TakeOver Challenge
Write-Up

Rinshad

Room Name	TakeOver
Platform	TryHackMe
Target IP	10.10.195.112
Domain	futurevera.thm
Room URL	https://tryhackme.com/room/takeover

Introduction

The "TakeOver" room on TryHackMe simulates a real-world security scenario involving a potential subdomain takeover. A fictional company, FutureVera, has reportedly been approached by blackhat hackers who claim they have found a way to hijack one of the organization's services. The company is seeking help to understand the vulnerability and assess the threat.The objective of this challenge is to investigate the website hosted at futurevera.thm, identify any vulnerable subdomains, and uncover any leaked or hidden flags that indicate a potential subdomain takeover scenario.

1. Environment Setup

Before beginning the assessment, I added the target domain to my local /etc/hosts file to resolve it correctly in a browser:

echo "10.10.195.112 futurevera.thm" | sudo tee -a /etc/hosts

This allows **futurevera.thm** and any subdomains to resolve to the target IP.

2. Enumeration

Nmap Scan

I began by conducting a basic Nmap scan to identify open ports and running services on the target:

nmap -sC 10.10.195.112

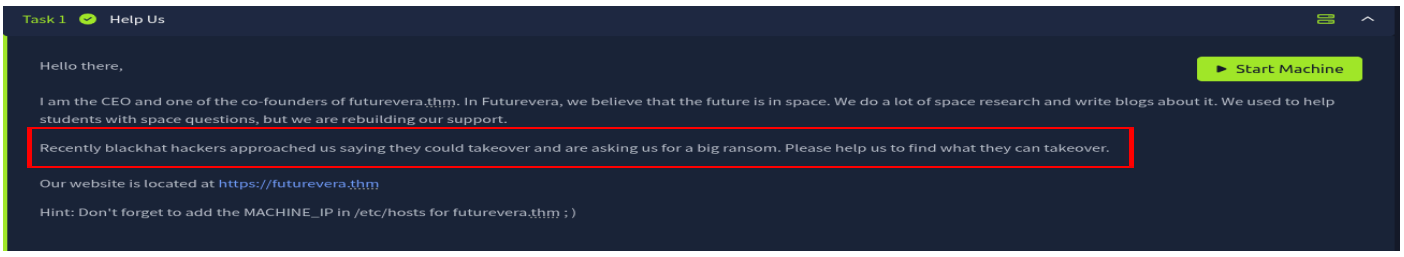
```
┌──(kid㉿dr4g0n)-[~/Desktop/Labs/THM/TakeOver]
└─$ cat nmap_result
# Nmap 7.95 scan initiated Wed Jul  9 23:23:04 2025 as: /usr/lib/nmap/nmap --privileged -sV -oN nmap_result 10.10.195.112
Nmap scan report for futurevera.thm (10.10.195.112)
Host is up (0.19s latency).
Not shown: 997 closed tcp ports (reset)
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.13 (Ubuntu Linux; protocol 2.0)
80/tcp  open  http     Apache httpd 2.4.41 ((Ubuntu))
443/tcp open  ssl/http Apache httpd 2.4.41 ((Ubuntu))
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Wed Jul  9 23:30:07 2025 -- 1 IP address (1 host up) scanned in 423.00 seconds
```

The scan detected three open TCP ports: 22 (SSH), 80 (HTTP), and 443 (HTTPS). Port 22 is running OpenSSH 8.2p1 on Ubuntu, while ports 80 and 443 are both serving Apache httpd version 2.4.41, also on Ubuntu. The operating system was identified as Linux. Service version detection was enabled during the scan, providing detailed information about the services and their configurations. This information is useful for assessing the attack surface and identifying potential vulnerabilities.

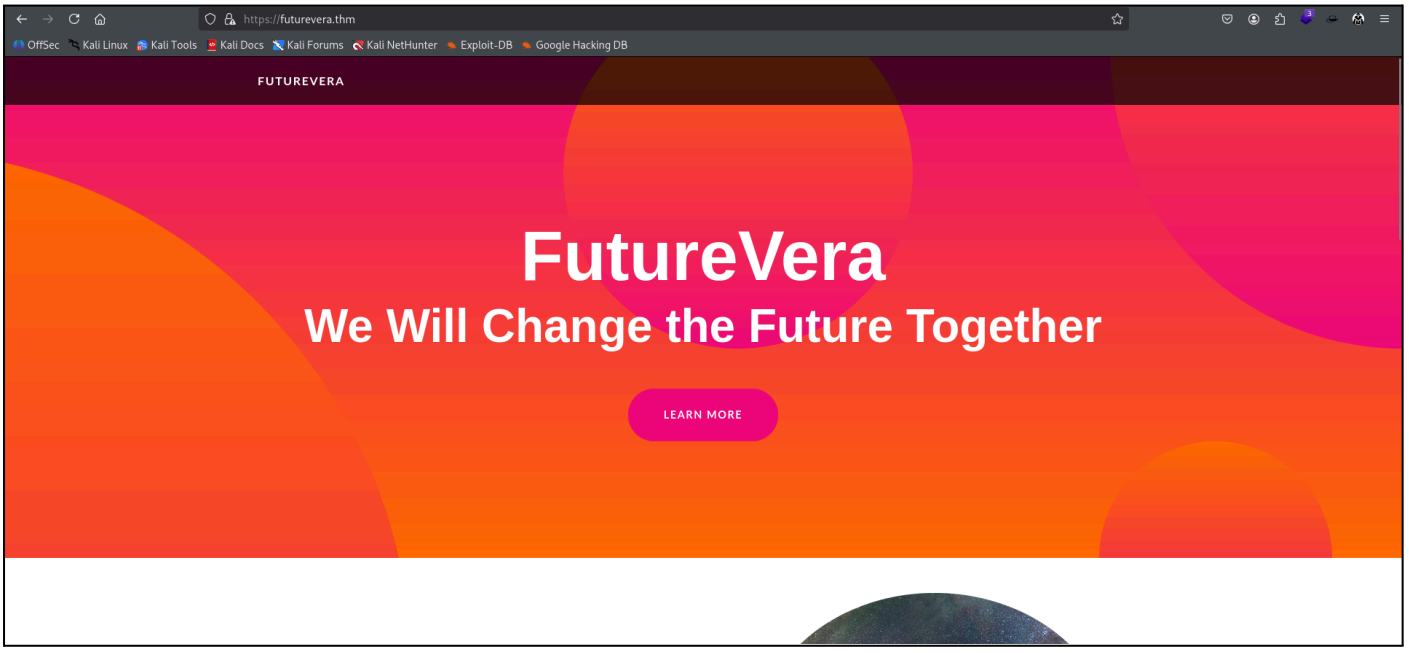
Web and DNS Enumeration

Initial Observations

Accessing **`https://futurevera.thm`** in the browser led to a corporate-style website. From the task description and the website's content, I inferred that the company's support system was in the process of being rebuilt, which became a key hint for subdomain exploration:



This suggested that there may be support-related subdomains still present or misconfigured.



Subdomain Enumeration

To investigate further, I performed DNS enumeration using **Gobuster** with the **`subdomains-top1million-110000.txt`** wordlist from **SecLists**. The command used was:

gobuster dns -d futurevera.thm -w /seclists/Discovery/DNS/subdomains-top1million-110000.txt

```
┌──(kid㉿dr4g0n)-[~/Desktop/Labs/THM/TakeOver]
└─$ gobuster dns -d futurevera.thm -w /usr/share/wordlists/seclists/Discovery/DNS/subdomains-top1million-110000.txt

===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Domain:     futurevera.thm
[+] Threads:    10
[+] Timeout:    1s
[+] Wordlist:   /usr/share/wordlists/seclists/Discovery/DNS/subdomains-top1million-110000.txt
===============================================================
Starting gobuster in DNS enumeration mode
===============================================================
Found: support.futurevera.thm

Found: piwik.futurevera.thm

Found: adserver1.futurevera.thm

Progress: 114442 / 114443 (100.00%)
===============================================================
Finished
===============================================================
```

The scan successfully identified three valid subdomains:

- `support.futurevera.thm`
- `piwik.futurevera.thm`
- `adserver1.futurevera.thm`

After identifying the subdomains through Gobuster, I updated my **`/etc/hosts`** file to map the discovered domains to the target IP address (**`10.10.195.112`**). This step was necessary to resolve the subdomains locally and access them via a browser or other tools. The entries added included **`support.futurevera.thm`**, **`piwik.futurevera.thm`**, and

adserver1.futurevera.thm. This setup ensured proper DNS resolution for further

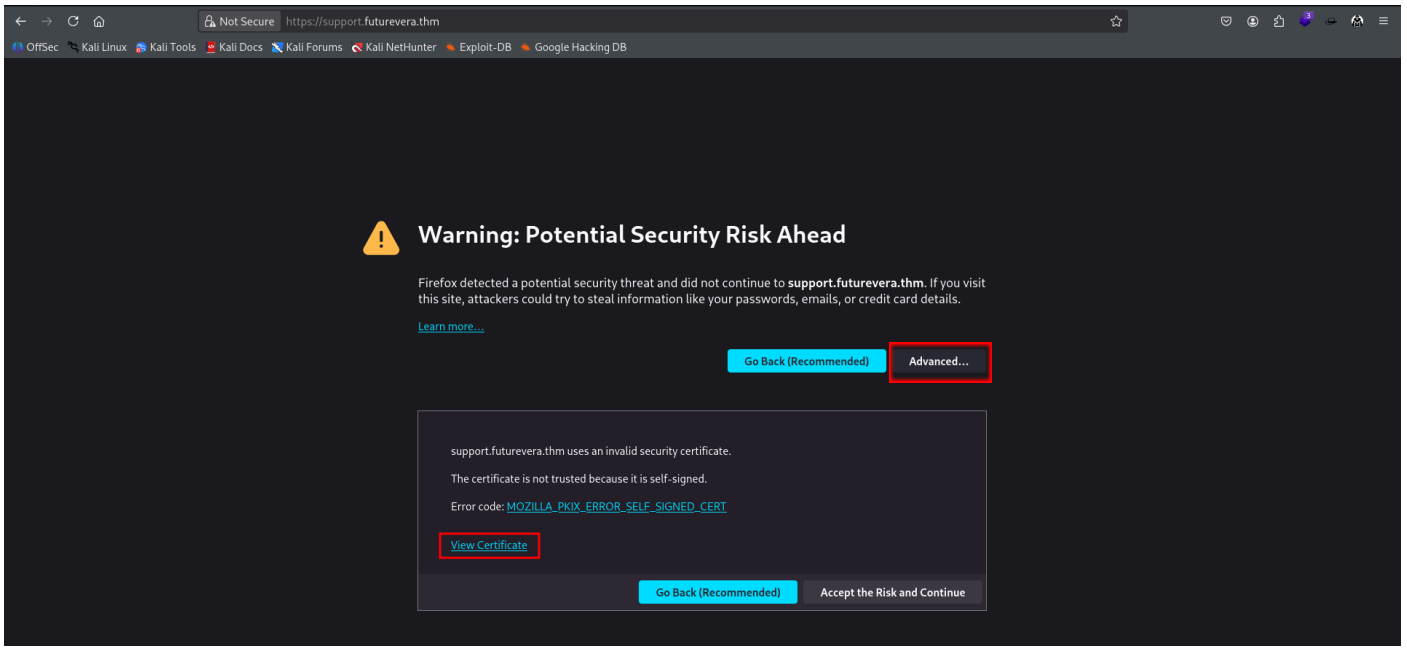
```
┌──(kid㉿dr4g0n)-[~/Desktop/Labs/THM/TakeOver]
└─$ echo "10.10.195.112 support.futurevera.thm" | sudo tee -a /etc/hosts
10.10.195.112 support.futurevera.thm

┌──(kid㉿dr4g0n)-[~/Desktop/Labs/THM/TakeOver]
└─$ echo "10.10.195.112 piwik.futurevera.thm" | sudo tee -a /etc/hosts
10.10.195.112 piwik.futurevera.thm

┌──(kid㉿dr4g0n)-[~/Desktop/Labs/THM/TakeOver]
└─$ echo "10.10.195.122 adserver1.futurevera.thm" | sudo tee -a /etc/hosts
10.10.195.122 adserver1.futurevera.thm
```

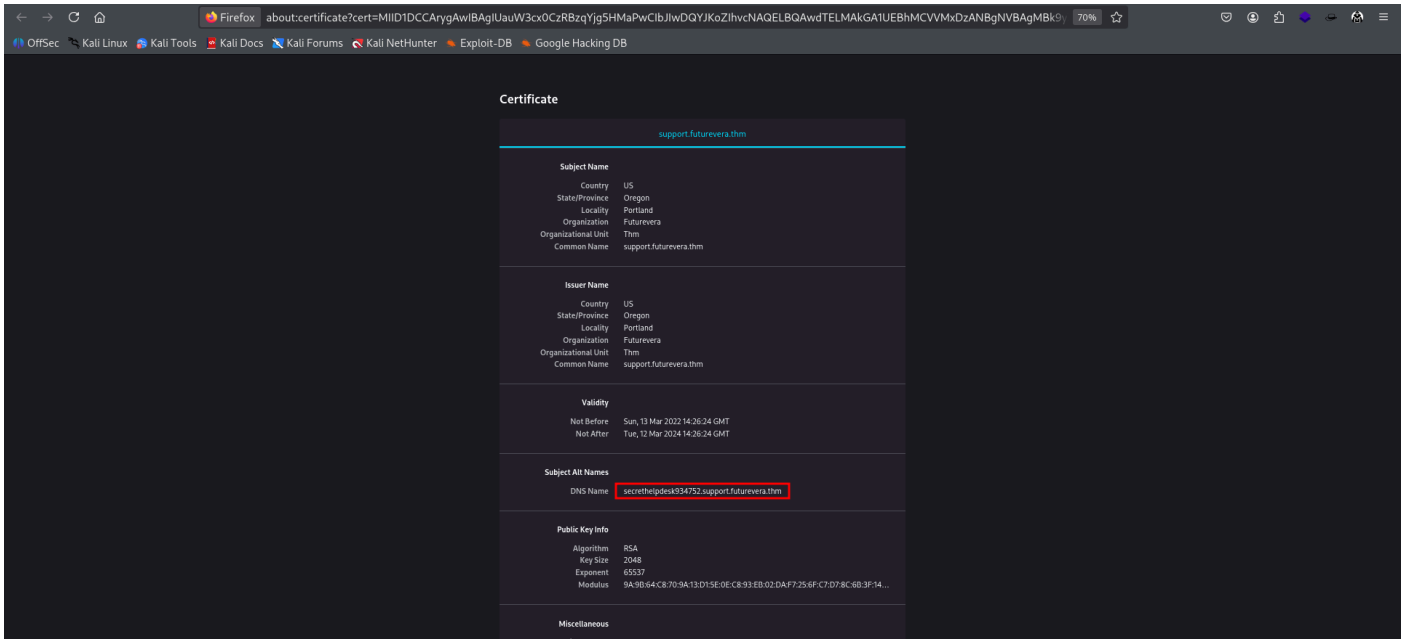
analysis of each subdomain.

Only **support.futurevera.thm** was valid and responsive. The other entries (`piwik.futurevera.thm` and `adserver1.futurevera.thm`) were false positives and did not return any meaningful content.

SSL Certificate Inspection

Upon visiting **https://support.futurevera.thm**, I encountered a certificate warning. Rather than bypassing it, I examined the SSL certificate through the browser's "View Certificate" option. In the "Subject Alternative Name (SAN)" section, I found a hidden subdomain: **secrethelpdesk934752.support.futurevera.thm.** This subdomain was not detected by Gobuster and was only visible in the SSL metadata. This is a classic case of information leakage through misconfigured SSL certificates.





I attempted to access the hidden subdomain:

http://secrethelpdesk934752.support.futurevera.thm

The page failed to load, but the browser displayed a **DNS resolution error**. Interestingly, the error message included a reference to a URL:

flag{beea0d6edfcee06a59b83fb50ae81b2f}.s3-website-us-west-3.amazonaws.com

This string format is typical of AWS S3-hosted websites. The inclusion of a flag value in the subdomain strongly indicated that this was the intended goal of the challenge.

Flag : `flag{beea0d6edfcee06a59b83fb50ae81b2f}`