Web-Security Essentials

Task 2 : Why Web?

The shift from desktop to web based apps has been going on for decades.

The shift from desktop to web apps brings some advantages at the cost of security concerns.

Web applications are among the most common entry points for attackers because they are always available and exposed. They often connect to back-end systems like databases and other infrastructure, offering attackers high-impact opportunities. A vulnerable web application is often the first stage in a larger attack sequence.

As a Web App Owner	As a Web App User
Your web application is always online and must be secured 24/7	Your data is stored in a web application, potentially insecurely
Anyone around the world can access your app at any time	Once your browser is breached, all of your accounts are at risk
It is challenging to stay up to date with so many emerging threats	A breach can result in identity theft or financial loss
You have the responsibility of securing your users' data	Your privacy can be permanently compromised

In 2017, Equifax’s sensitive customer data of nearly 150 million users got leaked due to Apache’s Vulnerability...by this attackers were able to access interal databases

Capital One faced a similar breach in 2019, in which a misconfigured web app firewall(waf) exposed over 100 million users sensitive personal and financial information.

Task 3 : Web Infrastructure

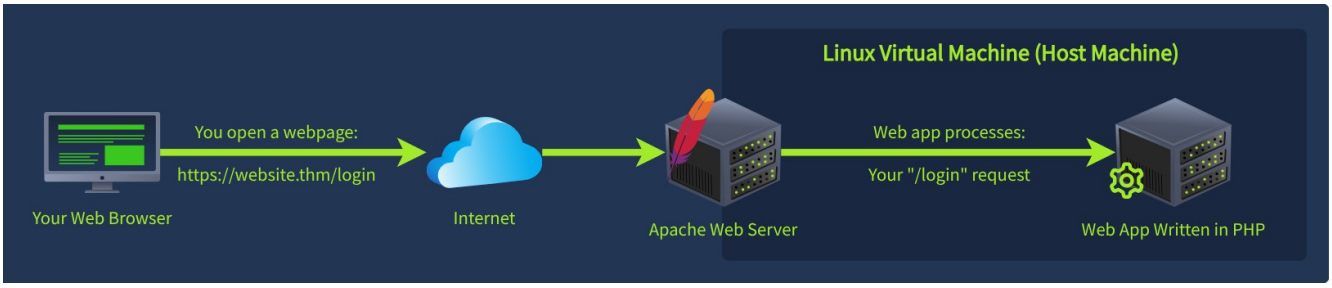
When we visit a Website, the server processes the request, verifies access, and returns a response to the user. The response can be webpage, image or some form of data like search results or account info. This request- response cycle is the foundation of how the website interacts/functions. Attackers can abuse this request-response cycle by overwhelming server with requests, bypassing access controls, or even tricking server into running/executing harmful commands.

There are three components and they are:

Application: The code, icons, images dictates how the website works and looks

Web Server: The component which hosts the application. Listens to the requests and responds(transfer a response to the user)

Host Machine: The underlying OS like Linux or Windows that runs the web server and the applications



WEB SERVERS:

As Discussed above, Web servers listen for requests and responds to them appropriately. They are positioned in front of websites and applications, making them crucial and potential target for Attackers as they are publicly exposed and handle all requests.

Common web servers:

Apache, Nginx,Internet Information Services(IIS)

Task 4 : Protecting the Web

Now lets see how to protect the components above:

Protecting the Application:

Secure Coding:

Avoid insecure functions, ensure proper handling of errors, and remove sensitive information.

Input Validation and Santization:

Validate and sanitize user inputs to prevent injection Attacks.

Access Control:
Restrict Access based on User Roles

Protecting The Web Server:

Logging: Keep a detailed record of all web requests with access logs.

Web App Firewall: Filter and Block harmful traffic based on defined rules.

Content Delivery Network(CDN): Reduce direct exposure to your server and use integrated WAFs.

Protecting the Host Machine:

Least Privilege: Use low-privilege users for service.

System Hardening: Disable unnecessary services and close unused ports.

Antivirus: Add endpoint-level protection that blocks known malware.

For all three:

Strong Authentication: Don't just let anyone access your code, admin panels, or host machine.

Patch Management: Ensure your app dependencies, web server, and host machine are up to date.

Logging:

Web servers can create logs for every request they receive. We call these access logs, and they are incredibly valuable from a security perspective because they track information about every interaction with the server, including the client's IP address, timestamp, requested page or data, response status from the server, and user agent. These fields can play an important role in investigations, helping analysts detect potential malicious activity and trace attacker behavior.

1. The user, from the client IP `10.10.10.100`, visits the website's homepage at `/index.html`.
2. Next, they navigate to the login page at `/login.html`.
3. They then enter their credentials and submit the form, signified by the `POST` request.
4. Finally, they access their account page at `/myaccount.html`.

Although this series of events is expected and not out of the ordinary, you can see how the verbosity of these logs can help analysts and incident responders reconstruct a possible attack sequence.

What cyber security concept involves stopping or limiting damage from threats?

Mitigation

What security control involves ensuring all software and components are up to date?

Patch Management

Task 5 : Defense Systems

Content Delivery Network(CDN):

CDNs store and serve cached content from servers closer to the user to reduce latency. Imagine you have a main server housed in a central location. This main server provides information to edge servers worldwide so your customers can access data more quickly and safely. Aside from speed, CDNs also help in a security sense by acting as a buffer between the user and the origin server.

Security Benefits:

IP Masking: Hides the origin server IP address, which makes it harder for attackers to target.

DDoS Protection: CDNs can absorb a large amount of traffic, making denial-of-service attacks less effective.

Enforced HTTPS: Encrypted communication via TLS is enforced by default by most CDNs.

Integrated WAF: Many CDNs, including Cloudflare CDN, Amazon CloudFront & Azure Front Door, integrate web application firewalls.

Web Application Firewall(WAF):

Types of WAF's:

Cloud-based (Reverse Proxy): Sits in front of the web server. These WAFs are easy to deploy and have great scalability.

Host-based: Software deployed directly on the web server and offers control for each application.

Network-based: A physical or virtual appliance situated on the network perimeter. More suited for enterprise environments.

As stated above, WAFs inspect HTTP requests to detect anomalies, attacks, or known suspicious patterns. Below are some of the methods used, along with examples of requests that may be blocked.

WAF Feature	Detection Method	Example
Signature-Based Detection	Matches known attack patterns or payloads	A request with a User-Agent that matches a known tool, `sqlmap/1.8.1`
Heuristic-Based Detection	Analyzes the context and behavior of requests	A long query string with special characters `search?q=%3Cscript%20(1)`
Anomaly & Behavioral Analysis	Flags deviations from normal traffic behavior	A single IP address makes repeated login attempts in a short period of time
Location & IP Reputation Filtering	Uses location and threat intel to block IPs	A request from an IP address that is outside of your normal business area

Antivirus:

Most AVs rely on signature-based detection, which means they compare files with a database of known malware or patterns.

While web attacks usually target the application layer, not the host machine, AVs still play an important role in host protection, as discussed in Task 3. They can help detect malicious file uploads, such as web shells, post-exploitation tools, and other malicious software.

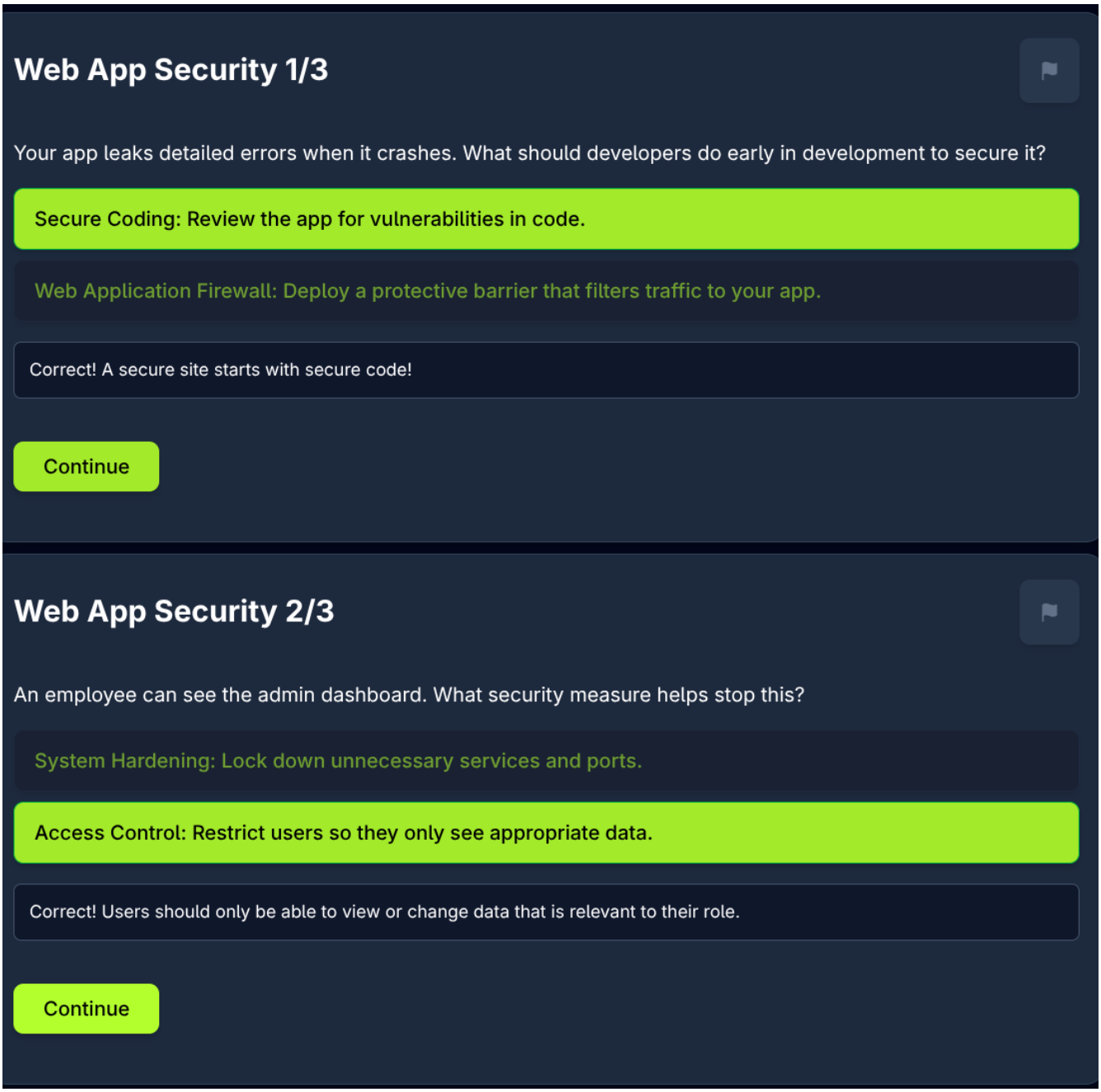
Which type of Web Application Firewall operates by running on the same system as the application itself?

Host-Based

Which common WAF detection technique works by matching incoming requests against known malicious patterns?

Signature-Based

Task 6 : Practice Scenario



Web App Security 3/3

Attackers can inject code into your login form. How do you block it?

Integrate Access Logging: Record web requests so you can investigate suspicious activity later.

Input Validation & Sanitization: Clean and check any user-submitted data before using it.

Correct! It is important to check and clean any data submitted by users to make sure it is safe.

Next Section

Web Server Security 1/3

When configuring your web server, what should you enable so unusual traffic patterns can be investigated later?

Antivirus: Set up endpoint-level protection to block known malware.

Access Logging: Maintain an access log to spot anomalies and support incident response.

Correct! Logs help analysts detect and investigate unusual activity.

Continue

Web Server Security 2/3

How can you reduce your servers' exposure while also speeding up content delivery?

Encryption: Protect sensitive data by encoding it during transmission.

Content Delivery Network: Serve cached content from edge servers to cut latency and improve security.

Correct! CDNs cache and deliver content through edge servers, shielding the origin IP from direct traffic.

Continue

Web Server Security 3/3

Which security measure helps ensure malicious requests never reach your server?

Web Application Firewall: Set up a protective barrier between your users and web server.

System Hardening: Lock down OS and server configurations.

Correct! A WAF filters requests and blocks any that match malicious patterns.

Next Section

Host Machine Security 1/3

Your web server runs with admin rights. What safer setup should you use?

Backups: Regularly copy important data to secure storage.

Least Privilege: Use dedicated low-privilege accounts to run your site.

Correct! Running services with minimal permissions limits attackers if they can gain access.

Continue

Host Machine Security 2/3

Unused ports are open and outdated services are still running. How do you reduce the risk?

System Hardening: Ensure only what you need is running and open.

Secure Coding: Write safe application code to avoid vulnerabilities.

Correct! Hardening limits what is exposed and available for exploitation.

Continue

Host Machine Security 3/3

How can you protect your endpoint from harmful or unauthorized software?

Web Server Logging: Gain visibility into web requests by keeping a detailed record.

Antivirus: Detects and blocks known malware.

Correct! AV tools help catch malicious files and behavior on the host.

Finish