Summative Security Coursework

Vulnerability	Exploit / Problem	Mitigation
1) Users have weak passwords	All user passwords are short and either don't use numbers or special characters or are easily guessable e.g. root password is 12345678.	User passwords should be a random selection of characters, numbers, and symbols. And passwords should be of suitable length such that they cannot easily be bruteforced.
2) Users have elevated permissions	Users e.g. user have read write permission on certain other users' home folders such as alice's. So user could access alice's private data such as confidential.txt	Users should not have read write or execute permissions on other users home directories so these permissions should be removed using chmod.
3) /etc/passwd is global readable and contains unsalted password hashes	Any user on the system can get access to hashed user passwords and pass them to a program like John The Ripper and get every users plaintext password, including root.	Passwords should be stored in a shadow file that is only readable by root.
4) Jess's selfie has been encrypted with aes-256-ecb	Jess's selfie has been encrypted with ECB mode enabled therefore key details of the image are still visible after encryption.	Re-encrypt the selfie using aes-256 without ECB enabled.
5) Bitcoin program has buffer overflow	The bitcoin program has a fixed input buffer on the stack that can be written over as gets takes an input of any length. So, by typing too long a password anyone can access the bitcoin wallet.	Use snprintf instead of gets as snprintf allows for limiting the size of the input to the buffer size. Could also compile with buffer overflow protection enabled.
6) Backup program allows for any command to be run as root	Anyone running the database backup program can perform any command as if they were root by passing it as input to the backup program surrounded by semicolons. As this passes it into the system call as a separate command to the cp command.	Instead of using a system call the program should instead take the database filename as input and make sure it exists. Then setuid to 0 and read the contents of the file into a buffer and write it into a new file at the backup location as this avoids the system call.
7) Users database table stores passwords in plain text	The Users table stores user passwords in plaintext. So, in the event of a database leak the adversary will have all the user passwords. A malicious	Store a hash of salted passwords in the database instead. Preferably using a 64 bit salt.

	employee could also steal them.	
8) Users have access to real database information	Users have access to the actual database information on their desktops. This means a malicious user could steal people credit card details and passwords.	Users should only have access to a masked database containing dummy information as it isn't necessary for every user on the webserver to have full access to the database information.
9) Website login page vulnerable to SQL injection	The password input on the login page can be used for SQL injection e.g. ' OR 1=1;-- allows you to see all user passwords and credit card numbers.	Use parameterised queries instead of just pasting the password input into the query.
10) Website chat is vulnerable to cross site scripting	It is possible to write html script tags containing malicious Javascript code in chat messages. This allows a malicious user to steal cookies or redirect people to other dangerous websites.	The messages should either be filtered, and script tags removed before posting or by encoding special characters.
11) Website uses http not https	Packets are not encrypted therefore if someone used a packet sniffer they could intercept and read packets containing password inputs.	The webserver should be reconfigured to use https instead of http.
12) Webserver has an open TCP connection to port 8888	Anyone can connect to the TCP socket and have full access to the root shell running on it. Allowing them to do anything on the server.	Close the TCP socket on the server. Or replace it with an SSH server instead to people have to login or have an SHH key to use it.
13) Website password field shows password in plaintext	The websites password field shows plaintext passwords so if someone is standing behind a user while they enter their password then they can steal it.	Change the login page to use an html form with the password field set to 'type password'.
14) GRUB Bootloader has no password	The GRUB bootloader has no password so before the webserver boots up it is possible to adjust the boot options to include init=/bin/sh to boot straight into a root shell.	Have a secure password in place to prevent people from editing the bootloader settings.
15) Database is vulnerable to inference attacks	It is possible to determine in what countries website users live in as on the stats page the list users and view user countries lists aren't sorted, so the entries matchup between them.	Sort the users and countries into some order (could be randomised) so it is not obvious which user corresponds to which country.

16) Webserver vulnerable to path traversal	By html encoding / as %2f and . as %2e it is possible to navigate the servers entire filesystem from a web browser and circumnavigate all user access permissions in the process.	The paths should first be verified by the webserver before serving the content at that path to the web browser.

Murder Mystery:

Jess (real name Charlotte Rebels) is the “Bleeder”. She has killed Karl after his investigations led to him discovering her real identity and role as the “Bleeder”. The police should use this YouTube video of Karl’s murder as evidence https://www.youtube.com/watch?v=iMfnXUYi5lo&feature=youtu.be. They should also look at her confidential.txt file as it contains a full confession and motive for her murders. And also Mark should come out as being Tux since it Alice has feelings for Tux but doesn’t know its Mark.