



SAMPLE REPORT (BRIEF)  
2020.08.01 (v1.0)

## Disclaimer

---

This report is a sample report for demonstration purpose only. All findings in this report are not real findings from any existing company but were written based on vulnerabilities labo00 LLC had identified from our project experience in order to show our strength and abilities. Customers could use this as a reference to choose their assessment service.

## Using This Report

This document has been divided into clearly marked sections so as to assist the readers to locate the information most relevant to themselves. In essence, these main sections are:

- **Overview** provides a brief synopsis of the assessment from a high-level perspective, including a complete test scope and risk distribution.
- **Detailed Findings** provides a thorough technical discussion, including reproducible steps (if applicable) and recommendations, for each individual security issue which was identified during the assessment.
- **Methodology** outlines the assessment-specific methodologies used by the test team.
- **Appendix** presents a summary of the security tools used during the engagement and the assessment team members. Any additional technical information or evidence which was too verbose to include in “Detailed Findings” section will also be included here.

## Client Confidentiality

This report has been made for ABC Limited and contains Client Confidential information. It is not for public dissemination and may not be copied without explicit written consent.

## Proprietary Information

This report contains proprietary and confidential information and should not be disclosed outside of the Client’s organization without permission.

Labo00 LLC is permitted to copy this report for the purposes of facilitating review and discussion on the results within Client’s organization and/or third parties who are the contributors of the test.

# Table of Contents

---

<b>CONTENT</b>	<b>PAGE</b>
<b>1. Overview</b>	5
<b>1.1 Report Information</b>	5
<b>1.2 Risk Distribution</b>	6
<b>1.3 Table of Findings</b>	6
<b>1.3.1 Summary - ABC Web Portal</b>	6
<b>2. Detailed Findings</b>	7
<b>3. Methodology</b>	8
<b>3.1 Type of Tests</b>	8
<b>3.2 Web Application Assessment Methodology</b>	9
<b>3.3 Risk Level Classifications</b>	11
<b>4. Appendix</b>	12
<b>4.1 Tools List</b>	12
<b>4.2 Assessment Team</b>	12
<b>4.3 OWASP Top 10 Web Application Security Risks (2017)</b>	12

# 1. Overview






---

## 1.1 Report Information

ITEM	DETAILS
REPORT NAME	Sample Assessment Report (Brief)
RELEASE DATE (VERSION)	2020.08.01 (v1.0)
PREVIOUS RELEASE (VERSION)	-
CLIENT NAME	ABC Limited
TEST TYPE	Web Application Assessment
TARGET	<a href="https://www.idontexist.com/">https://www.idontexist.com/</a>
ENVIRONMENT	Production
METHOD	Grey-box
TEST ACCOUNTS (ROLE/PRIVILEGE)	<a href="#">testuser1</a> (Basic User) <a href="#">admin</a> (Administrator)
ASSESSED FROM	Remote (labo00's office at Tokyo, Japan)
TEST DATES	2020.07.27 to 2020.07.31
TEST IPs	Remote (116.58.191.150)
REMARKS	-
REPORT WRITER	Satou Hiroshi

## 1.2 Risk Distribution

The tables and pie charts below show the number of vulnerabilities identified and their severity. Overall risk levels of tested targets were deduced from the findings.

RISK LEVEL		ABC Web Portal
CRITICAL		1
HIGH		1
MEDIUM		0
LOW		1
INFO		0
TOTAL		3



## 1.3 Table of Findings

The tables below show the vulnerabilities identified (classified based on OWASP Top 10 2017) and their severity.

### 1.3.1 Summary - ABC Web Portal

RISK ID	RISK CATEGORY	RISK LEVEL
A1	Other	CRITICAL
A2	Broken Access Control	HIGH
A3	Security Misconfiguration	LOW

## 2. Detailed Findings

---

Not shown in brief report.

## 3. Methodology

---

Based on Client's requirements and test scenarios, different methodologies can be used in penetration test. The detail of the methodology used in this assessment is described below.

### 3.1 Type of Tests

The test can be conducted in black-box or grey-box approach. According to Open Source Security Testing Methodology Manual (OSSTMM), the two types of tests are defined as follows:

TYPE	DESCRIPTION
<b>BLACK-BOX (BLIND)</b>	The pentester engages the target with no prior knowledge of its defenses, assets, or channels. The target is prepared for the audit, knowing in advance all the details of the audit. A blind audit primarily tests the skills of the pentester. The breadth and depth of a blind audit can only be as vast as the pentester's applicable knowledge and efficiency allows.
<b>GREY-BOX</b>	The pentester engages the target with limited knowledge of its defenses and assets and full knowledge of channels. The target is prepared for the audit, knowing in advance all the details of the audit. A grey-box audit tests the skills of the pentester. The nature of the test is efficiency. The breadth and depth depend upon the quality of the information provided to the pentester before the test as well as the pentester's applicable knowledge.

For more details, please refer to OSSTMM v3.



## 3.2 Web Application Assessment Methodology

<p><b>Phase 1 - Information Gathering</b></p>	<p>The penetration test was started by collecting information of the target application from various sources. The information, which was publicly available on the internet, includes the network infrastructure, domain name service, security systems in use, open services, etc. The information would help the consultants to understand the target environment and plan for further assessment.</p>
<p><b>Phase 2 - Vulnerability Identification &amp; Prioritization</b></p>	<p>After information gathering, assessors would try to identify any vulnerability on the application through automated scanning tools and manual inspection.</p> <p>Numerous test cases were conducted.</p> <p>For web applications, the OWASP Top 10 vulnerabilities would be covered.</p> <p>For mobile applications, the Mobile Top 10 were referenced during the assessment. In particular, the assessment will cover the following aspects:</p> <ul style="list-style-type: none"> <li>• Client-side attacks <p>Examine if there is any insecure data handling, such as unencrypted data storage for sensitive information, insecure file caching, etc.</p> </li> <li>• Network-side attacks <p>Examine the application traffic between the mobile apps and server and identify if there is any information leakage (e.g. personal data leakage, etc.)</p> </li> <li>• Server-side attacks <p>Examine the application traffic between the mobile applications and backend server and identify if there is any possible data manipulation between the client and server (e.g. injection, data tampering, session hijacking, etc)</p> </li> </ul> <p>The risks discovered will be correlated with the results in static security assessment to produce a more accurate result. The results were collected, reviewed, and prioritized for further exploit.</p>
<p><b>Phase 3 - Research &amp; Development</b></p>	<p>In this phase, the consultants conducted research on the vulnerabilities identified on the target application and developed the attack approaches, tools, scripts, etc and prepared for exploiting the vulnerabilities.</p>

**Phase 4 -  
Exploitation**

With the findings in the research and development phase, the consultants would then carry exploits on the target. This phase involved the use of real-world hacker tools and scripts to simulate attacks on the vulnerabilities. In this phase, a higher level of privileged or access to sensitive information can be achieved.

**Phase 5 -  
Post-Exploitation**

After exploitation, the consultants might gain privileged access to the target application. the consultants would explore further opportunities to see if it is possible to access other systems through the privileged access.

**Phase 6 -  
Risk Analysis and  
Reporting**

The results of the penetration test were documented in detail in this report. The risk rating of each vulnerabilities was assessed. The result and the recommendations for remediation will be documented in the report. To cater for different readers, the report will be clearly sectioned to consist of executive-level reporting and technical reporting. Labo00 shall endeavor to produce a report that is concise, well-structured and contain of solid recommendations and reproducible results.

### 3.3 Risk Level Classifications

This section of the report details the severity classification system used during the assessment:

SEVERITY RATING	DESCRIPTION
<b>CRITICAL</b>	These issues imply an immediate, easily accessible threat of large-scale total compromise. As such, they should be resolved as a matter of urgency to ensure the business is not operating with an excessive level of IT related business risk.
<b>HIGH</b>	These issues imply an immediate threat of system compromise. As such, they should be resolved as soon as possible to ensure the business is not operating with an excessive level of IT related business risk.
<b>MEDIUM</b>	These issues should be resolved in a timely manner where possible; however, they can often be mitigated in the short term until appropriate resolutions can be put in place.
<b>LOW</b>	These issues should be resolved if the improvement in the organization's security posture would justify the cost of the solution. In general, solutions to low severity issues should be implemented once higher severity issues have been addressed.
<b>INFO</b>	These issues are included in the report for completeness.

## 4. Appendix

---

### 4.1 Tools List

Not shown in sample report.

### 4.2 Assessment Team

Not shown in sample report.

### 4.3 OWASP Top 10 Web Application Security Risks (2017)

CATEGORY	DESCRIPTION
<b>Injection</b>	Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.
<b>Broken Authentication</b>	Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.
<b>Sensitive Data Exposure</b>	Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.
<b>XML External Entities (XXS)</b>	Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.
<b>Broken Access Control</b>	Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality

	and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.
<b>Security Misconfiguration</b>	Security misconfiguration is the most commonly seen issue. This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched/upgraded in a timely fashion.
<b>Cross-Site Scripting XSS</b>	XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.
<b>Insecure Deserialization</b>	Insecure deserialization often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.
<b>Using Components with Known Vulnerabilities</b>	Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.
<b>Insufficient Logging &amp; Monitoring</b>	Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data. Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.
<b>Other</b>	Vulnerabilities do not fall into above 10 categories.