bsi.

...making excellence a habit.™

# Web Application Test Report

Dragon InfoSec
**Dragon Wallet**

BSI Reference: IA14347-RPT-01

Version: 1.0

**21st September 2018**

**Testing Team**

Mark Woan

BSI Cybersecurity and Information Resilience (UK) Limited
Kitemark Court, Davy Avenue, Knowlhill, Milton Keynes MK5 8PP
Telephone: +44 345 080 9000 Email: cyber@bsigroup.com

INVESTORS IN PEOPLE

# Document Control Information

## 1.1 Document Details

| Property | Value |
|---|---|
| **Client** | Dragon InfoSec |
| **Title** | Dragon InfoSec Web User Interface Web Application Test Report |
| **Author** | Mark Woan |
| **Version** | 1.0 |
| **Date** | 21/09/2018 |
| **Document Reference** | IA14347-RPT-01 |
| **Status** | Definitive |

## 1.2 Revision History

| Version | Date | Author | Summary of Changes |
|---|---|---|---|
| 0.1 | 19/09/2018 | Mark Woan | Initial Draft |
| 0.2 | 21/09/2018 | Christian Hobbs | Internal QA |
| 1.0 | 21/09/2018 | Christian Hobbs | Definitive Issue |

## 1.3 Approvals

| Name | Organisation | Role |
|---|---|---|
| Peter Viranyi | BSI Cybersecurity and Information Resilience | Head of Security Testing |

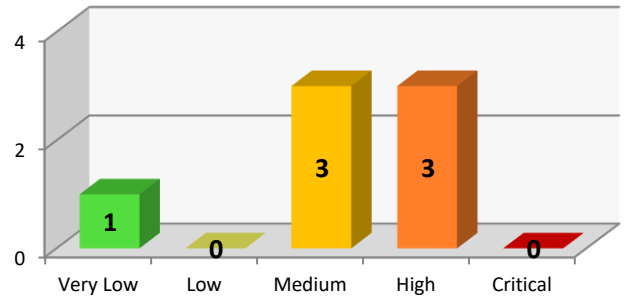
## 1.4 Distribution

| Name | Organisation | Role |
|---|---|---|
| Richard Dennis | Dragon InfoSec | CTO |

# Table of Contents

# 2 Executive Summary

BSI Cybersecurity and Information Resilience were engaged by Dragon InfoSec to perform a web application test against the Dragon Wallet application. Testing was undertaken remotely between the 17th and 18th September 2018.




This graph illustrates the level of risk that is exposed across the application tested. It shows the number of vulnerabilities identified during this assessment along with their severity.

As can be seen from the graphs above, high and medium areas of risk have been identified within the application. Of the issues identified three were of high risk, these issues are summarised below:

The application used a clear text method to communicate between the user's web browser and the server hosting the application. Due to the lack of encryption used by the application, an attacker who is able to monitor a session would be able to view all of the authentication credentials and data transmitted in the session.

The application did not correctly sanitise user input which could potentially allow an attacker to gain access to a user's session and perform actions as the targeted user.

The application login page did not protect against brute force password guessing attacks. If an attacker successfully guesses the victim's password they will gain access to the victim's account, take control of the account and potentially deny the victim access to the application

# 3 Introduction

BSI Cybersecurity and Information Resilience were engaged by Dragon InfoSec to perform a web application test against the Dragon Wallet application. Testing was undertaken remotely between the 17th and 18th September 2018.

## 3.1 Approach

All testing was carried out using BSI Cybersecurity and Information Resilience standard testing methodology. A full copy of this methodology can be provided on request.

## 3.2 Scope

The scope of the engagement was as follows:

- Perform security testing of the web application
- Ensure that users' private keys could not be exposed
- Ensure that transactions can only be performed with authorised accounts and valid balances

### 3.2.1 Limitations

The following limitations were identified:

- The application was under active development during testing, and therefore other issues may arise as the implementation changes
- The following areas were not tested as they appeared incomplete or unavailable; Two-Step-Verification, Smart auth protection, Payment password

# 4 Results of Web Application Testing

This section provides the detailed findings of the web application testing that was performed between the 17th and 18th September 2018.

## 4.1 Application Uses Clear Text HTTP Protocol

| Finding No 1. | | |
|---|---|---|
| | Systems Affected | `http://54.38.214.253` |
| | Finding | The web application was available over unencrypted HTTP protocol. |
| | CVE Number | N/A |
| | Root Cause | **Web Development** |
| | Impact | **5**  |
| | Likelihood | **3**  |
| | Overall Risk Rating | **15 (High Risk)** |
| | Status | **ONGOING** |

### 4.1.1 Summary

The clear text HTTP protocol provides no encryption of the communication between the web browser and the application, including any authentication details and data transferred.

### 4.1.2 Technical Details

Due to the lack of encryption provided by the HTTP service, an attacker who is able to monitor a session would be able to view all of the authentication credentials and data transmitted in the session. The attacker could then attempt to gain unauthorised access to the application using the authentication credentials extracted from the session and potentially gain access under the context of that user.

Further information relating to this issue can be found in the following OWASP document:

https://www.owasp.org/index.php/Transport_Layer_Protection_Cheat_Sheet

### 4.1.3 Recommendation

BSI recommends that the HTTP service be replaced with the cryptographically secure alternative HTTPS.

## 4.2 Stored Cross-Site Scripting

| Finding No 2. | | |
|---|---|---|
| | Systems Affected | http://54.38.214.253 |
| | Finding | A stored Cross-Site Scripting (XSS) vulnerability was identified which could be exploited by an authenticated user. |
| | CVE Number | CWE-79 |
| | Root Cause | **Web Development** |
| | Impact | **4**  |
| | Likelihood | **3**  |
| | Overall Risk Rating | **12 (High Risk)** |
| | Status | **ONGOING** |

### 4.2.1 Summary

Stored Cross-Site Scripting vulnerabilities occur when data entered by one user is stored within the application and then later displayed to other users without being sufficiently filtered or validated. A common scenario which may present this vulnerability would be a forum where users can submit their own posts.

This vulnerability would be exploited by an attacker by entering malicious code into a page which is then stored within the web application. A victim would then navigate to the page through normal use of the application and the malicious script would execute in the user's browser within their security context.

An attacker who successfully exploits this issue could hijack application user accounts and run malicious code on the client machines.

### 4.2.2 Technical Details

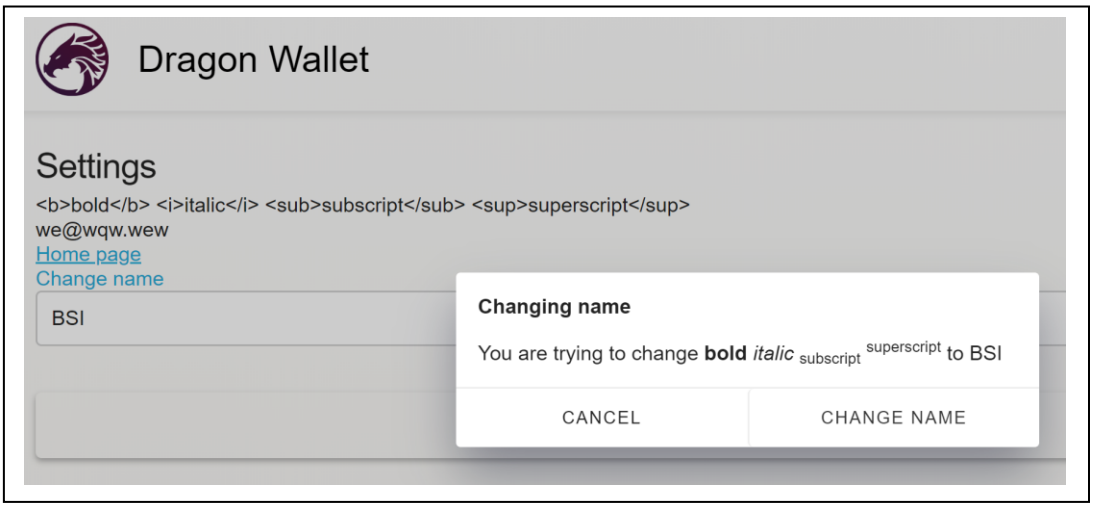
Stored Cross-Site Scripting vulnerabilities were identified in the **change name** functionality. The application attempted to validate the user input, however, the validation was only performed on the client-side.

An example of a HTTP POST request (edited for brevity) used to submit a name value with invalid characters is shown below:

```
POST /user/change/name HTTP/1.1
Host: 54.38.214.253:9001
Content-Length: 15
Accept: application/json, text/plain, */*
Origin: http://54.38.214.253
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjViODkzNGViNGE1ZDI3MDhiZmNhMmVjMCIsImV4cCI6MTUzNzE5M
Tc2MSwiaWF0IjoxNTM3MTkwODYxfQ.1nEisiHuL0SX_qIm-vodBP74WqaA_4RwsPGaH4XOp_E
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/68.0.3440.106 Safari/537.36
Content-Type: application/json;charset=UTF-8
Referer: http://54.38.214.253/
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Connection: close

{"name":"<b>bold</b> <i>italic</i> <sub>subscript</sub> <sup>superscript</sup>"}
```

The result of the injected HTML characters can be seen in the screenshot below:




The instance of XSS identified required that the user was authenticated to the application, reducing the likelihood of an unauthorised attacker exploiting this vulnerability. The issue could potentially be used to perform an attack against administrative users that have visibility of the user's name data.

Successful exploitation of the most critical of these vulnerabilities could allow an attacker to gain valid credentials to the application.

### 4.2.3 Recommendation

BSI recommends that all client supplied input is sufficiently filtered before being echoed back to the client's browser. Input validation should be carried out on all input fields to ensure that only input that matched an expected pattern is accepted. In addition, the application should implement output encoding to ensure that any potential unsafe data is properly encoded or escaped to prevent execution within the client's browser.

Further information relating to this issue can be found in the following OWASP documents:

https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)

https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet

## 4.3 No Account Lockout

| Finding No 3. | | |
|---|---|---|
| | Systems Affected | `http://54.38.214.253` |
| | Finding | The application did not enforce an account lockout policy |
| | CVE Number | N/A |
| | Root Cause | **Web Development** |
| | Impact | **4**  |
| | Likelihood | **3**  |
| | Overall Risk Rating | **12 (High Risk)** |
| | Status | **ONGOING** |

### 4.3.1 Summary

The application login page does not protect against brute force password guessing attacks

### 4.3.2 Technical Details

It was possible to attempt unlimited attempts to login to a given user account.

An attacker could use this vulnerability in order to carry out a password brute-force attack in order to obtain the correct password for known user accounts. If successful, this would allow the attacker to gain unauthorised access to the application and its data under the context of the compromised user account.

The password policy was enforced with the following parameters:

- Eight characters minimum length
- Mix of lower- and upper-case alphanumeric characters
- Mix of numeric and alphanumeric characters, and special characters

### 4.3.3 Recommendation

BSI recommends that an account lockout policy be enforced for incorrect login attempts in line with the organisation's password policy requirements.

An example of a secure lockout policy is:

- Lockout after 3-5 unsuccessful attempts.

The exact policy should be based on the sensitivity of the application.

BSI highly recommends that Two Factor or Multi-Factor authentication be implemented and enforced for all logins and transaction authorisations.

## 4.4 Unverified Password Change

| Finding No 4. | | |
|---|---|---|
| | Systems Affected | http://54.38.214.253 |
| | Finding | User password could be updated without verifying the current user |
| | CVE Number | N/A |
| | Root Cause | **Web Development** |
| | Impact | **4**  |
| | Likelihood | **2**  |
| | Overall Risk Rating | **8 (Medium Risk)** |
| | Status | **ONGOING** |

### 4.4.1 Summary

The password change functionality permitted the modification of the user's current password without verifying the authenticity of the user.

### 4.4.2 Technical Details

The application allows an account's password to be changed whilst authenticated without requiring the existing (old) password to be entered, i.e. without verifying the authenticity of the user.

Therefore, if the account is temporarily compromised e.g. via SQL Injection; XSS; CSRF; session hijacking; or similar, the attacker would be able to change the password on the account without requiring any knowledge of the existing password. As such they would lock out the legitimate user and effectively seize control of the account.

### 4.4.3 Recommendation

BSI recommends that the application be modified to require users to input the current password of the account prior to permitting the password change. The server should then use the password submitted to verify the authenticity of the user submitting the request and check that they are permitted to make such a change.

Best practice would be for the request to be given a threshold number of password change requests before temporarily locking the account in order to prevent attackers from brute-forcing the password on such a request.

## 4.5 Unverified Email Address Change

| Finding No 5. | | |
|---|---|---|
| | Systems Affected | http://54.38.214.253 |
| | Finding | User email address could be updated without verifying the current user |
| | CVE Number | N/A |
| | Root Cause | **Web Development** |
| | Impact | **4**  |
| | Likelihood | **2**  |
| | Overall Risk Rating | **8 (Medium Risk)** |
| | Status | **ONGOING** |

### 4.5.1 Summary

The password change functionality permitted the modification of the user's current email address without verifying the authenticity of the user.

### 4.5.2 Technical Details

The application allows an account's email address to be changed whilst authenticated without requiring the users current password to be entered, i.e. without verifying the authenticity of the user.

Therefore, if the account is temporarily compromised e.g. via SQL Injection; XSS; CSRF; session hijacking; or similar, the attacker would be able to change the email address on the account without requiring any knowledge of the existing password. The email account could subsequently be used to seize control of the account.

### 4.5.3 Recommendation

BSI recommends that the application be modified to require users to input the current password of the account prior to permitting the email change. The server should then use the password submitted to verify the authenticity of the user submitting the request and check that they are permitted to make such a change.

BSI further recommends that the application be modified verify email account ownership and to send an email to the existing (old) email account to provide visibility of the change.

## 4.6 User Account Enumeration

| Finding No 6. | | |
|---|---|---|
| | Systems Affected | http://54.38.214.253 |
| | Finding | The application was found to present error messages that facilitated the enumeration of valid user accounts. |
| | CVE Number | N/A |
| | Root Cause | **Web Development** |
| | Impact | **2**  |
| | Likelihood | **3**  |
| | Overall Risk Rating | **6 (Medium Risk)** |
| | Status | **ONGOING** |

### 4.6.1 Summary

It was possible to carry out user enumeration using the error messages returned by the login, forgotten password and sign up pages.

### 4.6.2 Technical Details

It was possible to carry out user enumeration using the error messages returned by the login, forgotten password and sign up pages. Using the difference in error messages returned, it was possible to determine valid user accounts. An attacker could leverage this vulnerability in order to identify user accounts, which could then be targeted in future attacks, such as brute force password guessing, once their accounts have been identified.

Testing of the application identified user enumeration was possible on the following pages:

- Login
- Forgotten password
- Sign up

The following indicates the revealing error messages returned by the respective pages:

- Authentication failed. User not found
- No account with that email address exists
- Username already exists

### 4.6.3 Recommendation

BSI recommends that the application be modified to return generic error messages to ensure that they do not indicate whether a valid user account exists.

## 4.7 Unauthorised Device History Deletion

| Finding No 7. | | |
|---|---|---|
| | Systems Affected | http://54.38.214.253 |
| | Finding | Authenticated users could delete the device history of other users |
| | CVE Number | N/A |
| | Root Cause | **Web Development** |
| | Impact | **1**  |
| | Likelihood | **2**  |
| | Overall Risk Rating | **2 (Very Low Risk)** |
| | Status | **ONGOING** |

### 4.7.1 Summary

Authenticated users could delete the device history of other users.

### 4.7.2 Technical Details

BSI determined that it was possible to delete individual records of login device history associated with other users.

An example of a HTTP POST request for device history deletion is shown below. The JSON **id** parameter value was used to perform the attack, as highlighted below:

```
POST /user/device/delete HTTP/1.1
Host: 54.38.214.253:9001
Content-Length: 33
Accept: application/json, text/plain, */*
Origin: http://54.38.214.253
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/69.0.3497.100 Safari/537.36
Content-Type: application/json;charset=UTF-8
Referer: http://54.38.214.253/
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Connection: close

{"id":"5ba49ad07c47ab6f08899cdf"}
```

An attacker would be required to know the device history **id** value belonging to another user. The **id** value is a 24-character token, and therefore the brute-forcing of such values would be unfeasible.

### 4.7.3 Recommendation

BSI recommends that the application be modified to ensure that appropriate authorisation checks are performed for all user actions.

## 4.8 Login IP Geo-Location Incorrect

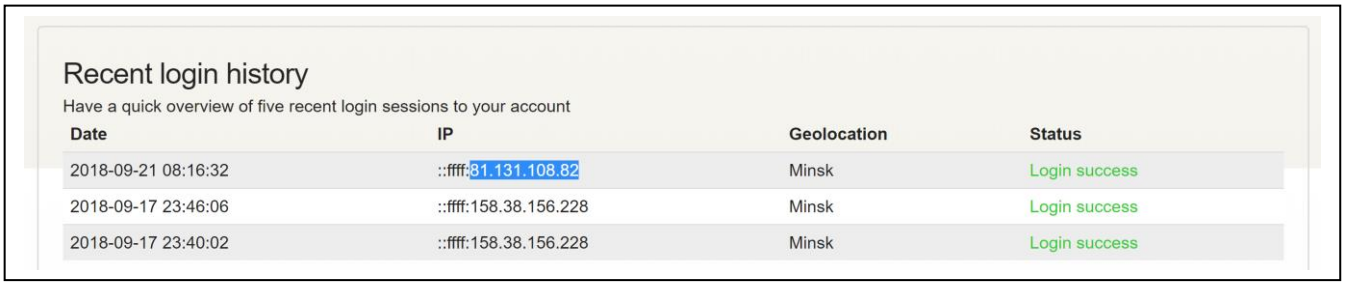
| Finding No 8. | | |
|---|---|---|
| | Systems Affected | `http://54.38.214.253` |
| | Finding | The login geo-location functionality did not correctly identify the location of the login IP addresses |
| | CVE Number | N/A |
| | Root Cause | **Web Development** |
| | Impact | **0** |
| | Likelihood | **0** |
| | Overall Risk Rating | **Informational Risk** |
| | Status | **ONGOING** |

### 4.8.1 Summary

The login geo-location functionality did not correctly identify the location of the login IP addresses.

### 4.8.2 Technical Details

All of the IP addresses displayed in the **Recent login history** section of the applications settings were identified as being in Minsk.




Whilst there is no direct security risk associated with the issue, the geo-location data would be used by the users to ensure that no authorised access had occurred.

### 4.8.3 Recommendation

BSI recommends that the geo-location functionality be modified to correctly identify the location associated with the login IP addresses.

# 5 Summary of Findings

## 5.1 Results of Web Application Testing

| Finding No. | Impact | Likelihood | | Overall Risk | Finding | Recommendation | Status |
|---|---|---|---|---|---|---|---|
| 1 | 5 | 3 | | **High** | **Application Available Over Unencrypted HTTP**<br>The web application was available over unencrypted HTTP protocol. | BSI recommends that the HTTP service be replaced with the cryptographically secure alternative HTTPS. | **ONGOING** |
| 2 | 4 | 3 | | **High** | **Stored Cross-Site Scripting**<br>A stored Cross-Site Scripting (XSS) vulnerability was identified which could be exploited by an authenticated user. | BSI recommends that all client supplied input is sufficiently filtered before being echoed back to the client's browser. Input validation should be carried out on all input fields to ensure that only input that matched an expected pattern is accepted. In addition, the application should implement output encoding to ensure that any potential unsafe data is properly encoded or escaped to prevent execution within the client's browser. | **ONGOING** |
| 3 | 4 | 3 | | **High** | **No Account Lockout**<br>The application did not enforce an account lockout policy | BSI recommends the implementation of anti-automation techniques to limit password-guessing and other attacks against the application login function. BSI highly recommends that Two Factor or Multi-Factor authentication be implemented and enforced for all logins and transaction authorisations. | **ONGOING** |
| 4 | 4 | 2 | | **Medium** | **Unvalidated Password Change**<br>User password could be updated without verifying the current user | BSI recommends that the application be modified to require users to input the current password of the account prior to permitting the password change. | **ONGOING** |

| Finding No. | Impact | Likelihood | | Overall Risk | Finding | Recommendation | Status |
|---|---|---|---|---|---|---|---|
| 5 | 4 | 2 | | **Medium** | **Unvalidated Email Change**<br>User email address could be updated without verifying the current user. | BSI recommends that the application be modified to require users to input the current password of the account prior to permitting the email change. | **ONGOING** |
| 6 | 2 | 3 | | **Medium** | **User Account Enumeration**<br>The application was found to present error messages that facilitated the enumeration of valid user accounts. | BSI recommends that the application be modified to return generic error messages to ensure that they do not indicate whether a valid user account exists. | **ONGOING** |
| 7 | 1 | 2 | | **Very Low** | **Unauthorised Device History Deletion**<br>Authenticated users could delete the device history of other users. | BSI recommends that the application be modified to ensure that appropriate authorisation checks are performed for all user actions. | **ONGOING** |
| 8 | N/A | N/A | | **Info** | **Login IP Geo-Location Incorrect**<br>The login geo-location functionality showed incorrect location data for some login IP addresses. | BSI recommends that the geo-location functionality be modified to correctly identify the location associated with the login IP addresses. | **ONGOING** |

# Appendix A - Testing Team

This project was undertaken using the following consultant:

- Mark Woan

Any queries regarding this testing and report should be directed to:

BSI Cybersecurity and Information Resilience Operations Team
Tel: +44 (0) 345 222 1711
Email: Operations.Cyber.UK@bsigroup.com

The primary point of contact at Dragon InfoSec was Richard Dennis (richard@dragoninfosec.com).

# Appendix B - Findings Definitions

BSI Cybersecurity and Information Resilience have developed a method for evaluating vulnerabilities and presenting the results in a way which enables clients to easily assess the risks they pose to the organisation.

## B.1. Risk Ratings

Each finding is categories by its "Seriousness" and "Likelihood". The overall risk rating is calculated as a multiple of the two values.

```
Overall risk = Seriousness x Likelihood
```

Below are guidance on rating definitions; exact ratings may depend on particular environment.

**Seriousness (Impact)**

**5** - Remotely gaining administrative access;

**4** - Remote privilege escalation or unauthorised read/write access;

**3** - Local privilege escalation or unauthorised read-only access to data;

**2** - Sensitive information disclosure. Minor security configuration weakness;

**1** - Minor non-sensitive information disclosure.

**Likelihood (exploitability)**

**5** - Trivial to exploit by unskilled person;

**4** - Require exploit code or tool which was in the public domain, or easy to exploit with some knowledge;

**3** - Require some exploit code development or effort to exploit, or require specific knowledge/skill;

**2** - Attacker may require specific access;

**1** - Theoretical vulnerability where there is no known exploit code and/or would require a lot of resources to exploit.

Rating may also take in to account existing defences which may restrict the exploitability.

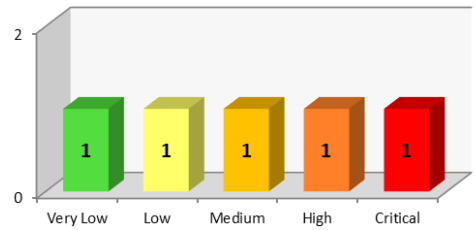
| Likelihood | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 5 | Low | Medium | High | Critical | Critical |
| 4 | Low | Medium | High | High | Critical |
| 3 | Low | Medium | Medium | High | High |
| 2 | V. Low | Low | Medium | Medium | Mediu m |
| 1 | V. Low | V. Low | Low | Low | Low |

**Seriousness** (Impact)

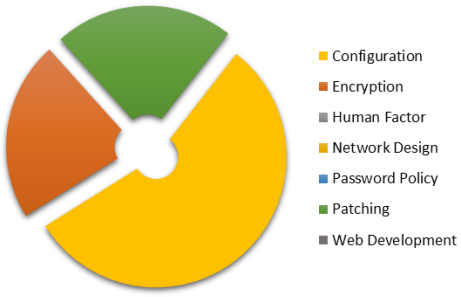
| Overall Risk Rating |
|---|
| Critical (20-25) |
| **High** (12-16) |
| **Medium** (6-10) |
| **Low** (3-5) |
| **Very Low** (1-2) |

## B.2. Executive Summary

The executive summary provides a number of graphical representations as to the most common root cause of the vulnerabilities identified. A summary of the number of different root cause categories are summarised in a graph in the management summary.




In addition, all findings are plotted onto a graph so that the severity of the vulnerabilities identified can easily be visualised. This enables the client to concentrate their efforts for resolution in specific areas




The pie chart depicts the most common root causes of the vulnerabilities identified.

### B.2.1. Root Causes

The root causes for infrastructure tests include:

- Configuration
- Encryption
- Human Factor
- Network Design
- Password Policy
- Patching
- Web Development

The root causes for application tests include:

- Authentication
- Client Side Controls
- Configuration
- Default Content
- Design Error
- Encryption
- Input Validation
- Logic Error
- Password Policy
- Session Control

## B.3. Findings Box

The table below provides a key to understand the findings description.

| | | |
|---|---|---|
| Finding No. X | Systems Affected | List of devices which are vulnerable. This will either take the form of IP addresses (DNS names) or URLs. |
| | Finding | An overview of the vulnerability identified. |
| | CVE number | Where possible, references will be made to a common reference identifier such as CVE or CWE. These references to external sources allow clients to find out additional details regarding the vulnerability and how to mitigate it. |
| | Root Cause | Each finding will be categorised as to the perceived root cause. Further details are discussed in the section below. |
| | Seriousness (Impact) | Impact if the vulnerability is successfully exploited. Rated from 5 (serious) to 1(not serious). |
| | Likelihood | How easy is the vulnerability to exploit? Ratings from 5 (easy) to 1 (difficult). |
| | Overall Risk rating | The overall risk rating takes into account the seriousness of the issue, the likelihood of the vulnerability being exploited, as well as other factors that could impact the overall risk. |


(visual representation)

Note: It should be noted that the definitions defined above for the seriousness and likelihood ratings are only guidelines.