Web Security Essentials

Task 2:- Why Web

The shift from desktop to web-based applications has been ongoing for decades. In the 1990s, desktop applications were the norm because of speed and connectivity limitations. As web technology advanced, the 2000s gave way to much more widely used dynamic web applications for email, social media, and banking. In the 2010s, there was a massive rise in cloud computing and software as a service (SaaS), and today, nearly everything can be done in a browser.

The shift to web apps brings some amazing advantages, including increased accessibility, faster updates, better compatibility, and reduced resource usage on the user's end. Think of it, you can browse online marketplaces and social networks, play games, edit images and video, and even run virtual machines all through your browser. However, these benefits come with trade offs in terms of security. The more powerful and widespread the web becomes, the more opportunities it introduces for attackers.

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

Real-World Examples

In 2017, Equifax's sensitive customer data of nearly 150 million Americans was compromised due to an Apache vulnerability. By abusing this vulnerability, the attackers were able to access internal databases storing valuable customer data.

Capital One faced a similar-scale breach in 2019, in which a misconfigured web application firewall (WAF) exposed over 100 million customers' sensitive personal and financial information. This misconfiguration allowed internal access to the company's cloud infrastructure and databases.

Answer the questions below

1)Have applications shifted from desktop to web over the past couple of decades (Yea/Nay)?

A)Yea

2)Who is ultimately responsible for ensuring the security of users' data within a web application?

A)Web App Owner

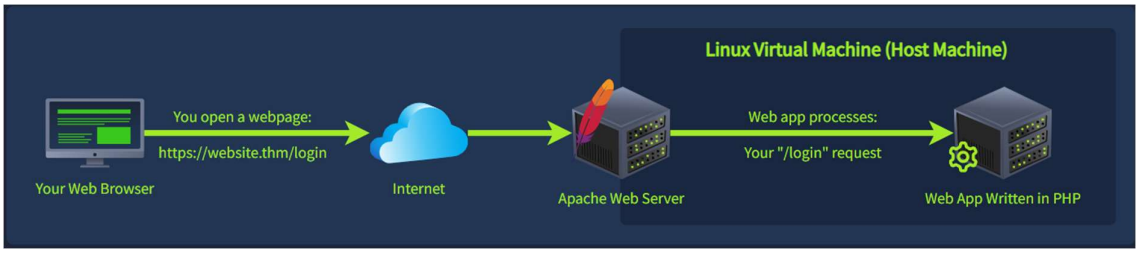
Task 3:- Web Infrastructure

When you visit a website, your browser sends a request to a web server. The server processes the request, verifies access, and returns a response to the user. This response can be a webpage, an image, or data like search results or your account information. This request-response cycle is the foundation of how the web functions. Attackers can abuse this request-response cycle by overwhelming servers with requests, bypassing access controls, or even tricking the server into executing harmful commands.

Components of a Web Service

For example, any web service, like tryhackme.com, requires three main components to function.

- **Application**: The code, images, styles, and icons that dictate how the website looks and functions.
- **Web Server**: This component hosts the application. It listens for requests and returns a response to the user.
- **Host Machine**: The underlying operating system, Linux or Windows, that runs the web server and the application.



Web Servers

When you visit a website, your web browser sends a request to a web server, as discussed above. Web servers listen for incoming requests and return an appropriate response. Web servers are positioned in front of websites and applications, making them a crucial aspect of

the internet's foundation. Because they are publicly exposed and handle all incoming web requests, web servers are a common target for attackers.

Here are some of the most common web servers that you will encounter.

- **Apache**: The most popular web server to host simple websites and blogs, most commonly WordPress.
- **Nginx**: An industry standard for high-performance web apps. Used by companies like Netflix, Airbnb, and GitHub.
- **Internet Information Services (IIS)**: A Microsoft-developed web server commonly used in enterprise environments.

Answer the questions below

1)What does your web browser send to a server to receive a web page?

A)Request

2)What web server is most commonly used to host WordPress websites?

A)Apache

3)What do we call the OS and environment that runs the web server and application?

A)Host Machine

Task 4:- Protecting The Web

Protecting the Application

- Secure Coding: Avoid insecure functions, ensure proper handling of errors, and remove sensitive information.
- Input Validation & Sanitization: Validate and sanitize user input 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 Application Firewall (WAF): 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 services.
- System Hardening: Disable unnecessary services and close unused ports.
- Antivirus: Add endpoint-level protection that blocks known malware.

Security Tips for All Three Components

- 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 behaviour.

Let's take a look at a benign series of events that we might find in an access log to get a feel for the type of data we can observe.
Note that GET requests are used to retrieve a resource from the server, like a specific web page.
POST requests are used to submit data to the server, such as login credentials.

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.

Answer the questions below

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

A)Mitigation

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

A)Patch Management

Task 5:- Defense Systems

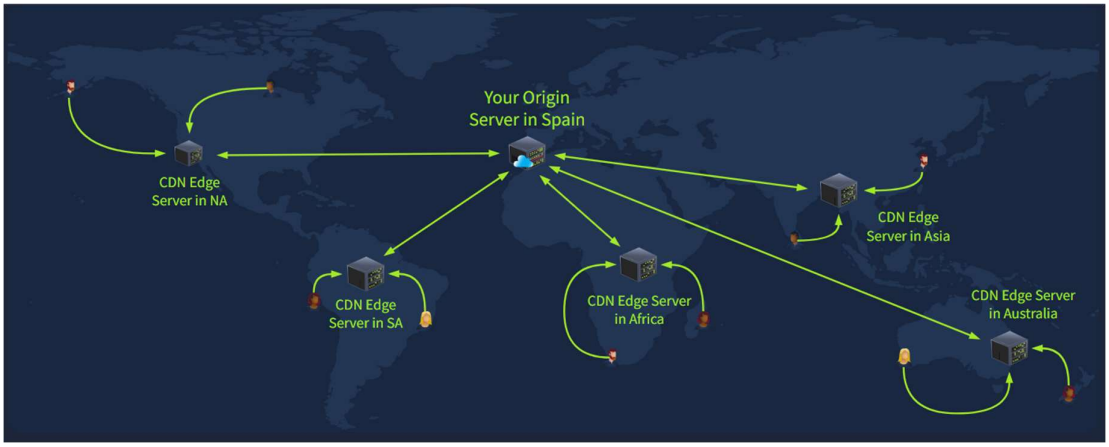
Content Delivery Network (CDN)

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.

In essence, CDNs allow web apps to deliver data to customers more efficiently and securely.



Web Application Firewall (WAF)

WAFs are a powerful tool that can be integrated as another layer of protection for websites and web applications. They inspect incoming HTTP traffic and block or log potentially harmful requests based on security rules. Think of the analogy of a bouncer at a bar or club. Every person (web request) that wants to enter must be checked by the bouncer (firewall). Anyone (any request) that doesn't meet the standard requirement will be rejected.

Let's take a closer look at the types of WAFs available to us as defenders, then dive deeper into their functionality.

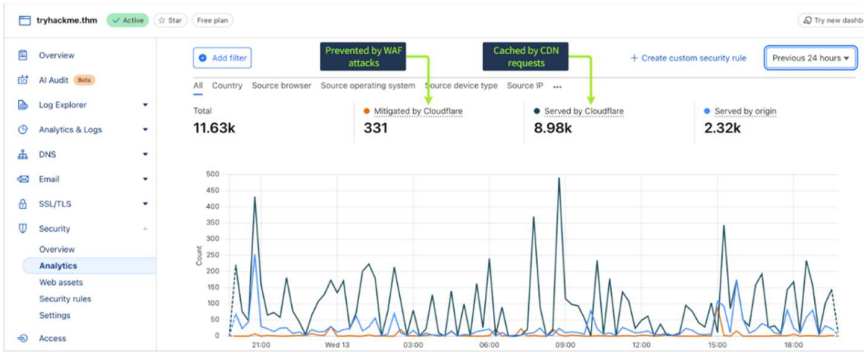
- Cloud-based (Reverse Proxy): Sits in front of the web server. These WAFs are easy to deploy and have great scalability.
- Host-based: Software deployed directly on the web server and offers control for each application.
- Network-based: A physical or virtual appliance situated on the network perimeter. More suited for enterprise environments.

Functionality

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

Below is a screenshot of the Cloudflare dashboard for tryhackme.thm, focused on the security panel. In it, we can see all requests for the last 24 hours, including requests blocked by the integrated web application firewall.



Antivirus (AV)

AVs are often misunderstood as a blanket protection measure, but they are primarily made to safeguard endpoints, such as desktops, laptops, and servers, from known malicious files and programs. 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. AVs are just one layer in a broader defense-in-depth strategy and should be combined with other security measures to provide stronger protection.

Answer the questions below

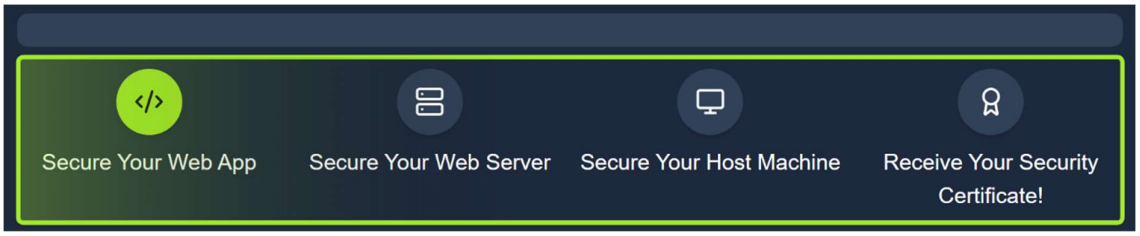
1)Which type of Web Application Firewall operates by running on the same system as the application itself?

A)Host-Based

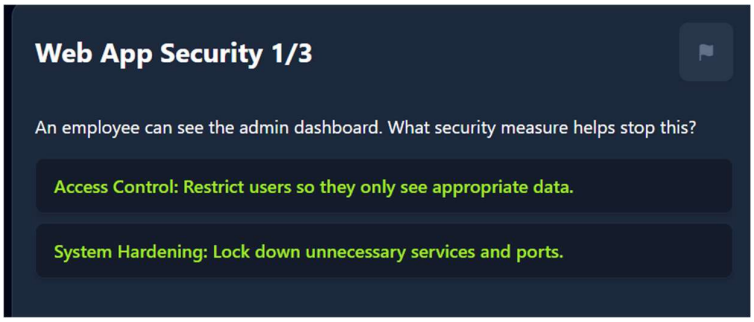
2)Which common WAF detection technique works by matching incoming requests against known malicious patterns?

A)Signature-Based

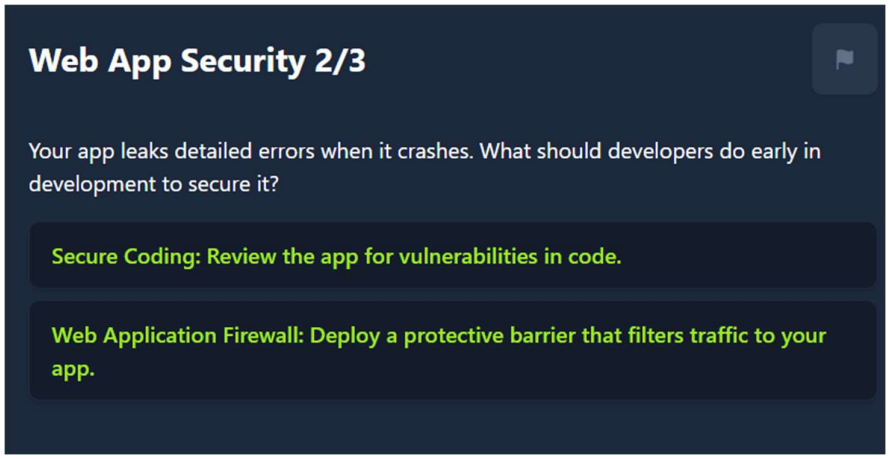
Task 6:- Practice Scenario



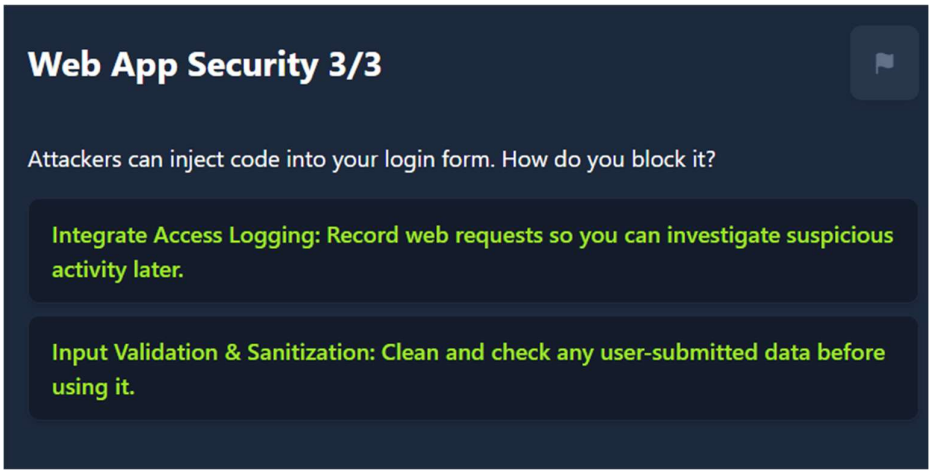
Step 1:-



Step 2:-



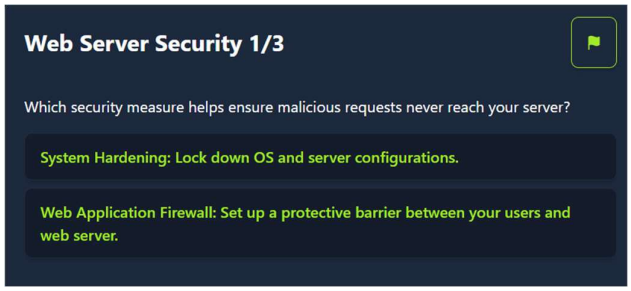
Step 3:-



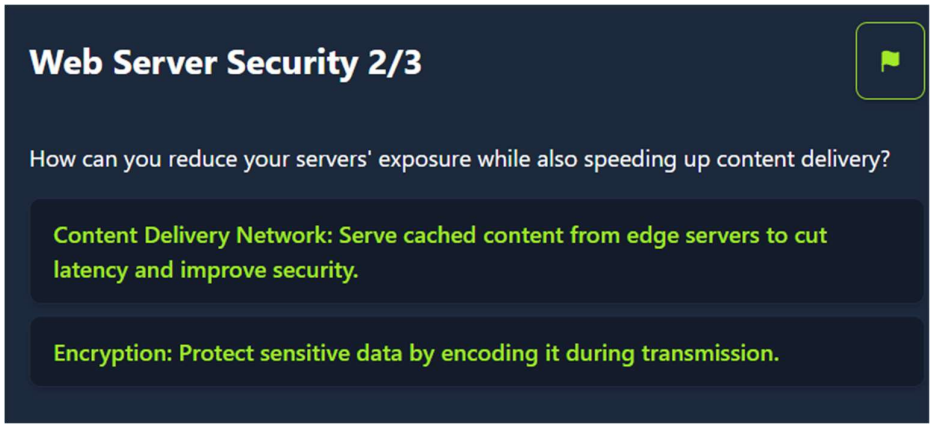
Step 4:-

THM{web_app_secured!}

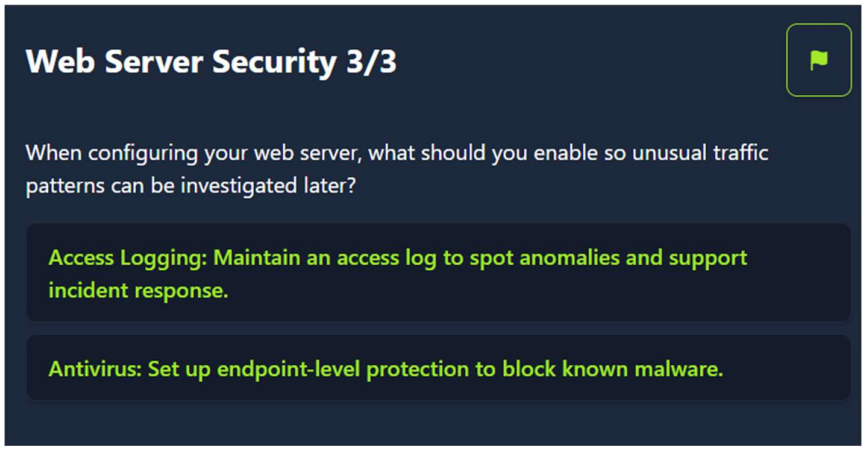
Step 5:-



Step 6:-



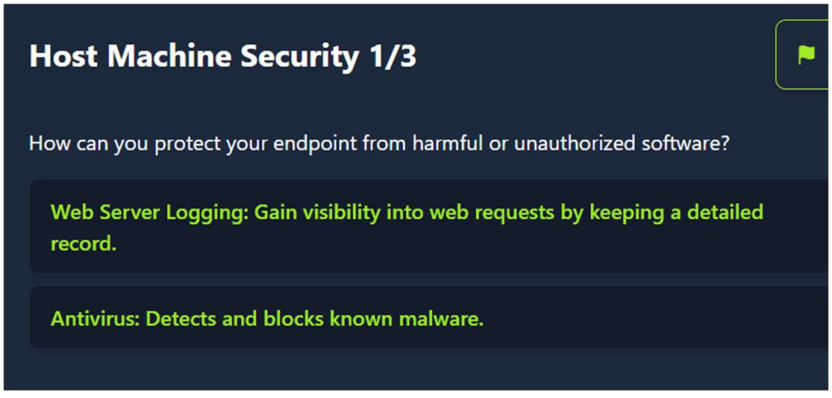
Step 7:-



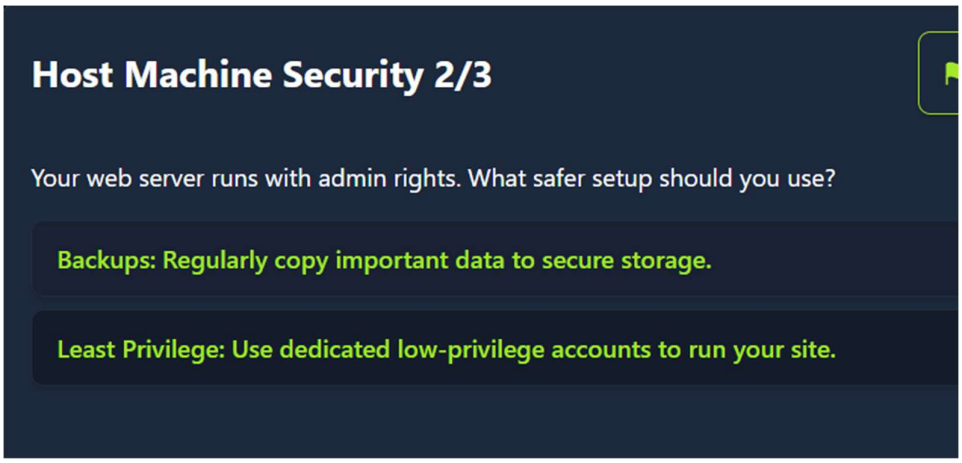
Step 8:-

THM{server_security_expert!}

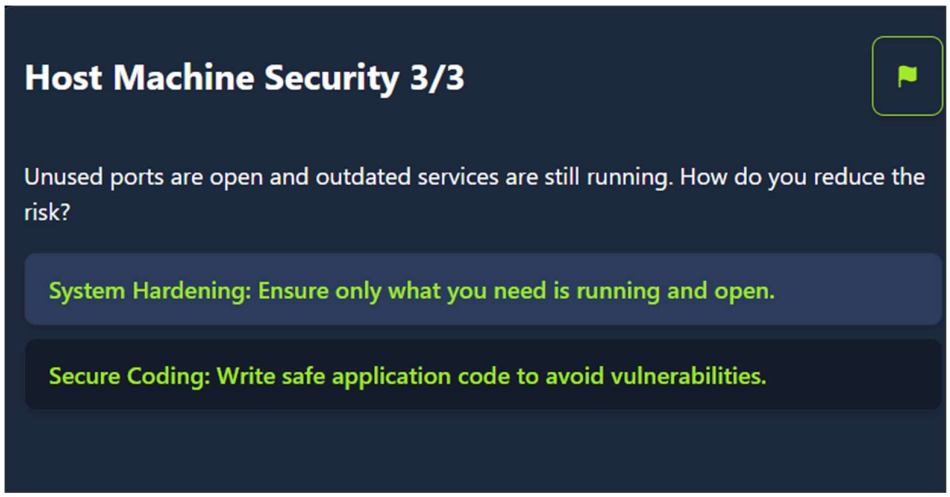
Step 9:-



Step 10:-



Step 11:-



Step 12:-

THM{the_final_security_layer!}

Answer the questions below

1)What flag did you receive for securing the Web Application?

A)THM{web_app_secured!}

2)What flag did you receive for securing the Web Server?

A)THM{server_security_expert!}

3)What flag did you receive for securing the Host Machine?

A)THM{the_final_security_layer!}