System Security Project Step 2

Group: The Group
Members: Andrew Bracken, Gavin Holliday, Logan Lay
CS 4371: Computer Security
Dr. Randy Klepetko

October 29, 2024

1 Introduction

In this project, we will test the security of our system by going on the offense and sending it attacks. By seeing how our system could be breached, it will help us effectively create defensive measures in the future.

2 Task I: Echo Program Vulnerability

2.1

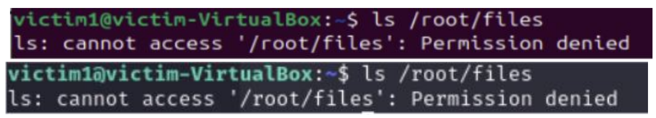

Figure 1: Root File Access

2.2 Seven bytes are needed to crash the echo program.

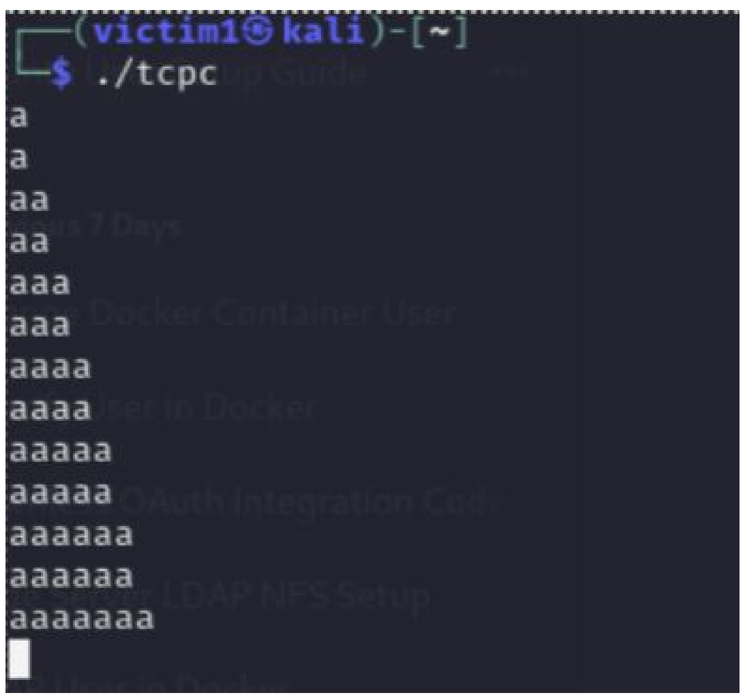

Figure 2: Echo Crash

2.3

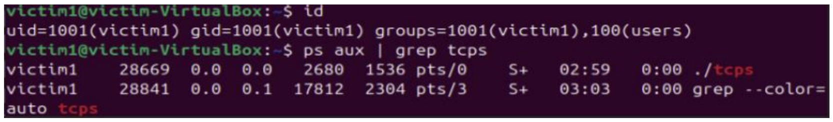

Figure 3: User Running Echo Program

2.4

Figure 4: User Running SSH Service in A.1

3 Task II: Analyze the Echo Program

3.1

Figure 5: Breakpoint on foo()

3.2

```
(gdb) info stack
#0  foo (in=0×7fffffffdc30 "a\n") at tcph.c:23
#1  0×0000000000401199 in main () at tcph.c:14
(gdb) 
```

Figure 6: Stack on foo()

3.3

```
(gdb) info registers rsp rbp
rsp            0×7fffffffdc00      0×7fffffffdc00
rbp            0×7fffffffdc20      0×7fffffffdc20
(gdb) print &buf
$1 = (char (*)[8]) 0×7fffffffdc18
(gdb) x/gx $rbp+8
0×7fffffffdc28: 0×0000000000401199
(gdb) 
```

Figure 7: Report Values, Address of buf, and foo() in B.1

3.4

```
(gdb) info registers rsp rbp
rsp            0x7fffffffdbb0      0x7fffffffdbb0
rbp            0x7fffffffdbd0      0x7fffffffdbd0
(gdb) print &buf
$1 = (char (*)[8]) 0x7fffffffdbc8
(gdb) x/gx $rbp+8
0x7fffffffdbd8: 0x00000000004011dd
(gdb) 
```

Figure 8: Report Values, Address of buf, and foo() in A.1

4 Task III: Exploit the Target Programs

4.1

4.2

4.3

4.4

4.5

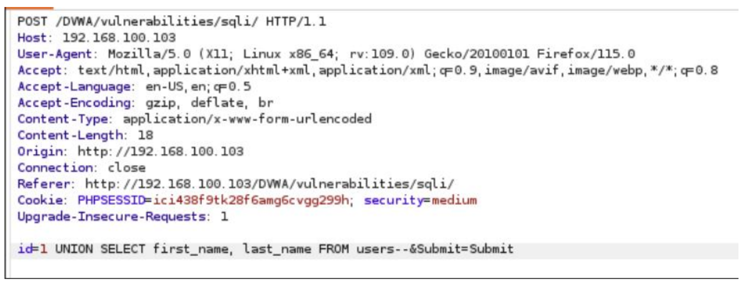

Figure 9: Injected SQL Statement

4.6

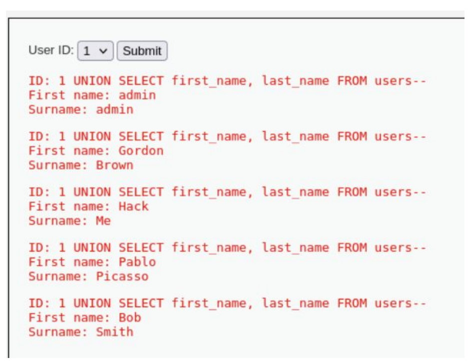

Figure 10: User IDs

5 Buffer Overflow and Address Randomization

5.1 Explanation

Because computer architectures follow common design principles, it is easy to rewrite the return address with a malicious one when attempting a buffer overflow attack. By randomizing the addresses, it will be more difficult to guess where the return address is stored in memory.

5.2 Calculation

Since only the low 16 bits are randomized, this randomizes 2^{16} addresses, equaling 65,536 addresses. The probability of hitting the return address is $1/65,536$. Given the attacker sends 10 packets a second, it will take:

$$\frac{65,536}{10} = 6,553.6 \text{ seconds} \approx 1 \text{ hour and } 49 \text{ minutes.}$$

6 Task IV: Dictionary Password Cracking

6.1

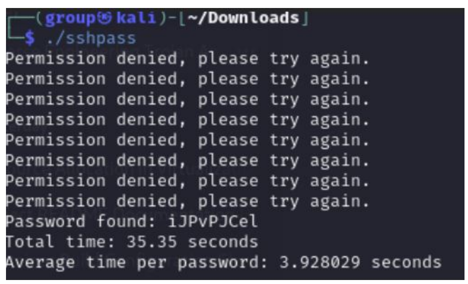

Figure 11: Password Testing

With an average of 3.928 seconds per password, 1 million passwords would take:

$$3,928,000 \text{ seconds} \approx 45 \text{ days, if the correct password was last in the file.}$$

7 Task V: Dictionary Cracking, Professional Tool

7.1

```
msf6 auxiliary(scanner/ssh/ssh_login) > info

       Name: SSH Login Check Scanner
     Module: auxiliary/scanner/ssh/ssh_login
    License: Metasploit Framework License (BSD)
       Rank: Normal

Provided by:
  todb <todb@metasploit.com>

Check supported:
  No

Basic options:
  Name              Current Setting  Required  Description
  ----              ---------------  --------  -----------
  ANONYMOUS_LOGIN   false            yes       Attempt to login with a blan
                                               k username and password
  BLANK_PASSWORDS   false            no        Try blank passwords for all
                                               users
  BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from
                                                0 to 5
  CreateSession     true             no        Create a new session for eve
                                               ry successful login
  DB_ALL_CREDS      false            no        Try each user/password coupl
                                               e stored in the current data
                                               base
  DB_ALL_PASS       false            no        Add all passwords in the cur
                                               rent database to the list
  DB_ALL_USERS      false            no        Add all users in the current
                                                database to the list
  DB_SKIP_EXISTING  none             no        Skip existing credentials st
                                               ored in the current database
                                                (Accepted: none, user, user
                                               &realm)
  PASSWORD                           no        A specific password to authe
                                               nticate with
  PASS_FILE         dictionary.txt   no        File containing passwords, o
                                               ne per line
  RHOSTS            99.68.230.147    yes       The target host(s), see http
                                               s://docs.metasploit.com/docs
                                               /using-metasploit/basics/usi
                                               ng-metasploit.html
  RPORT             22               yes       The target port
  STOP_ON_SUCCESS   true             yes       Stop guessing when a credent
                                               ial works for a host
  THREADS           1                yes       The number of concurrent thr
                                               eads (max one per host)
  USERNAME          user50           no        A specific username to authe
                                               nticate as
  USERPASS_FILE                      no        File containing users and pa
                                               sswords separated by space,
                                               one pair per line
  USER_AS_PASS      false            no        Try the username as the pass
                                               word for all users
  USER_FILE                          no        File containing usernames, o
                                               ne per line
  VERBOSE           true             yes       Whether to print output for
                                               all attempts

Description:
  This module will test ssh logins on a range of machines and
  report successful logins.  If you have loaded a database plugin
  and connected to a database this module will record successful
  logins and hosts so you can track your access.

References:
  https://nvd.nist.gov/vuln/detail/CVE-1999-0502

View the full module info with the info -d command.
```

Figure 12: SSH Login Module Parameters, klepetko.net

7.2

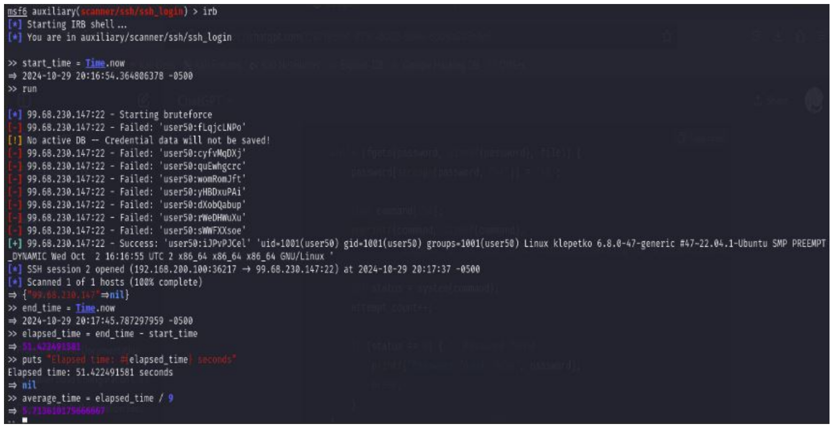

Figure 13: Correct Password and Average Test Time

7.3

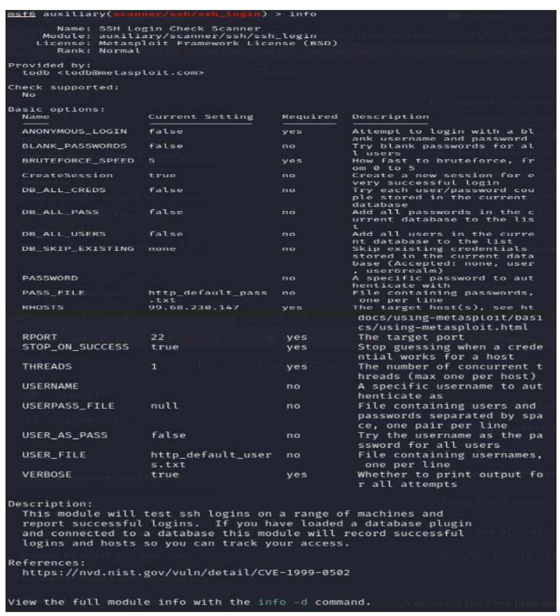

Figure 14: SSH Login Module Parameters, Username Dictionary

7.4

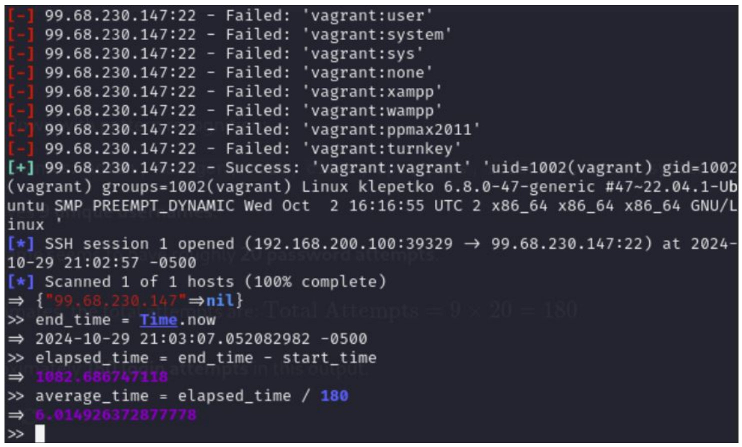

Figure 15: Correct Password and Average Test Time

8

9

9.1

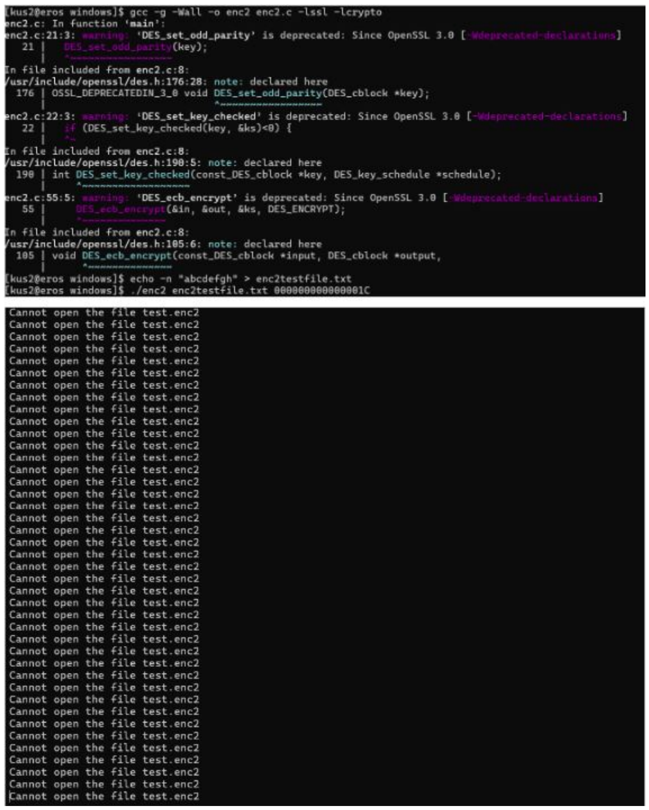

Figure 16: DES Program

10 Conclusion

In this project, we explored multiple ways of breaching a system's security methods. The first approach was using an attack file from the Kali machine to exploit the echo program of the Ubuntu machine. We then discussed randomization and how it can be an effective counter to an attack. After that we created a dictionary program to test multiple passwords in order to SSH as user50 to kleptko.net and then did it with a professional tool. We then used our dictionary tool to find another username and password for kleptko.net.