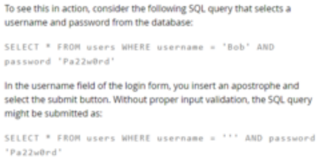
Pentest+ Lesson 13 - Web Application-Based Attacks

Study online at https://quizlet.com/_d1ezf6

code signing	The method of using a digital signature to ensure the source and integrity of programming code.
You have been asked to recommend a web framework for an application that incorporates HTML and/or JavaScript code. What would you suggest?	AngularJS, Ruby on Rails, or Django (Python).
You need to choose a query language for your client's application to write to and read from a database. Which language protocol would you suggest?	SQL
You have been asked to make a client presentation on the OWASP Top Ten. What are some of the critical security risks that you could discuss?	Answer can include any/all of the following: Injection Broken Authentication Sensitive Data Exposure XML External Entities (XXE) Broken Access Control Security Misconfiguration Cross-Site Scripting (XSS) Insecure Deserialization Using Components with Known Vulnerabilities Insufficient Logging & Monitoring
Session Hijacking	A malicious actor steals a user's session credential then uses it to impersonate the user. From the server's perspective, every request it receives from the client is new. If authentication is required, the user must either re-enter credentials for every new request, or some other mechanism must exist to tie all requests into a single continuous session. The most common mechanism for doing this is a cookie
cookie	Text file used to store information about a user when they visit a website. Some sites use cookies to support user sessions. It contains the session ID (SID) for that particular web session and is used as an authentication token. The browser keeps presenting the cookie every time a request is made. In this way, the user does not need to keep re-entering credentials at the website.
session fixation	An attack that forces a user to browse a website in the context of a known and valid session. As an example, this could be done through social engineering and providing a fake login page that will use the known SID.
session replay	This requires having access to the user authentication process itself, so that it can be intercepted and repeated. This could be achieved through a man-in-the-middle attack.
cross-site request forgery (XSRF/CSRF)	A malicious script hosted on the attacker's site that can exploit a session started on another site in the same browser. For example, this type of attack could leverage a web browser's trust in a user's unexpired browser cookies. The power of CSRF comes from the fact that it is extremely difficult to detect since the attack is carried out by the user's browser just as it normally would be if the user made the request themselves. The user could enter that same URL manually and get the same result. It is almost impossible for the browser to distinguish a successful CSRF attack from normal user activity.
server-side request forgery attack	An attack where an attacker takes advantage of the trust established between the server and the resources it can access, including itself.
Horizontal Privilege Escalation	When a user accesses or modifies specific resources that they are not entitled to. obtaining access to a regular user account with different access

	or permissions than the one currently in use. This approach has great potential for information gathering without raising possible suspicion, as irregular user activity is more likely to stay unnoticed than irregular admin activity.
Vertical Privilege Escalation	When an attacker can perform functions that are normally assigned to users in higher roles, and often explicitly denied to the attacker. obtaining access to an account of higher privilege than the one we currently have to enable resources that the regular user does not have permission for. In some cases we will need vertical PrivEsc, such as when we want to upgrade a "restrictive shell."
business logic flaws	Vulnerabilities that arise from implementation and design issues that lead to unintended behavior.
You have been asked to help make a presentation to your client's C-level executives. Your assignment is to explain Session Attacks. Which vulnerabilities could you discuss?	Correct answer can highlight any of the following: Session hijacking Cookie Session fixation Session replay
A user comes to you with a problem. They explain that they wanted to purchase some IT books from the online company bookstore but their shopping cart has changed its contents. They think this is strange because they don't want 50 of the same book for themself. What could be the cause?	This could be the result of Cross-Site Request Forgery (XSRF / CSRF).
You have been asked to PenTest a client's network. They have asked for you to only use horizontal privilege escalation. What is a benefit of this type of escalation?	This approach has great potential for information gathering without raising possible suspicion, as irregular user activity is more likely to stay unnoticed than irregular admin activity.
SQL injection	An attack that injects a database query into the input data directed at a server by accessing the client side of the application. you can modify one, or more, of the four basic functions of SQL querying (selecting, inserting, deleting, and updating) by embedding code in some input within the web app, causing it to execute your own set of queries using SQL. To identify SQL injection vulnerabilities in a web app, you should test every single input to include elements such as URL parameters, form fields, cookies, POST data, and HTTP headers.

single quote method	The simplest and most common method for identifying possible SQL injection vulnerabilities in a web app is to submit a single apostrophe and then look for errors. If an error is returned, you can see if it provides you with SQL syntax details that can then be used to construct a more effective SQL injection query.
	
	To see this in action, consider the following SQL query that selects a username and password from the database:
	`SELECT * FROM users WHERE username = 'Bob' AND password 'Pa22w0rd'`
	In the username field of the login form, you insert an apostrophe and select the submit button. Without proper input validation, the SQL query might be submitted as:
	`SELECT * FROM users WHERE username = ''' AND password 'Pa22w0rd'`

stack multiple queries	The process of modifying the SQL query to include new query type.
UNION SELECT '1', '2' FROM users—	For example, let's say you have a product search form that you've probed for SQL injection weaknesses. You could perform the following query on the search form to try to merge the users table with the products table, looking for the first two values from users
UNION SELECT '1', '2', '3', '4', '5' FROM users—	However, UNION operations only work when both queries (i.e., the initial SELECT from products and the UNION SELECT from users) have the same number of columns. So if the products table has five columns, you need to adjust your injection to include them

blind SQL injection	The process of injecting SQL queries when the web application's response does not contain the result of the query.
Boolean-based blind SQLi	The process of injecting SQL queries with values that are always true ('1=1') and false ('1=2').
time-based blind SQLi	The process of injecting SQL queries with time delays
Directory Traversal	An application attack that allows access to commands, files, and directories that may or may not be connected to the web document root directory. You can do this by inducing a web app to backtrack through the directory path so that the app reads or executes a file in a parent directory. The most simple example of directory traversal involves sending a ..\ or ../ command request to the application or API, which then traverses up one parent directory for each one of these commands.
null byte	A character with a value of zero that is used in most programming languages to indicate the termination of a string. With a poison null byte, you can use this termination character to exploit a web app that does not properly handle null terminators. The hexadecimal representation of the poison null byte is %00. The poison null byte can support several different attacks, including directory traversal.
code injection	Exploit technique that runs malicious code with the ID of a legitimate process. Injection attacks enable you to compromise an app in many ways, including: Causing a denial of service (DoS) of the app Escalating access privileges in the app Exposing and exfiltrating sensitive data in databases such as user credentials and PII Installing malicious software on the server hosting the app Defacing a website
command injection	Where a threat actor is able to execute arbitrary shell commands on a host via a vulnerable web application.
LDAP (Lightweight Directory Access Protocol)	Network protocol used to access network directory databases, which store information about authorized users and their privileges, as well as other organizational information. It can be used by web applications to perform tasks according to user input, so it is a possible location to attempt injection. The techniques employed look similar to SQL injection: x' or name()='username' or 'x'='y
Cross-Site Scripting (XSS)	A malicious script hosted on the attacker's site or coded in a link injected onto a trusted site designed to compromise clients browsing the trusted site, circumventing the browser's security model of trusted zones.
persistent XSS	also called a stored attack, you inject malicious code or links into a website's forums, databases, or other data. When a user views the stored malicious code, or clicks a malicious link on the site, the attack is perpetrated against them. As the name suggests, the injected code remains in the page because it is stored on the server.
reflected XSS	you craft a form or other request to be sent to a legitimate web server. This request includes your malicious script. You then send a link to the victim with this request and when the victim clicks that link, the malicious script is sent to the legitimate server and reflected off it. The script then executes on the victim's browser.

	Unlike a stored attack, the malicious code in a reflected attack does not persist on the server.
Document Object Model (DOM)-based attack	When attackers send malicious scripts to a web app's client-side implementation of JavaScript to execute their attack solely on the client.
Your coworker is stuck writing a report about SQL Injection attacks. However, they are in a hurry and cannot recall the four basic functions of SQL querying. You want to help them out. What are they?	Selecting, Inserting, Deleting, and Updating
You are on a security team and have found evidence of someone accessing a file from a location that the user is not authorized to access. What is one attack method that could be causing this prohibited process?	Directory traversal
You are on a PenTesting team and have decided to use a code injection attack to test a client's application. In what ways can code injection compromise an application?	Injection attacks enable you to compromise an application in many ways, including: Causing a denial of service (DoS) of the app Escalating access privileges in the app Exposing and exfiltrating sensitive data in databases such as user credentials and PII Installing malicious software on the server hosting the app Defacing a website
truffleHog	Git secrets search tool. It can automatically crawl through a repository looking for accidental commits of secrets. GitHub secrets allow code commits, this will allow an attacker to modify code in a repository.
OWASP ZAP (Zed Attack Proxy)	Proxy that allows for both automated and manual testing and identification of vulnerabilities. It has many components that allow for different tasks to be performed.
Burp Suite Community Edition	Proxy with a wide range of options to test web applications for different vulnerabilities. Its components allow you to perform particular types of automated testing, manually modifying requests, and passive analysis.
Gobuster	Can discover subdomains, directories, and files by brute-forcing from a list of common names. This can provide information that was otherwise not available.
DirBuster	Web application brute-force finder for directories and files. Comes with 9 different lists, including default directories and common names given by developers. Also allows for brute-force.
w3af	The Web Application Attack and Audit Framework allows you to identify and exploit a large set of web-based vulnerabilities, such as SQL injection and cross-site scripting.
Wapiti	A web application vulnerability scanner which will automatically navigate a webapp looking for areas where it can inject data. Several modules can be enabled/disabled to target different vulnerabilities.
BeEF (Browser Exploit Framework)	Focuses on web browser attacks by assessing the actual security posture of a target by using client-side attack vectors.
WPScan (WordPress Security Scanner)	Automatically gathers data about a WordPress site and compares findings such as plugins against a database of known vulnerabilities. Provides useful information on findings, including plugin version and references to the vulnerability such as CVE number and link.
Brakeman	Static code analysis security tool for Ruby on Rails applications. Checks for vulnerabilities and provides confidence level of finding (high, medium, weak).

SQLmap	SQL Injection scanner tool. Automates several of the attacks and supports many databases. Some of its features include database search, enumeration, and command execution.
SearchSploit	Exploit finder that allows to search through the information found in Exploit-DB. It also supports Nmap outputs in XML format to search for exploits automatically.
CrackMapExec	Post-exploitation tool to identify vulnerabilities in active directory environments.
hook	Connect a browser to another device, usually an attacker's tool or framework, to execute further attacks.
Your team is looking for a tool that can obtain secrets from a GitHub repository. What specific tool would you suggest as being best suited for this purpose?	truffleHog
Your client has a Ruby on Rails application. They want to check for vulnerabilities. Which tool would you suggest they use?	Brakeman
Your PenTest team has accessed an active directory environment. Which post-exploitation tool would you suggest the team use to identify vulnerabilities?	CrackMapExec