*Web Application Pen-Testing*

*AY 2022/2023*

*Week 13 Practical*

*OWASP Top 10 - 2021*

*A02:2021-* *Cryptographic Failures*

*A02 Related Challenges on OWASP Juice Shop*

#### OWASP Top 10 – 2021 – Image

Diagram

Description automatically generated

*Table of Contents*

[1. A02:2021- Cryptographic Failures 3](#_Toc124151713)

[*1.1.* *Description* 3](#_Toc124151714)

[*1.2.* *Example Attack Scenarios* 4](#_Toc124151715)

[2. Setup 5](#_Toc124151716)

[*2.1.* *Start and Login to Kali Linux VM with Host-only enabled* 5](#_Toc124151718)

[*2.2.* *[If not already done] Download, Extract, Power On, & Login: Ubuntu 64-bit-WAPT-VVM* 6](#_Toc124151719)

[*2.3.* *Run OWASP Juice Shop Docker Container* 7](#_Toc124151720)

[*2.4.* *Browse OWASP Juice Shop from Kali Linux VM* 8](#_Toc124151721)

[*2.5.* *[If not already done] Create Account & Login to JuiceBar on your HOST OS (Windows)* 8](#_Toc124151722)

[3. Prerequisite Challenges to be Done 10](#_Toc124151723)

[*3.1.* *Score Board: Find the carefully hidden 'Score Board' page* 10](#_Toc124151725)

[*3.2.* *Access Log: Gain access to any access log file of the server - Sensitive Data Exposure* 10](#_Toc124151726)

[*3.3.* *Forgotten Developer Backup: Access a developer's forgotten backup file - Sensitive Data Exposure* 11](#_Toc124151727)

[*3.4.* *Forgotten Sales Backup: Access a salesman's forgotten backup file - Sensitive Data Exposure* 11](#_Toc124151728)

[4. Attempt “A02:2021 - Cryptographic Failures” related challenges on OWASP Juice Shop 12](#_Toc124151729)

[*4.1.* *Download OWASP Juice Shop Challenge Walkthroughs from JuiceBar on your HOST OS (Windows)* 12](#_Toc124151734)

[5. Homework: Submit the 3 Red Circled completed challenges + 2 YouTube Challenges listed below as homework 13](#_Toc124151735)

[*5.1.* *Weird Crypto: Inform the shop about an algorithm or library it should definitely not use the way it does* 13](#_Toc124151737)

[*5.2.* *Nested Easter Egg: Apply some advanced cryptanalysis to find the real easter egg* 14](#_Toc124151738)

# A02:2021- Cryptographic Failures

A02:2021-Cryptographic Failures shifts up one position to #2, previously known as A3:2017-Sensitive Data Exposure, which was broad symptom rather than a root cause. The renewed name focuses on failures related to cryptography as it has been implicitly before. This category often leads to sensitive data exposure or system compromise. Notable Common Weakness Enumerations (CWEs) included are CWE-259: Use of Hard-coded Password, CWE-327: Broken or Risky Crypto Algorithm, and CWE-331 Insufficient Entropy.

## *Description*

The first thing is to determine the protection needs of data in transit and at rest. For example, passwords, credit card numbers, health records, personal information, and business secrets require extra protection, mainly if that data falls under privacy laws, e.g., EU's General Data Protection Regulation (GDPR), or regulations, e.g., financial data protection such as PCI Data Security Standard (PCI DSS). For all such data:

* Is any data transmitted in clear text? This concerns protocols such as HTTP, SMTP, FTP also using TLS upgrades like STARTTLS. External internet traffic is hazardous. Verify all internal traffic, e.g., between load balancers, web servers, or back-end systems.
* Are any old or weak cryptographic algorithms or protocols used either by default or in older code?
* Are default crypto keys in use, weak crypto keys generated or re-used, or is proper key management or rotation missing? Are crypto keys checked into source code repositories?
* Is encryption not enforced, e.g., are any HTTP headers (browser) security directives or headers missing?
* Is the received server certificate and the trust chain properly validated?
* Are initialization vectors ignored, reused, or not generated sufficiently secure for the cryptographic mode of operation? Is an insecure mode of operation such as ECB in use? Is encryption used when authenticated encryption is more appropriate?
* Are passwords being used as cryptographic keys in absence of a password base key derivation function?
* Is randomness used for cryptographic purposes that was not designed to meet cryptographic requirements? Even if the correct function is chosen, does it need to be seeded by the developer, and if not, has the developer over-written the strong seeding functionality built into it with a seed that lacks sufficient entropy/unpredictability?
* Are deprecated hash functions such as MD5 or SHA1 in use, or are non-cryptographic hash functions used when cryptographic hash functions are needed?
* Are deprecated cryptographic padding methods such as PKCS number 1 v1.5 in use?
* Are cryptographic error messages or side channel information exploitable, for example in the form of padding oracle attacks?

## *Example Attack Scenarios*

Scenario #1: An application encrypts credit card numbers in a database using automatic database encryption. However, this data is automatically decrypted when retrieved, allowing a SQL injection flaw to retrieve credit card numbers in clear text.

Scenario #2: A site doesn't use or enforce TLS for all pages or supports weak encryption. An attacker monitors network traffic (e.g., at an insecure wireless network), downgrades connections from HTTPS to HTTP, intercepts requests, and steals the user's session cookie. The attacker then replays this cookie and hijacks the user's (authenticated) session, accessing or modifying the user's private data. Instead of the above they could alter all transported data, e.g., the recipient of a money transfer.

Scenario #3: The password database uses unsalted or simple hashes to store everyone's passwords. A file upload flaw allows an attacker to retrieve the password database. All the unsalted hashes can be exposed with a rainbow table of pre-calculated hashes. Hashes generated by simple or fast hash functions may be cracked by GPUs, even if they were salted.

[Source: <https://owasp.org/Top10/A02_2021-Cryptographic_Failures/>]

# Setup



## *Start and Login to Kali Linux VM with Host-only enabled*

*Make sure the Virtual Machine Settings 🡪 Network Adapter 🡪 Host-only*

|  |  |
| --- | --- |
| *Graphical user interface, text  Description automatically generated* | ***Login*** *into this Kali Linux VM*  *Type in the KALI\_IP address below:*  *XXX.XXX.XX.XXX*  *Graphical user interface, application  Description automatically generated* |

|  |  |
| --- | --- |
| *Tools with solid fill* | *In case your Kali Linux is* ***not responding*** *to changing to NAT (i.e., still not connected to the Internet). You can restart Kali Linux’s Ethernet Interface (eth0) by typing the following* ***2 commands one after the other*** *into the Kali Linux’s Terminal Emulator and press Enter:* |

*sudo ifdown eth0*

*Text

Description automatically generated*

*sudo ifup eth0*

*Text

Description automatically generated*

## *[If not already done] Download, Extract, Power On, & Login:* *Ubuntu 64-bit-WAPT-VVM*

If not already done: Please download this VM (preferably using your home WiFi). This is a 6.36 GB, 7z compressed VMware Image File consisting of Vulnerable Applications: Juice Shop, Web Goat, Mutillidae II, crAPI, and vAPI. In case of limited storage space you can download this VM into your portable external storage and run the VM from portable external storage.

<https://drive.google.com/file/d/1EEhVRFbfRQFZxhGMu-HcjxHpzmee5ope/view?usp=sharing>

1. After download, extract Ubuntu 64-bit-WAPT-VVM.7z
2. Inside the Ubuntu 64-bit-WAPT-VVM folder, double click on Ubuntu 64-bit-WAPT-VVM.vmx
3. The VM should have opened in your VMWare Workstation, click “Power on this virtual machine”

Graphical user interface, application

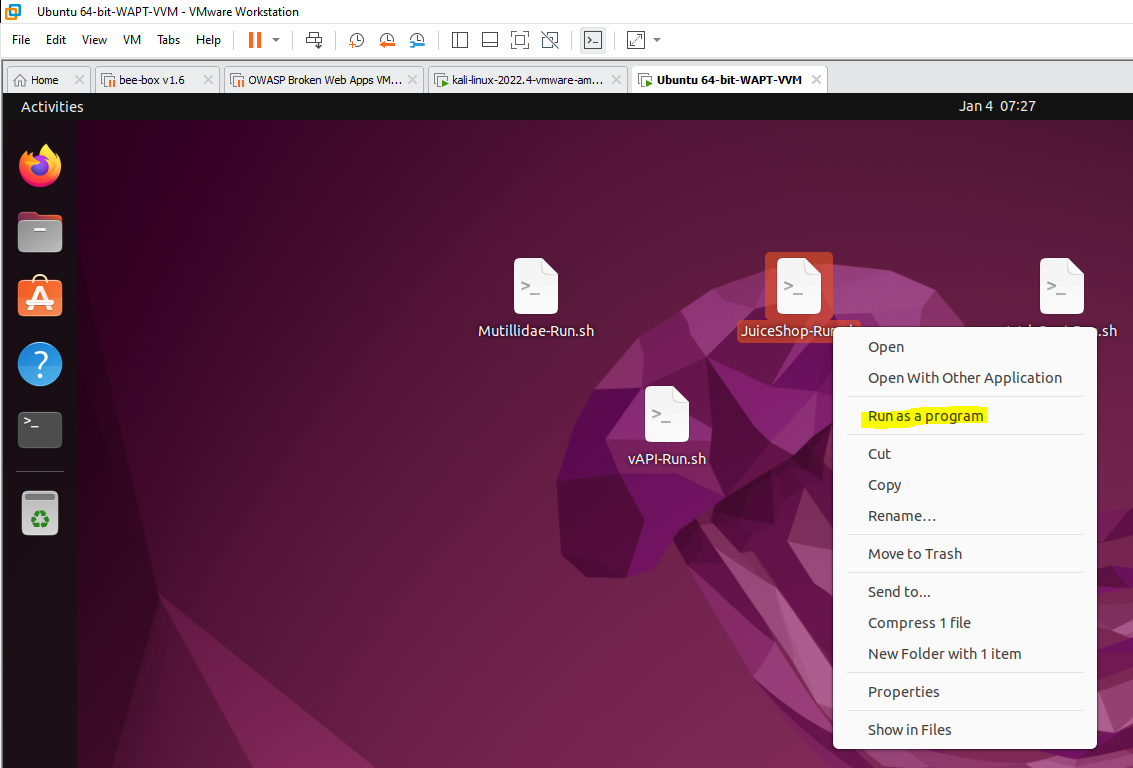
Description automatically generated

1. After the VM bootup, click on “wapt user” to enter the Password: wapt@123 and press Enter

|  |  |
| --- | --- |
| Graphical user interface, application, Teams  Description automatically generated |  |

## *Run OWASP Juice Shop Docker Container*

On the desktop right click on JuiceShop-Run.sh and select “Run as a program”



Make sure that OWASP Juice Shop Docker Container is running. Click the “Terminal” icon from the left-hand side menu. Type the following command into the Terminal and press Enter:

docker container ps

You should notice that the bkimminich/juice-shop status is Up and it is receiving requests on port 3000

|  |  |
| --- | --- |
| A screenshot of a cell phone  Description automatically generated with medium confidence |  |

Obtain the IP address of Ubuntu 64-bit-WAPT-VVM. Type the following command into the Terminal and press Enter:

ip -4 addr

In the output you should be able to see the IP address of Ubuntu 64-bit-WAPT-VVM. Type in the IP address of Ubuntu 64-bit-WAPT-VVM\_IP here: XXX.XXX.XXX.XXX

Text

Description automatically generated

## *Browse OWASP Juice Shop from Kali Linux VM*

Type the following into the address bar of the Kali Linux Web Browser and hit enter:

http:// Ubuntu 64-bit-WAPT-VVM\_IP:3000

Please give it some time to load the webpage

Graphical user interface

Description automatically generated

|  |  |
| --- | --- |
| *Tools with solid fill* | *In case you are not able to load the Juice Shop website, restart* Ubuntu 64-bit-WAPT-VVM and redo the steps in 2.3. |

## *[If not already done] Create Account & Login to JuiceBar on your HOST OS (Windows)*

JuiceBar was developed by Diploma in CSF Year 3 students (Yi Jing, Xin Min, Melvin, and Brayden) as part of their capstone project under the supervision of Mr. Tan Hock Guan. The aim of this capstone project is to use the OWASP Juice Shop to research and develop a set of reliable documentation on the different types of challenges found on the web application. The purpose of the documentation is to share with people who are interested in learning more about different security tools, and in the future, it can also be used in security training.

This website consolidates different challenges categorized into OWASP 2017 top 10 vulnerabilities, OWASP 2021 top 10 vulnerabilities and challenges with different difficulties based on the number of stars they have.

Graphical user interface, text, application, email

Description automatically generated

Type the following into the address bar of your HOST OS (Windows 10/11) Web Browser and hit enter:

<http://34.142.193.111/>

Click Register.

Graphical user interface, text, application

Description automatically generated

Proceed to create an account, use your personal Email address. Confirm by clicking on the link sent to your email inbox with the Subject Title: Confirm your email for JuiceBar.

Graphical user interface, application

Description automatically generated

Proceed to Login to JuiceBar.

Graphical user interface, text, application, chat or text message

Description automatically generated

# Prerequisite Challenges to be Done



## *Score Board: Find the carefully hidden 'Score Board' page*

[Source: <https://youtu.be/hPXfLc7U_PM>]

[](https://www.youtube.com/embed/hPXfLc7U_PM?feature=oembed)

## *Access Log: Gain access to any access log file of the server - Sensitive Data Exposure*

[Source: <https://youtu.be/RBTfGk-ZwnY>]

[](https://www.youtube.com/embed/RBTfGk-ZwnY?feature=oembed)

## *Forgotten Developer Backup: Access a developer's forgotten backup file - Sensitive Data Exposure*

[Source: <https://youtu.be/YvkuVZ6r2Rg>]

[](https://www.youtube.com/embed/YvkuVZ6r2Rg?feature=oembed)

## *Forgotten Sales Backup: Access a salesman's forgotten backup file - Sensitive Data Exposure*

[Source: <https://youtu.be/5g4WRASni6g>]

[](https://www.youtube.com/embed/5g4WRASni6g?feature=oembed)

# Attempt “A02:2021 - Cryptographic Failures” related challenges on OWASP Juice Shop



## *Download OWASP Juice Shop Challenge Walkthroughs from JuiceBar on your HOST OS (Windows)*

Click on “OWASP2021” on the top menu bar and select “A02 Cryptographic Failures” as shown below:

Graphical user interface, text, application, email

Description automatically generated

Select a challenge and once inside the challenge scroll down and you will notice “Steps” which could be used as hints to solve this challenge, or you can “Download Document” containing the step by step walkthrough to solve this challenge.

Graphical user interface, text, application

Description automatically generated

Either based on the “Steps” or the “Downloaded Document” you can now use your Kali Linux and OWASP Juice Shop that is running at http:// Ubuntu 64-bit-WAPT-VVM\_IP:3000 to attempt these challenges.

# Homework: Submit the 3 Red Circled completed challenges + 2 YouTube Challenges listed below as homework

Graphical user interface, text, application, email

Description automatically generated



## *Weird Crypto: Inform the shop about an algorithm or library it should definitely not use the way it does*

[Source: <https://youtu.be/GWJouiMUJno>]

[](https://www.youtube.com/embed/GWJouiMUJno?feature=oembed)

## *Nested Easter Egg: Apply some advanced cryptanalysis to find the real easter egg*

[Source: <https://youtu.be/yvatrnWvcGE>]

[](https://www.youtube.com/embed/yvatrnWvcGE?feature=oembed)