*Web Application Pen-Testing*

*AY 2022/2023*

*Week 12 Practical*

*OWASP Top 10 - 2021*

*A06:2021-Vulnerable and Outdated Components*

*Heartbleed Vulnerability*

*A06 Related Challenges on OWASP Juice Shop*

#### OWASP Top 10 – 2021 – Image

Diagram

Description automatically generated

*Table of Contents*

[1. A06:2021-Vulnerable and Outdated Components 3](#_Toc123713738)

[*1.1.* *Description* 3](#_Toc123713739)

[*1.2.* *Example Attack Scenarios* 3](#_Toc123713740)

[2. Setup 4](#_Toc123713741)

[*2.1.* *Start and Login to Kali Linux VM with Host-only enabled* 4](#_Toc123713743)

[*2.2.* *Start bee-box (bWAPP) VM* 5](#_Toc123713744)

[3. Heartbleed Vulnerability 6](#_Toc123713745)

[*3.1.* *Browse bWAPP website from Kali LInux* 7](#_Toc123713747)

[*3.2.* *Download attack script* 8](#_Toc123713748)

[*3.3.* *Scanning: Probe open ports to determine service/version info using nmap tool* 9](#_Toc123713749)

[*3.4.* *Discovery: Heartbleed Vulnerability Scan using nmap tool* 10](#_Toc123713750)

[*3.5.* *Discover: Heartbleed Vulnerability using sslyze tool* 11](#_Toc123713751)

[*3.6.* *Visit the Heartbleed Vulnerable Webserver* 12](#_Toc123713752)

[*3.7.* *Exploitation and Post Exploitation using the Attack Script* 13](#_Toc123713753)

[4. Attempt “A06:2021-Vulnerable and Outdated Components” related challenges on OWASP Juice Shop 14](#_Toc123713754)

[*4.1.* *Download, Extract, Power On, & Login: Ubuntu 64-bit-WAPT-VVM* 14](#_Toc123713756)

[*4.2.* *Run OWASP Juice Shop Docker Container* 15](#_Toc123713757)

[*4.3.* *Browse OWASP Juice Shop from Kali Linux VM* 16](#_Toc123713758)

[*4.4.* *Create Account & Login to JuiceBar on your HOST OS (Windows 10/11)* 16](#_Toc123713759)

[*4.5.* *Download OWASP Juice Shop Challenge Walkthroughs from JuiceBar on your HOST OS (Windows 10/11)* 18](#_Toc123713760)

[5. Homework: Attempt the 6 challenges under “A06:2021-Vulnerable and Outdated Components” and submit ANY 3 completed challenges as homework 19](#_Toc123713761)

# A06:2021-Vulnerable and Outdated Components

A06:2021-Vulnerable and Outdated Components was previously titled Using Components with Known Vulnerabilities and is #2 in the Top 10 community survey, but also had enough data to make the Top 10 via data analysis. This category moves up from #9 in 2017 and is a known issue that we struggle to test and assess risk.

## *Description*

You are likely vulnerable::

* If you do not know the versions of all components you use (both client-side and server-side). This includes components you directly use as well as nested dependencies.
* If the software is vulnerable, unsupported, or out of date. This includes the OS, web/application server, database management system (DBMS), applications, APIs and all components, runtime environments, and libraries.
* If you do not scan for vulnerabilities regularly and subscribe to security bulletins related to the components you use.
* If you do not fix or upgrade the underlying platform, frameworks, and dependencies in a risk-based, timely fashion. This commonly happens in environments when patching is a monthly or quarterly task under change control, leaving organizations open to days or months of unnecessary exposure to fixed vulnerabilities.
* If software developers do not test the compatibility of updated, upgraded, or patched libraries.
* If you do not secure the components’ configurations (see A05:2021-Security Misconfiguration).

## *Example Attack Scenarios*

Scenario #1: Components typically run with the same privileges as the application itself, so flaws in any component can result in serious impact. Such flaws can be accidental (e.g., coding error) or intentional (e.g., a backdoor in a component). Some example exploitable component vulnerabilities discovered are:

* CVE-2017-5638, a Struts 2 remote code execution vulnerability that enables the execution of arbitrary code on the server, has been blamed for significant breaches.
* While the internet of things (IoT) is frequently difficult or impossible to patch, the importance of patching them can be great (e.g., biomedical devices).

There are automated tools to help attackers find unpatched or misconfigured systems. For example, the Shodan IoT search engine can help you find devices that still suffer from Heartbleed vulnerability patched in April 2014.

[Source: <https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/>]

# 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*

## *Start bee-box (bWAPP) VM*

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

Graphical user interface, text, application

Description automatically generated

1. Type in the bee-box VM IP Address below: bWAPP\_IP
2. XXX.XXX.XX.XXX

Text

Description automatically generated

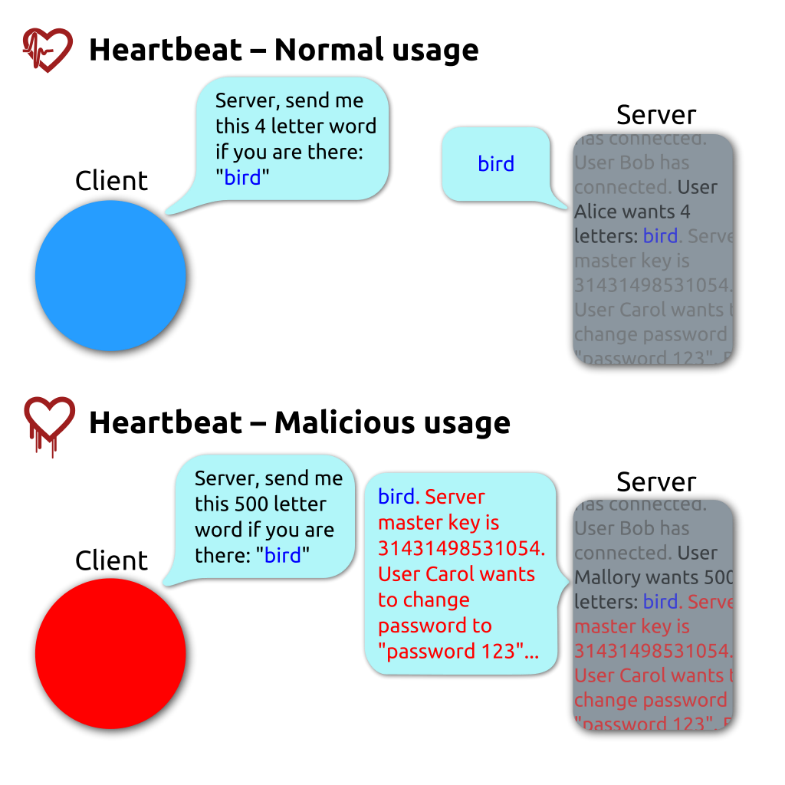
# Icon Description automatically generatedHeartbleed Vulnerability

Heartbleed was a security bug in the OpenSSL cryptography library, which is a widely used implementation of the Transport Layer Security (TLS) protocol. It was introduced into the software in 2012 and publicly disclosed in April 2014. It resulted from improper input validation (due to a missing bounds check) in the implementation of the TLS heartbeat extension. Thus, the bug's name derived from heartbeat. The vulnerability was classified as a buffer over-read, a situation where more data can be read than should be allowed.

Heartbleed was registered in the Common Vulnerabilities and Exposures database as CVE-2014-0160. A fixed version of OpenSSL was released on 7 April 2014, on the same day Heartbleed was publicly disclosed.

The RFC 6520 Heartbeat Extension tests TLS/DTLS secure communication links by allowing a computer at one end of a connection to send a Heartbeat Request message, consisting of a payload, typically a text string, along with the payload's length as a 16-bit integer. The receiving computer then must send exactly the same payload back to the sender.

Heartbleed is therefore exploited by sending a malformed heartbeat request with a small payload and large length field to the vulnerable party (usually a server) in order to elicit the victim's response, permitting attackers to read up to 64 kilobytes of the victim's memory that was likely to have been used previously by OpenSSL.[66] Where a Heartbeat Request might ask a party to "send back the four-letter word 'bird'", resulting in a response of "bird", a "Heartbleed Request" (a malicious heartbeat request) of "send back the 500-letter word 'bird'" would cause the victim to return "bird" followed by whatever 496 subsequent characters the victim happened to have in active memory. Attackers in this way could receive sensitive data, compromising the confidentiality of the victim's communications. Although an attacker has some control over the disclosed memory block's size, it has no control over its location, and therefore cannot choose what content is revealed.[citation needed]



[Source: <https://en.wikipedia.org/wiki/Heartbleed>]



## *Browse bWAPP website from Kali LInux*

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

http://bWAPP\_IP

We see a list of links. In this practical exercise we will **focus on “bWAPP” website**. Click on “bWAPP”.

Graphical user interface, text, website

Description automatically generated

Login: bee & Password: bug 🡪 Click “Login”

A screenshot of a computer

Description automatically generated with medium confidence

Under “Choose your bug”: Select “Heartbleed Vulnerability” 🡪 Click the button “Hack”



## *Download attack script*

Text, letter

Description automatically generated

Graphical user interface, website

Description automatically generated

## *Scanning: Probe open ports to determine service/version info using nmap tool*

Type the following command into the Kali Linux’s Terminal Emulator and press Enter:

sudo nmap -h

Text

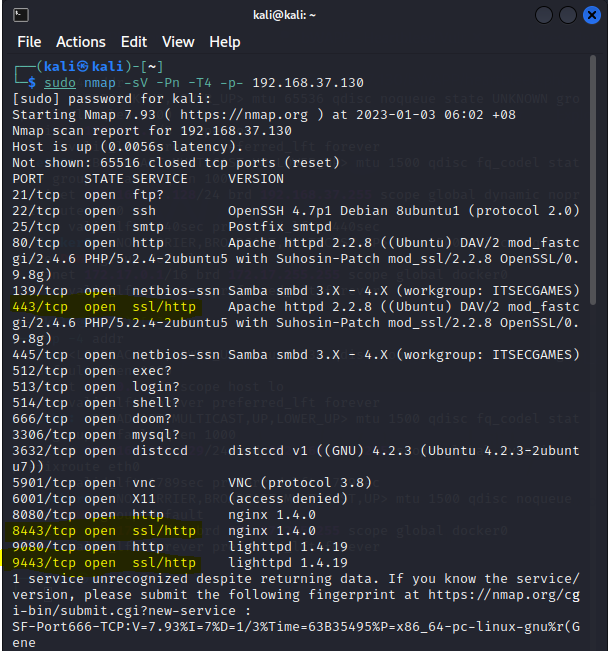
Description automatically generated

Type the following command into the Kali Linux’s Terminal Emulator and press Enter:

sudo nmap **-sV** -Pn -T4 -p- bWAPP\_IP

* -sV: Probe open ports to determine service/version info
* -Pn: Treat all hosts as online -- skip host discovery
* -T<0-5>: Set timing template (higher is faster)
* -p <port ranges>: Only scan specified ports
* bWAPP\_IP: bWAPP VM IP Address

In the output below we can identify 3 ports 443, 8443, and 9443 are open and are running the service ssl/http. We will check whether these services are vulnerable to Heartbleed bug.



## *Discovery: Heartbleed Vulnerability Scan using nmap tool*

Type the following command into the Kali Linux’s Terminal Emulator and press Enter:

sudo nmap --script ssl-heartbleed -sV -p 443,8443,9443 bWAPP\_IP

* --script: Defines which script to run
* ssl-heartbleed: is a Nmap Script Engine (NSE) script that detects whether a server is vulnerable to the OpenSSL Heartbleed bug (CVE-2014-0160). The code is based on the Python script ssltest.py authored by Katie Stafford (katie@ktpanda.org)
* -p <port ranges>: Only scan specified ports
* bWAPP\_IP: bWAPP VM IP Address

In the output below we can discover that among the 3 ports, port 8443 is vulnerable to Heartbleed bug.

Text

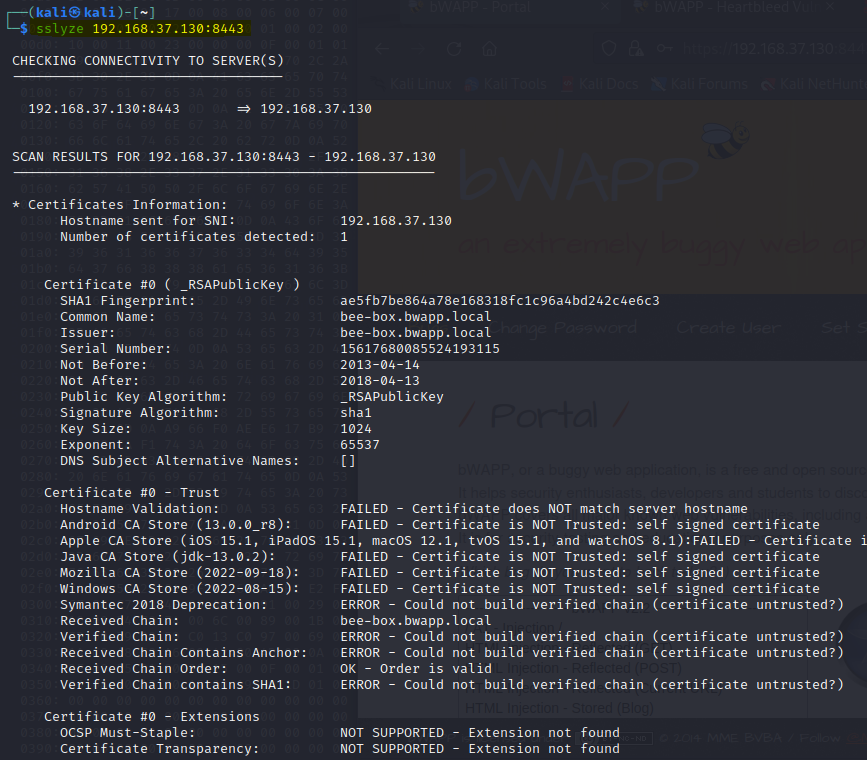
Description automatically generated

## *Discover: Heartbleed Vulnerability using sslyze tool*

SSLyze is a fast and powerful SSL/TLS scanning tool and Python library. SSLyze can analyze the SSL/TLS configuration of a server by connecting to it, in order to ensure that it uses strong encryption settings (certificate, cipher suites, elliptic curves, etc.), and that it is not vulnerable to known TLS attacks (Heartbleed, ROBOT, OpenSSL CCS injection, etc.). [Source: <https://github.com/nabla-c0d3/sslyze>]

Type the following command into the Kali Linux’s Terminal Emulator and press Enter:

sslyze bWAPP\_IP:8443

Text

Description automatically generated

## *Visit the Heartbleed Vulnerable Webserver*

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

http://bWAPP\_IP:8443

Login with the given credentials: bee/bug

Graphical user interface, website

Description automatically generated

## *Exploitation and Post Exploitation using the Attack Script*

Type the following commands into the Kali Linux’s Terminal Emulator one after the other and press Enter:

cd Downloads

python2 heartbleed.py -p 8443 bWAPP\_IP

You will notice in the output that the attack script mounted the Heartbleed attack and dumped the vulnerable server’s memory, where we can clearly see the (Login=bee&password=bug) credentials that we entered in the previous step.

Text

Description automatically generated

# Attempt “A06:2021-Vulnerable and Outdated Components” related challenges on OWASP Juice Shop

OWASP Juice Shop is probably the most modern and sophisticated insecure web application! It can be used in security trainings, awareness demos, CTFs and as a guinea pig for security tools! Juice Shop encompasses vulnerabilities from the entire OWASP Top Ten along with many other security flaws found in real-world applications! [Source: <https://owasp.org/www-project-juice-shop/>]



## *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

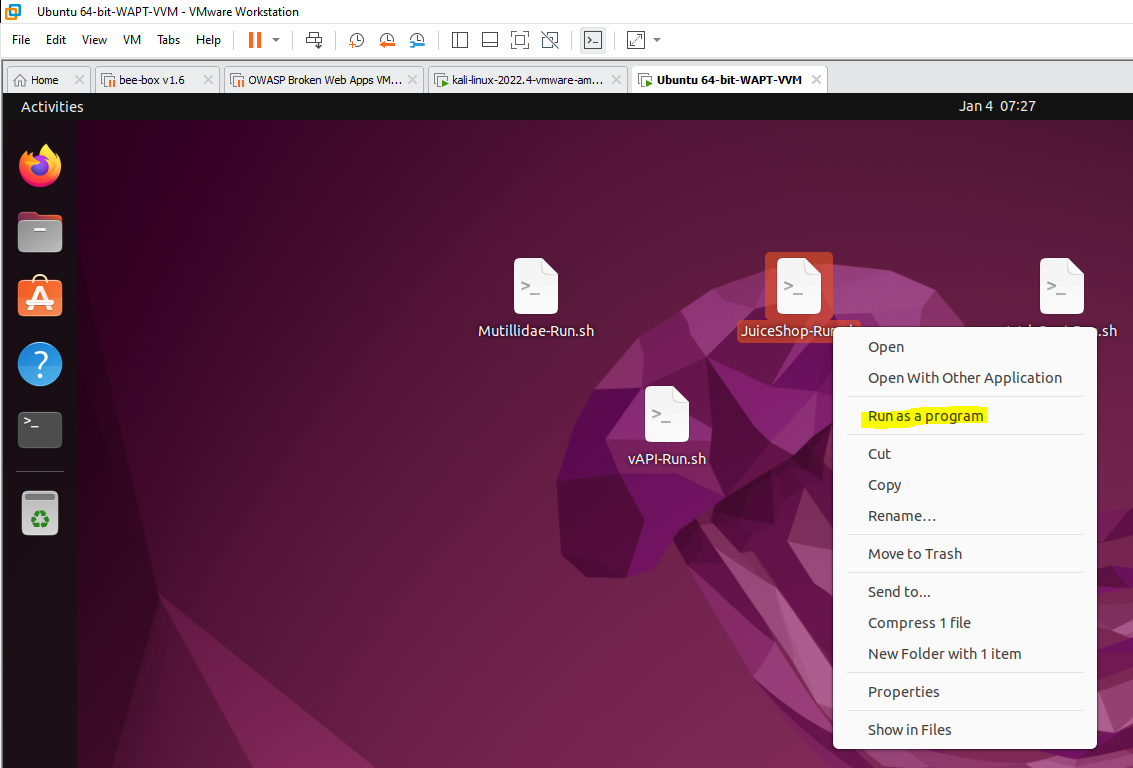
Description automatically generated

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

## *Create Account & Login to JuiceBar on your HOST OS (Windows 10/11)*

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

## *Download OWASP Juice Shop Challenge Walkthroughs from JuiceBar on your HOST OS (Windows 10/11)*

Click on “OWASP2021” on the top menu bar and select “A06 Vulnerable and Outdated Components” as shown below:

Graphical user interface, text, application, email

Description automatically generated

Select the challenge “Forged Signed JWT”

Graphical user interface, text, application

Description automatically generated

Once inside the challenge “Forged Signed JWT” 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: Attempt the 6 challenges under “A06:2021-Vulnerable and Outdated Components” and submit ANY 3 completed challenges as homework

Graphical user interface, text, application

Description automatically generated