**Attack In-Depth! - VDD - Accompanying Speech**

**Opening**

* Hi there, hope everyone is going well.
* My name is Riley Peters and welcome to Attack in depth!
* Today, I will be breaking down the steps and processes taken whilst obtaining unauthorised workstation access during a mock Red Team Engagement, as part of the Final Project for the Cybersecurity Bootcamp.

**Contents**

* The presentation will cover the network topology and critical vulnerabilities and weaknesses found on a target machine,
* As well as the exploits carried out on said machine,
* And a quick look into avoiding detection and maintaining unauthorised access in the future.

**Network Topology**

* Here we have a simple diagram to outline the network topology that was provisioned for this project.
* As you can see, we have a total of five machines running on a Hyper-V host.
* The ELK VM is a Monitoring stack capable of aggregating logs and visualising data through the use of Kibana and other services.
* However, in the interest of time, this presentation focuses on weaknesses found on the Target 1 machine alone.

**Critical Vulnerabilities**

* Here's a table outlining the most critical vulnerabilities that were found on the Target 1 Machine.
* Each weakness had its own direct impact, all of which ultimately led to the red team gaining root access on the Target 1 machine.
* So let me break down these direct impacts and show the operations that uncovered these gaps in security.

**Exploits Used**

**WPscan Enumeration**

* Wordpress is a widely used open source management system, with a large amount of plugins and tools available on top of the product.
* One popular tool used for testing the security configuration of wordpress servers is WPSCAN.
* The tool can perform a great range of enumeration tasks and scans to help security professionals gauge how vulnerable their server is.
* In this instance, it was clear after navigating to the website for the server that it was running Wordpress, so a WPSCAN was run to attempt to find any user accounts with this command on screen, ultimately identifying both user accounts for michael and steven.

**SSH Login**

* From here, the well known brute force tool Hydra was used to crack michaels password, however upon doing so using the command onscreen, it was apparent that the password could have actually been guessed.
* Employing very poor cyber security practice, Michael had set his password the same as his username, ultimately allowing the Red Team to gain access to the Target 1 machine through an SSH connection on his user account.
* It was now possible to access potentially sensitive directories, such as the /html folder for the server.

**SLIDE CHANGE**

* Using passwords with SSH is nowhere near as secure as standard forms of key authentication used today such as RSA 2048 and 4096.
* Here you can see even using a 12 letter password, which is only really just becoming the norm for digital accounts, is still only as strong as a 1024 bit RSA key, which are fairly outdated now.

**Plaintext Password Storage**

* After accessing the /html folder of the server while logged in as Michael, it was possible to navigate further into a Wordpress folder containing all configuration files for the software, including a file called WP-CONFIG.
* Inside this file, plaintext login information for the server's MySQL database was stored.
* On top of this, the account details were for the ROOT account, meaning once logged into the database, any action could be undertaken.

**SLIDE CHANGE**

* Once the Red Team gained ROOT access to the MySQL database, it was possible to use a tool called MY SQL DUMP, used for backups, to exfiltrate the contents of the database.
* Here, sensitive information was captured such as the password hashes for both users that were enumerated in the initial stages of this engagement.

**Weak Password Hashing**

* Now that the Red Team had attained a password hash for the other user account, Steven, it was time to crack this hash to find its corresponding plaintext password.
* This proved to be rather simple, as the hashing algorithm used to fingerprint Steven's password was MD5, an algorithm that has already been cryptographically broken and is by all standards insecure.
* Steven’s password was found to be PINK 84, as cracked by the tool John the Ripper, and once logged into the target system as Steven, it was possible to see what permissions he held on the machine.
* It was found Steven could use the whole python library without a password, and using a simple python shell exploit, it was possible to gain a root shell on the target machine, leading to sensitive data being found, in this case, a final flag for the activity.

**Avoiding Detection**

* This section details potential alerts or searches that may detect parts of the activity conducted by the Red Team, as well as ways to mitigate these alerts if they had been configured in Kibana.

**WPscan Enumeration**

* Assuming a Kibana dashboard alert is in place that is measuring the USER AGENT and DESTINATION IP of incoming traffic, the Red Team can instead use WPSCAN in stealth mode.
* This randomises the user agent, ensuring the scan can succeed without alerting the SO team.
* Below is an example of a stealth mode command.

**SSH Login**

* In the case there was an alert measuring traffic count on port 22 from unique IPs, and blocking said traffic if it exceeded a timed threshold, like the one shown onscreen, the Red Team could use functions built into Hydra to slow its brute force process down.
* These options let you set the number of tasks as well as the time to wait between these tasks, making brute force traffic much harder to detect, albeit at the cost of speed.

**SQL Dump**

* This is another action that could be detected by SOC members.
* Kibana plugins can be configured to include aggregation of MySQL slow logs, which would hold evidence of any My SQL DUMPS performed.
* Shown here is an example of how a logged My SQL DUMP event would appear.
* For the Red Team to get around such a precaution, they could simply copy and paste contents of the database after outputting tables to the command line. No need to dump.

**Maintaining Access**

* This section refers to methods taken to implement forms of persistence on the target 1 machine.
* In the interest of time, I will cover one, however there were two other forms of persistence acquired during the Final project.

**SSH Key Drop**

* The aim here was to generate a new public and private key pair while logged into the target machine as root, as shown in figure 1.
* From here, the public key contents were added to a file known as **authorized\_keys**
* This is an SSH file used to reference stored keys for authorised logins,
* by adding this new public key, the corresponding private key can be used to gain persistent access to the root account.
* Once the contents of the private key were copied to the Kali machine, it was just a matter of creating a key file and then using said file to SSH login into the target machine as root.
* as shown down here in figure 3. Root level persistence was gained.

**The End**

* And with that marks the end of this presentation.
* I hope you enjoyed this rather brief recap of the Red Team Operations undertaken as part of this Final Project,
* and I invite you to ask any questions you may have!
* This presentation was a sort of high level overview of aspects of the project and there was much that I wish I could have included as well, so please, ask away!
* Thank you again for your time.