ISEC2700 Glossary

*Kyle Walker*

*ISEC2700*

*09/24/2024*

Table of Contents

[Assignment 2: Research and Recommendation Checklist 2](#_Toc179887968)

[Question 1: 5](#_Toc179887969)

[Question 2: 6](#_Toc179887970)

[Question 3: 6](#_Toc179887971)

[Question 4: 7](#_Toc179887972)

[Assignment 3: Research and Recommendation Checklist 8](#_Toc179887973)

# Assignment 2: Research and Recommendation Checklist

1. **Web Server:**

A web server is hardware or software (in the case of VMs) that serves web content to users over the Internet. Apache HTTP is a popular example of a common web server.

1. **Firewall Server:**

A firewall server is a network device that controls traffic coming in and out of the network based on certain rules. pfSense is an example of a popular, open-source firewall server.

1. **TCP/IP Stack**

The TCP/IP model is a set of network protocols that helps deliver data across networks. It is made up of 4 layers: Application, Transport, Internet and Network interfaces. For example, HTTP operates at the Application layer, while TCP works at the Transport layer.

1. **SSH Keys**

SSH keys are used to authenticate users for secure, remote access to servers. There is both a private key and public key to access. A few security risks associated with SSH keys are theft of a private key, usually through a computer virus or malware, poor encryption algorithms, leading to easier brute force hacking, and misconfiguration, leaving the keys open for unauthorized users.

1. **DMZ (Demilitarized Zone)**

A DMZ acts like a barrier between two networks. Keeping an internal network safe on one side, and an untrusted network (like the internet) on the other side. A DMZ typically hosts things like a web server to access the internet, but also keeps nodes within the network safe.

1. **LAMP/WAMP/MAMP/XAMMP**

These are all acronyms for different software stacks related to web development. LAMP stands for Linux, Apache, MySQL, PHP. WAMP stands for Windows, Apache, MySQL, PHP, MAMP stands for macOS, Apache, MySQL, PHP and ZAMPP stands for cross-platform, Apache, MySQL, PHP, Perl.

1. **Cross Site Tracing Attack**

Cross Site Tracing (XST) is an attack that targets the HTTP TRACE method to pull server information from a website. Someone could potentially send a fake request to a server that would return sensitive information if the TRACE method is not disabled on the server.

1. **Apache ServerTokens and ServerSignature**

In Apache, the ServerTokens and ServerSignatures control information displayed in headers and errors from web servers. With these settings enabled, sensitive information can be shown on certain error pages when trying to access webpages hosted on a server.

**Identify your top 10 recommended security configurations for an IIS Web Server:[[1]](#footnote-1)[[2]](#footnote-2)**

1. **Set Up and Force HTTPS:** I strongly recommend configuring the server to reroute HTTP traffic to HTTPS. It’s hard to find a webpage nowadays that doesn’t adhere to the HTTPS protocol, so forcing traffic to use it is a no-brainer for me. This prevents any “eavesdropping” and almost guarantees data integrity.
2. **Keep Things Up to Date:** Security issues and attacks happen fast and often. I think the most important security step for most machines is keeping them up to date with the latest security. It’s well-known that attackers aim for older networks and machines to exploit.
3. **Use DDoS Protection:** A Web Application Firewall (WAF) can help prevent many common web-based strikes like DDoS attacks.
4. **Start Logging:** Use things like Windows Event Viewer to help catch unusual activity on the network, helping recognize and stop attacks sooner.
5. **Manage Permissions:** Simply restricting access to files to only the people who are meant to use/edit/read them is one of the most basic security practices on any machine, including web servers. This prevents anyone without permission from accessing the web apps.
6. **Use Request Filtering:** Block certain files and requests, like TRACE and TRACK mentioned previously, to prevent attacks from exploiting vulnerabilities and gaining access.
7. **Remove Unneeded Services:** Things like open ports and FTP protocols can leave holes in your network, especially if they’re left on and aren’t being used. Disabling things like this is a way to keep on eye on the things that matter, and not on what doesn’t.
8. **Use HTTP Security Headers:** Use headers like “Strict-Transport-Security” and “Content-Security-Policy” to help prevent attacks like cross-site scripting and clickjacking.
9. **Limit IP Access:** Use services like Cloudflare to help determine that IP addresses accessing your network are legitimate and ensuring you know who is accessing your server at any time.
10. **Authentication:** Use things like Windows Authentication for basic authentication. Like permissions, this helps ensures that users can only access what they’re meant to.

**Identify your top 10 recommended security configurations for an Apache Web Server:**

1. **Set Up and Force HTTPS:** My reasoning for having HTTPS at the top of my list for Apache is the same as IIS and implementation is very similar for both. Encrypting data in transit is always very important.
2. **Install ModSecurity:** The benefits of using an open-source Linux server, is having access to a powerful, free firewall as well. ModSecurity helps prevent against common exploits like SQL injections and filtering HTTP traffic.
3. **Update:** There will be some similarities between the two lists here. Being able to quickly update your network for Day Zero attacks and stay on top of developing exploits is paramount.
4. **Limit HTTP Methods:** Being able to quickly and simply prevent attacks on TRACE and TRACK methods are too easy to not implement, especially when the information your protecting is so sensitive (IP, OS, etc.)
5. **Logging:** Different from IIS, but still just as important. Instead of using Windows Event Viewer, something like the CustomLog directive can achieve the same result.
6. **DDoS Protection:** Slightly lower on my list than IIS because of ModSecurity, but still important. Apache can take some extra precautions against DDoS attacks with the “mod\_evasive” and “mod\_reqtimeout” directive to limit resource usage.
7. **Restrict Access:** Like managing permissions with IIS, with Apache, we would be using “.htaccess” to block access to config files.
8. **Authentication:** Going together with permissions and restricting access, adding Basic or Digest authentication on sensitive areas of the server helps divide data into sensitive and non-sensitive.
9. **Directory Listing:** In Apache, it’s possible for users to view the contents of a directory without an index file, so setting “Options -Indexes” in the configuration file is a quick way to prevent this.
10. **Security Headers:** Exactly the same system and integration method as IIS, security headers like “Content-Security-Policy” and “Strict-Transport-Security” help prevent XSS attacks and clickjacking.

**Describe your 4-5 recommend server logging recommendations for both an IIS and Apache web server. (4-5 logging recommendations PER server)**

**IIS Web Server:**

1. **Detailed Logging:** This one speaks for itself. Detailed logging includes things like timestamps, client IP and user agent among many others in its logs. Exactly the kind of information someone would need to quickly prevent an attack.
2. **Monitor and Alert:** Enable monitoring tools that create reports and alert administrators to things like repeated login attempts. Personally, I think the speed in which an attack can be identified is one of the most important parts of web security.
3. **Errors:** If an attacker is targeting a vulnerability, there’s a good chance an error report could point out the vulnerability before the attacker even has access to the network.
4. **Centralized Logging:** I discovered some interesting tools like ELK Stack and Splunk that keep all the logs for multiple servers in one place, so monitoring doesn’t have to be done on each specific node and instead can be done for the entire network at once. This seems invaluable to me.
5. **Log Retention Policies:** Keeping logs is important, but keeping too many can bog down your servers and bloat your storage devices. Setting up policies that delete logs after a certain amount of time automatically is a good way to solve this.

# Question 1:

*Apache separates all informational messages into categories depending on how important it considers the information. The Log Level setting is stored in****etc/apache2/apache2.conf****file on an Ubuntu server. What are the available Apache LogLevels that “Control the severity of messages logged to the error\_log” and what do they do? Review your apache2.conf file to help complete the following.*

|  |  |  |
| --- | --- | --- |
| *Level* | *Description* | *Example* |
|  |  |  |
|  |  |  |
|  |  |  |
| *1 – Debug* | *Logs all messages, including debugging information* | *Useful for developers, to spot problems exactly where and when they occur* |
| *2 – Info* | *Logs all informational messages* | *Startup messages, processing requests* |
| *3 – Notice* | *Logs significant events* | *Server startup, shutdown* |
| *4 – Warn* | *Logs potentially harmful events* | *Configuration file changes* |
| *5 – Error* | *Logs requests that cannot complete* | *“Error 404”, website/file not found* |
| *6 – Crit* | *Logs critical events* | *Hardware failures* |
| *7 – Alert* | *Logs events that need immediate attention* | *An entire service not functioning* |
| *8 – Emerg* | *Logs events that make the server completely unusable* | *A server crash* |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Question 2:

*What is the Apache Log level set by default in your apache2.conf?*

By default, it is set to “LogLevel warn”.

# Question 3:

*What are the TCP/IP request default settings in /etc/apache2/apache2.conf and what do they mean? You may review your apache2.conf file to complete the following.*

|  |  |  |
| --- | --- | --- |
| Name | Apache2.conf request Setting | Description of setting |
| Timeout | 300 | The number of seconds before receives and sends time out |
| KeepAlive | On | Whether or not to allow persistent connections (more than one request per connection). Set to “Off” to deactivate. |
| MaxKeepAliveRequests | 100 | The maximum number of requests to allow during a persistent connection. Set to 0 to allow an unlimited amount. We recommend you leave this number high, for maximum performance. |
| KeepAliveTimeout | 5 | Number of seconds to wait for the next request from the same client on the same connection. |

# Question 4:

*ServerTokens can have various values which will have different outputs to the HTTP headers. What are the****6 ServerToken Directive options****available and what do they display?*

|  |  |
| --- | --- |
| **ServerToken Directive** | **Displayed response header field** |
|  |  |
| ServerTokens Full (or not specified) | Server: Apache/2.4.2 (Unix) PHP/4.2.2 MyMod/1.2 |
| ServerTokens Prod | Server: Apache |
| ServerTokens Major | Server: Apache/2 |
| ServerTokens Minor | Server: Apache/2.4 |
| ServerTokens Min | Server: Apache/2.4.2 |
| ServerTokens OS | Server: Apache/2.4.2 (Unix) |
|  |  |
|  |  |
|  |  |
|  |  |

# Assignment 3: Research and Recommendation Checklist

1. **Application Detection such as OpenAppID**

Application Detection is a way of detecting certain applications being run on a network and therefor being able to block them, limit their bandwidth or otherwise control the traffic from specific sources. (ScienceDirect, n.d.)

OpenAppID from Cisco is an application-layer security plug-in for Snort capable of Application Detection for servers. (Snort)

1. **IDS (in reference to security servers)**

IDS stands for Intrusion Detection System, which alerts server administrators to any vulnerability exploits and traffic on a server. There are five different types of IDS, but the most common are NIDS (network-based) and HIDS (host-based). The NIDS runs on the server itself to detect traffic while the HIDS runs on end-user machines. (PaloaltoNetworks)

1. **IPS (in reference to security servers)**

IPS stands for Intrusion Prevention System and works similarly to an IDS but can actively defend against attacks. An IPS is usually part of the network itself, acting similarly to a firewall. Like an IDS, there are several different types, but the common ones are HIPS and NIPS, being host-based and network-based as well. (PaloaltoNetworks)

1. **SNORT (Network Intrusion Detection & Prevention System)**

SNORT is another example of an NIPS. It can be used as an IDS and/or an IPS, generating alerts for administrators based on traffic or blocking threats altogether. (Snort)

1. **Virtual Private Network, how do they work and why have them?**

VPNs work by creating a connection between you and a secure server to protect your data, traffic and IP addresses. A VPN adds a level of security to everyday browsing, but it also allows employees to access company data in a safe and secure manner when working remotely.

# Assignment 4:

1. **Proxy Server. What is this and what are the advantages and disadvantages is using a Proxy?**

* A proxy server is a device on a network that acts as a gateway to the internet with its own separate IP to protect other devices on the private network. (Fortinet, n.d.)

It aids in internet anonymity but has some security disadvantages. Although anonymous, your traffic is not encrypted and traffic going into the server cannot be controlled.

1. **Reverse Proxy Server. What is this and what are the advantages and disadvantages is using a Reverse Proxy?**

* A *forward* proxy server mentioned above sits on a private network and masks the IP of the client interacting with the Internet, while a reverse proxy instead sits on the receiving network. Usually in front of a web server and instead masks the IP of the web server instead. (CloudFlare, n.d.)

Being the same as a forward proxy server, just placed in a different place, it has the same advantages and disadvantages as any proxy server. It masks the IP of the server but still does not fully protect it from any attacks without any encryption.

1. **Transparent Proxy. What is this and what are the advantages and disadvantages is using a Transparent Proxy?**

* A transparent proxy is placed on a private network and is usually used as a web filter to prevent access to unauthorized websites. However, this can also be used to prevent DDoS attacks by detecting suspicious incoming activity, as the process can work both ways.

A transparent proxy can route traffic through a firewall without the sender of the data knowing, aiding in quietly monitoring incoming and outgoing data. However, it creates some security issues of its own. Because all the user data on the private network will be passing through it, hackers could break into the transparent proxy and obtain all of the networks outgoing traffic. (Fortinet, n.d.)

1. **Proxy Port, what is the default port? Would you change this port and if so why otherwise, why not?**

* A proxy port is the port on the proxy server that all incoming and outgoing traffic passes through for the network. Common default ports are usually HTTP (8080, 80, 8008), HTTPS (443), SOCKS (1080, 1081) and Squid (3128). (BrowserStack, n.d.)

1. **Google Safe Browsing. What is it and what resources are required to configure it? Would you recommend using Google Safe Browsing and why or why not?**

* Google Safe Browsing is a service offered by Google who will identify unsafe websites and notify the users and website owners of any risks. (Google, n.d.)

I see this service as a win-win for both users and website owners seeing as this service is run outside of the networks. The service is not taking up bandwidth or adding any third-party programs or services to potentially confidential servers and instead acting off of known security breaches, so I don’t see the harm.

1. **How does radius authentication work?**

* Radius authentication works by verifying users against a database. Instead of having user credentials sent directly to the database where the passwords and other critical information is stored, RADIUS acts as a barrier between the two.

1. (Microsoft) [↑](#footnote-ref-1)
2. (National Institute of Standards and Technology) [↑](#footnote-ref-2)