Project Report:

Identifying and

Mitigating Network

Vulnerabilities using Nessus

# Submitted by

**Sourav Bhongade**

**CYBER SECURITY INTERNSHIP**

**AT**

**EXTION INFOTECH**

1. Introduction

Overview of Nessus

1. Vulnerabilities Identification
2. Mitigation Plan for Identified Vulnerabilities
3. Conclusion

# Introduction

## 1.1 Project Objective

The main goal of this project is to use Tenable Nessus to identify and address network vulnerabilities. This report covers the entire process, including setting up the environment, conducting vulnerability scans, analyzing the results, and implementing mitigation strategies to improve the network's overall security.

## 1.2 Background

Network vulnerabilities can lead to data breaches, unauthorized access, and denial of service attacks. It's essential to identify these vulnerabilities to maintain a secure and strong network infrastructure. Tenable Nessus has established itself as a cornerstone in the field of vulnerability assessment, providing organizations with the tools they need to identify and address security weaknesses effectively.

## 1.3 Scan Objectives

* Identify at least five Vulnerabilities in the network.
* Provide detailed reports on each vulnerability.
* Provide recommendations for mitigating identified vulnerabilities.

# Overview of Nessus

## 1. Historical Development and Evolution

The journey of Nessus began in the late 1990s when Renaud Deraison, motivated by the need for better vulnerability assessment tools, created the initial version of Nessus. Originally an open-source project, Nessus provided a platform for security professionals to identify vulnerabilities in their systems. Its open-source nature allowed for community contributions and rapid updates, fostering a collaborative approach to cybersecurity.

In 2005, Tenable, Inc. acquired Nessus, transitioning it from an open-source project to a commercial product. This transition introduced new features, enhanced functionality, and a more structured support model. Despite this shift, Tenable continued to maintain a free version of Nessus, known as Nessus Essentials, for personal and educational use.

**2. What is Nessus?**

Nessus is a widely-used vulnerability scanning tool developed by Tenable, Inc. It is designed to identify security vulnerabilities in computer systems, networks, and applications. By detecting these vulnerabilities, Nessus helps organizations to mitigate potential security risks before they can be exploited by attackers. Originally developed by Renaud Deraison in 1998, Nessus has evolved into a robust tool used globally by security professionals to enhance their cybersecurity posture.

## 3. Key Features of Nessus

1. **Vulnerability Scanning**

Nessus performs comprehensive scans of systems and networks to identify known vulnerabilities. It uses a continually updated database of vulnerability signatures and threat intelligence to detect security issues across various components, including operating systems, applications, and network devices.

1. **Extensive Plugin Library**

Nessus utilizes a large library of plugins that are regularly updated to address new vulnerabilities and threats. Each plugin is designed to check for specific vulnerabilities or configuration issues. This extensive library ensures that Nessus can provide thorough coverage of known security issues.

1. **Customizable Scan Policies**

Users can create and customize scan policies to tailor assessments to their specific needs. Nessus allows users to define scan parameters, select specific plugins, and configure settings to focus on particular areas of interest or compliance requirements.

1. **Detailed Reporting**

After completing a scan, Nessus generates detailed reports that provide insights into identified vulnerabilities. These reports include information on the nature and severity of each vulnerability, along with recommendations for remediation. Reports can be customized to include specific details and formats.

1. **Integration and Automation**

Nessus integrates with a variety of security tools and platforms, such as security information and event management (SIEM) systems, ticketing systems, and network management tools. It also supports automation through its API, allowing organizations to incorporate vulnerability scanning into their continuous security monitoring and incident response processes.

1. **User-Friendly Interface**

Nessus features a user-friendly web-based interface that simplifies the configuration and management of scans. The interface provides easy access to scan settings, results, and reports, making it accessible for users with varying levels of expertise.

## 4. Advantages of Nessus

* **Comprehensive Vulnerability Detection**

Nessus provides a thorough assessment of systems and networks by identifying a wide range of vulnerabilities. Its extensive plugin library and regular updates ensure that it can detect both common and emerging threats.

* **User-Friendly Interface**

The web-based interface of Nessus is intuitive and easy to navigate. It simplifies the process of configuring scans, reviewing results, and generating reports, making it accessible to users with varying levels of expertise.

* **Customizability**

Nessus offers flexible configuration options, allowing users to create customized scan policies and tailor assessments to their specific needs. This customization enhances the effectiveness of vulnerability management and ensures that scans align with organizational requirements.

* **Integration and Automation**

Nessus integrates with other security tools and platforms, facilitating seamless integration into broader security operations. Its automation capabilities, through API support, enable organizations to incorporate vulnerability scanning into continuous monitoring and incident response workflows.

* **Cost-Effective**

Nessus provides a cost-effective solution for vulnerability assessment compared to other tools in the market. Its scalability makes it suitable for organizations of all sizes, and its comprehensive features help avoid costly security breaches and incidents.

* **Regular Updates and Support**
  + Tenable provides regular updates to Nessus, including new plugins and features, to address evolving threats. Additionally, users have access to support resources, including documentation, forums, and customer support, to assistant.

# Vulnerability Identification

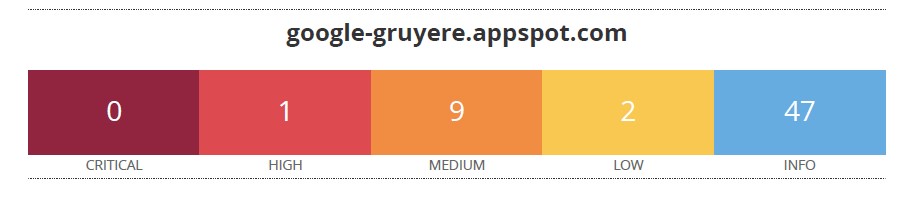
* Summary :

**Scan Date:** 2025-01-29

**Target:** google-gruyere.appspot.com

**Scan Type:** Basic Network Vulnerability Scan

**Nessus Plugin Set:** Standard



* Top 5 Vulnerabilities Overview:

1. **High (7.5):**
   * + **Vulnerability:** SSL Medium Strength Cipher Suites

Supported (SWEET32)

* + - **Plugin ID:** 42873
    - **Description:** The remote host supports the use of SSL ciphers that offer medium strength encryption. Nessus regards medium strength as any encryption that uses key lengths at least 64 bits and less than 112 bits, or else that uses the 3DES encryption suite. Note that it is considerably easier to circumvent medium strength encryption if the attacker is on the same physical network.

1. **Medium (6.5):**
   * **Vulnerability:** HSTS Missing from HTTPS Server (RFC 6797)
   * **Plugin ID:** 142960
   * **Description:** The remote web server is not enforcing HSTS, as defined by RFC 6797. HSTS is an optional response header that can be configured on the server to instruct the browser to only communicate via HTTPS. The lack of HSTS allows downgrade attacks, SSL-stripping man-in-the-middle attacks, and weakens cookie-hijacking protections.

1. **Medium (6.5):**
   * **Vulnerability:** TLS Version 1.0 Protocol Detection
   * **Plugin ID:** 104743
   * **Description:** The remote service accepts connections encrypted using TLS 1.1. TLS 1.1 lacks support for current and recommended cipher suites. Ciphers that support encryption before MAC computation, and authenticated encryption modes such as GCM cannot be used with TLS 1.1 As of March 31, 2020, Endpoints that are not enabled for TLS 1.2 and higher will no longer function properly with major web browsers and major vendors.
2. **Medium (5.3):**
   * **Vulnerability:** SSL/TLS Protocol Initialization Vector Implementation Information Disclosure Vulnerability (BEAST)
   * **Plugin ID:** 58751
   * **Description:** The SSL/TLS protocol used by the remote server is susceptible to the BEAST attack, a known vulnerability that exploits the way SSL 3.0 and TLS 1.0 handle block cipher encryption. The vulnerability occurs due to a flaw in the implementation of the Cipher Block Chaining (CBC) mode, which allows an attacker to decrypt portions of encrypted data (e.g., HTTP cookies or session tokens) by conducting a man-in-the-middle attack and injecting malicious JavaScript into the victim's browser.
3. **Medium (4.3):**
   * + **Vulnerability:** CGI Generic HTML Injections
     + **Plugin ID:** 49067
     + **Description:** The web application running on the server contains a CGI script that is vulnerable to generic HTML injection attacks. This vulnerability allows an attacker to inject arbitrary HTML or JavaScript into the web page, which may be executed by the user's browser. This can lead to various attacks, such as defacement, redirection, or even cross-site scripting (XSS), depending on the context of the injection.

## Mitigation Plan for Identified Vulnerabilities

### 1. SSL Medium Strength Cipher Suites Supported (SWEET32)

* **Steps for Remediation:**
  + **Identify Affected Servers:**
    - Locate all servers using SSL/TLS that support 3DES or other medium-strength cipher suites.
  + **Update SSL/TLS Configuration:**
    - Disable all 64-bit block ciphers, such as 3DES, in the server’s SSL/TLS configuration. o Configure the server to support only strong cipher suites, such as those using AES with a minimum key size of 128 bits (e.g., AES-GCM).
  + **Test the New Configuration:** o Use tools like SSL Labs’ SSL Test or Nessus to verify that medium-strength ciphers are no longer supported.
  + **Roll Out the Changes:**
    - Apply the updated configurations across all affected servers.

* **Estimated Timeline:** 1 week

* **Required Resources:**
* Access to SSL/TLS configuration files.
* SSL testing tools.
* Network administrator.

### 2. HSTS Missing from HTTPS Server (RFC 6797)

* **Steps for Remediation:**
  1. **Review HTTPS Configuration:**
     + Ensure all content is served over HTTPS. o Identify all subdomains that should be included in the HSTS policy.
  2. **Enable HSTS:**
     + Add the following header to the HTTPS server configuration:

[ Strict-Transport-Security: maxage=31536000;includeSubDomains; preload ] o Test the configuration to ensure it is working correctly.

* 1. **Monitor and Adjust:**
     + Monitor for any issues, particularly with legacy systems that may not fully support HSTS.
  2. **Submit for Preload (Optional):**
     + If the domain is ready, submit it to the HSTS preload list for browsers.

* **Estimated Timeline:** 2 weeks

* **Required Resources:**
* Web server configuration access.
* Web developers or administrators.
* Testing tools (e.g., browser developer tools).

### 3. TLS Version 1.0 Protocol Detection

 **Steps for Remediation:** 1. **Identify Servers Using TLS 1.0:**

o Audit the network to identify servers and services that still support TLS 1.0.

1. **Update SSL/TLS Protocols:**
   * Disable support for TLS 1.0 and 1.1 in the server’s SSL/TLS configuration.
   * Ensure the server supports TLS 1.2 or TLS 1.3.
2. **Test Compatibility:**
   * Test all client connections and applications to ensure compatibility with the updated protocols.
3. **Deploy Updates:**
   * Apply changes to all identified servers and services.

* **Estimated Timeline:** 3 weeks

* **Required Resources:**
* SSL/TLS configuration access.
* Compatibility testing tools.
* Network administrator.

### 4. SSL/TLS Protocol Initialization Vector Implementation Information Disclosure Vulnerability (BEAST)

* **Steps for Remediation:**
  1. **Disable SSL 3.0 and TLS 1.0:** o Follow the steps outlined in the mitigation for TLS 1.0 to disable both SSL 3.0 and TLS 1.0.
  2. **Reconfigure Cipher Suites:** o If TLS 1.0 must be retained for compatibility, reconfigure the server to prioritize RC4 (despite its own weaknesses) over CBC-mode ciphers for TLS 1.0. However, it’s strongly recommended to upgrade to TLS 1.2 or higher.
  3. **Implement TLS 1.2/1.3:** o Ensure TLS 1.2 or 1.3 is used, as these versions are not vulnerable to the BEAST attack.
  4. **Test the Configuration:** o Validate that the server is no longer vulnerable using tools like SSL Labs or Nessus.

* **Estimated Timeline:** 2 weeks

* **Required Resources:**
* SSL/TLS configuration access.
* Testing tools.
* Network administrator.

### 5. CGI Generic HTML Injections

* **Steps for Remediation:**
  1. **Identify Vulnerable CGI Scripts:** o Locate and review all CGI scripts on the server for input handling.
  2. **Sanitize Input:** o Implement proper input validation and sanitization in CGI scripts to prevent HTML injection.
     + Use libraries or frameworks that handle input sanitization.
  3. **Test for Vulnerabilities:** o Perform security testing using tools like Burp Suite to ensure that the vulnerability is fixed.
  4. **Deploy Updated Scripts:**
     + Replace the vulnerable scripts with the updated versions.
  5. **Monitor and Audit:**
     + Regularly audit the scripts and monitor logs for any signs of injection attempts.

* **Estimated Timeline:** 3 weeks

* **Required Resources:**
* Web developers familiar with CGI scripts.
* Security testing tools (e.g., Burp Suite).
* Code review tools.

## Conclusion

1. Project Outcomes

This project successfully identified and mitigated numerous network vulnerabilities using Nessus. The comprehensive scanning and analysis provided valuable insights into the security posture of the network, enabling targeted and effective remediation efforts.

2 Future Work

Future work includes establishing a regular vulnerability assessment schedule, continuous monitoring for new threats, and ongoing improvements to the network security infrastructure. Adopting a proactive approach to network security will help maintain a robust and resilient network environment.

**Project Report:**

**Investigation of a Data Breach**

**ABC SecureBank**

By Sourav Bhongade

29 January, 2025

**Table of Contents**

1. Objective
2. Scenario Overview
3. Tasks and Findings
4. Incident Analysis
5. Forensic Analysis
6. Data Recovery
7. Regulatory Compliance
8. Communication and Notification
9. Post-Incident Review

4. Conclusion

# Objective

This report investigates a data breach at ABC SecureBank, a highly respected financial institution. It covers how the breach occurred, the damage extent, forensic analysis, data recovery, regulatory compliance, communication strategies, and post-incident evaluation. This case evaluates forensic and investigative skills in managing a data breach.

# Scenario Overview

**Scenario:**

Imagine that there has been a data breach at a renowned website, and task is to investigate this breach. While the website's name is fictional, the scenario will test the investigative and forensic skills.

**Details:**

**Company Name:** ABC SecureBank, a highly reputable financial institution.

**Breach Discovery:** The breach was discovered during a routine security audit, and it appears that sensitive customer data may have been exposed.

**Scope of Breach:** The breach involves potential exposure of customer account information, including names, account numbers, and transaction history.

# Tasks and Findings

## 1. Incident Analysis

### 1.1 Overview of the Breach

ABC SecureBank, a highly reputable financial institution, recently discovered a significant data breach during a routine security audit. The breach potentially exposed sensitive customer data, including names, account numbers, and transaction history. Given the nature of the data involved, this breach poses serious risks to customer privacy, the bank's reputation, and its legal standing.

### 1.2 Identification of the Breach

The breach was identified during a scheduled security audit, a critical component of the bank's cybersecurity strategy. Auditors found anomalies in the access logs of customer databases, indicating unauthorized access to sensitive data. A detailed analysis revealed that the breach might have persisted for an extended period before detection, raising concerns about the adequacy of the bank's monitoring systems.

### 1.3 Initial Response

Upon discovering the breach, ABC SecureBank immediately initiated its incident response protocol. The response team was mobilized to contain the breach, assess the damage, and initiate communication with stakeholders, including customers, regulatory bodies, and law enforcement.

### 1.4 Point of Entry

To determine how the breach occurred, the investigation focused on identifying the point of entry used by the attackers. The forensic team began by analyzing firewall logs, Intrusion Detection System (IDS) alerts, and server access logs. They discovered that the breach likely originated from a phishing attack targeting employees of ABC SecureBank.

The phishing emails were crafted to appear as legitimate communications from internal departments, tricking recipients into clicking on malicious links. These links directed users to a compromised website that installed malware on their systems. The malware provided the attackers with remote access to the compromised machines, allowing them to escalate privileges and gain access to the bank's internal network.

### 1.5 Attack Vector

Once inside the network, the attackers exploited a known vulnerability in the bank's web application. The vulnerability was a result of outdated software that had not been patched, despite the availability of a security update. The attackers used SQL injection (SQLi) techniques to exfiltrate customer data from the database. This type of attack involves inserting malicious SQL queries into input fields, which the application then executes, allowing unauthorized access to the database.

### 1.6 Extent of the Breach

The investigation revealed that the attackers accessed several critical databases containing sensitive customer information. The extent of the breach included the following data:

* Customer names
* Account numbers
* Transaction history

The attackers accessed this data over a period of three months, during which they exfiltrated large volumes of information. It is suspected that the data was sold on underground forums or used for fraudulent activities.

## 2. Forensic Analysis

### 2.1 Overview

The forensic analysis aimed to gather evidence of the breach, identify any malware or suspicious activities on the affected systems, and trace the activities of the attackers.

### 2.2 Evidence Collection

The forensic team began by creating a forensic image of the affected systems. This process involved making an exact copy of the hard drives to preserve the state of the systems at the time of the breach. The forensic image was analyzed using specialized tools to identify traces of the attackers' activities.

### 2.3 Malware Analysis

The team identified the malware used in the attack by analyzing the forensic images. The malware was a Remote Access Trojan (RAT) that allowed the attackers to control the compromised systems remotely. The RAT was designed to evade detection by traditional antivirus software, making it difficult to identify during routine scans.

The malware analysis revealed that the RAT communicated with a command-and-control (C2) server located in a foreign country. This server issued commands to the compromised systems, allowing the attackers to exfiltrate data, escalate privileges, and move laterally within the network.

### 2.4 Log Analysis

Log analysis was a critical component of the forensic investigation. The team analyzed firewall logs, server access logs, and IDS alerts to trace the attackers' activities. The analysis revealed the following key points:

* The initial point of entry was through a phishing email that targeted employees.
* The attackers used the compromised systems to access the internal network and escalate privileges.
* The attackers exploited a known vulnerability in the web application to gain access to the database.
* Data exfiltration occurred over a period of three months.

The logs also revealed that the attackers attempted to cover their tracks by deleting certain log entries. However, the forensic team was able to recover these logs using specialized tools.

### 2.5 Timeline of the Breach

Based on the forensic analysis, the following timeline of the breach was established:

* **Day 1:** Attackers send phishing emails to ABC SecureBank employees.
* **Day 2:** An employee clicks on a malicious link, resulting in the installation of a RAT on their system.
* **Day 5:** Attackers gain access to the internal network and escalate privileges.
* **Day 7:** Attackers identify and exploit a vulnerability in the web application to access the database.
* **Day 10 - Month 3:** Attackers exfiltrate customer data from the database.
* **Month 4:** The breach is discovered during a routine security audit.

## 3. Data Recovery

### 3.1 Overview

The data recovery phase focused on determining the type and quantity of customer data that was potentially exposed and developing a strategy for data recovery and incident containment.

### 3.2 Identification of Exposed Data

The forensic team identified the types of data that were potentially exposed during the breach. This included:

* Customer names
* Account numbers
* Transaction history

The quantity of data exposed was significant, with millions of customer records potentially compromised. The attackers targeted the most valuable data, which could be used for fraudulent activities such as identity theft, unauthorized transactions, and financial fraud.

### 3.3 Data Recovery Strategy

Given the nature of the breach, the data recovery strategy focused on the following key areas:

#### 3.3.1 Containment

The first priority was to contain the breach and prevent further data loss. This involved:

* Disconnecting the compromised systems from the network to stop further exfiltration.
* Patching the web application vulnerability to prevent further exploitation.
* Scanning the entire network for additional malware or suspicious activities.
* Implementing additional security controls, such as enhanced monitoring and multifactor authentication (MFA), to prevent future attacks.

3.3.2 Communication

Effective communication was critical in managing the breach. ABC SecureBank needed to inform customers, regulatory bodies, and law enforcement about the breach. The communication strategy included:

* Sending notifications to affected customers, explaining the breach and advising them on steps to protect themselves.
* Coordinating with law enforcement to investigate the breach and track down the attackers.
* Working with regulatory bodies to ensure compliance with legal requirements and avoid penalties.

#### 3.3.3 Customer Support

To assist customers affected by the breach, ABC SecureBank set up a dedicated customer support team. This team provided assistance with:

* Monitoring customer accounts for suspicious activities.
* Offering identity theft protection services, such as credit monitoring and fraud alerts.
* Reimbursing customers for any unauthorized transactions that occurred as a result of the breach.

#### 3.3.4 Legal and Regulatory Compliance

The breach exposed ABC SecureBank to potential legal and regulatory consequences. The data recovery strategy included:

* Conducting a thorough legal review to determine the bank's liability and potential exposure to lawsuits.
* Ensuring compliance with data breach notification laws, such as the General Data Protection Regulation (GDPR) in the European Union and the California Consumer Privacy Act (CCPA) in the United States.
* Coordinating with legal counsel to prepare for potential litigation and regulatory investigations.

### 3.4 Long-Term Mitigation

In addition to immediate containment and recovery, ABC SecureBank developed a longterm mitigation plan to prevent future breaches. This plan included:

* **Security Awareness Training:** Enhancing security awareness training for employees to prevent phishing attacks and other social engineering tactics.
* **Regular Security Audits:** Increasing the frequency of security audits to identify vulnerabilities and address them promptly.
* **Advanced Threat Detection:** Implementing advanced threat detection tools, such as Security Information and Event Management (SIEM) systems and endpoint detection and response (EDR) solutions, to detect and respond to threats in realtime.

## 4. Communication and Notification

**Objective:** Create a communication plan to notify affected customers, stakeholders, and regulatory bodies, ensuring clarity and compliance with privacy laws.

**Communication Plan:**

* Customers: Notify affected customers via email and letter, providing breach details, exposed information, and recommended actions like monitoring their accounts.
* Stakeholders: Hold meetings with key stakeholders to explain the breach's impact and the measures taken.
* Regulatory Bodies: Submit a comprehensive incident report to relevant authorities, including a breach timeline and remediation steps.

## 5. Post-Incident Review

**Objective:** Perform a detailed review to identify security vulnerabilities and recommend enhancements after containing and mitigating the breach.

**Security Weaknesses Identified:**

* Insufficient employee training on phishing threats. o Lack of adequate network segmentation, enabling lateral movement of attackers. o Inadequate monitoring of network activity for suspicious behavior.

**Recommendations:**

* Training: Implement regular cybersecurity training for all employees, focusing on phishing and social engineering threats.
* Network Segmentation: Enhance network segmentation to restrict access to sensitive data.
* Monitoring: Improve network monitoring and deploy intrusion detection systems (IDS) to detect and respond to suspicious activity more rapidly.
* Regular Audits: Conduct more frequent security audits and vulnerability assessments.

# Conclusion

The data breach at ABC SecureBank highlights the importance of a comprehensive cybersecurity strategy that includes robust incident response, forensic analysis, and data recovery processes. The breach was a result of a sophisticated phishing attack combined with the exploitation of a known vulnerability in the bank's web application. The forensic analysis provided valuable insights into the attackers' activities, while the data recovery strategy focused on containment, communication, and long-term mitigation.

The lessons learned from this breach underscore the need for continuous improvement in cybersecurity practices, including regular security audits, employee training, and the adoption of advanced threat detection technologies. By implementing these measures, ABC SecureBank can better protect its customers' data and maintain its reputation as a trusted financial institution.