Logotipo

Descripción generada automáticamente

C15

Security Assessment  
Findings Report

Business Confidential

* Date of the project: 2024-12-19
* Project name: Demo for Master Thesis (TFM)

# Table of Contents

[Table of Contents 3](#_Toc143269467)

[Confidentiality Statement 4](#_Toc143269468)

[Disclaimer 4](#_Toc143269469)

[Contact Information 4](#_Toc143269470)

[Assesment overview 5](#_Toc143269471)

[Assessment Components 5](#_Toc143269472)

[Internal Penetration Test 5](#_Toc143269473)

[Finding severity ratings 6](#_Toc143269474)

[Risk factors 6](#_Toc143269475)

[Likelihood 6](#_Toc143269476)

[Impact 6](#_Toc143269477)

[Scope 7](#_Toc143269478)

[Hosts analyzed 7](#_Toc143269479)

[Scope Exclusions 7](#_Toc143269480)

[Client Allowances 7](#_Toc143269481)

[Executive summary 8](#_Toc143269482)

[Scoping and time limitations 8](#_Toc143269483)

[Testing summary 8](#_Toc143269484)

[Tester Notes and Recommendations 8](#_Toc143269485)

[Vulnerability summary & report card 9](#_Toc143269486)

[Internal Penetration Test Findings 9](#_Toc143269488)

# Confidentiality Statement

This document is the exclusive property of C15 and uit (C15).

This document contains proprietary and confidential information. Duplication, redistribution, or use, in whole or in part, in any form, requires consent of both C15 and C15).

C15 may share this document with auditors under non-disclosure agreements to demonstrate penetration test requirement compliance.

# Disclaimer

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

Time-limited engagements do not allow for a full evaluation of all security controls. C15 prioritized the assessment to identify the weakest security controls an attacker would exploit. C15 recommends conducting similar assessments on an annual basis by internal or third-party assessors to ensure the continued success of the controls.

# Contact Information

|  |  |  |
| --- | --- | --- |
| **Name** | **Title** | **Contact Information** |
| **C15** | | |
| **UIT** | Student | Email: |
| **uit** | | |
| **C15** | Penetration Tester | Email: 22520645@gm.uit.edu.vn |

# Assesment overview

From to 2024-12-19, C15 engaged uit, from now on, C15, to evaluate the security posture of its infrastructure compared to current industry best practices that included an internal network penetration test. All testing performed is based on the NIST SP 800-115 Technical Guide to Information Security Testing and Assessment, OWASP Testing Guide (v4), and customized testing frameworks.

Phases of penetration testing activities include the following:

* **Planning** – Customer goals are gathered and rules of engagement obtained.
* **Discovery** – Perform **scanning** and **enumeration** to identify potential vulnerabilities, weak areas, and exploits.
* **Attack** – Confirm potential vulnerabilities through exploitation and perform additional discovery upon new access.
* **Reporting** – **Document** all found vulnerabilities and exploits, failed attempts, and company strengths and weaknesses.

# Assessment Components

## Internal Penetration Test

An internal penetration test emulates the role of an attacker from inside the network. An engineer will scan the network to identify potential host vulnerabilities and perform common and advanced internal network attacks, such as: LLMNR/NBT-NS poisoning and other man- in-the-middle attacks, token impersonation, kerberoasting, pass-the-hash, golden ticket, and more. The engineer will seek to gain access to hosts through lateral movement, compromise domain user and admin accounts, and exfiltrate sensitive data.

# Finding severity ratings

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

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

# Risk factors

Risk is measured by two factors: Likelihood and impact.

## Likelihood

Likelihood measures the potential of a vulnerability being exploited. Ratings are given based on the difficulty of the attack, the available tools, attacker skill level, and client environment.

## Impact

Impact measures the potential vulnerability’s effect on operations, including confidentiality, integrity, and availability of client systems and/or data, reputational harm, and financial loss.

# Scope

## Hosts analyzed

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Host | Information about the host | | | |
| Ports | Service | Version | CVEs |
| 192.168.254.136 | 22 | ssh | 4.7p1 Debian 8ubuntu1 | CVE-2015-8325 CVE-2010-4478 CVE-2016-10010 CVE-2008-1657 CVE-2016-3115 CVE-2023-48795 CVE-2016-6210 CVE-2017-15906 CVE-2016-20012 CVE-2014-2532 CVE-2010-4755 CVE-2011-5000 CVE-2011-4327 CVE-2023-51385 CVE-2023-38408 CVE-2021-36368 CVE-2020-15778 CVE-2019-6111 CVE-2019-6110 CVE-2019-6109 CVE-2018-20685 CVE-2018-15473 CVE-2016-6515 CVE-2016-1908 CVE-2016-10708 CVE-2016-10012 CVE-2016-10011 CVE-2016-10009 CVE-2015-6564 CVE-2015-6563 CVE-2015-5600 CVE-2015-5352 CVE-2014-2653 CVE-2014-1692 CVE-2012-0814 CVE-2010-5107 CVE-2008-3259 |

## Scope Exclusions

Per client request, C15 did not perform any of the following attacks during testing:

* Denial of service (DoS)
* Phishing/Social Engineering

All other attacks not specified above were permitted by C15.

## Client Allowances

C15 provided C15 the following allowances:

* Internal access to network via TBD.

# Executive summary

C15 evaluated C15’s internal security posture through penetration testing from to 2024-12-19. The following sections provide a high-level overview of vulnerabilities discovered, successful and unsuccessful attempts, and strengths and weaknesses.

Executive Summary:  
  
1. Description of Found IP and Ports:  
 - IP 192.168.254.136:  
 - Open Ports: 22 (SSH)  
 - Vulnerabilities:  
 - CVE-2015-8325 (High): This vulnerability allows local users to gain privileges by triggering a crafted environment for the /bin/login program. It poses a significant risk as it can lead to privilege escalation.  
 - CVE-2010-4478 (High): This vulnerability allows remote attackers to bypass authentication mechanisms and successfully authenticate by sending crafted values in the J-PAKE protocol.  
 - CVE-2016-10010 (High): This vulnerability allows local users to gain privileges by creating forwarded Unix-domain sockets as root. It poses a risk of privilege escalation.  
 - CVE-2008-1657 (Medium): This vulnerability allows remote authenticated users to bypass the sshd\_config ForceCommand directive. It poses a risk of unauthorized access.  
 - CVE-2016-3115 (Medium): This vulnerability allows remote authenticated users to bypass intended shell-command restrictions. It poses a risk of unauthorized access.  
  
2. Health Posture of SSH Service:  
 - The SSH service on port 22 has multiple high and medium severity vulnerabilities that can lead to privilege escalation and unauthorized access. It is crucial to address these vulnerabilities promptly to secure the SSH service.  
  
3. Mitigation Strategies:  
 - CVE-2015-8325, CVE-2010-4478, CVE-2016-10010: Apply patches provided by the vendor to address these high-severity vulnerabilities. Implement strict access controls and monitor user privileges to prevent unauthorized access.  
 - CVE-2008-1657, CVE-2016-3115: Enforce strong authentication mechanisms and regularly review and update access controls to mitigate the risks associated with these medium-severity vulnerabilities.  
  
Additional Information:  
- Regularly updating the OpenSSH software to the latest version can help mitigate known vulnerabilities.  
- Implementing multi-factor authentication and least privilege access can enhance the security of the SSH service.  
- Conducting regular security audits and penetration testing can help identify and address security weaknesses in the SSH service.  
  
Addressing the identified vulnerabilities and implementing the recommended mitigation strategies are essential to improve the security posture of the organization and protect against potential security threats.

## Scoping and time limitations

Scoping during the engagement did not permit denial of service or social engineering across all testing components.

Time limitations were in place for testing. Internal network penetration testing was permitted for days.

## Testing summary

The network assessment evaluated C15’s internal security posture. From an internal perspective, the C15 performed vulnerability scanning against the IP addresses provided by C15 to evaluate the overall patching health of the network.

TBD

## Tester Notes and Recommendations

TBD overall security (good/regular/bad).

TBD constants that stood out in the process

TBD reccomendations.

We recommend that the C15 team reviews the patching recommendations made in the TBD section of the report along with reviewing the provided scans for a full overview of the items to be patched. We also recommend that Demo corp improve their patch management policies and procedures to help prevent potential attacks within their network.

TBD alerts triggered.

Overall, the C15 network performed as expected for the penetration test. We recommend that the C15 team thoroughly review the recommendations made in this report, patch the findings, and re-test annually to improve their overall security posture.

# Vulnerability summary & report card

The following tables illustrate the vulnerabilities found by impact and recommended remediations:

## Internal Penetration Test Findings

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 10 | 19 |  | 0 |
| **Critical** | **High** | **Moderate** | **Low** | **Informational** |

### Finding 1

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2015-8325**
* Severity score: **7.8** (**HIGH** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: Privilege Escalation via Crafted Environment (High)  
  
#### Severity Rationale:  
The vulnerability of privilege escalation via a crafted environment is classified as high due to the significant impact it can have on the organization's security. In this scenario, when the UseLogin feature is enabled and PAM is configured to read .pam\_environment files in user home directories, a local user can exploit this vulnerability to gain elevated privileges by manipulating the environment variables, such as setting a malicious LD\_PRELOAD variable.  
  
The high severity rating indicates that this vulnerability poses a serious threat to the organization's security as it allows an attacker with local access to escalate their privileges and potentially gain unauthorized access to sensitive information, execute arbitrary code, or perform malicious actions on the system.  
  
#### Impact on the Organization:  
If this vulnerability is exploited, the organization could face severe consequences such as unauthorized access to critical systems, data exfiltration, system compromise, and potential disruption of operations. An attacker could leverage the privilege escalation to carry out further attacks, manipulate system resources, and potentially cause significant harm to the organization's infrastructure and data.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the privilege escalation vulnerability via a crafted environment, an attacker could gain elevated privileges on the system, allowing them to execute arbitrary commands, access sensitive data, modify system configurations, and potentially take control of the affected system. This could lead to unauthorized access, data theft, system compromise, and other malicious activities that could have a detrimental impact on the organization's security and operations.

#### Public exploits related to this finding

## Description of the finding

The do\_setup\_env function in session.c in sshd in OpenSSH through 7.2p2, when the UseLogin feature is enabled and PAM is configured to read .pam\_environment files in user home directories, allows local users to gain privileges by triggering a crafted environment for the /bin/login program, as demonstrated by an LD\_PRELOAD environment variable.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2015-8325, it is recommended to disable the UseLogin feature in the OpenSSH configuration if it is not required. This can be done by setting "UseLogin no" in the sshd\_config file. Additionally, it is advised to review and restrict the permissions of the .pam\_environment files in user home directories to prevent unauthorized access. Regularly updating OpenSSH to the latest version and monitoring for any suspicious activity related to environment variables can also help in mitigating this vulnerability.

### Finding 2

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2010-4478**
* Severity score: **7.5** (**HIGH** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: Authentication Bypass via J-PAKE Protocol (High)  
  
#### Severity Rationale:  
The vulnerability of authentication bypass via the J-PAKE protocol is classified as high due to the critical impact it can have on the organization's security. In this case, OpenSSH versions 5.6 and earlier, when J-PAKE is enabled, do not properly validate the public parameters in the J-PAKE protocol. This flaw allows remote attackers to bypass the need for knowledge of the shared secret and successfully authenticate by sending crafted values in each round of the protocol.  
  
The high severity rating indicates that this vulnerability poses a serious threat to the organization's security as it enables remote attackers to bypass authentication mechanisms and potentially gain unauthorized access to the system without the required credentials or shared secret knowledge.  
  
#### Impact on the Organization:  
If this vulnerability is exploited, the organization could face severe consequences such as unauthorized access to critical systems, data compromise, unauthorized privilege escalation, and potential disruption of services. An attacker could leverage this authentication bypass to gain unauthorized access to sensitive information, execute arbitrary commands, and potentially compromise the integrity and confidentiality of the organization's data.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the authentication bypass vulnerability via the J-PAKE protocol, remote attackers could successfully authenticate to the system without the necessary credentials or shared secret knowledge. This could allow them to gain unauthorized access to the organization's systems, manipulate data, execute malicious commands, and potentially carry out further attacks within the network. The attacker could exploit this vulnerability to maintain persistence, exfiltrate sensitive information, or disrupt the organization's operations, leading to significant security risks and potential financial losses.

#### Public exploits related to this finding

## Description of the finding

OpenSSH 5.6 and earlier, when J-PAKE is enabled, does not properly validate the public parameters in the J-PAKE protocol, which allows remote attackers to bypass the need for knowledge of the shared secret, and successfully authenticate, by sending crafted values in each round of the protocol, a related issue to CVE-2010-4252.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2010-4478 related to J-PAKE in OpenSSH, it is recommended to disable the J-PAKE feature if it is not required for the specific use case. This can be achieved by ensuring that the "J-PAKE" option is set to "no" in the sshd\_config file. Additionally, it is crucial to regularly update OpenSSH to the latest version to patch any known vulnerabilities. Implementing network segmentation and access controls to restrict access to the SSH service can also help in reducing the attack surface and mitigating the risk of exploitation.

### Finding 3

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2016-10010**
* Severity score: **7.0** (**HIGH** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: Privilege Escalation via Forwarded Unix-Domain Sockets (High)  
  
#### Severity Rationale:  
The vulnerability of privilege escalation via forwarded Unix-domain sockets in OpenSSH versions before 7.4, when privilege separation is not used, is classified as high due to the significant impact it can have on the organization's security. In this scenario, the sshd process creates forwarded Unix-domain sockets as the root user, which could potentially allow local users to gain elevated privileges through unspecified vectors related to serverloop.c.  
  
The high severity rating indicates that this vulnerability poses a serious threat to the organization's security as it enables local users to exploit the privilege escalation issue and potentially gain unauthorized access to sensitive resources, execute arbitrary commands, or perform malicious actions on the system.  
  
#### Impact on the Organization:  
If this vulnerability is exploited, the organization could face severe consequences such as unauthorized access to critical systems, data exfiltration, system compromise, and potential disruption of operations. An attacker could leverage the privilege escalation to escalate their privileges, manipulate system resources, and potentially cause significant harm to the organization's infrastructure and data.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the privilege escalation vulnerability via forwarded Unix-domain sockets, local users could gain elevated privileges on the system, allowing them to execute arbitrary commands, access sensitive data, modify system configurations, and potentially take control of the affected system. This could lead to unauthorized access, data theft, system compromise, and other malicious activities that could have a detrimental impact on the organization's security and operations.

#### Public exploits related to this finding

#### Exploit ID 40962.

* Description: OpenSSH &lt; 7.4 - &#039;UsePrivilegeSeparation Disabled&#039; Forwarded Unix Domain Sockets Privilege Escalation
* Download URL: https://www.exploit-db.com/exploits/40962

## Description of the finding

sshd in OpenSSH before 7.4, when privilege separation is not used, creates forwarded Unix-domain sockets as root, which might allow local users to gain privileges via unspecified vectors, related to serverloop.c.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2016-10010 in OpenSSH, it is crucial to enable privilege separation in the OpenSSH configuration. This can be done by setting "UsePrivilegeSeparation yes" in the sshd\_config file. By enabling privilege separation, forwarded Unix-domain sockets will not be created as root, reducing the risk of local users gaining privileges. Additionally, it is recommended to regularly update OpenSSH to versions 7.4 and above, where this vulnerability has been addressed. Conducting regular security audits and monitoring for any unauthorized access or suspicious activity can also help in detecting and mitigating potential privilege escalation attempts.

### Finding 4

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2008-1657**
* Severity score: **6.5** (**MEDIUM** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: Bypass of ForceCommand Directive via .ssh/rc Session File (Medium)  
  
#### Severity Rationale:  
The vulnerability of bypassing the ForceCommand directive in OpenSSH versions 4.4 up to versions before 4.9 by modifying the .ssh/rc session file is classified as medium due to the moderate impact it can have on the organization's security. In this case, remote authenticated users can bypass the restrictions imposed by the ForceCommand directive in the sshd\_config file by manipulating the .ssh/rc session file.  
  
The medium severity rating indicates that while this vulnerability has the potential to allow unauthorized actions by authenticated users, it may not directly lead to critical system compromise or data loss. However, it still poses a security risk that needs to be addressed to prevent unauthorized access and potential misuse of system resources.  
  
#### Impact on the Organization:  
If this vulnerability is exploited, the organization could face consequences such as unauthorized command execution, unauthorized access to resources, and potential manipulation of system configurations by authenticated users. While the impact may not be as severe as critical vulnerabilities, it still poses a risk to the organization's security posture and could lead to unauthorized activities within the system.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the bypass of the ForceCommand directive via the .ssh/rc session file, authenticated users could potentially execute unauthorized commands, access resources beyond their intended permissions, and manipulate the system in ways not allowed by the configured restrictions. This could lead to unauthorized actions within the system, potential information disclosure, and misuse of system resources, albeit not as severe as critical vulnerabilities.

#### Public exploits related to this finding

## Description of the finding

OpenSSH 4.4 up to versions before 4.9 allows remote authenticated users to bypass the sshd\_config ForceCommand directive by modifying the .ssh/rc session file.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2008-1657 in OpenSSH, it is recommended to carefully review and restrict the permissions of the .ssh/rc session file to prevent unauthorized modifications by remote authenticated users. Additionally, it is advised to avoid using the ForceCommand directive in the sshd\_config file if it is not necessary for the specific use case. Regularly monitoring and auditing the .ssh/rc file for any unauthorized changes can help in detecting and preventing potential bypass attempts. Upgrading to a version of OpenSSH 4.9 or above, where this vulnerability has been addressed, is also recommended to enhance security.

### Finding 5

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2016-3115**
* Severity score: **6.4** (**MEDIUM** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: CRLF Injection in OpenSSH session.c (Medium)  
  
#### Severity Rationale:  
The vulnerability of CRLF injection in session.c in OpenSSH before 7.2p2, allowing remote authenticated users to bypass intended shell-command restrictions via crafted X11 forwarding data, is classified as medium due to the moderate impact it can have on the organization's security. In this scenario, multiple CRLF injection vulnerabilities in the do\_authenticated1 and session\_x11\_req functions can be exploited by authenticated users to bypass shell-command restrictions.  
  
The medium severity rating indicates that while this vulnerability can lead to unauthorized actions by authenticated users, it may not directly result in critical system compromise or data loss. However, it still poses a security risk that needs to be addressed to prevent unauthorized access and potential misuse of system resources.  
  
#### Impact on the Organization:  
If this vulnerability is exploited, the organization could face consequences such as bypassing shell-command restrictions, unauthorized command execution, and potential manipulation of system configurations by authenticated users. While the impact may not be as severe as critical vulnerabilities, it still poses a risk to the organization's security posture and could lead to unauthorized activities within the system.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the CRLF injection vulnerabilities in OpenSSH session.c, authenticated users could bypass intended shell-command restrictions, execute unauthorized commands, and potentially manipulate the system in ways not allowed by the configured restrictions. This could lead to unauthorized actions within the system, potential information disclosure, and misuse of system resources, albeit not as severe as critical vulnerabilities.

#### Public exploits related to this finding

#### Exploit ID 39569.

* Description: OpenSSH 7.2p1 - (Authenticated) xauth Command Injection
* Download URL: https://www.exploit-db.com/exploits/39569

## Description of the finding

Multiple CRLF injection vulnerabilities in session.c in sshd in OpenSSH before 7.2p2 allow remote authenticated users to bypass intended shell-command restrictions via crafted X11 forwarding data, related to the (1) do\_authenticated1 and (2) session\_x11\_req functions.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2016-3115 in OpenSSH, it is recommended to upgrade to version 7.2p2 or later, where this vulnerability has been addressed. Additionally, it is important to review and restrict the X11 forwarding settings in the sshd\_config file to limit potential CRLF injection vulnerabilities. Enforcing proper input validation and sanitization mechanisms for X11 forwarding data can help prevent remote authenticated users from bypassing shell-command restrictions. Regularly monitoring and auditing X11 forwarding activities for any suspicious behavior can also aid in detecting and mitigating potential exploitation attempts.

### Finding 6

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2023-48795**
* Severity score: **5.9** (**MEDIUM** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: Terrapin Attack in SSH Transport Protocol (Medium)  
  
#### Severity Rationale:  
The vulnerability known as the Terrapin attack in the SSH transport protocol, found in OpenSSH before 9.6 and other affected products, is classified as medium due to the moderate impact it can have on the security of the communication channel. This vulnerability allows remote attackers to bypass integrity checks, leading to the omission of certain packets during the extension negotiation phase. As a result, a connection may be established where some security features are downgraded or disabled, potentially exposing the communication to security risks.  
  
The medium severity rating indicates that while this vulnerability can lead to the downgrading or disabling of security features in the SSH connection, it may not directly result in immediate critical system compromise or data loss. However, it still poses a security risk that needs to be addressed to prevent potential exploitation by attackers.  
  
#### Impact on the Organization:  
If this vulnerability is exploited, the organization's SSH connections could be at risk of having security features downgraded or disabled, potentially exposing sensitive data to interception or manipulation. While the impact may not be as severe as critical vulnerabilities, the compromise of security features in the SSH transport protocol could lead to unauthorized access or manipulation of data in transit.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the Terrapin attack in the SSH transport protocol, remote attackers could manipulate the handshake phase and sequence numbers, leading to the bypass of integrity checks and potential downgrading or disabling of security features such as the ChaCha20-Poly1305 encryption algorithm. This could result in the exposure of sensitive information, unauthorized access to SSH connections, and potential security risks for the affected systems and communication channels.

#### Public exploits related to this finding

## Description of the finding

The SSH transport protocol with certain OpenSSH extensions, found in OpenSSH before 9.6 and other products, allows remote attackers to bypass integrity checks such that some packets are omitted (from the extension negotiation message), and a client and server may consequently end up with a connection for which some security features have been downgraded or disabled, aka a Terrapin attack. This occurs because the SSH Binary Packet Protocol (BPP), implemented by these extensions, mishandles the handshake phase and mishandles use of sequence numbers. For example, there is an effective attack against SSH's use of ChaCha20-Poly1305 (and CBC with Encrypt-then-MAC). The bypass occurs in chacha20-poly1305@openssh.com and (if CBC is used) the -etm@openssh.com MAC algorithms. This also affects Maverick Synergy Java SSH API before 3.1.0-SNAPSHOT, Dropbear through 2022.83, Ssh before 5.1.1 in Erlang/OTP, PuTTY before 0.80, AsyncSSH before 2.14.2, golang.org/x/crypto before 0.17.0, libssh before 0.10.6, libssh2 through 1.11.0, Thorn Tech SFTP Gateway before 3.4.6, Tera Term before 5.1, Paramiko before 3.4.0, jsch before 0.2.15, SFTPGo before 2.5.6, Netgate pfSense Plus through 23.09.1, Netgate pfSense CE through 2.7.2, HPN-SSH through 18.2.0, ProFTPD before 1.3.8b (and before 1.3.9rc2), ORYX CycloneSSH before 2.3.4, NetSarang XShell 7 before Build 0144, CrushFTP before 10.6.0, ConnectBot SSH library before 2.2.22, Apache MINA sshd through 2.11.0, sshj through 0.37.0, TinySSH through 20230101, trilead-ssh2 6401, LANCOM LCOS and LANconfig, FileZilla before 3.66.4, Nova before 11.8, PKIX-SSH before 14.4, SecureCRT before 9.4.3, Transmit5 before 5.10.4, Win32-OpenSSH before 9.5.0.0p1-Beta, WinSCP before 6.2.2, Bitvise SSH Server before 9.32, Bitvise SSH Client before 9.33, KiTTY through 0.76.1.13, the net-ssh gem 7.2.0 for Ruby, the mscdex ssh2 module before 1.15.0 for Node.js, the thrussh library before 0.35.1 for Rust, and the Russh crate before 0.40.2 for Rust.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2023-48795 affecting the SSH transport protocol with certain OpenSSH extensions, it is recommended to upgrade OpenSSH to version 9.6 or later, where this vulnerability has been addressed. Additionally, it is important to regularly update any other affected products or libraries listed in the description to the latest secure versions. Implementing strict firewall rules and network segmentation to restrict access to SSH services can help reduce the attack surface. Monitoring SSH traffic for any unusual or suspicious activity can aid in detecting potential Terrapin attacks. Conducting regular security assessments and staying informed about security updates and patches for SSH implementations can also help in mitigating the risk of exploitation.

### Finding 7

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2016-6210**
* Severity score: **5.9** (**MEDIUM** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: User Enumeration via Timing Attack in OpenSSH (Medium)  
  
#### Severity Rationale:  
The vulnerability in sshd in OpenSSH before 7.3, where BLOWFISH hashing is used on a static password when SHA256 or SHA512 are used for user password hashing and the username does not exist, is classified as medium due to the moderate impact it can have on the organization's security. This vulnerability allows remote attackers to enumerate users by exploiting the timing difference between responses when a large password is provided.  
  
The medium severity rating indicates that while this vulnerability can lead to user enumeration and potential information disclosure, it may not directly result in immediate critical system compromise or data loss. However, it still poses a security risk that needs to be addressed to prevent unauthorized user enumeration and potential misuse of the information.  
  
#### Impact on the Organization:  
If this vulnerability is exploited, remote attackers could potentially enumerate valid user accounts on the system by leveraging timing differences in responses when providing large passwords. This information could be used for further targeted attacks, unauthorized access attempts, or social engineering tactics. While the impact may not be as severe as critical vulnerabilities, user enumeration can still pose a risk to the organization's security posture.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the timing attack vulnerability in OpenSSH, remote attackers could determine the existence of valid user accounts on the system by observing the timing differences in responses from the server. This could allow attackers to gather intelligence on potential targets, plan further attacks, or attempt to gain unauthorized access to the system using the enumerated user accounts. While the vulnerability may not directly lead to system compromise, it can aid attackers in their reconnaissance efforts and potentially lead to security breaches if not mitigated.

#### Public exploits related to this finding

#### Exploit ID 40136.

* Description: OpenSSH 7.2p2 - Username Enumeration
* Download URL: https://www.exploit-db.com/exploits/40136

#### Exploit ID 40113.

* Description: OpenSSHd 7.2p2 - Username Enumeration
* Download URL: https://www.exploit-db.com/exploits/40113

## Description of the finding

sshd in OpenSSH before 7.3, when SHA256 or SHA512 are used for user password hashing, uses BLOWFISH hashing on a static password when the username does not exist, which allows remote attackers to enumerate users by leveraging the timing difference between responses when a large password is provided.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2016-6210 in OpenSSH, it is recommended to upgrade to OpenSSH version 7.3 or later, where this vulnerability has been addressed. Additionally, it is important to ensure that user password hashing algorithms such as SHA256 or SHA512 are used instead of BLOWFISH hashing to prevent the timing difference enumeration attack. Implementing rate limiting mechanisms for authentication attempts can help mitigate the risk of user enumeration through timing differences. Regularly monitoring authentication logs for unusual patterns or repeated failed login attempts can aid in detecting potential enumeration attacks. Conducting regular security assessments and staying informed about security best practices for password hashing can also help enhance the overall security of the SSH implementation.

### Finding 8

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2017-15906**
* Severity score: **5.3** (**MEDIUM** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: Write Operations in Readonly Mode in OpenSSH (Medium)  
  
#### Severity Rationale:  
The vulnerability in the process\_open function in sftp-server.c in OpenSSH before 7.6, which does not properly prevent write operations in readonly mode, allowing attackers to create zero-length files, is classified as medium due to the moderate impact it can have on the organization's security. This vulnerability enables attackers to perform unauthorized write operations in a readonly mode, leading to the creation of zero-length files.  
  
The medium severity rating indicates that while this vulnerability can result in unauthorized file creation and potential disruption of operations, it may not directly lead to critical system compromise or data loss. However, it still poses a security risk that needs to be addressed to prevent unauthorized write operations and potential misuse of the system.  
  
#### Impact on the Organization:  
If this vulnerability is exploited, attackers could create zero-length files on the system by bypassing the readonly mode restrictions, potentially causing confusion, resource wastage, or disruption of services. While the impact may not be as severe as critical vulnerabilities, unauthorized write operations can still pose a risk to the organization's data integrity and operational stability.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the vulnerability in OpenSSH that allows write operations in readonly mode, attackers can create zero-length files on the system, potentially leading to resource exhaustion, denial of service, or other disruptive activities. While the creation of zero-length files may not directly lead to data compromise, it can still impact the system's functionality and availability. Attackers could leverage this vulnerability to disrupt operations, cause confusion, or potentially pave the way for further malicious actions on the affected system.

#### Public exploits related to this finding

## Description of the finding

The process\_open function in sftp-server.c in OpenSSH before 7.6 does not properly prevent write operations in readonly mode, which allows attackers to create zero-length files.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2017-15906 in OpenSSH, it is recommended to upgrade to OpenSSH version 7.6 or later, where this vulnerability has been addressed. Additionally, it is important to review and adjust the permissions and configurations of the SFTP server to prevent unauthorized write operations in readonly mode. Implementing proper file system permissions and access controls can help restrict users from creating zero-length files. Regularly monitoring file system activities and auditing SFTP server logs for any suspicious write operations can aid in detecting and preventing unauthorized file creation. Conducting regular security assessments and staying informed about security updates for OpenSSH can also help in enhancing the overall security posture of the system.

### Finding 9

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2016-20012**
* Severity score: **5.3** (**MEDIUM** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: User Enumeration in OpenSSH (Medium)  
  
#### Severity Rationale:  
The vulnerability in OpenSSH through version 8.7 that allows remote attackers to test whether a certain combination of username and public key is known to an SSH server is classified as medium in severity. This vulnerability enables attackers to determine the validity of a username and public key combination by observing the server's response behavior, potentially leading to user enumeration.  
  
The medium severity rating indicates that while user enumeration may not be recognized as a vulnerability by the vendor, it still poses a security risk by potentially disclosing valid username and public key combinations to attackers. While this vulnerability may not directly result in critical system compromise or data loss, it can aid attackers in reconnaissance efforts and potentially lead to targeted attacks.  
  
#### Impact on the Organization:  
If this vulnerability is exploited, attackers could gather information about valid username and public key combinations on the SSH server, potentially aiding them in targeted attacks or unauthorized access attempts. While the impact may not be as severe as critical vulnerabilities, user enumeration can still pose a risk to the organization's security posture by revealing potentially sensitive information.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the user enumeration vulnerability in OpenSSH, attackers can test the validity of username and public key combinations by observing the server's response behavior. This information can be used to identify valid accounts, plan targeted attacks, or attempt unauthorized access to the system. While the vendor may not consider user enumeration a vulnerability, attackers can leverage this information to refine their attack strategies and potentially compromise the security of the SSH server and associated systems.

#### Public exploits related to this finding

## Description of the finding

OpenSSH through 8.7 allows remote attackers, who have a suspicion that a certain combination of username and public key is known to an SSH server, to test whether this suspicion is correct. This occurs because a challenge is sent only when that combination could be valid for a login session. NOTE: the vendor does not recognize user enumeration as a vulnerability for this product

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2016-20012 in OpenSSH, it is recommended to implement additional security measures such as rate limiting, account lockout policies, and strong password policies to prevent unauthorized access even if user enumeration is possible. Regularly monitoring authentication logs for unusual patterns or repeated login attempts can help in detecting potential enumeration attacks. It is also important to educate users about the importance of using unique and strong passwords to mitigate the risk of unauthorized access. Conducting regular security assessments and staying informed about security best practices for SSH server configurations can help enhance the overall security of the system.

### Finding 10

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2014-2532**
* Severity score: **4.9** (**MEDIUM** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: Bypass of Environment Restrictions in OpenSSH (Medium)  
  
#### Severity Rationale:  
The vulnerability in sshd in OpenSSH before version 6.6, where wildcards on AcceptEnv lines in sshd\_config are not properly supported, allowing remote attackers to bypass intended environment restrictions by using a substring located before a wildcard character, is classified as medium in severity. This vulnerability enables attackers to circumvent environment restrictions set by administrators, potentially leading to unauthorized access or manipulation of the system.  
  
The medium severity rating indicates that while this vulnerability can lead to the bypass of environment restrictions, it may not directly result in immediate critical system compromise or data loss. However, it still poses a security risk that needs to be addressed to prevent unauthorized access and potential misuse of system resources.  
  
#### Impact on the Organization:  
If this vulnerability is exploited, remote attackers could bypass intended environment restrictions by manipulating AcceptEnv lines in sshd\_config, potentially gaining unauthorized access or manipulating the system environment. While the impact may not be as severe as critical vulnerabilities, the bypass of environment restrictions can still pose a risk to the organization's security posture.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the vulnerability in OpenSSH that allows the bypass of environment restrictions, attackers can use substrings located before wildcard characters to circumvent intended restrictions set by administrators. This could allow attackers to introduce unauthorized environment variables, potentially gaining access to restricted resources, manipulating system behavior, or launching further attacks within the system. While the severity is medium, the exploitation of this vulnerability could lead to unauthorized actions and compromise the security of the SSH server and associated systems.

#### Public exploits related to this finding

## Description of the finding

sshd in OpenSSH before 6.6 does not properly support wildcards on AcceptEnv lines in sshd\_config, which allows remote attackers to bypass intended environment restrictions by using a substring located before a wildcard character.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2014-2532 in OpenSSH, it is recommended to upgrade to OpenSSH version 6.6 or later, where this vulnerability has been addressed. Additionally, it is important to review and update the AcceptEnv lines in the sshd\_config file to avoid using wildcards or carefully restrict the use of wildcards to prevent unintended environment restrictions bypass. Implementing strict input validation and sanitization mechanisms for environment variables can help prevent remote attackers from exploiting this vulnerability. Regularly reviewing and auditing the sshd\_config file for any misconfigurations or insecure settings can aid in enhancing the security of the SSH server. Conducting regular security assessments and staying informed about security updates for OpenSSH can also help in mitigating the risk of exploitation.

### Finding 11

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2010-4755**
* Severity score: **4.0** (**MEDIUM** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: Denial of Service via Crafted Glob Expressions in OpenSSH (Medium)  
  
#### Severity Rationale:  
The vulnerability in the remote\_glob function in sftp-glob.c and the process\_put function in sftp.c in OpenSSH 5.8 and earlier, which allows remote authenticated users to cause a denial of service (CPU and memory consumption) via crafted glob expressions that do not match any pathnames, is classified as medium in severity. This vulnerability can lead to resource exhaustion, impacting the availability and performance of the affected system.  
  
The medium severity rating indicates that while this vulnerability can result in denial of service due to excessive CPU and memory consumption, it may not directly lead to critical system compromise or data loss. However, it still poses a security risk that needs to be addressed to prevent service disruption and potential impact on system operations.  
  
#### Impact on the Organization:  
If this vulnerability is exploited, remote authenticated users could cause a denial of service by sending crafted glob expressions that do not match any pathnames, leading to excessive CPU and memory consumption. This could result in service degradation, unresponsiveness, or system instability, impacting the availability of the affected SSH server and potentially disrupting operations.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the vulnerability in OpenSSH that allows a denial of service via crafted glob expressions, attackers can consume excessive CPU and memory resources on the target system, leading to service disruption and potential unavailability. This could impact the performance of the SSH server, affect other services running on the system, and cause operational challenges for the organization. While the severity is medium, the exploitation of this vulnerability could lead to significant disruption and resource exhaustion.

#### Public exploits related to this finding

## Description of the finding

The (1) remote\_glob function in sftp-glob.c and the (2) process\_put function in sftp.c in OpenSSH 5.8 and earlier, as used in FreeBSD 7.3 and 8.1, NetBSD 5.0.2, OpenBSD 4.7, and other products, allow remote authenticated users to cause a denial of service (CPU and memory consumption) via crafted glob expressions that do not match any pathnames, as demonstrated by glob expressions in SSH\_FXP\_STAT requests to an sftp daemon, a different vulnerability than CVE-2010-2632.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2010-4755 in OpenSSH, it is recommended to upgrade to OpenSSH version 5.8 or later, where this vulnerability has been addressed. Additionally, it is important to review and restrict the use of glob expressions in SSH\_FXP\_STAT requests to the sftp daemon to prevent remote authenticated users from causing denial of service by consuming excessive CPU and memory resources. Implementing proper input validation and limiting the complexity of glob expressions can help mitigate the risk of exploitation. Regularly monitoring system resources and performance metrics can aid in detecting abnormal CPU and memory consumption patterns that may indicate a denial of service attack. Conducting regular security assessments and staying informed about security updates for OpenSSH can also help in enhancing the overall security of the system.

### Finding 12

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2011-5000**
* Severity score: **3.5** (**LOW** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: Denial of Service via Large Value in Length Field in OpenSSH (Low)  
  
#### Severity Rationale:  
The vulnerability in the ssh\_gssapi\_parse\_ename function in gss-serv.c in OpenSSH 5.8 and earlier, when gssapi-with-mic authentication is enabled, allowing remote authenticated users to cause a denial of service (memory consumption) via a large value in a certain length field, is classified as low in severity. This vulnerability may have limited relevance and impact, leading to memory consumption but not resulting in critical system compromise or data loss.  
  
The low severity rating indicates that while this vulnerability can result in denial of service due to memory consumption, it is not likely to have a significant impact on the organization's security posture or operations. The limited scenarios in which this issue is relevant contribute to the low severity rating.  
  
#### Impact on the Organization:  
If this vulnerability is exploited in specific scenarios where gssapi-with-mic authentication is enabled, remote authenticated users could cause a denial of service by triggering memory consumption with a large value in a certain length field. While this may lead to service disruption, the impact is limited and may not have widespread consequences for the organization.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the vulnerability in OpenSSH that allows a denial of service via a large value in a length field, attackers can trigger memory consumption, potentially leading to service disruption for the affected SSH server. However, due to the limited relevance and impact of this issue, the practical implications for attackers may be minimal, and the overall risk to the organization is low.

#### Public exploits related to this finding

## Description of the finding

The ssh\_gssapi\_parse\_ename function in gss-serv.c in OpenSSH 5.8 and earlier, when gssapi-with-mic authentication is enabled, allows remote authenticated users to cause a denial of service (memory consumption) via a large value in a certain length field. NOTE: there may be limited scenarios in which this issue is relevant.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2011-5000 in OpenSSH, it is recommended to upgrade to OpenSSH version 5.8 or later, where this vulnerability has been addressed. Additionally, it is important to review and adjust the configuration settings related to gssapi-with-mic authentication to prevent remote authenticated users from causing denial of service by consuming excessive memory. Implementing proper input validation and limiting the size of certain length fields can help mitigate the risk of exploitation. Regularly monitoring system resources and performance metrics can aid in detecting abnormal memory consumption patterns that may indicate a denial of service attack. Conducting regular security assessments and staying informed about security updates for OpenSSH can also help in enhancing the overall security of the system.

### Finding 13

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2011-4327**
* Severity score: **2.1** (**LOW** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: Information Disclosure via ssh-rand-helper in OpenSSH (Low)  
  
#### Severity Rationale:  
The vulnerability in ssh-keysign.c in ssh-keysign in OpenSSH before 5.8p2, where ssh-rand-helper is executed with unintended open file descriptors on certain platforms, allowing local users to obtain sensitive key information via the ptrace system call, is classified as low in severity. This vulnerability may lead to information disclosure but is limited in impact and may not result in critical system compromise or data loss.  
  
The low severity rating indicates that while this vulnerability can potentially expose sensitive key information to local users, the impact is limited and may not pose a significant risk to the organization's security posture. The specific conditions required for exploitation and the local nature of the attack contribute to the low severity rating.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by local users on certain platforms, they may be able to obtain sensitive key information via the ptrace system call. While this could lead to information disclosure, the impact is limited to local access and may not have widespread consequences for the organization.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the vulnerability in OpenSSH that allows local users to obtain sensitive key information via unintended open file descriptors in ssh-rand-helper, attackers could potentially access key material through the ptrace system call. However, due to the local nature of the attack and the specific conditions required for exploitation, the practical implications for attackers are limited. The overall risk to the organization is low, and the impact is constrained to specific scenarios on certain platforms.

#### Public exploits related to this finding

## Description of the finding

ssh-keysign.c in ssh-keysign in OpenSSH before 5.8p2 on certain platforms executes ssh-rand-helper with unintended open file descriptors, which allows local users to obtain sensitive key information via the ptrace system call.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2011-4327 in OpenSSH, it is recommended to upgrade to OpenSSH version 5.8p2 or later, where this vulnerability has been addressed. Additionally, it is important to review and secure the execution environment of ssh-keysign to prevent unintended open file descriptors and potential sensitive key information leakage. Implementing proper file descriptor management and access controls can help mitigate the risk of unauthorized access to key information via the ptrace system call. Regularly monitoring system calls and auditing file descriptor usage can aid in detecting any suspicious activity that may indicate unauthorized access attempts. Conducting regular security assessments and staying informed about security updates for OpenSSH can also help in enhancing the overall security of the system.

### Finding 14

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2023-51385**
* Severity score: **0.0** (**MEDIUM** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: OS Command Injection in OpenSSH (Medium)  
  
#### Severity Rationale:  
The vulnerability in ssh in OpenSSH before version 9.6, where OS command injection might occur if a username or hostname contains shell metacharacters and is referenced by an expansion token in certain situations, is classified as medium in severity. This vulnerability can lead to OS command injection, potentially allowing attackers to execute arbitrary commands on the system.  
  
The medium severity rating indicates that while this vulnerability can result in OS command injection, it may not directly lead to immediate critical system compromise or data loss. However, it still poses a security risk that needs to be addressed to prevent unauthorized command execution and potential system manipulation.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by attackers using shell metacharacters in usernames or hostnames, it could lead to OS command injection and unauthorized command execution on the system. This could potentially result in system compromise, data exfiltration, or disruption of operations, impacting the organization's security and integrity.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OS command injection vulnerability in OpenSSH, attackers could inject arbitrary commands into the system if a username or hostname with shell metacharacters is referenced by an expansion token. This could allow attackers to execute unauthorized commands, escalate privileges, manipulate system resources, and potentially compromise the security of the affected system. While the severity is medium, the exploitation of this vulnerability could lead to significant security risks and unauthorized actions on the system.

#### Public exploits related to this finding

## Description of the finding

In ssh in OpenSSH before 9.6, OS command injection might occur if a user name or host name has shell metacharacters, and this name is referenced by an expansion token in certain situations. For example, an untrusted Git repository can have a submodule with shell metacharacters in a user name or host name.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2023-51385 in OpenSSH, it is recommended to upgrade to OpenSSH version 9.6 or later, where this vulnerability has been addressed. Additionally, it is important to carefully review and sanitize user input, especially when it involves expansion tokens that may contain shell metacharacters. Implementing proper input validation and sanitization mechanisms can help prevent OS command injection attacks. Regularly monitoring and auditing user input for any suspicious characters or patterns can aid in detecting and preventing potential exploitation. Educating users about the risks of including shell metacharacters in usernames or hostnames can also help in mitigating the risk of OS command injection vulnerabilities. Conducting regular security assessments and staying informed about security best practices for OpenSSH configurations can further enhance the security posture of the system.

### Finding 15

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2023-38408**
* Severity score: **0.0** (**CRITICAL** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: Remote Code Execution via PKCS#11 in OpenSSH (Critical)  
  
#### Severity Rationale:  
The vulnerability in the PKCS#11 feature in ssh-agent in OpenSSH before version 9.3p2, with an insufficiently trustworthy search path, leading to remote code execution if an agent is forwarded to an attacker-controlled system, is classified as critical in severity. This vulnerability allows remote attackers to execute arbitrary code on the target system, posing a significant risk to the security and integrity of the affected environment.  
  
The critical severity rating indicates that this vulnerability has the potential to result in severe consequences, including unauthorized access, system compromise, data exfiltration, and potential disruption of operations. The ability for remote attackers to execute code on the system poses a critical threat that requires immediate attention and remediation.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by forwarding an agent to an attacker-controlled system, remote attackers could execute arbitrary code on the target system, leading to unauthorized access, system compromise, and potential data breaches. The impact could be severe, affecting the organization's security posture, confidentiality of data, and overall operational stability.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the remote code execution vulnerability via the PKCS#11 feature in OpenSSH, attackers can execute arbitrary code on the target system when an agent is forwarded to an attacker-controlled environment. This could lead to complete compromise of the system, unauthorized access to sensitive information, manipulation of system resources, and potential disruption of critical services. The critical severity of this vulnerability underscores the significant risk posed by remote attackers gaining the ability to execute code on the affected system.

#### Public exploits related to this finding

## Description of the finding

The PKCS#11 feature in ssh-agent in OpenSSH before 9.3p2 has an insufficiently trustworthy search path, leading to remote code execution if an agent is forwarded to an attacker-controlled system. (Code in /usr/lib is not necessarily safe for loading into ssh-agent.) NOTE: this issue exists because of an incomplete fix for CVE-2016-10009.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2023-38408 in OpenSSH, it is crucial to upgrade to OpenSSH version 9.3p2 or later, where this vulnerability has been addressed. Additionally, it is important to review and secure the search path used by the PKCS#11 feature in ssh-agent to prevent remote code execution if an agent is forwarded to an attacker-controlled system. Implementing proper access controls and restricting the search path to trusted locations can help mitigate the risk of unauthorized code execution. Regularly monitoring and auditing the configuration of ssh-agent for any suspicious changes or unauthorized access can aid in detecting and preventing potential exploitation. Educating users about the risks of forwarding ssh-agent to untrusted systems can also help in mitigating the risk of remote code execution vulnerabilities. Conducting regular security assessments and staying informed about security updates for OpenSSH can further enhance the security posture of the system.

### Finding 16

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2021-36368**
* Severity score: **0.0** (**LOW** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: Ambiguity in FIDO Authentication with None Option in OpenSSH (Low)  
  
#### Severity Rationale:  
The issue discovered in OpenSSH before version 8.9, where ambiguity arises in FIDO authentication with the None option when a client is using public-key authentication with agent forwarding but without -oLogLevel=verbose, is classified as low in severity. This ambiguity can lead to confusion for users regarding the authentication process but does not directly result in an authentication bypass or immediate security compromise.  
  
The low severity rating indicates that while this issue may introduce uncertainty in the authentication process, it does not lead to a direct security risk that could result in critical system compromise or data loss. The impact is limited, and the vulnerability does not bypass any authentication mechanisms.  
  
#### Impact on the Organization:  
If this issue is encountered in OpenSSH environments where FIDO authentication with the None option is used without clear logging settings, users may face uncertainty regarding the authentication process. While this ambiguity may cause confusion, it does not directly lead to unauthorized access or compromise of the organization's security.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
In scenarios where an attacker has silently modified the server to support the None authentication option, users may face uncertainty about the authentication process when using FIDO authentication with agent forwarding. However, as noted by the vendor, this issue does not result in an authentication bypass or direct security compromise. The impact is limited to potential confusion for users and does not pose a significant risk to the organization's security posture.

#### Public exploits related to this finding

## Description of the finding

An issue was discovered in OpenSSH before 8.9. If a client is using public-key authentication with agent forwarding but without -oLogLevel=verbose, and an attacker has silently modified the server to support the None authentication option, then the user cannot determine whether FIDO authentication is going to confirm that the user wishes to connect to that server, or that the user wishes to allow that server to connect to a different server on the user's behalf. NOTE: the vendor's position is "this is not an authentication bypass, since nothing is being bypassed.

## Mitigation steps

Mitigation steps:  
To mitigate the issue described in CVE-2021-36368 in OpenSSH, it is recommended to upgrade to OpenSSH version 8.9 or later, where this issue has been addressed. Additionally, users can mitigate the risk by ensuring that the -oLogLevel=verbose option is used when using public-key authentication with agent forwarding. This can help provide more visibility into the authentication process and potential security risks. It is also important to regularly monitor and audit SSH connections for any suspicious activity or unauthorized modifications to the server configuration. Educating users about the importance of verifying authentication prompts and being cautious when connecting to unknown servers can help prevent potential security risks associated with this issue. Conducting regular security assessments and staying informed about security best practices for SSH configurations can further enhance the security posture of the system.

### Finding 17

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2020-15778**
* Severity score: **0.0** (**HIGH** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: Command Injection in scp in OpenSSH (High)  
  
#### Severity Rationale:  
The vulnerability in scp in OpenSSH through version 8.3p1, allowing command injection in the scp.c toremote function, as demonstrated by backtick characters in the destination argument, is classified as high in severity. This vulnerability enables attackers to inject and execute arbitrary commands on the target system, posing a significant risk to the security and integrity of the affected environment.  
  
The high severity rating indicates that this vulnerability has the potential to result in severe consequences, including unauthorized command execution, system compromise, and potential data exfiltration. The ability for attackers to inject commands through scp poses a critical threat that requires immediate attention and remediation.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by injecting commands in the scp destination argument, attackers can execute arbitrary commands on the target system, leading to unauthorized access, system compromise, and potential data breaches. The impact could be severe, affecting the organization's security posture, confidentiality of data, and overall operational stability.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the command injection vulnerability in scp in OpenSSH, attackers can inject and execute arbitrary commands on the target system, potentially leading to complete compromise of the system, unauthorized access to sensitive information, manipulation of system resources, and potential disruption of critical services. The high severity of this vulnerability underscores the significant risk posed by attackers gaining the ability to execute commands on the affected system.

#### Public exploits related to this finding

## Description of the finding

scp in OpenSSH through 8.3p1 allows command injection in the scp.c toremote function, as demonstrated by backtick characters in the destination argument. NOTE: the vendor reportedly has stated that they intentionally omit validation of "anomalous argument transfers" because that could "stand a great chance of breaking existing workflows."

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2020-15778 in OpenSSH, it is recommended to upgrade to OpenSSH version 8.3p1 or later, where this vulnerability has been addressed. Additionally, users can mitigate the risk by being cautious when using the scp command and avoiding the use of backtick characters in the destination argument. It is important to review and validate all input arguments to prevent command injection attacks. Implementing proper input validation and sanitization mechanisms can help prevent unauthorized command execution. Regularly monitoring and auditing scp commands for any suspicious activity or unusual arguments can aid in detecting and preventing potential exploitation. Educating users about the risks of command injection vulnerabilities and promoting secure file transfer practices can also help in mitigating the risk of unauthorized access. Conducting regular security assessments and staying informed about security updates for OpenSSH can further enhance the security posture of the system.

### Finding 18

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2019-6111**
* Severity score: **0.0** (**MEDIUM** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: Arbitrary File Overwrite in OpenSSH scp (Medium)  
  
#### Severity Rationale:  
The issue discovered in OpenSSH 7.9, where a malicious scp server or Man-in-The-Middle attacker can overwrite arbitrary files in the scp client target directory due to insufficient validation of object names, is classified as medium in severity. This vulnerability allows for potential file overwrite attacks, especially when recursive operations are performed, posing a risk to the integrity and security of the affected systems.  
  
The medium severity rating indicates that while this vulnerability can lead to arbitrary file overwrites and manipulation of directories, it may not directly result in immediate critical system compromise or data loss. However, it still poses a security risk that needs to be addressed to prevent unauthorized modifications to files and directories.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by a malicious scp server or a Man-in-The-Middle attacker, it could lead to the overwrite of arbitrary files in the scp client target directory. This could potentially result in unauthorized modifications to critical files, such as .ssh/authorized\_keys, leading to unauthorized access or system compromise. The impact could be significant, affecting the organization's security posture and data integrity.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the arbitrary file overwrite vulnerability in OpenSSH scp, attackers can manipulate the scp client target directory and potentially overwrite arbitrary files, including sensitive configuration files like .ssh/authorized\_keys. This could lead to unauthorized access, privilege escalation, and compromise of the affected system. The ability to manipulate subdirectories during recursive operations further increases the risk of unauthorized modifications and potential security breaches. The medium severity of this vulnerability highlights the importance of addressing this issue to prevent unauthorized file overwrites and maintain the security of the system.

#### Public exploits related to this finding

#### Exploit ID 46193.

* Description: SCP Client - Multiple Vulnerabilities (SSHtranger Things)
* Download URL: https://www.exploit-db.com/exploits/46193

#### Exploit ID 46516.

* Description: OpenSSH SCP Client - Write Arbitrary Files
* Download URL: https://www.exploit-db.com/exploits/46516

## Description of the finding

An issue was discovered in OpenSSH 7.9. Due to the scp implementation being derived from 1983 rcp, the server chooses which files/directories are sent to the client. However, the scp client only performs cursory validation of the object name returned (only directory traversal attacks are prevented). A malicious scp server (or Man-in-The-Middle attacker) can overwrite arbitrary files in the scp client target directory. If recursive operation (-r) is performed, the server can manipulate subdirectories as well (for example, to overwrite the .ssh/authorized\_keys file).

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2019-6111 in OpenSSH, it is recommended to upgrade to OpenSSH version 7.9 or later, where this vulnerability has been addressed. Additionally, users can mitigate the risk by being cautious when using the scp command and avoiding interactions with untrusted or malicious servers. It is important to exercise caution when transferring files using scp and to verify the integrity of the files being transferred. Implementing strict file permissions and access controls on the target directory can help prevent unauthorized file overwrites. Regularly monitoring and auditing file transfers for any suspicious activity or unexpected changes can aid in detecting and preventing potential exploitation. Educating users about secure file transfer practices and the risks associated with untrusted servers can also help in mitigating the risk of unauthorized access. Conducting regular security assessments and staying informed about security updates for OpenSSH can further enhance the security posture of the system.

### Finding 19

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2019-6110**
* Severity score: **0.0** (**MEDIUM** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: Manipulation of Client Output in OpenSSH (Medium)  
  
#### Severity Rationale:  
The vulnerability in OpenSSH 7.9, where a malicious server or Man-in-The-Middle attacker can manipulate the client output by sending arbitrary stderr output, such as using ANSI control codes to hide additional files being transferred, is classified as medium in severity. This vulnerability allows for the manipulation of client output, potentially leading to confusion or obfuscation of the actual actions being performed during the SSH session.  
  
The medium severity rating indicates that while this vulnerability can result in manipulation of client output and potentially hide additional files being transferred, it may not directly lead to immediate critical system compromise or data loss. However, it still poses a security risk that needs to be addressed to prevent unauthorized manipulation of client information.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by a malicious server or a Man-in-The-Middle attacker, it could lead to the manipulation of client output, potentially hiding additional files being transferred or misleading the user about the actions being performed. This could create confusion, impact the integrity of data transfers, and potentially lead to security risks within the organization.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the manipulation of client output vulnerability in OpenSSH, attackers can use arbitrary stderr output to manipulate the client's display, potentially hiding additional files or actions being performed during the SSH session. This could lead to confusion, misdirection, or obfuscation of the actual activities taking place, potentially aiding attackers in concealing malicious actions or unauthorized transfers. While the severity is medium, the exploitation of this vulnerability could lead to security risks and impact the trustworthiness of the SSH session.

#### Public exploits related to this finding

#### Exploit ID 46193.

* Description: SCP Client - Multiple Vulnerabilities (SSHtranger Things)
* Download URL: https://www.exploit-db.com/exploits/46193

#### Exploit ID 46516.

* Description: OpenSSH SCP Client - Write Arbitrary Files
* Download URL: https://www.exploit-db.com/exploits/46516

## Description of the finding

In OpenSSH 7.9, due to accepting and displaying arbitrary stderr output from the server, a malicious server (or Man-in-The-Middle attacker) can manipulate the client output, for example to use ANSI control codes to hide additional files being transferred.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2019-6110 in OpenSSH, it is recommended to upgrade to OpenSSH version 7.9 or later, where this vulnerability has been addressed. Additionally, users can mitigate the risk by being cautious when using the scp command and avoiding interactions with untrusted or malicious servers. It is important to exercise caution when transferring files using scp and to verify the integrity of the files being transferred. Implementing strict file permissions and access controls on the target directory can help prevent unauthorized file manipulation. Regularly monitoring and auditing file transfers for any suspicious activity or unexpected changes can aid in detecting and preventing potential exploitation. Educating users about secure file transfer practices and the risks associated with untrusted servers can also help in mitigating the risk of unauthorized access. Conducting regular security assessments and staying informed about security updates for OpenSSH can further enhance the security posture of the system.

### Finding 20

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2019-6109**
* Severity score: **0.0** (**MEDIUM** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: Client Output Manipulation in OpenSSH (Medium)  
  
#### Severity Rationale:  
The issue discovered in OpenSSH 7.9, where missing character encoding in the progress display allows a malicious server or Man-in-The-Middle attacker to manipulate the client output, potentially hiding additional files being transferred using crafted object names and ANSI control codes, is classified as medium in severity. This vulnerability enables attackers to manipulate the client output, potentially leading to confusion or obfuscation of the actual file transfer operations.  
  
The medium severity rating indicates that while this vulnerability can result in the manipulation of client output and potentially hide additional files being transferred, it may not directly lead to immediate critical system compromise or data loss. However, it still poses a security risk that needs to be addressed to prevent unauthorized manipulation of client information.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by a malicious server or a Man-in-The-Middle attacker, it could lead to the manipulation of client output during file transfers, potentially hiding additional files or misleading the user about the transfer operations. This could create confusion, impact the integrity of data transfers, and potentially lead to security risks within the organization.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the client output manipulation vulnerability in OpenSSH, attackers can use crafted object names and ANSI control codes to manipulate the client output during file transfers. This could lead to the hiding of additional files being transferred or misrepresentation of the transfer progress, potentially aiding attackers in concealing malicious activities or unauthorized file transfers. While the severity is medium, the exploitation of this vulnerability could lead to security risks and impact the transparency and reliability of file transfer operations.

#### Public exploits related to this finding

## Description of the finding

An issue was discovered in OpenSSH 7.9. Due to missing character encoding in the progress display, a malicious server (or Man-in-The-Middle attacker) can employ crafted object names to manipulate the client output, e.g., by using ANSI control codes to hide additional files being transferred. This affects refresh\_progress\_meter() in progressmeter.c.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2019-6109 in OpenSSH, it is recommended to upgrade to OpenSSH version 7.9 or later, where this vulnerability has been addressed. Additionally, users can mitigate the risk by being cautious when using the scp command and avoiding interactions with untrusted or malicious servers. It is important to exercise caution when transferring files using scp and to verify the integrity of the files being transferred. Implementing strict file permissions and access controls on the target directory can help prevent unauthorized file manipulation. Regularly monitoring and auditing file transfers for any suspicious activity or unexpected changes can aid in detecting and preventing potential exploitation. Educating users about secure file transfer practices and the risks associated with untrusted servers can also help in mitigating the risk of unauthorized access. Conducting regular security assessments and staying informed about security updates for OpenSSH can further enhance the security posture of the system.

### Finding 21

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2018-20685**
* Severity score: **0.0** (**MEDIUM** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: Access Restriction Bypass in OpenSSH scp (Medium)  
  
#### Severity Rationale:  
The vulnerability in scp.c in the scp client in OpenSSH 7.9, allowing remote SSH servers to bypass intended access restrictions by using the filename of "." or an empty filename, resulting in modifying the permissions of the target directory on the client side, is classified as medium in severity. This vulnerability enables remote servers to manipulate access restrictions and potentially modify permissions on the client system.  
  
The medium severity rating indicates that while this vulnerability can lead to access restriction bypass and modification of directory permissions, it may not directly result in immediate critical system compromise or data loss. However, it still poses a security risk that needs to be addressed to prevent unauthorized modifications to client directories.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by remote SSH servers using the filename of "." or an empty filename, it could lead to bypassing access restrictions and modifying permissions on the target directory on the client side. This could potentially result in unauthorized changes to directory permissions, impacting the security and integrity of the client system.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the access restriction bypass vulnerability in OpenSSH scp, remote SSH servers can manipulate access restrictions and modify permissions on the client system by using specific filenames. This could lead to unauthorized modifications to directory permissions, potentially allowing attackers to manipulate files, escalate privileges, or disrupt the integrity of the client system. While the severity is medium, the exploitation of this vulnerability could lead to security risks and unauthorized actions on the client side.

#### Public exploits related to this finding

## Description of the finding

In OpenSSH 7.9, scp.c in the scp client allows remote SSH servers to bypass intended access restrictions via the filename of . or an empty filename. The impact is modifying the permissions of the target directory on the client side.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2018-20685 in OpenSSH, it is recommended to upgrade to OpenSSH version 7.9 or later, where this vulnerability has been addressed. Additionally, users can mitigate the risk by being cautious when using the scp command and avoiding interactions with untrusted or malicious servers. It is important to exercise caution when transferring files using scp and to verify the integrity of the files being transferred. Implementing strict file permissions and access controls on the target directory can help prevent unauthorized modifications. Regularly monitoring and auditing file transfers for any suspicious activity or unexpected changes can aid in detecting and preventing potential exploitation. Educating users about secure file transfer practices and the risks associated with untrusted servers can also help in mitigating the risk of unauthorized access. Conducting regular security assessments and staying informed about security updates for OpenSSH can further enhance the security posture of the system.

### Finding 22

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2018-15473**
* Severity score: **0.0** (**MEDIUM** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: User Enumeration in OpenSSH (Medium)  
  
#### Severity Rationale:  
The user enumeration vulnerability in OpenSSH through version 7.7, which allows for user enumeration due to not delaying bailout for an invalid authenticating user until after the packet containing the request has been fully parsed, is classified as medium in severity. This vulnerability can aid attackers in identifying valid user accounts on the system, potentially leading to targeted attacks or unauthorized access.  
  
The medium severity rating indicates that while this vulnerability can result in user enumeration and potentially aid attackers in identifying valid user accounts, it may not directly lead to immediate critical system compromise or data loss. However, it still poses a security risk that needs to be addressed to prevent unauthorized access and potential misuse of system resources.  
  
#### Impact on the Organization:  
If this vulnerability is exploited, attackers could enumerate valid user accounts on the system by taking advantage of the user enumeration weakness in OpenSSH. This information could be used for targeted attacks, unauthorized access attempts, or social engineering tactics. While the impact may not be as severe as critical vulnerabilities, user enumeration can still pose a risk to the organization's security posture.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the user enumeration vulnerability in OpenSSH, attackers can identify valid user accounts on the system by not delaying bailout for an invalid authenticating user until after the packet containing the request has been fully parsed. This could aid attackers in reconnaissance efforts, targeted attacks, or unauthorized access attempts. While the severity is medium, the exploitation of this vulnerability could lead to unauthorized actions and compromise the security of the SSH server and associated systems.

#### Public exploits related to this finding

#### Exploit ID 45233.

* Description: OpenSSH 2.3 &lt; 7.7 - Username Enumeration
* Download URL: https://www.exploit-db.com/exploits/45233

#### Exploit ID 45210.

* Description: OpenSSH 2.3 &lt; 7.7 - Username Enumeration (PoC)
* Download URL: https://www.exploit-db.com/exploits/45210

#### Exploit ID 45939.

* Description: OpenSSH &lt; 7.7 - User Enumeration (2)
* Download URL: https://www.exploit-db.com/exploits/45939

## Description of the finding

OpenSSH through 7.7 is prone to a user enumeration vulnerability due to not delaying bailout for an invalid authenticating user until after the packet containing the request has been fully parsed, related to auth2-gss.c, auth2-hostbased.c, and auth2-pubkey.c.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2018-15473 in OpenSSH, it is recommended to upgrade to OpenSSH version 7.7 or later, where this vulnerability has been addressed. Additionally, users can mitigate the risk by implementing proper access controls and monitoring mechanisms to detect and prevent user enumeration attempts. It is important to review and adjust the authentication settings in OpenSSH to prevent unauthorized access and user enumeration. Regularly monitoring authentication logs for any unusual patterns or repeated login attempts can aid in detecting potential enumeration attacks. Educating users about the risks of user enumeration vulnerabilities and promoting secure authentication practices can also help in mitigating the risk of unauthorized access. Conducting regular security assessments and staying informed about security updates for OpenSSH can further enhance the security posture of the system.

### Finding 23

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2016-6515**
* Severity score: **0.0** (**HIGH** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: Denial of Service via Long Password in OpenSSH (High)  
  
#### Severity Rationale:  
The vulnerability in the auth\_password function in sshd in OpenSSH before version 7.3, where password lengths for password authentication are not limited, allowing remote attackers to cause a denial of service (crypt CPU consumption) via a long string, is classified as high in severity. This vulnerability can lead to a denial of service condition due to excessive CPU consumption, impacting the availability and performance of the affected system.  
  
The high severity rating indicates that this vulnerability has the potential to result in severe consequences, including service disruption, resource exhaustion, and potential system unavailability. The ability for remote attackers to cause a denial of service through crypt CPU consumption poses a critical threat that requires immediate attention and remediation.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by remote attackers using long password strings, it could lead to a denial of service condition on the SSH server due to excessive CPU consumption during password authentication. This could result in service disruption, unresponsiveness, or system instability, impacting the availability and reliability of SSH services within the organization.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the denial of service vulnerability in OpenSSH, attackers can cause crypt CPU consumption by sending long password strings during authentication, leading to a denial of service condition on the SSH server. This could result in service degradation, unavailability, or system instability, impacting the organization's operations and potentially disrupting critical services. The high severity of this vulnerability underscores the significant risk posed by remote attackers causing a denial of service through excessive CPU consumption.

#### Public exploits related to this finding

#### Exploit ID 40888.

* Description: OpenSSH 7.2 - Denial of Service
* Download URL: https://www.exploit-db.com/exploits/40888

## Description of the finding

The auth\_password function in auth-passwd.c in sshd in OpenSSH before 7.3 does not limit password lengths for password authentication, which allows remote attackers to cause a denial of service (crypt CPU consumption) via a long string.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2016-6515 in OpenSSH, it is recommended to upgrade to OpenSSH version 7.3 or later, where this vulnerability has been addressed. Additionally, users can mitigate the risk by implementing proper password length restrictions and input validation mechanisms to prevent denial of service attacks via long password strings. It is important to review and adjust the password authentication settings in OpenSSH to enforce password length limits and prevent excessive CPU consumption. Regularly monitoring system resources and performance metrics can aid in detecting abnormal CPU consumption patterns that may indicate a denial of service attack. Educating users about the importance of using strong and secure passwords can also help in mitigating the risk of denial of service vulnerabilities. Conducting regular security assessments and staying informed about security updates for OpenSSH can further enhance the security posture of the system.

### Finding 24

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2016-1908**
* Severity score: **0.0** (**CRITICAL** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: X11 Forwarding Privilege Escalation in OpenSSH (Critical)  
  
#### Severity Rationale:  
The vulnerability in the client component of OpenSSH before version 7.2, where failed cookie generation for untrusted X11 forwarding can lead to remote X11 clients triggering a fallback and obtaining trusted X11 forwarding privileges by exploiting configuration issues on the local X11 server, is classified as critical in severity. This vulnerability allows for privilege escalation and unauthorized access to trusted X11 forwarding, posing a significant risk to the security of the affected systems.  
  
The critical severity rating indicates that this vulnerability has the potential to result in severe consequences, including unauthorized access, privilege escalation, and potential compromise of sensitive information. The ability for remote X11 clients to obtain trusted X11 forwarding privileges through this vulnerability poses a critical threat that requires immediate attention and remediation.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by remote X11 clients leveraging configuration issues on the local X11 server, it could lead to privilege escalation and unauthorized access to trusted X11 forwarding privileges. This could result in unauthorized access to sensitive information, compromise of system integrity, and potential security breaches within the organization.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the X11 forwarding privilege escalation vulnerability in OpenSSH, remote X11 clients can trigger a fallback and obtain trusted X11 forwarding privileges by exploiting configuration issues on the local X11 server. This could lead to unauthorized access, privilege escalation, and potential compromise of sensitive data through trusted X11 forwarding. The critical severity of this vulnerability underscores the significant risk posed by attackers gaining unauthorized access to trusted X11 forwarding privileges.

#### Public exploits related to this finding

## Description of the finding

The client in OpenSSH before 7.2 mishandles failed cookie generation for untrusted X11 forwarding and relies on the local X11 server for access-control decisions, which allows remote X11 clients to trigger a fallback and obtain trusted X11 forwarding privileges by leveraging configuration issues on this X11 server, as demonstrated by lack of the SECURITY extension on this X11 server.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2016-1908 in OpenSSH, it is recommended to upgrade to OpenSSH version 7.2 or later, where this vulnerability has been addressed. Additionally, users can mitigate the risk by ensuring proper configuration of X11 forwarding and access control settings on the local X11 server. It is important to review and adjust the X11 server configuration to enforce access control decisions and prevent unauthorized X11 clients from obtaining trusted X11 forwarding privileges. Implementing strict access control policies and monitoring X11 forwarding activities can help prevent unauthorized access and privilege escalation. Regularly monitoring and auditing X11 server configurations for any misconfigurations or security issues can aid in detecting and preventing potential exploitation. Educating users about secure X11 forwarding practices and the risks associated with untrusted X11 clients can also help in mitigating the risk of unauthorized access. Conducting regular security assessments and staying informed about security updates for OpenSSH can further enhance the security posture of the system.

### Finding 25

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2016-10708**
* Severity score: **0.0** (**HIGH** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: Denial of Service in OpenSSH (High)  
  
#### Severity Rationale:  
The vulnerability in sshd in OpenSSH before version 7.4, which allows remote attackers to cause a denial of service (NULL pointer dereference and daemon crash) via an out-of-sequence NEWKEYS message, is classified as high in severity. This vulnerability can lead to a denial of service condition, resulting in the crash of the SSH daemon and potential disruption of SSH services.  
  
The high severity rating indicates that this vulnerability has the potential to result in severe consequences, including service disruption, system unavailability, and potential impact on critical operations. The ability for remote attackers to cause a denial of service through a NULL pointer dereference poses a critical threat that requires immediate attention and remediation.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by remote attackers sending out-of-sequence NEWKEYS messages, it could lead to a denial of service condition on the SSH daemon, resulting in a crash and disruption of SSH services. This could impact the availability and reliability of SSH connections, potentially affecting critical operations within the organization.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the denial of service vulnerability in OpenSSH, remote attackers can trigger a NULL pointer dereference and crash the SSH daemon by sending out-of-sequence NEWKEYS messages. This could lead to service disruption, system unavailability, and potential impact on the organization's operations. The high severity of this vulnerability underscores the significant risk posed by attackers causing a denial of service through a daemon crash.

#### Public exploits related to this finding

## Description of the finding

sshd in OpenSSH before 7.4 allows remote attackers to cause a denial of service (NULL pointer dereference and daemon crash) via an out-of-sequence NEWKEYS message, as demonstrated by Honggfuzz, related to kex.c and packet.c.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2016-10708 in OpenSSH, it is recommended to upgrade to OpenSSH version 7.4 or later, where this vulnerability has been addressed. Additionally, users can mitigate the risk by implementing proper input validation mechanisms to handle out-of-sequence messages and prevent NULL pointer dereference issues. It is important to review and adjust the SSH configuration settings to ensure robust error handling and resilience against denial of service attacks. Regularly monitoring SSH daemon logs for any unusual patterns or crashes can aid in detecting potential exploitation attempts. Educating users about the risks of denial of service vulnerabilities and promoting secure SSH practices can also help in mitigating the risk of service disruptions. Conducting regular security assessments and staying informed about security updates for OpenSSH can further enhance the security posture of the system.

### Finding 26

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2016-10012**
* Severity score: **0.0** (**HIGH** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: Privilege Escalation in OpenSSH (High)  
  
#### Severity Rationale:  
The vulnerability in the shared memory manager associated with pre-authentication compression in sshd in OpenSSH before version 7.4, where a bounds check is not enforced by all compilers, allowing local users to gain privileges by leveraging access to a sandboxed privilege-separation process, is classified as high in severity. This vulnerability can lead to privilege escalation, potentially enabling local users to gain elevated privileges and compromise the security of the system.  
  
The high severity rating indicates that this vulnerability has the potential to result in severe consequences, including privilege escalation, unauthorized access, and potential compromise of sensitive information. The ability for local users to gain privileges through this vulnerability poses a critical threat that requires immediate attention and remediation.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by local users leveraging access to a sandboxed privilege-separation process, it could lead to privilege escalation and unauthorized access to sensitive resources on the system. This could result in the compromise of critical data, unauthorized actions, and potential security breaches within the organization.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the privilege escalation vulnerability in OpenSSH, local users can gain elevated privileges by leveraging access to the shared memory manager associated with pre-authentication compression. This could lead to unauthorized access, privilege escalation, and potential compromise of sensitive information on the system. The high severity of this vulnerability underscores the significant risk posed by attackers gaining elevated privileges through this vulnerability.

#### Public exploits related to this finding

## Description of the finding

The shared memory manager (associated with pre-authentication compression) in sshd in OpenSSH before 7.4 does not ensure that a bounds check is enforced by all compilers, which might allows local users to gain privileges by leveraging access to a sandboxed privilege-separation process, related to the m\_zback and m\_zlib data structures.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2016-10012 in OpenSSH, it is recommended to upgrade to OpenSSH version 7.4 or later, where this vulnerability has been addressed. Additionally, users can mitigate the risk by ensuring proper privilege separation and access controls within the SSH daemon process. It is important to review and adjust the shared memory manager settings associated with pre-authentication compression to enforce bounds checks and prevent unauthorized privilege escalation. Implementing strict access controls and monitoring mechanisms can help prevent local users from gaining unauthorized privileges. Regularly monitoring system processes and auditing privilege separation configurations can aid in detecting and preventing potential exploitation attempts. Educating users about secure SSH practices and the risks associated with unauthorized privilege escalation can also help in mitigating the risk of unauthorized access. Conducting regular security assessments and staying informed about security updates for OpenSSH can further enhance the security posture of the system.

### Finding 27

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2016-10011**
* Severity score: **0.0** (**MEDIUM** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: Sensitive Private-Key Information Disclosure in OpenSSH (Medium)  
  
#### Severity Rationale:  
The vulnerability in authfile.c in sshd in OpenSSH before version 7.4, where realloc on buffer contents is not properly considered, allowing local users to obtain sensitive private-key information by leveraging access to a privilege-separated child process, is classified as medium in severity. This vulnerability can lead to the disclosure of sensitive private-key information, potentially compromising the security of the system.  
  
The medium severity rating indicates that while this vulnerability can result in the disclosure of sensitive private-key information, it may not directly lead to immediate critical system compromise or data loss. However, it still poses a security risk that needs to be addressed to prevent unauthorized access to private keys and potential misuse of sensitive information.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by local users leveraging access to a privilege-separated child process, it could lead to the disclosure of sensitive private-key information. This could result in unauthorized access to cryptographic keys, compromising the confidentiality and integrity of encrypted communications and potentially leading to security breaches within the organization.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the vulnerability in OpenSSH that allows local users to obtain sensitive private-key information through improper handling of realloc on buffer contents, attackers can access and disclose private keys by leveraging access to a privilege-separated child process. This could lead to unauthorized access to encrypted data, compromise of cryptographic keys, and potential security risks for the organization. While the severity is medium, the exploitation of this vulnerability could lead to unauthorized access to sensitive information and compromise the security of the system.

#### Public exploits related to this finding

## Description of the finding

authfile.c in sshd in OpenSSH before 7.4 does not properly consider the effects of realloc on buffer contents, which might allow local users to obtain sensitive private-key information by leveraging access to a privilege-separated child process.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2016-10011 in OpenSSH, it is recommended to upgrade to OpenSSH version 7.4 or later, where this vulnerability has been addressed. Additionally, users can mitigate the risk by ensuring proper access controls and monitoring mechanisms to prevent unauthorized access to sensitive private-key information. It is important to review and adjust the SSH configuration settings to enforce secure buffer handling and prevent leakage of private-key information. Implementing strict access controls and privilege separation mechanisms can help prevent unauthorized access to sensitive data. Regularly monitoring system processes and auditing privilege separation configurations can aid in detecting and preventing potential exploitation attempts. Educating users about secure SSH practices and the risks associated with unauthorized access to private keys can also help in mitigating the risk of unauthorized access. Conducting regular security assessments and staying informed about security updates for OpenSSH can further enhance the security posture of the system.

### Finding 28

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2016-10009**
* Severity score: **0.0** (**HIGH** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: Arbitrary Execution of Local PKCS#11 Modules in OpenSSH (High)  
  
#### Severity Rationale:  
The untrusted search path vulnerability in ssh-agent.c in ssh-agent in OpenSSH before version 7.4, allowing remote attackers to execute arbitrary local PKCS#11 modules by leveraging control over a forwarded agent-socket, is classified as high in severity. This vulnerability enables attackers to execute arbitrary code through PKCS#11 modules, potentially leading to unauthorized access, privilege escalation, and compromise of the system.  
  
The high severity rating indicates that this vulnerability has the potential to result in severe consequences, including arbitrary code execution, unauthorized access to cryptographic modules, and potential system compromise. The ability for remote attackers to execute local PKCS#11 modules through this vulnerability poses a critical threat that requires immediate attention and remediation.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by remote attackers leveraging control over a forwarded agent-socket, it could lead to the arbitrary execution of local PKCS#11 modules, potentially resulting in unauthorized access to cryptographic keys, privilege escalation, and compromise of the system's security. This could have severe implications for the organization's confidentiality and integrity of cryptographic operations.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the untrusted search path vulnerability in OpenSSH, remote attackers can execute arbitrary local PKCS#11 modules by controlling a forwarded agent-socket. This could lead to unauthorized access to cryptographic modules, compromise of cryptographic keys, and potential system compromise. The high severity of this vulnerability underscores the significant risk posed by attackers gaining the ability to execute arbitrary code through PKCS#11 modules and compromise the security of the system.

#### Public exploits related to this finding

#### Exploit ID 40963.

* Description: OpenSSH &lt; 7.4 - agent Protocol Arbitrary Library Loading
* Download URL: https://www.exploit-db.com/exploits/40963

## Description of the finding

Untrusted search path vulnerability in ssh-agent.c in ssh-agent in OpenSSH before 7.4 allows remote attackers to execute arbitrary local PKCS#11 modules by leveraging control over a forwarded agent-socket.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2016-10009 in OpenSSH, it is recommended to upgrade to OpenSSH version 7.4 or later, where this vulnerability has been addressed. Additionally, users can mitigate the risk by implementing proper access controls and monitoring mechanisms to prevent unauthorized execution of arbitrary local PKCS#11 modules. It is important to review and secure the SSH agent configuration settings to prevent attackers from leveraging control over a forwarded agent-socket. Implementing strict access controls and monitoring the SSH agent activities can help prevent unauthorized execution of arbitrary code. Regularly monitoring system processes and auditing SSH agent configurations can aid in detecting and preventing potential exploitation attempts. Educating users about secure SSH practices and the risks associated with arbitrary library loading can also help in mitigating the risk of unauthorized access. Conducting regular security assessments and staying informed about security updates for OpenSSH can further enhance the security posture of the system.

### Finding 29

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2015-6564**
* Severity score: **0.0** (**MEDIUM** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: Use-After-Free in OpenSSH (Medium)  
  
#### Severity Rationale:  
The use-after-free vulnerability in the mm\_answer\_pam\_free\_ctx function in monitor.c in sshd in OpenSSH before version 7.0 on non-OpenBSD platforms, which might allow local users to gain privileges by leveraging control of the sshd uid to send an unexpectedly early MONITOR\_REQ\_PAM\_FREE\_CTX request, is classified as medium in severity. This vulnerability can lead to privilege escalation and unauthorized access by local users, posing a risk to the security of the system.  
  
The medium severity rating indicates that while this vulnerability can result in privilege escalation and unauthorized access, it may not directly lead to immediate critical system compromise or data loss. However, it still poses a security risk that needs to be addressed to prevent unauthorized actions and potential misuse of privileges.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by local users leveraging control of the sshd uid to send an early MONITOR\_REQ\_PAM\_FREE\_CTX request, it could lead to privilege escalation and unauthorized access on the system. This could result in unauthorized actions, compromise of sensitive data, and potential security breaches within the organization.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the use-after-free vulnerability in OpenSSH, local users can gain privileges and escalate their access by sending an early MONITOR\_REQ\_PAM\_FREE\_CTX request. This could lead to unauthorized access, privilege escalation, and potential compromise of the system's security. While the severity is medium, the exploitation of this vulnerability could lead to unauthorized actions and compromise the integrity of the system.

#### Public exploits related to this finding

## Description of the finding

Use-after-free vulnerability in the mm\_answer\_pam\_free\_ctx function in monitor.c in sshd in OpenSSH before 7.0 on non-OpenBSD platforms might allow local users to gain privileges by leveraging control of the sshd uid to send an unexpectedly early MONITOR\_REQ\_PAM\_FREE\_CTX request.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2015-6564 in OpenSSH, it is recommended to upgrade to OpenSSH version 7.0 or later, where this vulnerability has been addressed. Additionally, users can mitigate the risk by ensuring proper privilege separation and access controls within the SSH daemon process. It is important to review and adjust the SSH configuration settings to prevent unauthorized privilege escalation through use-after-free vulnerabilities. Implementing strict access controls and monitoring mechanisms can help prevent local users from gaining unauthorized privileges. Regularly monitoring system processes and auditing privilege separation configurations can aid in detecting and preventing potential exploitation attempts. Educating users about secure SSH practices and the risks associated with unauthorized privilege escalation can also help in mitigating the risk of unauthorized access. Conducting regular security assessments and staying informed about security updates for OpenSSH can further enhance the security posture of the system.

### Finding 30

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2015-6563**
* Severity score: **0.0** (**LOW** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: Impersonation Attack in OpenSSH (Low)  
  
#### Severity Rationale:  
The vulnerability in the monitor component of sshd in OpenSSH before version 7.0 on non-OpenBSD platforms, where extraneous username data in MONITOR\_REQ\_PAM\_INIT\_CTX requests allows local users to conduct impersonation attacks by leveraging SSH login access and control of the sshd uid to send a crafted MONITOR\_REQ\_PWNAM request, is classified as low in severity. This vulnerability can lead to impersonation attacks but may not directly result in critical system compromise or data loss.  
  
The low severity rating indicates that while this vulnerability can result in impersonation attacks and unauthorized access, the impact may be limited and may not have significant consequences for the organization's security posture. However, it still poses a security risk that needs to be addressed to prevent unauthorized actions and potential misuse of privileges.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by local users leveraging SSH login access and control of the sshd uid to conduct impersonation attacks, it could lead to unauthorized access and potential impersonation of other users on the system. While the impact may be low, impersonation attacks can still pose a risk to the organization's security and user privacy.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the vulnerability in OpenSSH that allows local users to conduct impersonation attacks through crafted MONITOR\_REQ\_PWNAM requests, attackers can potentially impersonate other users on the system. This could lead to unauthorized access, privilege escalation, and potential misuse of user accounts. While the severity is low, the exploitation of this vulnerability could still result in unauthorized actions and compromise the integrity of user identities on the system.

#### Public exploits related to this finding

## Description of the finding

The monitor component in sshd in OpenSSH before 7.0 on non-OpenBSD platforms accepts extraneous username data in MONITOR\_REQ\_PAM\_INIT\_CTX requests, which allows local users to conduct impersonation attacks by leveraging any SSH login access in conjunction with control of the sshd uid to send a crafted MONITOR\_REQ\_PWNAM request, related to monitor.c and monitor\_wrap.c.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2015-6563 in OpenSSH, it is recommended to upgrade to OpenSSH version 7.0 or later, where this vulnerability has been addressed. Additionally, users can mitigate the risk by ensuring proper access controls and monitoring mechanisms to prevent impersonation attacks through extraneous username data in MONITOR\_REQ\_PAM\_INIT\_CTX requests. It is important to review and adjust the SSH configuration settings to prevent unauthorized access and impersonation attempts. Implementing strict access controls and monitoring the SSH daemon activities can help prevent unauthorized access and impersonation attacks. Regularly monitoring system logs and auditing SSH activities can aid in detecting and preventing potential exploitation attempts. Educating users about secure SSH practices and the risks associated with impersonation attacks can also help in mitigating the risk of unauthorized access. Conducting regular security assessments and staying informed about security updates for OpenSSH can further enhance the security posture of the system.

### Finding 31

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2015-5600**
* Severity score: **0.0** (**HIGH** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: Brute-Force Attack and Denial of Service in OpenSSH (High)  
  
#### Severity Rationale:  
The vulnerability in the kbdint\_next\_device function in sshd in OpenSSH through version 6.9, which does not properly restrict the processing of keyboard-interactive devices within a single connection, making it easier for remote attackers to conduct brute-force attacks or cause a denial of service (CPU consumption) via a long and duplicative list in the ssh -oKbdInteractiveDevices option, is classified as high in severity. This vulnerability can lead to unauthorized access through brute-force attacks and resource exhaustion through denial of service.  
  
The high severity rating indicates that this vulnerability has the potential to result in severe consequences, including unauthorized access, service disruption, and potential impact on critical operations. The ability for remote attackers to conduct brute-force attacks and cause denial of service through CPU consumption poses a critical threat that requires immediate attention and remediation.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by remote attackers using long and duplicative lists in the ssh -oKbdInteractiveDevices option, it could lead to unauthorized access through brute-force attacks and denial of service due to CPU consumption. This could result in compromised accounts, service disruption, and potential impact on the availability and performance of the SSH server.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the vulnerability in OpenSSH that allows remote attackers to conduct brute-force attacks and cause denial of service through CPU consumption, attackers can potentially gain unauthorized access and disrupt SSH services. This could lead to compromised accounts, service unavailability, and potential security risks for the organization. The high severity of this vulnerability underscores the significant risk posed by attackers conducting brute-force attacks and causing denial of service through CPU consumption.

#### Public exploits related to this finding

## Description of the finding

The kbdint\_next\_device function in auth2-chall.c in sshd in OpenSSH through 6.9 does not properly restrict the processing of keyboard-interactive devices within a single connection, which makes it easier for remote attackers to conduct brute-force attacks or cause a denial of service (CPU consumption) via a long and duplicative list in the ssh -oKbdInteractiveDevices option, as demonstrated by a modified client that provides a different password for each pam element on this list.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2015-5600 in OpenSSH, it is recommended to upgrade to OpenSSH version 6.9 or later, where this vulnerability has been addressed. Additionally, users can mitigate the risk by implementing proper access controls and monitoring mechanisms to prevent brute-force attacks or denial of service through the processing of keyboard-interactive devices. It is important to review and adjust the SSH configuration settings to restrict the processing of keyboard-interactive devices within a single connection. Implementing rate limiting mechanisms and monitoring for unusual authentication patterns can help prevent brute-force attacks. Regularly monitoring system resources and performance metrics can aid in detecting abnormal CPU consumption patterns that may indicate a denial of service attack. Educating users about secure authentication practices and the risks associated with brute-force attacks can also help in mitigating the risk of unauthorized access. Conducting regular security assessments and staying informed about security updates for OpenSSH can further enhance the security posture of the system.

### Finding 32

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2015-5352**
* Severity score: **0.0** (**MEDIUM** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: Access Restriction Bypass in OpenSSH (Medium)  
  
#### Severity Rationale:  
The vulnerability in the x11\_open\_helper function in channels.c in ssh in OpenSSH before version 6.9, which lacks a check of the refusal deadline for X connections when ForwardX11Trusted mode is not used, making it easier for remote attackers to bypass intended access restrictions via a connection outside of the permitted time window, is classified as medium in severity. This vulnerability can lead to access restriction bypass and unauthorized X connections, potentially compromising the security of the system.  
  
The medium severity rating indicates that while this vulnerability can result in access restriction bypass and unauthorized connections, it may not directly lead to immediate critical system compromise or data loss. However, it still poses a security risk that needs to be addressed to prevent unauthorized access and potential misuse of X connections.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by remote attackers bypassing access restrictions via X connections outside of the permitted time window, it could lead to unauthorized access and potential compromise of X services. This could result in unauthorized actions, data exposure, and potential security breaches within the organization.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the access restriction bypass vulnerability in OpenSSH, remote attackers can bypass intended access restrictions and establish unauthorized X connections outside of the permitted time window. This could lead to unauthorized access, potential data exposure, and compromise of X services. While the severity is medium, the exploitation of this vulnerability could result in unauthorized actions and compromise the security of the system.

#### Public exploits related to this finding

## Description of the finding

The x11\_open\_helper function in channels.c in ssh in OpenSSH before 6.9, when ForwardX11Trusted mode is not used, lacks a check of the refusal deadline for X connections, which makes it easier for remote attackers to bypass intended access restrictions via a connection outside of the permitted time window.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2015-5352 in OpenSSH, it is recommended to upgrade to OpenSSH version 6.9 or later, where this vulnerability has been addressed. Additionally, users can mitigate the risk by ensuring proper configuration of ForwardX11Trusted mode to enforce access restrictions for X connections. It is important to review and adjust the SSH configuration settings to restrict X connections within permitted time windows and enforce access controls. Implementing strict access controls and monitoring mechanisms can help prevent unauthorized access to X connections. Regularly monitoring system logs and auditing SSH activities can aid in detecting and preventing potential exploitation attempts. Educating users about secure SSH practices and the risks associated with unauthorized access to X connections can also help in mitigating the risk of unauthorized access. Conducting regular security assessments and staying informed about security updates for OpenSSH can further enhance the security posture of the system.

### Finding 33

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2014-2653**
* Severity score: **0.0** (**MEDIUM** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: SSHFP DNS RR Checking Bypass in OpenSSH (Medium)  
  
#### Severity Rationale:  
The vulnerability in the verify\_host\_key function in sshconnect.c in the client component of OpenSSH 6.6 and earlier, which allows remote servers to trigger the skipping of SSHFP DNS RR checking by presenting an unacceptable HostCertificate, is classified as medium in severity. This vulnerability can lead to the bypass of SSHFP DNS RR checking, potentially compromising the authenticity and security of SSH connections.  
  
The medium severity rating indicates that while this vulnerability can result in the skipping of SSHFP DNS RR checking, it may not directly lead to immediate critical system compromise or data loss. However, it still poses a security risk that needs to be addressed to prevent unauthorized bypass of security mechanisms.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by remote servers triggering the skipping of SSHFP DNS RR checking, it could lead to the bypass of a security mechanism designed to verify the authenticity of SSH connections. This could result in potential man-in-the-middle attacks, unauthorized access, and compromise of the integrity of SSH communications within the organization.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the SSHFP DNS RR checking bypass vulnerability in OpenSSH, remote servers can trigger the skipping of SSHFP DNS RR checking by presenting an unacceptable HostCertificate. This could lead to the bypass of security checks, potentially enabling man-in-the-middle attacks and unauthorized access to SSH connections. While the severity is medium, the exploitation of this vulnerability could result in security risks and compromise the authenticity of SSH communications.

#### Public exploits related to this finding

## Description of the finding

The verify\_host\_key function in sshconnect.c in the client in OpenSSH 6.6 and earlier allows remote servers to trigger the skipping of SSHFP DNS RR checking by presenting an unacceptable HostCertificate.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2014-2653 in OpenSSH, it is recommended to upgrade to OpenSSH version 6.7 or later, where this vulnerability has been addressed. Additionally, users can mitigate the risk by ensuring proper configuration of SSHFP DNS RR checking and host key verification settings. It is important to review and adjust the SSH configuration settings to enforce SSHFP DNS RR checking and prevent the skipping of host key verification. Implementing strict host key verification mechanisms and monitoring for unacceptable HostCertificates can help prevent unauthorized access. Regularly monitoring system logs and auditing SSH connections can aid in detecting and preventing potential exploitation attempts. Educating users about secure SSH practices and the risks associated with skipping host key verification can also help in mitigating the risk of unauthorized access. Conducting regular security assessments and staying informed about security updates for OpenSSH can further enhance the security posture of the system.

### Finding 34

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2014-1692**
* Severity score: **0.0** (**HIGH** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: Memory Corruption in OpenSSH (High)  
  
#### Severity Rationale:  
The vulnerability in the hash\_buffer function in schnorr.c in OpenSSH through version 6.4, which occurs when the Makefile.inc is modified to enable the J-PAKE protocol and certain data structures are not initialized, might allow remote attackers to cause a denial of service (memory corruption) or have unspecified other impacts via vectors that trigger an error condition. This vulnerability is classified as high severity due to the potential for memory corruption and unspecified impacts.  
  
The high severity rating indicates that this vulnerability has the potential to result in severe consequences, including denial of service, memory corruption, and potential system instability. The ability for remote attackers to trigger memory corruption and potentially cause unspecified impacts poses a critical threat that requires immediate attention and remediation.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by remote attackers triggering memory corruption via the hash\_buffer function in OpenSSH, it could lead to denial of service, memory corruption, and potential system instability. This could impact the availability and reliability of SSH services, potentially disrupting critical operations within the organization.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the memory corruption vulnerability in OpenSSH, remote attackers can trigger memory corruption and potentially cause a denial of service or unspecified impacts by manipulating certain data structures. This could lead to system instability, service disruption, and potential security risks for the organization. The high severity of this vulnerability underscores the significant risk posed by memory corruption and potential impacts on the affected system.

#### Public exploits related to this finding

## Description of the finding

The hash\_buffer function in schnorr.c in OpenSSH through 6.4, when Makefile.inc is modified to enable the J-PAKE protocol, does not initialize certain data structures, which might allow remote attackers to cause a denial of service (memory corruption) or have unspecified other impact via vectors that trigger an error condition.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2014-1692 in OpenSSH, it is recommended to upgrade to OpenSSH version 6.5 or later, where this vulnerability has been addressed. Additionally, users can mitigate the risk by ensuring that the J-PAKE protocol is disabled in the Makefile.inc configuration to prevent the initialization issues in the hash\_buffer function. It is important to review and adjust the OpenSSH configuration settings to disable any unnecessary or experimental protocols that may introduce vulnerabilities. Implementing strict access controls and monitoring mechanisms can help prevent memory corruption and denial of service attacks. Regularly monitoring system resources and performance metrics can aid in detecting abnormal behavior that may indicate a memory corruption issue. Educating users about secure SSH practices and the risks associated with experimental protocols can also help in mitigating the risk of unauthorized access. Conducting regular security assessments and staying informed about security updates for OpenSSH can further enhance the security posture of the system.

### Finding 35

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2012-0814**
* Severity score: **0.0** (**LOW** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: Information Disclosure in OpenSSH (Low)  
  
#### Severity Rationale:  
The vulnerability in the auth\_parse\_options function in auth-options.c in sshd in OpenSSH before version 5.7, which provides debug messages containing authorized\_keys command options, allowing remote authenticated users to obtain potentially sensitive information by reading these messages, is classified as low in severity. This vulnerability can lead to information disclosure but may not directly result in critical system compromise or data loss.  
  
The low severity rating indicates that while this vulnerability can result in the disclosure of potentially sensitive information, the impact may be limited and may not have significant consequences for the organization's security posture. However, it still poses a security risk that needs to be addressed to prevent unauthorized access to sensitive data.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by remote authenticated users reading debug messages containing authorized\_keys command options, it could lead to the disclosure of potentially sensitive information. While the impact may be low, unauthorized access to such information could still pose a risk to the organization's security and confidentiality.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the information disclosure vulnerability in OpenSSH, remote authenticated users can obtain potentially sensitive information by reading debug messages containing authorized\_keys command options. This could lead to unauthorized access to command options and potentially sensitive data, posing a risk to the confidentiality of information within the organization. While the severity is low, the exploitation of this vulnerability could still result in unauthorized access to sensitive data and compromise the security of the system.

#### Public exploits related to this finding

## Description of the finding

The auth\_parse\_options function in auth-options.c in sshd in OpenSSH before 5.7 provides debug messages containing authorized\_keys command options, which allows remote authenticated users to obtain potentially sensitive information by reading these messages, as demonstrated by the shared user account required by Gitolite. NOTE: this can cross privilege boundaries because a user account may intentionally have no shell or filesystem access, and therefore may have no supported way to read an authorized\_keys file in its own home directory.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2012-0814 in OpenSSH, it is recommended to upgrade to OpenSSH version 5.7 or later, where this vulnerability has been addressed. Additionally, users can mitigate the risk by ensuring that debug messages containing authorized\_keys command options are not exposed to remote authenticated users. It is important to review and adjust the SSH configuration settings to restrict the visibility of potentially sensitive information in debug messages. Implementing strict access controls and monitoring mechanisms can help prevent unauthorized access to sensitive information. Regularly monitoring system logs and auditing SSH activities can aid in detecting and preventing potential information disclosure. Educating users about secure SSH practices and the risks associated with debug messages containing sensitive information can also help in mitigating the risk of unauthorized access. Conducting regular security assessments and staying informed about security updates for OpenSSH can further enhance the security posture of the system.

### Finding 36

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2010-5107**
* Severity score: **0.0** (**MEDIUM** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: Denial of Service in OpenSSH (Medium)  
  
#### Severity Rationale:  
The default configuration of OpenSSH through version 6.1, which enforces a fixed time limit between establishing a TCP connection and completing a login, making it easier for remote attackers to cause a denial of service (connection-slot exhaustion) by periodically making many new TCP connections, is classified as medium in severity. This vulnerability can lead to denial of service due to connection-slot exhaustion, impacting the availability and performance of the SSH service.  
  
The medium severity rating indicates that while this vulnerability can result in denial of service and connection-slot exhaustion, it may not directly lead to immediate critical system compromise or data loss. However, it still poses a security risk that needs to be addressed to prevent service disruption and potential impact on critical operations.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by remote attackers causing connection-slot exhaustion through many new TCP connections, it could lead to denial of service and impact the availability of the SSH service. This could result in service disruption, unavailability, and potential impact on the organization's operations.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the denial of service vulnerability in OpenSSH, remote attackers can cause connection-slot exhaustion by periodically making many new TCP connections within the fixed time limit enforced by the default configuration. This could lead to denial of service, service disruption, and potential impact on the availability of SSH services. While the severity is medium, the exploitation of this vulnerability could result in service disruption and impact the organization's operations.

#### Public exploits related to this finding

## Description of the finding

The default configuration of OpenSSH through 6.1 enforces a fixed time limit between establishing a TCP connection and completing a login, which makes it easier for remote attackers to cause a denial of service (connection-slot exhaustion) by periodically making many new TCP connections.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2010-5107 in OpenSSH, it is recommended to upgrade to OpenSSH version 6.2 or later, where this vulnerability has been addressed. Additionally, users can mitigate the risk by adjusting the configuration settings in OpenSSH to modify the time limit between establishing a TCP connection and completing a login. It is important to review and adjust the SSH configuration settings to prevent connection-slot exhaustion attacks by adjusting the time limits for completing logins. Implementing rate limiting mechanisms and monitoring for excessive TCP connections can help prevent denial of service attacks. Regularly monitoring system resources and network connections can aid in detecting abnormal connection patterns that may indicate an attack. Educating users about secure SSH practices and the risks associated with denial of service attacks can also help in mitigating the risk of service disruptions. Conducting regular security assessments and staying informed about security updates for OpenSSH can further enhance the security posture of the system.

### Finding 37

* Affected IP: **192.168.254.136**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2008-3259**
* Severity score: **0.0** (**LOW** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: X11 Forwarding Port Hijacking in OpenSSH (Low)  
  
#### Severity Rationale:  
The vulnerability in OpenSSH before version 5.1, where the SO\_REUSEADDR socket option is set when the X11UseLocalhost configuration setting is disabled, allowing local users on some platforms to hijack the X11 forwarding port via a bind to a single IP address, as demonstrated on the HP-UX platform, is classified as low in severity. This vulnerability can lead to X11 forwarding port hijacking but may not directly result in critical system compromise or data loss.  
  
The low severity rating indicates that while this vulnerability can result in X11 forwarding port hijacking, the impact may be limited and may not have significant consequences for the organization's security posture. However, it still poses a security risk that needs to be addressed to prevent unauthorized access to X11 forwarding ports.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by local users hijacking the X11 forwarding port via a bind to a single IP address, it could lead to unauthorized access to X11 forwarding services. While the impact may be low, unauthorized access to X11 forwarding ports could still pose a risk to the organization's security and integrity.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the X11 forwarding port hijacking vulnerability in OpenSSH, local users can hijack the X11 forwarding port via a bind to a single IP address when the SO\_REUSEADDR socket option is set. This could lead to unauthorized access to X11 forwarding services and potential misuse of X11 connections. While the severity is low, the exploitation of this vulnerability could result in unauthorized access to X11 forwarding ports and compromise the security of the system.

#### Public exploits related to this finding

## Description of the finding

OpenSSH before 5.1 sets the SO\_REUSEADDR socket option when the X11UseLocalhost configuration setting is disabled, which allows local users on some platforms to hijack the X11 forwarding port via a bind to a single IP address, as demonstrated on the HP-UX platform.

## Mitigation steps

Mitigation steps:  
To mitigate the vulnerability described in CVE-2008-3259 in OpenSSH, it is recommended to upgrade to OpenSSH version 5.1 or later, where this vulnerability has been addressed. Additionally, users can mitigate the risk by ensuring that the X11UseLocalhost configuration setting is enabled to prevent local users from hijacking the X11 forwarding port. It is important to review and adjust the SSH configuration settings to enforce secure X11 forwarding practices and prevent unauthorized access to X11 forwarding ports. Implementing strict access controls and monitoring mechanisms can help prevent unauthorized access to X11 forwarding ports. Regularly monitoring system logs and auditing SSH activities can aid in detecting and preventing potential exploitation attempts. Educating users about secure SSH practices and the risks associated with X11 forwarding vulnerabilities can also help in mitigating the risk of unauthorized access. Conducting regular security assessments and staying informed about security updates for OpenSSH can further enhance the security posture of the system.