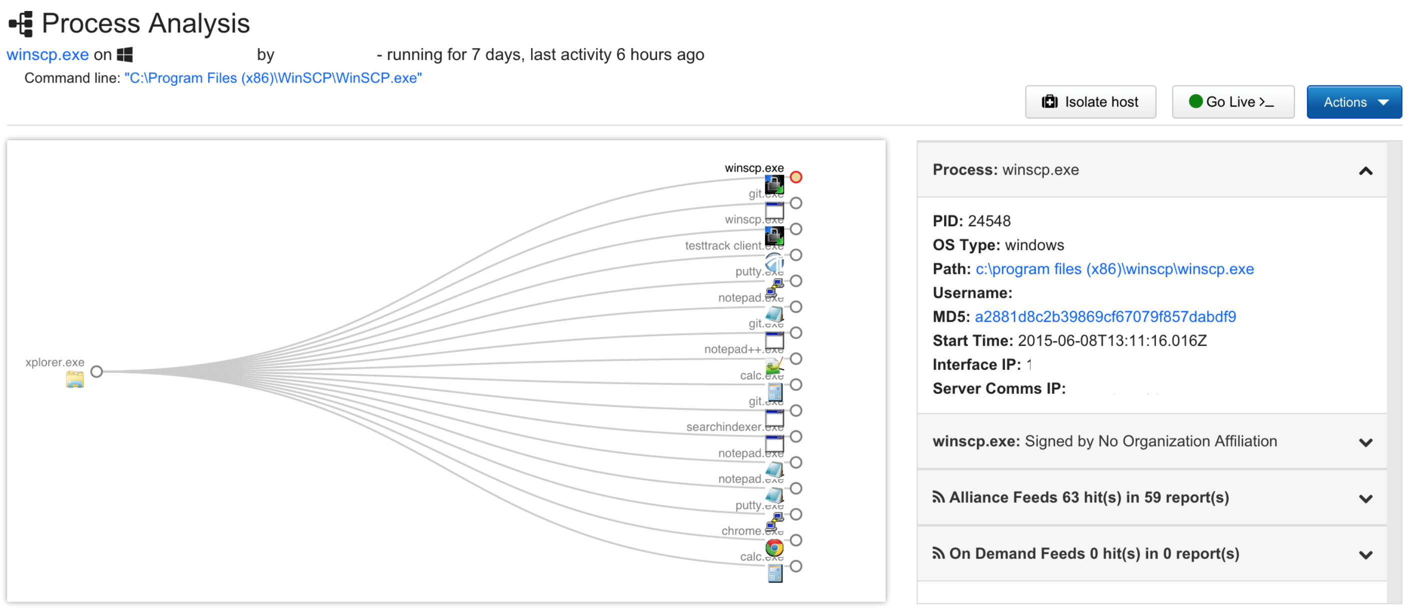
I chose the sensor ID because it will limit my search to a single system. I chose to view any process with a network connection of one or greater on this single system because I know logically that a network connection had to occur to connect to the cloud system to generate the authorization request. I chose to look back 10 minutes because the events had occurred five minutes prior to me starting this search.

The first system I checked was the MacBook Pro, which returned zero results! This means that the MacBook Pro did not generate any network connections during this time window and is now ruled out.

My search is down to one system!

I repeat my previous search and only modify the sensor ID to associate the search with the windows laptop now.

Bingo! 1 returned result!



Process Analysis

winscp.exe on by - running for 7 days, last activity 6 hours ago

Command line: "C:\Program Files (x86)\WinSCP\WinSCP.exe"

Process: winscp.exe

PID: 24548
OS Type: windows
Path: c:\program files (x86)\winscp\winscp.exe
Username:
MD5: a2881d8c2b39869cf67079f857dabdf9
Start Time: 2015-06-08T13:11:16.016Z
Interface IP: 1
Server Comms IP:

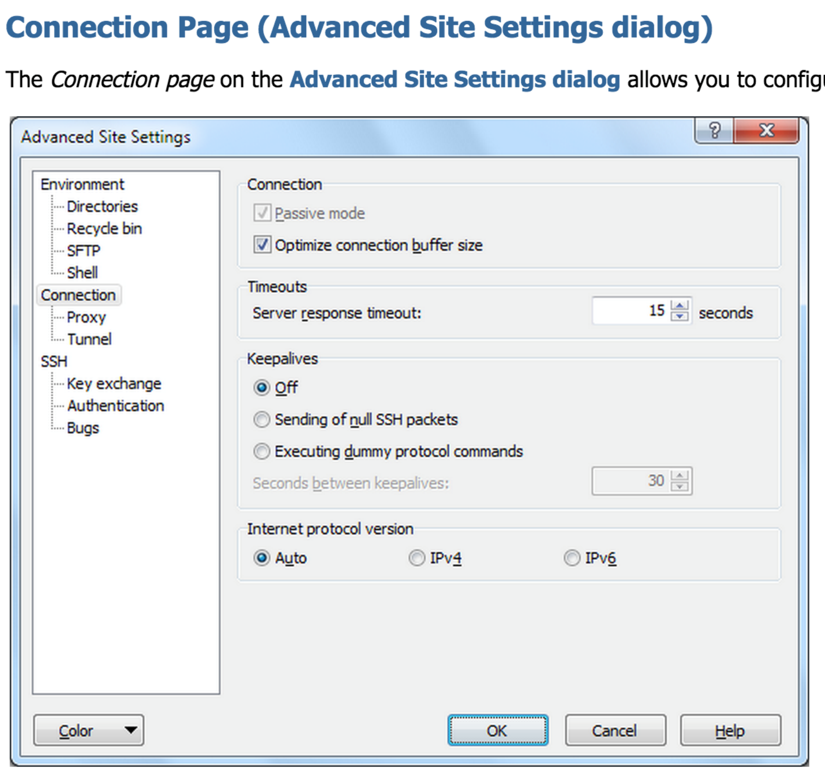
winscp.exe: Signed by No Organization Affiliation

Alliance Feeds 63 hit(s) in 59 report(s)

On Demand Feeds 0 hit(s) in 0 report(s)

The process in question was WinSCP, a popular file transfer application that supports SCP, FTP, SFTP, and other file transfer protocols. I use it myself when I need to transfer files between Windows and Linux-based systems.

I do a quick Google search to confirm my suspicions.



Connection Page (Advanced Site Settings dialog)

The *Connection page* on the **Advanced Site Settings dialog** allows you to configure general protocol-independent options for the connection.

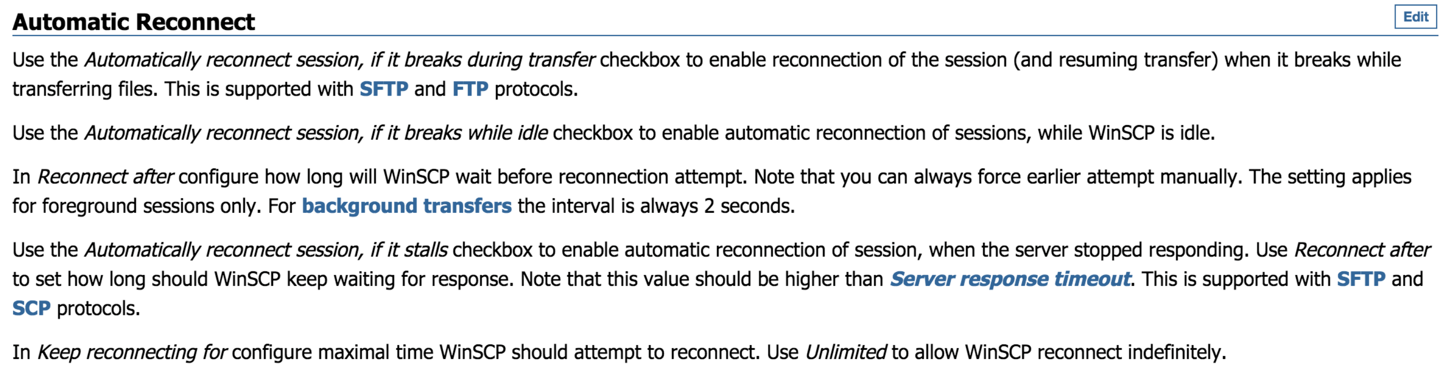
Advanced Site Settings

Connection
- [✓] Passive mode
- [✓] Optimize connection buffer size

Timeouts
- Server response timeout: 15 seconds

Keepalives
- [● Off]
- ○ Sending of null SSH packets
- ○ Executing dummy protocol commands
- Seconds between keepalives: 30

Internet protocol version
- [● Auto] ○ IPv4 ○ IPv6



Automatic Reconnect

Edit

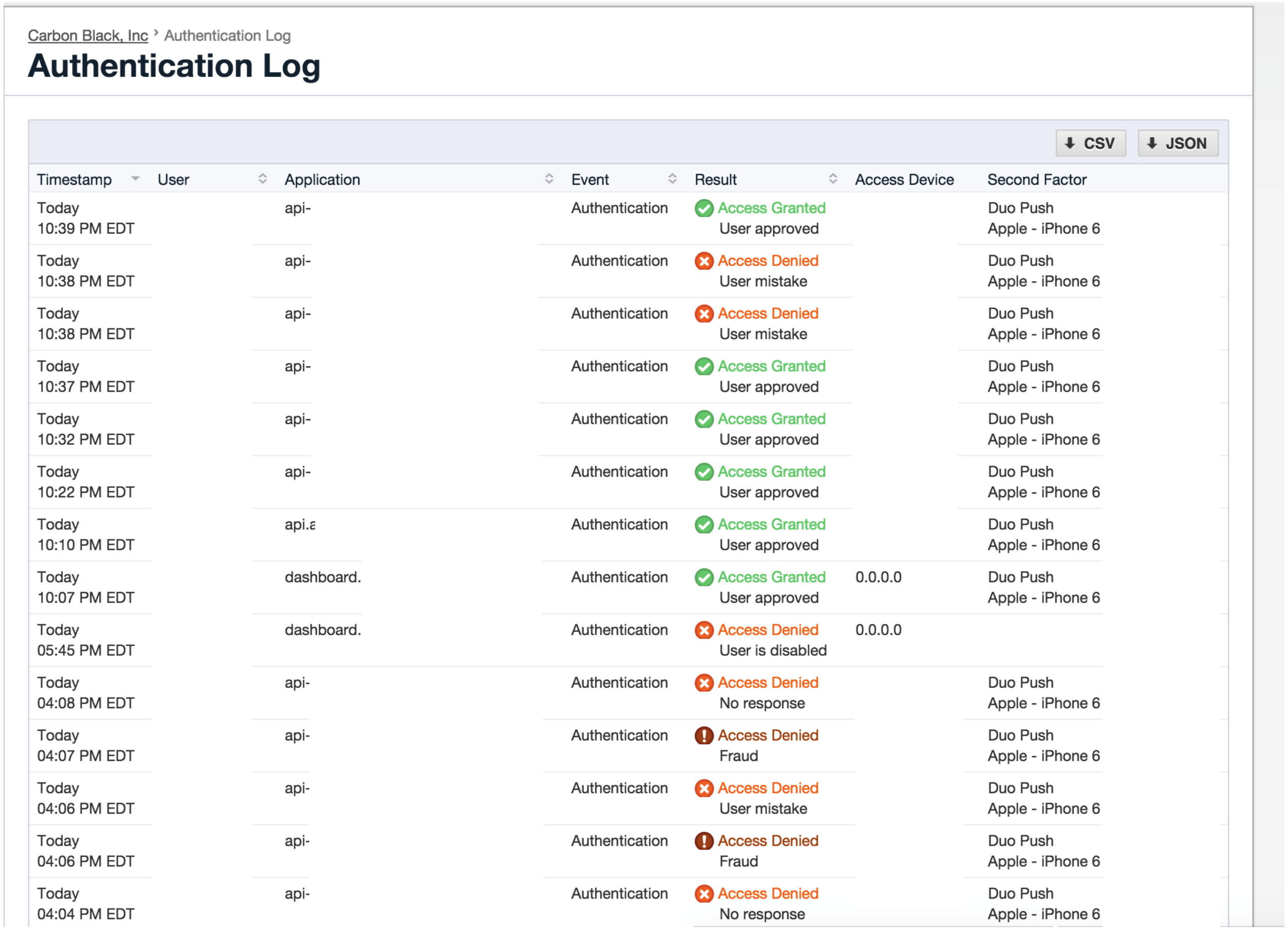
Use the *Automatically reconnect session, if it breaks during transfer* checkbox to enable reconnection of the session (and resuming transfer) when it breaks while transferring files. This is supported with **SFTP** and **FTP** protocols.

Use the *Automatically reconnect session, if it breaks while idle* checkbox to enable automatic reconnection of sessions, while WinSCP is idle.

In *Reconnect after* configure how long will WinSCP wait before reconnection attempt. Note that you can always force earlier attempt manually. The setting applies for foreground sessions only. For **background transfers** the interval is always 2 seconds.

Use the *Automatically reconnect session, if it stalls* checkbox to enable automatic reconnection of session, when the server stopped responding. Use *Reconnect after* to set how long should WinSCP keep waiting for response. Note that this value should be higher than ***Server response timeout***. This is supported with **SFTP** and **SCP** protocols.

In *Keep reconnecting for* configure maximal time WinSCP should attempt to reconnect. Use *Unlimited* to allow WinSCP reconnect indefinitely.



Carbon Black, Inc › Authentication Log

Authentication Log

CSV JSON

Timestamp	User	Application	Event	Result	Access Device	Second Factor
Today 10:39 PM EDT		api-	Authentication	Access Granted User approved		Duo Push Apple - iPhone 6
Today 10:38 PM EDT		api-	Authentication	Access Denied User mistake		Duo Push Apple - iPhone 6
Today 10:38 PM EDT		api-	Authentication	Access Denied User mistake		Duo Push Apple - iPhone 6
Today 10:37 PM EDT		api-	Authentication	Access Granted User approved		Duo Push Apple - iPhone 6
Today 10:32 PM EDT		api-	Authentication	Access Granted User approved		Duo Push Apple - iPhone 6
Today 10:22 PM EDT		api-	Authentication	Access Granted User approved		Duo Push Apple - iPhone 6
Today 10:10 PM EDT		api.ε	Authentication	Access Granted User approved		Duo Push Apple - iPhone 6
Today 10:07 PM EDT		dashboard.	Authentication	Access Granted User approved	0.0.0.0	Duo Push Apple - iPhone 6
Today 05:45 PM EDT		dashboard.	Authentication	Access Denied User is disabled	0.0.0.0	
Today 04:08 PM EDT		api-	Authentication	Access Denied No response		Duo Push Apple - iPhone 6
Today 04:07 PM EDT		api-	Authentication	Access Denied Fraud		Duo Push Apple - iPhone 6
Today 04:06 PM EDT		api-	Authentication	Access Denied User mistake		Duo Push Apple - iPhone 6
Today 04:06 PM EDT		api-	Authentication	Access Denied Fraud		Duo Push Apple - iPhone 6
Today 04:04 PM EDT		api-	Authentication	Access Denied No response		Duo Push Apple - iPhone 6

Armed with this knowledge, I call up the user and start asking some very specific questions. The conversation went a bit like this:

Me: "Hello User, thank you for alerting us about the possible unauthorized access attempt. I have conducted a quick investigation and I believe I have found the culprit. Can you answer a few follow up questions for me to confirm my findings?"

User: "Sure."

Me: "Do you use winSCP to transfer files from you windows laptop <HOSTNAME> to the host <CLOUDHOST> for your regular work?"

User: "Yes, I do. Just yesterday I uploaded a new version of ProgramX to <CLOUDHOST> to perform some testing."

Me: “When you initiate a connection to <CLOUDHOST> from your windows system, do you get a text prompting you for a two-factor authentication code?”

User: “Yes I do. Every time I open WinSCP to upload a new file I get prompted.”

Me: “Excellent. I think I know what happened. Can you do me a favor and help me reproduce the issue that I believe happened?”

User: “Sure thing. Just let me log in.”

Me: “Can you please upload a file to <CLOUDHOST> via WinSCP?”

User: “File uploaded.”

Me: “Are you still connected to your transfer session in WinSCP?”

User: “Yes, I am.”

Me: “Are you connected via wireless or wired connection?”

User: “Wired.”

Me: “Can you disable your wireless card and pull out the Ethernet cable from your laptop please?”

User: “Done.”

Me: “Are you seeing a reconnection attempt from WinSCP?”

User: “Yes, I am.”

Me: “Can you plug the Ethernet cable back in please?”

User: “I just got another text asking for approval. I think you need to disable my account again. I think they are attacking again.”

Me: “Actually you just helped me confirm my theory and reproduce what happened.”

User: “I did?”

Me: “Yes, you did. WinSCP was trying to reinitiate a connection that you did not close out of fully. This reconnect attempt tried to connect to a stale session, which caused WinSCP to then try and initiate a new session, which then required a new second factor code to connect. To confirm this we just forced that behavior to happen again manually while I was watching. I can see in the logs that the behavior was repeated. Thank you for your time today and for reporting the anomalous behavior.”

The one thing I didn’t show in screen shots is the results from my Bit9 searches. The reason I didn’t include them here is because there were no interesting new files on either of the systems during this time period. The Bit9 data was able to confirm for me that no application with Internet connectivity executed during this time window. That tells me that the process was already running during the time window and I used Carbon Black to confirm process spawn time and match up my results.

And that, in a nutshell, is how I quit freaking out over strange network traffic. It took me less than 10 minutes from the original alert to confirming my hypothesis with the user.

Until next time, remember my motto: “Flag it, Tag it and Bag it.”

Now, I’m going to get back to my coffee.

12 5

17
Shares

Tags: bit9 Carbon Black detection information security network traffic Prevention Response Ryan Nolette security operations

More Posts

June 20, 2016 / *by Ryan Murphy*

June 20, 2016 – Morning Cyber Coffee Headlines – Summer Solstice Edition

Good morning! Sit with Carbon Black this morning over a cup of coffee (or tea) and browse a few industry headlines to get the

March 18, 2016 / *by Ryan Murphy*

March 18, 2016 – Morning Cyber Coffee Headlines – March Madness Edition

Good morning! Sit with Carbon Black this morning over a cup of coffee (or tea) and browse a few industry headlines to get the