Deception in Depth - Building Deceptions from Breaches

blog.spookysec.net/DnD-building-from-breaches/

16 Sep 2022

So, here we are… Another day, another company hacked. Marketing and Sales teams are swarming like flies trying to pick up new clients claiming that the Uber hack could have prevented if they had just bought their SaaS platform or Dark Web monitoring service. *sigh* Well, this may be true, some of it may just be sales teams trying to make a quick buck, here's something you can do to detect adversarial presence faster. Adversary Deception

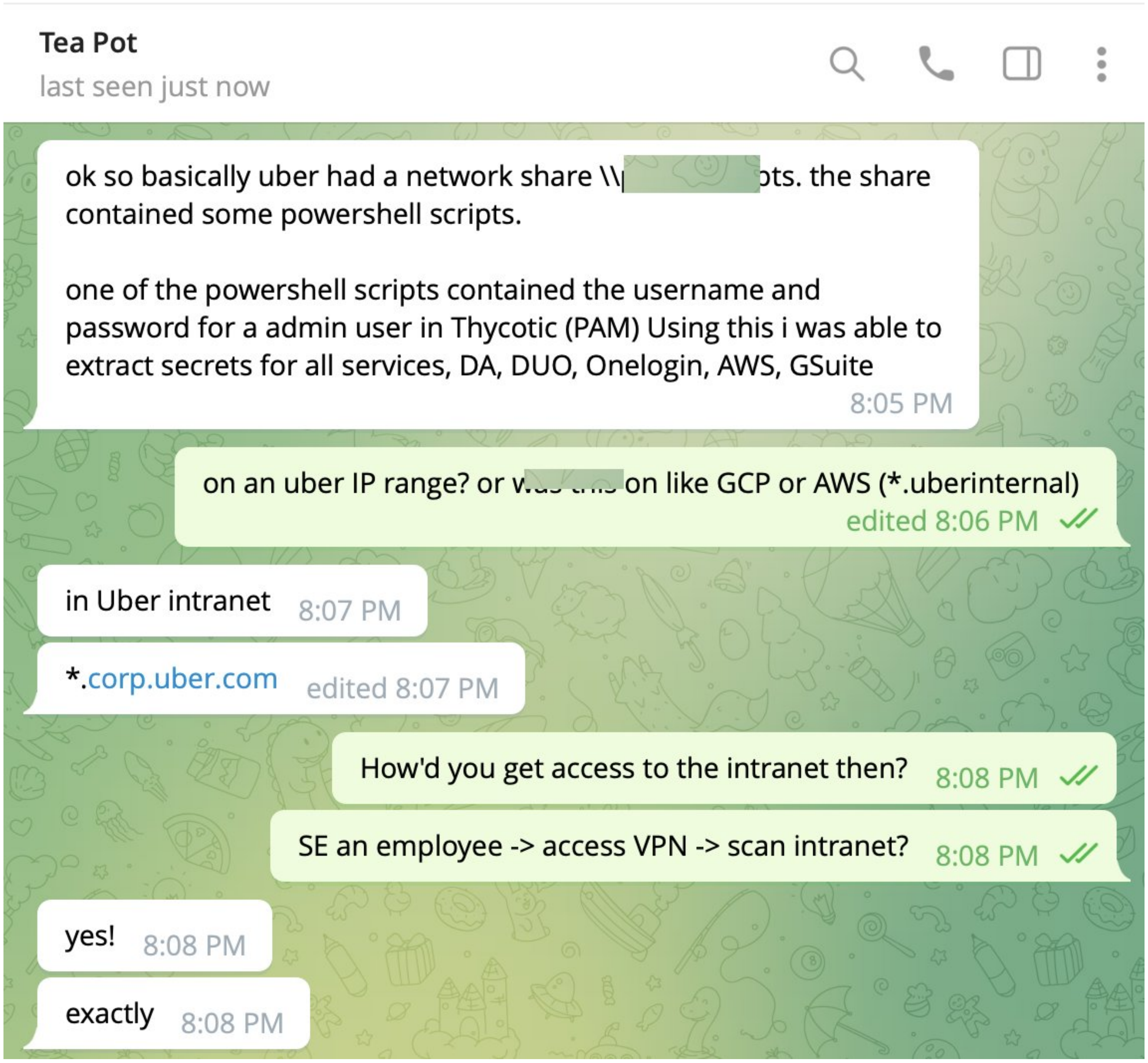
You know your environment far better than any adversary does (hopefully), so leverage this.

We know that when most adversaries breach a network, they don't just want to stop there. They want to collect data, learn more about their victim, move deeper into the network and ultimately gain Domain Admin/Enterprise Admin access so they can move freely throughout their environment. This is where the theory of deception starts coming into play, we know their goals, we know what they're after, so let's paint a picture of a tunnel at the end of a dead end road.

Disrupting Their Processes

As stated before, we know roughly what their goals are, so what do their processes look like? Thats a great question. There’s this age old saying “Don’t ever let a breach go to waste”. This is particuarly important with breaches like Uber where adversaries are **way too cocky** and they disclose their TTPs. Fortunately this helps us develop better deceptions so we can detect adversary presence faster.

So, let’s take a look at the information they provided from the Uber breach.



Tea Pot
last seen just now

ok so basically uber had a network share \\[redacted]pts. the share contained some powershell scripts.

one of the powershell scripts contained the username and password for a admin user in Thycotic (PAM) Using this i was able to extract secrets for all services, DA, DUO, Onelogin, AWS, GSuite 8:05 PM

on an uber IP range? or [redacted] on like GCP or AWS (*.uberinternal) edited 8:06 PM

in Uber intranet 8:07 PM

*.corp.uber.com edited 8:07 PM

How'd you get access to the intranet then? 8:08 PM

SE an employee -> access VPN -> scan intranet? 8:08 PM

yes! 8:08 PM

exactly 8:08 PM

*Image Credit: @hacker_*

This screenshot was proivded by Corben Leo @(hacker_). In here, we can see three key pieces of information:

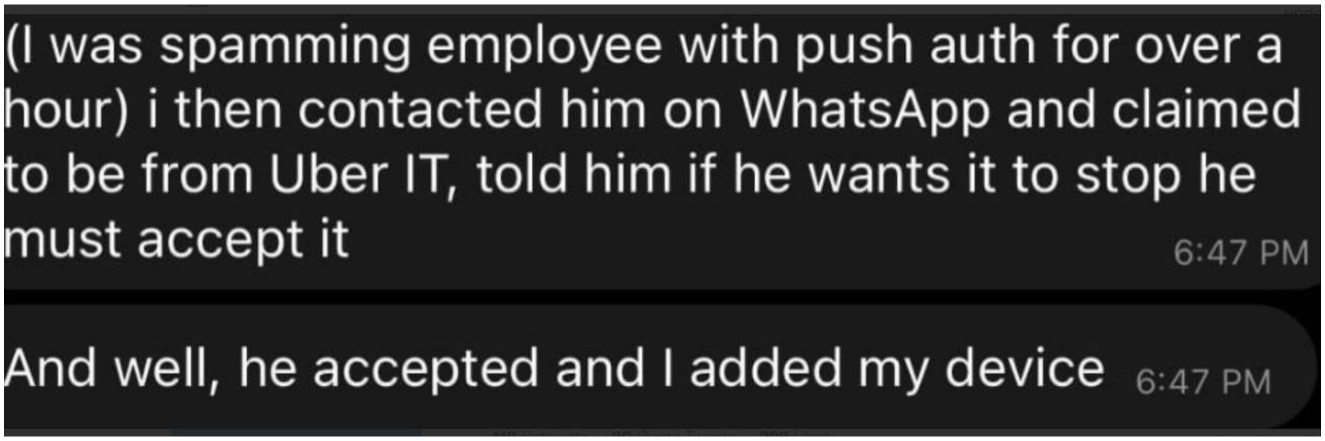
1. Initial Access Vector
2. Reconisance Information
3. Privilege Escalation

I'm going to make some inferences here; some of this is going to be backed by screenshots taken by the threat actor, some of it is influenced by my personal knowlege of Red Team & Pentest operations, some of it may be accurate or inaccurate; Ultimately, I wasn't there, but it will still be applicable regardless of accuracy, so let's get down to it.

Initial Access

This was claimed to be done via Social Engineering an employee (This could have been via phishing, vishing, smsishing, or any of the other ishings) to give up some piece of information leveraged by the attacker. This could have been a password *or* OTP/push authentication method. There is generally not much we can do in this front other than **educate the end users**. This isn't the main focus of this post, so we'll move on.

Additional information was disclosed; its accuracy is questionable, I don't know the original source, but here it is:


(I was spamming employee with push auth for over a hour) i then contacted him on WhatsApp and claimed to be from Uber IT, told him if he wants it to stop he must accept it 6:47 PM

And well, he accepted and I added my device 6:47 PM

Image Credit: Please contact me if you know the original source!

This could be the aspect of social engineering we're looking for. Dangerous. Once again, educate the end users. Ensure they know how to contact the Security Operations Center/Security Department to report these kinds of things. If you have a 24/7 SOC, great. Make them know this, if you don't again, make them know this.

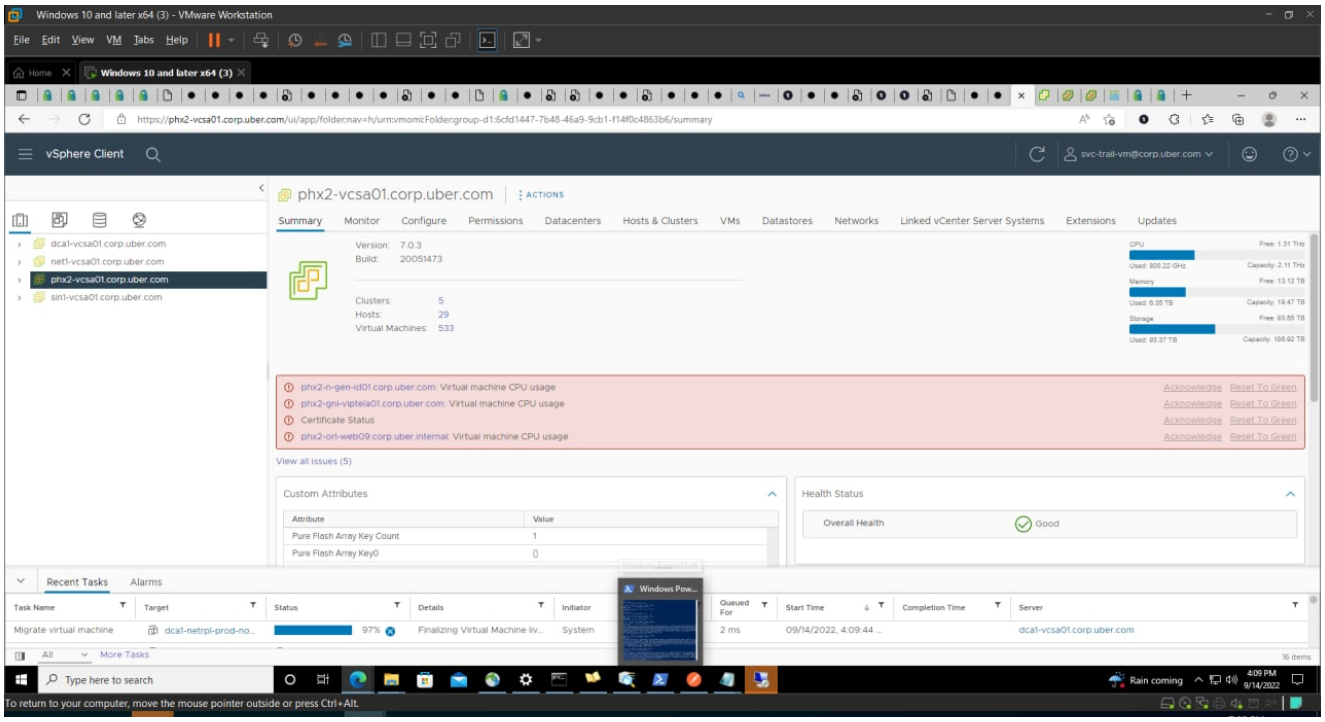
Reconnissance Information

From the screenshots we've seen so far, we know that they simply accessed a "Network" share (most likely SMB, possibly NFS?). But, how did they discover this Network share? What tools did they use? These are all really important questions. Making an educated guess, SMB shares can be enumerated using tools from PowerView like Invoke-ShareFinder/Invoke-FileFinder and other tool suites like PowerSploit with Find-DomainShare. These are all tools you should be using to detect and enumerate on your network as well. There's also generic PowerShell scripting one can do with Get-DomainComputer and Get-SmbShare, so don't always assume a tool is in use.

Anyways - This provides a great base for a Deceptive Object. We could add a decoy SMB share with full **read** access onto any device of your choosing. I personally recommend this be a dedicated device **only** for this purpose. If you do so, make this device a wolf among sheep. Ensure it's in the same OU as other devices (i.e. file server lives in the Servers OU) and make sure your naming scheme is consistent.

The next (and arguably most important step) is to monitor for SMB connection attempts. You could use a tool or service like Corelight or Bro, or utilize Windows Event logs (Microsoft-Windows-SMBServer/Connectivity and Microsoft-Windows-SMBServer/Security). How you do it doesn't exactly matter as you've verified logging is working in full expected capacity.

This next screenshot provides some context that may be helpful if you read into it enough:



Windows 10 and later x64 (3) - VMware Workstation

https://phx2-vcsa01.corp.uber.com/ui/app/folder;nav=h/urn:vmomi:Folder:group-d1:6cfd1447-7b48-46a9-9cb1-f14f0c4863b6/summary

vSphere Client — svc-trail-vm@corp.uber.com

- dca1-vcsa01.corp.uber.com
- net1-vcsa01.corp.uber.com
- phx2-vcsa01.corp.uber.com
- sin1-vcsa01.corp.uber.com

phx2-vcsa01.corp.uber.com — ACTIONS

Summary | Monitor | Configure | Permissions | Datacenters | Hosts & Clusters | VMs | Datastores | Networks | Linked vCenter Server Systems | Extensions | Updates

Version: 7.0.3
Build: 20051473

Clusters: 5
Hosts: 29
Virtual Machines: 533

CPU — Free: 1.31 THz — Used: 800.22 GHz — Capacity: 2.11 THz
Memory — Free: 13.12 TB — Used: 6.35 TB — Capacity: 19.47 TB
Storage — Free: 93.55 TB — Used: 93.37 TB — Capacity: 186.92 TB

- phx2-n-gen-id01.corp.uber.com: Virtual machine CPU usage — Acknowledge Reset To Green
- phx2-gni-viptela01.corp.uber.com: Virtual machine CPU usage — Acknowledge Reset To Green
- Certificate Status — Acknowledge Reset To Green
- phx2-orl-web09.corp.uber.internal: Virtual machine CPU usage — Acknowledge Reset To Green

View all issues (5)

Custom Attributes

Attribute	Value
Pure Flash Array Key Count	1
Pure Flash Array Key0	0

Health Status — Overall Health: Good

Recent Tasks | Alarms

Task Name	Target	Status	Details	Initiator	Queued For	Start Time	Completion Time	Server
Migrate virtual machine	dca1-netrpl-prod-no...	97%	Finalizing Virtual Machine liv...	System	2 ms	09/14/2022, 4:09:44 ...		dca1-vcsa01.corp.uber.com

All — More Tasks — 16 items

Windows Pow...

Type here to search — Rain coming — 4:09 PM 9/14/2022

To return to your computer, move the mouse pointer outside or press Ctrl+Alt.

Image Credit: @praise_terryd

This screenshot shows some valuable information, if you look in the Task bar PowerShell is a running process. As I previously mentioned, PowerShell is a very popular tool to use for searching for SMB Shares. We can also see some other tools like Advanced Port Scanner (lol noobs, use nmap ffs [sorry, I had to]) and ADExplorer from SysInternals, it is essentially the "Active Directory Users and Computers" commandlet but portable. All of these tools can be used to easily (and effectively) identify key assets within the domain. If your OU structure is anything like my preferred structure, there will be an OU for Devices, then a sub-ou for Domain Controllers, one for Servers, and one for Workstations with plenty of sub-sub-ous dividing workstations, servers, and paw workstations. Anyways - AD nerd out aside; Make sure the honey devices look and feel like they're in the right place. Always leverage existing infrastructure if you can.

Oh boy, that was a lot.

Privilege Escalation Information

So, this one is probably going to be the shortest section. It's short, it's sweet, they gave us the most information.

- The adversary found an open SMB share
- They located credentials to their Privileged Access Management System within a PowerShell Script
- They signed into the PAM system and they got Domain Admin access
- Finally, they pivoted throughout enterprise management and security infrastructure

    Slack, vSphere, G-Suite, AWS, HackerOne, SentinelOne, Thycotic, etc.

Now that we've completed our analysis on how they got to Domain Admin, we can then start developing Deceptions around this.

Developing Deceptions

Putting everything together, you could do a number of things to develop deceptions for detection purposes. Let's start:

- On a open portion of the network, you could deploy **Domain Joined** windows hosts with Firewalls disabled and SMB shares where Everyone has Read/Write privileges.
    - You could take this a step further and customize this and develop a story. Is this an IT Infrastructure SMB Share? Is there IT Service manuals? Documentation? Spreadsheets? Network Maps? Scripts? The world is yours. Invent a story
    - Use **real usernames** with fake credentials. Track login failures and correlate them with file reads to determine if you have a problem vs a serious problem.
    - Monitoring could then be deployed on the host; when a user connects with credentials via SMB, send an alert to your Security Operations Center.
- You could build out deceptive **PAM solutions**; use Single File (a Chrome/FireFox addon) to create a ""phishing"" pages for the attacker. Tie this into your scripts from the previous idea. Build out a HoneyNet. Make them think they hit the jackpot and found real valid credentials to your PAM system.

    Once a user authenticates with your planted credentials, generate an alert. Send it off to the Security Operations Center
- Piggy back off of existing File Servers.

    This takes the element of building a real looking host away. It'll often just be a quick chat with your Identity Team (Domain Admins) and they'll be happy to help out as long as it provides no new risk.
- Create Deceptive Service Accounts

    Kerberoasting is still relevant! Throw an easy-to-crack password out there. You never know.

        User accounts are free to make. It literally costs nothing and can improve your deception footprint massively!
- Develop intelligent alerting on Domain Enumeration; Plant real Domain Admin accounts that are designed to look like a very inticing target and monitor for probing of those account(s).

- Develop detection for SharpHound via planting real deceptive user sessions (see my blog post on Spoofing privileged users via Remote Registry)
- Sprinkle HoneyDocuments in there.

  Thinkst Canary Tokens is an awesome free service that will notify you when a HoneyDoc is opened. They support a whole bunch of different types of tokens. You never know who might be poking and probing.

Lastly, be creative. Theres a ton of things you can do for little to no cost to develop detections.

Closing Remarks

Anyways - I hope you all enjoyed this entry in the Deception in Depth series. I have absolutely no idea when the next post will come out. I don’t have anything hidden up my sleeve, but that doesn’t mean I’ll stop being passionate about deception. If you’d like to chat with me about Deception, feel free to reach out to me on Twitter or LinkedIn. I’d love to discuss new ideas, help develop and flesh out new methods, or even give some advice or share some experiences if Deception is something you’re interested in bringing to your organization. As always - Thank you for all for the love and support <3

~ Ronnie

Comments