2024 Q1 OneTrust Main Web App and API Penetration Test Re-Test

OneTrust

October 2024

Disclaimer

This report is intended solely for the use of management of OneTrust ("Client" or "OneTrust") and is not to be used or relied upon by others for any purpose whatsoever. This report and the related findings and recommendations detailed herein provide management with information about the condition or risks and internal controls at one point in time. Future changes in environmental factors and actions by personnel may significantly and adversely impact these risks and controls in ways that this report did not and cannot anticipate.

This report presents the results of a web application penetration test and API (Application Program Interface) penetration test performed by Protiviti between March 4th, 2024, and April 19th, 2024. **A re-test of the finding W.3 – Formula Injection was performed on October 16, 2024.** The scope of the review was limited to specific target systems which were agreed upon during project scoping. This executive summary report is designed for the reader to understand the level of security assessed, to identify security deficiencies, to identify areas of strength and weakness, and to develop a course of action to correct vulnerabilities and mitigate associated risks.

Penetration testing is an uncertain process which is based upon past experiences, currently available information, and known threats. It should be understood that all information security systems, which by their nature are dependent on their human operators, are vulnerable to some degree. Therefore, while the team believes to have identified the major security vulnerabilities on the systems analyzed, there can be no assurance that any exercise of this nature will identify all possible vulnerabilities or propose exhaustive and operationally viable recommendations to mitigate those exposures. This report identifies known vulnerabilities that were detected during the test period; new devices, configuration changes and new/future vulnerabilities were not tested. While the matters presented herein are the result of the review, had additional procedures been performed, other matters may have been identified that would have been reported to OneTrust.

Additionally, this report contains information concerning potential vulnerabilities of OneTrust network(s)/system(s) and methods for exploiting them. The team recommends that special precautions be taken to protect the confidentiality of both this document and the information contained herein.

Table of Contents

Executive Summary

Background

In Q1 of 2024, OneTrust ("Client" or "OneTrust") engaged a third-party, Protiviti to perform a web application penetration test and API penetration test. The project focused on evaluating controls that directly correlate to threats and risks that may compromise the confidentiality, integrity, and availability of sensitive information that resides on OneTrust's technology environment.

Fieldwork was performed remotely from Protiviti security labs between March 4th, 2024, and April 19th, 2024.

A re-test of the W.3 – Formula Injection finding was performed on October 16, 2024 and confirmed to be remediated.

Objectives and Scope

This engagement was executed with the intent of assessing controls that are in place within the applications and are designed to minimize the risks of the organization. Emphasis was placed on evaluating the application safeguards that restrict unauthorized access to the OneTrust application, the data it transmits, and the critical data (e.g., credentials, personally identifiable data, credit card information, etc.) it stores.

The scope of the engagement included the following:

- **Web Application Penetration Test** – Performed a series of tests on the OneTrust Main web application using automated commercial application scanning tools and web proxies to crawl and map the target. Specifically, application entry points, programming languages, structure, and error codes were identified to complete the application mapping process. Protiviti leveraged the data collected from the crawling and mapping phase to perform a series of automated tests, manual tests, and validation activities to evaluate the security posture of the application.
- **API Penetration Test** – Performed a series of tests against OneTrust Main's API using automated commercial application scanning tools and web proxies to crawl and map the associated endpoints. Protiviti leveraged endpoints collected from manual inspection and proxying of the front-end user interface (UI) as well as OneTrust provided documentation to perform a series of automated tests, manual tests, and validation activities to evaluate the security posture of the API endpoints.

Impact Summary

Penetration testing is a goal-driven exercise where Protiviti attempts to emulate a real-world attacker in order to obtain a specific objective. Therefore, Protiviti worked with OneTrust to establish the following goals and targets for this assessment:

Web Application Penetration Test

During the web application penetration test, Protiviti was unable to obtain the defined objectives.

Objectives:

- From an unauthenticated perspective, obtain access to any user-level application data or functionality. **(Not Achieved)**
- Compromise a user-level account or data from another user's account. **(Not Achieved)**
- Achieve Administrator-level compromise by escalating privileges to perform privileged functions. **(Not Achieved)**
- Achieve remote code execution on the application's supporting infrastructure. **(Not Achieved)**
- Impact the availability of the application for some or all users. **(Not Achieved)**

API Penetration Test

During the API penetration test, Protiviti was unable to obtain the defined objectives.

Objectives:

- From an unauthenticated perspective, obtain access to any user-level application data or functionality. **(Not Achieved)**
- Compromise a user-level account or data from another user's account. **(Not Achieved)**
- Achieve Administrator-level compromise by escalating privileges to perform privileged functions. **(Not Achieved)**
- Achieve remote code execution on the application's supporting infrastructure. **(Not Achieved)**
- Impact the availability of the application for some or all users. **(Not Achieved)**

Summary of Results

The following provides a summary of results by phase for the engagement:

Web Application Penetration Test

The OneTrust Main application implemented sound access control checks, live client-side input validation monitoring and performed server-side validation. Additionally, by including robust Content-Security Policies, the application significantly reduced the execution of malicious scripts on the OneTrust Main domain. These safeguards resulted in minimal findings as Protiviti discovered four (4) low priority issues listed below, two (2) of which were discovered during the test from June 2023 and two (2) of which are new.

- **W.1 – HTML Injection** – The OneTrust Main application allows the submission of HTML tags within certain input fields, which allows an attacker to inject scripts to run client-side in the user's browser. While these scripts are constrained by the Content-Security Policy (CSP) from making most outbound data requests, alternative exfiltration tactics, such as loading scripts from external storage blobs are not categorically ruled out by these measures. This was previously discovered during the test from June 2023.
- **W.2 – Use of Insecure Third-Party Libraries** - The OneTrust Main web application utilizes outdated third-party library JavaScript libraries known to be vulnerable to Cross-Site Scripting (XSS). This is a new finding from this year's testing.
- **[Remediated] W.3 – Formula Injection** – The OneTrust Main application allows attackers to inject spreadsheet formulas that OneTrust will in-turn inject into exported spreadsheet files without proper sanitization, which can lead to formula injection. This vulnerability permits attackers to manipulate data, execute unintended commands, or compromise the integrity of the spreadsheets and associated systems. This was previously discovered during the test from June 2023. **Additionally, a re-test was performed on October 16, 2024 and confirmed to be remediated.**
- **W.4 – Information Disclosure** – The OneTrust Main application discloses sensitive information within its HTTP response headers and body; specifically, the exposure of internal IP addresses, WebSocket, NATS URLs and server banner. This disclosure provides attackers with valuable information about internal messaging systems and the technology stack, increasing the risk of targeted attacks and data exfiltration. While the internal IP addresses were previously discovered during the December 2023 test, the remaining findings are new.
- **W.5 – Lack of File input validation** – The OneTrust Main web application allows files to be uploaded without proper input validation, permitting potentially malicious files to be stored within the application's Azure Storage blobs. This was previously noted during the test from June 2023.

API Penetration Test

During the API penetration test, Protiviti was unable to identify any issues affecting in-scope assets. Our testing did not reveal vulnerabilities such as authentication flaws, data leakage, or broken access controls, indicating that the API's current security measures are effectively configured to mitigating the commonly identifiable vulnerabilities associated with API's.

Strategic Recommendations

The following high-level recommendations are provided to help OneTrust strategically mitigate the risks identified in this report:

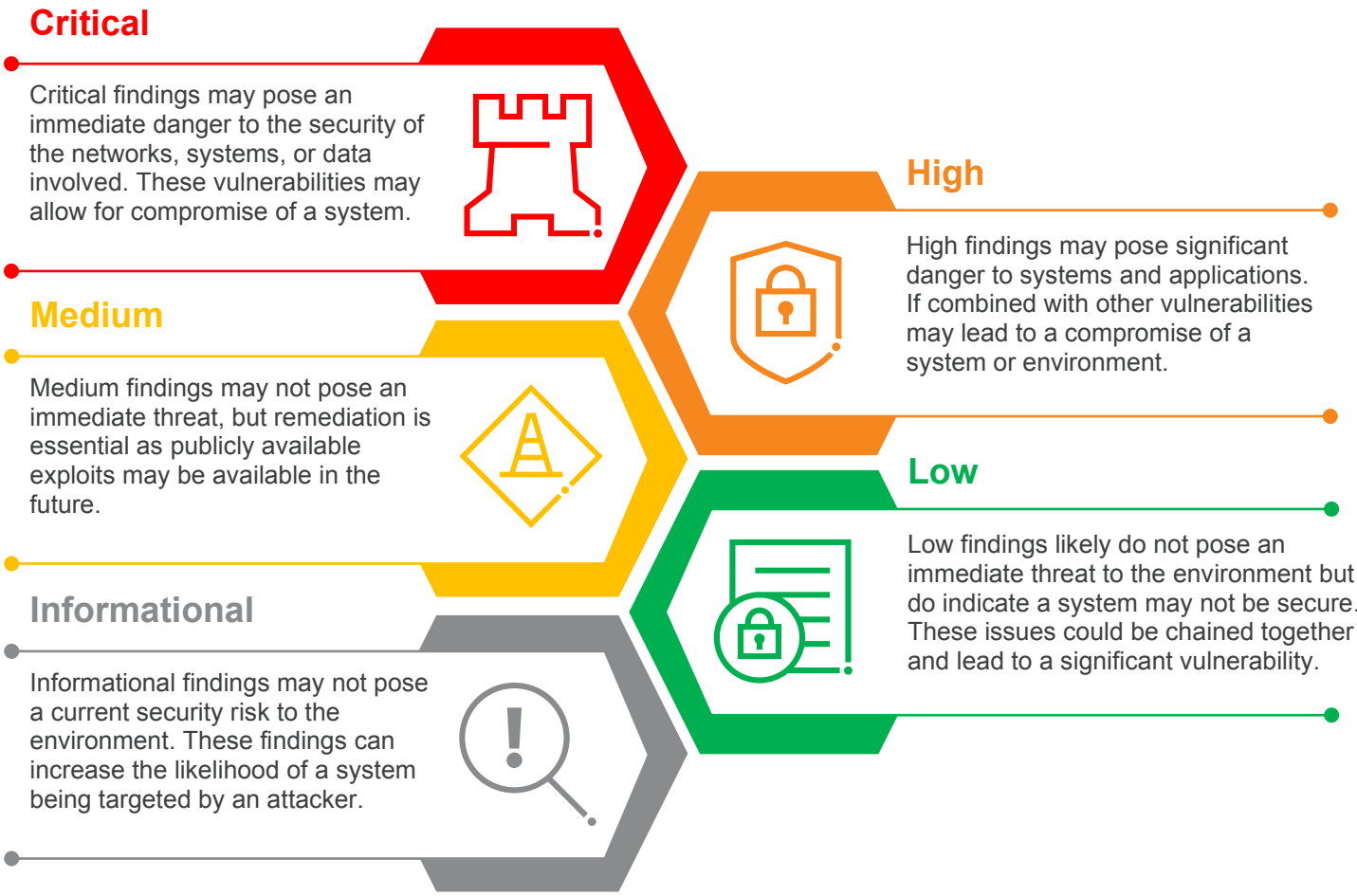
- **Enhance Secure Coding Practices:** Continue working with developers to validate they are following secure coding best practices and techniques with a specific emphasis on input validation, output sanitization, and handling of HTML tags to prevent HTML Injection. Additionally, ensure that no formula injection vulnerabilities exist by properly escaping or sanitizing inputs used in formulas. Implement thorough input validation mechanisms for file uploads to prevent malicious files from being accepted. Finally, implement a process whereby the application is subject to regular secure code reviews using automated tools and manual methods as part of the development cycle.
- **Update Data Classification Procedures:** Review data security standards to ensure they include definitions of non-public information that would address items like internal technical information. Ensure that data handling procedures require documents and other content released outside the organization to be scrubbed of metadata, to prevent disclosing valuable information to attackers. In this scenario, consultants discovered the disclosure of WebSocket and NATS URLs, which can enable direct access to internal messaging systems, as well as internal IP addresses and server banner disclosure, revealing information about the technology stack that can be used for more targeted attacks.

Many of Protiviti's recommendations contain instructions for specific system configuration changes (e.g., version upgrade), and these recommendations should be properly evaluated and tested in a non-production environment prior to implementation on any production systems.

Protiviti's Approach to Evaluating Results

Observations and recommendations made during this review have received one of the following risk rankings. These rankings address the significance of the risk and likelihood the risk could occur in the business environment. The rankings should be reviewed by management and used as a tool to determine the level of attention and effort that should be given to each observation and recommendation.

Each vulnerability or risk identified has been labeled with a particular significance rating of critical, high, medium, low, or informational risk levels, defined as follows:



Summary of Observations

The following table summarizes the total number of findings in each ranking:

Phases	Critical	High	Medium	Low	Info
Web Application Penetration Test	0	0	0	4	1
API Penetration Test	*No Associated Findings*				
Total	**0**	**0**	**0**	**4**	**1**

The table below summarizes the observations identified during the 2024 Q1 OneTrust Main Web App and API Penetration Test:

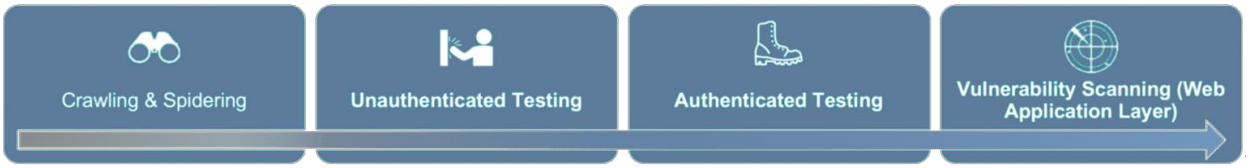
Ref.	Observation	Criticality	Remediated?
Web Application Penetration Test			
W.1	HTML Injection	**Low**	**N/A**
W.2	Use of Insecure Third-Party Libraries	**Low**	**N/A**
W.3	Formula Injection	**Low**	**Yes**
W.4	Lack of File Input Validation	**Low**	**N/A**
W.5	Information Disclosure	**Info**	**N/A**
API Penetration Test			
No Associated Findings			

Appendix A - Testing Methodologies

Web Application Penetration Test Methodology

Overview: During web application penetration testing, Protiviti attempts to identify insecure configurations, failures in business logic, and exploitable vulnerabilities that a malicious actor could abuse. Protiviti leverages found vulnerabilities to gain unauthorized access to sensitive information being handled by the application, to gain elevated or privileged access to the application itself, and/or to gain access to the web application's underlying infrastructure (e.g., the server hosting the web application). Protiviti initially tests from an unauthenticated perspective to simulate an attacker who does not have valid credentials for the application. Additionally, Protiviti continues testing with valid credentials and attempts to identify privilege escalation vectors and other attacks that could be performed from an authenticated perspective.

Steps listed in this section detail the general flow of activities and the type of tasks performed during web application penetration tests.



Crawling and Spidering: Protiviti profiles or "footprints" the in-scope web applications. This includes "crawling" and "spidering" the applications to identify pages and resources available to end users. Protiviti also attempts to identify pages or resources not directly referenced by the application. In addition, Protiviti attempts to identify the underlying technology utilized by the application (e.g., programming languages, libraries, frameworks, etc.).

Purpose: The purpose of application mapping is to build an inventory of the application's static and dynamic pages and to identify the application's architecture. This information is leveraged in later phases to identify attack vectors and vulnerabilities.

Result: Protiviti did not identify any issues during this phase.

Unauthenticated Testing: Protiviti leverages the data collected during the application mapping phase to perform a series of tests to evaluate the security posture of the application from an unauthenticated perspective. Protiviti attempts to identify application misconfigurations, coding errors, and other vulnerabilities, especially those outlined in the OWASP Top 10. Common web application vulnerabilities include Structured Query Language (SQL) injection, cross-site scripting (XSS), broken authentication/access controls, etc. Tests are also performed to better understand the functionality of the in-scope applications, which aids in identifying exploitable lapses in application logic.

Purpose: The purpose of unauthenticated testing is to evaluate the security posture of the application against an attacker who has not compromised valid credentials.

Result: Protiviti did not identify any issues during this phase.

Authenticated Testing: Protiviti performs testing of the web application(s) from the perspective of an authenticated user. Protiviti specifically tests the authenticated areas of the application and focuses on identifying issues like session hijacking, privilege escalation, authentication bypass, access control deficiencies, or unauthorized account "hopping."

Purpose: Performing authenticated testing of the in-scope application(s) provides an understanding of what an attacker with compromised credentials or a malicious user could potentially exploit.

Result: Protiviti identified the following issues:

- W.1 - HTML Injection
- W.2 - Use of Insecure Third-Party Libraries
- W.3 - Formula Injection
- W.4 - Information Disclosure
- W.5 - Lack of File Input Validation

Vulnerability Scanning (Web Application Layer): Protiviti uses automated vulnerability scanners to identify exploitable web application vulnerabilities. Automated vulnerability scanners identify thousands of known vulnerabilities while also attempting common attack vectors. Protiviti then performs false-positive analysis to verify discovered vulnerabilities.

Purpose: The objective of automated vulnerability scanning is to identify as many potential vulnerabilities as possible and to augment manual testing to provide a comprehensive view of the security posture of the in-scope web application(s).

Result: Protiviti did not identify any issues during this phase.

Appendix B - Testing Scope

Web Application Penetration Test
• https://pentest-app.onetrust.com/

API Penetration Test
• https://pentest-app.onetrust.com/api/*