SECURITY ASSESSMENT

Udajuicer

Submitted to: **Development Department, Udajuicer**
Security Analyst: Michael Wolff

Date of Testing: 14.04.2021
Date of Report Delivery: 19.04.2021

Table of Contents

Security Engagement Summary

Engagement Overview

The Development Team requested a vulnerability assessment of a legacy web-application.
The (Udajuicer) application was under attack, but the issue has been mitigated, the system was recovered and secured.
The goal of the engagement is to identify any potential areas of concern associated with the web application in its current state, provide solutions for reducing risks and fix vulnerabilities.
The engagement will be completed by the Information Security Department. All testing activities were performed on the staging environment provided by the customer and completely isolated from the production data.

Scope

Vulnerability assessment can identify potential problems and weaknesses in an environment.

The web-application is highly accessible from the internet and hosts the company’s ecommerce solution. The website is used to handle customer requests.

This report summarizes what the Information Security Department believes are the most important issues to address in the application. The chart below outlines a number of issues identified, that are grouped by risk factors. Note the risk ratings were given to help assist in prioritizing remediation efforts.

Risk Level	Description
High	These issues identity conditions that could directly result in the compromise or unauthorized access of a network, system, application or sensitive Information. Examples of High-Risk issues include remote execution of commands, known buffer overflows; unauthorized access and disclosure of sensitive information.
Medium	The issues identify conditions that do not immediately or directly result in the compromise or unauthorized access of a network, system, application or sensible information, but do provide a capability or information that could, in combination with other capabilities or information, result in the compromise or unauthorized access of a network, application or information. Examples of Medium-Risk issues include directory browsing, partial access to files on the system, disclosure of security mechanisms and unauthorized use of services.

Low	These issues identify conditions that do not immediately or directly result in the compromise or unauthorized access of a network, system, application or sensitive information, but do provide information that could be used in combination with other information to gain insights into how to compromise or gain unauthorized access.to a network, system, application or information.
Informational	These issues, also known as information leakage, appear when a website unintentionally reveals sensitive information to its users.

Identified issues by risk factor:

Risk Level	Number of Alerts
High	0
Medium	1
Low	1
Informational	2

Executive Risk Analysis

The website shows medium and low risk vulnerabilities. The web-application shows misconfiguration which could be used by an attacker to access data that is available in an unauthenticated manner.

Name	Risk Level	Number of Instances
Cross-Domain Misconfiguration	Medium	30
Cross-Domain JavaScript Source File Inclusion	Low	6
Information Disclosure - Suspicious Comments	Informational	6
Timestamp Disclosure - Unix	Informational	23

Exploration of these flaws is inevitable. An attack can have a severe impact on the business. The Information Security Department strongly recommends to remediate all issues detected to mitigate against the possible risk of a sensitive data compromise.

Executive Recommendation

- Issue a maintenance window to perform the necessary fixes.
- Send a message informing customer of the downtime.
- Create backups to restore the system in case of failure.
- Conduct follow-up scanning

Significant Vulnerability Summary

During the course of this assessment, the Information Security Department did not identify any critical vulnerabilities that could lead to full compromise of the system. However, several medium and low severity issues were found, which should be addressed promptly.

🟧 Medium Risk Vulnerability: **10098 - Cross-Domain Misconfiguration** *(Page 6/7)*

🟨 Low Risk Vulnerability: **10017 - Cross-Domain JavaScript Source File Inclusion** *(Page 8)*

🟦 Information Risk Vulnerability: **10027 - Information Disclosure - Suspicious Comments** *(Page 9)*

🟦 Information Risk Vulnerability: **10096 - Timestamp Disclosure** *(Page 9)*

🟧 AC** **CVE-2020-14145***	4.3	https://vulners.com/cve/CVE-2020-14145	*(Page 9)*
🟥 AC** **CVE-2021-28041***	4.6	https://vulners.com/cve/CVE-2021-28041	*(Page 9)*
🟨 AC** **CVE-2020-12062***	5.0	https://vulners.com/cve/CVE-2020-12062	*(Page 10)*
🟧 AC** **CVE-2020-15778***	6.8	https://vulners.com/cve/CVE-2020-15778	*(Page 10)*

*CVE stands for Common Vulnerabilities and Exposures. The Common Vulnerabilities and Exposures (CVE) system 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 National Cyber Security Division of the United States Department of Homeland Security.

**AC = Access Complexity

Significant Vulnerability Detail

Medium Risk Vulnerability: **10098 - Cross-Domain Misconfiguration**

Description: Web browser data loading may be possible, due to a Cross Origin Resource Sharing (CORS) misconfiguration on the web server

Information: The CORS misconfiguration on the web server permits cross-domain read requests from arbitrary third party domains, using unauthenticated APIs on this domain. Web browser implementations do not permit arbitrary third parties to read the response from authenticated APIs, however. This reduces the risk somewhat. This misconfiguration could be used by an attacker to access data that is available in an unauthenticated manner, but which uses some other form of security, such as IP address white-listing.

Solution: Ensure that sensitive data is not available in an unauthenticated manner (using IP address white-listing, for instance).
Configure the "Access-Control-Allow-Origin" HTTP header to a more restrictive set of domains, or remove all CORS headers entirely, to allow the web browser to enforce the Same Origin Policy (SOP) in a more restrictive manner.

Reference:
http://www.hpenterprisesecurity.com/vulncat/en/vulncat/vb/html5_overly_permissive_cors_policy.html

Code:

```
<!--
  ~ Copyright (c) 2014-2020 Bjoern Kimminich.
  ~ SPDX-License-Identifier: MIT
  -->

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>OWASP Juice Shop</title>
  <meta name="description" content="Probably the most modern and sophisticated insecure web
application">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link id="favicon" rel="icon" type="image/x-icon" href="assets/public/favicon_js.ico">
  <link rel="stylesheet" type="text/css"
href="//cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.css" />
  <script src="//cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.js"></script>
  <script src="//cdnjs.cloudflare.com/ajax/libs/jquery/2.2.4/jquery.min.js"></script>
  <script>
    window.addEventListener("load", function(){
      window.cookieconsent.initialise({
        "palette": {
          "popup": { "background": "#546e7a", "text": "#ffffff" },
          "button": { "background": "#558b2f", "text": "#ffffff" }
        },
        "theme": "classic",
        "position": "bottom-right",
        "content": { "message": "This website uses fruit cookies to ensure you get the juiciest tracking
experience.", "dismiss": "Me want it!", "link": "But me wait!", "href":
"https://www.youtube.com/watch?v=9PnbKL3wuH4" }
      })});
  </script>
<link rel="stylesheet" href="styles.css"></head>
<body class="mat-app-background bluegrey-lightgreen-theme">
  <app-root></app-root>
<script src="runtime-es2015.js" type="module"></script><script src="runtime-es5.js" nomodule
defer></script><script src="polyfills-es5.js" nomodule defer></script><script src="polyfills-
es2015.js" type="module"></script><script src="vendor-es2015.js" type="module"></script><script
src="vendor-es5.js" nomodule defer></script><script src="main-es2015.js"
type="module"></script><script src="main-es5.js" nomodule defer></script></body>
</html>
```

Low Risk Vulnerability: **Passive 10017 - Cross-Domain JavaScript Source File Inclusion**

Description: The page includes one or more script files from a third-party domain.

Solution: Ensure JavaScript source files are loaded from only trusted sources, and the sources can't be controlled by end users of the application.

```
<!--
  ~ Copyright (c) 2014-2020 Bjoern Kimminich.
  ~ SPDX-License-Identifier: MIT
  -->

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>OWASP Juice Shop</title>
  <meta name="description" content="Probably the most modern and sophisticated insecure web
application">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link id="favicon" rel="icon" type="image/x-icon" href="assets/public/favicon_js.ico">
  <link rel="stylesheet" type="text/css"
href="//cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.css" />
  <script src="//cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.js"></script>
  <script src="//cdnjs.cloudflare.com/ajax/libs/jquery/2.2.4/jquery.min.js"></script>
  <script>
    window.addEventListener("load", function(){
      window.cookieconsent.initialise({
        "palette": {
          "popup": { "background": "#546e7a", "text": "#ffffff" },
          "button": { "background": "#558b2f", "text": "#ffffff" }
        },
        "theme": "classic",
        "position": "bottom-right",
        "content": { "message": "This website uses fruit cookies to ensure you get the juiciest tracking
experience.", "dismiss": "Me want it!", "link": "But me wait!", "href":
"https://www.youtube.com/watch?v=9PnbKL3wuH4" }
      })});
  </script>
<link rel="stylesheet" href="styles.css"></head>
<body class="mat-app-background bluegrey-lightgreen-theme">
  <app-root></app-root>
<script src="runtime-es2015.js" type="module"></script><script src="runtime-es5.js" nomodule
defer></script><script src="polyfills-es5.js" nomodule defer></script><script src="polyfills-
es2015.js" type="module"></script><script src="vendor-es2015.js" type="module"></script><script
src="vendor-es5.js" nomodule defer></script><script src="main-es2015.js"
type="module"></script><script src="main-es5.js" nomodule defer></script></body>
</html>
```

Information Risk Vulnerability: **Passive 10027 - Information Disclosure - Suspicious Comments**
Description: The response appears to contain suspicious comments which may help an attacker. Note: Matches made within script blocks or files are against the entire content not only comments.
Solution: Remove all comments that return information that may help an attacker and fix any underlying problems they refer to.
URL: http://192.168.0.20:3000/polyfills-es5.js

Information Risk Vulnerability: **Passive 10096 - Timestamp Disclosure**
Description: A timestamp was disclosed by the application/web server - Unix
Information: 33333333, which evaluates to: 1971-01-21 14:15:33
Solution: Manually confirm that the timestamp data is not sensitive, and that the data cannot be aggregated to disclose exploitable patterns.
Reference: http://projects.webappsec.org/w/page/13246936/Information%20Leakage
URL: http://192.168.0.20:3000/styles.css

Vulnerabilities discovered with NMAP-Vulners / port 22:

Description: Common Vulnerabilities and Exposures
ID: CVE-2020-14145
Information: The client side in OpenSSH 5.7 through 8.4 has an Observable Discrepancy leading to an information leak in the algorithm negotiation. This allows man-in-the-middle attackers to target initial connection attempts (where no host key for the server has been cached by the client).

Description: Common Vulnerabilities and Exposures
ID: CVE-2021-28041
Information: ssh-agent in OpenSSH before 8.5 has a double free that may be relevant in a few less-common scenarios, such as unconstrained agent-socket access on a legacy operating system, or the forwarding of an agent to an attacker-controlled host.

Description: Common Vulnerabilities and Exposures

ID: CVE-2020-12062

Information: DISPUTED The scp client in OpenSSH 8.2 incorrectly sends duplicate responses to the server upon a utimes system call failure, which allows a malicious unprivileged user on the remote server to overwrite arbitrary files in the client’s download directory by creating a crafted subdirectory anywhere on the remote server. The victim must use the command scp -rp to download a file hierarchy containing, anywhere inside, this crafted subdirectory. NOTE: the vendor points out that “this attack can achieve no more than a hostile peer is already able to achieve within the scp protocol” and “utimes does not fail under normal circumstances.”

Description: Common Vulnerabilities and Exposures

ID: CVE-2020-15778

Information: DISPUTED scp in OpenSSH through 8.3p1 allows command injection in the scp.c to remote function, as demonstrated by backtick characters in the destination argument. NOTE: the vendor reportedly has stated that they intentionally omit validation of “anomalous argument transfers” because that could “stand a great chance of breaking existing workflows.”

Methodology

The Information Security Department based the findings and recommendations, outlined in this report, on application vulnerability scans performed against the application.

Assessment Tools Selection

OWASP ZAP is an open-source web application security scanner.

Nmap ("Network Mapper") is a free and open source (license) utility for network discovery and security auditing.

Vulners is an Nmap NSE script, using some well-known services to provide info on vulnerabilities.

Assessment Methodology Detail

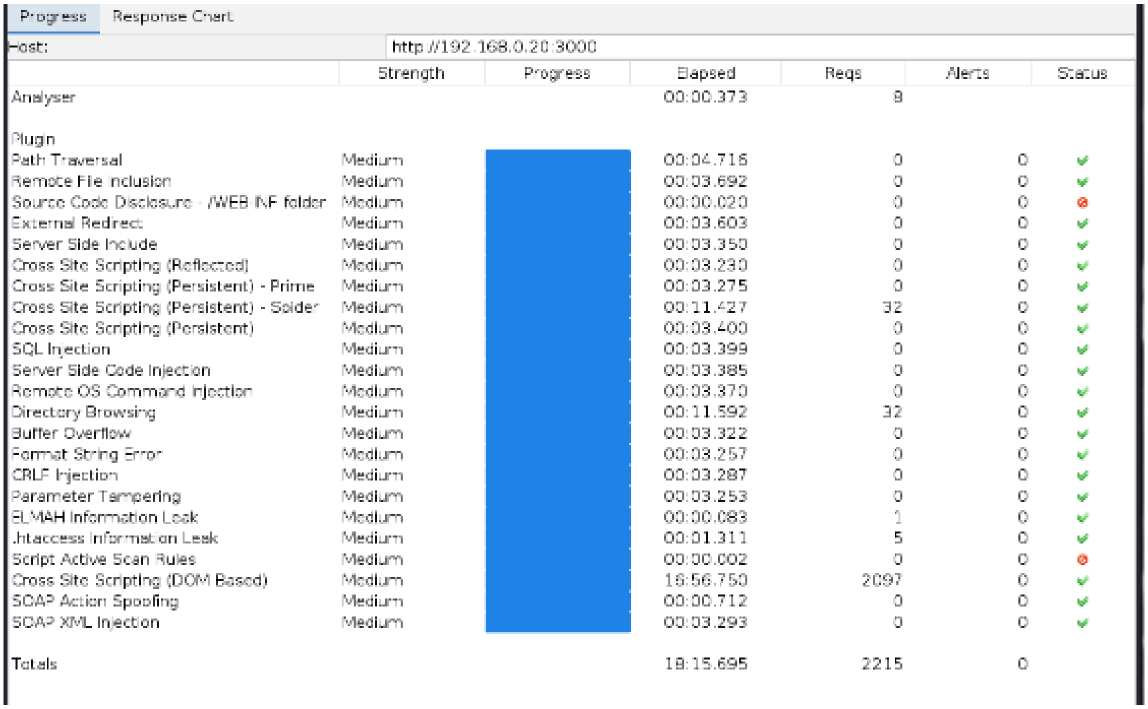
OWASP ZAP Automated Application Scan

The Information Security Department used several Open-Source tools to survey the targeted environment and identify potential vulnerabilities. The automated scanning software identifies application-level vulnerabilities.

The scope of testing with OWASP ZAP includes the following:

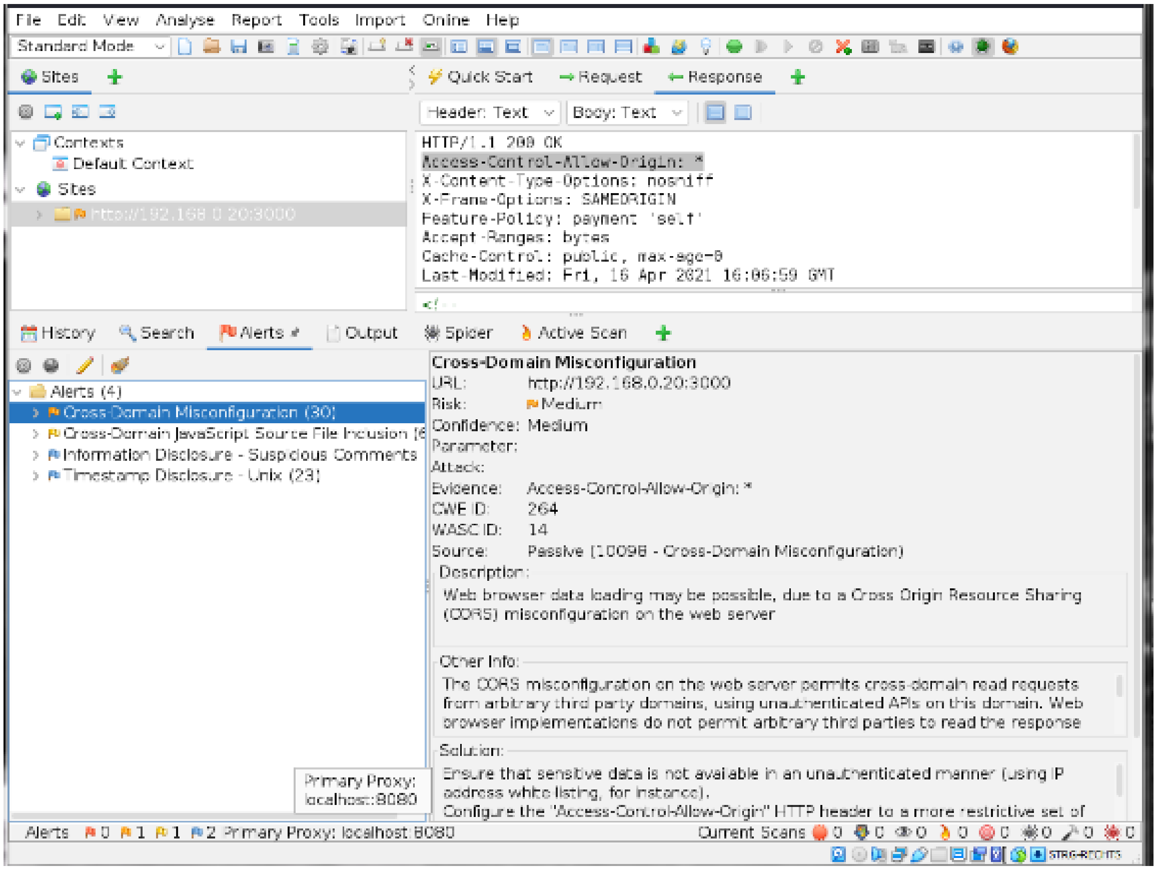
- SQL Injection
- Path Traversal
- Remote File Inclusion
- Source Code Disclosure
- External Redirect
- Server Side Include (Reflected)
- Cross Site Scripting (Persistent) - Prime
- Cross Site Scripting (Persistent) - Spider
- Cross Site Scripting (Persistent)
- Server-Side Code Injection
- Remote OS Command Injection
- Directory Browsing
- Buffer Overflow
- Format String Overflow
- CRLF Injection
- Parameter Tampering
- ELMAH Information Leak
- .htaccess Information Leak
- Script Active Scan Rules
- Cross Site Scripting (DOM Based)
- SOAP-Action Spoofing
- SOAP-XML Injection

The screenshots below show set up and vulnerability scan results in **OWASP Zap**:



Host: http://192.168.0.20:3000

	Strength	Progress	Elapsed	Reqs	Alerts	Status
Analyser			00:00.373	8		
Plugin						
Path Traversal	Medium		00:04.716	0	0	✓
Remote File Inclusion	Medium		00:03.692	0	0	✓
Source Code Disclosure - /WEB-INF folder	Medium		00:00.020	0	0	⊗
External Redirect	Medium		00:03.603	0	0	✓
Server Side Include	Medium		00:03.350	0	0	✓
Cross Site Scripting (Reflected)	Medium		00:03.230	0	0	✓
Cross Site Scripting (Persistent) - Prime	Medium		00:03.275	0	0	✓
Cross Site Scripting (Persistent) - Spider	Medium		00:11.427	32	0	✓
Cross Site Scripting (Persistent)	Medium		00:03.400	0	0	✓
SQL Injection	Medium		00:03.399	0	0	✓
Server Side Code Injection	Medium		00:03.385	0	0	✓
Remote OS Command Injection	Medium		00:03.370	0	0	✓
Directory Browsing	Medium		00:11.592	32	0	✓
Buffer Overflow	Medium		00:03.322	0	0	✓
Format String Error	Medium		00:03.257	0	0	✓
CRLF Injection	Medium		00:03.287	0	0	✓
Parameter Tampering	Medium		00:03.253	0	0	✓
ELMAH Information Leak	Medium		00:00.083	1	0	✓
.htaccess Information Leak	Medium		00:01.311	5	0	✓
Script Active Scan Rules	Medium		00:00.002	0	0	⊗
Cross Site Scripting (DOM Based)	Medium		18:56.750	2097	0	✓
SOAP Action Spoofing	Medium		00:00.712	0	0	✓
SOAP XML Injection	Medium		00:03.293	0	0	✓
Totals			19:15.695	2215	0	



```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
Accept-Ranges: bytes
Cache-Control: public, max-age=0
Last-Modified: Fri, 16 Apr 2021 16:06:59 GMT
```

Alerts (4)
- Cross-Domain Misconfiguration (30)
- Cross-Domain JavaScript Source File Inclusion (6
- Information Disclosure - Suspicious Comments
- Timestamp Disclosure - Unix (23)

Cross-Domain Misconfiguration

URL: http://192.168.0.20:3000
Risk: Medium
Confidence: Medium
Parameter:
Attack:
Evidence: Access-Control-Allow-Origin: *
CWE ID: 264
WASC ID: 14
Source: Passive (10098 - Cross-Domain Misconfiguration)

Description:
Web browser data loading may be possible, due to a Cross Origin Resource Sharing (CORS) misconfiguration on the web server

Other Info:
The CORS misconfiguration on the web server permits cross-domain read requests from arbitrary third party domains, using unauthenticated APIs on this domain. Web browser implementations do not permit arbitrary third parties to read the response

Solution:
Ensure that sensitive data is not available in an unauthenticated manner (using IP address white-listing, for instance).
Configure the "Access-Control-Allow-Origin" HTTP header to a more restrictive set of

Primary Proxy: localhost:8080

Alerts 0 1 1 2 Primary Proxy: localhost:8080 Current Scans 0 0 0 0 0 0 0 0

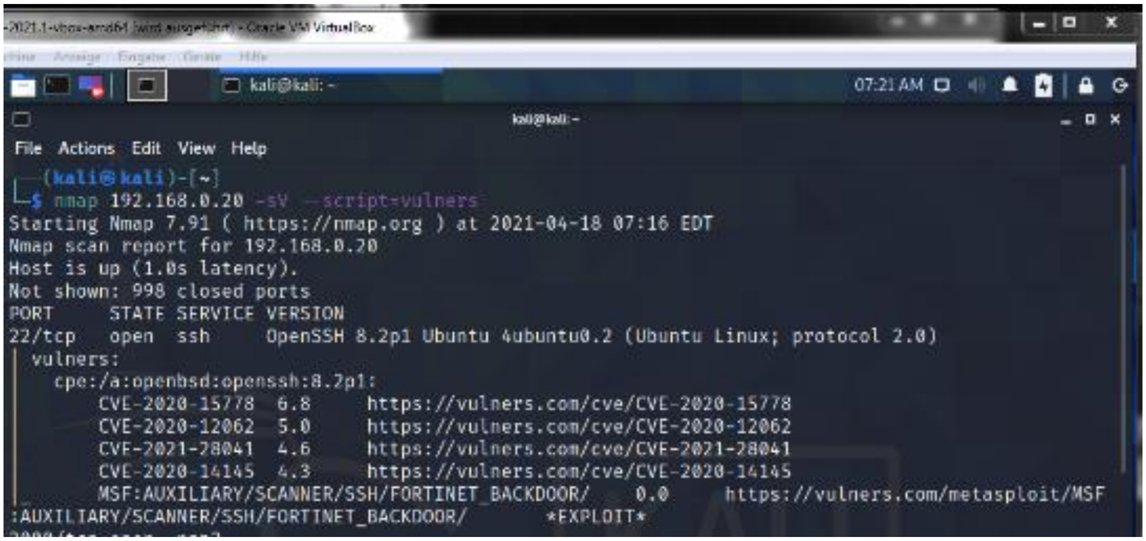
Nmap Vulners NSE script

Nmap-vulners queries the Vulners exploit database every time we use the NSE script.

```
nmap 192.168.0.20 -sV --script=vulners (-p 3000)
```

Output:

```
- - -
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.2p1 Ubuntu 4 ubuntu0.2 (Ubuntu Linux; protocol 2.0)
| vulners:
|   cpe:/a:openbsd:openssh:8.2p1:
|       CVE-2020-15778  6.8     https://vulners.com/cve/CVE-2020-15778
|       CVE-2020-12062  5.0     https://vulners.com/cve/CVE-2020-12062
|       CVE-2021-28041  4.6     https://vulners.com/cve/CVE-2021-28041
|       CVE-2020-14145  4.3     https://vulners.com/cve/CVE-2020-14145
- - -
```



-2021.1-vbox-amd64 [wird ausgeführt] - Oracle VM VirtualBox
kali@kali: ~
07:21 AM
kali@kali: ~
File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ nmap 192.168.0.20 -sV --script=vulners
Starting Nmap 7.91 ( https://nmap.org ) at 2021-04-18 07:16 EDT
Nmap scan report for 192.168.0.20
Host is up (1.0s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
| vulners:
|   cpe:/a:openbsd:openssh:8.2p1:
|       CVE-2020-15778  6.8     https://vulners.com/cve/CVE-2020-15778
|       CVE-2020-12062  5.0     https://vulners.com/cve/CVE-2020-12062
|       CVE-2021-28041  4.6     https://vulners.com/cve/CVE-2021-28041
|       CVE-2020-14145  4.3     https://vulners.com/cve/CVE-2020-14145
|_      MSF:AUXILIARY/SCANNER/SSH/FORTINET_BACKDOOR/    0.0     https://vulners.com/metasploit/MSF
:AUXILIARY/SCANNER/SSH/FORTINET_BACKDOOR/       *EXPLOIT*
```

All vulnerabilities have related references, definitions and severity which complete full information of any known bulletins. Visit https://vulners.com/ for detailed information.

Conclusion

The Information Security Department completed the vulnerability testing of the web application. This testing was based on the technologies and known threats as of the date of this document. All the security issues discovered during that exercise were analyzed and described in this report.

Please note that as technologies and risks change over time, the vulnerabilities associated with the operation of systems described in this report, as well as the actions necessary to reduce the exposure to such vulnerabilities, will also change.