
# Penetration Test Report

## Nossas Cidades

V 1.0
Amsterdam, May 20th, 2022
Confidential

## Document Properties

| | |
|---|---|
| Client | Nossas Cidades |
| Title | Penetration Test Report |
| Targets | https://cade-meu-absorvente.pentest.bonde.org/<br>https://tem-gente-com-fome.pentest.bonde.org/<br>https://cadavotoconta.pentest.bonde.org/ |
| Version | 1.0 |
| Pentester | Abhinav Mishra |
| Authors | Abhinav Mishra, Marcus Bointon |
| Reviewed by | Marcus Bointon |
| Approved by | Melanie Rieback |

## Version control

| Version | Date | Author | Description |
|---|---|---|---|
| 0.1 | February 25th, 2022 | Abhinav Mishra | Initial draft |
| 0.2 | May 13th, 2022 | Abhinav Mishra | Retest Report |
| 0.3 | May 13th, 2022 | Abhinav Mishra | Ready For Review |
| 1.0 | May 20th, 2022 | Marcus Bointon | Review |

## Contact

For more information about this document and its contents please contact Radically Open Security B.V.


| | |
|---|---|
| Name | Melanie Rieback |
| Address | Science Park 608<br>1098 XH Amsterdam<br>The Netherlands |
| Phone | +31 (0)20 2621 255 |
| Email | info@radicallyopensecurity.com |

# Table of Contents

# 1 Executive Summary

## 1.1 Introduction

Between February 10, 2022 and February 25, 2022, Radically Open Security B.V. carried out a penetration test for Nossas Cidades.

This report contains our findings as well as detailed explanations of exactly how ROS performed the penetration test.

## 1.2 Scope of work

The scope of the penetration test was limited to the following targets:

- https://cade-meu-absorvente.pentest.bonde.org/
- https://tem-gente-com-fome.pentest.bonde.org/
- https://cadavotoconta.pentest.bonde.org/

The scoped services are broken down as follows:

- Penetration Testing: 2 days
- Reporting: 0.5 days
- **Total effort: 2.5 days**

## 1.3 Project objectives

ROS will perform a penetration test of the target applications with Nossas Cidades in order to assess the security. To do so ROS will access the applications and guide Nossas Cidades in attempting to find vulnerabilities, exploiting any such found to try and gain further access and elevated privileges.

## 1.4 Timeline

The Security Audit took place between February 10, 2022 and February 25, 2022.

## 1.5 Results In A Nutshell

During this grey-box penetration test we found 1 Moderate and 1 Elevated-severity issues.

The attack surface of the application is very small, and the application implements some additional security controls such as request throttling, input validation etc. This reduces the opportunities for security issues dramatically. However, some minor security issues were discovered during the audit, including insecure TLS implementations and missing security headers.

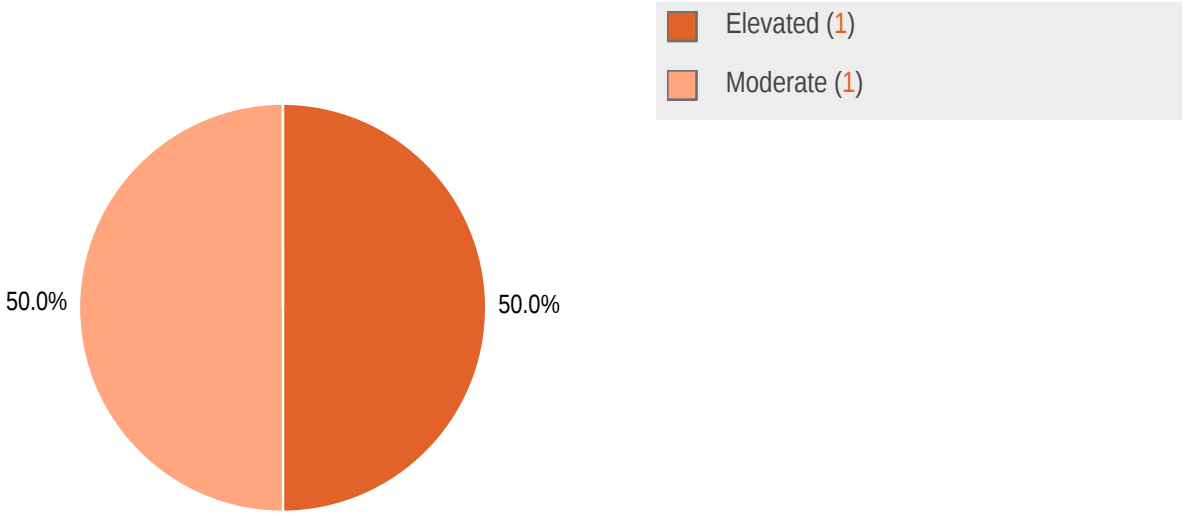
By exploiting these issues, an attacker might be able to affect data in transit between the clients and server.

In a follow-up retest we found that most issues had been resolved, but one remains incomplete.

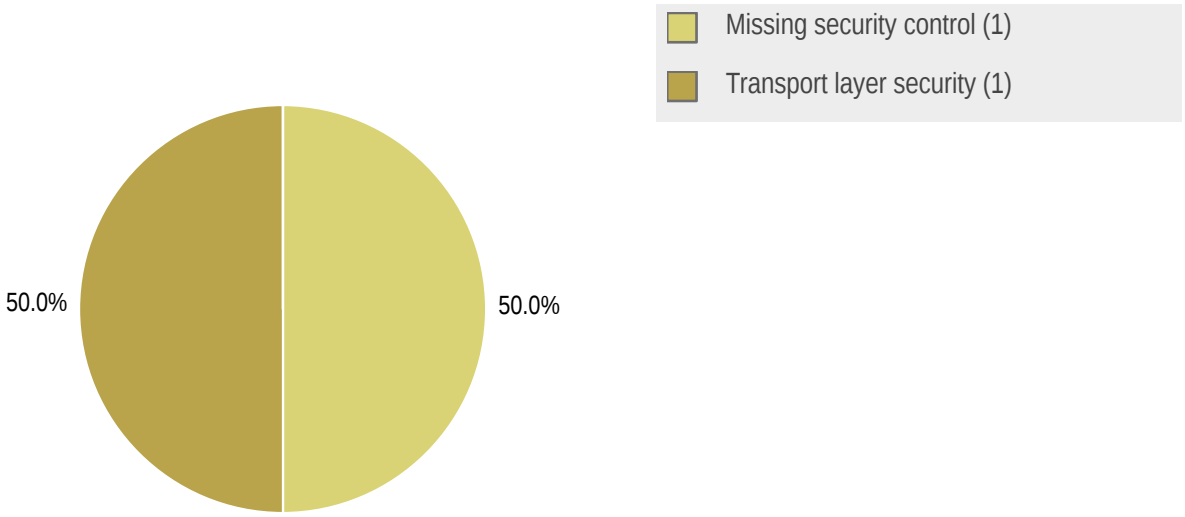
## 1.6 Summary of Findings

| ID | Type | Description | Threat level |
|---|---|---|---|
| CLN-001 | Transport Layer Security | The TLS/SSL implementation on the web application is insecure as it accepts vulnerable protocol versions and weak cipher suites. | Elevated |
| CLN-002 | Missing Security Control | The applications do not implement some important security headers. | Moderate |

### 1.6.1 Findings by Threat Level



### 1.6.2 Findings by Type




## 1.7 Summary of Recommendations

| ID | Type | Recommendation |
|---|---|---|
| CLN-002 | Missing Security Control | • Add these security headers to all application responses. |
| CLN-001 | Transport Layer Security | • Disable support of TLS 1.0 and TLS 1.1.<br>• Use TLS 1.2 and higher<br>• Do not accept weak TLS cipher suites |

# 2 Methodology

## 2.1 Planning

Our general approach during penetration tests is as follows:

1. **Reconnaissance**
   We attempt to gather as much information as possible about the target. Reconnaissance can take two forms: active and passive. A passive attack is always the best starting point as this would normally defeat intrusion detection systems and other forms of protection afforded to the app or network. This usually involves trying to discover publicly available information by visiting websites, newsgroups, etc. An active form would be more intrusive, could possibly show up in audit logs and might take the form of a social engineering type of attack.
2. **Enumeration**
   We use various fingerprinting tools to determine what hosts are visible on the target network and, more importantly, try to ascertain what services and operating systems they are running. Visible services are researched further to tailor subsequent tests to match.
3. **Scanning**
   Vulnerability scanners are used to scan all discovered hosts for known vulnerabilities or weaknesses. The results are analyzed to determine if there are any vulnerabilities that could be exploited to gain access or enhance privileges to target hosts.
4. **Obtaining Access**
   We use the results of the scans to assist in attempting to obtain access to target systems and services, or to escalate privileges where access has been obtained (either legitimately though provided credentials, or via vulnerabilities). This may be done surreptitiously (for example to try to evade intrusion detection systems or rate limits) or by more aggressive brute-force methods. This step also consist of manually testing the application against the latest (2017) list of OWASP Top 10 risks. The discovered vulnerabilities from scanning and manual testing are moreover used to further elevate access on the application.

## 2.2 Risk Classification

Throughout the report, vulnerabilities or risks are labeled and categorized according to the Penetration Testing Execution Standard (PTES). For more information, see: http://www.pentest-standard.org/index.php/Reporting

These categories are:

- **Extreme**
  Extreme risk of security controls being compromised with the possibility of catastrophic financial/reputational losses occurring as a result.

- **High**

  High risk of security controls being compromised with the potential for significant financial/reputational losses occurring as a result.
- **Elevated**

  Elevated risk of security controls being compromised with the potential for material financial/reputational losses occurring as a result.
- **Moderate**

  Moderate risk of security controls being compromised with the potential for limited financial/reputational losses occurring as a result.
- **Low**

  Low risk of security controls being compromised with measurable negative impacts as a result.

# 3 Reconnaissance and Fingerprinting

We were able to gain information about the software and infrastructure through the following automated scans. Any relevant scan output will be referred to in the findings.

- nmap – http://nmap.org
- testssl – https://github.com/drwetter/testssl.sh
- Burp Suite Pro – https://portswigger.net/burp/pro
- Nuclei – https://github.com/projectdiscovery/nuclei
- Naabu – https://github.com/projectdiscovery/naabu

# 4 Findings

We have identified the following issues:

## 4.1 CLN-002 — Missing security headers

| | |
|---|---|
| **Vulnerability ID:** CLN-002 | **Status:** Unresolved |
| **Vulnerability type:** Missing Security Control | |
| **Threat level:** Moderate | |

### Description:

The applications do not implement some important security headers.

### Technical description:

Affected Endpoints:

- `https://cade-meu-absorvente.pentest.bonde.org/`
- `https://tem-gente-com-fome.pentest.bonde.org/`
- `https://cadavotoconta.pentest.bonde.org/`

`Missing Security Headers`

- `Strict-Transport-Security` – HTTP Strict Transport Security (HSTS) is an excellent feature to support on your site and strengthens your implementation of TLS by getting the User Agent to enforce the use of HTTPS. Recommended value: `Strict-Transport-Security: max-age=31536000; includeSubDomains`.
- `Content-Security-Policy` – CSP is an effective measure to protect your site from XSS attacks. By whitelisting sources of approved content, you can prevent the browser from loading malicious assets. CSP can be complex to set up, and policies need to be tested carefully in order to avoid problems.
- `X-Content-Type-Options` – This header stops a browser from trying to sniff the MIME content type and forces it to stick with the declared content-type. The only valid value for this header is `X-Content-Type-Options: nosniff`.

`Note:` The security headers can be analyzed using a tool like Security Headers Checker (https://github.com/koenbuyens/securityheaders).

**Retest Result:** As per the retest done on 13th May 2022, this issue has been partially resolved. The `Strict-Transport-Security` and `X-Content-Type-Options` headers are now present, but the `Content-Security-Policy` header has not been implemented.

### Impact:

These security headers impact the security of the application and the API. For example; the lack of the CSP header means that resources may be loaded from any source without constraint, which is a primary source of XSS vulnerabilities, should other measures fail.

### Recommendation:

- Add these security headers to all application responses.

## 4.2 CLN-001 — Insecure TLS implementation

| | |
|---|---|
| **Vulnerability ID:** CLN-001 | **Status:** Resolved |
| **Vulnerability type:** Transport Layer Security | |
| **Threat level:** Elevated | |

### Description:

The TLS/SSL implementation on the web application is insecure as it accepts vulnerable protocol versions and weak cipher suites.

### Technical description:

**Affected Instances:**

- https://cade-meu-absorvente.pentest.bonde.org
- https://tem-gente-com-fome.pentest.bonde.org
- https://cadavotoconta.pentest.bonde.org/

**Vulnerable TLS versions accepted:**

- TLS 1.0

- TLS 1.1

**Weak TLS Ciphers**

| Ciphers |
| --- |
| TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013) ECDH secp521r1 (eq. 15360 bits RSA) FS WEAK |
| TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) ECDH secp521r1 (eq. 15360 bits RSA) FS WEAK |
| TLS_RSA_WITH_AES_128_GCM_SHA256 (0x9c) WEAK |
| TLS_RSA_WITH_AES_256_GCM_SHA384 (0x9d) WEAK |
| TLS_RSA_WITH_AES_128_CBC_SHA (0x2f) WEAK |
| TLS_RSA_WITH_AES_256_CBC_SHA (0x35) WEAK |
| TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA (0xc012) ECDH secp521r1 (eq. 15360 bits RSA) FS WEAK |
| TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa) WEAK |

**Tool Used & Approach:**
These issues were discovered using testssl.sh. Simply run the command `# ./testssl.sh <URL>`.

**Retest Result:** As per the retest done on 13th May 2022, this issue has been resolved. TLS versions 1.0 and 1.1 are no longer supported, and weak cipher suites are no longer accepted.

## Impact:

Weak SSL/TLS configuration may lead to attacks like machine-in-the-middle attack on the application traffic. TLS 1.0 has a number of cryptographic design flaws. Modern implementations of TLS 1.0 mitigate these problems, but newer versions of TLS (TLS 1.2 and higher) are designed against these flaws and should be preferred whenever possible. Accepting TLS 1.0 and TLS 1.1 makes the data in transit vulnerable to attacks in which an attacker can capture the encrypted data and decrypt it.

## Recommendation:

- Disable support of TLS 1.0 and TLS 1.1.
- Use TLS 1.2 and higher

- Do not accept weak TLS cipher suites

# 5 Non-Findings

In this section we list some of the things that were tried but turned out to be dead ends.

## 5.1 NF-004 — Non Finding Test cases

Summary of test cases which did not result in a vulnerability finding:

- Input validation tests
- Cross site scripting tests
- Test for server side request forgery
- Verify that access to sensitive records is protected, such that only authorized objects or data is accessible to each user (for example, protect against users tampering with a parameter to see or alter another user's account).
- Verify that directory browsing/indexing is disabled unless deliberately desired. Additionally, applications should not allow discovery or disclosure of file or directory metadata, such as Thumbs.db, .DS_Store, .git or .svn folders.
- Verify that the same access control rules implied by the presentation layer are enforced on the server side.
- Verify that all SQL queries, HQL, OSQL, NOSQL and stored procedures, calling of stored procedures are protected by the use of prepared statements or query parameterization, and thus not susceptible to SQL injection.
- Verify if the security controls are implemented to protect against anti-automation

# 6 Future Work

- **Source Code Review**
  External testing can only discover so much: Perform a source code review of all the code to try to discover vulnerabilities in it.
- **Retest of findings**
  When mitigations for the vulnerabilities described in this report have been deployed, a repeat test should be performed to ensure that they are effective and have not introduced other security problems.
- **Regular security assessments**
  Security is an ongoing process and not a product, so we advise undertaking regular security assessments and penetration tests, ideally prior to every major release or every quarter.

# 7 Conclusion

We discovered 1 Moderate and 1 Elevated-severity issues during this penetration test.

The issues we found were minor and easily resolved. However, while most of them were fixed when we retested, one issue was not completely resolved.

We recommend fixing all of the issues found and then performing a retest in order to ensure that mitigations are effective and that no new vulnerabilities have been introduced.

Finally, we want to emphasize that security is a process – this penetration test is just a one-time snapshot. Security posture must be continuously evaluated and improved. Regular audits and ongoing improvements are essential in order to maintain control of your corporate information security. We hope that this pentest report (and the detailed explanations of our findings) will contribute meaningfully towards that end.

Please don't hesitate to let us know if you have any further questions, or need further clarification on anything in this report.

# Appendix 1 Testing team

| | |
|---|---|
| Abhinav Mishra | Abhinav has 10+ years of experience in the penetration testing of web, mobile, infrastructure, API and other fields. He has received numerous accolades from multiple organisations for responsible disclosure of vulnerabilities. He is also known for providing trainings on web, mobile and infrastructure security. |
| Melanie Rieback | Melanie Rieback is a former Asst. Prof. of Computer Science from the VU, who is also the co-founder/CEO of Radically Open Security. |