Assignment on Cybersecurity II for CS50L

Total points 10/10

Starting in 2021, all assignments in CS50L are out of 10 points. A score of 7 points or better (70%) is required to be considered to have "passed" an assignment in this course. Please do not resubmit an assignment if you have already obtained a passing score. You don't receive a final grade at the end of the course, so it will have no bearing on your certificate, and it will only slow down our graders!

Unlike CS50x, assignments in this course are graded on a set schedule, and depending on when you submitted, it may take up to three weeks for your work to be graded. Do be patient! Project scores and assignment status on cs50.me/cs50l (e.g. "Your submission has been received...") will likely change over time and are not final until the scores have been released.

Email *

maz786@outlook.com

Name *

Mazafer Ul-Raqib

edX Username *

Mazafer

What is your GitHub username?

You only need to tell us if you are concerned about checking your progress in the course and/or you want a free CS50 Certificate after you satisfy all of the requirements of the course. If you do not already have a GitHub account, you can sign up for one at https://github.com/join. You can then use this account to log in to cs50.me/cs50l to track your progress in the course (your progress will only show up after you have received at least one score release email from CS50 Bot, so do be patient!). Don't worry about seeing a 'No Submissions' message on submit.cs50.io, if you find that. The course collects submissions using Google Forms, and only the gradebook on cs50.me/cs50l is important! If you do decide to provide us with a GitHub username, BE CERTAIN IT IS CORRECT. If you provide the wrong username, you will not be able to see your scores.

https://github.com/maz786

This course is graded by human graders, and has a ZERO TOLERANCE plagiarism and collaboration policy. If *any* of your answers are copied and pasted from, or obviously based on (a) an online source or (b) another student's work in the course, in *any* of the course's ten assignments, you will be reported to edX and removed from the course immediately. There is no opportunity for appeal. There are no warnings or second chances. *

It is far better, we assure you, to leave an answer blank rather than risk it. This may be an online course, but it is offered by Harvard, and we're going to hold you to that standard.

◉ I understand this policy and agree to its terms; I hereby affirm that I will not plagiarize any answers or collaborate with any other students in this course .

✓ Why is it important to implement proper security measures when using GitHub as a source code repository? 1/1

A well-liked website for hosting and working together on software projects is GitHub. It's crucial to put in place suitable security measures to safeguard sensitive data on any platform from unwanted access or manipulation.

When using GitHub as a source code repository, putting security measures in place is crucial for a number of reasons:

The confidentiality of your source code can be protected with the right security measures. It's crucial to make sure that no unauthorised people can access your code if it contains sensitive data or trade secrets.

Integrity: It's crucial to guarantee the integrity of your source code to make sure that it hasn't been modified or interfered with in any manner. This is particularly crucial if several individuals are working together on a project because any modifications made by one person could potentially cause problems for others.

Access management: You may manage who has access to your source code repository by putting security measures in place. This is necessary to protect the privacy and integrity of your code, as well as to guarantee that only people with the proper authorization can modify the source.

Compliance: Depending on the specifics of your project, you might need to follow specific security laws or guidelines. You can comply with these criteria and stay clear of any potential legal or regulatory difficulties by putting in place appropriate security measures.

In conclusion, putting in place appropriate security measures is crucial for safeguarding the privacy, consistency, and accessibility of your source code on GitHub. Additionally, observing any applicable security standards or legislation is crucial.

✓ In your own words, how does an SSH key provide greater security than simply using a username/password combination? 1/1

A form of authentication technique used to protect connections to remote servers is the SSH (Secure Shell) key. It uses a pair of distinct identifying keys to verify a connection, adding an extra layer of protection over a username/password combination.

An SSH key offers more security than a username/password combination for a number of reasons, including:

Passwords are less safe than keys: SSH keys authenticate connections using a public/private key pair. While the public key is retained on the server, the private key is kept on the client computer. This makes the private key more secure than a basic password since it makes it impossible for an attacker to remotely guess or brute-force the private key.

An SSH key is more secure than a password on its own, but you can further safeguard your key by adding a passphrase. Keys can be password-protected. This adds an extra layer of security since in order to use the private key, an attacker would also need to know the passphrase in addition to having the private key.

Keys are simpler to use: After setting up and adding an SSH key to the server, you can use it to authenticate connections without repeatedly entering your username and password. Because of this, using it becomes simpler and more convenient, especially if you connect to the server regularly.

In conclusion, because SSH keys use a public/private key pair to authenticate a connection, they may be password-protected, and they are simpler to use, they offer higher security than a username/password combination.

✓ Two-factor authentication is often lauded for providing "better" security than "regular" login. But what's a potential downside of using 2FA? 1/1

A user must give two distinct authentication factors in order to access their account while using two-factor authentication (2FA). Since it adds an additional layer of security against unwanted access, it is frequently praised for offering better security than a single-factor authentication procedure.

However, utilising 2FA could have drawbacks as well.

Added steps: The fact that 2FA adds an extra step to the login process and may be cumbersome for consumers is one potential drawback. This could be particularly troublesome if the second authentication factor is a code sent to the user's phone or email because it necessitates the user's physical possession of that device in order to log in.

Dependence on outside factors: If the user lacks access to the second authentication factor, such as a phone with a valid SIM card or an email account with functional internet connectivity, 2FA may be disrupted. The user may find it frustrating if this makes accessing their account difficult or impossible.

Potential weaknesses: Although 2FA adds an extra layer of protection, it is not perfect. An attacker might still be able to access a user's account, for instance, if they are able to intercept the second authentication factor (for instance, by getting access to the user's email or phone).

In conclusion, while two-factor authentication (2FA) can offer more security than single-factor authentication, it can also be more cumbersome for consumers and less reliable. Before utilising 2FA, it is crucial to thoroughly consider the advantages and potential drawbacks.

✓ Why, technically, are DoS attacks considered easier to stop than DDoS attacks? 1/1

Cyber attacks of the DoS (Denial of Service) and DDoS (Distributed Denial of Service) variety aim to prevent a network or service from operating normally. While both kinds of assaults have the potential to be disruptive, DDoS attacks are typically thought to be more challenging to neutralise than DoS attacks.

DoS attacks are thought to be simpler to counter than DDoS attacks for a number of technical reasons:

DoS assaults frequently originate from a single source, making it simpler to recognise and block the traffic. DDoS attacks, on the other hand, have numerous sources, which might make it more challenging to recognise and stop the attack traffic.

DoS assaults typically have a single source, making it simpler to identify the attack's origin and take precautions against it. DDoS assaults make it more challenging to identify the origin of the attack traffic because it originates from numerous sources.

Less traffic: Unlike DDoS attacks, DoS attacks often include less traffic. Due to the reduced amount of attack traffic, it might be simpler to filter or stop it.

DoS attacks are frequently more focused since they are typically initiated by a single entity with a particular objective in mind. On the other hand, DDoS assaults are frequently launched by numerous parties and may not have a clear aim or objective. This can make it more challenging to lessen the impact of a DDoS attack.

In conclusion, DoS attacks are thought to be simpler to thwart than DDoS attacks because they originate from a single source, are simpler to track, include less bandwidth, and are more focused.

✓ Cryptography and security rely extensively on the notion of trust, despite our inclination to always be skeptical. In considering the HTTPS protocol specifically, in which third-parties do we need to place trust? 1/1

Numerous factors contribute to how much cryptography depends on the concept of trust.

First off, robust and secure algorithms and protocols are crucial for cryptography to be effective in securing communication and data. Therefore, users must have confidence that the cryptographic primitives and methods used are secure against threats and are difficult for adversaries to crack.

Second, trust in the execution of these algorithms and protocols is necessary for the application of cryptography. Even though the cryptographic algorithms themselves are safe, the system's overall security can be jeopardised if they are used improperly or in a vulnerable manner. This is why it's crucial to develop cryptographic systems using libraries and frameworks that have been thoroughly validated and trusted.

Finally, many security methods and systems place a strong emphasis on trust. For instance, a number of third parties are trusted in the HTTPS protocol, which is used to protect web connections. These consist of:

Digital certificates, needed to verify the identification of websites and other online entities, are issued by certificate authorities (CAs), which are in charge of doing so. Users must have confidence in the CAs' ability to authenticate the websites and other entities for which they issue certificates and in the security of those certificates in order for HTTPS to function properly.

Server administrators: For HTTPS to function, users must have confidence that the administrators of the servers they are interacting with are correctly setting up and looking after their networks to guarantee the security of the communication.

Developers of browsers: Users must have confidence that the developers of the browsers they use have correctly implemented the HTTPS protocol and are rigorously examining the legitimacy of digital certificates.

Overall, trust is a crucial element of cryptography and security, and it's crucial that we put our faith in the correct people, organisations, and systems to guarantee the confidentiality and integrity of our data and communications.

✓ If a site doesn't offer HTTPS encryption, should you stop using that site? Why or why not? 1/1

An internet protocol called HTTPS (Hypertext Transfer Protocol Secure) is used to encrypt and secure conversations. Sensitive data, like login credentials and financial activities, are frequently protected by it.

The communication between your device and the site's server is not secure and may be intercepted by third parties if a site does not support HTTPS encryption. This implies that any confidential data you supply to the website, like passwords or credit card details, may be hacked.

Because of this, it is typically advised to stay away from utilising websites that do not support HTTPS encryption, especially for delicate tasks like online banking or shopping. Be aware of the hazards and, whenever possible, select an alternative site if you must use one that does not support HTTPS encryption.

It's important to keep in mind that some websites, such as blogs and news sites, may not offer HTTPS encryption for non-sensitive uses. In certain circumstances, the absence of HTTPS encryption might not be seriously risky. However, it is always advisable to proceed with caution when submitting any personal information online and to confirm that the website you are using has implemented the necessary security measures to safeguard your data.

✓ In your own words, distinguish the nature of the two types of "cross-site" attacks we discussed, cross-site scripting (XSS) and cross-site request forgeries (CSRF). 1/1

A vulnerability known as cross-site scripting (XSS) enables an attacker to insert malicious code into a website. When the victim's web browser runs this code, the attacker is able to steal sensitive data, like login credentials or personal information, as well as carry out other illegal operations on the victim's behalf.

XSS attacks can be classified as either stored or reflected. Malicious code is injected into a website's database during a stored XSS attack, where it is permanently saved and performed each time the page is loaded. Infected code is injected into a website's URL in reflected XSS attacks, which subsequently cause the malicious code to be reflected back to the user's browser and executed there.

A form of attack known as cross-site request forgery (CSRF) deceives a victim into taking undesired activities on a website. To accomplish this, the victim is shown a URL or form that appears legal but actually contains harmful demands. The fraudulent requests are carried out on behalf of the victim when they click the link or submit the form, potentially enabling the attacker to carry out illegal actions on the website.

Using adequate input validation and sanitization as well as putting security measures like content security policies and same-site cookies into place will help defend against XSS and CSRF attacks. Web applications and libraries must also be kept up to date because vulnerabilities are frequently found and resolved in new versions.

✓ In your own words, what is "information leakage," and what are some basic things that can be done to avoid it? 1/1

Information leakage is the unintentional or deliberate release of private or sensitive data to individuals or groups that are not authorised to have access to or view it. This can happen via a variety of channels, including emails, network traffic, and even physical papers.

To prevent information leakage, a few fundamental steps can be taken:

Put in place strict access controls: You can lessen the risk of leakage due to unauthorised or malicious access by allowing only authorised people to access sensitive information.

Use encryption: Even if sensitive information is unintentionally or accidently leaked, encryption can help prevent unauthorised individuals from viewing it.

Use secure communication channels: To prevent unauthorised parties from intercepting your transmission of critical information, always use secure communication channels. This can entail adopting safe email and file transfer methods or safe, encrypted chat services for communication.

Implement proper disposal procedures: It's critical to follow the correct steps when discarding physical papers or devices that contain sensitive information to prevent unwanted access to the data. This might entail erasing data from devices, shredding documents, or even physically destroying them.

Employee education can assist lower the risk of information leakage. Employees can be taught the value of securing sensitive information and the right ways to handle it.

✓ In your own words, what would someone do to perpetrate a SQL injection attack? 1/1

A cyberattack known as a SQL injection occurs when an attacker inserts malicious code into a website's database in an effort to access the data that is stored there without authorization. This is frequently accomplished by inserting carefully prepared SQL statements into the input or URL of a web form in an effort to deceive the database into executing unauthorised commands.

An attacker would often begin by locating a weak online application that uses a SQL database to store data in order to carry out a SQL injection attack. Then, after creating a malicious SQL statement, they would attempt to inject it into the web application so that the database would run it. This could be accomplished in a number of ways, including by changing the parameters of a URL, changing the input fields on a web form, or employing other strategies to deceive the programme into executing the malicious SQL statement.

If the attack is successful, the attacker might have access to confidential information kept in the database, be able to change or remove information, or even take full control of the database. Use prepared statements and parameterized queries, correctly sanitise all user input, maintain the database and web application software updated with the most recent security updates, and all of these precautions can help prevent SQL injection attacks.

✓ Why is "phishing" such a difficult problem to prevent? 1/1

Phishing is a type of cyberattack in which an attacker poses as a reliable entity in an email, message, or website in an attempt to fool a victim into disclosing sensitive information, including login credentials or financial information.

Phishing is a challenging issue to stop for a number of reasons:

It might be challenging to tell the difference between trustworthy and malicious communications: Attackers that engage in phishing frequently produce extremely convincing clones of reliable websites or send communications that look to come from reliable sources. Users find it challenging to distinguish between real and false messages as a result, especially if they are not aware of the strategies attackers employ.

Attackers' strategies are always changing: Phishers are always coming up with new ways to avoid detection and deceive their targets. As a result, organisations find it challenging to stay abreast of new dangers and safeguard their users.

Phishing attacks can be very particular in their targets: Because of this, it can be more challenging to identify and stop some phishing assaults from succeeding. Because they are customised to the specific interests and concerns of the victim, these attacks frequently have greater success.

Even if a company has strong security measures in place, individual individuals might still fall victim to phishing attacks if they are not vigilant. Users are vulnerable to these assaults. If users are unaware of the hazards and are unable to recognise a phishing attempt, they risk being deceived into disclosing critical information.

In conclusion, phishing is a challenging issue to prevent since it requires persuading users to divulge critical information, it is ever-evolving, it can be highly targeted, and users may not be aware of the risks.

Debrief

About how many MINUTES would you say you spent on this assignment in total? *

Just to set expectations for future students.

666