KULeuven

Development of secure software

Mooc
Web security
Intermediate Test

Kevin Loonen

1 Securing the communication channel

1.1 Key concepts of HTTPS

Quetion 1 Which of these statements about TLS is not true?

- TLS can be used for all kinds of traffic, including HTTP, POP, IMAP, SMTP, ...
- The security properties that TLS offers are confidentiality, integrity and authenticity
- **TLS ensures that an eavesdropper cannot observe the messages being exchanged ✓**
- HTTPS is nothing more than the HTTP protocol over a TLS-secured connection

Question 2 Confidentiality and integrity are achieved by algorithms that depend on a shared key. True or false?

- **True ✓**
- False

Question 3 Authenticity depends on the use of a certificate. True or false?

- **True ✓**
- False

Question 4 What happens if there is a problem establishing the authenticity of an HTTPS connection?

- The connection is aborted, and the browser shows a network error
- The connection is established as usual, but the browser marks it as insecure (no lock icon)
- **The connection is aborted, and the browser shows a warning explaining the problem ✓**
- The connection is established, because the browser falls back to a less secure protocol

1.2 Perfect forward secrecy

Question 1 Which of these statements are false when Perfect Forward Secrecy is enforced?

- **The private key belonging to the server's certificate changes every connection ✓**
- An attacker with possession of the private key belonging to the server's certificate can impersonate a legitimate server
- **An attacker can decrypt a recorded session if he obtains the private key belonging to the server's certificate ✓**
- **Perfect Forward Secrecy replaces authenticity ✓**

Question 2 The use of a public/private key pair and a certificate to exchange the pre-master secret ensures Perfect Forward Secrecy. True or false?

- True
- **False ✓**

Question 3 The plain Diffie-Hellman algorithm is not sufficient to setup a secure TLS connection with Perfect Forward Secrecy. True or false?

- **True ✓**(Does not offer authenticity)
- False

1.3 HTTPS in your application

Question 1 Which of the following statements about mixed content are true?

- Mixed content is always blocked by modern browsers
- Mixed content is HTTPS content being included in HTTP pages
- **The only way to address mixed content is by rewriting all URLs ✓**
- **Mixed content is technically an easy problem, but becomes hard in a large content base ✓**

Question 2 What is the biggest problem with partial HTTPS deployments?

- It can easily result in mixed content problems
- It allows the attacker to break the security properties of the HTTPS traffic
- **It allows the attacker to compromise the application before the upgrade to HTTPS happens ✓**
- It undermines the performance of the website

Question 3 Which statements become true if you turn off support for HTTP on a webserver?

- **The server will never serve an insecure HTTP resource ✓**
- Browsers detect the error and upgrade the connection to HTTPS
- **Existing links to the HTTP version of the application will break ✓**
- A network attacker can no longer attack a user trying to connect to the HTTP version of your application

Question 4 The most user-friendly and secure way to deploy HTTPS is to redirect all HTTP URLs to their HTTPS counterparts. True or false?

- **True ✓**
- False

1.4 HTTP Strict Transport Security

Question 1 If you enable HSTS, browsers will never send an HTTP request to your server. True or false?

- True
- **False ✓**

Question 2 Which statements are true if you consider the following HSTS configuration: "max-age=432000;preload"?

- **The HSTS policy is valid ✓**
- If the user visits the application again within the next 7 days, every request will be sent over HTTPS
- The HSTS policy applies to all subdomains as well
- The domain sending the HSTS header can be added to the preload list

Question 3 What is the biggest drawback of the "includeSubdomains" flag

- The browser needs to visit the top-level domain to receive the HSTS policy
- **The HSTS policy may make legacy HTTP services running on any subdomain unavailable ✓**
- There is no easy way to disable the HSTS policy
- The "includeSubdomains" flag can only be set on the top-level domain

1.5 The trust model of HTTPS

Question 1 If you request a legitimate certificate for your besite, it will be signed by a root CA. True or false?

- True
- **False ✓**

Question 2 Which statement most accurately describes the biggest weakness in the HTTPS ecosystem

- CAs fail to provide adequate security, and get hacked frequently
- Users have no idea what a certificate is, and cannot correctly assess insecure situations
- **Browser accept certificates from hundreds of different root CAs ✓**
- Certificates are too error-prone, and errors are often false positives

Question 3 There are multiple levels of certificate validation. Which level of certificate unlocks the most powerful cryptographic features?

- Extended validation certificates
- Organization validation certificates
- Domain validation certificates
- **All of the above ✓**

Question 4 Which statements are true with regard to domain validated certificates?

- Domain validation requires a manual sign-off by the CA, so it cannot be automated
- **Domain validation checks if the requester controls the domain ✓**
- Domain validation checks if the requester is also the official registrant of the domain
- **Domain validation is enough for the browser to show a (green) padlock ✓**

1.6 The fragility of certificates

Question 1 Which of the following problems is solved by Certificate Authority Authorization (CAA)?

- A browser accepting any valid certificate for a website, regardless of which CA signed it
- A CA issuing a certificate to someone that does not control the domain
- **The fact that any CA can issue certificate for any domain ✓**
- The lack of a connection between DNS records an certificates

Question 2 Which of the following problems is solved by Certificate Transparency (CT)?

- The lack of a catalog of all issued certificates
- **The fact that it is hard to detect a fraudulent certificate being used ✓**
- The fact that a browser cannot determine if a certificate is legitimate
- The lack of transparency about which CAs have been compromised

Question 3 Certificate Transparency by itself does not prevent anattacker from using a fraudulent certificate. True or false?

- **True ✓**
- False

Question 4 Which of these mechanisms is not a valid way to transport SCT information to the browser?

- Embedding the SCT information into the certificate
- Adding the SCT information to an OCSP response
- **Adding the SCT in a dedicated DNS record ✓**
- Sending the CT information during the TLS handshake

1.7 Securing the communication channel

Question 1 Which of these statements about HTTPS are true?

- Authenticity follows from confidentiality and integrity
- **Confidentiality and integrity are useless without authenticity ✓**
- **Authenticity is useless without integrity ✓**
- Authenticity is useless without confidentiality

Question 2 Confidentiality and integrity is achieved by algorithms that depend on a shared key. True or false?

- **True** ✓
- False

Question 3 Which of these statements are false when Perfect Forward Secrecy is enforced?

- **An attacker can decrypt a recorded session if he obtains the private key belonging to the server's certificate** ✓
- **The private key belonging to the server's certificate changes every connection** ✓
- An attacker with possession of the private key belonging to the server's certificate can impersonate a legitimate server
- **Perfect Forward Secrecy replaces authenticity** ✓

Question 4 Browsers block mixed content because an attacker can eavesdrop on the HTTP request. True or false?

- True
- **False** ✓(block for malicious code in the HTTP responses)

Question 5 The most user-friendly and secure way to deploy HTTPS is to redirect all HTTP URLs to the homepage of the HTTPS application. True or false?

- True
- **False** ✓

Question 6 A website's public pages are served over HTTP, but from the moment you need to login, it switches HTTPS. In which cases can this be considered to be secure?

- When the TLS configuration ensure Perfect Forward Secrecy
- When the HTTPS pages do not contain mixed content
- When the HTTPS pages are deployed on a different domain and HSTS is enabled on this domain
- **Never. They will always be insecure** ✓(Attacker can prevent the upgrade to HTTPS)

Question 7 A university has one registered domain, and each faculty administers their own subdomain, under which numerous applications are hosted. Which of these actions should the university take right now to start working towards the mandatory use of HTTPS and HSTS?

- **Enable an HSTS policy, without the includeSubdomains flag, on the main university website, which already runs over HTTPS ✓**
- Enable an HSTS policy, with the includeSubdomains flag, on the registered domain of the university
- **Enable an HSTS policy, without the includeSubdomains flag, on each application running over HTTPS ✓**
- Enable an HSTS policy, with the includeSubdomains and preload flag, on the registered domain of the university, and add the domain to the preload list

Question 8 Certificate Transparency (CT) will achieve its goal, but only if every domain administers monitors the logs. True or false?

- **True ✓**
- False

Question 9 Which of these consequences is an unexpected side-effect of Certificate Transparency (CT)?

- Every website needs to update their certificate to become compliant with CT
- Browsers need to be forced to enforce the presence of an SCT
- Administrators will have to jump to extra hoops to configure HTTPS on their servers
- **The registration of all certificates in a log leaks information about the existence of a website or host ✓**

Question 10 The CA will request proof of your identity, investigate your business and check if your business has the right to use the domain name you're requesting a certificate for. Which level of validation corresponds with this statement?

- Domain validation
- Organization validation
- **Extended validation ✓**
- None of the above

2 Preventing unauthorized access

2.1 Passwords

Question 1 Which of these statements about a password manager are true?

- A password manager is always more secure, even if you use it to store your old passwords
- **A password manager allows youo to use a unique password for each site ✓**
- A password manager prevents phishing by looking at the visual elements in a website
- **If you use a password manager as recommended, a data breach in one application will not result in the compromise of your account in a second application ✓**

Question 2 Why is it so easy to break MD5/SHA1 hashes?

- Because MD5 and SHA1 are weak algorithms. SHA256 or SHA3 hashes are not easy to break.
- Because of implementation errors by the developer.
- **Because hashingg algorithms are designed to be fast, so brute forcing is also really fast. ✓**
- Because the passwords were not salted

Question 3 If you use a dedicated password hashing algorithm, you no longer need to use a salt. True or false?

- True
- **False ✓**

Question 4 What is the best approach to gradually improve the strength of your password storage algorithm over time?

- Reset all paswords, and have them choose a new password when they login
- Keep the old hashes, but have the user choose a new password when they login
- **Keep the old hashes, but upgrade the hash when a user logs in ✓**
- Use the current hashes as input for the new algorithm

2.2 Enumeration attacks

Question 1 Which statement most accurately describes an enumeration vulnerability?

- An enumeration vulnerability allows an attacker to collect information about valid users
- **An enumeration vulnerability leaks information about the existence of objects in the system ✓**
- An enumeration vulnerability is the enabler of a brute force attack

Question 2 Which of the following scenarios indicates an enumeration vulnerability in the application?

- The registration form tells the user that an email has been sent to the given address
- **The registration form asks the user to choose a username and a password ✓**
- The account recovery form tells the user that an email has been sent to the given address
- The authentication form informs the user that the username and password are invalid

Question 3 Both locking an account and slowing down authentication attempts are valid defenses against a brute force authentication attack. True or false?

- **True ✓**
- False

2.3 Cookie-based session management

Question 1 To which naming scheme do cookies containing a session identifier have to adhere?

- **Any name goes, as long as it uses characters that are allowed in a cookie ✓**
- Any name goes, as long as it contains the word "SESSIONID"
- Any name goes, as long as it contains the word "SESSION"
- Only one of the following names is allowed: SESSION, SESSIONID, PHPSESSIONID, JSESSIONID

Question 2 Without the Secure flag, the cookie will not be attached to requests to an HTTPS page. True or false?

- True
- **False ✓**

Question 3 If the application does not use cookie flags, how can an attacker obtain the victim's session identifier?(Indicate all valid responses)

- **The attacker can guess the session identifier ✓**
- **The attacker can predict the session identifier based on his own session identifier ✓**
- **The attacker can steal the session identifier by eavesdropping on HTTP requests on the network ✓**
- **The attacker can steal the session identifier by executing malicious JavaScript code in the application's browsing context ✓**

Question 4 The HttpOnly and Secure flag are only relevant to protect the cookie with the session identifier. True or false?

- True
- **False ✓**

2.4 Common authorization problems

Question 1 Which statement most accurately describes a CSRF attack?

- The attacker sends a request from an unrelated browsing context in his browser to the server, to perform an operation in the user's name
- **The attacker sends a request from an unrelated browsing context in the victim's browser to the the server, to perform an operation in the user's name ✓**
- The attacker sends a request from the application's browsing context in his browser to the server, to perform an operation in the user's name
- The attacker sends a request from the application's browsing context in the victim's browser to the server, to perform an operation in the user's name

Question 2 Which security property oes a CSRF mitigation technique using hidden form tokens depend upon?

- The fact that cookies are associated with a domain, not an origin
- The fact that an HttpOnly cookie cannot be read from JavaScript
- **The fact that the browser prevents the attacker from reading the token ✓**
- The fact that the browser removes the token from the request

Question 3 An insecure direct object reference vulnerability only occurs with identifiers generated by the database. True or false?

- True
- **False ✓**

Question 4 In essence, an insecure direct object reference vulnerability is nothing more than a missing authorization check. True or false?

- **True ✓**
- False

2.5 Preventing unauthorized access

Question 1 Using a salt makes the password unique, yet password reuse across applications remains a problem. True or false?

- **True ✓**
- False

Question 2 Which of these statements about storing passwords is true?

- Running a large number of iterations of SHA3 is just as good as using a password-based hashing function
- **The benefit of a password-based hashing function is its expensiveness to execute ✓**
- Brute forcing of MD5 and SHA1 hashes is only possible because the algorithms are weak
- Brute forcing bcrypt hashes is slow because they use a long salt

Question 3 An application is currently using plaintext passwords in the database. Which upgrade strategy would you advise?

- Delete all passwords, and have each user reset their password
- Modify the authentication procedure, so that it will update the stored hash when a user logs in
- Use the MD5 hash of these passwords as input for bcrypt, and update the hashes when the user logs in
- **Calculate bcrypt hashes from the passwords, and change the verification procedure during login ✓**

Question 4 Which of these statements about enumeration attacks make no sense?

- **An enumeration attack allows an attacker to steal user credentials from the database ✓**
- Advanced enumeration attacks can triage responses on other criteria than visible error messages
- **Enumeration attacks are irrelevant if you deploy strong brute force defenses ✓**

Question 5 An American airline currently uses the following authentication scheme: a username and password in combination with the answer to a secret question. Is this two-factor authentication?

- Yes
- **No ✓**(Both are knowledge based)

Question 6 Which of these statements most accurately describes most security challenges with session management?

- **Cookies are attached to every outgoing request to a particular domain ✓**
- Not every application uses HTTPS by default
- Cookies are not always set with the Secure flag
- Server-side sessions can be taken over by simply stealing an identifier

Question 7 Using the HttpOnly and Secure flag for the session cookie prevents all session hijacking attacks. True or false?

- True
- **False ✓**(guessing/predicting the identifier, ...)

Question 8 Client-side sessions are different from server-side sessions. Which of the following statements are true?

- **Client-side session data requires additional integrity checks before it can be used ✓**
- **Client-side sessions are also vulnerable to session hijacking ✓**
- Client-side session object are not limited in size
- Client-side sessions do not work well with cookies

Question 9 Switching to HTTPS does not make it more difficult to execute a CSRF attack. True or false?

- **True ✓**
- False

Question 10 Which of these strategies can be used to mitigate insecure direct object reference vulnerabilities?

- Only exposing SHA3 hashes of the numerical identifier to the client
- **Only exposing randomly generated identifiers to the client ✓**
- **Only exposing indirect object references to the client ✓**
- Implementing rate limits on requesting different objects of the same type

3 Securely handling untrusted data

3.1 Command injection

Question 1 What is the most accurate cause of a command injection attack?

- The dynamic construction of a command string before executing it
- The use of user input in the command string
- **The use of untrusted data to construct the command string ✓**
- The execution of a command from within another program

Question 2 How would you most accurately describe the result of a successful command injection attack?

- The attacker takes full control over the server
- **The attacker executes a command with the privileges of the script that executed the system command ✓**
- The attacker executes a command as the root user
- The attacker executes a command with the privileges of the web server

Question 3 Command injection problems only exist in web applications. True or false?

- True
- **False ✓**

Question 4 Which defense can not be used to mitigate command injection attacks?

- **Preventing the submission of dangerous characters when validating the form in the browser ✓**(Client-side defenses are easy to bypass)
- Use an API that separates the command from its parameters, if supported by the language
- Encode untrusted data before appending to the command string
- Input validation on the server, before using the data

Question 5 What is the best defense against command injection attacks?

- Preventing the submission of dangerous characters when validating the form in the browser
- Preventing the use of the semicolon (;)
- **Use an API that separates the command from its parameters, if supported by the language ✓**
- Encode untrusted data before appending to the command string
- Input validation on the server, before using the data

Question 6 Why are Java and .NET not vulnerable to command injection attacks?

- The Java/.NET standard libraries escape dangerous characters before running the command in a shell
- The Java/.NET virtual machine executes bytecode, and not low-level binary code
- The Java/.NET virtual machine has no privileges to run system commands
- **The Java/.NET standard libraries do not execute the command in a shell, but invoke it directly ✓**

3.2 SQL Injection

Question 1 What is the most accurate description of the cause of a SQL injection vulnerability?

- The application uses dynamically created SQL statements
- The application uses user input to create a SQL statement
- The application creates the statement and the database server executes it
- **The application does not provide enough context information to distinguish data from code ✓**

Question 2 In which of the following SQL statements would this payload result in a successful SQL injection attack: "OR 1=1–

- "SELECT * FROM " + table + "WHERE visibility = 1"
- **"SELECT * FROM users WHERE username = " + username + " AND password = " + password ✓**
- **"SELECT * FROM notes WHERE owner = " + userId ✓**
- "SELECT id, title, " + column + " FROM notes"

Question 3 Which one of these application features is least likely to be vulnerable to SQL injection

- The authentication procedure
- **Retrieving a list of all notes ✓**
- Creating a new tasting note
- Deleting a note from the database

Question 4 What is the best practice mitigation strategy against SQL injection vulnerabilities?

- Strict input validation
- **Using prepared statements with variable binding ✓**
- Encoding user data before adding it to the statement
- Using prepared statements

Question 5 Which mitigation strategies apply if you want to build a SQL statement where the table name comes from an untrusted value?

- **Selecting a value from a whitelist based on the input ✓**
- Using prepared statements with variable binding
- **Encoding the untrusted data for the right DBMS system ✓**
- Putting the table name between double quotes

3.3 Cross-Site Scripting

Question 1 Which most accurately describes the result of the successful exploitation of a cross-site scripting vulnerability?

- The attacker can execute arbitrary commands on the server
- The attacker has full control over the browser
- The attacker can execute arbitrary commands on the client
- **The attacker has full control over the application's browsing context ✓**

Question 2 A cross-site scripting attack always involves one site sending malicious code to another site. True or false?

- True
- **False ✓**

Question 3 Which statements are correct regarding the difference between a reflected and stored XSS attack?

- Reflected XSS attacks happen on the client, and stores XSS attacks happen on the server
- Reflected XSS attacks give the attacker less capabilities than stored XSS attacks
- **It is harder to make a lot of victims with a reflected XSS attack than with a stored XSS attack ✓**
- **Reflected XSS attacks can be stopped by the browser, and stored XSS attacks cannot ✓**

Question 4 Which of the following mitigation strategies are effective to prevent cross-site scripting vulnerabilities?

- Always put JavaScript code in seperate files
- **Sanitize untrusted data before using it as output ✓**
- Validate all input against a list of known script injection attack vectors
- Encode all HTML characters in untrusted data
- **Encode dangerous characters in untrusted data, based on the context where it will be used ✓**

Question 5 Which of the following statements are true?

- XSS defenses are only relevant for rich-text output, not for simple strings
- XSS payloads can only come from user input
- **It is better to encode data for a specific context than to encode all dangerous characters, regardless ✓**
- The mitigation strategies for reflected XSS are different than those for stored XSS
- **Output encoding applies to all pieces of code, regardless of whether it is benign or malicious ✓**

Question 6 In which case is sanitization the best mitigation strategy?

- When the untrusted data contains a large piece of text
- When the untrusted data does not contain user input
- **When the untrusted data needs to contain benign code ✓**
- When the untrusted data is composed from different sources

3.4 DOM-based XSS

Question 1 Which definition most accurately describes DOM-based XSS

- DOM-based XSS occurs when the injected JavaScript modifies the DOM
- **DOM-based XSS is an XSS vulnerability caused byt client-side modifications of the DOM ✓**
- DOM-based XSS is an XSS vulnerability that is neither reflected or stored
- DOM-based XSS attacks occur because the application uses untrusted data from the DOM

Question 2 DOM-based vulnerabilities cannot occur if you do not have dynamic server)generated pages. True or false?

- True
- **False ✓**

Question 3 Which mitigation strategies are effective against DOM-based XSS?

- **Using safe DOM APIs to create new content ✓**
- **Applying context-sensitive encoding to untrusted data before putting it into the page ✓**
- **Applying sanitization to untrusted data before putting it into the page ✓**
- Using client-side libraries such as jQuery for all DOM modifications

Question 4 DOM-based vulnerabilities are as powerful as traditional XSS vulnerabilities. True or false?

- **True ✓**
- False

3.5 Advanced attacks and defenses

Question 1 Cross-site scripting, HTML injection and CSS injection are all caused by the same vulnerability. True or false?

- **True ✓**
- False

Question 2 Which mitigation strategies are effective against various types of content injection attacks (XSS, HTML injection, CSS injection, ...)?

- Server-side sanitization
- Client-side sanitization
- Input validation
- **Context-sensitive output encoding ✓**

Question 3 For which scenarios would a sandboxed iframe be a good mitigation strategy?

- **To isolate a block of untrusted content from the main application ✓**
- **To isolate untrusted content without having to deploy your application across multiple origins ✓**
- To prevent harmful consequences from CSS injection attacks
- **To enforce behavioral restrictions on the content loaded in the iframe ✓**

Question 4 Which of the following statements is true?

- Isolating content in an iframe with a different origin offers weaker isolation as a sandbox with a unique origin
- **Isolating content in an iframe with a different origin offers the same isolation as a sandbox with a unique origin ✓**
- Isolating content in an iframe with a different origin offers stronger isolation as a sandbox with a unique origin

Question 5 Which of the following sandbox configurations fails to isolate malicious script content?

- sandbox
- sandbox="allow-same-origin"
- **sandbox="allow-same-origin allow-scripts" ✓**
- sandbox="allow-scripts"
- sandbox="allow-scripts allow-top-navigation"

3.6 Content Security Policy

Question 1 Which defenses does CSP offer that help mitigate script injection attacks?

- **CSP prevents the execution of inline scripts ✓**
- CSP sanitizes inline scripts before executing them
- CSP prevents the loading of external scripts
- **CSP only loads external scripts if they are whitelisted ✓**

Question 2 If a CSP policy contains a default-src directive and an img-src directive, then the following rules will be applied to images in the page:

- Only the default-src directive applies
- **Only the img-src directive applies ✓**
- The default-src directive and the img-src directive both apply
- The img-src directive applies first, and if no match is found, the default-src directive applies

Question 3 If CSP is deployed in report-only mode with a report-uri directive, and the browser encounters an injected script block, it will:

- Block the script from being executed
- Block the script from being executed and send a report to the reporting endpoint
- Execute the script
- **Execute the script and send a report to the reporting endpoint ✓**

Question 4 If a CSP policy contains a hash, as well as the *unsafe-inline* keyword. What happens when a browser supporting CSP Level 2 encounters an inline script block?

- It prevents the script from executing
- It allows the script block to execute
- **It allows the script block to execute if it matches the hash ✓**(The 'unsafe-inline' keyword is ignored when hashes are used, te script block needs to match the hash)
- It allows the script block to execute if it uses the proper DOM APIs

Question 5 What is the most accurate description of the new *strict-dynamic* keyword?

- It instructs the browser to only trust scripts that have a valid nonce
- It instructs the browser to let trusted scripts load additional resources
- It instructs the browser to let trusted scripts load additional resources, if they have a valid nonce
- **It instructs the browser to let trusted scripts load additional resources, if they use the proper DOM APIs ✓**

3.7 Securely handling untrusted data

Question 1 What is the most accurate description of the root cause of injection attacks?

- Combining untrusted data and code
- Failure to properly encode untrusted data
- **The lack of context information to distinguish between data and code ✓**
- The lack of input validation on untrusted data

Question 2 Which of these statements about command injection are true?

- **The best defense against command injection is using a context-aware API ✓**
- The best defense against command injection is input validation and output encoding
- Higher-level languages do not suffer from command injection because they run in a bytecode VM
- Low-level languages have command injection vulnerabilities because they are inherently insecure

Question 3 Which of these statements most accurately describes the effect of using parametrized statements with variable binding?

- **They separate data and code, so that the data can be handled securely ✓**
- The count the number of variables, and report an error if more variables are detected in the statement
- They enforce type safety on the parameters, so that they cannot be interpreted as code
- They force the developer to think about SQL injection

Question 4 Cross-site scripting attacks can be defined as follows: one site uses the victim's browser to inject script code into another site. True or false?

- True
- **False ✓**

Question 5 Which of these statements about XSS does not make sense?

- A good way to prevent XSS vulnerabilities is to use statically generated pages
- XSS allows the attacker to launch attacksagainst the victim's browser
- **XSS allows the attacker to violate the Same-Origin Policy in the victim's browser ✓**(SOP prevents XSS attacks from different origins, not the same.)
- XSS allows the attacker to remotely control the compromised browsing context

Question 6 What is the most accurate description of the effect of this sandbox attribute: sandbox="allow-scripts"?

- Code in the iframe does not run
- **Code in the iframe runs, but is isolated from the rest of the page ✓**
- Code in the iframe runs, but can still access its parent page
- The use of allow-scripts void any protection guarantees

Question 7 Blocking the execution of JavaScript is insufficient to prevent an attacker from abusing an injection vulnerability. True or false?

- **True ✓**
- False

Question 8 What is the use of the default-src directive?

- To specify commonparts for all types of resources
- **To specify a policy for content types that are not specified in the CSP policy ✓**
- As a complementary policy to enforce on all types of resources
- As a shorthand to setup a CSP policy

Question 9 Defining a strict CSP policy is a good replacement for traditional defenses against content injection attacks. True or false?

- True
- **False** ✓ (It's a second line of defense)

Question 10 Which of these problems with CSP are addressed by the introduction of hashes?

- The blocking of inline event handlers
- **The blocking of inline code blocks** ✓
- The blocking of additional resources required by whitelisted scripts
- The need to specify overly long whitelists

4 Final Exam

4.1 Securing the communication channel

Question 1 What will happen if a page contains mixed content from its own domain, but the application has an HSTS policy configured?

- **The browser refuses to lead HTTP resources, so it does not even check if HSTS is enabled ✓**
- The browser will prompt the user to ask if it should attempt to load the resources securely
- The browser will upgrade the URLs to HTTPS and load the resources securely
- Mixed content is only relevant for other domains, not for your own domain

Question 2 Supporting the older SSL is only acceptable if you need to support legacy clients. True or false?

- **True ✓**
- False

Question 3 Which of these statements is false, assuming the browser has an active HSTS policy for the application?

- If the user does not type a protocol in the address bar, the browser defaults to HTTPS
- **If the user explicitly types http:// in the address bar, the browser does not upgrade the URL to HTTPS ✓**
- If an external page links to the application with an HTTP URL, the browser upgrades it to HTTPS
- If the application sends a redirect to an HTTP resource, the browser upgrades it to HTTPS

Question 4 Deploying TLS using a third-party service such as Cloudfare offers the same or better security properties than deploying it yourself. True or false?

- True
- **False ✓**(third party has access to the plain text traffic, so it's weaker for full end-to-end encryption)

Question 5 In which of these scenarios is Certificate Transparency useful? Assume Certificate Transparency is used correctly.

- **An attacker obtaining a valid certificate that is not published in a log ✓**
- An attacker obtaining a valid certificate for a phishing site with an unrelated domain name
- **An attacker obtaining a certificate by circumventing domain validation ✓**
- **An attacker obtaining a certificate by hacking a CA ✓**

4.2 Preventing unauthorized access

Question 6 What should you do to upgrade the cost factor of your bcrypt algorithm over time?

- Reset all passwords, and have them choose a new password when they login
- Keep the old hashes, but have the user choose a new password when they login
- **Keep the old hashes, but upgrade the hash when a user logs in ✓**
- Use the current bcrypt hashes as input for the new hash with a higher cost factor

Question 7 An application offers two-factor authentication. The two factors are a username/password combination, and a verification code sent over SMS. Which of these statements are false?

- Mobile phone numbers are easy to compromise, so the site should refrain from using them
- **Mobile phone numbers are easy to compromise, so users should not enable the second factor ✓**
- **Mobile phone numbers are easy to compromise, so the site should send the code via email instead ✓**

Question 8 The HttpOnly and Secure flag are only relevant to protect the cookiewith the session identifier. True or false?

- True
- **False ✓**

Question 9 Which of the following statements about CSRF do not make sense?

- **A CSRF attack cannot be used to perform transactions that require multiple steps ✓**
- CSRF allows the attacker to perform actions in the user's name
- **Every session management mechanism suffers from CSRF vulnerabilities ✓**
- **CSRF allowsthe attacker to transfer the user's session to a different browser ✓**

Question 10 In essence, an insecure direct object reference vulnerability is nothing more than a missing authorization check. True or false?

- **True ✓**
- False

4.3 Securely handling untrusted data

Question 11 In which of these injection attacks does the server inject the malicious code into the context where it is executed?

- **SQL injection ✓**
- **Command injection ✓**
- **Traditional cross-site scripting ✓**
- DOM-based cross-site scripting

Question 12 How would you most accurately describe the result of a successful command injection attack?

- The attacker executes a command as the root user
- **The attacker executes a command with the privileges of the script that executed the system command ✓**
- The attacker executes a command with the privileges of the web server
- The attacker takes full control over the server

Question 13 Which of the following statements is true?

- **Isolating content in an iframe with a different origin offers the same isolation as a sandbox with a unique origin ✓**
- Isolating content in an iframe with a different origin offers stronger isolation as a sandbox with a unique origin
- Isolating content in an iframe with a different origin offers weaker isolation as a sandbox with a unique origin

Question 14 Which of the following statements are true?

- **It is better to encode data for a specific context than to encode all dangerous characters, regardless of the context ✓**
- **Output encoding applies to all pieces of code, regardless of whether it is benign or malicious ✓**
- XSS defenses are only relevant for rich-text output, not for simple strings
- XSS payloads can only come from user input
- The mitigation strategies for reflected XSS are different than those for stored XSS

Question 15 Which features in CSP do hashes and nonces override?

- **The use of 'unsafe-inline' ✓**
- The use of 'unsafe-eval'
- The use of whitelists
- The use of 'self'

4.4 Course overview

Question 16 Which of these statements with regard to the Same-Origin Policy (SOP) are true?

- The use of HTTPS ha nothing to do with the SOP
- **The CSRF defense using hidden tokens relies on the protections offered by the SOP ✓**
- The sandbox attribute inherently modifies the way SOP works
- If the SOP would be stricter, applications would not suffer from XSS

Question 17 Because of common vulnerabilities in popular libraries, it's better to write your own defenses. True or false?

- True
- **False ✓**

Question 18 What can a local network attacker not do if your application uses HTTPS with HTTP Strict Transport Security enabled (HSTS)?

- Impersonate your application with a fraudulent but legitimate certificate
- **Impersonate your application with an invalid certificate, and trick the user into accepting the SSL warning ✓**
- Trick the user into authenticating to a phishing site that also uses HTTPS
- **Fake a network error to cause the browser to drop the HSTS policy ✓**

Question 19 Which of the following statements describes the best way to protect your application against server-side injection attacks?

- Handle user input carefully, and ensure it cannot be interpreted as code
- Apply strict input validation to all user-provided data
- **Treat every piece of data in the application as potentially untrusted, regardless of where it comes from ✓**
- Use a higher-level language, such as Java or .NET

Question 20 Which of these attacks becomes harder if cookies would be associated with an origin, and not a domain?

- Session hijacking from JavaScript
- **Session hijacking on the network ✓**
- Cross-Site Request Forgery (CSRF)
- Reflected cross-site scripting