EXTION INFOTECH PROJECT 1

NETWORK VULNERABILITY ASSESSMENTS

By – Sai Raghavendra . Baswa

Application id/ registration number-SAIR200472

# Table of Contents

1. INTRODUCTION TO PROJECT:
2. NETWORK VULNERABILITY TESTING:
3. ASSESSMENT METHODOLODY
4. TOOLS:
5. COMPLIANCE AND REGULATORY STANDARDS
6. VULNERABILITY CLASSIFICATIONS
7. ASSESSMENT FINDINGS
8. MITIGATION STRATEGIES
9. CONCLUSION
10. KEY RECOMMENDATIONS:
11. REFERENCES
12. REFERENCE 1: SWEET32 ATTACK
13. REFERENCE 2: nginx < 1.17.7 Information Disclosure
14. REFERENCE 3: TLS 1.0 VERSION
15. REFERENCE 4: Cross domain and client access policies
16. REFERENCE 5: SSL certificate cannot be trusted
17. REFERENCE 6: phpMyAdmin 2.6.4

# INTRODUCTION TO THE PROJECT-

The project titled "Network Vulnerability Assessment" was prepared by the security team operating at ABC Infotech, an imaginary company, that aimed to determine what kinds of vulnerabilities the company had, as a way not only to solve them but also as a proactive action towards avoiding future recurrences of vulnerabilities.

As security team, it is absolutely important to safeguard the company's systems from unauthorized personnel however, with the continuous innovation of technology,

This is the motivation and rationale behind the project since staying on top of new software's and mitigation measures to safeguard our systems is imperative.

A company system or object is said to be vulnerable when it has a weakness in its system that makes it vulnerable to attacks such as denial of service (DoS) or incidences of unauthorized access by third parties. Unless addressed immediately through active defense mechanisms and preventive methods, these vulnerabilities pave the way for cybercriminals to exploit them and result in the company system being jeopardized and the Confidentiality,

CIA triad not being upheld. Since the CIA triad must be ensured by systems, vulnerability scanning is required. Furthermore, ABC Infotech needs to uphold operational integrity by adhering to Cyber security regulations and industry compliance standards. The systems, software's and methodology adopted should be in accordance with the NIST, PCI and GDPR. Hence, scanning for vulnerabilities and areas that require patching It is necessary to be in accordance with the standards we are required to implement.

The following project aims at determining, protecting and preventing present and future weak points by first discovering them with a company IP address then it identifies the methods with which they can be worked on to protect them and safeguard them for the future .Through an application of well-trusted Vulnerability Scanner: Nessus and the use of Nmap, the project has adopted various sources into ensuring that all vulnerabilities have been found. By doing so, ABC infotech is able to maintain sensitive data, ensure only authorized personnel have access to data they require, ensure that company is complying with industry standards and maintaining integrity throughout the organization.

# NETWORK VULNERABILITY TESTING-

Network vulnerability testing, also known as network vulnerability assessment, is an essential process to detect, analyze, and remove security weaknesses in a network infrastructure. It identifies flaws in the network that could be exploited by attackers to gain unauthorized access, cause damage, or disrupt network operations.

Network vulnerability testing is usually carried out by network administrators, security professionals, and ethical hackers, also known as penetration testers, to ensure that the network remains secure against possible cyber threats.

*Key Objectives of Network Vulnerability Testing-*

1. **Identify Vulnerabilities**: This involves identifying weaknesses in the network that could be exploited by attackers. These may include out-of-date software, misconfigured network devices, weak passwords, or insecure communication protocols.
2. **Assess Risk:** Once vulnerabilities have been identified, the next step is to assess their potential impact on the organization's assets, data, and overall business operations.
3. **Prioritize Fixes**: Not all vulnerabilities are created equal. Some could be very serious and expose the network to severe threats while others could be of negligible impact. Vulnerability testing helps to prioritize which of these issues needs to be addressed first.
4. **Strengthen Network Security**: In this way, detecting and repairing vulnerabilities helps to strengthen the network's security and guard against unauthorized access, data breaches, and other types of attacks.

*Common Tools Utilized in Network Vulnerability Testing-*

1. **Nessus:** This is a commonly used vulnerability scanner that checks for vulnerabilities in operating systems, network devices, and databases. It also provides compliance checks.
2. **OpenVAS:** This is an open-source vulnerability scanner with identical functionality to Nessus, but free to use.
3. **Qualys:** This is a cloud-based vulnerability management solution used for identifying and managing network security vulnerabilities.
4. **Nmap:** A network scanning tool that can be used to discover devices, open ports, and services running on the network. It's often used as part of the scanning process before a vulnerability assessment.
5. **Wireshark:** A packet analyzer that helps in monitoring network traffic to detect suspicious activity or potential security issues at the packet level.
6. **Metasploit**: This is a very popular penetration testing tool. It can be used to simulate real-world attacks such as exploiting vulnerabilities to gain access to network systems.

# TOOLS-

**1) Nessus:** Nessus is a Tenable-powered vulnerability scanner which scans to try and help to identify potential vulnerabilities in the system, out-of-compliance settings, and configurations that may be exploited maliciously.

**2)Nmap:** Nmap often shortened for "Network Mapper," Nmap is a tool which could be employed using Linux to scan networks, host security audit purposes of discovering hosts and OS and it enables network administrators find which devices are running on their network, discover open ports andservices, and detect vulnerabilities.

**Nmap** was applied to find the company's network structure as well as the knowledge of attack surfaces on the network, including open ports and the tool used was **Nessus** to identify vulnerabilities within the company network, such as software that is not in compliance with the industry standards, potential attacks ABC Infotech is susceptible to and CVE vulnerabilities according to NIST.

# COMPLIANCE AND REGULATORY STANDARDS-

1. **ISO 27000-**

The ISO/IEC 27000 family of standards is an international family of frameworks for the management of information security. It involves best practices and guidelines to set up, implement, maintain, or improve information security management systems (ISMS) that are necessary to protect information confidentiality, integrity, and availability. It has been developed by ISO and IEC in order to help organizations manage sensitive data and ensure compliance with legal, regulatory, and contractual requirements.

1. **NIST Cyber security Framework (CSF):**

The NIST Cyber security Framework, officially called the NIST Framework for Improving Critical Infrastructure Cyber security, is a framework of guidelines designed to assist organizations in managing and mitigating cyber security risks. The NIST framework was developed by the National Institute of Standards and Technology and has gained widespread adoption across a broad range of industries, from government to healthcare, finance, and energy.

First conceived as a set of standards for critical infrastructure sectors like energy, banking, and telecommunications, the NIST CSF is now flexible and voluntary enough to apply to organizations of any size and type. The ultimate purpose is to enhance the management of cyber security risk in an integrated, structured approach based on protecting confidentiality, integrity, and availability of information systems.

1. **NIST SP 800-171:** Protecting Controlled Unclassified Information (CUI)

NIST Special Publication 800-171 is the guidelines developed by the National Institute of Standards and Technology to protect Controlled Unclassified Information in non-federal systems and organizations. This guideline provides a framework for the protection of sensitive but unclassified information when it is stored, processed, or transmitted by contractors, vendors, and other third parties outside the U.S. federal government. This publication is of great importance to organizations dealing with the U.S. federal government or its contractors, and they have specific requirements on how to handle CUI.

# VULNERABILITY CLASSIFICATIONS-

**CVSS Score Classification**

The classification of vulnerabilities is performed based on the final base score into the following:

1. **Critical: 9.0** through 10.0-the vulnerability is extremely exploitable and has high impact to the system. Therefore, they are to be acted upon with urgency.
2. **High: 7.0** through 8.9-the flaw is critical but not with the urgency of critical classification. Remedy should be expedited.
3. **Medium (4.0 - 6.9):** This vulnerability is of moderate risk. It may need to be mitigated but is not as urgent as more severe vulnerabilities.
4. **Low (0.1 - 3.9):** The vulnerability has a low impact and low exploitability. These are usually dealt with after the more serious vulnerabilities.
5. **None (0.0):** The vulnerability has no impact or cannot be exploited.

# ASSESSMENT FINDINGS-

Below are the vulnerabilities found that are found from the various sources. Evidence of the collated vulnerabilities can be referenced to at the end of the document.

1. **CVE ID**: **CVE-2016-2183**

 **Name**: SSL Medium Strength Cipher Suites Supported (SWEET32)

 **Severity**: High

 **CVSS Score**: **7.5 (High)**

**Description-**CVE-2016-2183; a known weakness in certain implementations of 64-bit block ciphers, a SWEET32-vulnerability - is known in terms of its capacity for attacking SSL/TLS communication by making use of weaker cipher suites that are described by medium-strength for usage in these protocols; such vulnerability being of significant degree. The SWEET32 vulnerability specifically takes aim at the use of 64-bit block ciphers, such as 3DES (Triple DES) and RC2, that remain in many older versions of SSL/TLS. Those ciphers are insecure for long-duration communications and may be decrypted with birthday attacks against data after sufficient amounts of traffic have flowed over a connection.

1. **CWE ID**: **CWE-327**

 **Name**: **TLS Version 1.0 Protocol Detection**

 **Severity**: **Medium**

 **CVSS Score**: **6.5 (Medium)**

**Description-**CWE-327 isa security weakness that involves the use of TLS version 1.0, which is now an outdated and deprecated protocol used to secure communications on the internet**.** TLS 1.0 was initially developed in 1999 as a successor to the SSL (Secure Sockets Layer) protocol. Since then, it has been replaced by more secure versions, namely TLS 1.2 and TLS 1.3.

1. **CVE ID**: **CVE-2019-20372**

 **Name**: **nginx < 1.17.7 Information Disclosure**

 **Severity**: **Medium**

 **CVSS Score**: **5.3 (Medium)**

**Description-**CVE-2019-20372 is identified as the vulnerability in the version of nginx web server under 1.17.7 and prior versions in which disclosure of information related to sensitive details about configurations or settings on the target server, or environment possibly can occur.

1. **Plugin #51192**

 **Name**: **SSL Certificate Cannot Be Trusted**

 **Severity**: **Medium**

 **CVSS Score**: **6.5 (Medium)**

**Description-**The plugin #51192 refers to a security issue that arises when an SSL certificate used by a server cannot be trusted. This can happen for various reasons, such as the certificate not being issued by a trusted Certificate Authority (CA), it being self-signed, or the certificate being expired or misconfigured. This vulnerability undermines the security of SSL/TLS communication because it enables attackers to intercept, manipulate, or impersonate the legitimate server during the communication process.

1. **Cross domain and client access policies**

**CVSS Base Score**: **6.5 (Medium)**

**Severity: Likely vulnerable (Medium)**

**Description-** Cross-Domain and Client Access Policies are core concepts in web security when it comes to resource sharing and services across different domains and client applications. This means if these policies have not been set up right or are overly permissive, it could lead to vulnerabilities in sensitive resources and even cross-site scripting (XSS) attacks.

1. **CVE ID**: **CVE-2005-3299**

 **Name**: **phpMyAdmin 2.6.4 (Multiple vulnerabilities)**

 **Severity**: **Medium**

 **CVSS Score**: **5.0 (Medium)**

**Description-CVE-2005-3299** refers to a **medium-severity vulnerability** discovered in **php My Admin 2.6.4**, a widely used web-based tool for managing MySQL databases. This specific vulnerability allows an attacker to potentially **bypass security controls** and execute actions without proper validation, leading to potential **unauthorized access** or **manipulation of dat**

**abase content**.

# MITIGATION STRATEGIES-

1. **CVE-2016-2183**

• Reconfigure the affected application in order to ensure other parts of the system are not compromised.

• Disable and deprecate the current cipher suites in the TLS or SSL configuration.

• Disable all 3DES Ciphers

• Use of stronger encryption algorithms such as AES for stronger and trusted protection from remote user attacks.

1. **CWE 327**

• Eliminate all the protocol dependencies for TLS 1.0 within the application

• Change system protocols usage to TLS 1.2 and TLS 1.3

1. **CVE-2019-20372**

• Upgrade to nginx version 1.17.7 or later versions

1. **Plugin #51192**

• Renew SSL certificate in order to see if it is going to update to a trustedversion

• Buy new SSL certificate

1. **Cross domain and client access policies**

• Verify permissions set to various web clients

• Grant permissions with the Principle of Least privilege to web clients to preserve confidentiality and prevent risk of Cross- Site forgery attacks.

• Use Token Synchronization, which is effective in preventing CSRF attacks because it makes sure that requests can only be made from a valid user session. That is, even if an attacker can generate a request that looks like it comes from the user, they will not have the correct token and the request will be rejected.

1. **CVE-2005-3299**

• White listing. This is a list of trusted email addresses, IP addresses, domain names or applications or even executable files, while denying all others. By having this and only allowing trustedsources, it eliminates the risk of third parties accessing files they arenot authorized to access.

• Use of databases rather than servers. Instead of saving files or information that can be compromised and have sensitive information on a web server, saving them on a database is much safer. This keeps CIA in place.

• Limit execution permissions for upload directories as well as upload file sizes.

• Perform dynamic application security testing to determine if your code is vulnerable to file inclusion exploits.

• Sanitize user-supplied inputs, including GET/POST and URL parameters, cookie values, and HTTP header values. Validate on the server side, not the client side.

Through the mitigation and remediation of these measures, it is possible to sustain the security that the system of the organizations holds. Besides this, the different risks the organization faces makes us to remain ahead by upgrading the patches on our system and its areas that should be reconfigured fully.

# CONCLUSION-

The project conduced by the security team at CDF was generally a success since it has enabled us to identify, evaluate and protect our systems from vulnerabilities we are vulnerable to as a company. By employing Nessus and Nmap, six Main vulnerabilities found to exist with one being categorized as "High"While the rest maintained an overall scoring of "Medium".

# KEY RECOMMENDATIONS-

The information below is a summary of the found mitigation and preventative methods that will be implemented to addressing each of the six vulnerabilities found within the systems;

1. **Patching and Updates of system**: This includes updating of un trusted SSL certificates and update to newer versions of protocols such as the TLS 1.2 in order to ensure that system has up to date security measures and does not pave the way for man-in-the-middle attacks that are seen in older software versions.
2. **Implementation of safer security measures:** This entails the implementation of safer habits that are more secure to the system such as use of White listing that ensures only trusted and authorized sources have access rights. Furthermore, this includes reviewing of current permissions set to web servers and reconfiguring them to increase security within our systems. Finally, use of databases rather than web serves for data storage to avoid

information being compromised.

1. **Regular monitoring:** This includes consistent vulnerability assessments to make sure not only the system is geared for newer cyber attacks methods but to protect against current vulnerabilities and current exploits.

In addition, this also encompasses monitoring with logs regularly in order faster detection of potential attacks or unusual activity for a more proactive approach.

1. **Network protection and safeguarding**: This involves proactive

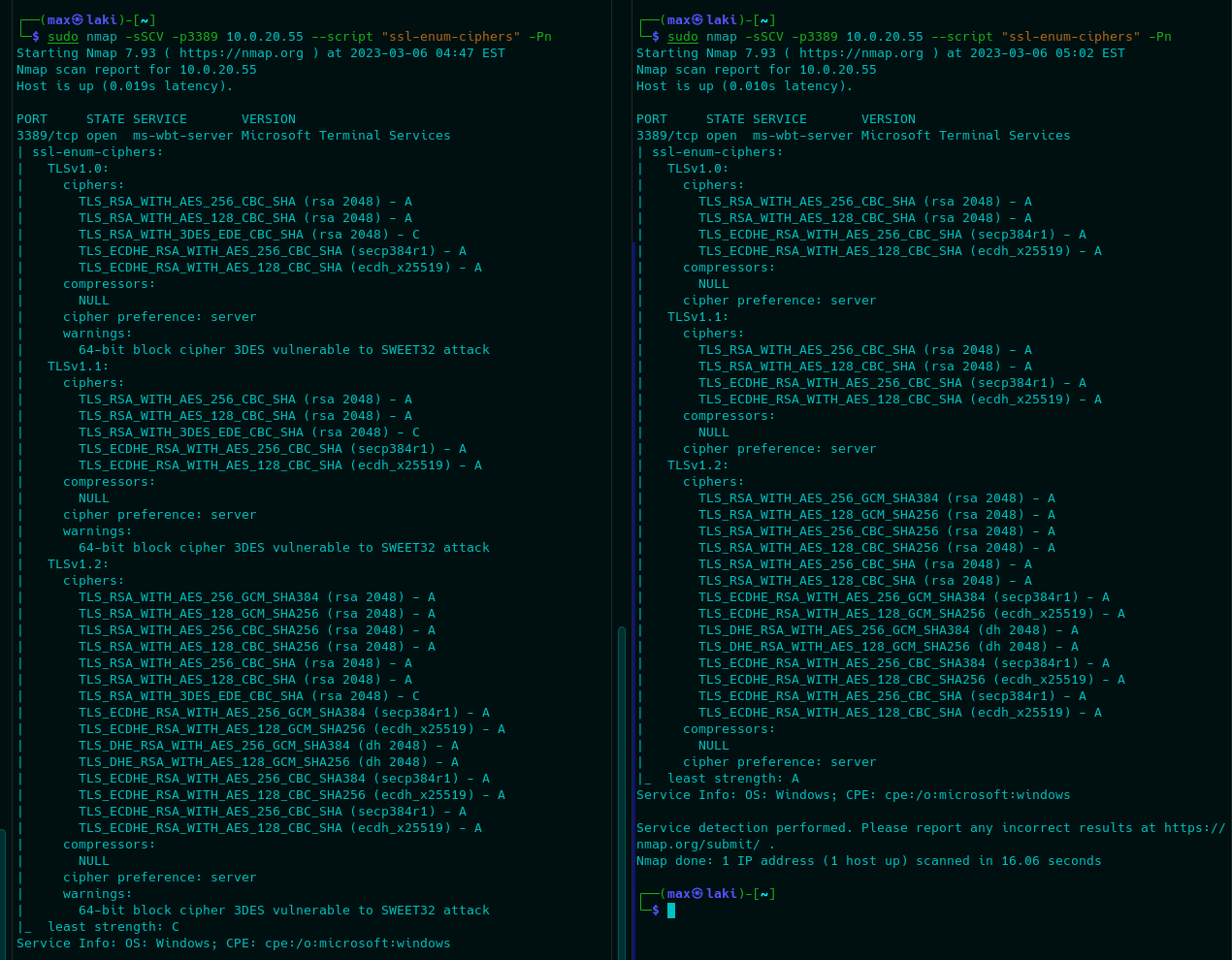
safeguarding measures to ensure CIA of the organization systems including

Implementation of stronger firewalls as well as permissions to web clients

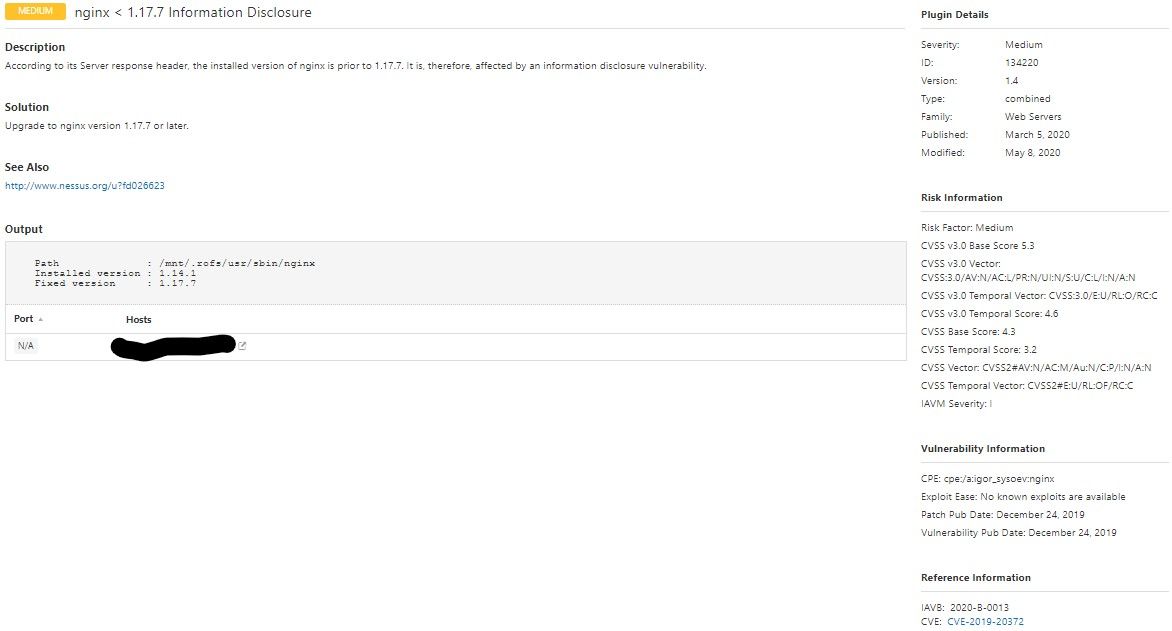
being offered under the Principle of Least Privilege agenda.

# REFERENCES

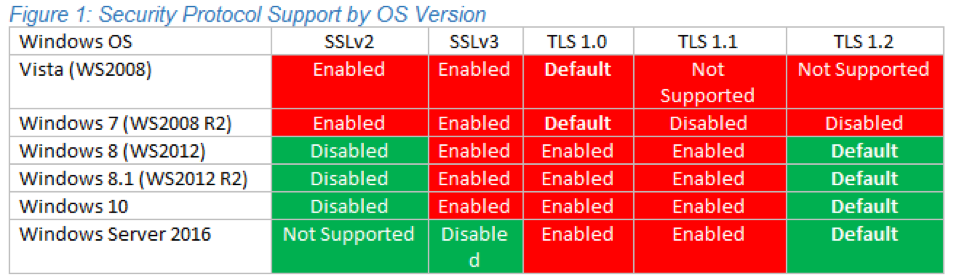
1. **Reference 1: Sweet 32 attack**

****

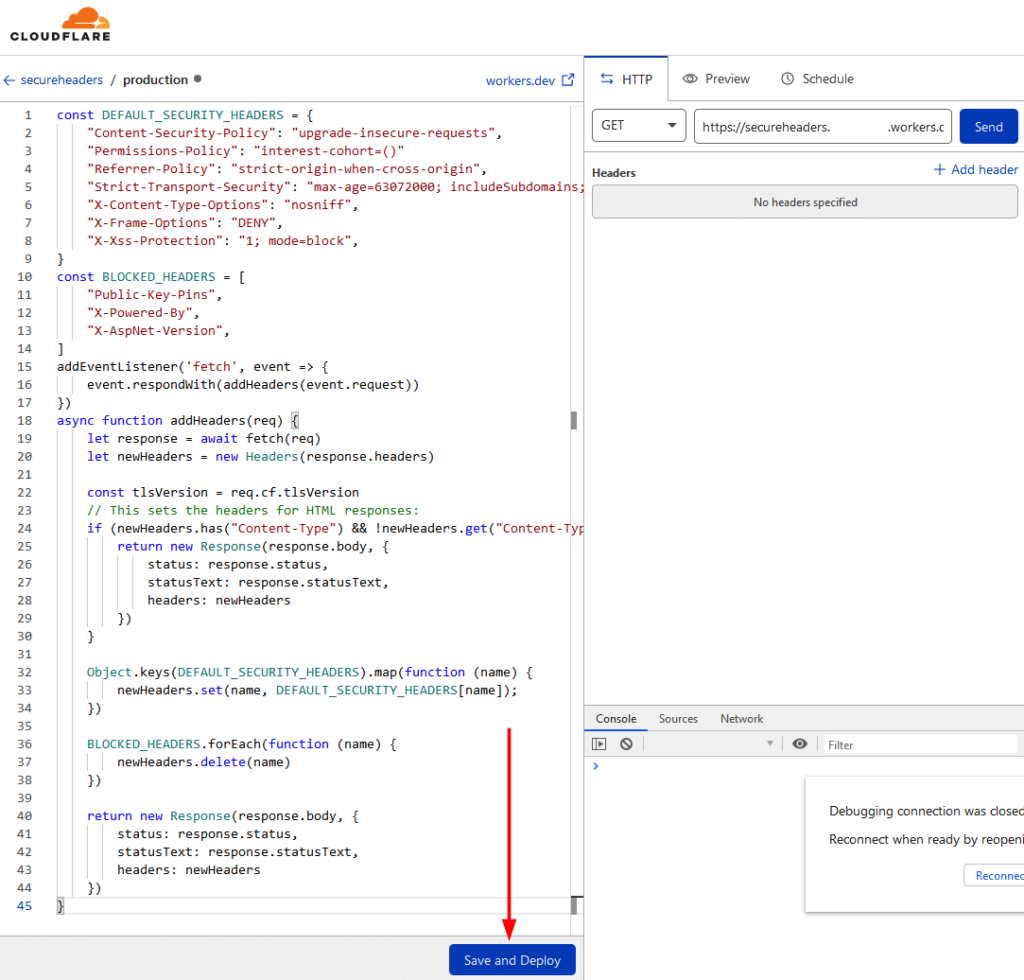
1. **Reference 2: Nginx < 1.17.7 Information Disclosure**



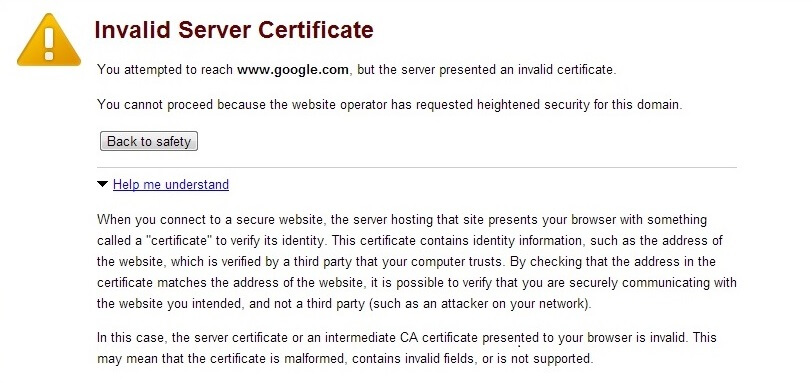
1. **Reference 3: TLS 1.0 VERSION**



1. **Reference 4: Cross domain and client access policies**



1. **Reference 5:** **SSL Certificate Cannot be trusted**



1. **Reference 6: phpMyAdmin 2.6.4**

