VulnWebApp (VWA) Security Report

Code Revision: 1.0.0.0

Company: Acme Inc.

Report: VWA20231129

Author: [Hsin-Wen Chang]

Date: [2023/08/30]

[**VWA20230830 – Cryptographic Failures - HIGH**](#_30j0zll) **3**

[**VWA20230901 – Cryptographic Failures - HIGH**](#_royzagy73tzr)

[**VWA20230902 – Injection - HIGH**](#_ubm3z6g4we0n)

[**VWA20230903 – Injecction - MEDIUM**](#_6884g3k2b1of)

[**VWA20230904 – Security Misconfiguration - MEDIUM**](#_n4sc8m3ivdl2)

[**VWA20230905 – FALSEPOSITIVE - LOW**](#_jk8g4abjxnlx)

[**VWA20230906 – FALSEPOSITIVE - LOW**](#_sfe0dnad33m8)

# 

# **VWA20230830 – Cryptographic Failures - High**

**Vulnerability Exploited:** **Cryptographic Failures**

**Severity:[Critical, High, Medium, Low, Info]**

**System:** VWA Web Application

**Vulnerability Explanation**:

Using deprecated cryptographic functions MD5 in password.

**Vulnerability Walk-thru**:

**Recommendations:**

MD5 should never be used in passwords, we need to migrate to bcrypt or another hash that is slow hashing.

https://cheatsheetseries.owasp.org/cheatsheets/Password\_Storage\_Cheat\_Sheet.html

# 

# 

# 

# 

# 

# 

# **VWA20233008 – Cryptographic Failures - Medium**

**Vulnerability Exploited: Cryptographic Failures**

**Severity:[Critical, High, Medium, Low, Info]**

**System:** VWA Web Application

**Vulnerability Explanation**:

Hard-coded passwords frequently result in substantial authentication failures that are challenging for system administrators to identify. The administrator might be compelled to completely disable the product because once discovered, it can be challenging to fix. The back-end credentials could simply be hard-coded into the front-end product by the coder. The password might potentially be extracted by any user of that program. Given how easy it is to extract a password from a binary, client-side system with hard-coded passwords are considerably more dangerous.

**Vulnerability Walk-thru**:

**Recommendations:**

We should never use default passwords.

<https://bandit.readthedocs.io/en/1.7.0/plugins/b106_hardcoded_password_funcarg.html>

<https://cwe.mitre.org/data/definitions/259.html>

# 

# **VWA20230830 – Injection - High**

**Vulnerability Exploited:** **Injection**

**Severity:[Critical, High, Medium, Low, Info]**

**System:** VWA Web Application

**Vulnerability Explanation**:  
By using deprecated cryptographic functions MD5 in password will expose to an attacker's perspective, the optimal place to inject malicious content is in an area that is displayed to either many users or particularly interesting users. Interesting users typically have elevated privileges in the application or interact with sensitive data that is valuable to the attacker. The attacker can engage in a number of malicious actions after injecting the malicious script. Private data, including cookies that may contain session information, could be sent from the victim's computer to the attacker by the attacker. When the victim has administrator rights to control a website, the attacker may make malicious requests to that website on the victim's behalf, which might be extremely risky for that website. To compromise the victim's account on a trustworthy website, an attacker may use phishing attacks to imitate reputable websites and deceive the victim into entering their password. Finally, the script might take advantage of a flaw in the victim's web browser to potentially take control of their computer, a technique known as "drive-by hacking."

**Vulnerability Walk-thru**:

**Recommendations:** https://bandit.readthedocs.io/en/1.7.0/plugins/b703\_django\_mark\_safe.html

# 

# 

# 

# **VWA20233008 – Injection - Medium**

**Vulnerability Exploited:** **Injection**

**Severity:[Critical, High, Medium, Low, Info]**

**System:** VWA Web Application

**Vulnerability Explanation**:

This problem is low severity. Just a warning for those who might not be aware of the potential security concerns surrounding the library.

**Vulnerability Walk-thru**:

**Recommendations:**

By excluding the warning with # nosec B404 on the relevant line, or by altering scan behavior in your Bandit settings, you can disable the alert and have [B602: subprocess\_popen\_with\_shell\_equals\_true](https://bandit.readthedocs.io/en/latest/plugins/b602_subprocess_popen_with_shell_equals_true.html?highlight=b602) and [B603: subprocess\_without\_shell\_equals\_true](https://bandit.readthedocs.io/en/latest/plugins/b603_subprocess_without_shell_equals_true.html#b603-subprocess-without-shell-equals-true) both turned on, which are where actual security issues could happen.

<https://docs.openstack.org/bandit/1.4.0/blacklists/blacklist_imports.html#b404-import-subprocess>

# 

# 

# 

# 

# 

# **VWA20230904 – Security Misconfiguration - Medium**

**Vulnerability Exploited:** **Security Misconfiguration**

**Severity:[Critical, High, Medium, Low, Info]**

**System:** VWA Web Application

**Vulnerability Explanation**:

The product employs XML documents and permits the definition of their structure using a Document Type Definition (DTD), however it does not adequately regulate the quantity of recursive entity definitions.

**Vulnerability Walk-thru**:

**Recommendations:**

The majority of this is based on Christian Heimes' defusedxml work, which may be found at: https://pypi.python.org/pypi/defusedxml/#defusedxml-sax

It is well known that using different XLM methods to parse untrusted XML input leaves one open to XML assaults. The defusedxml equivalents of methods should be used in their place.

# 

# 

# 

# 

# 

# **VWA20233008 – FALSEPOSITIVE - Low**

**Vulnerability Exploited:** **FALSEPOSITIVE**

**Severity:[Critical, High, Medium, Low, Info]**

**System:** VWA Web Application

**Vulnerability Explanation**:

This is a false positive, the code is generate a random resetcode.

**Vulnerability Walk-thru**:

**Recommendations:** This is a false positive, the code is generate a random resetcode.

# 

# 

# 

# 

# 

# 

# 

# **VWA20233008 – FALSEPOSITIVE - Low**

**Vulnerability Exploited:** **FALSEPOSITIVE**

**Severity:[Critical, High, Medium, Low, Info]**

**System:** VWA Web Application

**Vulnerability Explanation**:

This plugin test is part of a family of tests built to check for process spawning and warn appropriately.

Specifically, this test looks for the spawning of a subprocess without the use of a command shell.

**Vulnerability Walk-thru**:

**Recommendations:**

This type of subprocess invocation is not vulnerable to shell injection attacks, but care should still be taken to ensure validity of input.

https://docs.openstack.org/bandit/1.4.0/plugins/subprocess\_without\_shell\_equals\_true.html