Design Document

# Introduction

The report’s scope is to present a series of tools, challenges and findings of the vulnerability assessment performed against a target host. In addition, the report will include the selected methodology and the timeline to complete the performed evaluation.

# Domain

“An e-commerce site used to provide payment services and advice for commercial website operators”.

# Target Host

URL: <http://nismphp-env.eba-qgvjnwpz.us-east-1.elasticbeanstalk.com/>

IP: 184.73.185.129

# Vulnerability Assessment

## Tools

### Nmap

Nmap is the most popular network scanner globally and is used by millions of network specialists to perform several complex scanning techniques. In addition, Nmap was selected to determine the services running on the target host (Lyon, 2008).

### OpenVas

OpenVas is an open-source tool selected for its vulnerability scanning and reporting services. The tool assesses the provided target and compares the results with the CVE database to evaluate each vulnerability (incl. a solution) (Openvas.org, 2021).

### OWASP ZAP

The open-source web application security scanner OWASP ZAP (short for Zed Attack Proxy) scans web applications to discover vulnerabilities such as SQL injection, broken authentication, sensitive data exposure, insecure deserialisation and security misconfiguration. (Triad, 2021).

### TestSSL

TestSSL is an open-source tool that assesses every server’s port to define any support of SSL/TLS protocols and ciphers. Additionally, based on the results and the selection of that parameter in the execution command, it provides a report which contains every well-known vulnerability (connected with SSL/TLS, e.g., Heartbleed, etc.) applicable against each port.

### TheHarvester

theHarvester was selected for its ability to retrieve open-source intelligence (OSINT) to determine the domain's external threat landscape on the web and include the human factor within the assessment. The tool accumulates personal information such as names and e-mails, including network information (e.g. IPs, subdomains) and URLs using numerous public data sources.

# Methodology

## Threat Landscape

The term “Threat Landscape” consists of several threats and vulnerabilities under a specific environment/field that includes sensitive credentials, assets, threats, risks, and detected patterns (EISA, 2014).

After further inspection of the selected domain, we have identified and evaluated the following environments/fields as the most critical against any e-commerce site used to provide payment services.

### Human Resources

In order to evaluate the human resources field, it was necessary to scan the website for any information related to a user (i.e. e-mail address, PII and sensitive data). For that operation, **theHarvester** tool was utilised.

##### Commands

###### theHarvester -g -p -s -v -d <http://nismphp-env.eba-qgvjnwpz.us-east-1.elasticbeanstalk.com/>

**Command-Line flags:**

* -g: Google Dorks, used to include google dorks in google search.
* -p: Proxies, used to add proxies in the requests.
* -s: Shodan, used to query discovered hosts via Shodan.
* -v: Virtual, used to verify hostnames via DNS resolutions and virtual hosts.
* -d: Domain, to specify the target.

##### Results

The scan results revealed no personal information (names, e-mails, etc.) connected with the website.

### Network & Server Infrastructure

In our attempt to evaluate layers 3 - 7 (OSI model), it was necessary to perform a series of scans with the following tools:

#### Nmap

##### Commands

###### nmap -v -A -p- 184.73.185.129

**Command-Line Flags:**

* -v: Verbose (outputs all background processes to the console)
* -A: Aggressive (Enables script scanning, operating system detection and tracerouting)
* -p-: Scan the full range of all ports excluding zero.

###### nmap -v --script ssl-cert,ssl-enum-ciphers 184.73.185.129

**Command-Line Flags:**

* -v: Verbose (outputs all background processes to the console)
* --script ssl-cert: Attempts to retrieve the servers SSL certificate.
* --script ssl-enum-ciphers: Determines the compressors and cypher suites the server accepts.

##### Results

(Command 1) The results of the scan (see Appendix A) revealed three open ports and their services, namely 22 (SSH), 80(HTTP/Apache) and 443 (SSL/TLS).

(Command 2) Executes two scripts on the target host; the first, ssl-cert, attempts to retrieve the hosts SSL certificate (Nmap, 2021). In the case of the target host, we have observed that a network security control implemented by AWS prevents any reconnaissance activities indicated by “11 out of 11 dropped probes since last increase” (see Appendix B). Secondly, the ssl-enum-ciphers script initiates SSLv3/TLS connections to determine which compressors and cypher suites the server supports. We have observed no output in this script, indicating that SSL/TLS certificates are not issued or installed on the server (Nmap, 2021).

#### TestSSL

TestSSL was used to compare the results against the Nmap (command 2) report. It was discovered that the host had not implemented any certificates based on the scan outcome nor implemented SSL/TLS services on any port.

##### Commands

testssl 184.73.185.129

#### OpenVas

##### Commands

* gvm start
* “Full and Fast” Scan profile.

##### Results

As it can be seen in the report (Appendix D), the following results were found:

1. Port 22 (TCP) revealed an enabled SSH service.
2. Port 80 (TCP) revealed vulnerability regarding access to hidden files/ folders.

### Web Application

Regarding the evaluation of the threat landscape of the web application, the following tools were selected for the assessment:

* OpenVas
* OWASP ZAP
* Sqlmap
* testSSL

#### OWASP ZAP

OWASP ZAP revealed one significant vulnerability: the lack of an X-Frame Options HTTP header; this could lead to ‘ClickJacking’ attacks (see appendix C).

#### OpenVas

As observed in Appendix D, a vulnerability was detected in the web application that could potentially lead to information disclosure.

#### SqlMap

SqlMap was utilised to determine whether the server contained any SQL injection flaws. However, no results were provided (Refer to Appendix F).

##### Command

sqlmap -u <http://nismphp-env.eba-qgvjnwpz.us-east-1.elasticbeanstalk.com/add> -a

**Command-Line Flags:**

* -u: Target URL
* -a: Retrieve everything

#### TestSSL

In order to confirm the implementation of certificates in the web application, we have deployed TestSSL. As shown in Appendix (G), no certificates are implemented on the web application (incl. SSL/TLS services).

##### Command

testssl -U [http://nismphp-env.eba-qgvjnwpz.us-east-1.elasticbeanstalk.com](http://nismphp-env.eba-qgvjnwpz.us-east-1.elasticbeanstalk.com/)

**Command-Line Flags:**

-U: Test all known SSL/TLS vulnerabilities

## OWASP Web Application Security Guide

The OWASP Web Application Security Guide provides a comprehensive resource outlining approaches for testing web applications and servers. The scope of this report aligns directly with section 4.1 of the Web Application Security Guide, which outlines procedures for information gathering (OWASP, 2021). Therefore, we have selected this guideline to compare our methodology against industry best practices.

## Risk & Business Impact Assessment

In order to classify the discovered vulnerabilities and assess their impact against the business services, a risk and business impact assessment was performed based on the products of the methodology (please refer to the Risk & Business Impact Assessment excel file).

# Security Challenges

## Assumptions

As previously mentioned, the target host provides payment services for commercial website providers. Therefore, when performing information gathering or reconnaissance on the target host, we make assumptions regarding the types of security vulnerabilities we expect to discover, especially in insecure systems. Below you will find vulnerabilities that are specifically important to the target host domain.

### Sensitive Data Exposure (OWASP, 2017).

#### Typical Vulnerabilities

* Exposure of Sensitive Information Through Data Queries (cwe.mitre.org, 2021d)
* Inadequate Encryption Strength (cwe.mitre.org, 2021f).

### Broken Access Control (OWASP, 2021)

#### Typical Vulnerabilities

* Improper Limitation of a Pathname to a Restricted Directory (cwe.mitre.org, 2021a).
* Improper Access Control (cwe.mitre.org, 2021e).

### Cross-Site Scripting (OWASP, 2017).

#### Typical Vulnerabilities

* Improper Neutralization of Special Elements used in an SQL Command (cwe.mitre.org, 2021c).
* Improper Neutralization of Special Elements used in a Command (cwe.mitre.org, 2021b).

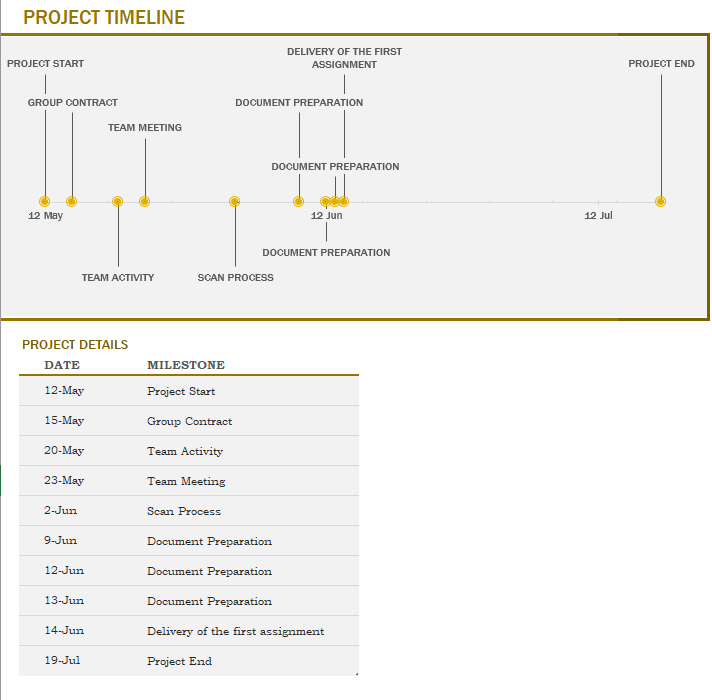
## Limitations

* The target host does not provide the expected services running on a commercial payment provider—resulting in limited scan results.

### Tools with no outcome

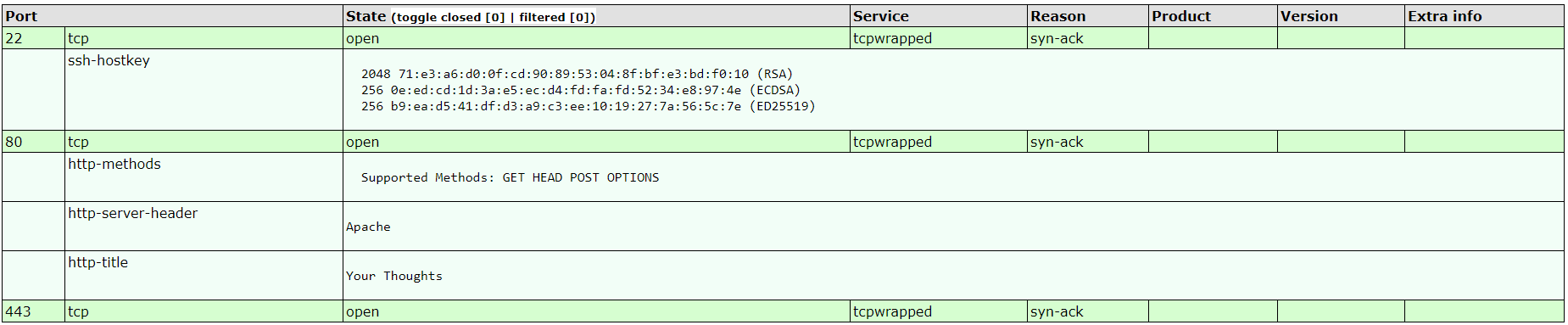
* DirBuster
* Nikto
* TheHarvester
* SQLmap
* testSSL

# Timeline

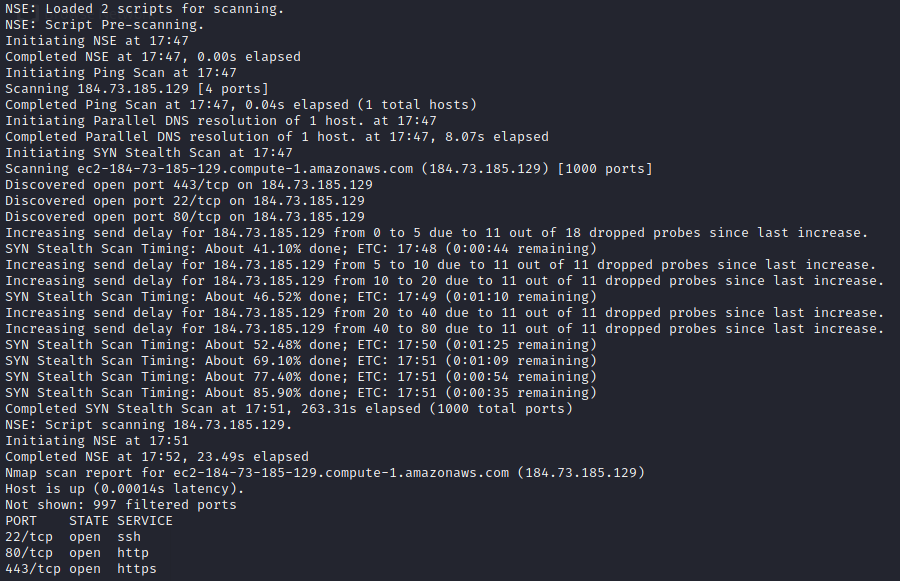


# Appendices

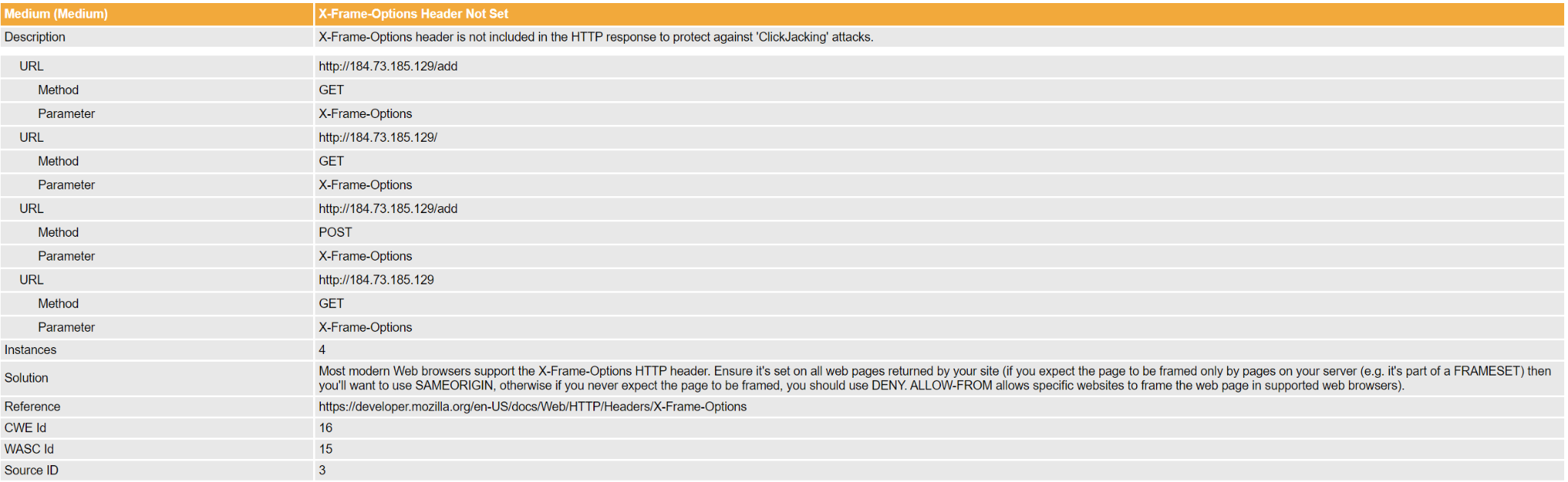
## Appendix A



## Appendix B

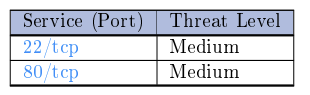


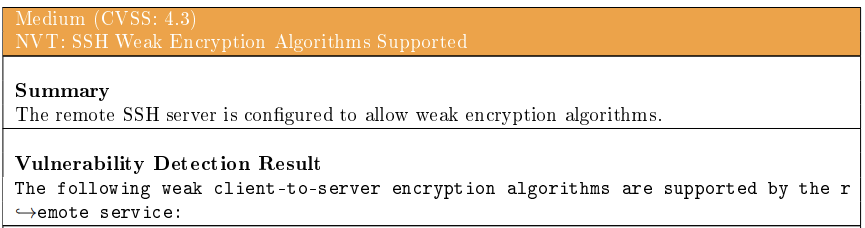
## Appendix C

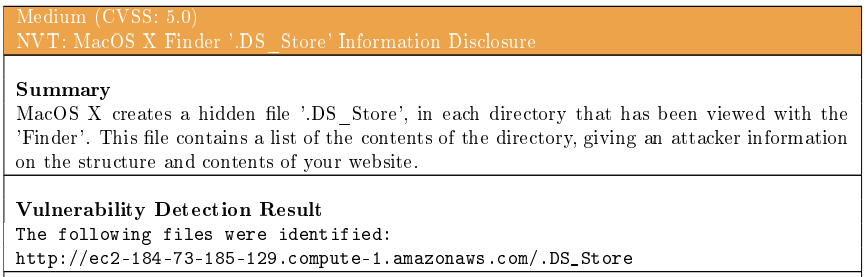


## 

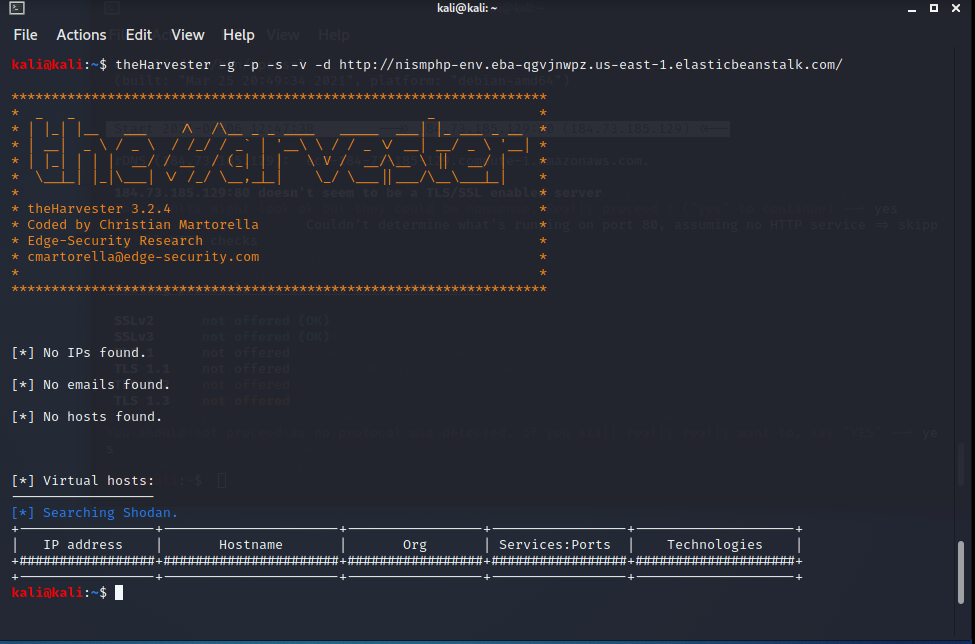
## Appendix D







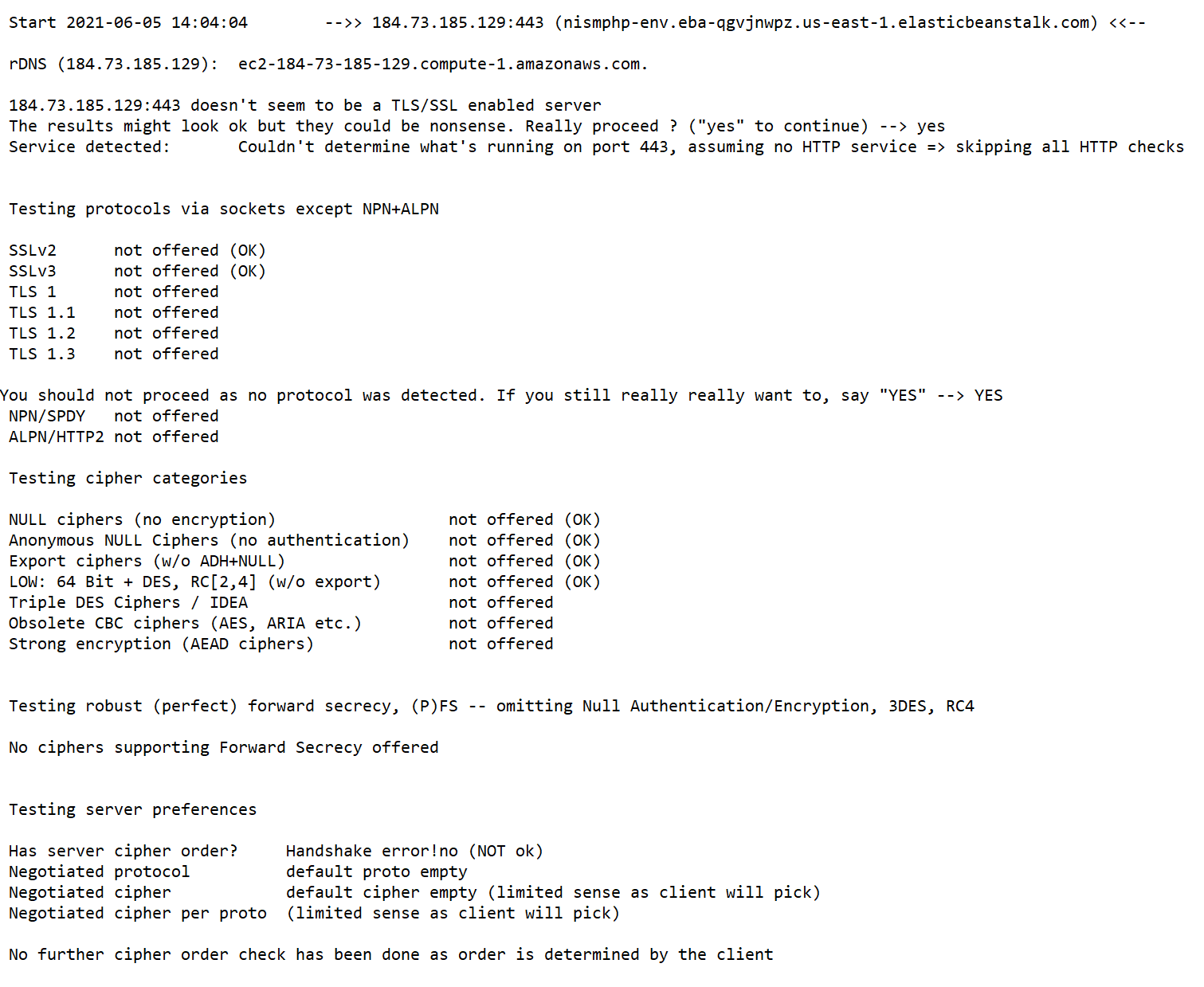
## Appendix E



## Appendix F



## Appendix G



# References

cwe.mitre.org (2021a) *CWE - CWE-22: Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal') (4.4)*. Available at:<https://cwe.mitre.org/data/definitions/22.html> [Accessed: 13 June 2021].

cwe.mitre.org (2021b) *CWE - CWE-77: Improper Neutralization of Special Elements used in a Command ('Command Injection') (4.4)*. Available at:<https://cwe.mitre.org/data/definitions/77.html>. [Accessed: 13 June 2021]

cwe.mitre.org (2021c) *CWE - CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') (4.4)*. Available at:<https://cwe.mitre.org/data/definitions/89.html> [Accessed: 13 June 2021].

cwe.mitre.org (2021d) *CWE - CWE-202: Exposure of Sensitive Information Through Data Queries (4.4)*. Available at:<https://cwe.mitre.org/data/definitions/202.html> [Accessed: 13 June 2021 2021].

cwe.mitre.org (2021e) *CWE - CWE-284: Improper Access Control (4.4)*. Available at:<https://cwe.mitre.org/data/definitions/284.html> [Accessed: 13 June 2021].

cwe.mitre.org (2021f) *CWE - CWE-326: Inadequate Encryption Strength (4.4)*. Available at:[https://cwe.mitre.org/data/definitions/326.htm](https://cwe.mitre.org/data/definitions/326.html%5BAccessed) [Accessed: 13 June 2021].

Digital.ai. (2021). *OWASP ZAP*. Available at: <https://digital.ai/technology/owasp-zap> [Accessed 11 June 2021].

EISA (2014). *ENISA Threat Landscape 2014*. Available at: <https://www.enisa.europa.eu/publications/enisa-threat-landscape-2014/at_download/fullReport> [Accessed 13 June 2021].

Lyon, G. (2008). *Nmap Network Scanning*.

Nmap (2021). *NSEDoc Reference Portal*. Available at: <https://nmap.org/nsedoc/>.

Nmap (2021). *ssl-cert NSE Script*. Available at:<https://nmap.org/nsedoc/scripts/ssl-cert.html>.

Nmap (2021). *SSL-enum-ciphers NSE Script*. Available at:<https://nmap.org/nsedoc/scripts/ssl-enum-ciphers.html>.

Openvas.org. (2021). *OpenVAS - OpenVAS - Open Vulnerability Assessment Scanner*. Available at: <https://www.openvas.org/> [Accessed 6 June 2021].

OWASP (2021). *4.1 Information Gathering*. Available at: <https://owasp.org/www-project-web-security-testing-guide/stable/4-Web_Application_Security_Testing/01-Information_Gathering/README.html>.

OWASP (2017). *A3:2017-Sensitive Data Exposure*. Available at: <https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure> [Accessed: 12 June 2021].

Owasp.org. (2021). *WSTG - v4.2 | OWASP*. Available at: <https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/00-Introduction_and_Objectives/README.html> [Accessed 11 June 2021].

Owasp.org. (2021). *WSTG - Stable | OWASP*. Available at: <https://owasp.org/www-project-web-security-testing-guide/stable/4-Web_Application_Security_Testing/01-Information_Gathering/> [Accessed 9 June 2021].

Owasp.org. (2021). *Broken Access Control for Software Security | OWASP Foundation*. Available at: <https://owasp.org/www-community/Broken_Access_Control> [Accessed 13 June 2021].

Owasp.org. (2017). *Cross Site Scripting (XSS) Software Attack | OWASP Foundation*. Available at: <https://owasp.org/www-community/attacks/xss/> [Accessed 13 June 2021].

Triad. (2021). *Using OWASP ZAP to find web app security vulnerabilities - Triad article*. Available at: <https://www.triad.co.uk/news/owasp-zap/> [Accessed 11 June 2021].