COMMAND AND CONTROL

Using Covenant C2 to Manage Access and Lateral Movement Post Exploitation

Sarah Yeoell

Capstone Project - BAS Cybersecurity

Clover Park Technical College

Lakewood, WA

March 18, 2022

Abstract

Hacking and Penetration Testing tutorials, textbooks, and courses put most of the focus on finding and exploiting software vulnerabilities to gain access to a target machine. Once access is gained, the focus shifts to gaining root and finding a flag. This leads to a “smash and grab” approach with little regard to persistence, operational security, lateral movement, or asset management once within the target network. APTs, cybercrime groups, and even security consulting firms operate with these considerations in mind. This project is intended to explore the theoretical operations of a more advanced adversarial campaign, focusing on use of the Covenant C2 [Command and Control] framework for centralized operations.

Contents

1 Introduction

The initial thoughts for this project came about from my desire to learn as much as I could about offensive security/penetration testing. During my time in the CNISS/NOS Associates degree and BAS-Cybersecurity programs at Clover Park Technical College, I was exposed to much of the defensive side of information and computer security. Best practices for network administration, Windows Server administration, Linux servers, forensics, database security, even incident response.

There was a component of penetration testing, but for me it fell into the same pitfalls mentioned in the abstract of this report. Penetration Testing and Hacking tutorials and classes are always focused on the beginner, and do not tread much further than teaching how to use nMap and Metasploit to attack vulnerable systems.

When I participated in the 2021 Pacific Rim Collegiate Cyber Defense Competition, our team had to manage and defend multiple systems against a Redteam who's only goal was to break in and make life hard for us. Given there were multiple teams of defenders, and multiple teams of attackers, there surely had to be a way that they were managing their access to our networks in a centralized manner. During debrief, it was revealed to us that they had been using a Command and Control framework known as MythicC2 to manage their operations against us.

This lead me to the question that spawned this project, and the problem I wanted to solve:

"How does command and control work, and how does someone learn how to use it?"

It is my belief that studying this area of security is important because it is highly relevant in the real world, as threat actors use these technologies and understanding how they work can defenders more of an edge of knowing what they're up against. Some argue that exposing this knowledge to the public only helps educate new threat actors, but I think that falls under the same attitudes as security through obscurity. Ultimately for this project I wanted to give an overview of a C2 framework, and demonstrate its use against an Active Directory environment, giving future students a good starting platform for a more in depth project.

2 Literature Review

Reviewing literature around the subject of C2 frameworks turned out to be a somewhat difficult task. As with much of offensive security, much of the information surrounding the subject is either institutional, recorded as tooling documentation, or as tutorials in online blog posts and videos. Some larger players in the field of offensive security like Jeff Dimmock and Raphael Mudge have created repositories of tools and techniques as well as some tutorials and blogs on the subject of C2 operations. In Mudge's case, they focused on his product "Cobalt Strike", which is a sold as a "Adversarial Simulation Framework" and not intended to be used maliciously. As for Dimmock, he is the head of Adversarial Simulations at SpectreOps, and his writings focus on Red Team practices, with the assumption that most tools mentioned are familiar to the user.

That being said, there were various sources I did find helpful during my initial searches, and some later in the project. Some were research by security organizations into the prevalence of some tool-sets used by malicious actors, which helped to confirm that these tools were actually being used maliciously. Some were technique based, giving insight into larger operations of adversarial actions, and others were specific tooling tutorials. The literature listed below is what I found to be the most helpful in gaining insight into the prevalence of C2 usage, adversarial infrastructure, and specific usage of the Covenant C2 framework.

2.1 Research

Blackford, D. and Larson, S. (2021). Cobalt strike: Favorite tool from APT to crimeware. Proofpoint. Retrieved January 10, 2022, from https://www.proofpoint.com/us/blog/threat-insight/cobalt-strike-favorite-tool-apt-crimeware

This blog post by ProofPoint employees reveals the results of research into the prevalence of Cobalt Strike being used in recent cybercrime. Blackford and Larson used data both from internal sources at ProofPoint, as well as data from other security vendors to track the usage of Cobalt Strike by commodity malware actors and advanced persistent threat APT groups. What they found was since 2017, usage of Cobalt Strike had been gaining popularity with major cybercrime groups and APTs. However since 2019, its usage among smaller actors had exploded, and only 15 percent of observed attacks were attributed to established large groups.

2.2 Techniques

Dimmock, J. (1999). Red Team Infrastructure Wiki. Retrieved January 15, 2022, from https://github.com/bluscreenofjeff/Red-Team-Infrastructure-Wiki

This GitHub repository by Jeff Dimmock serves as a resource for overall Red Team infrastructure setup and management. Its contents provide a mostly tool agnostic set of practices and techniques for building resilient and operationally secure infrastructure for adversarial simulation. The sections on C2 do focus on Cobalt Strike (being the industry standard) and Empire/Powershell Empire, as those were products of SpectreOps at the time. I found this to be a valuable resource when considering the initial design of my own lab attack infrastructure.

Mudge, R. (2013). Tradecraft (Playlist). Retrieved January 10, 2022, from https://www.youtube.com/playlist?list=PL9HO6M_MU2nesxSmhJjEvwLhUoHPHmXvz

Raphael Mudge created a fantastic series of videos on Red Team tactics and tradecraft using Cobalt Strike, and some videos not using Cobalt Strike. While he does not demonstrate the tactics and techniques live, he provides a holistic approach to an engagement, including using C2, but also importantly noting on techniques and tactics to use once inside of a network. Techniques for credential gathering, persistence, and lateral movement being the most helpful towards the writing of this report.

Dimmock, J. (2018). HTTPS Payload and C2 Redirectors. Retrieved March 13, 2022, from https://bluescreenofjeff.com/2018-04-12-https-payload-and-c2-redirectors/

After reading the Red Team Infrastructure Wiki, this blog post provided a solid look at redirection techniques for C2 and Payload traffic when considering Operational Security and Infrastructure resilience. It covered everything from quick and dirty redirection to advanced filtered redirection based on specific web queries using Apache's mod_rewrite. This post provided a good insight and some design considerations when initially planning my project.

2.3 Tutorials + Most Used Resources

Cobb, R. (2019). Covenant wiki. Covenant Wiki. Retrieved January 10, 2022, from https://github.com/cobbr/Covenant/wiki

Ryan Cobb's Command and Control framework Covenant was the focus of my project. Cobb created a multiplatform C2 framework using .NET. The project wiki was an invaluable resource for the setup and familiarization with Covenant, and I wouldn't have gotten nearly as a far as I did without it.

Infinitelogins. (2020). Installing Covenant C2 on windows and reviewing basic features. YouTube. Retrieved January 10, 2022, from https://www.youtube.com/watch?v=3-I_UZfqLZA

InfiniteLogins' video tutorial on the actual basic use of Covenant C2 and setting up a launcher with a Grunt. It was a simple video to follow, but provided enough context to get a user up and running. I found it helpful as he went a little more in depth than the Covenant Wiki did on how a grunt actually connects back to the C2 server with the listener when you launch it through PowerShell.

Art, S. (2017a). Pentest Home Lab - 0x2 - Building Your AD Lab on Premises. Retrieved March 13, 2022, from https://sethsec.blogspot.com/2017/06/pentest-home-lab-0x2-buildingyour-ad.html

Art, S. (2017b). Pentest Home Lab - 0x3 - Kerberoasting: Creating SPNs So You Can Roast Them. Retrieved March 13, 2022, from https://sethsec.blogspot.com/2017/08/pentesthome-lab-0x3-kerberoasting.html

Seth Art's "Pentest Homelab" series provides a simple tutorial of suggestion on building a virtual Active Directory Lab for pentesting techniques. He covers setting up multiple servers, some users, and creating Service Principal Names for Kerberoasting attacks. These two posts in the series specifically formed much of the guidance in the design of my own Active Directory lab. Despite using Windows Server 2012, Kerberoasting is still a valid attack path in today's environments, and thus this resource on how to create an environment based upon it ended up being incredibly useful.

3 Methodology

3.1 Plan

The overall plan for the project, with timing based on the schedule of the class, and activities selected based the knowledge I gained during the CNISS program to date, and knowledge gained previously outside of education through personal projects, research, and interaction with offensive security professionals was as follows:

- **Week 1** – Research
- **Week 2** – Proposal Writing
- **Week 3** – Build an Active Directory lab using VMWare and Windows Sever 2016/Windows 10 LTSC. Manually create a pathway through the domain from Domain User to Domain Admin through credential harvesting and Kerberoasting.
- **Week 4** – Build a Covenant C2 Server using Ubuntu 20.04. Test launch Grunts on one of the domain computers to ensure functionality.
- **Week 5** – Build an Apache HTTP redirector using the information provided by Dimmock(2018) Conda(2020)
- **Week 6** – Attempt a full Adversarial Campaign against the AD environment
- **Week 7** – Reflect + Reset
- **Week 8** – Cloud Deployment of AD Lab + C2 Server (AWS)
- **Week 9** – Simulated Adversarial Campaign over the Internet
- **Week 10** – Final Report Writing

3.2 PoC Domain Setup

The planned concept for the project is illustrated in Figure 3.1. According to the proposed project timeline, the PoC network would be where a majority of the learning and project design occurs. After Week 7, the goal was to move the C2 server and Apache Redirector from Local Virtual Machines, onto a cloud provider with real registered domain names.

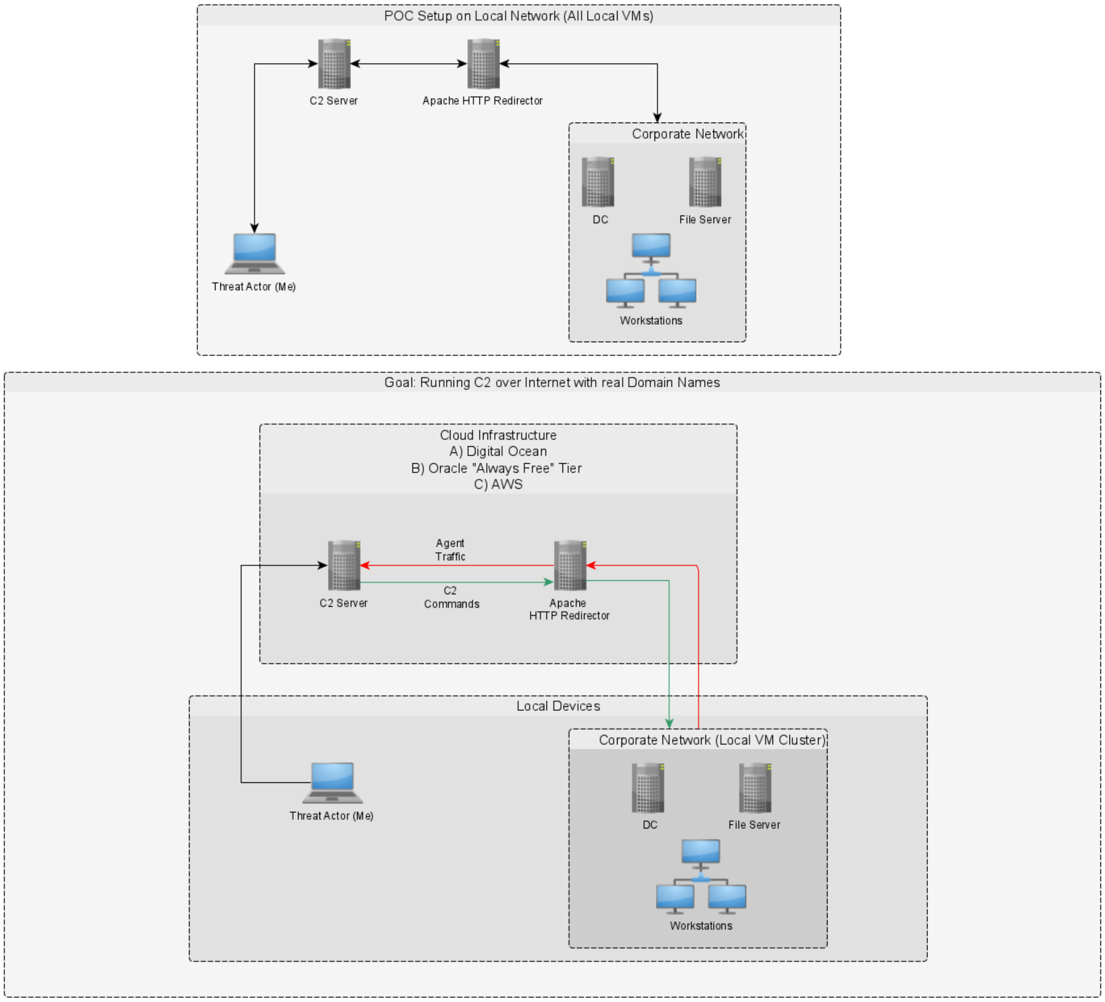

Figure 3.1: Initial project plan and stretch goals

This was translated into an PoC Domain design according to the following tables:

Computers			
Name	OS	IP	Roles/Notes
ALLIANCE-DC01	Win Server 2016	NAT: 192.168.178.130 LAN:172.16.10.1	Domain Controller DNS RRAS
ALLIANCE-SRV01	Win Server 2016	172.16.10.2	General Purpouse
ALLIANCE-DEV01	Win 10 LTSC	172.16.10.10	TicketService is LA Has SPN assigned
ALLIANCE-IT01	Win 10 LTSC	172.16.10.20	
ALLIANCE-WS01	Win 10 LTSC	172.16.10.30	Initial access vector
C2	Ubuntu 20.04	192.168.178.140	Covenant C2 Server
REDIRECTOR	Ubuntu 20.04	192.168.178.141	Apache HTTP/HTTPS Redirector Server

Users		
Username	Groups	Roles/Notes
CS	Domain Users	Unprivileged user that will launch first Covenant Grunt in network.
IT	IT Admins	IT Admins are members of Local Admin group on every workstation and SRV01 via Group Policy
DEV	DEV Admins	DEV Admins are members of Local Admin group on DEV01 machine
ADA	Domain Admins	The target domain admin account. Logged into DC and SRV01 (for credential stealing via mimikatz)
TicketService	Domain Users	Has an SPN set for DEV01 Member of Local Admin group on DEV01. Intended to emulate a service being run by the dev

3.3 Intended Attack Path

Because the intent of this project was to showcase lateral movement and compromised asset management, the pivot and privilege escalation path was designed to rely as little as possible on specific exploits and software vulnerabilities. Kerberoasting and other forms of credential harvesting were the main targets, as suggested by Seth Art(2017b) and Fatrozdianko(2019c).

Initially I had concerns that this path was potentially too simple. Consultation with industry professionals verified this path and methodology as valid for the purpose of demonstrating a C2 framework based on their experience in industry, being that "in most cases you are like 3 hops from DA [Domain Admin]. I have been as low as 1 hop." (Kelly, 2022)

1. Initial Access on ALLIANCE-WS01
    a. Launch a Covenant PowerShell HTTP Grunt and establish persistence
    b. Perform Active Directory Enumeration using Covenant tooling and other tools like BloodHound or PowerView/SharpView.
    c. Kerberoast TicketService and crack weak password or perform a Pass-The-Hash attack.
    d. Use TicketService account to pivot from ALLIANCE-WS01 to ALLIANCE-DEV01.
2. Pivot to ALLIANCE-DEV01
    a. Launch a PowerShell SMB Grunt via WMI on ALLIANCE-DEV01 and connect it back to HTTP Grunt on WS01.
    b. Use TicketService's Local Administrator access to harvest credentials using Mimikatz and obtain passwords/NTLM Hashes for DEV and IT.
3. Pivot to ALLIANCE-IT01 and ALLIANCE-SRV01 using IT account.

   a. Launch a PowerShell SMB Grunt via WMI on ALLIANCE-IT01 and ALLIANCE-SRV01, connect them back to Grunt on ALLIANCE-DEV01.
   b. Use IT account's Local Administrator Access to harvest credentials via Mimikatz and obtain password/NTLM Hash for ADA (Domain Admin) account.

4. Pivot to ALLIANCE-DC01 using ADA account.
   a. Launch a PowerShell HTTP Grunt via WMI on ALLIANCE-DC01 and establish persistence.
5. Success: Domain is fully compromised.

At the outset I was aware that this project was ambitious in scope, and potentially vulnerable to scope creep, but I felt the plan was detailed enough and had achievable milestones that were not critical to the basic function of Covenant. This would allow for readjustment or removal of milestones without putting the project in a dead stop.

4 Findings

The findings for this report are divided into two sections: Major Issues encountered and their remediation or adjustement, and Attack Path completion with explanations.

4.1 Major Issues

During the project I hit several stumbling blocks in my proposed weekly plan that necessitated alteration of the plan even before the Week 7 reflect and reset. These were:

1. **Week 5** - Apache HTTP C2 Redirection sever was not built

    - **Cause:** This week was busy due to unforeseen external events and I was unable to dedicate appropriate time to building and testing the C2 Redirector. I was able to deploy and install Apache with mod_rewrite, but was unsuccessful in configuration of mod_rewrite despite following the procedures outlined in Conda(2020).
    - **Remediation:** The decision was made to adjust the plan and ignore the C2 redirector. It was not directly necessary to the operation of the C2 server itself, and was originally planned as a demonstration of good operational security (OpSec).

2. **Week 6** - Attack Plan was not completed.

    - **Cause:** My VM host machine ran out of disk space with all VMs running. Multiple VMs reported disk space issues and I was unable to gracefully shut them down. After fixing this issue and restarting the host and VMs, the VMs exhibited unstable behavior and PowerShell Grunts would no longer activate. These behaviors continued to occur even after resetting to earlier checkpoints.

- **Remediation:** Week 7 was taken to reflect and reset. Week 8 was then used to rebuild the Active Directory environment from scratch, resulting in only the DC01, SRV01, and WS01 machines being built, for a simplified attack path.

4.2 Attack Path Completion

Because of the drive space incident in Week 6, the attack path was modified to be shorter, resulting in the following:

1. Initial Access on WS01
    a. Launch a Covenant PowerShell HTTP Grunt and establish persistence
    b. Perform Active Directory Enumeration using Covenant tooling and other tools like BloodHound or PowerView/SharpView.
    c. Kerberoast TicketService and crack weak password or perform a Pass-The-Hash attack.
    d. Use TicketService account to pivot from WS01 to SRV01.
2. Pivot to SRV01
    a. Launch a PowerShell SMB Grunt via WMI on SRV01 and connect it back to HTTP Grunt on WS01.
    b. Use TicketService's Local Administrator access to harvest credentials using Mimikatz and obtain passwords/NTLM Hashes for IT Account [Now a Domain Admin].
3. Pivot to DC01 using IT account.
    a. Launch a PowerShell HTTP Grunt via WMI on DC01 and establish persistence.
    b. Launch a Powershell SMB Grunt via WMI on DC01 and connect it back to WS01 or SRV01.
4. Success: Domain is fully compromised.

4.2.1 Completed

During Week 9, I was able to complete 1a, 1c, 1d, 2a, and 2b of the modified attack path. Demonstration of these techniques is shown in the accompanying video file to this paper.

4.2.2 Not Completed

I was unable to complete the following parts of the modified attack path. Reasoning is expanded upon below:

1. Initial Access on WS01

    b. Perform Active Directory Enumeration using Covenant tooling and other tools like BloodHound or PowerView/SharpView.

    This was intensely frustrating to not complete as I felt it was an important part of the demonstration. Uploading BloodHound worked, however when trying to run it using Covenant's **ShellCmd** function, BloodHound would error out. Even trying to run it locally resulted in errors.

    Covenant also has a function called **PowerShell Import**, which is intended to allow PowerShell based Grunts to add PowerShell scripts into their memory and execute them similar to PowerShell's **Import-Module** or **Invoke-Expression**. My thinking, was to use this to import the **Powerview.ps1** script from PowerShellEmpire/PowerSploit with its Find-LocalAdminAcccess function to scan the domain for places that TicketService was a member of the Administrators group on the local machine. This also resulted in errors and even the Grunts crashing.

    There is another function, called Execute Assembly, which is intended to allow full executables to be loaded into the Grunt's memory and run them. I tried to use SharpView, a C# executable implementation of PowerView to use **Find-LocalAdminAccess** but this too resulted in errors.

I think most of these are related to the same issue: Because the domain was built and left intentionally bare, dependencies such as the dotNET run-time were not installed on the machines, like they would be in a corporate environment where other third party software is present.

3. Pivot to DC01 using IT account.
    a. Launch a PowerShell HTTP Grunt via WMI on DC01 and establish persistence.
    b. Launch a Powershell SMB Grunt via WMI on DC01 and connect it back to WS01 or SRV01.

   Covenant is a work in progress, and this was evident in its Launcher/Grunt generation and hosting. While having one SMB based grunt was fine, trying to chain a SMB grunt to an SMB grunt did not work for unknown reasons. Additionally, attempting to launch grunts on DC01 from the SMB grunt on SRV01 was unstable and caused the SRV01 Grunt to be lost multiple times.

4. Success: Domain is fully compromised.

While in this attack path Domain Admin credentials were gained, using them was difficult and persistence on the Domain Controller wasn't achieved. Thus I consider this as "Not Completed".

5 Conclusion and Recommendations

Overall, I feel that I achieved an acceptable level of success compared to my original plan. Every point in that plan which I did not directly achieve the goal of, was still a learning opportunity that was worth putting the time into. C2 as well as asset management during offensive engagements is important to understand for those wishing to enter offensive security, and I still believe its worth continuing to study.

Over the course of this project I did hit these large milestones:

- Built an Active Directory domain with specific configuration vulnerabilities.
- Deployed and instance of Covenant C2.
- Used Covenant C2 to pivot halfway through a small Active Directory domain starting at Domain User level privilege without the use of more common and real-time oriented tools like Metasploit or Meterpreter reverse shells and powerful privilege escalation exploits.

There are recommendations of things to keep in mind should a student wish to expand upon this project:

1. Open Source software projects are a tossup on function and stability.

   Covenant is visually well polished, but has its fair share of bugs and instability. At one point after clicking around menus too much, the dotNET runtime that it is built on threw an exception and crashed the whole server. It is a promising project, however I don't feel it is easy for someone to learn without much more dedicated time and potentially instruction from someone with real industry experience.

2. Familiarize yourself with all your tools, and test their function extensively.

I got stuck on the Active Directory Enumeration. I was attempting to use tools that are industry standard from an offensive security standpoint (Bloodhound, Powersploit), but they weren't working for reasons I didn't understand. I had not had much exposure to them before starting this project, and thus did not have the familiarity with them to debug the environment and recognize if I was missing dependencies or just using them incorrectly.

3. Dedicated Command and Control solutions that are available to the public are often not documented well or in depth, and finding information on them past basic usage is difficult. If not impossible.

   Information on actual use of more advanced C2 solutions is hard to find. HelpSystems holds Cobalt Strike behind an expensive corporate pay-wall, MythicC2 is open source and has decent documentation compared to other options but it is still difficult to learn, and Covenant C2 is a fledgling project with promise but usability and documentation issues. Tutorials and explanations of features of any of them are hard to find and very surface level, mostly reiterating what is available in the documentation without providing much more example usage. This may be for good reason as they are extremely powerful and potentially dangerous tools, but anyone dedicated enough will spend the time to learn the tool even without external assistance.

For a future project, a more solid demonstration of Active Directory enumeration and credential harvesting should be included. It is possible do to using C2 implants that arent realtime, like Covenant's Grunts, but knowing exactly how the tool works and planning its execution is more important because you are not directly running the tool yourself with a shell on the machine, you are tasking the Grunt to do it for you.

If a student is familiar with that and wants to have a more realistic progression of this project, I think that step is to do it in entirely in AWS, Digital Ocean, or another cloud provider, using information learned previously in the program. Setting up servers in the cloud is not much different from local VMs, and applying actual domain names to them is not difficult. Adding the C2 Redirector to this would make it a much more real-world demo than what I was able to come up with, as the traffic would now be running over the open internet. AWS and Digital Ocean do allow for penetration testing activities on its platform and against assets you own, so long as it does not affect other customers.

For a future research question, I am not sure what a future student may ask, but it may still be along the lines of "How do I efficiently maintain persistence in a network once Domain Admin or Root access is secured?". Perhaps they may focus on Operational Security and redundant modes of persistence both on single machines and through pivot points, rather than establishing a foothold and working through the network like this project was.

And finally: You're working with programs classed as malware. Turn off Windows Defender. It will save you many headaches. Windows Firewall too.

References

Art, S. (2017a). Pentest Home Lab - 0x2 - Building Your AD Lab on Premises. Retrieved March 13, 2022, from https://sethsec.blogspot.com/2017/06/pentest-home-lab-0x2-building-your-ad.html

Art, S. (2017b). Pentest Home Lab - 0x3 - Kerberoasting: Creating SPNs So You Can Roast Them. Retrieved March 13, 2022, from https://sethsec.blogspot.com/2017/08/pentest-home-lab-0x3-kerberoasting.html

Blackford, D., & Larson, S. (2021). Cobalt Strike: Favorite tool from APT to Crimeware. Retrieved January 10, 2022, from https://www.proofpoint.com/us/blog/threat-insight/cobalt-strike-favorite-tool-apt-crimeware

Cobb, R. (2019). Covenant Wiki. Retrieved January 10, 2022, from https://github.com/cobbr/Covenant/wiki

Conda. (2020). How to setup Covenant C2 with HTTP redirector. Retrieved January 10, 2022, from https://www.youtube.com/watch?v=1uh5-OzBEqM

Dimmock, J. (2017). Red Team Infrastructure Wiki. Retrieved January 15, 2022, from https://github.com/bluscreenofjeff/Red-Team-Infrastructure-Wiki

Dimmock, J. (2018). HTTPS Payload and C2 Redirectors. Retrieved March 13, 2022, from https://bluescreenofjeff.com/2018-04-12-https-payload-and-c2-redirectors/

Fatrozdianko. (2019a). Creating an Active Directory Lab in AWS. Retrieved March 13, 2022, from https://fatrodzianko.com/2019/08/05/creating-an-active-directory-lab-in-aws/

Fatrozdianko. (2019b). Getting started with Covenant C2. Retrieved March 13, 2022, from https://fatrodzianko.com/2019/08/14/getting-started-with-covenant-c2/

Fatrozdianko. (2019c). Kerberoasting. Retrieved March 13, 2022, from https://fatrodzianko.com/2019/09/07/kerberoasting/

InfiniteLogins. (2020). Installing Covenant C2 on windows and reviewing basic features. Retrieved January 10, 2022, from https://www.youtube.com/watch?v=3-l_UZfqLZA

Kelly, J. (2022). Interview with Jareth Kelly (Senior Managing Consultant). Informal interview with Jareth Kelly via Discord chat platofirm. Transcript available on upon request. Sarah Yeoell January 26, 2022.

Mudge, R. (2013). Tradecraft (Playlist). Retrieved January 10, 2022, from https://www.youtube.com/playlist?list=PL9HO6M_MU2nesxSmhJjEvwLhUoHPHmXvz

PowerShellMafia. (2015). Powersploit. Retrieved March 13, 2022, from https://github.com/PowerShellMafia/PowerSploit

Shcroeder, W. (2015). Find-LocalAdminAccess. Retrieved March 13, 2022, from https://powersploit.readthedocs.io/en/latest/Recon/Find-LocalAdminAccess/

Snaplabs. (2021). Covenant C2 for Red Teaming. Retrieved March 13, 2022, from https://www.snaplabs.io/insights/covenant-c2-for-red-teaming-2

TevoraThreat. (2018). SharpView. Retrieved March 13, 2022, from https://github.com/tevora-threat/SharpView