HyperionDev

Server Security

Task

Visit our website

Introduction

Server configuration is a crucial part of cyber security and is important for protecting a system from security threats. Firewall rules help to filter incoming and outgoing traffic to ensure only authorised services are allowed. Access controls can help to prevent unauthorised users from accessing sensitive data. Identifying and deleting vulnerabilities is also important, as well as keeping the software, applications, and system up to date with the latest security patches. In this task, you will be exposed to the most important concepts in server security, including server configuration, why it is important, and real-world applications.

Server configuration

Cyber security server configuration refers to the security measures and settings that are put in place on a server to protect it from cyber attacks.

There are many industry standards that help to guide IT administrators on how to securely configure their servers to maximise their security. The most common include:

1. **NIST cyber security framework**
2. **CIS**
3. **NCSC**
4. **Cyber Essentials**
5. **PCI**
6. **ISO 27002**
7. **ISO 27001**

Let's explore the components of server configuration in more detail.

Secure by design

As a programmer, it is important to understand the concept of **secure by design**. This means that security should be built into the design and development of a system from the start rather than trying to add it on afterwards. This can involve things like using secure coding practices, implementing access controls and authentication mechanisms, and following industry standards and best practices. By considering security at every stage of the development process, you can help to ensure that the systems you create are less vulnerable to threats and attacks. It is also important to

regularly review and update your systems to address any new security issues that may arise.

In most organisations, security is still considered a ‘bolt-on’ that is applied after a solution is built. This creates problems and reduces the security posture of the solution. Building applications with a secure-by-design mindset from the outset will help to avoid vulnerabilities once the service goes live.

Configuration management (assets)

When building applications, it’s important to understand the type of data being processed and the devices and users that will access the data. For example, building an application intended to process payments or personnel data will require tighter controls than might be necessary when building an application intended to publish a public event calendar.

Most organisations utilise **managed endpoints**, which is where the company controls the PC, laptop, or mobile phone from a central management system. This enables them to ensure the configuration of the device that accesses the company server is as secure as possible.

Bring your own device (BYOD) has become more popular in recent years, which has presented a security challenge, as these devices are not managed. Most organisations severely limit which resources (servers) a BYOD endpoint can access.

Any services that are open to the general public require a significantly more stringent security configuration, since these are most likely to be attacked.

Internal and external threats

Internal cyber security threats come from within an organisation, such as from employees or contractors. Examples of internal threats could include employees sharing sensitive information on unsecured networks or accidentally downloading malware from a malicious website.

External cyber security threats come from outside an organisation. These could include things like hacking attempts, malware infections, or phishing scams.

For instance, an external attacker may try to gain access to an organisation's network by sending fake emails to employees to trick them into revealing their login credentials.

Internal and external threats can be equally dangerous, and organisations must be prepared to defend against both. To address these threats with server configuration, an IT administrator can implement the following measures:

1. Installing and configuring a firewall to control incoming and outgoing network traffic and prevent unauthorised access.
2. Enforcing strong password policies, such as requiring complex and unique passwords for each user, and regularly updating and rotating these passwords.
3. Disabling unnecessary services and features on the server, such as those that are not being used or are known to be vulnerable to attack.
4. Regularly applying software updates and patches to fix known vulnerabilities and protect against new threats.
5. Implementing access controls to limit which users and devices have access to the server and what actions they can perform.
6. Encrypting sensitive data to protect it from unauthorised access, even if an attacker somehow intercepts it.
7. Regularly monitoring the server for any signs of suspicious activity or potential security breaches and responding quickly to address any identified issues.
8. Implementing **multi-factor authentication (MFA)**, which is a security measure that requires users to provide multiple pieces of evidence (or 'factors') to verify their identity before being granted access to a system or service. For instance:
    a. Something the user knows, like a password or a security question.
    b. Something the user has, like a physical token or a smartphone that can receive a **one-time password (OTP)** via SMS or a mobile app.
    c. Something the user is, such as a fingerprint or a face scan, that can be used for biometric authentication.

Malware (ransomware, viruses, and Trojans)

You will recall from a previous task in the bootcamp that malware is short for malicious software, and refers to any software that is specifically designed to harm or exploit a computer or network. There are many different types of malware, including viruses, ransomware, and Trojans. Let's recap briefly:

- A **virus** is a type of malware that replicates itself by attaching to other programs or files. Once a virus has infected a computer, it can spread to other computers on the same network, causing damage to files and programs.
- **Ransomware** is a type of malware that encrypts a victim's files and demands a ransom payment to decrypt them. Ransomware attacks can be particularly damaging for organisations, as they can result in the loss of important data and disrupt business operations.
- **Trojans** (also called Trojan Horses) are a type of malware that disguises itself as legitimate software to trick users into installing it. Once installed, a Trojan can give an attacker access to a victim's computer, allowing them to steal sensitive information or perform other malicious actions.

Information security pillars

This section acts as a refresher, but with respect to the Server Security task.

Confidentiality

The information security pillar of confidentiality refers to the protection of sensitive information from unauthorised access or disclosure. This means ensuring that only authorised individuals or systems can access sensitive data, and that the data is only used for the purposes for which it was intended.

There are several ways to improve a server's configuration to maintain confidentiality, including **implementing access controls, encrypting data,** and properly **disposing of sensitive information** when it is no longer needed. Ensuring confidentiality is important for protecting the privacy of individuals and organisations, as well as maintaining the security of critical assets.

Integrity

The information security pillar of integrity refers to the preservation of the accuracy and completeness of data and systems. This means ensuring that information cannot be modified or corrupted by unauthorised individuals or systems, and that it is always possible to trust the accuracy and reliability of the data.

There are several ways to improve a server's configuration to maintain integrity, including implementing a **security baseline** and **monitoring deviations** from this **baseline**, which could indicate a breach. In addition, it's important to regularly **backup** and test data, and **monitor systems for unusual activity**. Ensuring integrity is important for maintaining the reliability and trustworthiness of information and systems, as well as protecting against threats like malware and data corruption.

Availability

The information security pillar of availability refers to the ability of authorised individuals or systems to access data and systems when needed. This means ensuring that information and systems are always available for use, and that authorised users can access them without interruption or delay.

There are several ways to improve a server's configuration to maintain availability, such as implementing **redundant systems** so that if one fails, another will 'kick in' without resulting in an outage to the end user. It is also important to establish **backup plans,** monitoring for and responding to outages or failures, and regularly **testing** and **maintaining** systems to ensure they are functioning properly. Ensuring availability is important for maintaining the usability and reliability of information and systems, as well as protecting against threats like denial-of-service attacks.

Secure defaults

Default versus hardened configuration

The default configuration of a server is the set of settings and configurations that are applied when the server is first installed or set up. This default configuration is designed to provide a basic level of functionality and security, but may only be sufficient for some environments or use cases.

A hardened configuration, on the other hand, is a server configuration that has been specifically designed to provide a higher level of security. This can involve disabling or removing unnecessary services and applications, implementing stricter access controls, and applying security patches and updates.

CIS benchmark level (baselining)

CIS (Centre for Internet Security) benchmarking compares the configuration of a server or other system against a set of recommended security settings. The CIS benchmarks provide detailed guidance on configuring systems securely, covering a wide range of security-related topics.

Using CIS benchmarks can help organisations ensure that their systems are correctly configured and secured. By comparing the configuration of a system against the recommended settings in the benchmarks, organisations can identify any potential security vulnerabilities or gaps in their configuration. This can help them make informed decisions about how to improve the security of their systems and protect against threats.

Qualys Security Configuration Assessment (SCA) is a tool that can measure the level of compliance of a server's configuration against the CIS benchmarks. Using Qualys SCA, organisations can scan their servers and compare their configuration against the CIS benchmarks to determine if they are compliant or if any deviations from the expected configuration exist. This information can then be used to identify and address any security vulnerabilities or compliance issues, and ensure that the organisation's servers are secure and compliant.

When attempting to improve an organisation’s security posture, you will conduct a posture review using NIST or Cyber Essentials, for example, which will highlight areas of improvement. A remediation plan is then formulated to tackle the issues in order of priority. A CIS benchmark level is useful, because it enables you to capture a snapshot of the current server configuration and associated vulnerabilities, and then once you have made changes to the server configuration, you can re-run the test to see if you are compliant with the relevant benchmark level.

Principles and strategies

Organisations should consider the following principles and strategies to take a proactive approach to cyber security.

Zero trust

The principle of zero trust is a cyber security principle that assumes that all users and systems are potentially untrusted, and that no user or system should be automatically trusted. This means that all users and systems, regardless of whether they are internal or external, should be subject to the same level of scrutiny and verification before they are granted access to sensitive information or resources.

To implement a zero-trust approach, organisations can use a variety of security measures, such as MFA, access controls, and network segmentation. By applying these measures consistently and rigorously, organisations can help to ensure that only authorised users and systems can access sensitive information and resources.

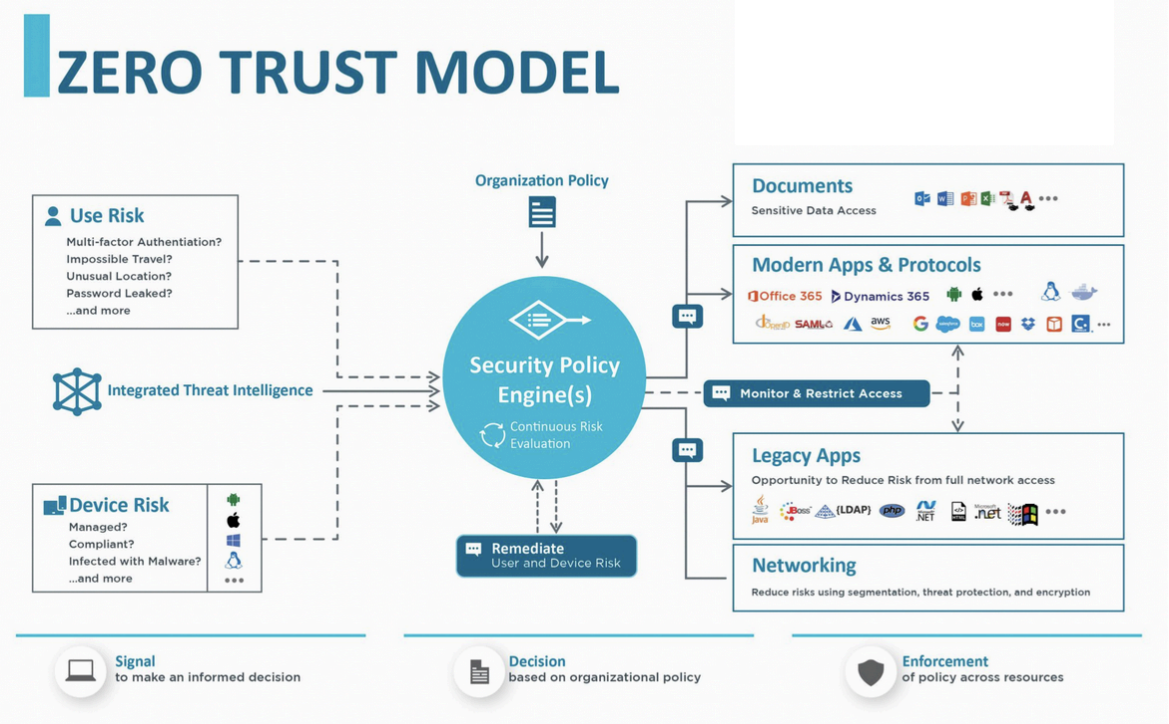

Diagram (Proveho Networks, 2020)

Principle of least privilege

An important component of zero trust is the principle of least privilege. This is a fundamental principle of cyber security that states that individuals and systems should only have the minimum level of access necessary to perform their duties. This means that users should only be given access to the specific resources and information they need to do their jobs, and should not be able to access other resources or information unless necessary.

The principle of least privilege is closely related to the concept of separation of duties, which is the practice of dividing responsibilities among different individuals or groups to prevent unauthorised access or actions. For example, an organisation might separate the duties of creating and approving transactions to prevent fraud.

The principle of least privilege is particularly relevant to programming, because it involves designing systems and applications in a way that ensures that users and systems only have the minimum level of access necessary to perform their functions. This can involve things like implementing access controls, using secure authentication mechanisms, and following industry standards and best practices.

From a programming perspective, implementing the principle of least privilege can involve designing systems and applications in a modular and flexible way that allows for fine-grained control over access and privileges. This can involve using techniques like **role-based access control (RBAC)**, which allows different users to have different levels of access based on their roles within the organisation.

Principle of least privilege in action

Secure Shell (SSH) is a network protocol that allows users to securely connect to and manage remote computers over the internet. Instead of using a password to authenticate, SSH uses cryptographic keys to identify and authenticate users securely.

To use SSH and SSH keys, users first need to generate a public/private key pair. The private key is kept on the user's local computer, while the public key is copied to the remote server the user wants to access. When the user attempts to connect to the server, their private key is used to authenticate their identity. If the server recognises the key, the user is granted access to the server.

Using SSH and SSH keys offers several advantages over using passwords for authentication. Because the keys are generated using cryptography, they are much more difficult for an attacker to guess or crack. Additionally, because the keys are stored on the user's local computer, they are not vulnerable to the same kinds of attacks that can compromise passwords, such as phishing or social engineering.

Defence in depth

Cyber security defence in depth is a strategy that involves implementing multiple layers of security to protect against threats and attacks. This can include things like segmentation (separating servers from users), access monitoring, and regular security training for employees. By using a defence-in-depth approach, organisations can help to reduce the risk of security breaches and protect against a wide range of threats.

Fail securely

The principle of failing securely is a cyber security principle that states that systems and applications should be designed to fail in a safe and controlled manner. This means that if a system or application encounters an error or failure, it should fail in a way that prevents or minimises the potential damage or impact.

There are several ways that systems and applications can be designed to fail securely. For example, systems can be designed to automatically switch to a backup or redundant system in the event of a failure or to shut down gracefully to prevent further damage. Applications can be designed to validate input and handle unexpected conditions in a way that prevents or minimises the potential impact of those conditions.

Minimise attack surface

The principle of minimising the attack surface is a cyber security principle that involves reducing the number of potential entry points or vulnerabilities that attackers can use to gain access to a system or network. This can involve things like disabling or removing unnecessary services and applications, applying security patches and updates, and implementing access controls.

By minimising the attack surface of a system or network, organisations can help to reduce the risk of security breaches and protect against a wide range of threats. This can involve conducting regular security assessments to identify and address potential vulnerabilities, and implementing security controls to prevent unauthorised access.

Server Security and Modern System Hardening

Modern server environments face increasingly sophisticated attacks that target not only misconfigurations but also the underlying system architecture, authentication mechanisms, virtualisation layers, and supply chain components. Beyond the foundational principles already covered, organisations rely on advanced strategies that improve visibility, reduce attacker persistence, enforce system integrity, and strengthen trust across all layers of the server ecosystem.

These advanced practices ensure that servers remain secure even when attackers bypass or evade basic controls.

Kernel-Level Protection and Secure Boot Mechanisms

As attackers develop more advanced rootkits and kernel-level exploits, organisations must protect the lowest layers of the operating system. Kernel hardening prevents unauthorised modification of system components, making it significantly harder for attackers to maintain persistence or escalate privileges.

Modern approaches include:

- **Secure Boot**, which ensures only trusted, cryptographically signed components are loaded during the boot process.
- **TPM-backed integrity checks**, which verify that critical boot files have not been tampered with.
- **Kernel lockdown modes**, which restrict even privileged users from accessing sensitive kernel interfaces.
- **Limiting kernel modules**, reducing the number of modules that can be dynamically loaded and potentially exploited.

By reinforcing the operating system at the kernel level, organisations significantly reduce the risk of stealthy, long-term compromise.

File Integrity Monitoring and Tamper Detection

Once an attacker gains access to a system, one of their first goals is often to modify configuration files, system binaries, or log files to hide their tracks. File Integrity Monitoring (FIM) helps organisations detect these changes quickly.

Advanced FIM tools (such as AIDE, Wazuh, or Tripwire) continuously monitor:

- Critical system binaries
- Authentication logs
- Configuration files
- Cron jobs and scheduled tasks
- Installed packages
- Important directories

If an unauthorised modification occurs, the system issues alerts or triggers automated responses. This early detection capability is essential for catching intrusions before they escalate into full-scale breaches.

Secrets Management and Secure Credential Handling

One of the most common—and dangerous—server weaknesses is storing passwords, API keys, and private certificates in plaintext configuration files. Attackers frequently search for exposed credentials after gaining initial access.

Advanced secret-management solutions mitigate this by:

- Encrypting sensitive keys at rest and in transit
- Automating key rotation
- Enforcing strict access policies
- Storing secrets in centralised vaults rather than on servers

Tools like **HashiCorp Vault**, **AWS Secrets Manager**, and **Azure Key Vault** ensure credentials never reside openly on the server and cannot be harvested by attackers during a breach.

Behaviour-Based Threat Detection

Traditional log monitoring and signature-based tools are limited in their ability to detect new or unknown attacks. Behaviour-based detection adds a deeper layer of security by analysing how systems, users, and applications normally behave, then flagging abnormalities.

Examples of anomalous activity include:

- A service suddenly spawning unexpected child processes
- User accounts logging in outside normal hours
- A server transferring unusually large amounts of data
- A process repeatedly modifying system files
- Unexpected attempts to disable logging or security controls

Modern behavioural tools use machine learning or pattern analysis to identify suspicious operations long before clear indicators of compromise appear.

Immutable Server Architecture

Immutable infrastructure is a modern approach in which servers are not modified after deployment. Instead of updating or patching a live server, the system is replaced with a fresh, fully configured image.

This approach offers several advantages:

- Eliminates configuration drift
- Ensures consistent, predictable server state
- Prevents attackers from persisting changes on the server
- Reduces human error during patching or updates

In many cloud environments, immutable servers are deployed using tools like Terraform, Packer, or Kubernetes, which rebuild systems automatically when updates are required.

Hypervisor and Virtualisation Security

Most organisations run servers in virtualised environments, making the hypervisor a high-value target. Hypervisor compromises are rare but extremely damaging.

Key advanced controls include:

- Restricting access to hypervisor management interfaces
- Securing VM migration channels (e.g., vMotion)
- Isolating workloads with different sensitivity levels
- Applying strict role separation for host administrators
- Monitoring for abnormal VM behaviour or unauthorised snapshots

Ensuring the hypervisor layer is protected is critical, because a single compromise can expose every VM running on the host.

Container Security and Build Pipeline Protection

With the rise of containerised applications, securing server environments also requires securing the software supply chain.

Advanced practices include:

- Using **minimal, trusted base images**
- Scanning images for vulnerabilities before and after deployment
- Enforcing **non-root container execution**
- Validating image signatures before a container is allowed to run
- Using network policies in Kubernetes to restrict pod communication

A compromised container image can easily spread across an entire cluster, making container security essential for modern server deployments.

Continuous Configuration Monitoring and Drift Detection

Servers become insecure over time as changes accumulate—some intentional, others accidental. Continuous configuration monitoring (using tools like Ansible, Puppet, Chef, or Salt) ensures servers remain aligned with their security baseline.

This type of monitoring detects:

- Unauthorised configuration changes
- Unexpected services running
- New ports opening
- Changed permissions
- Altered firewall rules

Early detection of drift allows teams to correct issues before attackers exploit them.

Industry standards

In addition to general principles and strategies, there are several different cyber security industry standards that organisations can use to help ensure the security of their systems and information. Some common examples of these standards include:

1. **Cyber Essentials:** Cyber Essentials is a UK, government-backed certification scheme that provides a set of basic controls to help protect organisations against common cyber threats. It covers areas such as secure configuration, access controls, and malware protection.
2. **NIST (National Institute of Standards and Technology):** NIST provides a range of cyber security standards and guidelines, including the NIST Cybersecurity Framework, which is a set of best practices for managing and reducing cyber risks.
3. **CIS (Center for Internet Security):** CIS provides a set of security benchmarks that organisations can use to help ensure that their systems are properly

configured and secure. The benchmarks cover a wide range of security-related topics, including operating systems, applications, and networking.

4. **NCSC (National Cyber Security Centre):** The NCSC is the UK government's national technical authority for cyber security. It provides guidance and best practices for organisations looking to improve the security of their systems and information.
5. **ISO 27002:** ISO 27002 is an international standard that provides guidelines for information security management. It covers a wide range of topics, including security policies, access controls, and incident management.
6. **PCI DSS (Payment Card Industry Data Security Standard):** PCI DSS is a set of requirements for organisations that handle credit card information. It provides a framework for securing cardholder data and protecting consumers against fraud.

Take note

It's important to note that some of the above are guiding frameworks that need to be heavily interpreted and adapted for each organisation, such as the NCSC. Others are standards that have prescriptive settings that need to be applied in order to be compliant with the standard, such as the PCI DSS and ISO 27002.

Vulnerability management

Vulnerability management is the process of detecting, analysing, and remediating vulnerabilities in an organisation's systems and applications. This can involve conducting regular security assessments to identify potential vulnerabilities, and implementing controls and processes to reduce the risk of those vulnerabilities being exploited.

- **Detection** is the process of identifying potential security threats or vulnerabilities in an organisation's systems and applications. This can involve using tools and techniques like intrusion-detection systems, vulnerability scanners, and log analysis to identify potential security issues.
- **Nessus** and **Qualys** are both examples of commercial vulnerability-assessment and management tools. These tools can be used to scan an organisation's

systems and applications for potential vulnerabilities and provide guidance on how to remediate those vulnerabilities.

- **Analysis** involves examining the results of the vulnerability scan data and information to identify trends, patterns, and potential threats. This can include using tools and techniques like data visualisation and statistical analysis to help identify potential security issues. To move from analysis to remediation, it is essential to prioritise threats. The mechanism for doing this is risk management.
- A **risk** is the potential for something negative to happen to an organisation. It is measured by assessing the potential impact of a security threat or vulnerability against the likelihood that it will happen. **Risk management** is the process of prioritising vulnerabilities and focusing on the most critical first.
- **Remediation** is the process of addressing and fixing identified vulnerabilities. This can involve things like applying security patches and updates, implementing controls and safeguards, and conducting training to help prevent future vulnerabilities.
- **Patching** is the process of applying updates and fixes to an organisation's systems and applications to address identified vulnerabilities. This can involve applying security patches provided by vendors, as well as implementing custom fixes and controls.

Prevention versus cure

Prevention is always preferred to cure when it comes to cyber security, and many practices can be adopted to help an organisation prevent a cyber security breach.

These can include:

1. Taking regular **backups** and completing regular test restores to ensure the integrity of the data.
2. Utilise **snapshots** before any changes, where possible.
3. Ensure **immutable** copies of the data exist, i.e. data that cannot be infected with malware.
4. Ensure **network segmentation** is in place, i.e. separation of users from servers. In practice, this means that users can only access the services that they need for their job role, and therefore, the associated servers that provide those services. For instance, users within HR may need to access the HR system. However, most other users will not require this access, so this should be blocked at the network level by default and only users/devices from HR will be permitted to access these resources. In most organisations today, all users can potentially communicate with all servers, which poses a significant security risk as anyone walking in from

the street could plug into a network port and begin attacking servers. Removing this default access through segmentation reduces the attack surface dramatically.

5. Ensure **firewalls, intrusion-detection systems (IDS),** and **intrusion-prevention systems (IPS)** are in place at the perimeter and within the network.
6. Make sure your organisation has carefully planned and thoroughly tested **business continuity plans (BCP)** and **disaster recovery plans (DRP).**

Potential careers

The following careers utilise cyber security and programming skills to configure servers securely and address server vulnerabilities, or to test and ensure that they have been configured securely.

- **Penetration testing**, also known as 'pen testing' or 'ethical hacking', is a career that involves simulating real-world cyber attacks to identify and address vulnerabilities in an organisation's systems and applications. Pen testers use a range of tools and techniques to try to gain unauthorised access to systems and networks, and then report their findings and provide recommendations for addressing any vulnerabilities they find.
- **Software engineering and analysis** are careers that involve designing, developing, and testing software applications. These careers often involve working with various technologies and languages, such as Java, Python, and C++, and can involve working on both front-end and back-end systems.
- **Automation** uses technology to automate processes and tasks, such as testing or data analysis. In the context of cyber security, automation can be used to help organisations scale their security efforts and reduce the amount of manual work involved in tasks like vulnerability management or intrusion detection. Automation can involve using tools like scripting languages and automation frameworks to automate security-related tasks.

Overall, careers in **pen testing, software engineering and analysis, and automation** can all involve a mix of programming and cyber security skills. These careers can be challenging and rewarding, and can offer opportunities to work on a wide range of projects and technologies.

Take note

In the following practical task, you will answer questions related to vulnerability scanning. Ensure that you save the practical task files in this task folder before requesting a review.

Practical task

Most websites are driven by **web servers** and are an ideal target for learning about vulnerabilities, because they are publicly accessible and are influenced by secure server configurations.

Follow these steps:

1. Go to Pentest-Tools' **Website Vulnerability Scanner**.
2. In the "**Light scan**" box, enter the URL of the website you want to scan, such as www.gov.uk, www.talesfromthekitchenshed.com, or www.yourmechanic.com.
3. Ensure you have selected the "**Light scan**" tab.
4. Click on the "**Start scan**" button.
5. Wait for the scan to complete.
6. Review the results of the scan, which will show any vulnerabilities found on the website.
7. Research **one of the specific vulnerabilities found in the scan results** and learn about common ways to resolve or mitigate it:
    a. Some resources for researching vulnerabilities and their solutions include:
        - The **OWASP Top Ten project**,
        - The **National Vulnerability Database (NVD)**, and
        - The **Common Vulnerabilities and Exposures (CVE) database.**
    b. To use the CVE database:
        - Go to **CVE.org**.

- In the search bar icon, enter the CVE number or a keyword related to the vulnerability.
- Review the information provided about the vulnerability, including its description, impact, and any known solutions or mitigations.

8. Create a document called **web_scan.**
9. Describe the appropriate solution(s) to address one of the vulnerabilities that you discovered during your website scan in step six.

   For instance, if the scan results show a vulnerability with the CVE number **CVE-2021-22222**, and the vulnerability is related to a missing security patch on a software component, the solution would be to apply the latest security patch to the affected software component. **Please note:** sometimes your scan will not provide a CVE number.

   Note that this is just an example and not all vulnerabilities can be mitigated by applying patches, and it is always recommended to consult with experts or vendors and follow industry best practices to mitigate the vulnerabilities.

10. Submit your answer as a PDF in your task folder (**web_scan.pdf**).

Important: Be sure to upload all files required for the task submission inside your task folder and then click "Request review" on your dashboard.

Share your thoughts

Please take some time to complete this short feedback **form** to help us ensure we provide you with the best possible learning experience.

Reference list

Proveho Networks. (2020, January 6). *Security Architect Challenges*. Proveho Networks. **https://www.provehonetworks.com.au/security-architecture-challenges/**