“Game of Pwns” CTF Pen Test Report Sheet

By

Owen Carver, Matt Henderson, Ian Kane, and Daniel Feibush

- **Project Summary of the capstone project:** “Our intention for this project is to showcase the core concepts of Penetration-Testing in Cybersecurity. Using the ‘Game of Thrones’ CTF, we will cover the vulnerabilities specific to this challenge and offer solutions for how one would resolve these vulnerabilities.”
- **Context for the CTF:** “‘Game of Thrones CTF: 1’ is a Capture the Flag challenge designed by Oscar Alfonso, known on Github as ‘OscarAkaElvis.’ The challenge consists of 11 total objectives: seven main flags, three secret flags, and one final flag that involves gaining root access. Each objective requires different means of access.”
- **Scope:** “For this challenge, our main goal is the discovery of confidential information. It is important that we keep in mind how the vulnerabilities that we use to gain access could be exploited by a bad actor. At the same time, we need to be cautious with the data ourselves, making sure that the data is intact. Our focus for the engagement is exfiltration of information.

  The digital scope was the Game of Thrones CTF hackable box, also referred to as ‘7kingdoms,’ hosted on the address 192.168.57.1/24. It should be noted that the IP address may vary depending on the network where it is downloaded. Apart from the description on Vulnhub, we aimed to work through the challenge without using any outside knowledge unless we encountered unfamiliar concepts.”
- **Assumptions:** “The CTF’s Vulnhub description gave us an impression of the engagement’s difficulty, ranging from Medium to High. The assumption was that the challenge would mainly come from any unfamiliar tools and concepts required for completion.”
- **Methodology:** “The information gathered from the Vulnhub description disclosed that a set order for the challenges could lead to the completion of the challenge. Using the nmap and dirbuster tools, we confirmed that the target system was live and what directories were immediately accessible to us without first obtaining credentials. Our initial scans were important toward the recon phase since it allowed us to discover open ports for accessing the system and a hosted webpage named ‘/secret-island/’ that provided a guide for completing each task. After gaining more directions as to where we should begin our search, the main tools that we used for the engagement were the Command Line Terminal and a web browser, specifically Burp Suite since the proxy can intercept information from request forms. Knowing that the CTF was live, we used directions from the map and context clues hidden in its files to solve each objective.

  LinPEAS became an important tool for discovering root access.”
- **Results:** At the end of each objective, we received a flag hashed in MD5 format. Unhashing each flag would reveal one word out of 11 that would form a message from the creator upon the challenge’s completion. Metasploit did not work as effectively as

intended, but other tools were useful in our recon phase. From a vulnerability standpoint, these are the results from each objective (using the MITRE ATT&CK Matrix):

- -Collecting Data from Information Repositories
- -Exploiting Public-Facing Applications
- -Credentials Access in Files
- -Command Line Execution
- -Account Discovery
- -File and Directory Discovery
- -Collecting Data from Local System
- -Exfiltration Over Web Service
- -Account Manipulation
- -Privilege Escalation

1. FTP: **Vulnerable initial access**. Since port 21 was open and running this service, **public-facing applications** were exploitable. This provides an attack vector that is initiated from visiting the target’s IP address in a web browser and looking for directories to gain access. Attempting anonymous access is not possible on the target and a fail2ban feature locks out users for 2 minutes, making a bad actor’s password attack easily detectable. Additional Note: **Collecting Data from Information Repositories** was the technique used to pull information from the FTP files.
2. HTTP: Measures are taken to obscure information such as login credentials (mainly hiding clues within the website’s source code and using features like the User-agent), but they are also written in plain text, which could raise the risk factor in an attack scenario. The “/h/i/d/d/e/n/index.php" file also discloses that Docker is used to increase the site’s security. Available clues taken from the FTP server revealed use of a hash formatted as “md5(md5($s).$p)” and an mcrypt-encrypted file. Using specific tools for each problem, Hashcat and Mcrypt, we were ableto gather new information to continue the engagement. Doing so led to a webpage where the HTML hid the flag.
3. DNS: Another instance of an **exploitable public-facing application**. Using the “dig” command in addition to a web domain located in the HTML from the previous challenge. This eventually directed us to the next flag, which was embedded in the command’s output. The flag also provides **credential access** to another website disclosed within the output.
4. Webmin: A security parameter is set in place for the “Stormlands” website where this flag is located. Accessing this page at all, even with the credentials revealed by the previous objective’s output, cannot be achieved without adding the necessary web address to your machine’s “etc/hosts/”

file. After reaching the website, a Webmin vulnerability from Metasploit can be used to leverage the “/file/show.cgi” folder so that the flag is revealed if signed in as “aryastark.” Through a **Command-Line Execution** vulnerability within this web application, vital information is made vulnerable.

5. PostgreSQL: The objective was achieved using credentials achieved from the previous challenge, directing us to a database containing multiple tables. The message was encoded in ROT16 format, which can be easily decoded using a tool like CyberChef. Most likely, this vulnerability could fall under the categories of **Account Discovery** and **File Discovery**.
6. IMAP: Achieving the objective required using the IMAP protocol. At first, this port was closed but we were able to a knock command on the IP address with a unique passcode to open it. Initiating an IMAP connection can be done using different commands like netcat or telnet. With the right command input, we retrieved information from the only message in the inbox for “oleannatyrell.” The solution came as a result of **Credential Access in Files** and **File Discovery**.
7. Gitlist, MySQL: Using Credential Access from the previous achievement’s output, we were able to visit a Gitlist website. The flag’s location could not be discovered by searching the website normally, but we instead used URL Encoding to find where the location of the next hint. Doing so led us to a “checkpoint.txt” file. This seems to be an example of **Exploiting a Public-Facing Application** (the website’s search query parameters should most likely be sanitized). The MySQL portion was difficult since Command Line Execution gave no output, but we leveraged the same Gitlist vulnerability to uncover a clue written in Morse Code. Further Exploitation was achieved by using the vulnerability to view the file located at “/etc/mysql/flag.”
8. Secret 1: AKA the “Savages” flag. This puzzle was completed by examining the metadata from the challenge’s homepage further. Within a music folder containing an MP3 file in the sources, After copying the MP3 file’s contents into a new file and using the strings command, the flag is revealed. This appears to be an instance of **File Discovery** in the MITRE ATT&CK Framework.
9. Secret 2: AKA the “Braavos” flag. Similar factors to the PostgreSQL objective (located within the same database). Falls under the same category as **Account** and **File Discovery**.

10. Secret 3: AKA the “Dragonglass Mine” flag. Achieved using Credentials from File Access, entering an SSH session as user “daenerystargaryen.” Using **Command Line Execution**, we were able to Escalate Privileges using the Hydra command to determine the password for Root access. Logging into the Root user account will allow you to find the
11. SSH: AKA the “Final Battle” flag. Using Credential Access as “branstark,” we are able to **Escalate Privileges** using a Docker command to become Root. The final flag is held within a password-protected Zip file, but this is unlocked by first **Collecting Data from the Local System** and then transferring it back to our client-side machines. Then, forming the password based on context clues we are able to unlock the Zip file and obtain the final flag.

- **Recommendations: (1)** In general, better encryption for credentials will be useful for ensuring unwanted parties cannot easily access systems. It may be better to use more secure versions of running services to ensure that unintended parties can’t access confidential information. Changing FTP to SFTP for file transfers could improve the owner’s ability to keep information confidential. **(2)** Updating applications with the latest security patches would be essential for keeping intruders away. Public-facing applications such as Webmin and Gitlist can be exploited to access files in unintended ways, which can become an issue if left unchecked. This is especially important as most of the applications are outdated versions of themselves. **(3)** Practicing least-privilege will be essential to making sure that attackers cannot escalate privileges from any one account. Implement role-based access control for each account can be useful to make sure that they cannot use Root access easily.