# FINAL REPORT

## Penetration testing
## XYZ-Group

# Table of Contents

# 1. Introduction

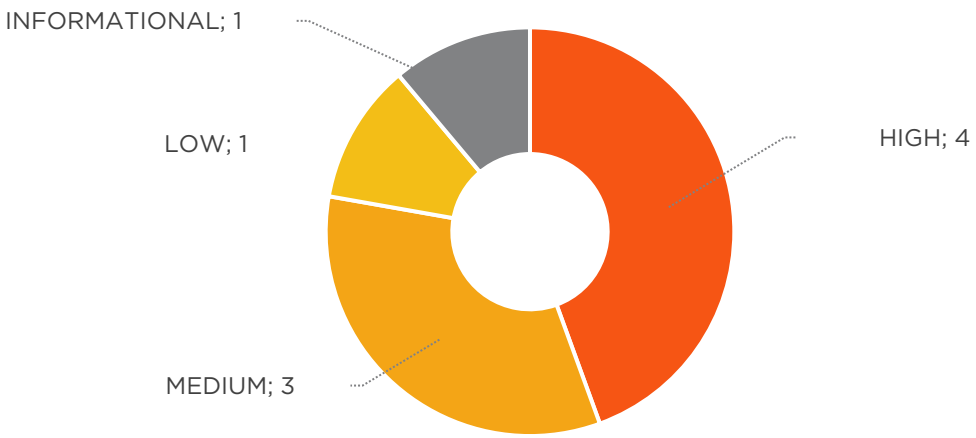
The works were carried out under this Agreement between XYZ Company (hereinafter - the “Client”) and Active Audit Agency, LLC (hereinafter - the “Contractor”).

This document describes the penetration test results for XYZ-Group, conducted by the Active Audit Agency team led by Eugene Ermolaev from xx.xx.2024 to yy.yy.2024. The report provides a detailed assessment of identified cybersecurity risks and recommendations for mitigating these risks.

To evaluate the actual security of the systems, the team simulated the actions and capabilities of the following threat profiles: External Attacker and Insider with Remote Access. Additionally, we assessed the security of specific processes and systems within a 'White Box' type evaluation.

The team conducted a comprehensive information security assessment across selected vectors, using various tools and methods to identify cybersecurity deficiencies and manually check potential vulnerabilities.

As a result of this assessment, the team identified and confirmed the presence of cybersecurity risks, particularly:




We evaluated the overall state of cybersecurity as follows:

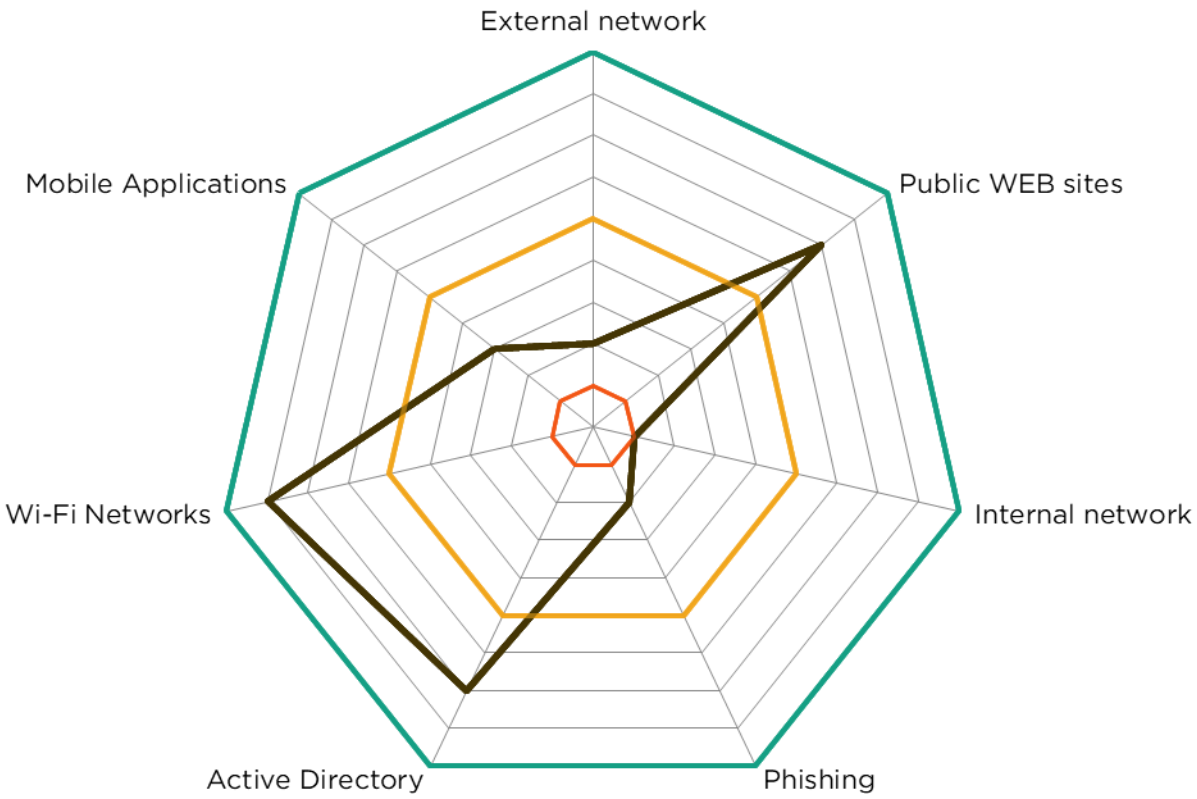

Critical vulnerabilities that an external attacker can easily exploit to cause a significant impact on the company have been identified. Immediate actions are required to prevent the possibility of a breach.

A thorough assessment of the identified risks showed that they could lead to such consequences as full system compromise, leakage of confidential data, partial system control with potential leakage of confidential data, and disruption of system availability.

The risks identified during testing indicate deficiencies in the following cybersecurity processes: vulnerability and patch management, secure system hardening, secure development practices, and cybersecurity training of employees.

## Level of information security (expert review)

| Attack vector | Protection level | Risks | | | |
|---|---|---|---|---|---|
| | | High | Medium | Low | Informational |
| External network | 3 | 1 | 1 | 0 | 0 |
| Public WEB sites | 8 | 0 | 0 | 0 | 0 |
| Internal network | 2 | 1 | 2 | 1 | 1 |
| Phishing | 3 | 1 | 0 | 0 | 0 |
| Active Directory | 8 | 0 | 0 | 0 | 0 |
| Wi-Fi Networks | 9 | 0 | 0 | 0 | 0 |
| Mobile Applications | 4 | 1 | 0 | 0 | 0 |




**Legend:** protection level gradation scale from 0 to 10, where the level:

2 — Systems are not protected. Flaw exists, that lead to full compromise of the system or its part leading to a breach of the integrity of accessibility and confidentiality. IS processes are not performed consistently.

6 — Sufficient level of security. There is no possibility of easy systems compromise. Basic IS processes are performed, but not fully. Systems can be compromised only partially, or vulnerabilities exist, that are not publicly known, or vulnerabilities exist only exploitable from complex contexts (for example, locally)

10 — IS processes are performed on an ongoing basis, the systems are reliably protected, best security practices applied.

— Current protection level

## Assessment of Simulated Threat Capabilities

As a result of the testing, several cybersecurity risks were identified, including 4 **HIGH**, 3 **MEDIUM**, 1 **LOW** and 1 **INFORMATIONAL**

The capabilities of simulated threats in the context of the identified risks are described below:

### External Attacker

Risks unique to this threat were identified, namely, 4 risks are relevant for threats, of them: 3 **HIGH**, 2 **MEDIUM**
Using vulnerabilities associated with these risks, external attacker can use the following scenarios:

- An adversary can launch phishing attacks against a majority of the company's users, given the mail server's vulnerability to such threats. It permits the identification of existing users and the forging of internal sender addresses, thus crafting an ideal environment for phishing attacks.
- The API server at https://api.somecompany.com/someapi enables attackers to execute SQL Injection attacks, resulting in the total compromise of the PROD_INTERNAL database.
- Under certain conditions, such as physical access to a mobile phone or remote access to a rooted device, an attacker can extract user cardholder data and authentication data from the application's logs. In specific scenarios, this could lead to a widespread attack on application users, significantly elevating the risk level.
- Furthermore, an intruder can remotely deactivate antivirus software on the server at 123.zz.xx.yy.
- Additionally, we have identified a deficiency in vulnerability and patch management across the public-facing systems. Although there are currently no public exploits or detailed information available for these vulnerabilities, there is a potential for exploitation in the foreseeable future.

### Insider with Remote Access

All risks relevant to the above-mentioned profile are also relevant to this profile.
However, due to the threat agent's remote access to internal company resources, additional risks unique to this threat were identified, namely:
9 risks are relevant for threats, of them: 1 **HIGH**, 1 **MEDIUM**, 1 **LOW** and 1 **INFORMATIONAL**

Using vulnerabilities associated with these risks, the Insider with remote access can use the following scenarios:

- An attacker with regular user privileges can fully compromise a segment of the company's infrastructure utilizing the LDAP service. The vulnerability allows for the administrator's password of the LDAP service to be discerned via a web interface, which is then recorded in a file accessible to any user.
- Additionally, due to the server at http://voip.somecompany.com employing an unsecured connection for authentication, in certain contexts, an attacker can compromise this service and its users.
- Furthermore, two vulnerabilities do not enable an attacker to breach the systems but allow access to some non-critical information.

# 2. Scope, limitations & threat model

A simulation of actions by a group of hackers with various commercial or personal motives was conducted during the testing. The primary objective of simulating these threats was to enhance the understanding of the system and gain logical access to the company's internal resources at the highest possible level.

## Simulated threats:

| № | Threats | Scope (attack vector) | Test mode |
|---|---|---|---|
| 1 | External Attacker | • External perimeter:<br>  ○ Defined independently in "BlackBox" mode<br>• External website:<br>  ○ https://somexternalsite.local<br>• Android Mobile App<br>• Wi-Fi network at some address 12/123<br>• Phishing is aimed at company employees | • Does not have information about the external perimeter - Black Box mode.<br>• Has access with the rights of an ordinary user to the website https://somexternalsite.local - "Gray box" mode.<br>• Is a standard user in the Android Mobile App - "Gray Box" mode.<br>• Does not have employee email distribution lists and must discover them on his/her own. |
| 2 | Insider with Remote Access | • Hosts and sites on the local network:<br>  ○ 192.168.1.0/24<br>  ○ https://someinternalsite.local<br>  ○ https://someinternalsite2.local<br>• Active Directory | • Does not have information about the internal perimeter<br>• Has access with the rights of an ordinary user to the website https://somexternalsite.local - "Gray box" mode.<br>• Is a standard employee user in the Active Directory - "Gray Box" mode. |

## Testing conditions

During the works, the following conditions and limitations were established, affecting the result:

- All transactions that may impact the functioning of critical business processes were explicitly authorized by the Client and conducted at agreed times to minimize the impact.
- The Contractor was prohibited from modifying system resources and internal files in case of compromise.
- To achieve the desired level of accuracy in vulnerability scanning, at least two vulnerability scanners were used.
- The Contractor identified the external perimeter IPs and URLs and provided a final list to the Client for approval.
- The Performer independently identified the email addresses of the Client's staff for test phishing attacks and provided a list of such addresses for Client approval.
- For access to internal network segments, the Client provided the contractor with remote routed and unhindered access using VPN remote access technology to access the Client's information systems and resources subject to internal penetration testing.
- …other testing conditions…

# 3. Security assessment methods and brief results

The table below provides information about the stages of work, the tools employed, and concise results. Detailed reports on key activities can be found through the links provided.

| № | Stages | Link to report | Brief Report |
|---|---|---|---|
| 1 | Foreign intelligence (OSINT BlackBox) | | |
| 1.1 | Corporate profile reconnaissance, open-source intelligence gathering (domain name, hostname, IP address) | Report | As an outcome of the conducted work, 3 domains, 52 IP addresses, and 13 URLs of websites associated with the IT infrastructure of the subject under investigation were identified. |
| 1.2 | Scanning of open ports (TCP, UDP), Identification of active addresses and services | Report | *…brief report or the work done…* |
| 1.3 | Reconciliation of systems under test | Report | *…brief report or the work done…* |
| 2 | Testing an external network | | |
| 2.1 | Scan for open ports (TCP, UDP), determine services and software versions for open ports, determine operating system versions | Report | *…brief report or the work done…* |
| 2.2 | Scanning for network perimeter vulnerabilities | Report | *…brief report or the work done…* |
| 2.3 | Check for network perimeter vulnerabilities | Report | *…brief report or the work done…* |
| 2.4 | Analysis of outdated versions of software services that have published vulnerabilities. | Report | *…brief report or the work done…* |
| 3 | Testing the internal network | | |
| 3.1 | Scan for open ports (TCP, UDP), determine services and software versions for open ports, determine operating system versions | Report | *…brief report or the work done…* |
| 3.2 | Scanning for network perimeter vulnerabilities | Report | *…brief report or the work done…* |
| 3.3 | Check for network perimeter vulnerabilities | Report | *…brief report or the work done…* |
| 3.4 | Analysis of outdated versions of software services that have published vulnerabilities. | Report | *…brief report or the work done…* |
| 3.5 | Testing Lan-attacks | Report | *…brief report or the work done…* |
| 4 | Web application penetration testing according to OWASP v4 | | |
| 4.1 | Gathering information | Report | *…brief report or the work done…* |
| 4.2 | Identification of entry points | Report | *…brief report or the work done…* |
| 4.3 | Information Gathering | Report | *…brief report or the work done…* |
| 4.4 | Configuration and Deployment Management Testing | Report | *…brief report or the work done…* |
| 4.5 | Identity Management Testing | Report | *…brief report or the work done…* |
| 4.6 | Authentication Testing | Report | *…brief report or the work done…* |
| 4.7 | Session Management Testing | Report | *…brief report or the work done…* |
| 4.8 | Authorization Testing | Report | *…brief report or the work done…* |
| 4.9 | Input Validation Testing | Report | *…brief report or the work done…* |
| 4.10 | Testing for Error Handling | Report | *…brief report or the work done…* |
| 4.11 | Testing for weak Cryptography | Report | *…brief report or the work done…* |
| 4.12 | Business Logic Testing | Report | *…brief report or the work done…* |
| 4.13 | Client-Side Testing | Report | *…brief report or the work done…* |
| 5 | Mobile Application Testing (Android) | | |

| № | Stages | Link to report | Brief Report |
|---|---|---|---|
| 5.1 | Testing for Weak server-side controls | Report | *…brief report or the work done…* |
| 5.2 | Testing for Insecure data storage | Report | *…brief report or the work done…* |
| 5.3 | Testing for Insufficient transport layer protection | Report | *…brief report or the work done…* |
| 5.4 | Testing for Unintended information leakage | Report | *…brief report or the work done…* |
| 5.5 | Testing for Weak Authentication and Authorization Algorithms | Report | *…brief report or the work done…* |
| 5.6 | Testing for Weak cryptography mechanisms | Report | *…brief report or the work done…* |
| 5.7 | Testing for Client-side Injections | Report | *…brief report or the work done…* |
| 5.8 | Testing for Insecure incoming calls | Report | *…brief report or the work done…* |
| 5.9 | Testing for Incorrect session management | Report | *…brief report or the work done…* |
| 5.10 | Testing for Absence of binary protection | Report | *…brief report or the work done…* |
| 6 | Phishing | | |
| 6.1 | Mail servers survey and mailbox identification | Report | The subsequent results were obtained:<br>1. The mail server permits the utilization of a falsified sender's domain.<br>2. A universal pattern for naming email accounts was identified, facilitating the creation of a database of potentially valid user accounts. This database can be exploited for dictionary attacks to ascertain valid accounts.<br>These vulnerabilities in mail servers could enable subsequent phishing and brute-force attacks. |
| 6.2 | Development of testing scenarios for social engineering methods and agreeing them with the Customer. | Report | *…brief report or the work done…* |
| 6.3 | Sending of e-mails to the Customer's employees for the purpose of obtaining data for further development of the attack on the internal resources of the Customer | Report | A phishing attack was executed, which was not recognized by all users. As a result of the attack, 10 valid logins and user passwords were obtained. |
| 6.4 | Making telephone calls to employees in order to obtain data for further development of an attack on internal resources. | Report | *…brief report or the work done…* |
| 7 | Wi-Fi network testing | | |
| 7.1 | Mapping the unimpeded coverage of wireless access points | Report | *…brief report or the work done…* |
| 7.2 | The final determination and approval of the list of names and wireless access points | Report | *…brief report or the work done…* |
| 7.3 | Scanning and conducting attacks on WI-FI network | Report | *…brief report or the work done…* |
| 8 | Verification of potential vulnerabilities | | |
| 8.1 | Verification of potential vulnerabilities. Development of scenarios and models of potential attacks by attackers based on combinations of identified vulnerabilities. Test attempts to exploit and develop attacks. | Report | As a culmination of all preceding efforts and various types of vulnerability identification, 27 unique potential vulnerabilities were identified. Each was rigorously tested for the possibility of exploitation and the potential for further attack development.<br><br>The analysis yielded the following categorization of vulnerabilities:<br>- 4 vulnerabilities classified as high severity.<br>- 3 vulnerabilities classified as medium severity.<br>- 1 vulnerability classified as low severity.<br>- 1 vulnerability classified as informational severity.<br>The remaining 18 potential vulnerabilities were deemed to be false positives. |

# 4. Risk analysis results and recommendations.

The table below contains an excerpt from the risk assessment. Detailed risk reports are provided as attachments, which can be found by the links in the first column.

| ID, Link to report | Vektor | Vulnerable service | Risk Level | Description | Recommendation |
|---|---|---|---|---|---|
| R-01 | Internal network | https://samplecomp.intranet:443/ | HIGH | The service https://samplecomp.intranet is found to leak critical information within its log files. These logs include logins and passwords for the LDAP service, as well as locations of critical backup archives. The archives, in turn, contain passwords from other services. Credentials for the LDAP, SMTP, and database accounts were obtained. Further development of the attack was not pursued to avoid harm to production services. | Restrict access to the confidential file or remove it from the website. |
| R-02 | Phishing | mail.somecompany.com:25 (TCP) | HIGH | The present configuration settings of the mail server permit the following:<br>1. Identification of valid employee email addresses.<br>2. Acceptance of email messages from recipients utilizing a fabricated sender's address, including those from the @somecompany.com domain.<br><br>These vulnerabilities enable attackers to execute an efficient "phishing attack," the repercussions of which can range from severe to critical, up to and including the complete compromise or destruction of information.<br><br>The vulnerability was confirmed by executing a phishing attack targeting 50 users. | 1. Disable the feature that informs the sender about the presence or absence of the recipient in the server database. For instance, Google's server always provides a positive response, even if the recipient does not exist.<br>2. Implement message filtering to detect forged sender addresses by verifying if the sender's IP address matches its domain. |
| R-03 | External network | https://api.somecompany.com/ | HIGH | Insufficient input filtering facilitates the execution of an SQL Injection / HQL injection attack.<br>The vulnerability was validated in the parameter "_number" of https://api.somecompany.com/someapi API call.<br>In this scenario, the injection type is HQL Boolean, which was manipulated to execute a Boolean-based Blind SQL Injection.<br>Exploiting these vulnerabilities enabled auditors to access the full content of the PROD_INTERNAL database, which contains critical information that could be leveraged by malicious entities for subsequent attacks. | Ensure the special characters provided by users in the "_number" parameter are adequately filtered. It is recommended to exclusively utilize parametric queries or stored procedures for database queries.<br>Useful links:<br>https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet |
| R-04 | Mobile applications | https://api.somecompany.com/ | HIGH | The Android mobile application is found to log critical data into system-wide logs, including customer card numbers, card validity periods, CVV2 codes, and session information (Authorization: Bearer). This information is stored in a file accessible to attackers or any application granted the READ_LOGS permission. In the event of a mobile device theft without a screen lock and with file system access (superuser rights), an attacker could gain access to customer card data. | Disable the logging of critical application data in any log files. |

| ID, Link to report | Vektor | Vulnerable service | Risk Level | Description | Recommendation |
|---|---|---|---|---|---|
| | | | | Furthermore, the storage of such data on the client side contravenes the PCI DSS standard requirements. | |
| R-05 | Internal Network | http://voip.somecompany.com:8080/ | MEDIUM | The server facilitates authentication over the non-encrypted HTTP protocol. An attacker can intercept traffic between the victim and server, extracting credentials in cleartext. This vulnerability was demonstrated by an auditor who successfully intercepted traffic and captured a test authentication attempt. | Implement HTTPS support and redirections from HTTP to HTTPS or use secure authentication schemas over HTTP |
| R-06 | External Network | 123.zz.xx.yy:3310 (TCP) | MEDIUM | ClamAV version 0.99.2 permits the execution of remote commands SCAN and SHUTDOWN without requiring authentication. The 'SCAN' command can be utilized to enumerate system files, and the 'SHUTDOWN' command can terminate the ClamAV service. This vulnerability was confirmed through test exploitation. | Update to the latest stable release (https://www.clamav.net/downloads) |
| R-07 | External Network | Multiple systems | MEDIUM | The infrastructure shows multiple instances of using outdated software, which contains security vulnerabilities. There is a lack of vulnerability and patch management within the infrastructure. Even though these vulnerabilities currently have no public exploits or detailed information available, they may be exploited in the near future. | Implement vulnerability and patch management procedures to fix all available vulnerabilities in a timely manner. |
| R-08 | Internal Network | https://sql.somecompany.com<br>https://somecompany.com/ | LOW | Accessing a specific directory via URL discloses the content of the server directory. This practice is ill-advised as the directory might contain files not typically accessible through the website's links. An attacker could view a list of all files in this directory, potentially exposing sensitive information. | Configure the web server not to respond with a list of directory content.<br>Useful links:<br>https://stackoverflow.com/questions/2530372/how-do-i-disable-directory-browsing |
| R-09 | Internal Network | https://mail.somecompany.com<br>https://somecompany.com<br>https://sql.somecompany.com | INFORMATIONAL | The HTTP TRACE method has been activated on three web services. The HTTP TRACE method, intended for diagnostic purposes, can lead to potential security risks if enabled on a web server. When this method is used, the server echoes back the exact request received in its response. Although often harmless, this feature can sometimes lead to the disclosure of sensitive information, such as internal authentication headers that are appended by reverse proxies. Historically, this functionality could be exploited to bypass the HttpOnly cookie flag, although this is no longer feasible with modern web browsers. This vulnerability has been confirmed through tests conducted by auditors. | The TRACE method should be disabled on production web servers.<br>Useful links:<br>https://github.com/nu11secur1ty/Disabling-the-TRACE-method-in-Apache2/blob/master/Apache%20Tips:%20Disable%20the%20HTTP%20TRACE%20Method.md |

# ANNEX A – RISK ANALYSIS METHODIC

The purpose of our work is not only to identify and describe vulnerabilities but also to assess the actual level of risk they can create for your business. We consider real threats in the context of the technological environment and evaluate potential impacts that could arise from exploiting vulnerabilities.

Risk assessment is performed using the 'expert assessment' method, and the risk levels are determined by the project team. If necessary, the approach to risk analysis can be adapted according to the risk analysis methodology used in your company.

We use the following criteria to determine the levels of risk:

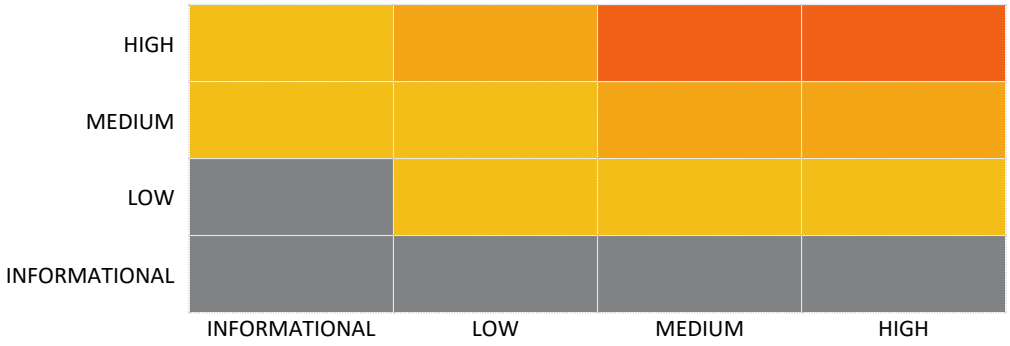
| Risk level | Description |
|---|---|
| HIGH | This level signifies that potential adversaries with a high likelihood of success could attain significant control over the system or its critical components. They might gain access to highly confidential information, manipulate critical data, initiate a sustained denial of service, or severely damage the Company's reputation. The associated risk has the potential to lead to substantial losses. |
| MEDIUM | At this level, attackers with a substantial chance of success could obtain partial access to confidential information, compromise the integrity of specific system components, induce short-term denial of service incidents, or impact the reputation of a subset of the Company's customers. This category also includes recurring low-level incidents and critical vulnerabilities with a low likelihood of exploitation. |
| LOW | The LOW-risk level indicates that attackers might acquire limited information about the system, initiate a minor denial of service events targeting specific elements, or cause other relatively insignificant impacts. |
| INFORMATIONAL | This level is designated for situations where no discernible risk is posed to the affected systems. The identified issue aligns with established best practices but does not present a direct threat. |
| **INSIGNIFICANT** | This classification is reserved for cases where the risk of impact is minimal, and its presence could be disregarded or deemed inapplicable under the current circumstances. |

Also, there were vulnerability, threats, and impact levels defined: HIGH, MEDIUM, LOW, INFORMATIONAL, **NOT APPLICABLE**

Risk assessment methodology defines the risk level by formula:

$$Risk = Fx( Fx( Vulnerability, Threat ), Impact )$$

Where Fx can be calculated from the matrix:



# SAMPLE RISK R-01 DETAILED REPORT

## RISK BRIEF

| | |
|---|---|
| ID | R-01 |
| RISK LEVEL | HIGH |
| SUMMARY | The service https://samplecomp.intranet is found to leak critical information within its log files. These logs include logins and passwords for the LDAP service, as well as locations of critical backup archives. The archives, in turn, contain passwords from other services. Credentials for the LDAP, SMTP, and database accounts were obtained. Further development of the attack was not pursued to avoid harm to production services. |
| VULNERABLE SERVICES | https://samplecomp.intranet |
| RECOMMENDATIONS | Restrict access to the confidential file or remove it from the website. |

## RISK ASSESSMENT

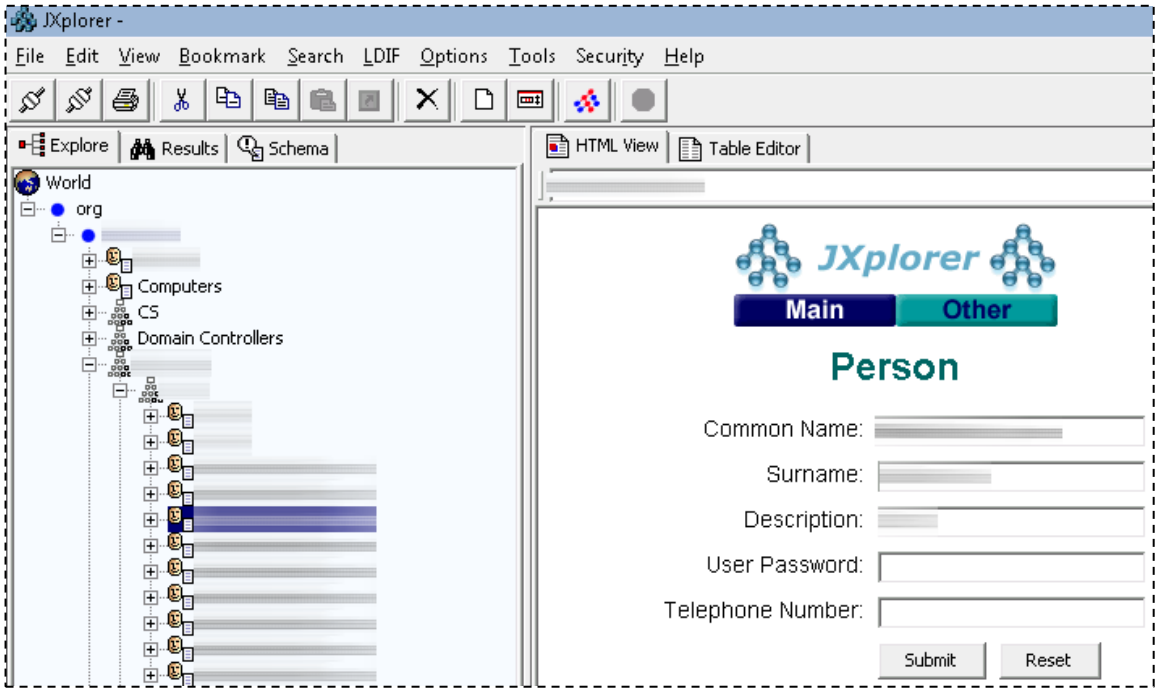
| | SEVERITY | DESCRIPTION | |
|---|---|---|---|
| VULNERABILITY | HIGH | The service https://samplecomp.intranet is found to leak critical information within its log files. | |
| THREAT | HIGH | Threat agents | Insider with Remote Access |
| | | Attack scenario | The insider obtains information from files accessible on the internal website. |
| | | Context | No special conditions are required. |
| MAXIMUM IMPACT | HIGH | LDAP service compromise | |

## VERIFICATION RESULT

### 1. https://samplecomp.intranet/log/error.log

The log file, accessible through the link https://samplecomp.intranet/log/error.log, includes portions that contain credentials for the LDAP service, highlighted in orange:

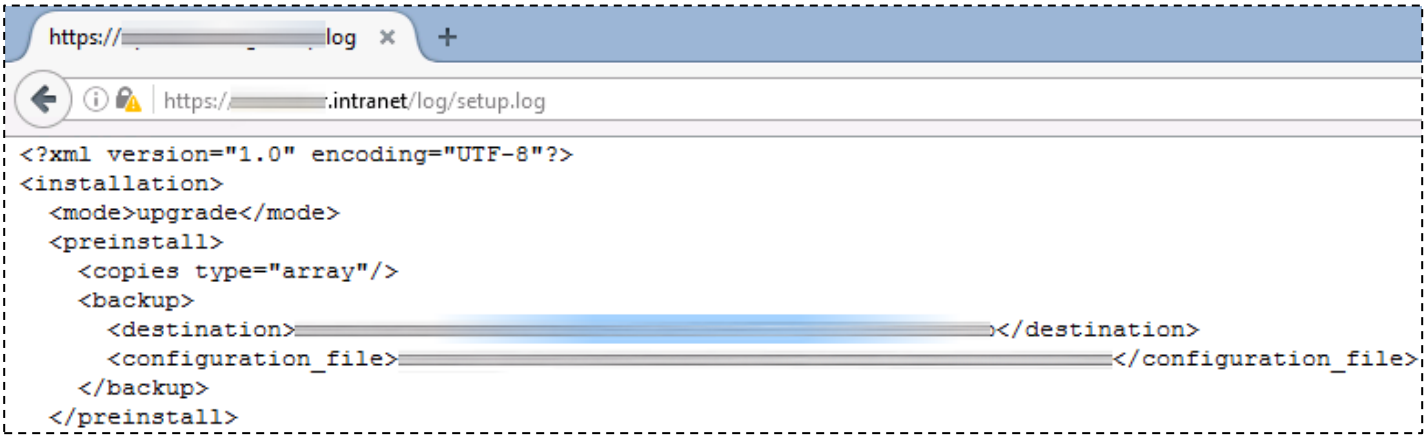
```
30 | Error | ldap_authentication: can not bind to the LDAP server 'ldaps://exc1.somecompany.org' (port:
636), user='ldap', pwd='#########'. Error: 'Can't contact LDAP server'. Check the configuration file
config-samlpe.php.
1970-01-01 11:11:11 | Error | ldap_set_option('17', '3') returned true
Debug trace:
#0 /www/example/web/application/cmdbabstract.class.inc.php(3173): DBObject->DBInsertNoReload()
#1 /www/example/web/core/dbobject.class.php(1704): cmdbAbstractObject->DBInsertNoReload()
#2 /www/example/web/application/ui.extkeywidget.class.inc.php(512): DBObject->DBInsert()
#3 /www/example/web/pages/ajax.render.php(483): UIExtKeyWidget->DoCreateObject(Object(ajax_page))
2016-06-16 10:05:45 | Warning | Email sending failed: Some recipients were invalid, aFailedRecipients
contains: sample@somecompany.org
```



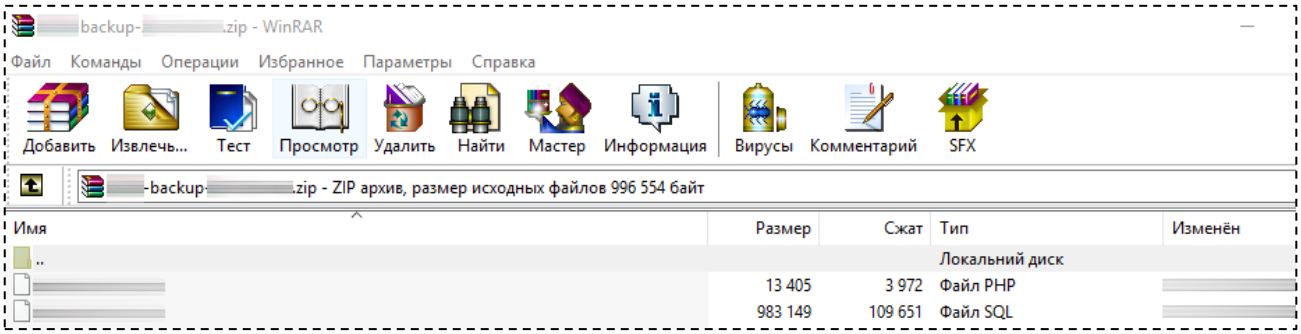

Disclosed credentials allow access to LDAP service

## 2. https://samplecomp.intranet/log/setup.log

The log file is accessible via the link https://samplecomp.intranet/log/setup.log. As evident, it includes the location of the backup archive. Here is a segment of the log file:




A copy of that archive is available in the annex "*R-01 Annex 1\backup-empty-sample*".

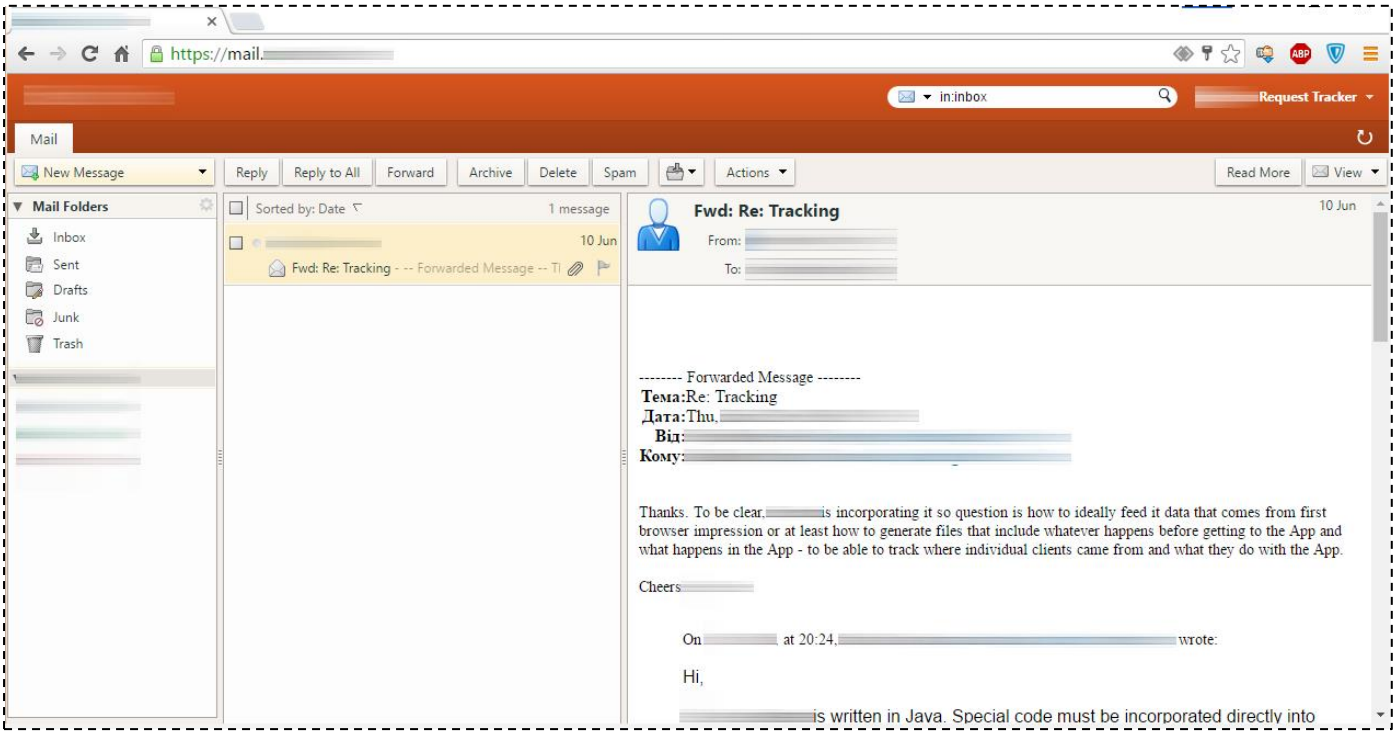



The archive contains Example service main configuration file and database dump.

```
// csv_import_history_display: Display the history tab in the im
//  default: false
'csv_import_history_display' => false,

'db_character_set' => '[redacted]',

'db_collation' => '[redacted]',

'db_host' => '[redacted]',

'db_name' => '[redacted]',

'db_pwd' => '[redacted]',

'db_subname' => '',

'db_user' => '[redacted]',
```

```
// email_transport: Mean to send emails: PHPMail (uses the function mail()) or SMTP (implements the client protocole)
//  default: 'PHPMail'
'email_transport' => '[redacted]',
'email_transport_smtp.encryption' => '[redacted]',
'email_transport_smtp.host' => '[redacted]',
'email_transport_smtp.port' => [redacted],
'email_transport_smtp.username' => '[redacted]',
'email_transport_smtp.password' => '[redacted]',

// email_validation_pattern: Regular expression to validate/detect the format of an eMail address
//  default: '[a-zA-Z0-9._&\'-]+@[a-zA-Z0-9.-]+\\.[a-zA-Z0-9-]{2,}'
'email_validation_pattern' => '[redacted]',

'encryption_key' => '[redacted]',
```

Some parts of config-samlpe.php file. The file contain passwords to other services.




The picture shows access to mailbox with leaked credentials

Further attack development was halted, thus preventing any impact on the production services.

# SAMPLE RISK R-02 DETAILED REPORT

## RISK BRIEF

| | |
|---|---|
| ID | R-02 |
| RISK LEVEL | HIGH |
| SUMMARY | The present configuration settings of the mail server permit the following:<br>1. Identification of valid employee email addresses.<br>2. Acceptance of email messages from recipients utilizing a fabricated sender's address, including those from the @somecompany.com domain.<br><br>These vulnerabilities enable attackers to execute an efficient "phishing attack," the repercussions of which can range from severe to critical, up to and including the complete compromise or destruction of information.<br><br>The vulnerability was confirmed by executing a phishing attack targeting 50 users. |
| VULNERABLE SERVICES | mail.somecompany.com:25 (TCP) |
| RECOMMENDATIONS | 1. Disable the feature that informs the sender about the presence or absence of the recipient in the server database. For instance, Google's server always provides a positive response, even if the recipient does not exist.<br>2. Implement message filtering to detect forged sender addresses by verifying if the sender's IP address matches its domain. |

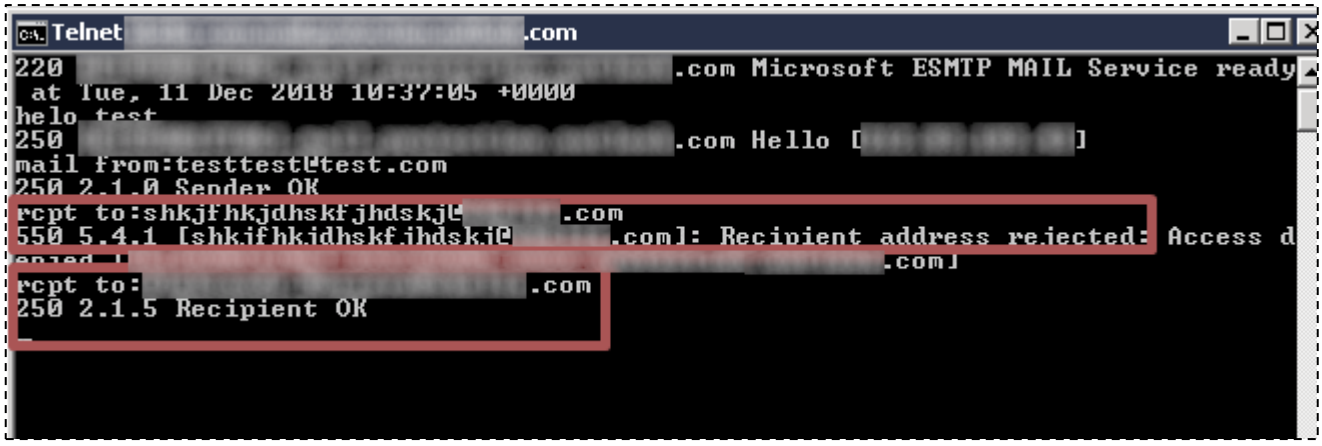
## RISK ASSESSMENT

| | SEVERITY | DESCRIPTION | |
|---|---|---|---|
| VULNERABILITY | HIGH | Identification of valid employee email addresses is possible. The company's mail server permits email messages to be sent with spoofed sender addresses, even from the @somecompany.com domain, allowing for email-based attacks. | |
| THREAT | HIGH | Threat agents | External Attacker<br>Insider with Remote Access |
| | | Attack scenario | The attacker first identifies a list of valid email addresses belonging to employees. Then, using specialized software, they craft a message with a forged sender address, such as that of an authoritative internal employee, containing a request designed to extract users' authentication data or to persuade the recipient to download malware, among other malicious objectives. |
| | | Context | No special conditions are required. |
| MAXIMUM IMPACT | MEDIUM | The attacker has acquired a list of user email addresses and may exploit it for various malicious activities such as phishing, spam, malware distribution, dictionary attacks, or others. | |

VERIFICATION RESULT

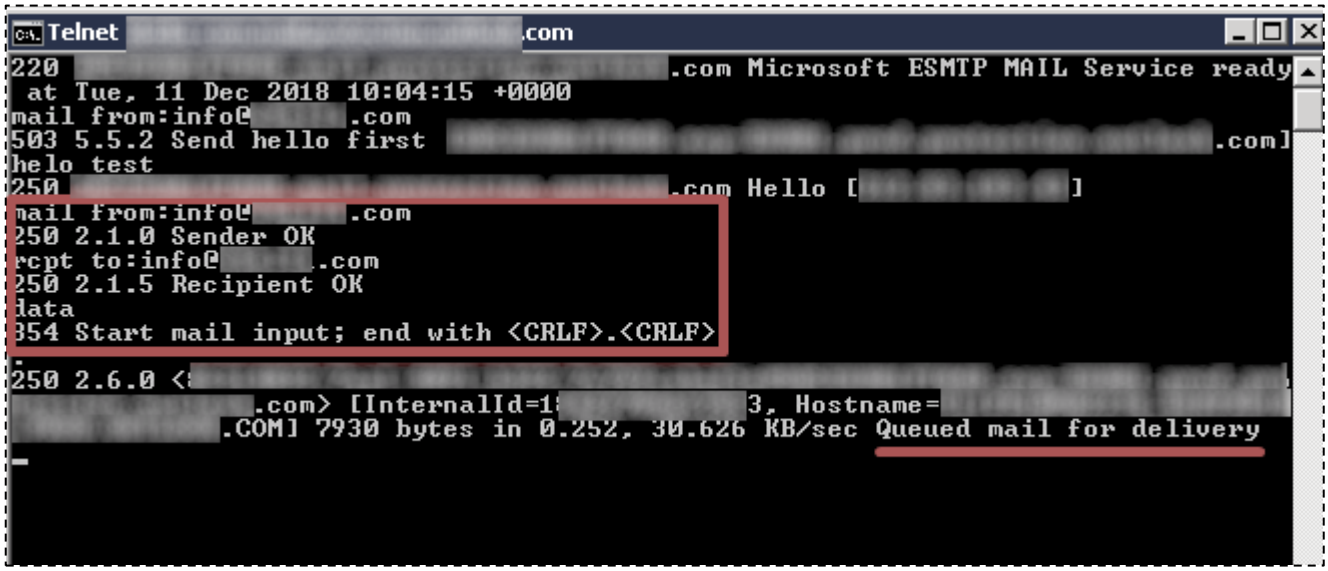
## User enumeration attacks are possible.

The mail server mail.somecompany.com is configured to disable the use of the SMTP commands VRFY, which allows checking the availability of a recipient's email account without sending an actual email, and the EXPN command, which is used for checking account information, such as a list of recipients. However, the server's response to the command "RCPT TO" for potential users of the mail domain presents an opportunity for attackers to identify valid email accounts. Here are the server responses to such attempts:




Response from the server allows user enumeration

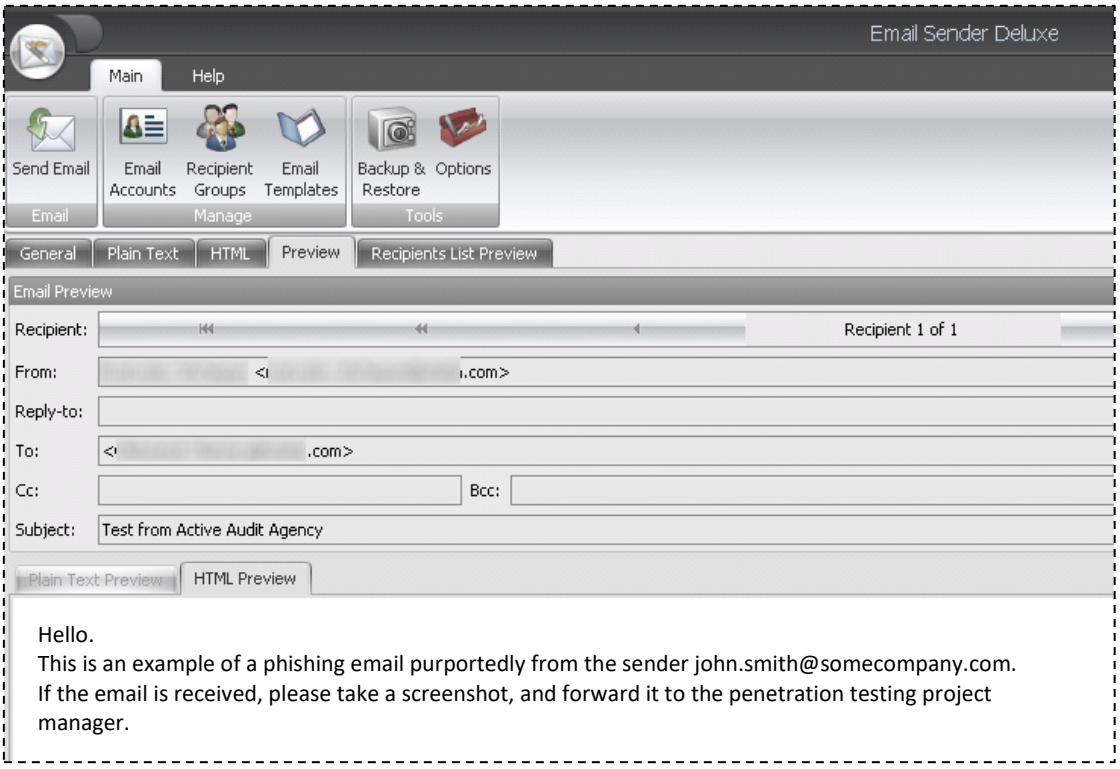
## Forging a sender's domain name.




Attempting to use the forged domain of the sender. Email sent from the fake sender

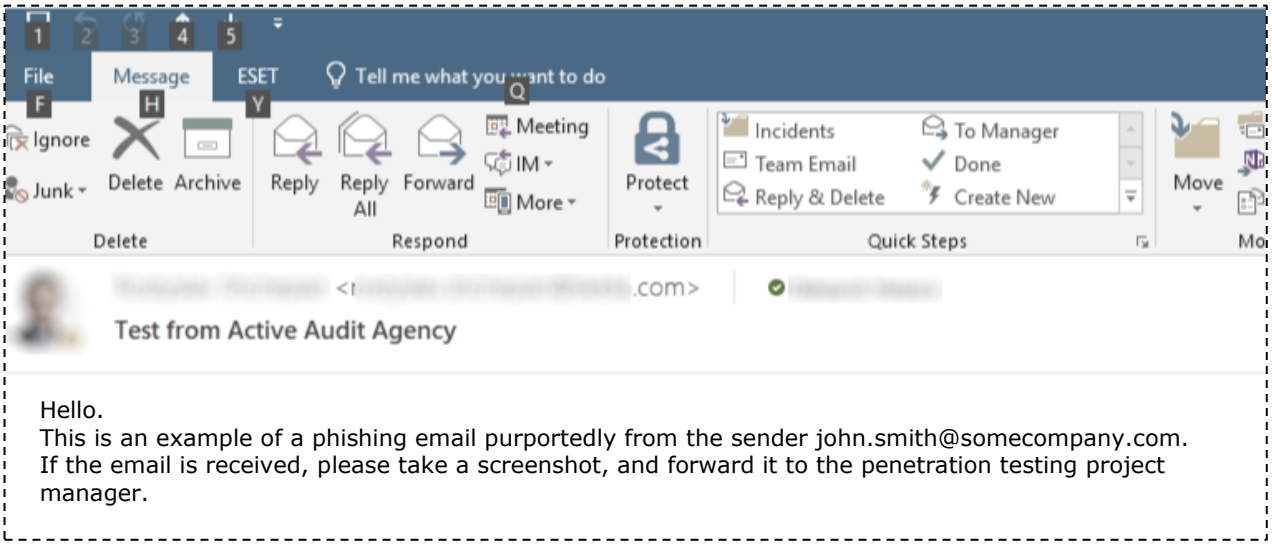
## Test phishing attack

"Email Sender Deluxe" software was used to send a fake email with spoofed sender address. The email template may have contained a link to a "Fishing site" or backdoor content.




Sending a test phishing email

The message was crafted and dispatched to the email address john.doe@somecompany.com. The mail server mail.somecompany.com received the email and forwarded it to the recipient without conducting any verifications. A screenshot from John Doe's mailbox confirms the attack's success:




Mailbox of John Doe

The mail server permits the utilization of a spoofed sender domain. However, no further advancement of the attack was pursued, thereby averting any damage to the production services.

# SAMPLE RISK R-03 DETAILED REPORT

## RISK BRIEF

| | |
|---|---|
| ID | R-03 |
| RISK LEVEL | HIGH |
| SUMMARY | Insufficient input filtering facilitates the execution of an SQL Injection / HQL injection attack.<br>The vulnerability was validated in the parameter "_number" of https://api.somecompany.com/someapi API call.<br>In this scenario, the injection type is HQL Boolean, which was manipulated to execute a Boolean-based Blind SQL Injection.<br>Exploiting these vulnerabilities enabled auditors to access the full content of the PROD_INTERNAL database, which contains critical information that could be leveraged by malicious entities for subsequent attacks. |
| VULNERABLE SERVICES | https://api.somecompany.com/ |
| RECOMMENDATIONS | Ensure the special characters provided by users in the "_number" parameter are adequately filtered. It is recommended to exclusively utilize parametric queries or stored procedures for database queries.<br>Useful links:<br>https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet |

## RISK ASSESSMENT

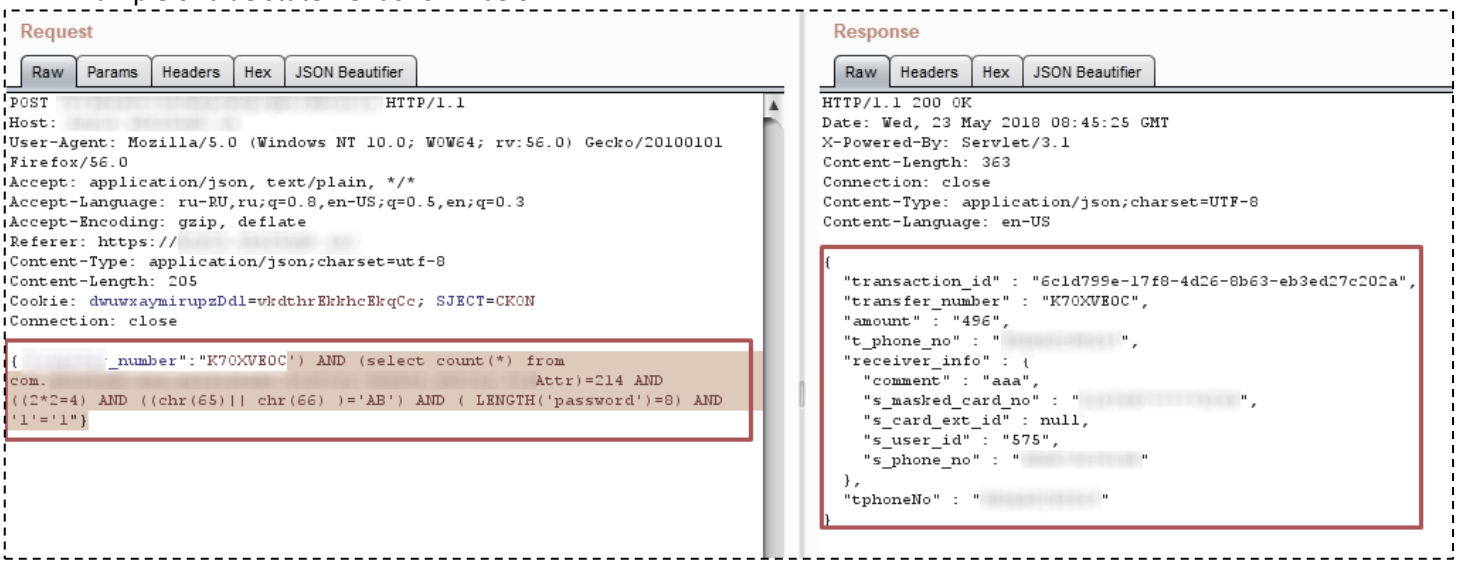
| | SEVERITY | DESCRIPTION | |
|---|---|---|---|
| VULNERABILITY | HIGH | Insufficient input filtering allows you to perform an SQL Injection \ HQL injection attack. | |
| THREAT | HIGH | Threat agents | External Attacker<br>Insider with Remote Access |
| | | Attack scenario | An attacker using a special software, using the vulnerability in the web service browses the contents of the database. |
| | | Context | No special conditions are required. |
| MAXIMUM IMPACT | HIGH | Full access to PROD_INTERNAL database | |

VERIFICATION RESULT

## Boolean-based Blind SQL Injection

The lack of input filtering in non-user-authenticated systems exposes them to a Boolean-based Blind SQL Injection attack. Attackers can exploit this vulnerability by manipulating the server's response to POST requests on the page https://somesite.com/transaction/card/card2card/. By injecting SQL code into the "_number" parameter, attackers can discern true or false conditions based on the server's responses. This allows them to extract sensitive information from the database or execute unauthorized actions. Immediate action is required to implement input validation and mitigate the risk of SQL injection attacks.

Example of true statement shown below.




Malicious SQL code with “True” response

Malicious code added:

```
') AND (select count(*) from com.somecompany.api.Attr)=214 AND ((2*2=4) AND ((chr(65)|| chr(66) )='AB') AND ( LENGTH('password')=8) AND '1'='1
```

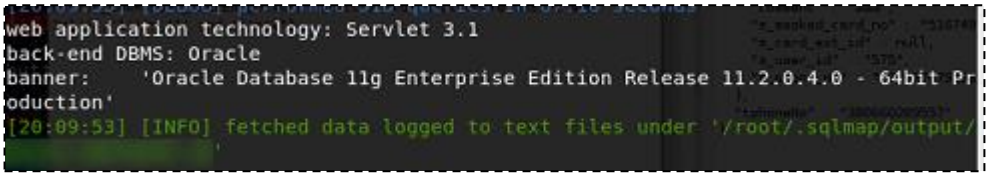
In this code, added True values, such as 2*2=4, the sum of ASCII codes 65 and 66 is **‘AB’**. And the count of records in object **com.somecompany.api.Attr equals** 214.

For the automatic vulnerability exploitation “sqlmap” was used with the following parameters:

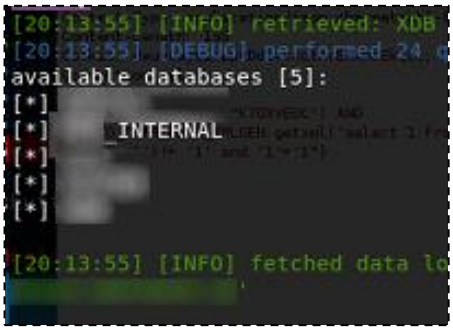
```
{"_number":"K7OXVE0C') AND NVL(TO_CHAR(DBMS_XMLGEN.getxml('select 1 from dual where 1=1*')),'1')!= '1' and '1'='1"}
```

Wildcard (*****) acts as a vulnerable parameter where sqlmap added payloads for SQL Injection exploitation SQL Injection.

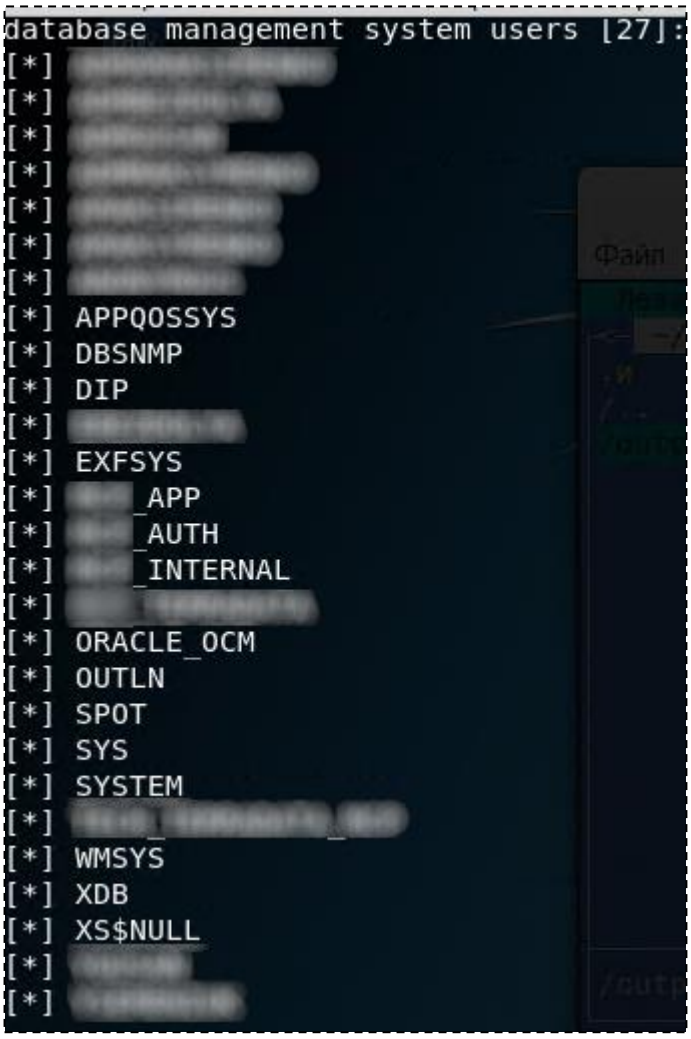
Next results were obtained:




DB version




Available databases for the current DB user




The list of the database users

The list of all tables of the **PROD_INTERNAL** and their records were obtained: Annex 1 – DB PROD_INERNAL