



eLodgment Security Assessment

Federal Court of Australia



2008 Australian Capital Territory Winner
Panasonic Australia Medium Business Award

30 August 2009

Table of Contents

- 1 Executive Summary..... 5
 - 1.1 Overview 5
 - 1.2 Key Findings 5
 - 1.3 Risk Exposure 6

- 2 Schedule of Recommendations 7
 - 2.1.1 Identified Issues 7

- 3 Review Approach 9
 - 3.1 Internal Vulnerability Assessment Study Method 9
 - 3.2 Application Security Study Method 9

- 4 Application Assessment 12
 - 4.1 eLodgment Application 12
 - 4.1.1 Summary 12
 - 4.1.2 Test Case Findings Summary 12
 - 4.1.3 TCA-01 Information Gathering 13
 - 4.1.4 TCA-02 Information Disclosure 13
 - 4.1.5 TCA-03 Authentication and Authorisation 14
 - 4.1.6 TCA-04 Session Management 16
 - 4.1.7 TCA-05 Client Security 17
 - 4.1.8 TCA-06 Injection 18
 - 4.1.9 TCA-07 Use of Cryptography 21
 - 4.1.10 TCA-08 Business Logic 22
 - 4.1.11 TCA-09 Denial of Service 23
 - 4.1.12 TCA-10 Logging Manipulation 23

- 5 External Vulnerability Assessment..... 25
 - 5.1 Assessment Summary 25
 - 5.2 Assessment Findings 25
 - 5.2.1 VA01 – Insecure / Inappropriate Services Exposed 25
 - 5.2.2 VA02 – Missing Patches 26
 - 5.2.3 VA03 – Outdated Software 27
 - 5.2.4 VA04 – Information Disclosure 27

5.2.5	VA05 – Misconfigured Security Settings	27
Annex A	Review Scope	29
	In Scope.....	29
	Project Administration, Preparation & Reporting	29
	Application Penetration Test	29
	Out of Scope.....	30
Annex B	Risk Rating Tables	32
	Likelihood.....	32
	Consequence.....	33
	Overall Risk.....	33

Index of Tables

Table 1: Issues identified, by risk rating.....	6
Table 2: Schedule of Issues and Recommendations.....	7
Table 3: eLodgment Application Test Results Risk Summary	12
Table 4: Test Case Findings	12
Table 5: External Vulnerability Assessment Risk Summary	25
Table 6: External Vulnerability Assessment Findings Summary	25
Table 7: Project High Level Task Analysis - Administration	29
Table 8: Scope task analysis – Application Penetration Test.....	29
Table 9: Likelihood Rating Table	32
Table 10: Consequence Rating Table	33
Table 11: Overall Risk Rating Table.....	34

Table of Figures

Figure 1: Accessing Application Error Log as CSOfficer.....	15
Figure 2: Access external user list as CSO Administrator.....	15
Figure 3: Before modification	18
Figure 4: After modification	19
Figure 5: User prompted to open batch file	20
Figure 6: A batch file running on the user's computer	20
Figure 7: Cross-site scripting via file upload	21

Amendment record

Version	Date	Description
0.1	12/8/2009	Internal Release
0.2	12/8/2009	Draft release to the Federal Court of Australia
1.0	24/8/2009	Final release

Copyright Notice:

This document contains information protected by copyright.

© STRATSEC.NET PTY LTD. ABN 14 111 187 270.

The material in this document may not be commercialised without prior written permission from STRATSEC.NET PTY LTD.

1 Executive Summary

1.1 Overview

stratsec was engaged by the Federal Court of Australia (the Court) to conduct a security assessment of the eLodgment application and hosting infrastructure. The assessment comprised of an application security assessment conducted from an unauthenticated and authenticated perspective, to assess the security profile of the application as it would appear to an unauthenticated external attacker, or a malicious internal user.

1.2 Key Findings

In general, eLodgment was found to be configured with appropriate security controls. While the application and infrastructure securely handled most common attack types, two Moderate and four Low risks were identified within the environment that introduces risks such as privilege escalation or bypass of business rules. However there were mitigating factors for all these issues such as the difficulty of exploitation or the manual validation process.

The four key findings of the assessment that should be addressed before the final deployment of the application are:

Application Assessment

1. The application was found to be affected by an access control bypass, which can be exploited by internal staff users to access unauthorised functionality within the application, such as the eLodgment logging functionality (Finding reference [APP-01]).
2. It is possible for malicious users to upload files with arbitrary file extensions, which could then be inadvertently opened by legitimate users. Anti-virus and content checking partially mitigate this risk, however this technique could potentially be exploited by an attacker to execute arbitrary code on legitimate users' systems. The risk of malicious content being uploaded is always going to be present in a system which allows uploading of content, however performing some simple checks on the server would greatly reduce the chance that users would execute this malicious content (Finding reference [APP-04]).

Network Vulnerability Assessment

3. A number of potentially unnecessary open ports were identified on the eLodgment web server during scanning. However, attempts to communicate with services on these ports were blocked, most likely by the firewall, and could not be exploited by **stratsec** during the course of the assessment (Finding reference [INF-01]).
4. The web server was found to be configured to accept connections using cryptographically weak SSL ciphers. A skilled attacker, with access to intercept traffic between a user and the eLodgment application can potentially exploit this to intercept encrypted communications in plaintext. However, this attack is unlikely due to the nature of Internet traffic routing (Finding reference [INF-02]).

The following table provides an indication of the risks and vulnerabilities in the current state of the application:

Table 1: Issues identified, by risk rating

Review Component	Extreme	High	Moderate	Low
eLodgment Application	0	0	2	4
eLodgment Webserver	0	0	0	2
Total	0	0	2	6

1.3 Risk Exposure

Based on the risk matrix used for this assessment the eLodgment application is considered to be at a **MODERATE** risk of exposure due to the lack of readily and easily exploitable security issues.

2 Schedule of Recommendations

The following schedule incorporates all risks and recommendations identified in this deliverable as a result of test cases performed. Ratings are in line with AS/NZS 4360. Further detail on the risk ratings can be found in Appendix B: Risk Rating Tables. The numbering scheme references the full recommendation, as provided within the report.

Domain references are as follows: eLodgment [APP]; Infrastructure [INF].

Likelihood ratings are as follows: Rare [RARE]; Unlikely [UNL]; Moderate [MOD]; Likely [LIK]; Almost Certain [AMC].

Consequence ratings are as follows: Insignificant [INS]; Minor [MIN]; Moderate [MOD]; Major [MAJ]; Catastrophic [CAT].

Risk Ratings are as follows: Extreme [EXT], High [HIGH], Moderate [MOD]; Low [LOW].

2.1.1 Identified Issues

Table 2: Schedule of Issues and Recommendations

Ref	Issue / Risk	Like.	Cons.	Risk	Recommendation	Identified Follow-up Actions
APP-01	It is possible to modify JavaScript requests used within the application in order to access administrative functionality.	RARE	MOD	MOD	Modify the web application to enforce access permissions on the server side for all sensitive functionality.	CDT will review the required effort to remedy this issue.
APP-04	It is possible to upload files with an arbitrary file extension, which can be exploited by attackers to upload malware onto the eLodgment servers.	RARE	MOD	MOD	Store files on disk with the same extension used when they are checked for integrity.	CDT will update the application to remedy this issue.
APP-02	There is no mechanism to prevent simultaneous logins to the one user account.	RARE	MIN	LOW	Ultimately the requirement for concurrent session prevention is a business decision, but if this is a requirement, then it is recommended that users be required to re-	The Court will review the business requirements for

					authenticate in the case where their credentials have been identified as being used in two places at the same time	multiple user logins.
APP-03	The cookie generated by the eLodgment portal application when the user logs in is not marked as HttpOnly, which may lead to session hijacking if certain other vulnerabilities are present.	RARE	INS	LOW	Although this attack is extremely unlikely it, altering the web application to mark cookies as "HttpOnly" would align it with security best practice.	CDT will update the application to remedy this issue.
APP-05	Cookies were not marked "Secure", allowing active attackers to intercept a session cookie, and hijack a user's browsing session.	RARE	MIN	LOW	Configure the application to mark cookies as "Secure" and only transmit them over encrypted connections (e.g. HTTP/S).	CDT will update the application to remedy this issue.
APP-06	It is possible to bypass business logic rules by submitting page requests out of order.	RARE	MIN	LOW	Modify the application to check all payment details and applicable business logic checks during the payment/lodgment finalisation stage.	CDT and the Court to consider alternative approaches to prevent or detect this issue.
INF-01	A number of ports were exposed on the eLodgment web server, but did not correspond to any accessible services.	RARE	INS	LOW	Configure the relevant network access control device/s to restrict the number of ports responding to external scanning.	The Court will review this issue.
INF-02	The web server was found to be configured to accept connections using cryptographically weak encryption protocols.	RARE	MIN	LOW	Configure the web application to only accept SSL connections using SSL v3 or TLS v1, and strong cryptographic cipher suites.	The Court will update the server to remedy this issue.

3 Review Approach

3.1 Internal Vulnerability Assessment Study Method

stratsec completed an external vulnerability assessment of the following IP address associated with the Federal Court eLodgment environment:

Table 3: Target Host for Vulnerability Assessment

IP Address	Hostnames
210.193.178.202	www.eLodgment.fedcourt.gov.au

Scanning was conducted both over the internet and locally from within the Court's offices, to assess the system's security posture from the perspective of a malicious internal user or potential external attacker. The vulnerability assessment consisted of the following tasks:

- Port scanning;
- Service identification;
- Host vulnerability scanning; and
- Manual vulnerability verification.

3.2 Application Security Study Method

stratsec conducted an application penetration test of the following systems using a structured verification approach. The URLs provided for the application security assessment are shown below:

- Federal Court eLodgment Application
<https://eLodgment.fedcourt.gov.au>
<https://fcadmzuelpa1/eLodgment/Default.aspx> (internal domain name for external site)
<http://fcadmztelpa2/eLodgmentAdmin/Default.aspx> (internal staff website)

Testing was performed using the following test accounts as provided by the Court:

- Username: stratsec1, stratsec2
Role: User
- Username: ELODGMNT1, ELODGMNT6
Role: Chamber Staff Officer
- Username: ELODGMNT2, ELODGMNT5
Role: Chamber Staff
- Username: ELODGMNT3, ELODGMNT7, ELODGMNT8
Role: Chamber Staff Officer - Administrator
- Username: ELODGMNT4
Role: Administrator

Application penetration testing comprised of application familiarisation followed by an in-depth assessment using the following test cases as a starting point for response and behaviour analysis:

- TCA-01 - Information Gathering
Information gathering is the most fundamental step in application security testing. It allows

the tester to become familiar with the application, identify all the components, and subsequently prioritise testing effort based on the highest risk areas of the system.

- **TCA-02 - Information Disclosure**
A common vulnerability in web applications is the accidental disclosure of sensitive information either directly or implicitly through application behaviour. This includes both confidential information, such as user data or company secrets, and internal application details which may aid an attacker in identifying vulnerabilities including application debug output, source code, application API versions, directory structure and network layout.
- **TCA-03 - Authentication and Authorisation**
If authentication is not conducted robustly, an attacker may be able to access application functionality without identifying themselves to the system or may be able to supply a fraudulent identity when performing application actions. It may also be possible for an attacker to masquerade as a legitimate user – accessing private information or executing actions on behalf of the victim. The failure of authorisation and access controls may allow an attacker to view data or perform actions which they are not entitled to access.
- **TCA-04 - Session Management**
It is common for applications to track an individual's navigation through the use of stored session information, especially when authentication is involved. Session management is closely linked to authentication, as sessions are typically used to prevent the need for a user to provide authentication credentials for every request. This means an attacker who successfully hijacks a valid user session or otherwise subverts session functionality, can access the web application as if they were the session's rightful owner.
- **TCA-05 - Client Security**
Client side attacks target the security of the client application or system. In a typical scenario, cross-site scripting attacks are used to compromise the integrity or privacy of the web browser to steal session tokens and impersonate legitimate users.
- **TCA-06 – Injection**
Appropriate data validation within an application allows it to detect and handle malformed or unexpected inputs before passing such data to subsystems for processing or execution. Insufficient or inappropriate data validation within an application may allow an attacker to supply malicious commands or parameters to subsystems which may affect the results of processing or cause unauthorised actions to be performed.
- **TCA-07 - Misuse of Cryptography**
Cryptography often provides a means of securing an application and its data however it is notoriously complex to design, implement, and configure securely. Issues with cryptography often result in the compromise of data held within the system as protections are usually applied to important components.
- **TCA-08 - Business Logic**
An individual application contains workflows and implements business rules and policies specific to that application. Business logic can be susceptible to flaws which allow for actions outside these workflows and business rules to be performed. Such issues impact applications in ways specific to their individual context.
- **TCA-09 - Denial of Service**
Denial of service attacks seek to disrupt the business function being provided an application. There are many forms of denial of service attacks however all target ability of an application to achieve its intended goal are therefore analysed in terms of the applications context.

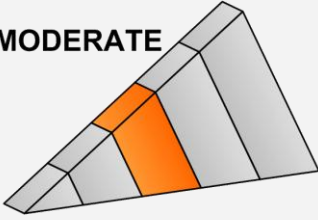
- TCA-10 - Logging Manipulation

Logs are a fundamental component of the intrusion detection process and often form much of the audit trail. In many applications all non-repudiation is provided by logs. Testing of log mechanisms seeks to verify that the data stored can be tampered with, disguised, or otherwise manipulated. Furthermore, it seeks to ensure that logs store a complete and thorough record of events.

4 Application Assessment

4.1 eLodgment Application

Table 3: eLodgment Application Test Results Risk Summary

Test Component	Location	Business Risk Exposure
eLodgment Application (internal and external)	https://eLodgment.fedcourt.gov.au https://fcadmzuelpa1/eLodgment/Default.aspx (internal domain name for external site) http://fcadmztelpa2/eLodgmentAdmin/Default.aspx (internal staff website)	MODERATE 

4.1.1 Summary

The majority of the security controls in the eLodgment application were found to be configured at standard industry practice levels, using a secure application framework which was configured appropriately, a number of specific point issues were identified. The more serious issues identified included:

- An internal or external user may upload an executable file type which may be executed on another user's computer;
- An internal user to access functionality for which they are not authorised by guessing URL parameters; and
- Any user may bypass business logic checks such as payment requirements by submitting out-of-order requests to the application.

4.1.2 Test Case Findings Summary

The following table is a summary of the security posture of the application component with reference to the **stratsec** Test Cases:

Table 4: Test Case Findings

Test Case	Issue Identified	No Issue Identified
TCA-01 Information Gathering		◆
TCA-02 Information Disclosure		◆
TCA-03 Authentication and Authorisation	◆	
TCA-04 Session Management	◆	
TCA-05 Client Security		◆
TCA-06 Injection		◆

TCA-07 Use of Cryptography	◆	
TCA-08 Business Logic	◆	
TCA-09 Denial of Service		◆
TCA-10 Logging Manipulation		◆

4.1.3 TCA-01 Information Gathering

Information gathering is the most fundamental step in application security testing. It allows the tester to become familiar with the application, identify all the components, and subsequently prioritise testing effort based on the highest risk areas of the system.

Server Technology Identification

The first step towards assessing any web application is determining which technologies it has been created with and is currently hosted on. The eLodgment application was found to comprise of the following technologies:

- Microsoft IIS 6.0;
- ASP.NET 2.0.50727;
- AjaxControlToolkit Version 3.0.20820.24771; and
- FCKEditor.

This is an informational item only and no risk is associated with this finding. Note that these versions number do not provide detail of specific patching levels. This is as per best practice, but also means that **stratsec** is unable to determine if the latest version of these software components are installed.

4.1.4 TCA-02 Information Disclosure

A common vulnerability in web applications is the accidental disclosure of sensitive information either directly or implicitly through application behaviour. This includes both confidential information, such as user data or company secrets, and internal application details which may aid an attacker in identifying vulnerabilities including application debug output, source code, application API versions, directory structure and network layout.

Verbose Error Messages

Information disclosure through error messages is one of the most prevalent issues in modern web applications. For the most part, the eLodgment application provided very good error handling and did not display any sensitive information to the user. The only potential issue was that the error message provided when the upload of a file fails provides possible more information than necessary.

It is noted however that the same message is displayed to the user as is displayed to the administrator in the eLodgment application event log. This is not an issue in itself and nothing particularly sensitive is revealed in the error message presently, however if error messages were made more verbose – perhaps to aid problem resolution – these messages are likely to be displayed to the user as well as the administrator.

4.1.5 TCA-03 Authentication and Authorisation

If authentication is not conducted in a robust manner, an attacker may be able to access application functionality without identifying themselves to the system or supply a fraudulent identity when performing application actions. It may also be possible for an attacker to masquerade a legitimate user – accessing private information or executing actions on behalf of the victim. The failure of authorisation and access controls may allow an attacker to view data or perform actions which they are not entitled to access.

Internal User Access Control Bypass

Issue Details

Internal accounts of varying privilege levels were supplied to **stratsec** in order to test controls. While the interface of the application was found to be appropriate, with the correct functions being displayed to the correct user, access control was found to not be checked consistently on the server for all functions.

Some ASP.NET “controls” appear to be invisible to the lower privileged users (i.e. there is not menu item for them), but are still functioning if the user can guess the ASP.NET control id. Malicious internal users could exploit this issue by modifying ASP.NET’s JavaScript Postback request when accessing a legitimate control to point to an otherwise disabled control, and thus gain access to privileged sections of the eLodgment application.

For example, it was found to be possible to show a list of external users (an admin function) from a CSOfficer account by performing the following Postback request:

- javascript:__doPostBack('nm\$dllv2\$ctl01\$dllv2\$ctl00\$lbItem',')

The screenshot shows the eLodgment Admin interface in a Mozilla Firefox browser. The address bar contains the URL: `javascript:__doPostBack('nm$dllv2$ctl01$dllv2$ctl00$lbItem',')`. The page displays a table of document synchronization events.

Date	Source	Status	Message	Action
13/08/2009 11:15:59	Cache Sync	Completed	Document Stamp Synchronization	View
13/08/2009 11:12:31	Cache Sync	Completed	Mandatory Questions Synchronizat	View
13/08/2009 11:12:30	Cache Sync	Completed	Matter Role Types Synchronization	View
13/08/2009 11:12:30	Cache Sync	Completed	Party Types Synchronization	View
13/08/2009 11:12:10	Cache Sync	Completed	Primary Acts Synchronization	View
13/08/2009 11:12:10	Cache Sync	Completed	Action Officers Synchronization	View
13/08/2009 11:11:05	Cache Sync	Completed	Action Documents Synchronization	View
13/08/2009 11:11:04	Cache Sync	Completed	Reasons For Waiver Synchronizatic	View
13/08/2009 11:11:03	Cache Sync	Completed	Sources Synchronization	View
13/08/2009 11:10:49	Cache Sync	Completed	Case Types Synchronization	View

Note the two numbers in bold in the above URL – these are the only parts that need to be guessed

by an attacker, as the structure of the URL is provided in the source code to most pages. These are typically an integer value less than five (05) so guessing these control IDs is very feasible for any internal user with a moderate understanding of ASP.NET

Further examples of privilege escalation are shown below. The following screenshot shows CSOfficer accessing the system event log via this method:

The screenshot shows the eLodgment Admin interface. The user is logged in as 'elodgment1'. The 'Reports' tab is selected. Below the navigation bar, there are filters for 'Document Action', 'Document User Type', and 'Document User Id'. A 'Filter' dropdown and a 'Clear Log' button are visible. The main content area displays a table of application error logs with columns for Date, Source, Status, and Message. The table shows five entries, all with a 'Fail' status and the message 'Asynchronous Postbacks Error'. Each entry has a 'View' button next to it.

Date	Source	Status	Message	
7/08/2009 3:52:19 PM	FEDCOURT\elodgment1	Fail	Asynchronous Postbacks Error	View
7/08/2009 3:52:11 PM	FEDCOURT\elodgment1	Fail	Asynchronous Postbacks Error	View
7/08/2009 3:52:04 PM	FEDCOURT\elodgment1	Fail	Asynchronous Postbacks Error	View
7/08/2009 3:50:54 PM	FEDCOURT\elodgment1	Fail	Asynchronous Postbacks Error	View
7/08/2009 3:49:22 PM	FEDCOURT\elodgment1	Fail	Asynchronous Postbacks Error	View

Figure 1: Accessing Application Error Log as CSOfficer

The screenshot shows the eLodgment Admin interface. The user is logged in as 'elodgment3'. The 'Users' tab is selected. Below the navigation bar, there are filters for 'New (6)', 'Active (21)', 'Rejected (1)', 'Pending', and 'Inactive'. A 'Search' dropdown is visible. The main content area displays a table of external user lists with columns for Name, Type, Username, Email, Client ID, and Status. The table shows ten entries, all with an 'Active' status. Each entry has a 'View' button next to it.

Name	Type	Username	Email	Client ID	Status	
Alaric	Government Agency	spottydog	x@x.com	BAR4511532	Active	View
Aspire A pathway to mental h	Non-Government Organisation	Aspire	aspire@aspire.org.au	ASP4511672	Active	View
Beling, Robinson	Self Represented Litigant	RobBel	frankmay2008@hotmail.com	ROB4481042	Active	View
Brown, John	Self Represented Litigant	johnbrown	johnbrown@live.com.au	bro4465808	Active	View
Brown, Susan	Self Represented Litigant	sbrown	susanbrownsl@gmail.com	BRO4511654	Active	View
Business Limited	Business	businessltd	businesslimited1@gmail.com	BUS4511673	Active	View
elodgment one	Business	elodgment1	ddd@gg.com	ELO4514707	Active	View
Energy and Conservation Der	Government Agency	Conservation	conservation@energy.gov.au	ENE4511692	Active	View

Figure 2: Access external user list as CSO Administrator

This attack was not successful for all controls – access control is enforced on the server side for some controls. Due to the manual trial and error nature of this attack, it is not possible to completely enumerate the possible privilege escalations and it is likely that further examples of this issue would be found.

Attack Scenario

A malicious internal user can modify a Postback request to gain access to unauthorised and hidden functionality which should be limited to a more privileged user.

Risk

As this issue requires authenticated access to the application and a significant level of technical knowledge to identify, the likelihood that this issue will be exploited is considered RARE. As successful exploitation of this issue may allow malicious internal users to access administrator functionality, the consequence of this issue being successfully exploited is considered MODERATE. Therefore, this issue is considered to present a **MODERATE** risk.

Recommendation

Modify the web application to enforce access permissions on the server side for all sensitive functionality [Recommendation APP-01].

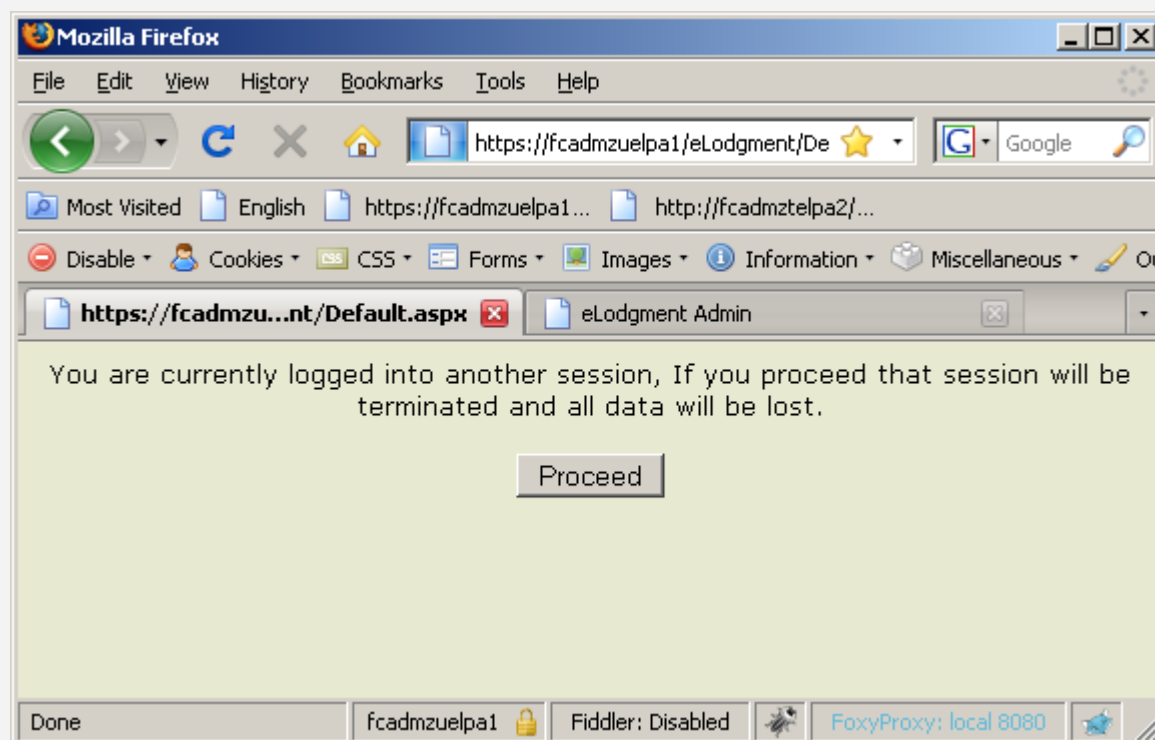
4.1.6 TCA-04 Session Management

It is common for applications to track an individual's navigation through the use of stored session information, especially when authentication is involved. Session management is closely linked to authentication, as sessions are typically used to prevent the need for a user to provide authentication credentials for every request. This means an attacker who successfully hijacks a valid user session or otherwise subverts session functionality, can access the web application as if they were the session's rightful owner.

Concurrent Session Prevention Ineffective

Issue Details

Currently when a user attempts to log in to their account from two different browsers, a warning message is presented to both users saying that their session is already being used as follows:



When the button is pressed, a new session identifier is allocated to the user, and they can start using the website again. It is not known whether prevention of concurrent sessions is a business

requirement, but if it is, then the currently implemented protection is not effective. This is because currently users are not required to re-authenticate when a concurrent session is detected.

Attack Scenario

An attacker who manages to hijack a session via cross-site scripting, or physical access to the user's machine could continue using that session even after a concurrent session was detected.

Risk

The likelihood of this issue is considered RARE as an attacker would have had to previously gain access to a user's session. The consequence of this issue is MINOR as it merely extends the attack window for an attacker. As such this is considered a **LOW** risk.

Recommendation

Ultimately the requirement for concurrent session prevention is a business decision, but if this is a requirement, then it is recommended that users be required to re-authenticate in the case where their credentials have been identified as being used in two places at the same time.

[[Recommendation APP-02](#)]

Cookie Not Marked as 'HttpOnly

Issue Details

The cookie generated by the eLodgment portal application when the user logs in is not marked as HttpOnly, and can be intercepted by an attacker who is able to successfully exploit a cross-site scripting condition in the eLodgment application.

It is noted that no cross-site scripting issues were identified during testing, except however enabling this setting is considered good security practice.

Attack Scenario

An attacker who found a cross-site scripting issue in the eLodgment website could inject script into a users browser to hijack their session.

Risk

As this vulnerability requires an attacker to successfully exploit a cross-site scripting issue within the eLodgment application, the likelihood is considered RARE (given that **stratsec** did not identify any of these vulnerabilities during testing). As the protection afforded by this setting is limited, not having it enabled is of INSIGNIFICANT consequence. Therefore, this issue is considered to present a **LOW** risk.

Recommendation

Although this attack is extremely unlikely it, altering the web application to mark cookies as "HttpOnly" would align it with security best practice [[Recommendation APP-03](#)].

4.1.7 TCA-05 Client Security

Client side attacks target the security of the client application or system. In a typical scenario, cross-site scripting attacks are used to compromise the integrity or privacy of the web browser to steal session tokens and impersonate legitimate users.

No client security issues were identified during the assessment. Dangerous input was HTML-encoded by the application, and was safely displayed to the user. Furthermore it was noted that ASP.NET dangerous input exceptions were enabled which provides another layer of protection against cross-site scripting.

4.1.8 TCA-06 Injection

Appropriate data validation within an application allows it to detect and handle malformed or unexpected inputs before passing such data to subsystems for processing or execution. Insufficient or inappropriate data validation within an application may allow an attacker to supply malicious commands or parameters to subsystems which may affect the results of processing or cause unauthorised actions to be performed.

Arbitrary File Extension Upload

Issue Details

An attacker is able to upload a file with an arbitrary file extension, as long as the file remains a valid file formats accepted by the eLodgment application. While **stratsec** was not able to cause code execution on the server, this issue does increase the risk that a malicious user could convince another user of the system to run malicious code.

The attack is performed by modifying the filename and mime type on the client side, by trapping the request in between the browser and the server, as shown below.


```

original request | edited request | response
raw | params | headers | hex
POST /eLodgment/AsyncFileUploadService.axd?id=60466899e23740c2876a0699f387e0ca HTTP/1.1
Host: fcadmzuelpal
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.13) Gecko/2009073022
Firefox/3.0.13
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: https://fcadmzuelpal/eLodgment/Default.aspx
Cookie: ASP.NET_SessionId=z15wzqr0gxdsnrqq2n2kg455;
.ASPXAUTH=23B56291615388242E8AF6E535DC7C7294AB70BB8CBD8FE808C96D8C025E2D6DFE0AF4F4C162366
3A446BD27733CDDAEOCDC2FA7594982ADBC2DD1A0FDFAC4A736FD18F8DA995B30EFE593
Content-Type: multipart/form-data; boundary=-----256672629917035
Content-Length: 542

-----256672629917035
Content-Disposition: form-data; name="cl$ad$s$afuDoc$fileUpload"; filename="test.rtf"
Content-Type: application/msexcel

```

Figure 3: Before modification



```

original request | edited request | response
raw | params | headers | hex
POST /eLodgment/AsyncFileUploadService.axd?id=60466899e23740c2876a0699f387e0ca HTTP/1.1
Host: fcadmzuelpal
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.13) Gecko/20090730
Firefox/3.0.13
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: https://fcadmzuelpal/eLodgment/Default.aspx
Cookie: ASP.NET_SessionId=z15wzqrOgxdsnrqg2n2kg455;
.ASPXAUTH=23B56291615388242E8AF6E535DC7C7294AB70BB8CBD8FE808C96D8C025E2D6DFE0AF4F4C1623
3A446BD27733CDOAEOCDC2FA7594982ADBC2DD1AOFDFAC4A736FD18F8DA995B30EFEF593
Content-Type: multipart/form-data; boundary=-----256672629917035
Content-Length: 542

-----256672629917035
Content-Disposition: form-data; name="cl$ad$s$afuDoc$fileUpload"; filename="test.bat"
Content-Type: application/msword

```

Figure 4: After modification

Note that the mime type of the file must also be changed to match the extension. This can also be achieved by uploading a file which is valid in an accepted type, and also the type of the desired extension. In the screenshots below, a file which was both a valid Rich Text and Windows Batch file. When any user later attempts to download an executable file, the download.aspx page will serve a Windows Batch file (.bat) rather than a Rich Text Format file. In this case the user will be prompted to download the file similar to the screenshot below:

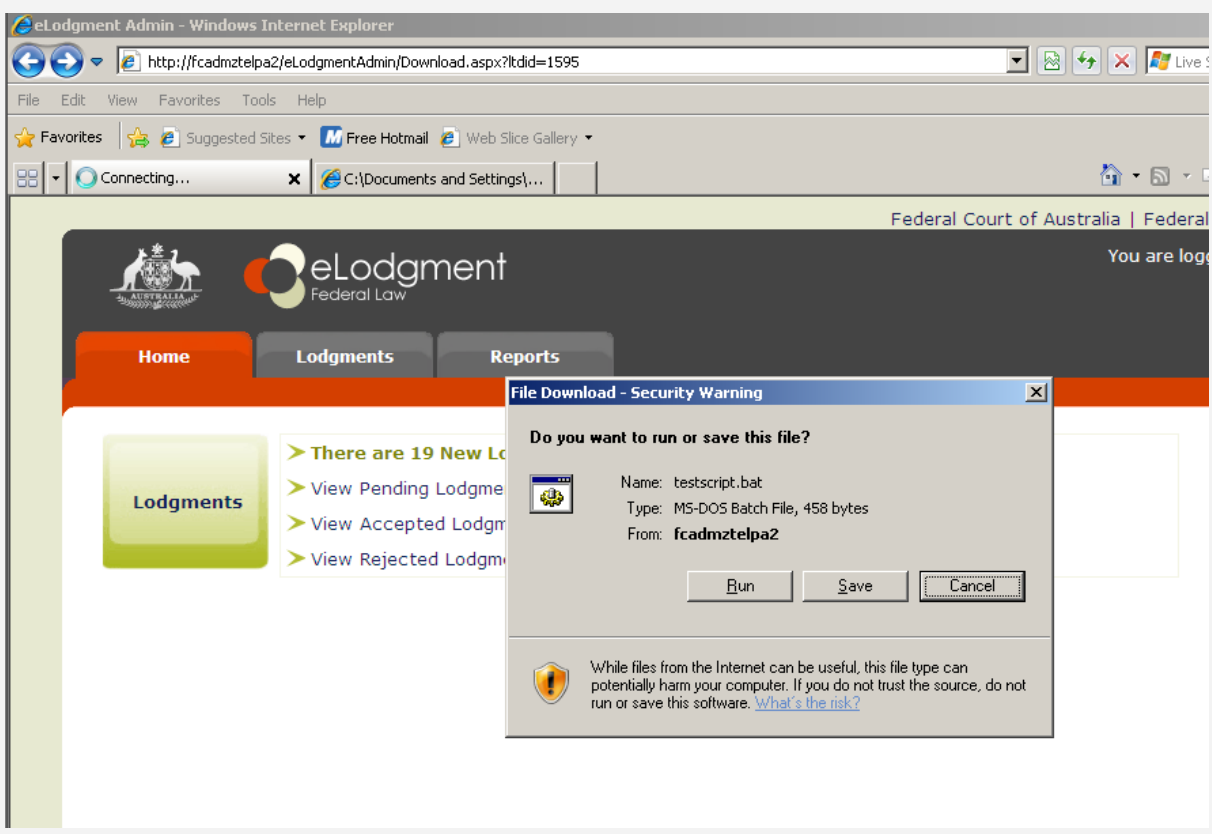


Figure 5: User prompted to open batch file

If the user clicks the Run button, or saves the file and then opens it, calculator will be opened as shown below:

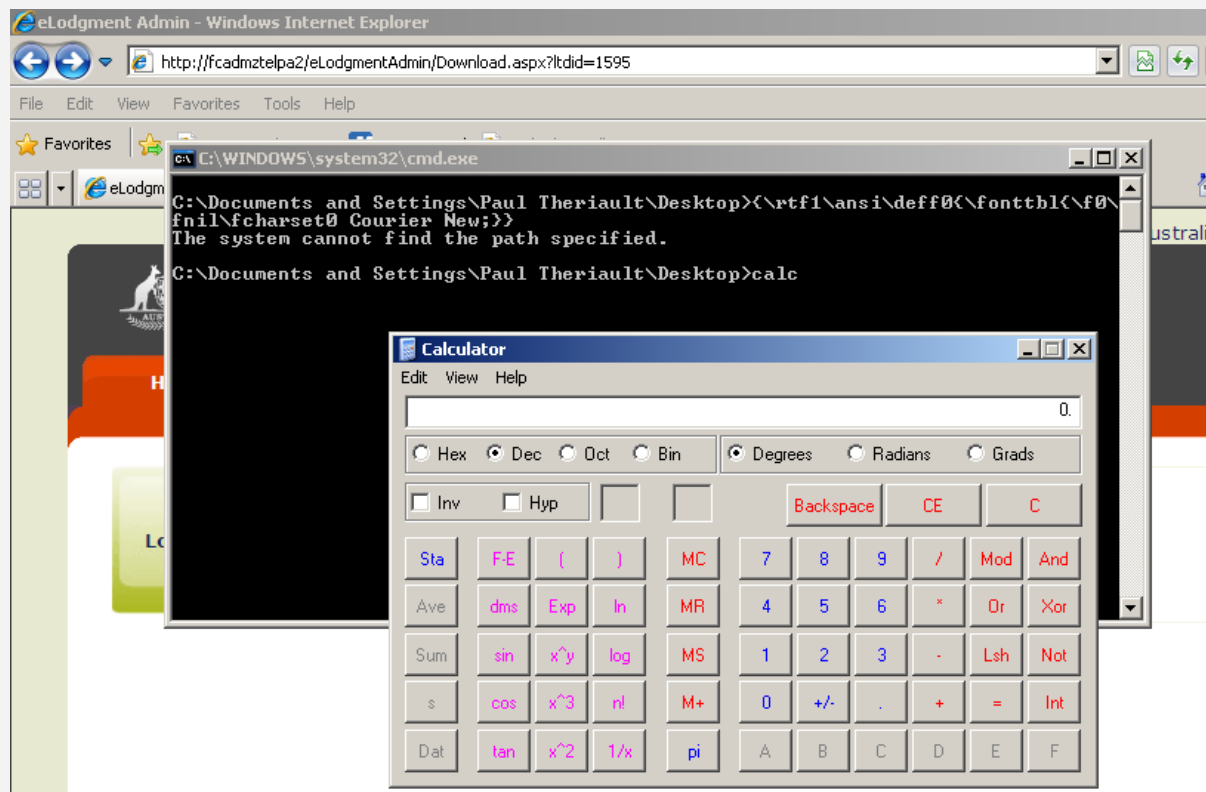


Figure 6: A batch file running on the user's computer

The example of opening calculator is benign, however an attacker could easily substitute this with a malicious command such as adding a user, or connecting to a remote server.

Another potential attack is to upload a html file, which results in a cross-site scripting attack, allowing the attacker to potentially hijack a users session. The screenshot below shows script executing in the context of the elodgment application – it merely opens an alerts box, but could easily perform a more sinister action, such as sending the users session token to an attacker.

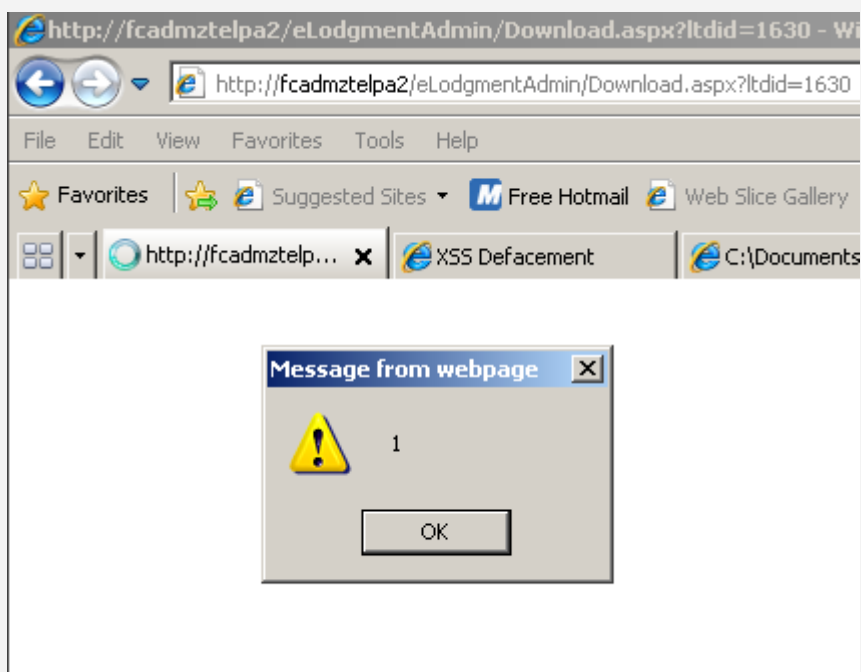


Figure 7: Cross-site scripting via file upload

Note that all of these examples require the user to click “open” when prompted by the browser (as opposed to saving the file to disk).

Attack Scenario

An attacker can create a specially crafted file which bypasses checks on the server. Any user who downloads this file and attempts to open it by double-clicking the saved file (Windows) will then launch the file and execute the malicious commands inserted by the attacker.

Risk

As this issue requires authenticated access to the application and a significant level of technical knowledge to identify, the likelihood that this issue will be exploited is considered RARE. As successful exploitation of this issue may allow attackers to execute arbitrary code on a legitimate user’s system, the consequence of this issue being successfully exploited is considered MODERATE. Therefore, this issue is considered to present a **MODERATE** risk.

Recommendation

Modify the web application to check the extension of the file against a whitelist, as currently occurs on the client side. Furthermore a check in Download.asp should be enabled so that it only serves files from a whitelist of allowed file types. [\[Recommendation APP-04\]](#).

4.1.9 TCA-07 Use of Cryptography

Cryptography often provides a means of securing an application and its data, however it is notoriously complex to design, implement, and configure securely. Issues with cryptography often result in the compromise of data held within the system as protections are usually applied to important components.

Cookie Not Marked 'Secure'

Issue Details

The cookie generated by the eLodgment portal application when the user logs in is not marked as secure, and as such can be intercepted by an active attacker who has an ability to perform a man-in-the-middle attack. This negates a lot of the benefit of using SSL for encryption as interception of session cookies typically allows unauthorised access while the session is active.

Attack Scenario

An attacker who could intercept traffic between the user and the web application could bypass the encryption to gain session tokens.

Risk

As this vulnerability requires an attacker to be able to intercept the user's communication or successfully exploit a cross-site scripting issue within the eLodgment application, the likelihood is considered RARE. As the cookies used by the eLodgment application only contain ASP.NET Session ID's and do not contain any sensitive data, the consequence of this issue being exploited is considered MINOR. Therefore, this issue is considered to present a **LOW** risk.

Recommendation

Alter the web application to mark cookies as 'secure' [Recommendation ELO-05].

4.1.10 TCA-08 Business Logic

An individual application contains workflows and implements business rules and policies specific to that application. Business logic can be susceptible to flaws which allow for actions outside these workflows and business rules to be performed. Such issues impact applications in ways specific to their individual context.

Bypass of Business Rules

Issue Details

As the lodgement process is a multiple stage process, it is possible to exploit the stateless nature of HTTP to reach unintended application states. The main issue is that business rule validation is performed throughout the process, rather than at the end, so an attacker can reach an unauthorised state by sending requests out of order.

For example, an attacker is able to lodge arbitrary documents (including originating documents) without immediate payment, by performing the following procedure:

1. Initiate a document lodgement procedure.
2. Add all relevant documents, and proceed to Payment page.
3. Select "Credit Card" for payment option, and enter valid Credit Card details
4. Proceed to the "Finalize Payment" page.
5. Use an intercepting proxy to save (but not send) the final request which finalises payment.

6. Using the web interface, proceed back to the Payment options page.
7. Select “Account” as the payment method (an error is displayed saying that you can not continue as there is no account available for you). At this point if you were just using the web interface you would not be able to continue, since there is not button available to go forward, but we have the request we saved earlier.
8. Use the intercepting proxy to now send the previously saved request which finalises the lodgement, even though the payment method is currently set to account.

This allows the user to submit a document with “Account” specified as the payment method, even if the user does not have an account with the Federal Court of Australia. Note that this is just one potential issue – it is likely that many of the business rules performed throughout the lodgement process might be bypassed in this manner. Although testing every permutation is infeasible other examples which **stratsec** was able to achieve included:

- Creating a lodgement with no file attachment
- Creating a lodgement without selecting a payment method (by selecting a free lodgement first, then switching to a paid lodgement afterwards, then submit before selecting payment method)

Attack Scenario

A malicious user can exploit this issue in order to bypass business rules, as described above.

Risk

As this issue requires authenticated access to the application and a significant level of technical knowledge to identify, the likelihood that this issue will be exploited is considered RARE. As manual controls were found to be in place for business rules such as payment processing, the impact of this issue being exploited is considered MINOR. Therefore, this issue is considered to present a **LOW** risk.

Recommendation

Modify the web application to check all payment details and applicable business logic when the lodgement is finalised, in addition to the current checks in place throughout the document lodgement process [**Recommendation APP-06**].

4.1.11 TCA-09 Denial of Service

Denial of service attacks seek to disrupt the business function being provided by an application. There are many forms of denial of service attacks however all target the ability of an application to achieve its intended goal and are therefore analysed in terms of the application’s context.

No potential denial of service conditions were identified during testing.

4.1.12 TCA-10 Logging Manipulation

Logs are a fundamental component of the intrusion detection process and often form much of the audit trail. In many applications all non-repudiation is provided by logs. Testing of log mechanisms seeks to verify that the data stored cannot be tampered with, disguised, or otherwise manipulated. Furthermore, it seeks to ensure that logs store a complete and thorough record of events.

No logging-related issues were identified during testing, as logging functionality in the administrator section correctly escaped injected malicious input.

5 External Vulnerability Assessment

5.1 Assessment Summary

Table 5: External Vulnerability Assessment Risk Summary

Test Component	Business Risk Exposure
External Vulnerability Assessment	<p>LOW</p> 

The following table is a summary of the security posture of the assessed server with reference to the following commonly identified security vulnerabilities:

- VA01 – Insecure / Inappropriate Services Exposed;
- VA02 – Missing Patches;
- VA03 – Outdated Software;
- VA04 – Information Disclosure; and
- VA05 – Misconfigured Security Settings.

Where a square is marked with a diamond (◆) at least one security vulnerability of that type was identified for the system.

Table 6: External Vulnerability Assessment Findings Summary

Ref	System	VA01	VA02	VA03	VA04	VA05
WS	Web Server	◆				◆

5.2 Assessment Findings

This section details vulnerabilities identified in the Federal Court eLodgment infrastructure through the execution of infrastructure security test cases as previously agreed. Vulnerability assessment was conducted from an anonymous perspective.

Specific details are only provided for test cases which resulted in findings.

5.2.1 VA01 – Insecure / Inappropriate Services Exposed

Servers typically provide access to TCP/IP ports in order to expose underlying services and/or functionality. From a functionality perspective, providing a large number of available ports and services means that the end user may communicate with the server in a variety of ways. However, from a security perspective having a large number of ports available to external users provides an

increased surface area for a malicious intruder to attempt to attack and therefore increases the likelihood of an attack.

Security best practice dictates that only the minimum number of services required in order for the system to perform its immediate business function should be made available. In addition, a number of services utilise protocols or provide access to underlying functionality that is considered to be insecure. Typically this occurs in services using protocols that transmit information (including authentication information) 'in the clear' or services providing access to outdated or legacy functionality.

Unknown Ports Exposed

Issue Details

During initial scanning of the target system, **stratsec** identified several ports to be open. However, no data was sent by these ports when a connection was initiated, and they did not respond to data sent during testing by **stratsec**. Specifically, the ports identified to be potentially unnecessarily open are:

- 1821/tcp
- 8402/tcp
- 8400/tcp
- 1433/tcp
- 8401/tcp
- 2200/tcp
- 210/tcp
- 389/tcp
- 53/tcp

Attack Scenario

No attack scenario is available for this issue as **stratsec** was unable to identify the services corresponding to the identified ports.

Risk

The likelihood and consequence of this issue being exploited are considered RARE and INSIGNIFICANT respectively, as the services corresponding to the identified open ports could not be identified. Therefore, this issue is considered to present a **LOW** overall risk.

Recommendation

Configure the relevant network access control device to only allow access to the ports required for business functionality on the assessed system [[Recommendation WS-01](#)].

5.2.2 VA02 – Missing Patches

Software vendors frequently issue updates in the form of patches and hotfixes to operating systems to address identified security vulnerabilities. The failure to apply these patches unnecessarily exposes a server system to potential compromise and patching is therefore a critical part of the security process.

Vulnerabilities in the server operating system may allow risks such as the disclosure of sensitive data, denial of service or system compromise to eventuate. Security best practice suggests that all vendor-issued patches that address known vulnerabilities be tested and applied promptly to server systems to address operating system security holes and minimise the possibility of a successful attack being performed on the server.

At the time of testing all software and associated patches reviewed by **stratsec** appear to be up-to-date. Note that even in internal testing, accurate version number were not able to be identified. This is inline with security best practice, however as such did not identify any instances of missing patches on the assessed system.

5.2.3 VA03 – Outdated Software

Outdated software may be vulnerable to exploits that have not been patched by the vendor as the product has been discontinued or superseded. Running outdated software may render a system vulnerable to attackers who may be able to exploit security weaknesses present in the outdated software.

To address this issue, security best practice requires that all software should be up-to-date and have all relevant security patches applied. Legacy software that is not able to be updated or patched must be subject to mitigating security controls such as reverse-proxy servers and host-based intrusion detection systems.

stratsec did not identify any outdated software on the reviewed system. All software identified on the assessed host appeared to be up-to-date.

5.2.4 VA04 – Information Disclosure

Server software installed on a system typically provides some information to a computer attempting to connect to the server, often including the name of the software and its version. This is standard behaviour for many servers, such as web and email servers. However, providing this sort of information to an attacker can enable them to identify a vulnerable piece of software and what exploits can potentially be used to compromise the server.

To address this issue, security best practice requires that where possible, server software should be configured to reveal as little information as possible without affecting the functionality of the program.

stratsec did not identify any services on the reviewed system(s) that disclose unnecessary information.

5.2.5 VA05 – Misconfigured Security Settings

Security settings include items such as configuration of audit logs, error handling behaviour, password management controls and default user accounts and associated access permissions. The impact of misconfigured security settings ranges from unauthorised information disclosure, non-compliance with regulatory requirements through to system compromise depending on the particular setting which has been misconfigured.

Hardening guides and security standards are available for the majority of operating systems, both from vendors and from independent organisations such as the Centre for Internet Security (CIS). It is recommended that systems be built in accordance with standards such as these, or internally

developed standards (if they exist), as the impact of individual settings can often be subtle but lead to security issues when combined with other issues.

Weak SSL Settings

Issue Details

stratsec identified that the web server software on the assessed system is configured to accept connections using cryptographically weak SSL ciphers suites and SSL 2.0. The specific weak cipher suites found to be used are as follows:

- SSLv2
 - EXP-RC2-CBC-MD5 Kx=RSA(512) Au=RSA Enc=RC2(40) Mac=MD5 export
 - EXP-RC4-MD5 Kx=RSA(512) Au=RSA Enc=RC4(40) Mac=MD5 export
- SSLv3
 - EXP-RC2-CBC-MD5 Kx=RSA(512) Au=RSA Enc=RC2(40) Mac=MD5 export
 - EXP-RC4-MD5 Kx=RSA(512) Au=RSA Enc=RC4(40) Mac=MD5 export

Attack Scenario

A malicious user could attempt to brute-force session keys where a weak cipher suite was used to decrypt data being transmitted between client and server. Alternatively, an attacker could perform a man-in-the-middle attack to compromise communications using SSL 2.0.

Risk

The likelihood and consequence of this issue being exploited are considered RARE and MINOR respectively, as the overall design of the Internet makes exploiting this issue technically infeasible. Therefore, this issue is considered to present a **LOW** overall risk.

Recommendation

Disable weak SSL ciphers and SSL 2.0 support, and instead implement TLS 1.0 or SSL 3.0 with ciphers considered cryptographically strong [[Recommendation INF-02](#)].

Annex A Review Scope

In Scope

The following tasks have been identified as components to be executed for the development of the eLodgment Security Assessment.

Project Administration, Preparation & Reporting

Table 7: Project High Level Task Analysis - Administration

Stream Item	Task Outline(s)
Project Planning & Project Management	<ul style="list-style-type: none"> • Develop project plan • Socialise project plan with client for approval • Ongoing project reporting and status reports as required
Information Gathering & Documentation Review	<ul style="list-style-type: none"> • Review design documents • System familiarisation exercise with Federal Court personnel • Review application context, functional & technical specifications • Review of dependencies on other components/systems
Document Findings & Deliver Report	<ul style="list-style-type: none"> • Document process followed • Document vulnerabilities and likely attack vectors • Document recommended remediation steps • Internal QA & Review • Deliver report and recommendations <ul style="list-style-type: none"> ○ Documentary form ○ Presentation

Application Penetration Test

Note that the application under review through this project is defined to be defined upon project initiation

Table 8: Scope task analysis – Application Penetration Test

Stream Item	Task Outline(s)
Test Case Development	<ul style="list-style-type: none"> • Document security test cases <ul style="list-style-type: none"> ○ Assess compliance to security principles

Stream Item	Task Outline(s)
	<ul style="list-style-type: none"> ○ Assess resilience to malicious use and/or manipulated data ● Deliver defined test cases to client for review & approval ● <i>Note that test cases provided at commencement of the project will provide a “point of departure” for testing and are subject to further revision and addition as testing progresses.</i>
Application Security Testing	<ul style="list-style-type: none"> ● Perform application penetration testing tasks on the system and related components <ul style="list-style-type: none"> ○ Validate design, architecture & other identified risks ○ Execute defined test cases ○ External assessment (application security issues identified via Internet presented application) ● Assess security mechanisms from an unauthenticated perspective ● Assess security mechanisms against a range of user types (four roles are known to exist) ● Review ability to withstand attack from injected or manipulated code ● Assess scenarios through which a denial of service condition can be introduced ● Review availability and appropriateness of audit trails ● Assess user access controls, user segregation and authentication security

Out of Scope

All other items not identified above are considered outside scope. These specifically include:

- Assessment of applications not specifically defined in “Application Security Assessment” phase
- Review of application source code
- Infrastructure, network or server security assessment
- Review, development or documentation of operational procedures and security policy unless included in a project phase
- Load testing as a vector for achieving Denial of Service
- Social engineering as a mechanism for obtaining network information
- Specific legal advice. Where stratsec believes or identifies that issues exist that could introduce legal liability or other considerations, these will be documented and raised.

However, as stratsec is not a law firm, specific interpretation of the impact of these issues on regulatory and legal compliance is not within the scope of the stratsec report.

Annex B Risk Rating Tables

Likelihood

The likelihood rating of an issue encompasses both the likelihood of the issue being identified and attacked as well as the likelihood of an attack being successful. This is evaluated by taking into consideration the following aspects:

- Exploitability
- Difficulty and technical knowledge or skill required to identify and exploit the issue
- Time or resources required to mount a successful attack
- Availability of exploit code and automated attack tools
- Reproducibility
- Ease of reproducing a successful attack
- Additional requirements for the attack to be successful, for example:
 - Victim user must be logged in
 - Some level of interaction by the victim user is required
- Discoverability
- Number of instances of the vulnerability identified in the system
- Level of authentication required to access affected components
- Accessibility of the system
- Degree of specific Insider knowledge required
- Frequency
- History of the issue in the industry
- Existence of self-propagating malware targeting the issue

These factors will be employed to formulate a final likelihood rating for a given issue and a table of examples is provided below.

Table 9: Likelihood Rating Table

Likelihood Rating	Example Frequency	Example Scenario
Rare	1 incident every 5+ years	Highly skilled and determined attacker with substantial resources
Unlikely	1 incident every 2 years	A skilled attacker with some degree of insider knowledge
Moderate	1 incident every year	An attacker with technical knowledge
Likely	1 incident every 6 months	Published and widely available exploit code exists
Almost Certain	1+ incidents every month	Worm propagating in the wild or widespread availability of an automated attack tool

Consequence

The consequence rating assesses the significance of exposure to a particular risk. This is evaluated by considering the impacts to the affected system and the underlying business. The factors under consideration are outlined in the following table.

Table 10: Consequence Rating Table

Determination of Consequence Rating					
Impact	Insignificant	Minor	Moderate	Major	Catastrophic
System – Confidentiality	Disclosure of public information	Minor disclosure of commercial-in-confidence information	Major disclosure of commercial-in-confidence information	Minor disclosure of highly-confidential information	Major disclosure of highly confidential information
System – Integrity	Unauthorised modification of public data	Small-scale unauthorised modification of private data	Large-scale unauthorised modification of private data	Small-scale unauthorised modification of trusted data	Large-scale unauthorised modification of trusted data
System – Availability	Minor increase in processing load	Minor outage in a business system	Outage or unavailability of a business system	Extended unavailability or outage of a business system	Unavailability or outage of a business-critical system
Brand or Reputation	Complaints from small number of customers	Complaints from small number of customers across a broader customer base	Complaints from a large number of customers and localised media coverage	Short term adverse large scale media coverage	Extended adverse large scale media coverage
Regulatory and Legal	Warnings for minor breaches	Formal caution for regulatory breaches or threat of legal proceedings	Targeted audit / investigation by regulator or minor legal proceedings brought against the organisation	Fines imposed and negative media coverage or major legal proceedings brought against the organisation	Service line closed down
Management Impact	A minor event or issue which causes minimal disruption	Minor disruption absorbed through normal management activities and no compromise of technology direction or policy	Disruption absorbed via additional effort to ensure technology direction or policy is not compromised	Considerable deviation and significant compromise of technology direction or policy	Cancellation of the service line and significant recovery and remediation costs incurred

Overall Risk

A risk measure or rating is determined by the likelihood and adjusted consequence ratings. Use the matrix below to determine each risk.

Table 11: Overall Risk Rating Table

		Consequence				
		Insignificant	Minor	Moderate	Major	Catastrophic
Likelihood	Almost Certain	HIGH	HIGH	EXTREME	EXTREME	EXTREME
	Likely	MODERATE	HIGH	HIGH	EXTREME	EXTREME
	Moderate	LOW	MODERATE	HIGH	EXTREME	EXTREME
	Unlikely	LOW	LOW	MODERATE	HIGH	EXTREME
	Rare	LOW	LOW	MODERATE	HIGH	HIGH