MOBIUS
CONSULTING

Actualising change

22SEVEN – WEB APPLICATION PENETRATION TEST REPORT

Nature of service: Web Application Penetration Test

Client: 22Seven

Date: July 2020

A member of the Mobius Group

MOBIUS
CONSULTING

Actualising change

TABLE OF CONTENTS

www.mobiusconsulting.co.za

A member of the Mobius Group

MOBIUS
CONSULTING
Actualising change

1. EXECUTIVE SUMMARY

Mobius Consulting performed a web application penetration test of the SpikeData API's for 22Seven that was completed during July 2020. The goal of the testing was to identify vulnerabilities and security weaknesses which could be exploited by an attacker to gain access to sensitive information or restricted functionality. An outcome of this testing was to provide 22Seven with a view of the potential security risks as well as recommended remedial advice which can be used to improve the resilience of the SpikeData web application and associated API endpoints.

The scope of the testing consisted of manual penetration testing to identify vulnerabilities and security weakness of the web applications, API's, and attempts to exploit these vulnerabilities in a similar manner as a hacker would to compromise the confidentiality, integrity, or availability. This report includes the outcomes of the testing including all vulnerabilities we found, categorisation of the vulnerabilities based on their severity, recommended remedial actions, and suspected root cause associated with each vulnerability.

Overall, the SpikeData web applications and API's were found to have a robust security posture. There were two vulnerabilities of a low severity:

- Client Side Content Blocking Bypass (Severity: **Low**) – Certain site functionality is accessible to the unauthenticated user by changing the DOM properties of the site. It is recommended that content blocking be revisited.
- Backup/Test files (Severity: **Low**) – These files may give potential attackers a better understanding of the overall functionality, or setup of the web application. It is recommended these files/folders be removed.

The following were identified as the suspected main root causes of the majority of vulnerabilities contained in this report:

- Misconfiguration of the application/system

22Seven should address the vulnerabilities and security weakness included in this report in order to improve the security posture of the SpikeData web application and API's. In addition to addressing the vulnerabilities, 22Seven should review the root cause areas to determine how these might be improved to prevent similar vulnerabilities from being re-introduced in the future.

MOBIUS CONSULTING
Actualising change

2. SCOPE AND APPROACH

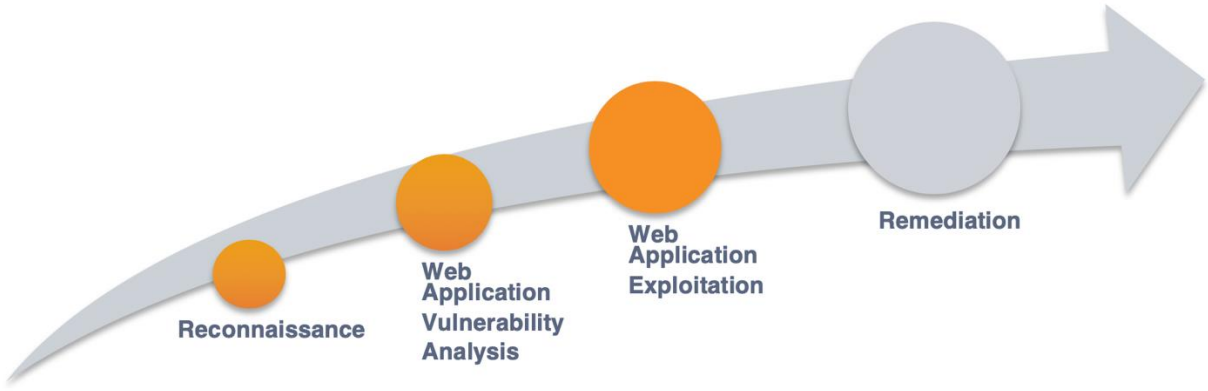
The following section contains a description of testing activities as well as the high-level findings and recommendations from the assessment.

2.1. SCOPE OF WORK

Mobius completed an authenticated and unauthenticated web application penetration test during June 2020, to identify vulnerabilities and measure the effectiveness of the in-scope web application.

2.2. TESTING METHODOLOGY

Web Application Penetration Testing Approach:



Reconnaissance

Through port enumeration, Mobius detected an open HTTP port for the identified web server and identified the type of service running on each port. A visibility profile, or "map", contained active web application services that were found, e.g. IIS webserver running an ASP.NET application. This information was critical for the next phases as it determined the scope and nature of the security.

Whenever possible, Mobius also determined the version or patch level of the service by fingerprinting or examining HTTP protocol requests (GET, HEAD etc.).

Our information gathering included cross-referencing with databases of known vulnerabilities relating to the Open Web Application Security Project (OWASP) top ten vulnerabilities.

Note: This phase was primarily conducted utilising the automated toolset at our disposal.

Web Application Vulnerability Analysis

Based on the results of the reconnaissance phase, we used our web application vulnerability assessment software, and other techniques in an attempt to identify and validate vulnerabilities that may allow unauthorised access to web application components (core application, the underlying application framework or the back-end database).

The vulnerability analysis consisted of the following tests:

- Examining publicly visible elements of the selected web applications;
- Testing the inputs and outputs of the web application:
    - Input validation (SQL Injection (SQLi), Cross-Site Scripting (XSS), input validation, and directory path traversal);
    - Session management (hijacking and fixation);
    - Authentication and authorisation (registration process validation, form / infrastructure authentication mechanisms, and password strength controls); and
    - Configuration (application patch levels, caching and sensitive information stored in configuration files).

Note: The main emphasis for this engagement was the web application vulnerability analysis that produced a ranking of the most severe vulnerabilities.

Web Application Exploitation

The exploitation phase included exploiting the major web application vulnerabilities identified in an attempt to gain successively greater levels of access to data or network resources, i.e. reverse shell through the web application onto the operating system of a server.

The exploited component could be used as a launching point for further information gathering and penetration attacks against other systems. It is important to note that we made every attempt not to disturb the status quo and business activities. In the case of a serious compromise, you would have been immediately notified. Prior to any exploitation, as part of the Mobius risk management strategy, we would inform you of our intent, seek authorisation for exploitation, specify the time of exploitation and work closely with you to minimise risk.

Reporting

This phase concluded testing with thorough documentation of the vulnerabilities that were discovered. This includes detailed steps to reproduce the vulnerability, screenshots, a description of the impact and exploitability as well as various options for remedial action to mitigate the risk.

3. DETAILED FINDINGS

3.1. CLIENT SIDE CONTENT BLOCKING BYPASS

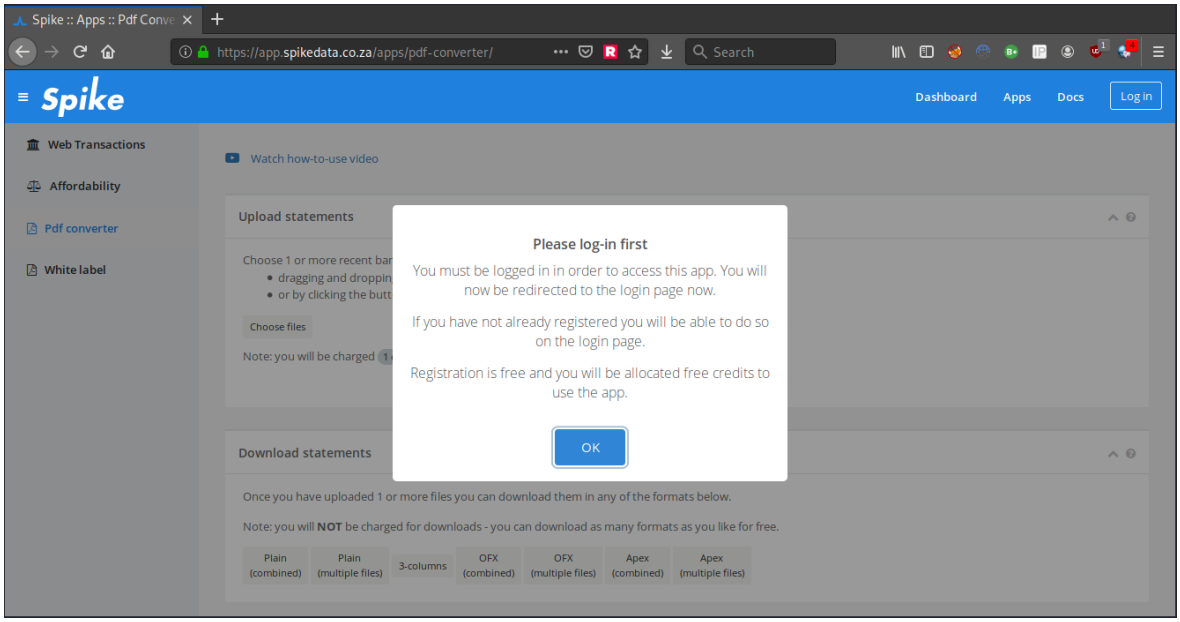
Risk Rating: **Low**

Affected Host(s):

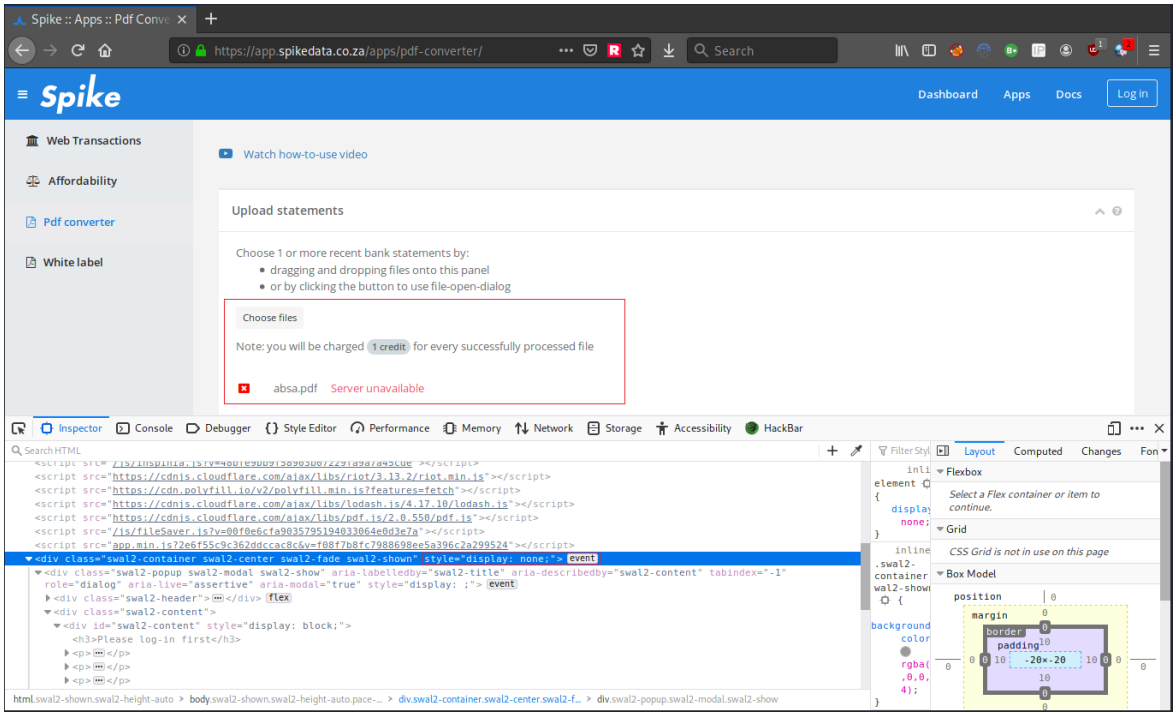
- https://app.spikedata.co.za/apps/

Description:

When trying to access the various apps on the site, an unauthenticated user is presented with the below screen requesting them to log in,



This "content blocking" is easily bypassed by changing the DOM settings of the rendered page. Below we can see the style set to "***display: none;***" giving us access to the pdf upload functionality,



It should also be noted that the full functionality still requires valid session, user and api tokens to function correctly. Hence this vulnerability being marked as a low severity item.

Remediation:

It's recommended that this content blocking perhaps be revisited. Whilst in its current form its safe from exploitation (due to the use of session tokens), it does disclose information about the sites functionality, and could help attackers with future exploitation efforts.

3.2. BACKUP/TEST FILES

Risk Rating: **Low**

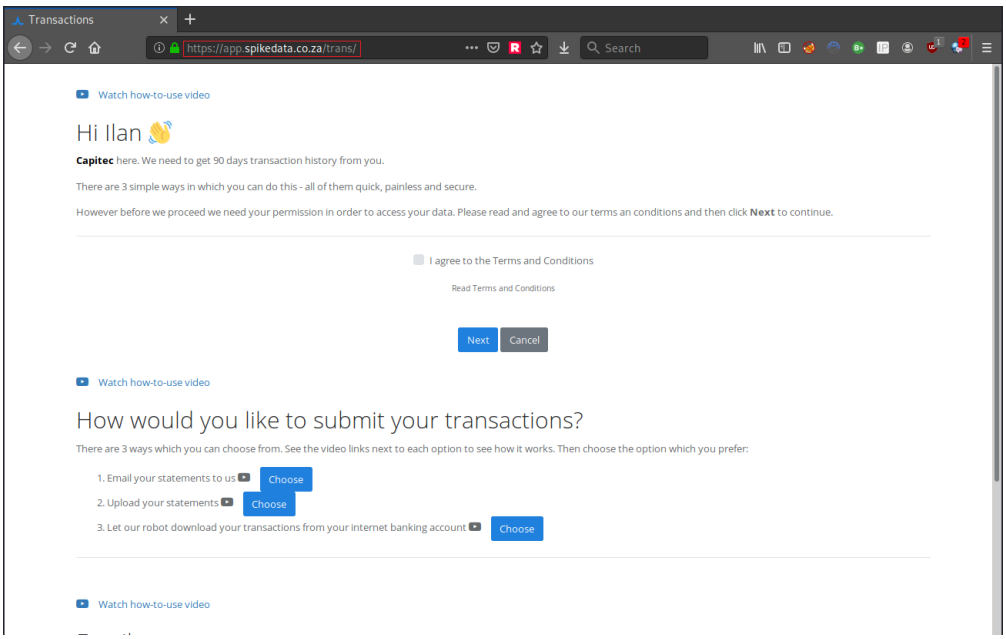
Affected Host(s):

- api-v6.spikedata.co.za
- app.spikedata.co.za

Description:

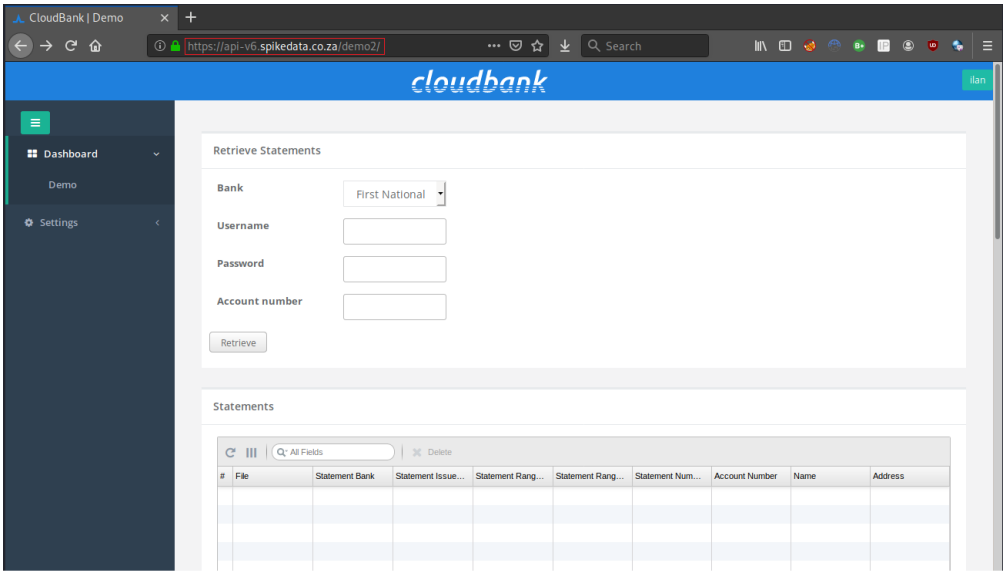
While testing the web applications and api endpoints a few backup/test files and folders were found. Whilst these files and folder themselves don't present themselves as a serious vulnerability, they do aid potential attackers in understanding the inner workings and setup of the backend server and system. Below are the affected links and some example screenshots:

- https://app.spikedata.co.za/hi.txt
- https://app.spikedata.co.za/trans/



- https://api-v6.spikedata.co.za/demo/
- https://api-v6.spikedata.co.za/beta/#overview

- https://api-v6.spikedata.co.za/demo2/
- https://api-v6.spikedata.co.za/demo3/



It should also be noted that these pages were accessible from an unauthenticated perspective.

Remediation:

It's recommended that all unnecessary files and folders be removed from internet facing production servers, as this minimises the amount of information a potential attacker could infer about the system and underlying application setup.

MOBIUS CONSULTING
Actualising change

4. APPENDIX A: GLOSSARY OF VULNERABILITY RATINGS

RISK RATING	DEFINITION
Critical	The vulnerability represents a fundamental risk to the confidentiality, integrity or availability of the system / application. These vulnerabilities could be used to ensure the network remains unavailable for extended periods of time. These vulnerabilities should receive a critical priority for remediation.
High	The vulnerability represents a significant risk to the confidentiality, integrity or availability of the system / application. These vulnerabilities could be used to gain access to privileged areas of the application or the underlying infrastructure as well as access to confidential or sensitive information. These vulnerabilities should receive a high priority for remediation.
Medium	The vulnerability represents a moderate risk to the confidentiality, integrity or availability of the system / application. These vulnerabilities could typically be used together as part of a larger attack or provide the attacker with further information about the application/system. In some cases, these vulnerabilities could be used to gain access to confidential of sensitive information. These vulnerabilities should receive a medium priority for remediation.
Low	The vulnerability represents a low risk to the confidentiality, integrity or availability of the system / application. These vulnerabilities will rarely result in a significant compromise when considered in isolation but could provide an attacker with further information or leverage to conduct a larger attack. These vulnerabilities should receive a low priority for remediation.

Table 1 – Risk rating glossary