

WebTMA Application Security Assessment

CONFIDENTIAL

Project Timeframe: June 2021

Submitted to: TMA Systems
1876 Utica Square, Third Floor
Tulsa, OK 74114



True Digital Security, Inc.

Corporate Address

P.O. Box 35623
Tulsa, OK 74153

Florida Office

1401 Forum Way
Suite 100
West Palm Beach, FL 33401

p. 800.757.6937

f. 561.835.0065

Oklahoma Office

1350 S. Boulder Ave
Suite 1100
Tulsa, OK 74119

p. 866.430.2595

f. 877.720.4030

New York Office

111 Smithtown By-pass
Suite 104
Hauppauge, NY 11788

p. 631.366.5155

f. 631.366.0979

CONTENTS

- 1. ASSESSMENT SUMMARY 3
 - 1.1 Project Information 3
 - 1.2 Summary of Assessment Findings 3
 - 1.3 Assessment Scope & Environment 3
 - 1.4 Methodology & Narrative 4
- 2. WEB APPLICATION PENETRATION TEST 6
 - 2.1 Summary 6
 - 2.2 Vulnerabilities & Exploitation Results 6
 - 2.2.1 Insecure Direct Object Reference (IDOR) Reveals Sensitive Information 6
 - 2.2.2 Outdated/Vulnerable JavaScript Libraries 8
 - 2.2.3 Verbose Error Pages Expose Internal Server Information 9
 - 2.2.4 Browser Security Hardening Header Flags Not Set 12
- APPENDIX A. GENERAL ASSESSMENT METHODOLOGY 13

DOCUMENT REVISION HISTORY


Revision Date	Authors	Notes
06/25/2021	Aaron Moss	Initial document
08/27/2021	Aaron Moss	Remediation Testing info added

1. ASSESSMENT SUMMARY

1.1 Project Information

Company	True Digital Security, Inc. (TRUE)
Client	TMA Systems
Project Type	WebTMA Application Security Assessment
Project Timeline	June 2021 Remediation – August 2021
Team Members	Aaron Moss, Senior Security Consultant (Project Lead) Oscar Gutierrez, Security Consultant

1.2 Summary of Assessment Findings

Assessment Component	Overall Risk Rating	Rationale
WebTMA Application Security Assessment	 <p>High Risk</p>	<p>Remediation Testing – TRUE found all vulnerabilities identified to be remediated. No residual risk exists at this point.</p> <p>During testing, TRUE’s Red Team identified an insecure direct object reference vulnerability which reveals secret keys and various cloud services API keys that could be more severe in the production version of WebTMA. In addition, TRUE discovered outdated JavaScript libraries and web server verbose errors that reveal physical server paths and internal IP addresses.</p> <p>TRUE rates the overall security of the WebTMA web application to be at a high risk of compromise.</p>

1.3 Assessment Scope & Environment

TMA Systems requested TRUE’s Red Team to perform a web application security assessment of the Sandbox version of the WebTMA web application. TMA Systems provided the Red Team with three sets of credentials for testing – User/Admin, Technician, and Requestor. The following application was in-scope for testing:

- <https://sandbox.webtma.com>

1.4 Methodology & Narrative

Testing was conducted in two phases, testing as unauthenticated and authenticated users. TRUE’s assessment methodology for the application testing follows a process of mapping, reconnaissance, exploitation, and post-exploitation.

TRUE began testing the web servers by mapping the hosted directories using **Gobuster** and **Dirb** in search of hidden directories. TRUE manually crawled the web apps using **Burp Suite Professional**, an intercepting web proxy. TRUE performed automated and manual testing of the pages and directories discovered in the application mapping phase. **Nikto** was used to perform in-depth analysis of the web servers and detect common vulnerabilities.






During testing, the Red Team identified an authenticated insecure direct object reference (IDOR) vulnerability which reveals secret keys and API keys to various cloud services (including Dropbox, Google, and SharePoint), out-of-date JavaScript libraries, and verbose error messages which reveal physical server paths and internal IP addresses.

Tools Used

Tool	Description
Nikto	Web application security scanner
Dirb	Web hidden directory scanner
Kali Linux	Linux-based security distribution
Burp Proxy Professional	Web application assessment platform and scanner
Google Chrome	Web browser used for manual web application testing
Burp Scanner	Web application scanning tool

Risk Rating

TRUE assigns a risk level to each identified issue discovered during the assessment. This risk level is based on expert analysis of the issue, its environment, and the severity of the identified issue. The suggested remediation timeline is derived from the potential for system compromise, overall damage to the environment/system, and criticality of information theft.

Risk Level	Description
Critical 	Risk of immediate exploitation or critical level of exposure that can lead to system or application compromise or information theft. Remediation should be conducted immediately or as soon as possible.
High 	Significant risk of severe impact to system or application security. Remediation should be prioritized or within (1) month.
Elevated 	Risk of an elevated nature that may expose sensitive information or may be used in conjunction with other issues aiding in exploitation. Remediation should be prioritized based on the criticality of the system and information exposed or within (3) months.
Moderate 	Risk of a less critical nature that may potentially lead to information theft or misuse. Remediation should be included in the next security update or within six (6) months.
Minimal 	Risk of a non-critical nature that may lead to misuse or stability loss or enhancement features, which will improve security. This issue should be noted for reference; however, remediation is not strictly necessary.

2. WEB APPLICATION PENETRATION TEST

2.1 Summary

TRUE’s Red Team performed penetration testing of TMA Systems’ Sandbox version of the WebTMA web application, mimicking both an external unauthenticated entity attacking from the internet and an authenticated user. The Red Team performed the application penetration test with all TMA Systems security controls in place. The test was designed to attempt to access any sensitive information, gain code execution, force unexpected server behavior, or access administrative functionality.

2.2 Vulnerabilities & Exploitation Results

2.2.1 Insecure Direct Object Reference (IDOR) Reveals Sensitive Information

Issue Risk Level	High	<div style="width: 100%; height: 15px; background: linear-gradient(to right, green, yellow, orange, red);"></div>
------------------	------	---

TRUE verified this vulnerability as remediated on 8/27/2021.

Description

Insecure direct object reference (IDOR) vulnerabilities are a type of access control vulnerability which can arise when an application uses user-supplied input to access an object directly. TRUE identified an authenticated IDOR vulnerability which reveals secret keys, client keys, and API keys for various cloud services, including SharePoint, Dropbox, Google Drive, and Microsoft OneDrive.

In this case, the Red Team was able to access `/LinkedDocument/AvailableProviders` directly with authentication. The link was first identified while testing the application as the Administrator user, however, during testing, the keys and other sensitive information were accessible from the Admin, Technician, and Requestor logins. The Red Team attempted access while unauthenticated, but this was unsuccessful.

Figure 2-1 – IDOR Vulnerability Displaying Client, Secret and API Keys – HTTP Request

```

Request
Pretty Raw Hex \n ☰
1 GET /LinkedDocument/AvailableProviders HTTP/1.1
2 Host: sandbox.webtma.com
3 Cookie: .AspNetCore.Antiforgery.ORB623DQXAA=
  CfDj8NpdtNvZAJhEoy6tHdM3UDDZB4kdL90ciI6Yt_KSINEJWlpnzC-7VBZM8mrdU5UE_nc9PisBZDze4EmOGyV9T3Fy
  xC9TvsVX_Je0Vklz1_2DsMT_4vraw_ENNx2IPXgl_GY4kQMalj9tsETBIJP36g; WebTMALogin=
  VHJ1ZURpZ210YwXUZWN0%7CVHJ1ZSBEaWdpdGFs; .AspNetCore.Session=
  CfDj8NpdtNvZAJhEoy6tHdM3UDAgNYMYcTgJjo2z59qawhmijWUYOqRsGpueoyOJUXNd9eeLaf%2BJ82NjbZdxI9epeZh
  qZ1MoxAU2SSkWnEmoBBjDhrp8%2FA%2BQMrKBj85RjmgdYY%2FIivrCe5PsOT%2FvBh1WWpGzNoz6tkSDrLVzm9x27BCC
  ; .AspNetCore.Identity.Application=
  CfDj8NpdtNvZAJhEoy6tHdM3UDCbuoNoEFA541hZeqaNGxakn0zeN1HOT1uwtGeo8kKkAI9oTuibPVvXonlpMrKTKJd0
  vap50DQ-YeHdFezRDZQ5iytjG8JVCvocWNADvBs6XgISR3RRcDudEcwr0_b73SCg7KKjAPlh8x2T3B2Nwmjx_WXTSd49x
  BVL9PqQvOFFz4dGjzA1KK00xHuQEzWflrWqA999F1I3XIjaCwblOb7kt9KKhNNigQ-P6vMoSAHifIa3Zw1ffhxyznDTI
  jeTkDFZS7kAUtOhJpdvhoCOjTEKBfeGo665YosozYjq92AQ59SMka1SiXB1OWZIHgvI
4 Sec-Ch-UA: "Chromium";v="91", " Not;A Brand";v="99"
5 Accept: application/json, text/plain, */*
  
```


- OWASP Cheat Sheet Series – Insecure Direct Object Reference Prevention – https://cheatsheetseries.owasp.org/cheatsheets/Insecure_Direct_Object_Reference_Prevention_Cheat_Sheet.html
- OWASP Cheat Sheet Series – Access Control Cheat Sheet – https://cheatsheetseries.owasp.org/cheatsheets/Access_Control_Cheat_Sheet.html

2.2.2 Outdated/Vulnerable JavaScript Libraries

Issue Risk Level	Moderate	
------------------	----------	---

TRUE verified this vulnerability as remediated on 8/27/2021.

Description

TRUE identified several instances of vulnerable and out-of-date JavaScript libraries on the WebTMA web application.

- AngularJS versions before 1.8.0 have known vulnerabilities, including cross-site scripting (XSS) and prototype pollution.
- Bootstrap versions before 4.3.1 are vulnerable to several vectors of cross-site scripting (XSS) attacks through various properties, including the data-template, data-content, data-title, and data-target properties.
- jQuery versions before 3.5.0 are vulnerable to Cross-site Scripting (XSS) attacks when a specially crafted input is passed to the jQuery.htmlPrefilter regex included with the library.

Affected Locations

- AngularJS 1.7.8 – <https://sandbox.webtma.com/lib/angular/angular.min.js>
- AngularJS Animate 1.7.8 – <https://sandbox.webtma.com/lib/angular/angular-animate.min.js>
- Bootstrap 3.3.7 – <https://sandbox.webtma.com/lib/bootstrap/dist/js/bootstrap.min.js>
- jQuery 3.4.1 – <https://sandbox.webtma.com/help/Resources/Scripts/jquery.min.js>
- jQuery 3.3.1 – <https://sandbox.webtma.com/lib/jquery/jquery.min.js>

Remediation Recommendations

Test and update all JavaScript libraries to the latest versions available. It is recommended to implement automated checks for security vulnerabilities within the application’s development processes.

References

- OWASP Top Ten 2017 – A9 – Using Components with Known Vulnerabilities – https://owasp.org/www-project-top-ten/2017/A9_2017-Using_Components_with_Known_Vulnerabilities

2.2.3 Verbose Error Pages Expose Internal Server Information

Issue Risk Level	Moderate	<div style="width: 30%; height: 15px; background: linear-gradient(to right, #92d050, #fff);"></div>
------------------	----------	---

TRUE verified this vulnerability as remediated on 8/27/2021.

Description

The WebTMA application exposes internal server information such as application code physical paths and internal IP addresses through certain error pages. TRUE’s Red Team caused the server to expose the physical path of WebTMA Sandbox application code through an *Unauthorized 401* HTTP request to a specific page as a Requestor user.

Figure 2-3 – Physical Path Disclosed

Requested URL	http://sandbox.webtma.com:80/QPContractor/ValidationFieldData/?pageFieldId=2707&configCode=1&searchValue
Physical Path	C:\Sites\WebTMA7.Sandbox\QPContractor\ValidationFieldData\
Logon Method	Not yet determined
Logon User	Not yet determined

The Red Team was also able to affect the server to disclose its internal IP address through a GET request to */aspnet_client/* using *HTTP/1.0*, an older version of HTTP, initially in the *Location* header, followed by an error page which also disclosed the physical path of the application code on the server.

Figure 2-4 - HTTP/1.0 GET Request and Response – Location Header

Request

Pretty Raw Hex \n

```
1 GET /aspnet_client HTTP/1.0
2
3
```

? ⚙️ ⏪ ⏩ Search...

Response

Pretty Raw Hex Render \n

```
1 HTTP/1.1 301 Moved Permanently
2 Content-Type: text/html; charset=UTF-8
3 Location: http://192.168.51.210/aspnet_client/
4 Server: Microsoft-IIS/10.0
5 X-Powered-By: ASP.NET
6 Date: Wed, 23 Jun 2021 19:39:35 GMT
7 Connection: close
8 Content-Length: 159
9
10 <head>
    <title>
      Document Moved
    </title>
  </head>
11 <body>
    <h1>
      Object Moved
    </h1>
    This document may be found <a href="http://192.168.51.210/aspnet_client/">here</a>
  </body>
```

Figure 2-5 – 403 Forbidden Error Page Discloses Internal IP Address and Physical Path

HTTP Error 403.14 - Forbidden

The Web server is configured to not list the contents of this directory.

Most likely causes:

- A default document is not configured for the requested URL, and directory browsing is not enabled on the server.

Things you can try:

- If you do not want to enable directory browsing, ensure that a default document is configured and that the file exists.
- Enable directory browsing using IIS Manager.
 - Open IIS Manager.
 - In the Features view, double-click Directory Browsing.
 - On the Directory Browsing page, in the Actions pane, click Enable.
- Verify that the configuration/system.webServer/directoryBrowse@enabled attribute is set to true in the site or application configuration file.

Detailed Error Information:

Module	DirectoryListingModule	Requested URL	http://192.168.51.210:80/aspnet_client/
Notification Handler	ExecuteRequestHandler	Physical Path	c:\inetpub\wwwroot\aspnet_client\
Handler	StaticFile	Logon Method	Anonymous
Error Code	0x00000000	Logon User	Anonymous

Affected Locations

The underlying web server (Microsoft IIS) itself appears to be affected.

Remediation Recommendations

Unless needed for business purposes, TRUE recommends that TMA Systems disable the use of HTTP 1.0 requests on all web servers. TRUE recommends creating a custom error page that does not disclose any sensitive information about the underlying application and web server infrastructure.

References

- CWE-212: Improper Cross-boundary Removal of Sensitive Data – <https://cwe.mitre.org/data/definitions/212.html>

2.2.4 Browser Security Hardening Header Flags Not Set

Issue Risk Level	Minimal 
------------------	---

TRUE verified this vulnerability as remediated on 8/27/2021.

Description

The web applications are missing two security focused flags which, when applied, will strengthen the application from attacks stemming from XSS, CSRF, and other attacks. These settings require support from the user’s browser; most current browsers support these options. If other vulnerabilities were present in the web application, the risk level for this vulnerability may be elevated.

Security Flag	Reference
Strict-Transport-Security (HSTS)	The HTTP Strict Transport Security policy defines a timeframe where a browser must connect to the web server via HTTPS. The ‘max-age’ parameter defines the length of time which the browser must connect over HTTPS. Without this flag, an attacker could possibly manipulate a user session to access insecure HTTP pages, and gain access to sensitive data.
X-XSS-Protection	The X-XSS-Protection flag instructs the browser to not render a web page in the event of a browser detected cross-site scripting (XSS) attack.

Affected Locations

Responses are being generated throughout the application without the flags.

Remediation Recommendations

TRUE recommends evaluating the inclusion of the identified security-related HTTP headers and flags.

References

- Mozilla – HTTP Headers – Strict-Transport-Security – <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security>
- Mozilla – HTTP Headers – X-XSS-Protection – <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-XSS-Protection>

APPENDIX A. GENERAL ASSESSMENT METHODOLOGY

The following phased methodology was employed by TRUE to assess the organization's security controls against real-world attacks.

Phase 1: Reconnaissance and System Identification

TRUE used automated toolsets and manual processes to identify known and unknown components of the system under review to gain a complete picture of the target environment and identify likely targets of attack. Examples of techniques TRUE may have used include network scanning, website crawling, forced browsing, public dataset research, network traffic capture, binary analysis, and reverse engineering.

Phase 2: Vulnerability Analysis

TRUE used automated vulnerability management toolsets and manual processes to identify and verify known vulnerabilities and misconfigurations. False positives and false negatives were reviewed, and a risk profile with impact and likelihood metrics was determined. Higher criticality security vulnerabilities, if detected, were verified and escalated in phase 3.

Phase 3: Exploitation and Vulnerability Verification

If higher criticality security vulnerabilities were uncovered, TRUE attempted to exploit any security weakness identified using sophisticated real-world techniques tailored to mimic actual attack methodologies. Examples of attacks TRUE may have used within this phase included, but are not limited to: network exploitation, application exploitation such as SQL Injection, wireless attack, password cracking, man-in-the-middle, lateral movement, and backdoor and shell access. TRUE ensured that common attack techniques such as those listed in the SANS Top 20 and the OWASP Top 10 were verified.

Phase 4: Reporting

After performing a thorough review of your environment to identify risks to your business, TRUE's team of experts produced a detailed report, consisting of all findings along with recommended, prioritized countermeasures to ensure security best practices can be successfully incorporated.