OPEN SECURITY

ACME, INC.: ACME ULTIMATE PENTEST REPORT

PREPARED BY:

GEOFFREY PAMERLEAU

MICHAEL PLEASANT

LAST REVISION: JULY 5, 2020

210-446-7691

OPENSECURITY

SUPPORT@OPENSECURITY.IO

TABLE OF CONTENTS

EXECUTIVE SUMMARY

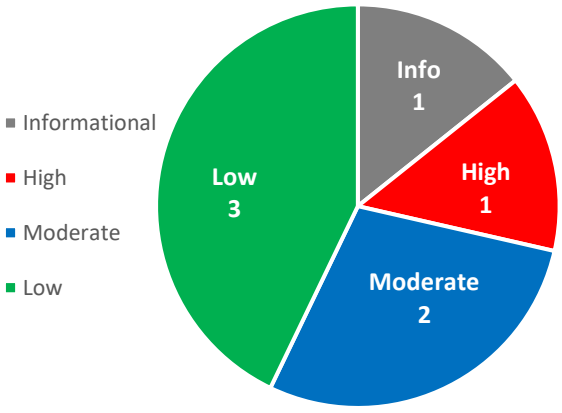
OVERVIEW

Open Security performed a Web Application Penetration test of the Acme Ultimate Web Application to identify security issues and recommend areas for improvement.

Acme Ultimate is a Software-as-a-Service (SaaS) solution utilized by customer hospitals and clinics for clinical workflows. The primary focus of the engagement was to asses if any discovered vulnerabilities could be used to access patient information without authentication or across account boundaries of customers using the Acme solution.

Testing occurred from 20-31 January 2020 and consisted of authenticated and unauthenticated access to the *staging* instance of the application which was hosted at *pentest.acme-ultimate-dev.com* as well as source code access for review of functions of interest. All functionality of the target site was considered in-scope. The Acme Ultimate app was assessed both with and without additional security controls in place. For more information see *Appendix B: Scope*.

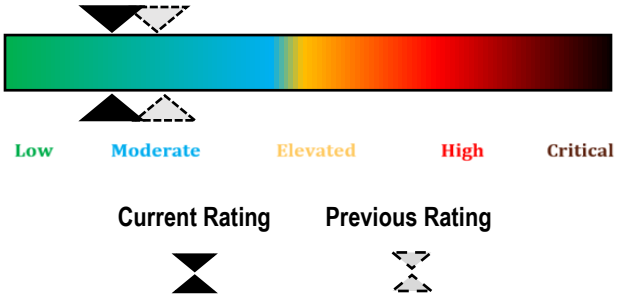
FINDINGS BREAKDOWN



Findings grouped by risk severity

CURRENT RISK ASSESSMENT

Open Security assesses the Acme Ultimate Web App's overall security risk to be: Low/Moderate



KEY FINDINGS

During this Penetration Test seven (7) new vulnerabilities were discovered. One (1) of these vulnerabilities was assessed as **high** risk and two (2) of were assessed as **moderate** risk, while the remaining three (4) were assessed as **low** or informational risk.

- Authentication was consistently enforced across the application, utilizing recognized best-practices for the underlying technology. **No method was discovered to access patient information without logging into the application as a valid user.**

Through a combination of both of the **moderate** risk vulnerabilities *it was possible* to change another user's password. This made it possible for a malicious user of one account to potentially login as users of other accounts. **The complexity of the attack path (chaining together 2 different vulnerabilities) combined with the requirements to bypass the WAF and have a valid account to use significantly decreases the likelihood of this happening. See the *Methodology* section for an explanation of this attack.**

IDENTIFIED TRENDS

- There was a lack of verification on user supplied values in many locations in the application. While the WAF successfully prevented low-skilled attackers from exploiting this issue, and entirely prevented it in some places, validation of user input should still be performed by the application.
- User Role checks were not consistently performed in the application. A malicious user *could* perform limited actions which were not intended.

RECOMMENDED ACTION ITEMS

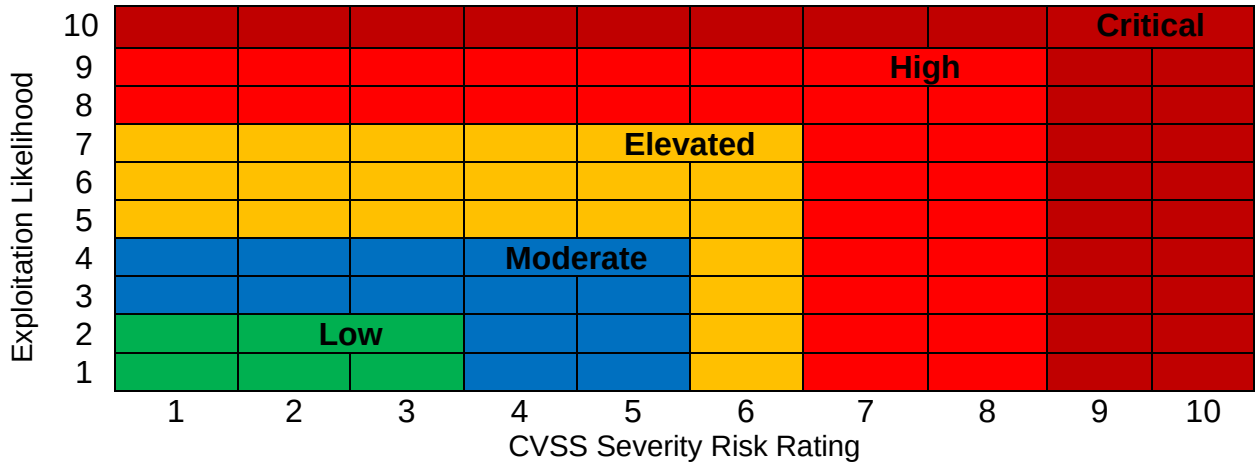
- **Validate User Input**: Best practices dictate that all user input should be verified both by the application prior to submission (client-side) and by the server (server-side) prior to further processing by the application. *See Findings 1 and 4 for more information.*
- **Ensure Consistent Role Enforcement:** Review and update code to ensure that role-based access controls (RBAC) are consistently enforced throughout the application. *See Finding 2 for more information.*
- **Require User Verification for Profile Updates:** The User Profile Update page should require the current user password prior to updating authentication related information (password, email, username). *See Finding 3 for more information.*
- **Add Support for Multi-Factor Authentication:** Since the Acme Ultimate application handled sensitive and controlled HIPAA information supporting, and potentially requiring, multi-factor authentication is recommended to significantly decrease the likelihood of account compromise and inadvertent patient information disclosure. *See Finding 8 for more information.*

RISK METHODOLOGY

Information security is not about eliminating risk. It is founded upon the science and discipline of risk management. This is an important distinction because computer systems are inherently designed to share information while security strives to guard it. Management's role, therefore, is to weigh the benefits of information sharing with the potential security risks of doing so, all while enabling the organization to achieve its objectives. Open Security has provided a recommended course action to accompany each vulnerability.

INFORMATION SECURITY RISK RATING SCALE

To effectively evaluate the security posture of a client's network, Open Security uses the Information Security Risk Rating Scale shown below. This scale is based on the open-industry Common Vulnerability Scoring System (CVSS) against the Common Vulnerabilities and Exposures (CVE) Dictionary maintained by the National Cybersecurity Federally Funded Research and Development Center (FFRDC) with funding from the National Cyber Security Division of the US Department of Homeland Security. This base CVSS score, the likelihood of exploitation, and the impact of exploitation are all taken into account to determine the overall risk presented by the vulnerability



RISK RATING KEY

When evaluating remediation timelines for your environment ***Critical*** network and system vulnerabilities should be addressed as quickly as feasible, the bulk of effort will likely involve those rated as ***High*** and ***Elevated.*** Open Security recommends that these risks be remediated within 30 days of report delivery. While it is of vital importance to identify solutions to all risks affecting the network, those rated ***Moderate*** and ***Low*** can be approached methodically, in line with general information security best practices without accepting significant risk of severe financial or data loss.

Critical risks: very high likelihood of exploitation and the possibility of ***catastrophic*** financial losses.
High risks: high likelihood of exploitation with the possibility of ***significant*** financial losses.
Elevated risks: average likelihood of exploitation with the possibility of ***material*** financial losses.
Moderate risks: below average likelihood of exploitation with the possibility of ***limited*** financial losses.
Low risks: below average likelihood of exploitation with ***little to no impact*** as a result.

ENGAGEMENT OVERVIEW

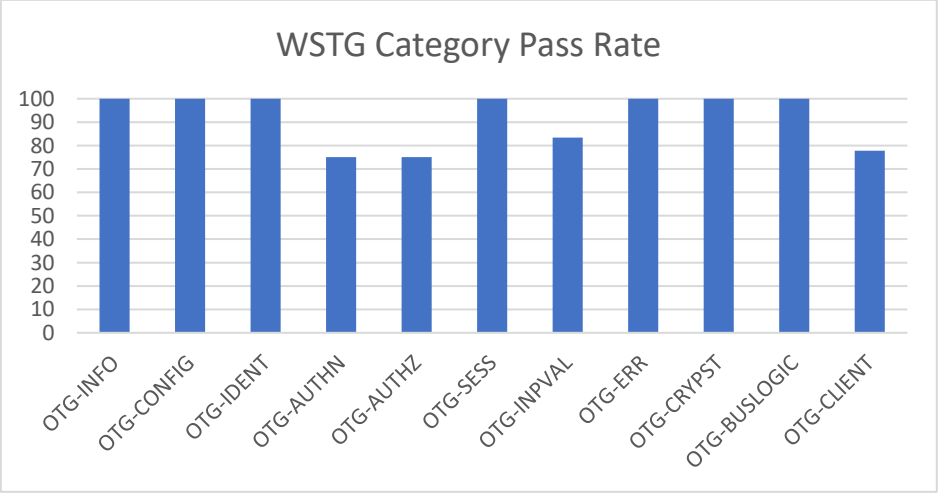
DESCRIPTION

Open Security performed a Web Application Penetration Test of the Acme Ultimate Web Application to identify security issues and recommend areas for improvement. Acme Ultimate is a Software-as-a-Service (SaaS) solution utilized by customer hospitals and clinics for workflows involving patient management, provider assessment, and product inventory operations. The application is multi-tenant, with different organizational accounts and users assigned within organizations.

The greatest business risk inherent in applications like this is the inadvertent exposure of customer records. The worst-case scenario is the exposure of customer records to unauthenticated attackers. A secondary concern, due to the multi-tenancy of the application, is the leakage of sensitive information across tenants. All discovered vulnerabilities were assessed to determine if they could be abused to lead to either of these outcomes.

When evaluating web applications, Open Security assesses not only based off our industry experience but also against recognized standards. One such standard that we use is the Web Security Testing Guide (WSTG) published by the Open Web Application Security Project (OWASP). By including the WSTG in our penetration testing methodology we ensure that our customers receive consistently thorough assessments evaluated against industry best practices.

The WSTG breaks analysis down across 11 categories. Each category is composed of several tests associated with some aspect of the underlying application. While these categories are high level, they can be used to quickly identify areas where the application has trends of problems. In our analysis of the Acme Ultimate web application Open Security found issues in four (4) of the eleven (11) categories:



Percentage Pass Rate for each WSTG Testing Category

The four (4) categories with identified issues were: *authentication*, *authorization*, *input validation*, and *client-side attacks*. Detailed information about the findings of the penetration test, to include how they map to the WSTG as well as recommendations are included in the *Key Findings* section of this report.

SCOPE AND RULES OF ENGAGEMENT

Penetration Testing occurred from 20-31 January 2020 and consisted of authenticated and unauthenticated access to the *staging* instance of the Acme Ultimate application which was hosted at ***pentest.acme-ultimate-dev.com***. Source code access was provided by Acme to enable more efficient testing.

Specific rules of engagement for the penetration test included the following:

- No timing restrictions on testing
- No storage/copying of sensitive data accessed during the engagement

The Acme Ultimate app was assessed both with and without additional security controls in place. An evaluation without security controls ensures that testing evaluates the web application and not the security devices protecting the application. A secondary benefit of this whitelisting is that it allows the tester to determine the skill level required by attackers to bypass current defenses which informs the analysis of the probability of exploitation.

SUMMARY OF HIGH-RISK FINDINGS

During the Penetration Test it was found that overall application security was well implemented: no **Critical** or **Elevated** risk findings were identified during the penetration test. The highest risk posed by any of the vulnerabilities was assessed to be **High**, and only one (1) vulnerability was assessed to pose this risk. For details on these findings, see the following *Key Findings* section.

KEY FINDINGS

During the Penetration Test, one (1) vulnerability was rated as high risk, two (2) vulnerabilities were rated as **Moderate** risk and three (3) vulnerabilities were rated as **Low** risk findings. An additional **Informational** finding is listed to highlight an area where Open Security recommends efforts are spent to greatly improve application security. Findings are listed once even if they pertain to multiple forms or URLs and vulnerabilities of common criteria are grouped together.

	Security Risk	*Account Required*	*Severity*
1	Stored CSV Injection Vulnerability	No	**High**
2	Role-based Access Control bypass via Direct Browsing	No	**Moderate**
3	Lack of Verification on Profile Update	Yes	**Moderate**
4	Password Policy – Weak Policy	No	**Low**
5	Password Policy – Remembered History Bypass	Yes	**Low**
6	Use of Out-of-Date Components	N/A	**Low**
7	Lack of Multi-Factor Access Controls	N/A	**Informational**

SECURITY ROADMAP

To strengthen overall defensive network capabilities, Open Security recommends upgrading end-of-life technologies and systems to their modern counterparts, such as eliminating use of outdated encryption techniques and upgrading vulnerable software, such as the *Acme Pro* application. Doing so will ensure that the latest improvements are always available and that the latest defensive mechanisms remain in place.

Efforts should also be made to configure SSL certificates and web servers to use trusted and valid certificates in all instances. This will ensure that users' communications are properly protected and meet compliance requirements. Additionally, server configuration improvements should be implemented to enforce HTTPS Strict Transport Security and prevent 'clickjacking' style attacks. These actions will mitigate or remediate 20% of identified issues.

SHORT TERM REMEDIATION

Acme should seek to incorporate all servers and applications into their scanning and patching routine, decommission any legacy servers, and update vendor supplied software where possible. In particular, the *Acme Pro* application hosted at samle.acme.com continues to account for 35% of identified issues. Upgrading to a newer version of this application will likely remediate a number of these findings.

Finding 1: Stored CSV Injection Vulnerability

VULNERABILITY RATING: ***High***

CVE/CWE: CWE-222

CVSS RATING: N/A

DISCOVERY METHOD: Uncredentialed Manual Review During Previous Penetration Test

OBSERVATION HISTORY: 2019Q3, 2020Q2

DESCRIPTION:

A vulnerability exists in the web application that allows an attacker to store an excel formula in a user name field, which can be downloaded by the user of the web application as a CSV file. When this CSV file is opened with Microsoft Excel, the command is executed on the user's computer. This vulnerability allows attackers to execute arbitrary commands on a user's computer.

AFFECTED ASSET(S):

2.2.2.242 (https://sample.acme-ultimate-dev.com)

ANALYSIS:

Open Security was able to weaponize CSV exports in combination with the Microsoft Dynamic Document Exchange (DDE) format. This vulnerability does not affect the underlying web application but instead attacks any system on which an exported CSV file is opened.

This vulnerability could be used by an attacker to execute attacks such as client-side command injection or code injection. The attack scenario targets users who download these files as part of the typical application workflow. Website developers should be aware that the information they export could be used as a vehicle to launch attacks against system users.

RECOMMENDATION:

Validate and filter input all user input so that only known-good values are accepted.

METHODOLOGY

The Web Application Penetration Test performed by Open Security began at 0700 MST on January 20th, 2020. Prior to testing, Acme and Open Security personnel coordinated to establish two testing accounts that would be used for the duration of the engagement.

Testers provided an IP address to whitelist in security appliances to allow unfettered access to the application to ensure maximum visibility into application vulnerabilities. Exploitable vulnerabilities were then tested with security controls in place to determine their current effectiveness.

When testing web applications, Open Security assesses not only based off our industry experience but also against recognized standards. One such standard that we use is the Web Security Testing Guide (WSTG) published by the Open Web Application Security Project (OWASP). By including the WSTG in our penetration testing methodology we ensure that our customers receive consistently thorough assessments evaluated against industry best practices.

OWASP – WEB SECURITY TESTING GUIDE

The WSTG breaks analysis down across 11 categories. Each category is composed of several tests associated with some aspect of the underlying application. These categories are high level concepts with more specific testing criteria called out for testers to evaluate against. As applications are all different, this testing guide is not meant to be a prescriptive step-by-step checklist on identifying issues but rather serves as a recommended framework of concepts and common issues to keep in mind while evaluating web applications.

Open Security and OWASP both recommend that security testing be an ongoing and continuous process. The WSTG recommendations apply to all aspects of the software development lifecycle (SDLC) and can be incorporated in every phase to ensure security is a conscious decision from product and feature inception through design, implementation, deployment and follow up maintenance.

At a high level the 11 categories of the WSTG cover:

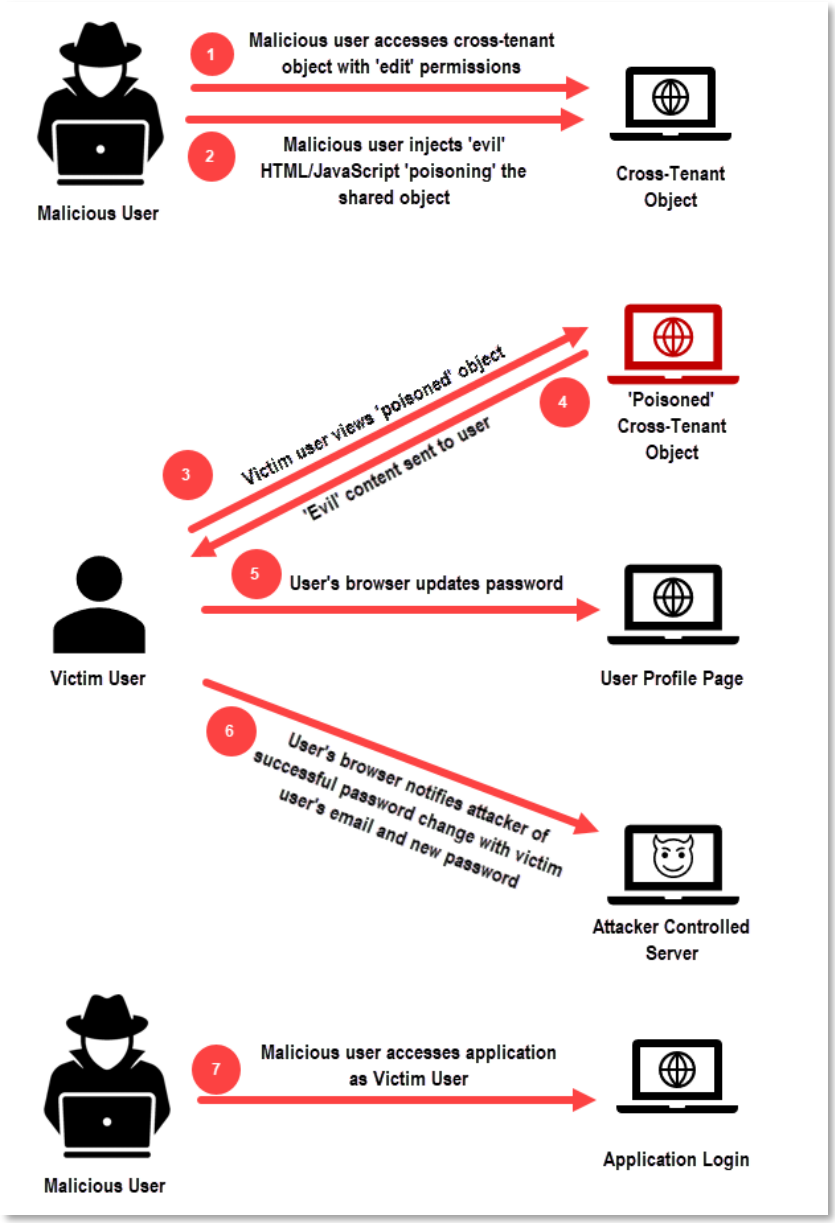
- **Information Gathering:** Gather information about the application to map attack surface and identify potential attack vectors in the application and supporting technologies.
- **Configuration and Deployment Management Testing:** Identify common configuration issues in the application itself, underlying technologies, and overall architecture.
- **Identity Management Testing:** Identify issues in user management processes of the application.
- **Authentication Testing:** Identify issues that could lead to account compromise.
- **Authorization Testing:** Identify issues that could lead to access of unauthorized information.
- **Session Management Testing:** Identify issues that could lead to session hijacking.
- **Input Validation Testing:** Identify issues in data handling and processing as it affects the application.
- **Error Handling:** Identify an information leakage as a result of errors and error handling.
- **Cryptography:** Validate that the application implements cryptography in accordance with industry standards and that sensitive information is properly protected.
- **Business Logic Testing:** Identify issues that result from unexpected application usage and invalid data.
- **Client-Side Testing:** Identify issues that allow for exploitation of clients accessing the application.

During the two-week testing window Open Security performed actions to thoroughly evaluate the Acme Ultimate application across all these areas. Issues affecting the application were identified in 4 of the 11 categories. The four affected categories were *Authentication, Authorization, Input Validation, and Client-Side attacks*. The vulnerabilities affecting these categories are called out in Findings 1-6. The *Account Takeover Attack* described in the next section shows how a malicious user could compromise other accounts via exploiting issues across these four categories.

ACCOUNT TAKEOVER ATTACK

The *account takeover attack* developed during testing of the application was the only method discovered by which patient information might be possibly accessed by an unauthorized party. It exploits a series of vulnerabilities to allow a malicious user to forcibly change the password of other valid accounts, which the attacker can then use to login as an affected user with the permissions granted to that user.

A high-level diagram of the attack is shown below:



Three phase 'account takeover' attack.
Attacker poisons a shared object, victim views it and has their password changed, attacker logs in as victim

The devised attack has three distinct phases:

1. A malicious user accesses a cross-tenant item with edit permissions due to issues in RBAC (**Finding 2**)
   Attacker injects malicious HTML/JavaScript into the shared object due to issues in input sanitization (**Findings 1 and 4**).
2. At some later point an unwitting 'victim' user browses to the 'poisoned' cross-tenant object
   The victim receives the malicious HTML/JavaScript and their web browser processes it due to application rendering malicious content as valid HTML (**Findings 1 and 4**).
   The victim's web browser executes the malicious JavaScript which uses the user profile update feature to submit a password change to a password of the attacker's choosing. This works because the password update feature does not require the user to enter their current password. (**Finding 3**)
   The victim's web browser notifies the attacker of the successful password change and provides the email for the account compromised.
3. The attacker logs in as the victim user. This could be prevented by implementation of MFA (**Finding 8**).

The remainder of the *Methodology* section will walk through the attacker's actions in more detail. It should be noted that Open Security validated that the WAF currently deployed ***will*** catch attempts by attackers to inject malicious content into the application. The penetration tester ***was able to bypass the WAF*** but this required ***significant effort*** and is not something that is easily done by unskilled attackers. Furthermore, this attack ***requires a malicious authenticated user*** which significantly decreases the likelihood of exploitation. Lastly, in the following sections the exploit shown and documented is the version caught by the WAF for security reasons.

POSTMORTEM

This attack requires the chaining together of both of the identified **moderate** vulnerabilities in the Acme Ultimate application. Implementing *any* of the recommendations from **Findings 2,3,4,** or **8** effectively disrupts the chain and prevents the attack from resulting in account compromise. Additionally, the attack requires a malicious user to not only perform the attack but to also perform it in such a way as to by existing security controls (the WAF). Lastly, the attack still relies on other 'victim' users browsing to 'poisoned' exercises to achieve execution in the context of other user accounts.

No other method of accessing sensitive information was identified. Review of code showed a good use of scoping sensitive information on a per account (tenant) basis. The only way to cross the tenant barrier is to compromise another account.

Existing brute-force counter measures mean that realistically the only attack paths viable for account compromise are the attack path referenced above and the common tactic of phishing targeted users and presenting them with fake login pages. As the later attack is not something that can be prevented by the Acme Ultimate team it is Open Security's opinion that fixing the vulnerabilities highlighted in this report will remove the only account compromise vector existing in the application as it was reviewed during this assessment (20-31 January 2020).

SECURITY TEAM SNAPSHOT

Passionate and forward-thinking, our team bring decades of combined technical experience as top-tier researchers, penetration testers, application security experts, and more. Drawing from experience in the US military and leading technology firms, we pride ourselves on the capabilities we make available to our clients. Open Security understands the importance of information security and appreciates the opportunity to have worked on this engagement.

GEOFFREY PAMERLEAU – SECURITY ENGINEER
GEOFF@OPENSECURITY.IO | 908.752.5183

Former Air Force officer and a Senior Engineer for Open Security. He performed computer network operations for the Intelligence Community, and it was here that he found his calling. Seeking the challenge that comes from private sector work he left active duty and joined our team. Geoff is passionate about security and loves to share his Military and Industry experience with our customers.

MICHAEL PLEASANT – CEO
MICHAEL@OPENSECURITY.IO | 210.446.7691

Michael is a former United States Marine Corps Intelligence Analyst and is the founder of Open Security. Following his uniformed service, Michael went on to develop curriculum and teach a wide range of topics such as military tactics and strategy, intelligence practices, sensitive site exploitation, and cybersecurity.

This report represents a “snapshot” of the security environment assessed at a specific point in time. Conditions may have improved, deteriorated, or remained the same since this assessment was completed. Open Security cannot guarantee the discovery of all system vulnerabilities, breaches, or attempted breaches. Should there be any questions regarding the contents of this report, please don’t hesitate to contact us.

APPENDIX A – GLOSSARY

Cascading Style Sheets (CSS)
A style sheet language used for describing the presentation of a document written in a markup language like HTML.

Cross-site Request Forgery (CSRF)
A type of malicious web exploit where an attack takes advantage of a trusted browser session to transmit unauthorized commands as the user.

Common Vulnerabilities & Exposures (CVE)
Provides a reference-method for publicly known information-security vulnerabilities and exposures. The National Cybersecurity FFRDC, operated by the Mitre Corporation, maintains the system, with funding from the US DHS.

Common Vulnerability Scoring System (CVSS)
A free and open industry standard for assessing the severity of computer system security vulnerabilities.

Common Weakness Enumeration (CWE)
A universal online dictionary of weaknesses that have been found in computer software. The dictionary is maintained by the MITRE Corporation and can be accessed free on a worldwide basis.

Domain Name System (DNS)
A hierarchical decentralized naming system for computers, services, or other resources connected to the Internet or a private network which translates human readable domain names to the numerical machine relevant IP addresses.

HTTP Strict Transport Security (HSTS)
A web security policy intended to protect against protocol downgrade attacks and cookie hijacking. It allows web servers to declare that web browsers should only interact with it using secure HTTPS connections, and never via the insecure HTTP protocol.

jQuery
A JavaScript library designed to simplify the client-side scripting of HTML. It is free, open-source software using the permissive MIT License. It is the most widely deployed JavaScript library by a large margin

National Institute of Standards and Technology (NIST)
Established in 1901, NIST provides technology, measurement, and standards as part of the US Department of Commerce.

Open Web Application Security Project (OWASP)
An online community dedicated to the security of web-based applications producing articles, methodologies, reports, and tools to further web-app security. Widely considered the industry standard for web-based application security.

Role Based Access Controls (RBAC)
A method of restricting access based on the roles of individual users within a system. RBAC lets employees have access rights only to the information they need to do their jobs and prevents them from accessing information that doesn't pertain to them.

Secure Sockets Layer (SSL)
A protocol in the Transport Layer Security family of protocols that provides communications security over a network.

Short Messaging Service (SMS)
A system that enables mobile phone users to send and receive text messages.

Software-as-a-Service (SaaS)
A method of software delivery and licensing in which software is accessed online via a subscription, rather than bought and installed on individual computers.

Web Application Firewall (WAF)
A firewall that monitors, filters or blocks data packets as they travel to and from a website or web application. A WAF can be either network-based, host-based or cloud-based and is often deployed through a reverse proxy and placed in front of one or more web sites.

Web Security Testing Guidelines (WSTG)
A comprehensive guide to testing the security of web applications and web services. Created by the collaborative efforts of security professionals and dedicated volunteers, the WSTG provides a framework of best practices used by penetration testers and organizations all over the world.

APPENDIX B – SCOPE

URLs: The following URLs were considered in-scope for this Penetration Test. All features and functionality contained therein were likewise considered in-scope.

pentest.acme-ultimate-dev.com

APPENDIX C – REFERENCES

1. **Open Web Application Security Project – Web Security Testing Guide**
    a. https://github.com/OWASP/wstg/tree/master/document
2. **NIST Password Recommendations**
    a. https://pages.nist.gov/800-63-3/sp800-63b.html
3. **Ruby-on-Rails Sample Password Gems**
    a. StrongPassword (https://github.com/bdmac/strong_password)
    b. PasswordBlacklist (https://github.com/gchan/password_blacklist)