**{{COMPANY\_NAME}}**

**Web Application Grey Box Testing**

**{date}**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version Control | | | | | |
| Version | Date | Created by | Reviewed By | Approved by | Report Stage |
| 1.0 | 30th Aug 2023 | {name} | Ummed Meel | Ravleen Kaur | Final Report |

|  |  |  |
| --- | --- | --- |
| Report Distribution | | |
| Name | Organization | Purpose |
| {client name} | {client company } | Web Application Grey Box Testing |

Contents

[**Report Guide** 3](#_Toc144905884)

[**Introduction** 5](#_Toc144905885)

[**Project Background** 5](#_Toc144905886)

[**Scope** 5](#_Toc144905887)

[**Project Team**  6](#_Toc144905888)

[**Executive Summary**  7](#_Toc144905889)

[**Risk Rating Criteria** 7](#_Toc144905890)

[**Observation Summary Table** 8](#_Toc144905891)

[**Snapshot of Observations** 10](#_Toc144905892)

[**Detailed Observations** 11](#_Toc144905893)

[**1.** **Insecure direct object reference** 11](#_Toc144905894)

[**2.** **Stored HTML injection** 14](#_Toc144905895)

[**Annexure** 17](#_Toc144905896)

# 

# **Report Guide**

The following table depicts the flow of this report.

|  |  |
| --- | --- |
| Section | Description |
| Introduction | This section basically sets the tone of the Web application penetration testing security assessment report and draws the boundaries of the report in terms of its objective, scope, project timeline, project team from both sides. |
| Executive Summary | This section is prepared for quick management reference. It contains a summary of observations from our review of Web application penetration testing. |
| Detailed Report | This section presents the detail of the observations/ gaps found in network infrastructure along with the following:   * Risk rating * Status * Revalidation Status * Description of observation * Impact * Recommendation to address the risk. * Proof of Concept |
| Annexures | Contains high level approach adopted for the Web application penetration testing assessment including assumptions. |

# **Introduction**

## **Project Background**

**Grant Thornton Bharat (GT)** was engaged to conduct Web application penetration testing for {client company} IT assets. Web application penetration testing was conducted using similar tools and techniques that a malicious attacker would use to try and compromise Security of Web application with respect to:

* Confidentiality
* Integrity
* Availability

The purpose of this assessment was to identify technical as well as logical vulnerabilities in the publicly exposed assets and provide recommendations for risk mitigation that may arise on successful exploitation of these vulnerabilities. The idea behind this testing was to discover whether an attacker may leverage flaws in the applications and supporting infrastructure to compromise the security at {client company}.

## **Scope**

The scope of the internal security assessment covers testing on the given URL addresses based on the Penetration Testing Execution Standards (PTES) guidelines. The following URL addresses were subjected to the Web application penetration testing:

{{SCOPE\_TABLE}}

|  |  |
| --- | --- |
| **S. No** | **In-scope URL** |
| 1 | https://www.google.com/scope |

## **Project Team**

Following team members were involved in this assessment:

|  |  |
| --- | --- |
| GT Security Team | Contact Information |
| Ravleen Kaur  Ummed Meel | [ravleen.kaur@in.gt.com](mailto:ravleen.kaur@in.gt.com)  [Ummed.meel@in.gt.com](mailto:Ummed.meel@in.gt.com) |
| Hubilo Team | Contact Information |
| {client name} | {client email} |

# **Executive Summary**

Based on our Web application penetration testing activity,we provide {{COMPANY\_NAME}}’s management with an indication of the significance of risk involved and the priority with which the same needs to be addressed. We have categorized the observations illustrated in this report in accordance with the classifications given below:

## **Risk Rating Criteria**

The risk rating is used to signify the level of risk due to gaps noted during the audit and is based on a qualitative criterion defined as follows:

|  |  |
| --- | --- |
| **Severity Rating** | |
| **Critical** | Critical risk vulnerability has a high potential of impacting business operations leading to downtime or disruption and provides an attacker with privileged access, resulting in significant outage. If exploited, it has a direct impact on confidentiality, integrity or availability of organizational information. |
| **High** | High risk vulnerability indicates that successful exploitation of the vulnerability may result in a significant impact to the confidentiality, integrity, or availability of the information accessible through the application/system or even the backend resources like databases, operating systems, etc. |
| **Medium** | Medium risk vulnerability reveals information about the application and its underlying infrastructure that can be used by an attacker in conjunction with another vulnerability to gain privileged control of the application or its underlying operating system. |
| **Low** | Low risk vulnerability that has the potential of revealing the information about the system and may lead to unauthorized access to a system, leading to compromise. Higher work factors would be involved for exploiting this type of vulnerability. |

To capture the risk rating, the following risk assessment matrix is used considering Impact and probability of risk in terms of ease of exploitation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk Assessment Matrix** | | | | |
| **Impact of Vulnerability - Consequence** | **Major** | High | Critical | Critical |
| **Moderate** | Medium | Medium | High |
| **Minor** | Low | Medium | Medium |
| **Risk Severity = Impact x Probability** | | **Hard** | **Moderate** | **Easy** |
| **Probability of Risk occurrence** | | |

Please note: Risk rating will also depend on the business criticality of the asset.

## **Observation Summary Table**

The chart given below represents the vulnerabilities found during Web application Penetration testing:

**Figure 1: Summary of Vulnerabilities**

The below table illustrates the distribution of observations of Web application Penetration testing based on the risk categorization i.e., Critical, High, Medium, and Low.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Domain** | **Critical** | **High** | **Medium** | **Low** | **Total** |
| Web Application Grey Box Testing | 0 | 4 | 5 | 12 | 21 |

**Revalidation Summary Table**

**Figure 2: Web Application Penetration Testing Vulnerabilities Revalidation Status**

The below table illustrates the revalidation status of vulnerabilities observed during Web Application Penetration testing.

|  |  |  |  |
| --- | --- | --- | --- |
| **Domain** | **Open** | **Closed** | **Total** |
| Web Application Grey Box Testing | **0** | **21** | **21** |

## **Snapshot of Observations**

This section provides an overview of observations identified in Web application Penetration testing security assessment for management reference. The detailed observations section provides a complete list of observations along with their individual risk categorization and suggested recommendations to mitigate the risks.

| **Vulnerabilities** | **Risk** | **Status** | **{Client Company} Comments** | **GT Comments** |
| --- | --- | --- | --- | --- |
| **Insecure direct object reference** | **High** | **Closed** | {client comment} | {GT Comment} |
| **Stored HTML injection** | **High** | **Closed** | {client comment} | {GT Comment} |
| **{vulnerability}** | **{Risk}** | **{status}** | {client comment} | {GT Comment} |

# **Detailed Observations**

The details of identified vulnerabilities, impact, severity, and recommendations for the same are explained below.

{{VULNERABILITIES}}

* 1. **Insecure direct object reference**

**Risk: High**

**Status: Open**

**Revalidation Status: Closed**

**Abstract:**

It was observed that the authorization header just using API authenticate sake not for authorization and *Team member id* parameter is the request where server know which profile it’s asking and it’s vulnerable to IDOR or BOLA.

**Impact:**

Insecure Direct Object References (IDOR) occur when an application provides direct access to objects based on user-supplied input. As a result of this vulnerability attackers can bypass authorization and access resources in the system directly, for example database records or files.

**Recommendations:**

It is recommended to implement access control validation.

**Affected URL:**

<https://www.google.com/scope> { Affected URL(s) }

**{Client Company} Team Justification:**

This vulnerability has been resolved.

**Artifact Submitted by {Client Company}:**

**A screenshot of a computer

Description automatically generated**

***Fig 1.1 Artifacts Given by {client company}***

**A screenshot of a computer

Description automatically generated**

***Fig 1.2 Artifacts Given by {client company}***

**Gt Team Comment:**

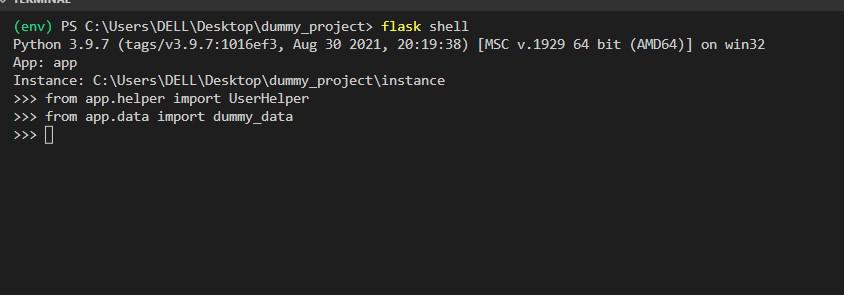
After revalidation we found that Insecure direct object reference has been remediated and no longer exists in the application.

**Proof of Concept:**

**A screenshot of a computer

Description automatically generated**

***FIGURE 1.4 PROOF OF CONCEPT***



***FIGURE 1.6 PROOF OF CONCEPT***

* 1. **Stored HTML injection**

**Risk: High**

**Status: Open**

**Revalidation Status: Closed**

**Abstract:**

Hypertext Markup Language (HTML) injection is a technique used to take advantage of non-validated input to modify a web page presented by a web application.

**Impact:**

Attackers take advantage of that the content of a web page is often related to a previous interaction with users. When applications fail to validate user data, an attacker can send HTML-formatted text to modify site content that gets presented to other users. A specifically crafted query can lead to inclusion in the web page of attacker-controlled HTML elements which change the way the application content gets exposed to the web.

.

**Recommendations:**

Following Fix are recommended:

1.Every input should be checked if it contains any script code or any HTML code. One should check, if the code contains any special script or HTML brackets – <script></script>, <html></html>.

2.There are many functions for checking if the code contains any special brackets."

**Affected URL:**

<https://www.google.com/scope> { Affected URL(s) }

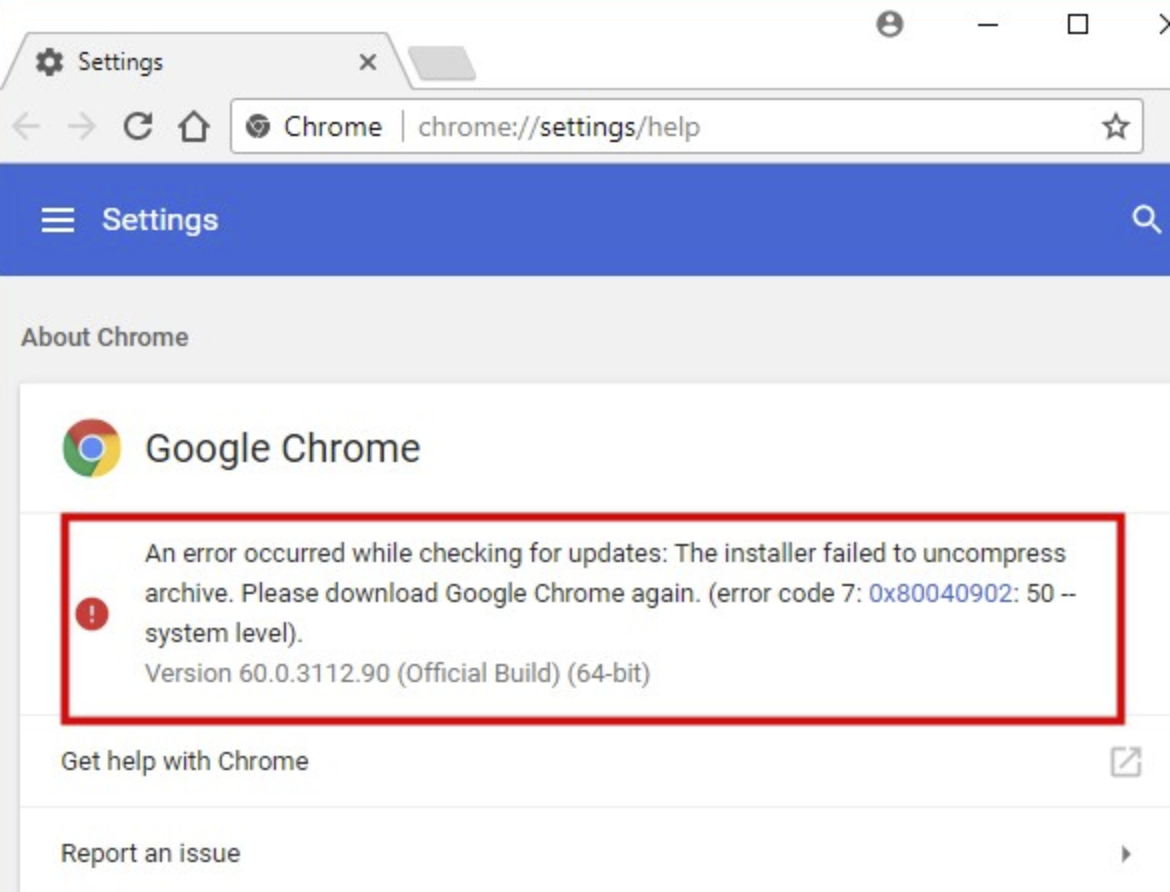
**{Client Company} Team Justification:**

This vulnerability has been resolved.

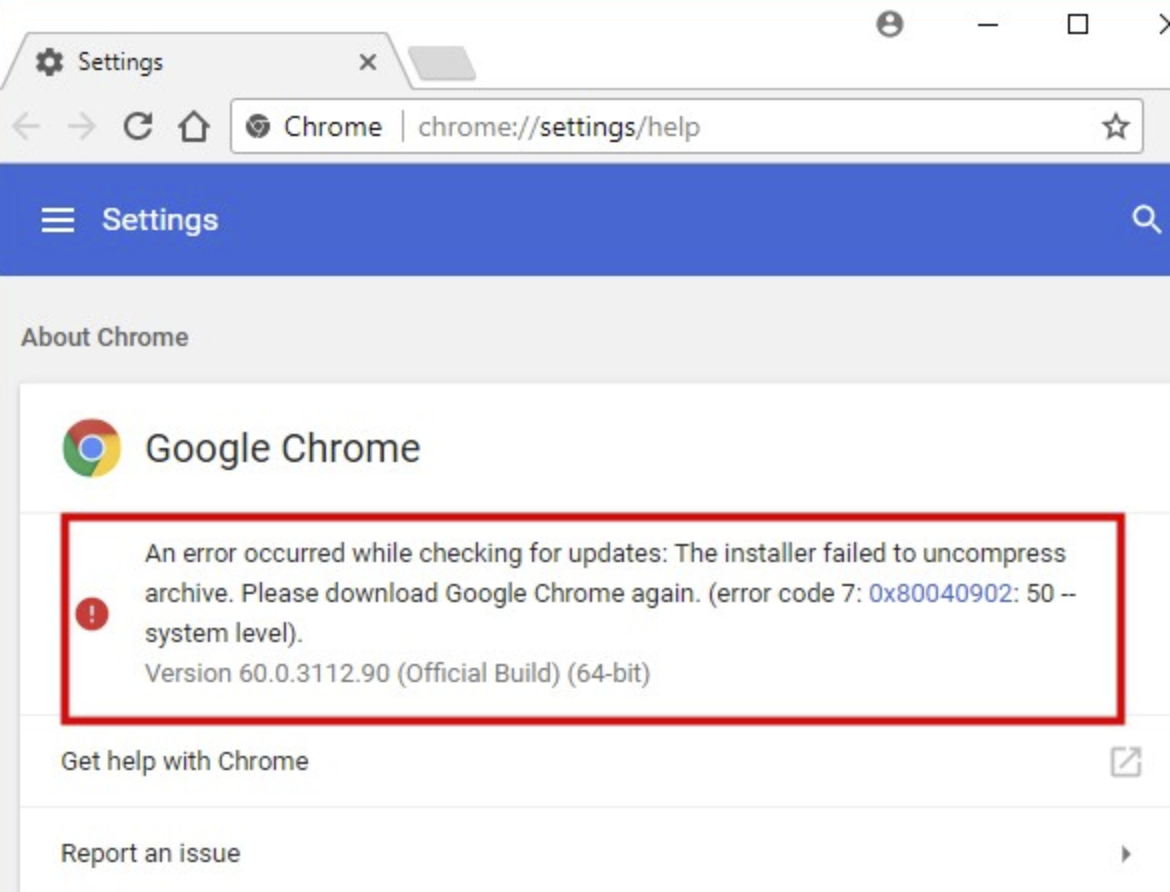
**Gt Team Comment:**

After revalidation we found that Stored HTML injection has been remediated and no longer exists in the application.

**Artifact Submitted by {client company}**

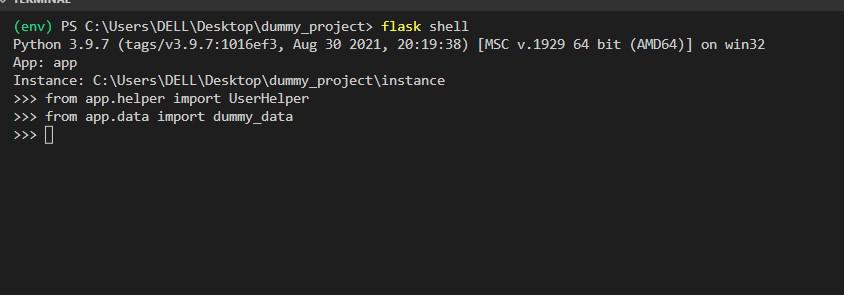


***FIGURE 2.1 Artifacts Given by {client company}***



***FIGURE 2.2 Artifacts Given by {client company}***

**Proof of Concept:**



***FIGURE 2.3 PROOF OF CONCEPT***

**A screenshot of a computer

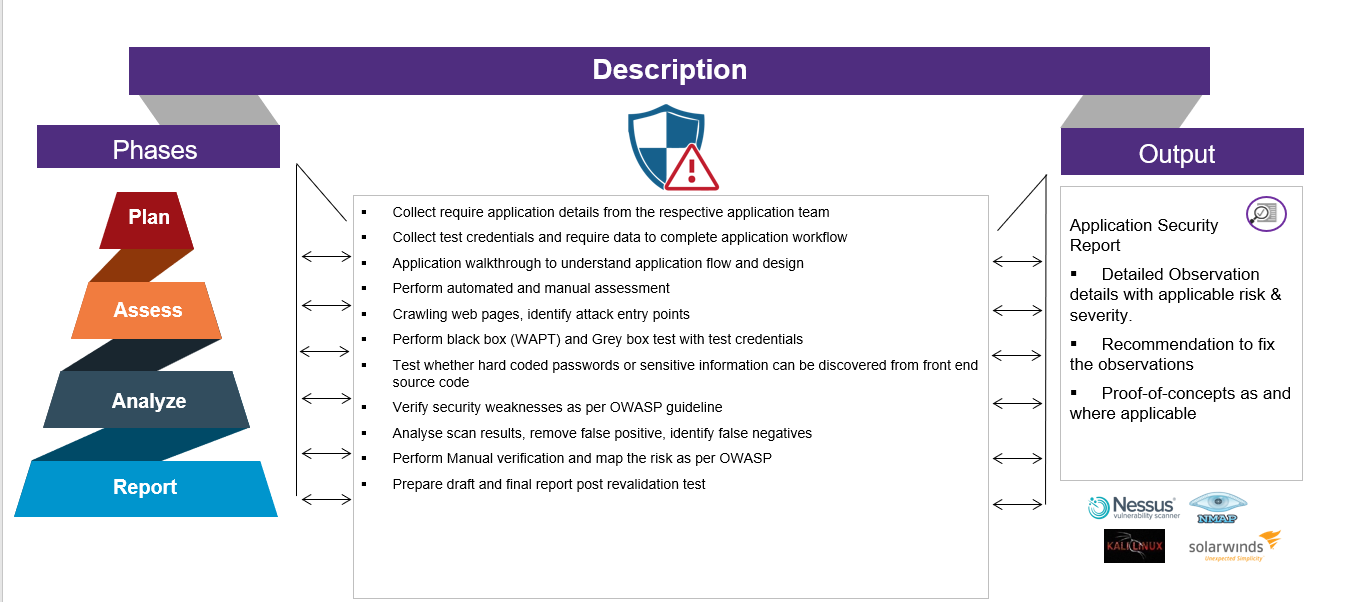
Description automatically generated**

***FIGURE 2.4 PROOF OF CONCEPT***

# **Annexure**

**Assessment Approach:**

The Web application security assessment was conducted as an exercise. This was done to simulate as closely as possible the viewpoint of a completely external attacker. Following approach is followed performing the assessment on the Web provided for testing.



**Web Application Penetration Testing Approach**

**Assumptions:**

Based on the scope, only the specified Web application was tested. This report has been produced based on the Web application security assessment & output of the scan that was conducted on a particular date tested. Vulnerability details provided in this report is based on the Web application security assessment performed on a production or identical to production environment.

It is recommended that prior to acting on the recommendations; following actions are assumed to be taken by {Client Comany} :

* Vulnerabilities identified were as on the date security assessment conducted and as per the scan policies selected (non-intrusive). Any vulnerabilities identified after GT assessment date may also not form part of this report.
* GT provided the reference link in the detailed vulnerability section are for {Client Comany} reference only.
* Any fix to application/system should be tested on UAT or non-production environment prior to any patch deployment on production environment.
* Appropriate backup and rollback plan are made prior to implementing the recommendation on the system.
* This report is intended solely for the information and internal use of {Client Comany}
* {Client Comany} team is responsible for applying security fixes and maintaining an effective security controls on application, network, and system.
* There are version specific vulnerabilities included in the report based on the outcome of the service enumeration. This needs to be validated by respective team to verify the correct version running on the system and apply the OEM specified patch (if version is vulnerable) after testing patch on non-production system and taking appropriate backup.

**End of Document**