<INSERT IMAGE>

{{ client.name }}

({{ client.short\_name }})  
Penetration Test Report

Prepared by:

{{ company.name }}

{{ report\_date }}

A red rectangle with black background

Description automatically generated

# Document Information

## Confidentiality

This document and all the information contained within are confidential and proprietary to {{ company.name }} and {{ client.name }}. The utmost care should be exercised when handling, referring to, or copying this document. {{ company.name }} authorizes {{ client.name }} to view and communicate this document as they see fit in accordance with {{ client.name }}’s data handling policies.

## Legal Disclaimer

No warranties are provided by {{ company.name }} for this penetration test report with respect to the accuracy, reliability, or correctness of the information in this document. This report is delivered “as is” with findings and recommendations reflecting only the information obtained during the assessment. {{ company.name }} does not assume liability from any damages, indirectly or directly, related to the reliance of information provided in this report, and it is highly recommended to thoroughly evaluate the business impact of changes before the implementation of them.

## Contact Information

|  |  |
| --- | --- |
| **{{ company.name }}** | |
| **Name** | Derya Yavuz Ionescu |
| **Role** | Lead Technical Representative |
| **Email** |  |
| **{{ client.name }}** | |
| **Name** |  |
| **Role** |  |
| **Email** |  |

# Table of Contents

[1. Document Information 2](#_Toc210896395)

[1.1. Confidentiality 2](#_Toc210896396)

[1.2. Legal Disclaimer 2](#_Toc210896397)

[1.3. Contact Information 2](#_Toc210896398)

[2. Table of Contents 3](#_Toc210896399)

[3. Executive Summary 6](#_Toc210896400)

[3.1. Assessment Overview 6](#_Toc210896401)

[3.2. Engagement Timeline 6](#_Toc210896402)

[3.3. Findings Count 6](#_Toc210896403)

[3.4. Scope 6](#_Toc210896404)

[3.5. Key Strengths 6](#_Toc210896405)

[3.6. Key Findings 6](#_Toc210896406)

[3.7. Strategic Recommendations 6](#_Toc210896407)

[4. Governance and Regulatory Compliance 7](#_Toc210896408)

[4.1. Payment Card Industry Data Security Standard (PCI DSS) 7](#_Toc210896409)

[4.1.1. PCI DSS Compliance Findings 7](#_Toc210896410)

[4.1.2. Prioritized Approach for PCI DSS 8](#_Toc210896411)

[4.2. International Maritime Organization (IMO) SOLAS (Safety of Life at Seas) 8](#_Toc210896412)

[4.2.1. IMO SOLAS Compliance Findings 8](#_Toc210896413)

[4.3. Standards for Privacy of Individually Identifiable Health Information - Health Insurance Portability and Accountability Act (HIPAA) 9](#_Toc210896414)

[4.3.1. HIPAA Compliance Findings 9](#_Toc210896415)

[5. Strategic Roadmap 10](#_Toc210896416)

[6. Testing Details 11](#_Toc210896417)

[6.1. Methodology Overview 11](#_Toc210896418)

[6.2. Scope 11](#_Toc210896419)

[6.2.1. Assessment Access Assets 11](#_Toc210896420)

[6.2.2. Authorized Engagement Assets 11](#_Toc210896421)

[6.3. Approach 11](#_Toc210896422)

[6.4. Timeframe 11](#_Toc210896423)

[6.5. Network Map 12](#_Toc210896424)

[7. Attack Narrative 13](#_Toc210896425)

[7.1. Pre-Engagement 13](#_Toc210896426)

[7.2. Day 1 13](#_Toc210896427)

[8. Finding Classifications 14](#_Toc210896428)

[8.1. Business Impact 14](#_Toc210896429)

[8.2. CVSS Score 14](#_Toc210896430)

[8.3. Naming Schema 14](#_Toc210896431)

[9. Finding Details 15](#_Toc210896432)

[9.1. Findings Summary 15](#_Toc210896433)

[9.2. Critical Risk Findings 17](#_Toc210896434)

[9.2.1. 17](#_Toc210896435)

[9.3. High Risk Findings 18](#_Toc210896436)

[9.3.1. 18](#_Toc210896437)

[9.4. Moderate Risk Findings 19](#_Toc210896438)

[9.4.1. 19](#_Toc210896439)

[9.5. Low Risk Findings 20](#_Toc210896440)

[9.5.1. 20](#_Toc210896441)

[9.6. Informational Findings 21](#_Toc210896442)

[9.6.1. 21](#_Toc210896443)

[10. Conclusion 23](#_Toc210896444)

[11. Appendix A: Non-Compliance Findings 24](#_Toc210896445)

[11.1. Payment Card Industry Data Security Standard (PCI DSS) 24](#_Toc210896446)

[11.2. International Maritime Organization SOLAS Compliance 24](#_Toc210896447)

[11.3. HIPAA Compliance 24](#_Toc210896448)

[12. Appendix B: Testing Methodology 26](#_Toc210896449)

[12.1. Penetration Testing Execution Standard (PTES) 26](#_Toc210896450)

[12.2. OWASP Top 10 27](#_Toc210896451)

[13. Appendix C: Findings Legend 32](#_Toc210896452)

[14. Appendix D: Logical Systems 33](#_Toc210896453)

[14.1. Logical System Findings: Web Applications (WA) 33](#_Toc210896454)

[14.2. Logical System Findings: Active Directory (AD) 33](#_Toc210896455)

[14.3. Logical System Findings: XX 33](#_Toc210896456)

[14.4. Logical System Findings: XX 34](#_Toc210896457)

[15. Appendix E: Tools Used 35](#_Toc210896458)

[15.1. Reconnaissance Tools 35](#_Toc210896459)

[15.2. Exploitation Tools 40](#_Toc210896460)

[15.3. Post-Exploitation Tools 42](#_Toc210896461)

[16. Appendix F: OSINT Assessment 44](#_Toc210896462)

[16.1. OSINT Findings 44](#_Toc210896463)

[16.1.1. Digital Media and Information Disclosure 44](#_Toc210896464)

[16.1.2. Missing Domain DMARC Record 44](#_Toc210896465)

[17. Appendix G: Phishing Assessment 45](#_Toc210896466)

[17.1. Spear Phishing 45](#_Toc210896467)

[18. Appendix H: Assessment Artifacts 46](#_Toc210896468)

# Executive Summary

## Assessment Overview

{{ company.name }} was contacted by {{ client.name }} (hereafter referred to as {{ client.short\_name }} ) to conduct a security assessment of their network assets. This assessment focused on assessing {{ client.short\_name }}’s current security posture across the following contexts:

* W
* W
* W
* W
* W

In total, {{ company.name }} identified XX total findings during our assessment with XX% of the findings from the previous assessment being resolved. It is recommended that {{ client.short\_name }} takes the necessary steps to evaluate and remediate these findings in order of severity. The discovered flaws in their current state have the potential to threaten public safety, including loss of life. Leaving these systems as they are exposes {{ client.short\_name }} to multiple risks: intrusion that could disrupt business operations, costly responses to cover impacted parties, loss of trust from customers and partners, and regulatory jeopardy. Regulatory compliance failures can result in monthly fines of up to $XXX,XXX until the issues are resolved, or a fine of $XXX,XXX if a data breach occurs within this period. Based on the identified weaknesses and their respective risk ratings, {{ company.name }} developed a comprehensive security strategy roadmap to assist {{ client.short\_name }} in addressing the outlined vulnerabilities. Please refer to **Strategic Roadmap** for further details.

## Engagement Timeline

* 09/22/2025 – VSA Submission Accepted
* 10/1/2025 – VSA Answers Received
* 10/18/2025 – Q4 Engagement Conducted

## Findings Count

Critical Risk Findings: ***X***

High Risk Findings: ***X***

Moderate Risk Findings: ***X***

Low Risk Findings: ***X***

Informational Findings: ***X***

## Scope

* Subnet 1X.X.X.X/24
* Subnet 2X.X.X.X/24
* Subnet 3X.X.X.X/24
* Subnet 4X.X.X.X/24
* Subnet 5X.X.X.X/24

## Key Strengths

## Key Findings

## Strategic Recommendations

{{ company.name }} deeply appreciates the opportunity to conduct this assessment for your organization. Your trust in our security expertise is valuable, and we stand committed to assist you to achieve a stronger and more robust security posture.

# Governance and Regulatory Compliance

{{ company.name }} would like to preface this section by stating that the purpose of this assessment performed for {{ client.name }} is to perform a best-effort penetration test, instead of a fully comprehensive compliance audit. However, during the course of our technical assessment, {{ company.name }} engineers identified several findings that may carry compliance implications. While these observations are framed primarily to provide further context into existing security vulnerabilities, they could also represent potential areas of non-compliance with applicable regulatory or industry standards.

It is important to note that unresolved issues of this nature can introduce not only heightened security risk but also regulatory and legal exposure for {{ client.name }}. Organizations that fail to remediate security shortcomings that overlap with compliance requirements may face penalties, reputational damage, or limitations in doing business in regulated environments. The observations provided in this section are therefore intended to help {{ client.name }} understand where technical gaps may translate into regulatory jeopardy, and to emphasize the importance of timely and effective remediation.

Below is context about industry-standard regulations, relevant chapters and associated findings.

## Payment Card Industry Data Security Standard (PCI DSS)

The Payment Card Industry Data Security (PCI DSS)[[1]](#footnote-2) is a twelve (12) requirement actionable framework to ensure data security when processing, transmitting, and storing cardholder data. This standard was created collaboratively by major credit card companies such as Visa, MasterCard, and American Express to reduce the risk of data breaches and fraud. Companies that are looking to process credit cards are required to continuously adhere to PCI DSS to ensure the security of payment card data. Failure to comply can lead to monthly fines (which can range from $XX,XXX to $XX,XXX based on {{ company.name }} classifying {{ client.short\_name }} as a Level X or Level X merchant from preliminary data), increased auditing requirements, inability to process credit cards, loss of public trust, lawsuits, and data breaches. In addition, data breaches that occur during periods of non-compliance can result in fines ranging from $50 to $90 per customer, which can equate to a total of $XXX,XXX.

### PCI DSS Compliance Findings

{{ company.name }} chose to assess {{ client.short\_name }} compliance utilizing version 4.0[[2]](#footnote-3) of PCI DSS as version 3.2.1 of PCI DSS is set to retire after March 2024. During our assessment, {{ company.name }} discovered XX potential PCI DSS violations.

[Summary of findings w/ Distribution]

A detailed list of the PCI DSS compliance findings can be found in Appendix A.

### Prioritized Approach for PCI DSS

The Prioritized Approach for PCI DSS[[3]](#footnote-4) is a roadmap of six risk-based security milestones provided by PCI DSS to help organizations incrementally protect against the highest risk factors while working to become PCI DSS compliant.

[Additional info]

|  |
| --- |
| **Prioritized Approach for PCI DSS Milestones** |
| 1. Do not store sensitive authentication data and limit cardholder data retention. |
| 1. Protect systems and networks and be prepared to respond to a system breach. |
| 1. Secure payment applications. |
| 1. Monitor and control access to your systems. |
| 1. Protect stored cardholder data. |
| 1. Complete remaining compliance efforts, and ensure all controls are in place. |

## International Maritime Organization (IMO) SOLAS (Safety of Life at Seas)

The International Maritime Organization Safety of Life at Sea (IMO SOLAS) Convention is the primary international treaty that establishes minimum safety standards for the construction, equipment, and operation of merchant ships. First adopted in 1914 after the Titanic disaster, SOLAS is binding on all contracting governments and enforced through national maritime administrations.

Adherence is not optional for cruise lines as it is a constant compliance requirement enforced by flag states, port state control and classification societies. Failure to comply with SOLAS can result in significant operational and legal consequences including but not limited to the detention of vessels, loss of certification and financial penalties.

Relevant cybersecurity-related chapters of SOLAS are below:

**Chapter XI-2: Ship and Port Facility Security (ISPS Code):** Cruise ships must implement ship security plans, access controls, surveillance systems, and designated Ship Security Officers to prevent unauthorized access or terrorist threats.

1 <https://www.imorules.com/SOLAS.html>

### IMO SOLAS Compliance Findings

[Summary of findings w/ Distribution]

A detailed list of the IMO SOLAS compliance findings can be found in Appendix A.

## Standards for Privacy of Individually Identifiable Health Information - Health Insurance Portability and Accountability Act (HIPAA)

The Health Insurance Portability and Accountability Act (HIPAA) is the main federal framework for protecting the privacy and security of individually identifiable health information (PHI). As cruise lines handle guest medical information on a day-to-day basis, this regulation becomes even more relevant. While HIPAA is primarily applied within U.S. healthcare systems, requirements extend to covered entities and business associates that collect and process PHI, including cruise lines that store crew and passenger medical information and operate medical centers on-board.

For cruise vessels, HIPAA compliance ensures the confidentiality, integrity, and availability of health records generated during voyages, such as passenger medical incidents, vaccination records, prescriptions, and emergency treatments delivered onboard. Adherence reduces liability exposure and safeguards passenger trust, especially as cruise operators increasingly integrate electronic health records (EHRs) and telemedicine services.

Relevant cybersecurity-related key provisions of HIPAA are below:

**Privacy Rule (45 CFR Part 160, Subparts A & E of Part 164):** Requires cruise medical staff to limit use and disclosure of PHI, provide notices of privacy practices, and obtain passenger authorization where required.

**Security Rule (45 CFR Part 160, Subparts A & C of Part 164):** Establishes technical, administrative, and physical safeguards for electronic PHI (ePHI), including encrypted storage, access control, and secure transmission to shoreside providers or insurers.

1 <https://www.hhs.gov/sites/default/files/privacysummary.pdf>

### HIPAA Compliance Findings

[Summary of findings w/ Distribution]

A detailed list of the HIPAA Privacy Rule compliance findings can be found in Appendix A.

# Strategic Roadmap

{{ company.name }} has created the following roadmap to provide {{ client.name }} with a tentative timeline by which remediation and mitigation strategies relevant to the assessment findings should be planned and executed. This roadmap was created to the best of our ability given the limited engagement timeline and visibility to the backend operations of {{ client.name }}. All remediation recommendations hereby proposed are to be interpreted by qualified industry professionals with experience working with the relevant systems to ensure proper implementation. Each recommendation is further explored in detail within the individual technical findings.

Furthermore, given that {{ client.name }} main operation is through cruise ships, an environment where some of the vulnerabilities identified in the assessment could lead to loss of human life and others may have operational consequences – it becomes extremely crucial to prioritize mitigation efforts based on criticality, business impact and ease of remediation. For each of the actions implemented in guidance of this roadmap, a cost-benefit analysis should be performed by {{ client.name }}. This approach would ensure that corporate investments are adequately directed towards measures that should be prioritized and that increase the overall security posture of the cruise line.

Immediate Efforts:

Immediate action items require a prioritized approach to remediate as these weaknesses could pose a direct threat to human life and safety – these can include critical data exposure or unauthorized code execution on critical systems.

Short-Term (1-3 Months) Efforts:

Short-term initiatives focus on addressing vulnerabilities that present material risk to operations or regulatory standing, where delayed remediation could increase the likelihood of exploitation or compliance violations.

Long-Term (6 Months – 1 Year) Efforts:

Long-term efforts are aimed at structural improvements and strategic investments that strengthen resilience, reduce systemic risk, and increase the security maturity of the organization.

# Testing Details

## Methodology Overview

To assess {{ client.short\_name }}’s internal network, {{ company.name }} utilizes the Penetration Testing Execution Standard (PTES)[[4]](#footnote-5) as it provides a comprehensive framework, covering all stages involved in an internal penetration test. For more details on the PTES methodology, please consult Appendix B.

## Scope

### Assessment Access Assets

{{ client.short\_name }} requested {{ company.name }} to assess {{ client.short\_name }}’s network using jump hosts that were to be utilized by a VPN connection. The details of these assets are as follows:



### Authorized Engagement Assets

{{ company.name }} was authorized by {{ client.short\_name }} to assess the following internal subnets during the penetration test:



## Approach

{{ company.name }}’s penetration test was performed with initial internal network access from provided Windows 10 and Kali Linux virtual machines under a “black-box” penetration testing approach where penetration testers had limited knowledge of network assets from, primarily from the Vendor Security Assessment answers posted by {{ client.short\_name }}, the network scope provided, and additional information supplied from {{ client.short\_name }} throughout the penetration testing period.

## Timeframe

{{ company.name }} was allotted X days, X, to perform an assessment over the authorized assets mentioned above where our engagement started at X and ended at X. After the assessment, our team worked diligently to deliver this report of our findings before X.

## Network Map

# Attack Narrative

## Pre-Engagement

In order to gather information that could potentially prove useful during the engagement, prior to live testing, {{ company.name }} conducted a scoping and intelligence-gathering phase against {{ client.name }}’ external attack surface. The goal of this research was to not only assist our team’s efficiency by getting familiar with {{ client.name }}’ technology stack but to also demonstrate that information that is publicly available could be used against {{ client.name }} in a malicious way.

{{ company.name }} scoured the internet extensively using a wide range of open-source enumeration tools and found a multitude of {{ client.name }} assets utilizing GitHub search, Google Dorks and social media which served useful in the investigation.

Findings from this phase could be found under the OSINT findings portion of the report in full detailed descriptions.

## Day 1

The first and only testing window focused on assessing internal network exposure from an assumed-compromise perspective, with engineers scanning the network through Kali hosts. Activities were carried out within the agreed 8-hour penetration testing scope, with an additional 7-hours of reporting activities. Access into the corporate environment began at <TIME> and ended <TIME>.

{{ company.name }} immediately conducted port scans on all in-scope subnets. Shortly after the initial reconnaissance, engineers identified….

Later in the assessment, the network segmentation measures were lifted within the corporate network and {{ company.name }} began performing reconnaissance on all other systems.

The remainder of time was allocated to actively exploit any identified weaknesses, prepare documentation and, to answer any questions that {{ client.name }} might have had in real time.

# Finding Classifications

{{ company.name }} utilized a two-dimensional matrix, see below, consisting of the business impact and Common Vulnerability Scoring System v4.0 (CVSS)[[5]](#footnote-6) score of each finding to categorize it within one of five overall security risk categories: informational, low, moderate, high, and critical. These categories were organized to prioritize the remediation of findings that would cause {{ client.short\_name }} financial loss, safety risks, non-compliance with governance requirements, and reputational impact. A detailed explanation of each section of the finding structure can be found at Appendix C.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Business Impact** | | | | |
| **CVSS Score** | **Informational (1)** | **Low (2)** | **Moderate (3)** | **High (4)** | **Critical (5)** |
| **N/A – 0.0 (a)** | 1a | 2a | 3a | 4a | 5a |
| **0.1 – 3.9 (b)** | 1b | 2b | 3b | 4b | 5b |
| **4.0 – 6.9 (c)** | 1c | 2c | 3c | 4c | 5c |
| **7.0 – 8.9 (d)** | 1d | 2d | 3d | 4d | 5d |
| **9.0 – 10.0 (e)** | 1e | 2e | 3e | 4e | 5e |
| **Overall Risk Key: ◼ Informational ◼ Low ◼ Moderate ◼ High ◼ Critical** | | | | | |

## Business Impact

{{ company.name }} incorporates business impact into the result for the categorization of a finding to help prioritize mitigation efforts and allocate resources effectively to address the most critical issues. We base our qualitative measurement on the ability of a finding to impact {{ client.short\_name }}’s ability to conduct business, ensure public safety and security, protect customer information, or stay in compliance with government regulations and business standards. As {{ company.name }} is operating under limited knowledge of the business operations of {{ client.short\_name }}, we would recommend {{ client.short\_name }} to review the business impact of these findings to provide a better understanding of the overall risk of said findings.

## CVSS Score

The Common Vulnerability Scoring System (CVSS) is a widely recognized industry standard used to evaluate and communicate the severity of security vulnerabilities in computer systems and software. It provides a structured framework for assessing a vulnerability's potential impact, exploitability, complexity, and privileges required for exploitation, assigning it a numeric score from 0 to 10, with higher scores indicating greater risk. CVSS assists organizations in prioritizing and addressing security flaws by considering their impact on confidentiality, integrity, and availability. In our security assessments, we adhere to the CVSS framework, which allows us to quantitatively gauge the severity of vulnerabilities.

## Naming Schema

{{ company.name }} utilizes a three-part structure for creating our findings’ unique identifiers which includes: a logical system abbreviation, a risk categorization abbreviation, and a numeric index within the risk categorization. For instance, a finding could be named “AD-H-05” to identify a finding within the active directory logical system (AD), that has been categorized as high based on our two-dimensional matrix (H), when it was the fifth finding within the high categorization. A list of all logical systems and their respective findings can be found at Appendix D.

# Finding Details

Findings Summary

|  |  |  |  |
| --- | --- | --- | --- |
| **Critical Risk Findings** | | | |
| **Unique ID** | **Finding Name** | **CVSS Score** | **Page Number** |
| **{%tr for finding in findings | filter\_severity([“Critical”]) %}** | | | |
| **XX-C-{{ “%02d” | format( loop.index | int ) }}** | **{{ finding.title }}** | **{{ finding.cvss\_score }}** |  |
| **{%tr endfor %}** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **High Risk Findings** | | | |
| **Unique ID** | **Finding Name** | **CVSS Score** | **Page Number** |
| **{%tr for finding in findings | filter\_severity([“High”]) %}** | | | |
| **XX-H-{{ “%02d” | format( loop.index | int ) }}** | **{{ finding.title }}** | **{{ finding.cvss\_score }}** |  |
| **{%tr endfor %}** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Moderate Risk Findings** | | | |
| **Unique ID** | **Finding Name** | **CVSS Score** | **Page Number** |
| **{%tr for finding in findings | filter\_severity([“Medium”]) %}** | | | |
| **XX-M-{{ “%02d” | format( loop.index | int ) }}** | **{{ finding.title }}** | **{{ finding.cvss\_score }}** |  |
| **{%tr endfor %}** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Low Risk Findings** | | | |
| **Unique ID** | **Finding Name** | **CVSS Score** | **Page Number** |
| **{%tr for finding in findings | filter\_severity([“Low”]) %}** | | | |
| **XX-L-{{ “%02d” | format( loop.index | int ) }}** | **{{ finding.title }}** | **{{ finding.cvss\_score }}** |  |
| **{%tr endfor %}** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Informational Findings** | | | |
| **Unique ID** | **Finding Name** | **CVSS Score** | **Page Number** |
| **{%tr for finding in findings | filter\_severity([“Informational”]) %}** | | | |
| **XX-I-{{ “%02d” | format( loop.index | int ) }}** | **{{ finding.title }}** | **{{ finding.cvss\_score }}** |  |
| **{%tr endfor %}** | | | |

Critical Risk Findings

{% for finding in findings | filter\_severity([“Critical”]) %}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **XX-C-{{ “%02d” | format( loop.index | int ) }}** | {{ finding.title }} | | |
| **Findings Categorization** | | | | |
| **Business Impact** | | <IMPACT> | **CVSS v4.0 Score** | {{ finding.cvss\_score }} |
| **CVSS Vector** | | {{ finding.cvss\_vector }} | | |

Technical Description

{{ finding.description | strip\_html }}

Business Impact Description

{{ finding.impact | strip\_html }}

Affected Systems

{{ finding.affected\_entities\_rt }}

Potential Compliance Violations

PCI DSS:

[OTHER]:

Finding Evidence

{{ finding.replication\_steps | strip\_html }}

Mitigations

{{ finding.mitigation | strip\_html }}

References

{{ finding.references | strip\_html }}

|  |  |
| --- | --- |
|  |  |



**END OF FINDING BLOCK**

{% endfor %}

High Risk Findings

{% for finding in findings | filter\_severity([“High”]) %}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **XX-H-{{ “%02d” | format( loop.index | int ) }}** | {{ finding.title }} | | |
| **Findings Categorization** | | | | |
| **Business Impact** | | <IMPACT> | **CVSS v4.0 Score** | {{ finding.cvss\_score }} |
| **CVSS Vector** | | {{ finding.cvss\_vector }} | | |

Technical Description

{{ finding.description | strip\_html }}

Business Impact Description

{{ finding.impact | strip\_html }}

Affected Systems

{{ finding.affected\_entities\_rt }}

Potential Compliance Violations

PCI DSS:

[OTHER]:

Finding Evidence

{{ finding.replication\_steps | strip\_html }}

Mitigations

{{ finding.mitigation | strip\_html }}

References

{{ finding.references | strip\_html }}

|  |  |
| --- | --- |
|  |  |



**END OF FINDING BLOCK**

{% endfor %}

Moderate Risk Findings

{% for finding in findings | filter\_severity([“Medium”]) %}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **XX-M-{{ “%02d” | format( loop.index | int ) }}** | {{ finding.title }} | | |
| **Findings Categorization** | | | | |
| **Business Impact** | | {{ finding.impact | strip\_html }} | **CVSS v4.0 Score** | {{ finding.cvss\_score }} |
| **CVSS Vector** | | {{ finding.cvss\_vector }} | | |

Technical Description

{{ finding.description | strip\_html }}

Business Impact Description

{{ finding.impact | strip\_html }}

Affected Systems

{{ finding.affected\_entities\_rt }}

Potential Compliance Violations

PCI DSS:

[OTHER]:

Finding Evidence

{{ finding.replication\_steps | strip\_html }}

Mitigations

{{ finding.mitigation | strip\_html }}

References

{{ finding.references | strip\_html }}

|  |  |
| --- | --- |
|  |  |



**END OF FINDING BLOCK**

{% endfor %}

Low Risk Findings

{% for finding in findings | filter\_severity([“Low”]) %}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **XX-L-{{ “%02d” | format( loop.index | int ) }}** | {{ finding.title }} | | |
| **Findings Categorization** | | | | |
| **Business Impact** | | {{ finding.impact | strip\_html }} | **CVSS v4.0 Score** | {{ finding.cvss\_score }} |
| **CVSS Vector** | | {{ finding.cvss\_vector }} | | |

Technical Description

{{ finding.description | strip\_html }}

Business Impact Description

{{ finding.impact | strip\_html }}

Affected Systems

{{ finding.affected\_entities\_rt }}

Potential Compliance Violations

PCI DSS:

[OTHER]:

Finding Evidence

{{ finding.replication\_steps | strip\_html }}

Mitigations

{{ finding.mitigation | strip\_html }}

References

{{ finding.references | strip\_html }}

|  |  |
| --- | --- |
|  |  |



**END OF FINDING BLOCK**

{% endfor %}

Informational Findings

{% for finding in findings | filter\_severity([“Informational”]) %}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **XX-I-{{ “%02d” | format( loop.index | int ) }}** | {{ finding.title }} | | |
| **Findings Categorization** | | | | |
| **Business Impact** | | {{ finding.impact | strip\_html }} | **CVSS v4.0 Score** | {{ finding.cvss\_score }} |
| **CVSS Vector** | | {{ finding.cvss\_vector }} | | |

Technical Description

{{ finding.description | strip\_html }}

Business Impact Description

{{ finding.impact | strip\_html }}

Affected Systems

{{ finding.affected\_entities\_rt }}

Potential Compliance Violations

PCI DSS:

[OTHER]:

Finding Evidence

{{ finding.replication\_steps | strip\_html }}

Mitigations

{{ finding.mitigation | strip\_html }}

References

{{ finding.references | strip\_html }}

|  |  |
| --- | --- |
|  |  |



**END OF FINDING BLOCK**

{% endfor %}

# Conclusion

The penetration test of {{ client.name }}’ corporate environment revealed multiple high and critical severity findings across both operational technology (OT) and traditional IT infrastructure.

Most notably, <INSERT\_VULNS\_HERE>. Attackers could manipulate or disrupt physical processes, leading to service outages or unsafe conditions.

To mitigate these risks, {{ company.name }} strongly recommends the prompt implementation of mitigation efforts outlined in the proposed strategic roadmap and technical security improvements. {{ company.name }} stands ready to provide guidance for any potential remediation efforts.

# Appendix A: Non-Compliance Findings

## Payment Card Industry Data Security Standard (PCI DSS)

|  |  |
| --- | --- |
| **PCI DSS Requirements** | **Related Findings** |
| **Build and Maintain a Secure Network and Systems** | |
| Requirement 1: Install and Maintain Network Security Controls |  |
| Requirement 2: Apply Secure Configurations to All System Components |  |
| **Protect Account Data** | |
| Requirement 3: Protect Stored Account Data |  |
| Requirement 4: Protect Cardholder Data with Strong Cryptography During Transmission Over Open, Public Networks |  |
| **Maintain a Vulnerability Management Program** | |
| Requirement 5: Protect All Systems and Networks from Malicious Software |  |
| Requirement 6: Develop and Maintain Secure Systems and Software |  |
| **Implement Strong Access Control Measures** | |
| Requirement 7: Restrict Access to System Components and Cardholder Data by Business Need to Know |  |
| Requirement 8: Identify Users and Authenticate Access to System Components |  |
| Requirement 9: Restrict Physical Access to Cardholder Data |  |
| **Regularly Monitor and Test Networks** | |
| Requirement 10: Log and Monitor All Access to System Components and Cardholder Data |  |
| Requirement 11: Test Security of Systems and Network Regularly |  |
| **Maintain an Information Security Policy** | |
| Requirement 12: Support Information with Organizational Policies and Programs |  |

## International Maritime Organization SOLAS Compliance

|  |  |
| --- | --- |
| **Other Regulation Requirements** | **Related Findings** |
| Chapter XI-2: Ship and Port Facility Security (ISPS Code): Cruise ships must implement ship security plans, access controls, surveillance. |  |
| Requirement 2: |  |
| Requirement 3: |  |
| Requirement 4: |  |

## HIPAA Compliance

|  |  |
| --- | --- |
| **Other Other Regulation Requirements** | **Related Findings** |
| Privacy Rule (45 CFR Part 160, Subparts A & E of Part 164) – Limit disclosure of PHI |  |
| Security Rule (45 CFR Part 160, Subparts A & C of Part 164) – Establish technical, administrative and physical safeguards for PHI |  |

# Appendix B: Testing Methodology

Penetration Testing Execution Standard (PTES)

For the assessment of {{ client.name }}’s internal network, {{ company.name }} utilized the Penetration Testing Execution Standard (PTES)[[6]](#footnote-7) due to its coherency and extensive coverage of all stages encountered throughout an internal penetration test. The PTES methodology separates each penetration test into 7 unique phases:

**1. Pre-Engagement Interactions:** This initial phase involves extensive communication and collaboration between the penetration testing team and the client organization. It's during this stage that the objectives, scope, and rules of engagement are defined, and a clear understanding of the target environment is established. By carefully addressing these aspects, the pre-engagement interactions lay the groundwork for a transparent, well-structured, and mutually beneficial penetration testing engagement that aligns with the client's specific security needs and goals.

**2. Intelligence Gathering:** Once pre-engagement interactions have concluded, the next phase of the methodology focuses on collecting information about the target organization and its assets. To collect this information, different techniques are utilized such as Open-Source Intelligence (OSINT), social engineering, and fingerprinting.

**3. Threat Modeling:** The primary goal of this stage is identifying and categorizing a business’s critical assets, mapping each asset to all probable attack vectors, and identifying and modeling the appropriate threat actors based on the nature of the assets.

**4. Vulnerability Analysis:** Next, the methodology then calls for an in-depth analysis of the client’s assets with the goal of discovering flaws in the systems and applications that are within the scope of the assessment. This process can involve the use of banner grabbing to identify services and versions, manual testing to discover vulnerabilities, and automated vulnerability scanners.

**5. Exploitation:** This stage involves revisiting all vulnerabilities gathered during the previous phases of the methodology, with the primary goal of exploiting these targets and gaining access to the client’s assets.

**6. Post-Exploitation:** Upon gaining access, the next step is evaluating the importance of the compromised asset and the risk that it poses, as well as searching for additional vulnerabilities such as privilege escalation or moving laterally within the client’s network.

**7. Reporting:** The final step of this methodology involves gathering all findings from the previous phases and generating a professional report for the client. The main purpose of the report is to convey all findings from the penetration test, as well as remediation techniques so that security is hardened as a result of the assessment.

OWASP Top 10

{{ company.name }} utilizes the OWASP Top 10[[7]](#footnote-8) as a foundational framework for evaluating Web Applications, focusing on identifying common vulnerabilities and misconfigurations. The overarching goal of the project is to establish a consensus among experts in web application security regarding the most prevalent issues in modern applications. The 2021 edition of the OWASP Top 10, which if the most recent, specifies the following web application security flaws:

|  |  |
| --- | --- |
| **OWASP Top 10** | |
| 1. Broken Access Control | 1. Cryptographic Failures |
| 1. Injection | 1. Insecure Design |
| 1. Security Misconfiguration | 1. Vulnerable and Outdated Components |
| 1. Identification and Authentication Failures | 1. Software and Data Integrity Failures |
| 1. Security Logging and Monitoring Failures | 1. Server-Side Request Forgery |

This structured approach empowers security professionals to comprehensively assess and address the most critical aspects of web application security based on the collective insights of the industry.

Below are brief explanations and potential technical and business impacts that each vulnerability category can reach:

|  |  |
| --- | --- |
| **Broken Access Control** | |
| **Description** | Broken access control occurs when a web app fails to enforce either authentication or the correct restrictions to sensitive web content. Attackers can exploit such flaws and gain unauthorized access to modify data or functions. |
| **Technical Impact** | Attackers can gain unauthorized access to sensitive files, data, or administrative functions connected to the website. They may also view, alter, or even delete critical user data for all users. |
| **Business Impact** | Loss of sensitive data or PII on users is a violation in data privacy / data protection regulations. Also, the loss or unauthorized access to sensitive data can cause reputational damage in releases such as data leaks. |

|  |  |
| --- | --- |
| **Cryptographic Failures** | |
| **Description** | Cryptographic failures are encryption errors caused by poor and weak algorithms that can either be decrypted or cracked easily. These failures can also be an issue of implementation of encryption or lack thereof, resulting in cleartext user data or passwords. |
| **Technical Impact** | Attackers can steal and edit sensitive information such as emails, names, passwords, credit card information, and other personal data. Weak encryption standards allow data to be decrypted and used to further attack the website. |
| **Business Impact** | Breaches in user privacy and exposure of sensitive information can result in financial loss for the customer and distrust in the business. Lack of encryption can result in regulatory fines and compliance issues. |

|  |  |
| --- | --- |
| **Injection** | |
| **Description** | Injection flaws occur when untrusted and untreated input is interpreted as code or commands by the web app. These can be database command or server commands that can allow unauthorized access to web resources. |
| **Technical Impact** | Attackers find “unsanitized” input fields and send payloads, which allows for reading of data or execution of commands. These injections can result in full database access, and in some cases a full system compromise. |
| **Business Impact** | Successful injection attacks often cause large data breaches, exposing entire customer databases. This allows attackers to also modify or destroy data, resulting in major downtime and loss of customer data. |

|  |  |
| --- | --- |
| **Insecure Design** | |
| **Description** | Insecure design refers to fundamental architectural or workflow decisions that fail to incorporate security principles from the start. This includes missing threat modeling, lack of abuse-case thinking, insecure default flows, or designs that assume trusted inputs or environments. |
| **Technical Impact** | Insecure design can permit entire classes of attacks across the system (privilege escalation, bypassing controls, logic abuse). Because the weakness exists at the architecture level, fixes are often non-trivial and may require re-architecting flows, adding authorization checks, or redesigning components. Attackers can chain multiple design flaws to achieve high-impact outcomes. |
| **Business Impact** | Design-level failures cause repeated security incidents, expensive rework, delayed releases, and loss of customer trust. They often require major development effort to remediate, increase time-to-market for new features, and can expose the organization to regulatory or contractual breaches |

|  |  |
| --- | --- |
| **Security Misconfigurations** | |
| **Description** | Security misconfiguration occurs when software, platforms, or infrastructure are left in insecure default states or configured incorrectly - such as default credentials, unnecessary services enabled, verbose error messages, or exposed admin interfaces. |
| **Technical Impact** | Misconfigurations often provide low-hanging fruit for attackers: easy authentication bypass, information leakage, or direct access to management interfaces. Attackers can exploit these to gain foothold, move laterally, or extract sensitive configuration and credential material. |
| **Business Impact** | These issues can lead to rapid compromise and data breaches, typically avoidable with basic hardening. Remediation may require patching configs across many systems, audits of infrastructure, and possible downtime – all of which raise operational costs and reputational risk. |

|  |  |
| --- | --- |
| **Vulnerable and Outdated Components** | |
| **Description** | Using libraries, frameworks, or platform components with known vulnerabilities (unpatched CVEs) creates risk. This includes third-party modules, containers, OS packages, and legacy systems that are no longer maintained. Relying on outdated components transfers responsibility for security to upstream projects that may have been fixed long ago. |
| **Technical Impact** | Exploitable third-party flaws can allow remote code execution, privilege escalation, or data exfiltration without any application-level bug. Attackers frequently target known CVEs in widely used components to scale attacks quickly. |
| **Business Impact** | The presence of vulnerable dependencies increases the likelihood of automated compromise, forces emergency patch cycles, and may violate compliance requirements. It can also create vendor/third-party management headaches and increase lifecycle costs for upgrades and testing. |

|  |  |
| --- | --- |
| **Identification and Authentication Failure** | |
| **Description** | These failures occur when user or system identity controls are weak or incorrectly implemented. Examples include broken or missing multi-factor authentication, predictable or exposed session identifiers, improper credential storage, or flawed password reset flows. |
| **Technical Impact** | Weak authentication enables account takeover, session hijacking, and unauthorized access to sensitive features or data. Attackers may impersonate users, escalate privileges, or abuse forgotten-password flows to gain persistence. |
| **Business Impact** | Account compromise leads to direct theft of data, fraud, customer churn, and potential regulatory fines. Restoring trust (password resets, notifications, forensic work) and fixing underlying identity issues can be costly and time consuming. |

|  |  |
| --- | --- |
| **Software and Data Integrity Failures** | |
| **Description** | Integrity failures appear when applications or systems accept or distribute code/data that has been tampered with or is not properly validated. This includes unsigned or unaudited updates, accepting untrusted serialized objects, or failing to validate package sources and CI/CD artifacts. |
| **Technical Impact** | Attackers can inject backdoors, tamper with binaries, or deliver malicious updates that run with elevated privileges. Compromised build artifacts or unverified dependencies become a vector for supply-chain attacks. |
| **Business Impact** | Integrity compromises can lead to widespread malware distribution, long-term undetected breaches, and catastrophic brand damage. Recovery often involves rebuilds, supply-chain audits, and substantial legal/PR consequences. |

|  |  |
| --- | --- |
| **Security Logging and Monitoring Failures** | |
| **Description** | This category covers insufficient or missing logging, lack of alerting, poor log retention, or blind spots in monitoring that prevent timely detection of security events. Even if defenses exist, they are useless without good telemetry and response capability. |
| **Technical Impact** | Without adequate logs and monitoring, intrusions go undetected, forensic investigations are limited, and attackers can persist longer. Incident response is slower and less effective, increasing the blast radius of compromises. |
| **Business Impact** | Delayed detection increases breach costs, regulatory exposure (failure to detect/report), and operational disruption. Investing in logging and monitoring maturity improves resilience but requires tooling, staffing, and process changes. |

|  |  |
| --- | --- |
| **Server-Side Request Forgery** | |
| **Description** | SSRF happens when an application fetches remote resources based on attacker-controlled input (URLs, hosts) without proper validation. Attackers manipulate the server into making requests to internal-only services, metadata endpoints, or external malicious sites. |
| **Technical Impact** | SSRF can expose internal networks, retrieve sensitive metadata (e.g., cloud instance credentials), or be chained into further attacks like port scanning and SSRF-to-RCE cases. Because the request originates from the server, it can bypass network restrictions placed on external clients. |
| **Business Impact** | Exploitation may lead to full cloud account compromise, data theft, or unauthorized lateral movement inside protected environments. Fixing SSRF involves input validation, egress filtering, and architectural controls — all of which can be time-consuming and critical to overall cloud security posture. |

# Appendix C: Findings Legend

To enhance clarity, {{ company.name }} has included this legend of our findings. This serves as a helpful reference guide to understanding the background behind each of the sections of our findings.

|  |  |
| --- | --- |
| **Section** | **Description** |
| **Unique ID** | The unique ID serves as an easy way to identify a specific finding as it is composed of the abbreviation of the relevant logical system, a risk categorization abbreviation, and a numeric index within the risk categorization. Additional information of the logical systems can be found in Appendix D. |
| **Finding Title** | The finding title is a short description of finding that can be utilized to understand the contents of the finding at a high-level overview. |
| **Business Impact** | {{ company.name }} utilizes a qualitative business impact rating based on the ability of the finding to impact the confidentiality of the business or customer data, the integrity of said data, the availability of business operations, the legal and regulatory compliance of the business, or the safety of employees and customers. The business impact is rated across one of five categories, Critical, High, Moderate, Low, Informational, and is used in tandem with the Common Vulnerability Scoring System (CVSS) v4.0 score to achieve the overall finding categorization. |
| **CVSS v4.0 Score** | The CVSS v4.0 score is a standardized numerical rating that quantifies the severity of a security vulnerability. It considers various factors, including the vulnerability's impact and how easy it is to exploit. A higher score indicates a more critical and potentially harmful vulnerability, aiding organizations in prioritizing and addressing security issues effectively. |
| **CVSS Vector** | The CVSS vector is an abbreviated string representation of the metrics utilized to calculate the CVSS score. This metric can be useful to identify reoccurring issues or understand the technical impact of the vulnerability at a brief glance. |
| **Technical Description** | The technical description gives a detailed explanation of a technical aspect of the finding, breaking down how the vulnerability exists and could be exploited. It helps technical teams understand fix the vulnerability more effectively. |
| **Business Impact Description** | The business impact description outlines the potential consequences of a finding on the organization’s operations, assets, and overall business continuity. |
| **Affected Systems** | The affected systems section includes all assets that are impacted by the specified finding. |
| **Potential Compliance Violations** | Potential compliance violations refer to aspects of regulatory, industry, or legal requirements that might be violated with respect to the contents of the findings. |
| **Mitigations** | The mitigations section offers practical strategies and recommendations to address and reduce the impact of identified findings. |
| **References** | The references section includes and references that might be helpful for resolving, reproducing, or understanding the finding. |
| **Finding Evidence** | Finding Evidence are a clear and concise set of instructions to allow {{ client.short\_name }} to verify findings and test potential solutions for remediation. |

# Appendix D: Logical Systems

During assessments, {{ company.name }} groups assets together logically based on their purpose as well as their relationship with other assets. This helps spot larger issues that might affect multiple devices as well as to understand the extent of a vulnerability. By organizing assets this way, we not only streamline the identification of problems but also provide valuable support to the teams managing these assets. A table of the logical systems, their abbreviation, and a description of said systems is as follows:

|  |  |  |
| --- | --- | --- |
| **Logical System** | **Abbreviation** | **Description** |
| Web Applications | WA | This logical system includes all web applications identified during the engagement: |
| Active Directory | AD | This logical system includes all Active Directory computers and domain controllers: |
|  |  |  |
|  |  |  |

Following this organization, our findings are presented in tables corresponding to each logical system by descending order of their severity. These tables provide a targeted and efficient remediation process, allowing teams to address issues specific to their areas of responsibility and enhance the overall security of the network.

Logical System Findings: Web Applications (WA)

|  |  |  |
| --- | --- | --- |
| **Unique ID** | **Finding Name** | **Page Number** |
| WA-C-01 |  |  |
| WA-H-02 |  |  |
| WA-M-03 |  |  |
| WA-L-04 |  |  |

Logical System Findings: Active Directory (AD)

|  |  |  |
| --- | --- | --- |
| **Unique ID** | **Finding Name** | **Page Number** |
| AD-C-01 |  |  |
| AD-H-02 |  |  |
| AD-M-03 |  |  |
| AD-L-04 |  |  |

Logical System Findings: XX

|  |  |  |
| --- | --- | --- |
| **Unique ID** | **Finding Name** | **Page Number** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Logical System Findings: XX

|  |  |  |
| --- | --- | --- |
| **Unique ID** | **Finding Name** | **Page Number** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Appendix E: Tools Used

## Reconnaissance Tools

|  |  |
| --- | --- |
| **AWS CLI** | |
| **Description** | The AWS Command Line Interface (AWS CLI) is a unified tool to manage your AWS services. Attackers can use the CLI to look into publicly exposed S3 buckets and more. |
| **Source** | <https://aws.amazon.com/cli/> |

|  |  |
| --- | --- |
| **BloodHound** | |
| **Description** | BloodHound is an Active Directory (AD) reconnaissance tool that can reveal hidden relationships and identify attack paths within an AD environment. |
| **Source** | <https://www.kali.org/tools/bloodhound/>  <https://github.com/SpecterOps/BloodHound> |

|  |  |
| --- | --- |
| **Enum4linux** | |
| **Description** | Enum4linux is a tool for enumerating information from Windows and Samba systems. |
| **Source** | <https://www.kali.org/tools/enum4linux/> |

|  |  |
| --- | --- |
| **Eyewitness** | |
| **Description** | EyeWitness is designed to take screenshots of websites, provide some server header info, and identify default credentials if possible. |
| **Source** | <https://www.kali.org/tools/eyewitness/>  <https://github.com/RedSiege/EyeWitness> |

|  |  |
| --- | --- |
| **Feroxbuster** | |
| **Description** | Feroxbuster is a tool designed to perform forced browsing. Forced browsing is an attack where the aim is to enumerate and access resources that are not referenced by the web application but are still accessible by an attacker. |
| **Source** | <https://www.kali.org/tools/feroxbuster/>  <https://github.com/epi052/feroxbuster> |

|  |  |
| --- | --- |
| **Gobuster** | |
| **Description** | Gobuster is a tool used to brute-force URIs including directories and files as well as DNS subdomains. |
| **Source** | <https://www.kali.org/tools/gobuster/>  <https://github.com/OJ/gobuster> |

|  |  |
| --- | --- |
| **NBTscan** | |
| **Description** | NBTscan is a program for scanning IP networks for NetBIOS name information. It sends NetBIOS status query to each address in supplied range and lists received information in human readable form. For each responded host it lists IP address, NetBIOS computer name, logged-in user name and MAC address (such as Ethernet). |
| **Source** | <https://www.kali.org/tools/nbtscan/>  <https://salsa.debian.org/pkg-security-team/nbtscan> |

|  |  |
| --- | --- |
| **Nikto** | |
| **Description** | Nikto is a web server scanner which performs comprehensive tests against web servers for multiple items, including potentially dangerous files/programs, checks for outdated versions of servers, and version specific problems. |
| **Source** | <https://www.kali.org/tools/nikto/>  <https://github.com/sullo/nikto> |

|  |  |
| --- | --- |
| **Onesixtyone** | |
| **Description** | Onesixtyone is a simple SNMP scanner which sends SNMP requests for the sysDescr value asynchronously with user-adjustable sending times and then logs the responses which gives the description of the software running on the device. |
| **Source** | <https://www.kali.org/tools/onesixtyone/>  <https://github.com/trailofbits/onesixtyone> |

|  |  |
| --- | --- |
| **Sharphound** | |
| **Description** | SharpHound is the official data collector for BloodHound. It is written in C# and uses native Windows API functions and LDAP namespace functions to collect data from domain controllers and domain-joined Windows systems. |
| **Source** | <https://github.com/BloodHoundAD/SharpHound> |

|  |  |
| --- | --- |
| **Smbmap** | |
| **Description** | SMBMap allows users to enumerate samba share drives across an entire domain. List share drives, drive permissions, share contents, upload/download functionality, file name auto-download pattern matching, and even execute remote commands. |
| **Source** | <https://www.kali.org/tools/smbmap/>  <https://github.com/ShawnDEvans/smbmap> |

|  |  |
| --- | --- |
| **Smtp-user-enum** | |
| **Description** | Smtp-user-enum is a username guessing tool primarily for use against the default Solaris SMTP service. |
| **Source** | <https://www.kali.org/tools/smtp-user-enum/> |

|  |  |
| --- | --- |
| **Sslscan** | |
| **Description** | SSLScan queries SSL services, such as HTTPS, in order to determine the ciphers that are supported. SSLScan is designed to be easy, lean and fast. The output includes preferred ciphers of the SSL service, the certificate and is in text and XML formats. |
| **Source** | <https://www.kali.org/tools/sslscan/>  <https://github.com/rbsec/sslscan> |

|  |  |
| --- | --- |
| **Wappalyzer** | |
| **Description** | Wappalyzer is an extension for browsers which allows you to graphically and simply visualize the technologies that are using an individual web page you visit, from the programming language used on the client and server side, to detect the CMS and more. |
| **Source** | <https://www.wappalyzer.com/> |

## Exploitation Tools

|  |  |
| --- | --- |
| **Burp Suite** | |
| **Description** | Burp Suite is a tool for testing web applications for security vulnerabilities through inspection and manipulation of their requests. |
| **Source** | <https://portswigger.net/burp> |

|  |  |
| --- | --- |
| **Certify / Certipy** | |
| **Description** | Certify/Certipy is an offensive tool for enumerating and abusing Active Directory Certificate Services (AD CS). |
| **Source** | <https://github.com/r3motecontrol/Ghostpack-CompiledBinaries>  <https://www.kali.org/tools/certipy-ad/> |

|  |  |
| --- | --- |
| **Kerbrute** | |
| **Description** | Kerbrute is an open-source tool to quickly bruteforce and enumerate valid Active Directory accounts through Kerberos Pre-Authentication |
| **Source** | <https://github.com/ropnop/kerbrute> |

|  |  |
| --- | --- |
| **Netexec** | |
| **Description** | Netexec formerly known as CrackMapExec is a network service exploitation tool that helps automate assessing the security of large networks. |
| **Source** | <https://github.com/Pennyw0rth/NetExec> |

|  |  |
| --- | --- |
| **Rubeus** | |
| **Description** | Rubeus is a C# toolset for raw Kerberos interaction and abuses. |
| **Source** | <https://github.com/r3motecontrol/Ghostpack-CompiledBinaries>  <https://github.com/GhostPack/Rubeus> |

|  |  |
| --- | --- |
| **Sqlmap** | |
| **Description** | sqlmap is an open-source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers. |
| **Source** | <https://www.kali.org/tools/sqlmap/>  <https://github.com/sqlmapproject/sqlmap> |

## Post-Exploitation Tools

|  |  |
| --- | --- |
| **Hashcat** | |
| **Description** | Hashcat supports five unique modes of attack for over 300 highly-optimized hashing algorithms. |
| **Source** | <https://www.kali.org/tools/hashcat/>  <https://hashcat.net/hashcat/> |

|  |  |
| --- | --- |
| **John The Ripper** | |
| **Description** | John the Ripper is a tool designed to help systems administrators to find weak (easy to guess or crack through brute force) passwords, and even automatically mail users warning them about it, if it is desired. |
| **Source** | <https://www.kali.org/tools/john/>  <https://github.com/openwall/john> |

|  |  |
| --- | --- |
| **Mimikatz** | |
| **Description** | Mimikatz uses admin rights on Windows to display passwords of currently logged in users in plaintext. |
| **Source** | <https://www.kali.org/tools/mimikatz/> |

|  |  |
| --- | --- |
| **Peass-ng** | |
| **Description** | Privilege Escalation Awesome Scripts SUITE. These scripts assist a penetration tester is identifying potential privilege escalation vulnerabilities as a part of their post-exploitation procedures. |
| **Source** | <https://www.kali.org/tools/peass-ng/>  <https://github.com/carlospolop/PEASS-ng> |

|  |  |
| --- | --- |
| **Sliver** | |
| **Description** | Sliver is an open source cross-platform adversary emulation/red team framework, it can be used by organizations of all sizes to perform security testing. Sliver's implants support C2 over Mutual TLS (mTLS), WireGuard, HTTP(S), and DNS. |
| **Source** | <https://github.com/BishopFox/sliver> |

|  |  |
| --- | --- |
| **XRDP** | |
| **Description** | XRDP provides a graphical login to remote machines using Microsoft Remote Desktop Protocol (RDP). |
| **Source** | <https://github.com/neutrinolabs/xrdp> |

# Appendix F: OSINT Assessment

For the assessment of {{ client.short\_name }} , {{ company.name }} engaged in OSINT (Open-Source Intelligence) prior to the start of the engagement. The primary objective of this Open-Source Intelligence initiative was to comprehensively understand {{ client.short\_name }} 's digital footprint, potential online vulnerabilities, and any publicly available data that could impact the engagement strategy. By seeking information from various publicly accessible sources such as social media, public websites, and public storefronts, {{ company.name }} sought to identify potential risks and uncover any relevant trends that might influence {{ client.short\_name }} 's security posture. This proactive approach ensured that the engagement was well-informed, allowing for strategic decision-making and a comprehensive understanding of the external factors that could positively impact {{ client.short\_name }} during the assessment taking place on 10/18/2025. The following information is the complete findings of everything found during {{ company.name }}'s OSINT engagement.

An overview of the OSINT artifacts identified by {{ company.name }} are as follows along with a section detailing each finding.

* XXX.XXX
* XXX.XXX
* XXX.XXX

## OSINT Findings

|  |  |  |
| --- | --- | --- |
|  | **OSINT-I-01** | Finding 1 |

Technical Description

Business Impact Description

Source

Finding Evidence

Mitigations

**END OF FINDING BLOCK**

|  |  |  |
| --- | --- | --- |
|  | **OSINT-I-02** | Finding 2 |

Technical Description

Business Impact Description

Source

Finding Evidence

|  |  |
| --- | --- |
|  |  |

Mitigations

**END OF FINDING BLOCK**

# Appendix G: Phishing Assessment

Spear Phishing

<INSERT\_PHISHING\_DESCRIPTION>

# Appendix H: Assessment Artifacts

During our assessments, we occasionally need to create objects such as files or users on a device, referred to as artifacts, to test potential vulnerabilities or gain additional access to a device or network. While our primary goal is to remove these artifacts after the assessment, there are instances where they may persist due to certain limitations in deletion. To mitigate any unintended consequences, we adhere to a standardized naming convention, using "{{ client.short\_name }} -[Artifact Number].[File Extension]", not only to provide easy identification of these artifacts but also to maintain transparency and accountability, aiming to minimize any lingering impact on the assessed systems. A table of all artifacts created during the assessment is as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Hostname** | **IP** | **Artifact Name** | **Creation Time** | **File Path** | **Deleted?** |
|  |  |  |  |  | Yes/No |
|  |  |  |  |  | Yes/No |
|  |  |  |  |  | Yes/No |
|  |  |  |  |  | Yes/No |
|  |  |  |  |  | Yes/No |
|  |  |  |  |  | Yes/No |
|  |  |  |  |  | Yes/No |
|  |  |  |  |  | Yes/No |
|  |  |  |  |  | Yes/No |

1. <https://docs-prv.pcisecuritystandards.org/PCI%20DSS/Supporting%20Document/PCI-DSS-v4-0-At-A-Glance.pdf> [↑](#footnote-ref-2)
2. <https://docs-prv.pcisecuritystandards.org/PCI%20DSS/Standard/PCI-DSS-v4_0.pdf> [↑](#footnote-ref-3)
3. <https://docs-prv.pcisecuritystandards.org/PCI%20DSS/Supporting%20Document/Prioritized-Approach-For-PCI-DSS-v4-0.pdf> [↑](#footnote-ref-4)
4. <http://www.pentest-standard.org/index.php/Main_Page> [↑](#footnote-ref-5)
5. <https://www.first.org/cvss/v4.0/specification-document> [↑](#footnote-ref-6)
6. <http://www.pentest-standard.org/index.php/Main_Page> [↑](#footnote-ref-7)
7. <https://owasp.org/www-project-top-ten/> [↑](#footnote-ref-8)