%COMPANY LOGO%

Report of investigation into “….”  
%KLANTNAAM / LOGO%

=====================================

Version: 1.0  
  
By: %COMPANYNAME% – info@%companyname%.nl  
Date: 03 december 2020  
TLP: [**ORANGE**] – Business Confidential  
=====================================

All rights reserved to %COMPANYNAME%. No part of this publication, in whole or in part, may be reproduced, copied, transferred or any other right reserved to its copyright owner, including photocopying and all other copying, any transfer or transmission using any network or other means of communication, any broadcast for distant learning, in any form or by any means such as any information storage, transmission or retrieval system, without prior written permission from %COMPANYNAME%.

# Table of Contents

[Table of Contents 2](#_Toc58876564)

[Confidentiality Statement 3](#_Toc58876565)

[Disclaimer 3](#_Toc58876566)

[Contact Information 4](#_Toc58876567)

[Involved Parties 4](#_Toc58876568)

[Assessment Overview 5](#_Toc58876569)

[Assessment Background 6](#_Toc58876570)

[Assessment Components 7](#_Toc58876571)

[Internal Penetration Test 7](#_Toc58876572)

[House Cleaning 7](#_Toc58876573)

[Finding Severity Ratings 8](#_Toc58876574)

[Timeline of events 9](#_Toc58876575)

[Scope 10](#_Toc58876576)

[Scope Exclusions 10](#_Toc58876577)

[Client Allowances 10](#_Toc58876578)

[Executive Summary 11](#_Toc58876579)

[Attack Summary 12](#_Toc58876580)

[Recommendations 12](#_Toc58876581)

[Vulnerability Overview 13](#_Toc58876582)

[Technical Report 14](#_Toc58876583)

[System X.X.X.X – %HOSTNAME% 15](#_Toc58876584)

[Pentest / Exploit Development step-by-step: 15](#_Toc58876585)

[Setting the basics: 15](#_Toc58876586)

[Findings and remediation 16](#_Toc58876587)

[MySQL (MariaDB) SQLi Injection Vulnerability (Error-Based) 16](#_Toc58876588)

[Using default credentials 17](#_Toc58876589)

[Exploitation | Proof of Concept 18](#_Toc58876590)

[Additional Items 19](#_Toc58876591)

[Appendix 1 – ???: 19](#_Toc58876592)

# Confidentiality Statement

This document is the exclusive property of %KLANTNAAM% and %COMPANYNAME%. This document contains proprietary and confidential information. Duplication, redistribution, or use, in whole or in part, in any form, requires consent of both %KLANTNAAM% and %COMPANYNAME%.

%KLANTNAAM% may share this document with auditors under non-disclosure agreements to demonstrate penetration test requirement compliance.

# Disclaimer

A penetration test is considered a snapshot in time. The findings and recommendations reflect the information gathered during the assessment and not any changes or modifications made outside of that period.

Time-limited engagements do not allow for a full evaluation of all security controls. %KLANTNAAM% prioritized the assessment to identify the weakest security controls an attacker would exploit. %COMPANYNAME% recommends conducting similar assessments on an annual basis by internal or third-party assessors to ensure the continued success of the controls.

# Contact Information

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

|  |  |  |
| --- | --- | --- |
| Name | Title | Contact Information |
| **%KLANTNAAM%** | | |
| John Doe | VP, Information Security (CISO) | Office: (555) 555-5555  Email: [john.doe@demo.com](mailto:john.doe@demo.com) |
| Jim Doe | IT Manager | Office: (555) 555-5555  Email: [jim.doe@demo.com](mailto:jim.doe@demo.com) |
| Joe Doe | Network Engineer | Office: (555) 555-5555  Email: [joe.doe@demo.com](mailto:joe.doe@demo.com) |
| **%COMPANYNAME%** | | |
| Jarno X | Lead Penetration Tester | Office: (0031)6-00000000  Email: info@%companmyname%.nl |
|  |  |  |
|  |  |  |
|  |  |  |

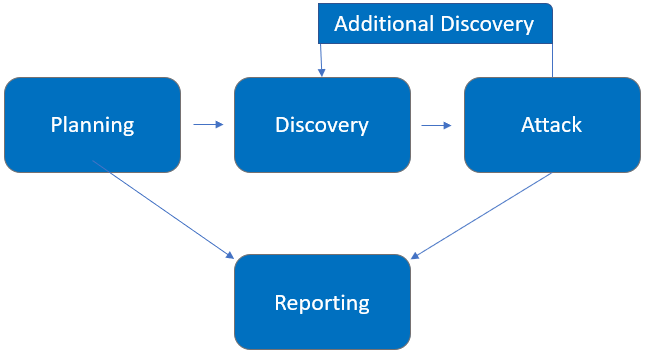
# Involved Parties

|  |  |  |
| --- | --- | --- |
| Name | Role | Contact Information |
| Party 1 | Supplier…. | Office: (555) 555-5555  Email: [john.doe@demo.com](mailto:john.doe@demo.com) |
| Party 2 | Supplier…. | Office: (555) 555-5555  Email: [jim.doe@demo.com](mailto:jim.doe@demo.com) |
| Party 3 | Supplier…. | Office: (555) 555-5555  Email: [joe.doe@demo.com](mailto:joe.doe@demo.com) |

# Assessment Overview

From December the 1st, 2020 and December the 2nd, 2020, %KLANTNAAM% engaged %COMPANYNAME% to evaluate the security posture of its Network that included an internal penetration test. %COMPANYNAME% is tasked with following methodical approach in obtaining access to the objective goals. This penetration test should simulate an actual penetration test from beginning to end, including the overall report. The following high-overview steps are taking during the labs and the exam.

* Planning – Analyze the network and plan the required steps.
* Discovery – Perform scanning and enumeration to identify potential vulnerabilities, weak areas, and exploits.
* Attack – Confirm potential vulnerabilities through exploitation and perform additional discovery upon new access.
* Reporting – Document all found vulnerabilities and exploits in a management report (high-level overview) and detailed, technical report.



# Assessment Background

On 01-01-2020 %KLANTNAAM% was compromised. %COMPANYNAME% was asked to analyze the events leading up to the compromise and the publicly disclosure of 200.000 customer records.

# Assessment Components

## Internal Penetration Test

A internal penetration test emulates the role of an attacker attempting to gain access to the information and devices located on the internal network without inside knowledge. We perform this test from inside the internal network as if the hacker already found a foothold on a inside, low privilege machine. %COMPANYNAME% attempts to gather sensitive information and access to internal machines. During the internal penetration test we perform scanning and enumeration to identify potential vulnerabilities. We will also exploit found vulnerabilities to verify the vulnerability. All steps to reproduce and evaluate found vulnerabilities will be explained in this report.

## House Cleaning

The house-cleaning portion of the assessment ensures that remnants of the penetration test are removed. Often times, fragments of tools or user accounts are left on an organization’s computer, which can cause security issues down the road. Ensuring that we are meticulous and no remnants of our penetration test are left over is paramount importance. After the objectives were successfully completed, %COMPANYNAME% removed all user accounts and passwords as well as the Meterpreter services installed on the system. %KLANTNAAM% should not have to remove any user accounts or services from any of the systems.

# Finding Severity Ratings

The following table defines levels of severity and corresponding CVSS score range that are used throughout the document to assess vulnerability and risk impact.

Medium, Low and Informational vulnerabilities are not reported in this document because we focus mainly on the high and critical vulnerabilities with serious implications. This is not a vulnerability scan report.

| Severity | CVSS V3 Score Range | Definition |
| --- | --- | --- |
| Critical | 9.0-10.0 | Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately. |
| High | 7.0-8.9 | Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible. |
| Moderate | 4.0-6.9 | Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved. |
| Low | 0.1-3.9 | Vulnerabilities are non-exploitable but would reduce an organization’s attack surface. It is advised to form a plan of action and patch during the next maintenance window. |
| Informational | N/A | No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation. |

# Timeline of events

The timeline below shows the most relevant events that occurred before and after the breach.

|  |  |
| --- | --- |
| Date | Description |
| 01-01-2020 14:00 | x.x.x.x |
| 01-01-2020 15:00 | x.x.x.x |
| 01-01-2020 16:00 | x.x.x.x |

# Scope

|  |  |
| --- | --- |
| Assessment | Details |
| Internal Network | x.x.x.x |

## Scope Exclusions

%COMPANYNAME% did not perform any “Denial of Service” attacks or “Auto Exploitation” attacks during testing.

## Client Allowances

%KLANTNAAM% did not provide any allowances to assist the testing.

# Executive Summary

%COMPANYNAME% evaluated %KLANTNAAM% internal security posture through an internal network penetration test on the 2th and 3th of December 2020. The overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to %KLANTNAAM%. By leveraging a series of attacks, %COMPANYNAME% found critical level vulnerabilities that could result in full internal network access into the %KLANTNAAM% network. It is highly recommended that %KLANTNAAM% addresses these vulnerabilities as soon as possible as the vulnerabilities are easily found through basic reconnaissance and exploit techniques.

## Attack Summary

The following table describes a high-over summary on how %COMPANYNAME% gained high privileged access to 4 internal systems:

|  |  |  |
| --- | --- | --- |
| System | Action | Recommendation |
| x.x.x.x | Action 1 | Recommendation 1 |
| x.x.x.x | Action 2 | Recommendation 2 |
| x.x.x.x | Action 3 | Recommendation 3 |
| x.x.x.x | Action 4 | Recommendation 4 |

The following table describes a high-over summary on how %COMPANYNAME% found vulnerabilities on a system (but was not able to get high-privilege access to):

|  |  |  |
| --- | --- | --- |
| x.x.x.x | Action 5 | Recommendation 5 |

### Recommendations

%COMPANYNAME% recommends patching the vulnerabilities identified during the penetration test to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that systems and applications require frequent patching and once patched, should remain on a regular patch program in order to mitigate additional vulnerabilities that may be discovered at a later date. Detailed (step-by-step) exploitation and mitigation is discussed in the “Technical “Report.

## Vulnerability Overview

The vulnerability overview shows all vulnerabilities found and shown in the technical report but categorized in a simple overview. This overview shows the number of vulnerabilities, the severity and the systems that where identified with the vulnerability.

**Critical :**

|  |  |  |
| --- | --- | --- |
| **ID** | **Vulnerability** | **Affected Systems** |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |

**High :**

|  |  |  |
| --- | --- | --- |
| **ID** | **Vulnerability** | **Affected Systems** |
| 1 |  |  |
| 2 |  |  |

# Technical Report

The technical report describes in detail the vulnerabilities that were found on the systems [in scope](#_Scope). The report also describes the details we found during information gathering. The report is created in a chronological order, meaning we describe the exploited systems in order by first-tested until the last system we tested. In this way it becomes clear by which path we escalated privileges along the way. We also note all enumeration information and found vulnerabilities on a "by system" view. By applying this view, all information about a machine is concentrated at one location. That makes it easier to search within this report.

## System X.X.X.X – %HOSTNAME%

Pentest / Exploit Development step-by-step:  
  
Brief details:

|  |  |
| --- | --- |
| IP: | X.X.X.X |
| Hostname: |  |
| Services / Ports of interest: |  |

#### Setting the basics:

Let’s setup my working directories:  
  
I will set the variables for later use:   
  
**$** export IP=x.x.x.x

Now I will create separate scan, payload and loot directories:  
  
**$** mkdir -p /home/user/Documents/%KLANTNAAM%/Exam/$IP/{Scans,Loot,Payloads}  
  
Enumeration:

Let’s first start with a nmap scan:  
  
**$** sudo nmap -Pn -sC -sV -v -p- -oA /home/user/Documents/%KLANTNAAM%/X.X.X.X/Scans/nmap-initial-X-X-X-X X.X.X.X  
  
The results:

=======================================================

=======================================================  
  
There are a few open ports. Etc….

### Findings and remediation

#### MySQL (MariaDB) SQLi Injection Vulnerability (Error-Based)

|  |  |
| --- | --- |
| Description: | A SQL injection attack consists of insertion or “injection” of a SQL query through the input data from the client to the application. A successful SQL injection exploit can bypass login mechanisms, read sensitive data from the database, modify database data (Insert/Update/Delete), execute administration operations on the database (such as shutdown the DBMS), recover the content of a given file present on the DBMS file system and in some cases issue commands to the operating system. SQL injection attacks are a type of injection attack, in which SQL commands are injected into data-plane input in order to effect the execution of predefined SQL commands. In this case we’ve identified an SQL Insert Error Based SQL Injection. |
| CVE: |  |
| Impact: | **Critical** |
| External references: | * <https://owasp.org/www-community/attacks/SQL_Injection> * <https://www.acunetix.com/websitesecurity/sql-injection/> * <https://hydrasky.com/network-security/mssql-server-injection-tutorial/> |
| Remediation | The only structural way to prevent SQL Injection attacks is input validation and parametrized queries including prepared statements. The application code should never use the input directly. The developer must sanitize all input, not only web form inputs such as login forms. They must remove potential malicious code elements such as single quotes. It is also a good idea to turn off the visibility of database errors on your production sites. Database errors can be used with SQL Injection to gain information about your database.  If you are unable to secure the code and sanitize all data input it is also possible to use a web application firewall (WAF) to sanitize your input temporarily. |

#### Using default credentials

|  |  |
| --- | --- |
| Description: | Applicatrion is currently secured by default or at least very simple credentials. The default username (admin) is used with an easy to guess or brute-force password. This means an unauthorized user can easily login and change settings or use the Application system as a stepping stone to gain more privileges. |
| CVE: |  |
| Impact: | **High** |
| External references: | * <https://en.wikipedia.org/wiki/Default_Credential_vulnerability> * <https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/04-Authentication_Testing/02-Testing_for_Default_Credentials> |
| Remediation | Make sure to harden your accounts. This means disabling or deleting default credentials, apply a password policy with strong passwords and if possible implement 2-factor authentication. |

#### 

### Exploitation | Proof of Concept

To proof above vulnerabilities exist and we are able to exploit them we will share our screenshots to prove our elevated rights on the machine and the user data we retrieved.

*-none-*

# Additional Items

## Appendix 1 – ???:

**%LOGO%**

- Last Page -  
End of report