Logotipo

Descripción generada automáticamente

C15

Security Assessment  
Findings Report

Business Confidential

* Date of the project: 2024-12-20
* 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 2024-12-20, 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 |
| 127.0.0.1 | 22 | ssh | 4.7p1 Debian 8ubuntu1 | CVE-2023-48795 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-2017-15906 CVE-2016-6515 CVE-2016-6210 CVE-2016-3115 CVE-2016-20012 CVE-2016-1908 CVE-2016-10708 CVE-2016-10012 CVE-2016-10011 CVE-2016-10010 CVE-2016-10009 CVE-2015-8325 CVE-2015-6564 CVE-2015-6563 CVE-2015-5600 CVE-2015-5352 CVE-2014-2653 CVE-2014-2532 CVE-2014-1692 CVE-2012-0814 CVE-2011-5000 CVE-2011-4327 CVE-2010-5107 CVE-2010-4755 CVE-2010-4478 CVE-2008-3259 CVE-2008-1657 |

## 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 2024-12-20. 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 127.0.0.1:  
 - Open Ports: 22 (SSH)  
 - Service: SSH (OpenSSH 4.7p1 Debian 8ubuntu1)  
 - Vulnerabilities:  
 - CVE-2023-48795 (Medium): This vulnerability allows remote attackers to bypass integrity checks in the SSH transport protocol, potentially downgrading or disabling security features. It poses a medium risk to the organization.  
 - CVE-2023-51385 (Medium): This vulnerability can lead to OS command injection if a user or host name contains shell metacharacters, potentially compromising the system.  
 - CVE-2023-38408 (Critical): This vulnerability in the PKCS#11 feature of ssh-agent can result in remote code execution if an agent is forwarded to an attacker-controlled system.  
  
2. Health Posture of SSH Service:  
 - The SSH service on port 22 is running an outdated version of OpenSSH with multiple vulnerabilities, including critical and medium severity issues. These vulnerabilities can lead to remote code execution, OS command injection, and integrity bypass, posing a significant risk to the organization.  
  
3. Mitigation Recommendations:  
 - Update OpenSSH: Upgrade the OpenSSH version to the latest release to patch known vulnerabilities and improve the security of the SSH service.  
 - Implement Access Controls: Enforce strict access controls and user permissions to mitigate the risk of OS command injection and unauthorized access.  
 - Monitor SSH Traffic: Implement monitoring and logging mechanisms to detect and respond to suspicious activities or potential attacks targeting the SSH service.  
  
Additional Information:  
- Regular security assessments and patch management are essential to address vulnerabilities and enhance the overall security posture of the organization.  
- Employee training on secure SSH practices and awareness of potential threats can help in preventing security incidents related to SSH vulnerabilities.

## 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: **127.0.0.1**
* 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: SSH Terrapin Attack (Medium)  
  
#### Severity Rationale:  
The severity of the SSH Terrapin Attack vulnerability is classified as medium due to the potential impact it can have on the security of the SSH communication protocol. This vulnerability allows remote attackers to bypass integrity checks and manipulate the connection to downgrade or disable certain security features, leading to a compromise in the confidentiality, integrity, and availability of the communication.  
  
The medium severity indicates that while the vulnerability poses a significant risk, it may not have an immediate and widespread impact on the organization's security posture. However, it still requires prompt attention and mitigation to prevent potential exploitation by threat actors.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by attackers, it could lead to unauthorized access to sensitive information, manipulation of data in transit, and potential disruption of secure communication channels. The attackers could compromise the confidentiality of data exchanged over SSH connections, tamper with the integrity of the communication, and potentially launch further attacks within the organization's network.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the SSH Terrapin Attack vulnerability, attackers could manipulate the SSH connection to bypass security checks, downgrade encryption algorithms, and disable critical security features. This could allow them to eavesdrop on sensitive communications, inject malicious payloads, and potentially gain unauthorized access to systems and data exchanged over SSH connections. The attackers could use this vulnerability to conduct man-in-the-middle attacks, intercept sensitive information, and compromise the overall security of the organization's network.  
  
In conclusion, while the SSH Terrapin Attack vulnerability is classified as medium severity, it still poses a significant risk to the organization's security and requires immediate remediation to prevent potential exploitation by malicious actors.

#### 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 related to the SSH transport protocol with certain OpenSSH extensions, it is recommended to update the affected OpenSSH versions to version 9.6 or later. This update should address the mishandling of the handshake phase and sequence numbers in the SSH Binary Packet Protocol (BPP) that leads to the Terrapin attack. Additionally, it is crucial to regularly monitor for security updates and patches for the SSH implementations listed in the vulnerability description to ensure that any potential security weaknesses are promptly addressed.

### Finding 2

* Affected IP: **127.0.0.1**
* 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: SSH OS Command Injection (Medium)  
  
#### Severity Rationale:  
The severity of the SSH OS Command Injection vulnerability is classified as medium due to the potential risk it poses to the security of the SSH protocol. This vulnerability allows for the injection of operating system commands if a user name or host name contains shell metacharacters and is referenced by an expansion token in specific scenarios.   
  
The medium severity indicates that while the vulnerability has the potential to be exploited by attackers to execute arbitrary commands on the target system, it may require specific conditions to be met for successful exploitation. However, it still presents a significant risk to the confidentiality, integrity, and availability of the system.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by attackers, it could lead to unauthorized execution of arbitrary commands on the target system, potentially resulting in data exfiltration, system compromise, and disruption of operations. The attackers could leverage this vulnerability to gain unauthorized access to sensitive information, escalate privileges, and carry out further malicious activities within the organization's network.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the SSH OS Command Injection vulnerability, attackers could inject malicious commands into the SSH connection, leading to unauthorized execution of commands on the target system. This could allow them to manipulate system configurations, install malware, exfiltrate sensitive data, and potentially take control of the affected system. The attackers could use this vulnerability to achieve persistence, move laterally within the network, and escalate their privileges.  
  
In conclusion, while the SSH OS Command Injection vulnerability is classified as medium severity, it still poses a significant risk to the organization's security posture and requires immediate attention to prevent potential exploitation by threat actors.

#### 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 related to OS command injection in ssh in OpenSSH before version 9.6, it is recommended to sanitize and validate user names and host names to prevent the inclusion of shell metacharacters. This can be achieved by implementing input validation mechanisms to ensure that user input does not contain any malicious characters that could be interpreted as commands by the shell. Additionally, it is advisable to restrict access to untrusted Git repositories and carefully review any submodules for potential shell metacharacters in user names or host names before referencing them in expansion tokens. Regularly updating OpenSSH to the latest version can also help mitigate this vulnerability by addressing any known security issues and implementing necessary fixes.

### Finding 3

* Affected IP: **127.0.0.1**
* 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: SSH PKCS#11 Remote Code Execution (Critical)  
  
#### Severity Rationale:  
The severity of the SSH PKCS#11 Remote Code Execution vulnerability is classified as critical due to the high impact it can have on the security of the SSH protocol. This vulnerability arises from an insufficiently trustworthy search path in the PKCS#11 feature of ssh-agent in OpenSSH before 9.3p2, which could lead to remote code execution if an agent is forwarded to an attacker-controlled system.   
  
The critical severity indicates that this vulnerability poses a severe risk to the confidentiality, integrity, and availability of the system. Exploitation of this vulnerability could result in unauthorized execution of arbitrary code on the target system, potentially leading to complete compromise of the system and sensitive data.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by attackers, it could lead to remote code execution on the target system, allowing them to take full control of the system, exfiltrate sensitive data, and disrupt operations. The attackers could leverage this vulnerability to escalate privileges, install backdoors, and carry out further malicious activities within the organization's network.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the SSH PKCS#11 Remote Code Execution vulnerability, attackers could forward a malicious agent to an attacker-controlled system, leading to the execution of arbitrary code on the target system. This could allow them to compromise the confidentiality of data, manipulate system configurations, and potentially gain persistent access to the network. The attackers could use this vulnerability to launch further attacks, move laterally within the network, and maintain control over the compromised systems.  
  
In conclusion, the critical severity of the SSH PKCS#11 Remote Code Execution vulnerability highlights the urgent need for immediate remediation to prevent potential exploitation by threat actors and protect the organization's assets and sensitive information.

#### 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 related to the PKCS#11 feature in ssh-agent in OpenSSH before version 9.3p2, it is crucial to update OpenSSH to version 9.3p2 or later. This update should address the insufficiently trustworthy search path issue that could lead to remote code execution if an agent is forwarded to an attacker-controlled system. Additionally, it is recommended to review and secure the search path used by ssh-agent to ensure that only trusted directories are included, and code from untrusted locations is not loaded into ssh-agent. Regularly monitoring for security updates and patches for OpenSSH is essential to stay protected against critical vulnerabilities like this one.

### Finding 4

* Affected IP: **127.0.0.1**
* 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: OpenSSH Authentication Issue (Low)  
  
#### Severity Rationale:  
The severity of the OpenSSH Authentication Issue vulnerability is classified as low due to the limited impact it has on the security of the SSH protocol. This vulnerability arises when a client is using public-key authentication with agent forwarding without the -oLogLevel=verbose option, and an attacker has silently modified the server to support the None authentication option.   
  
The low severity indicates that while this vulnerability can lead to confusion and potential misuse of authentication options, it does not directly result in unauthorized access or compromise of the system. The impact of this vulnerability is limited and does not pose a significant risk to the organization's security posture.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by attackers, it could lead to confusion regarding the authentication process, potentially allowing for misinterpretation of authentication requests. However, this vulnerability does not directly result in unauthorized access to the system or compromise of sensitive data. The impact is more related to user experience and potential confusion rather than a direct security threat.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH Authentication Issue vulnerability, attackers could potentially confuse users regarding the authentication process, leading to uncertainty about the authentication requests. This could result in users inadvertently allowing the server to connect to a different server on their behalf, potentially leading to unintended actions or connections. However, this vulnerability does not provide a direct avenue for unauthorized access or compromise of the system.  
  
In conclusion, the low severity of the OpenSSH Authentication Issue vulnerability indicates that while it may lead to confusion and potential misuse of authentication options, it does not pose a significant security risk to the organization. It is important to address this vulnerability to ensure clarity in the authentication process and prevent potential user confusion.

#### 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 vulnerability described in CVE-2021-36368 related to OpenSSH before version 8.9, it is recommended to update OpenSSH to version 8.9 or later. This update should address the issue where a client using public-key authentication with agent forwarding could be vulnerable to an attacker silently modifying the server to support the None authentication option. Additionally, it is advisable to enable the -oLogLevel=verbose option when using public-key authentication with agent forwarding to increase visibility and detect any suspicious authentication attempts. Regularly updating OpenSSH to the latest version and following best practices for secure authentication mechanisms can help prevent potential exploitation of this low-severity vulnerability.

### Finding 5

* Affected IP: **127.0.0.1**
* 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: OpenSSH SCP Command Injection (High)  
  
#### Severity Rationale:  
The severity of the OpenSSH SCP Command Injection vulnerability is classified as high due to the significant risk it poses to the security of the SCP (Secure Copy Protocol) functionality in OpenSSH. This vulnerability allows for command injection in the scp.c toremote function, enabling attackers to execute arbitrary commands by using backtick characters in the destination argument.  
  
The high severity indicates that this vulnerability can be easily exploited by attackers to execute malicious commands on the target system, potentially leading to unauthorized access, data exfiltration, and system compromise. The lack of validation of "anomalous argument transfers" by the vendor further increases the risk associated with this vulnerability.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by attackers, it could lead to the execution of arbitrary commands on the target system, allowing them to take control of the system, manipulate data, and potentially escalate privileges. The attackers could leverage this vulnerability to compromise the confidentiality, integrity, and availability of the organization's systems and data.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH SCP Command Injection vulnerability, attackers can inject malicious commands into the destination argument of SCP commands, leading to the execution of arbitrary commands on the target system. This could enable them to perform unauthorized actions, escalate privileges, and potentially gain persistent access to the system. The lack of validation by the vendor further facilitates the exploitation of this vulnerability.  
  
In conclusion, the high severity of the OpenSSH SCP Command Injection vulnerability highlights the urgent need for immediate remediation to prevent potential exploitation by attackers and protect the organization's systems and data from unauthorized access and compromise.

#### 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 related to command injection in scp in OpenSSH through version 8.3p1, it is recommended to update OpenSSH to a version beyond 8.3p1. This update should address the vulnerability in the scp.c toremote function that allows for command injection, particularly when backtick characters are present in the destination argument. Additionally, it is crucial to sanitize and validate input arguments to prevent the injection of malicious commands. Implementing strict input validation mechanisms and avoiding the use of potentially dangerous characters in scp commands can help mitigate the risk of exploitation. Regularly updating OpenSSH to the latest version is essential to ensure that known vulnerabilities are patched and security is maintained at a high level.

### Finding 6

* Affected IP: **127.0.0.1**
* 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: OpenSSH SCP Arbitrary File Overwrite (Medium)  
  
#### Severity Rationale:  
The severity of the OpenSSH SCP Arbitrary File Overwrite vulnerability is classified as medium due to the potential risk it poses to the security of the SCP (Secure Copy Protocol) implementation in OpenSSH. This vulnerability allows a malicious SCP server or a Man-in-The-Middle attacker to overwrite arbitrary files in the SCP client target directory by manipulating the object names returned by the server.  
  
The medium severity indicates that while this vulnerability can lead to unauthorized file overwrites and potential manipulation of subdirectories, it may require specific conditions to be met for successful exploitation. However, it still presents a significant risk to the confidentiality and integrity of the client's files and directories.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by attackers, it could lead to the unauthorized modification or deletion of critical files in the SCP client target directory. This could result in the compromise of sensitive data, manipulation of system configurations, and potential disruption of operations. The attackers could leverage this vulnerability to escalate privileges and carry out further malicious activities on the target system.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH SCP Arbitrary File Overwrite vulnerability, attackers can manipulate the object names returned by the SCP server to overwrite arbitrary files in the client target directory. This could allow them to modify important configuration files, overwrite sensitive data, and potentially escalate their privileges on the target system. If the recursive operation (-r) is performed, the attackers could also manipulate subdirectories, leading to further compromise of the system.  
  
In conclusion, while the OpenSSH SCP Arbitrary File Overwrite vulnerability is classified as medium severity, it still poses a significant risk to the organization's security posture and requires immediate attention to prevent potential exploitation by malicious actors. It is crucial to address this vulnerability to protect the confidentiality and integrity of the organization's files and directories.

#### 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 related to arbitrary file overwriting in OpenSSH 7.9's scp implementation, it is recommended to update OpenSSH to a version beyond 7.9. This update should address the issue where a malicious scp server or Man-in-The-Middle attacker could overwrite arbitrary files in the scp client target directory. Additionally, it is important to ensure that the scp client performs thorough validation of the object names returned by the server to prevent such attacks. Implementing strict validation checks and avoiding recursive operations (-r) when interacting with potentially untrusted scp servers can help reduce the risk of file manipulation. Regularly updating OpenSSH to the latest version is crucial to protect against known vulnerabilities and maintain a secure file transfer environment.

### Finding 7

* Affected IP: **127.0.0.1**
* 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: OpenSSH Arbitrary Output Manipulation (Medium)  
  
#### Severity Rationale:  
The severity of the OpenSSH Arbitrary Output Manipulation vulnerability is classified as medium due to the potential risk it poses to the integrity and reliability of the SSH client output in OpenSSH. This vulnerability allows a malicious server or a Man-in-The-Middle attacker to manipulate the client output by sending arbitrary stderr output, potentially using ANSI control codes to hide additional files being transferred.  
  
The medium severity indicates that while this vulnerability can lead to manipulation of client output and potential obfuscation of additional files being transferred, it may not directly result in unauthorized access or compromise of the system. However, it still presents a significant risk to the integrity and trustworthiness of the SSH client output.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by attackers, it could lead to confusion and potential manipulation of the client output, making it difficult for users to accurately interpret the information displayed. This could result in users being misled about the actions being performed by the SSH client, potentially leading to security incidents or unauthorized activities.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH Arbitrary Output Manipulation vulnerability, attackers can send arbitrary stderr output to the client, using ANSI control codes to hide additional files being transferred or manipulate the displayed information. This could lead to confusion among users, potentially allowing attackers to hide malicious activities or obfuscate the true nature of the SSH session. While this vulnerability may not directly lead to unauthorized access, it can impact the trustworthiness of the client output.  
  
In conclusion, the medium severity of the OpenSSH Arbitrary Output Manipulation vulnerability highlights the importance of ensuring the integrity and reliability of the SSH client output. While it may not directly result in unauthorized access, it can still lead to confusion and potential manipulation of information, requiring attention to prevent potential exploitation by malicious actors.

#### 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 related to arbitrary stderr output manipulation in OpenSSH 7.9, it is recommended to update OpenSSH to a version beyond 7.9. This update should address the issue where a malicious server or Man-in-The-Middle attacker could manipulate the client output by sending arbitrary stderr output, potentially using ANSI control codes to hide additional files being transferred. Additionally, it is important to ensure that the client properly handles and sanitizes the output received from the server to prevent manipulation and unauthorized actions. Implementing strict output validation mechanisms and avoiding interactions with untrusted servers can help mitigate the risk of exploitation. Regularly updating OpenSSH to the latest version is essential to address known vulnerabilities and maintain a secure communication environment.

### Finding 8

* Affected IP: **127.0.0.1**
* 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: OpenSSH Progress Display Manipulation (Medium)  
  
#### Severity Rationale:  
The severity of the OpenSSH Progress Display Manipulation vulnerability is classified as medium due to the potential risk it poses to the integrity and reliability of the progress display in OpenSSH. This vulnerability allows a malicious server or a Man-in-The-Middle attacker to manipulate the client output by employing crafted object names to hide additional files being transferred, using ANSI control codes.  
  
The medium severity indicates that while this vulnerability can lead to manipulation of the progress display and potential obfuscation of additional files being transferred, it may not directly result in unauthorized access or compromise of the system. However, it still presents a significant risk to the integrity and trustworthiness of the progress display.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by attackers, it could lead to confusion and potential manipulation of the progress display, making it difficult for users to accurately track the transfer of files and understand the actions being performed by the SSH client. This could result in users being misled about the progress of file transfers, potentially leading to security incidents or unauthorized activities.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH Progress Display Manipulation vulnerability, attackers can employ crafted object names to manipulate the client output, using ANSI control codes to hide additional files being transferred or manipulate the progress display. This could lead to confusion among users, potentially allowing attackers to hide malicious activities or obfuscate the true progress of file transfers. While this vulnerability may not directly lead to unauthorized access, it can impact the trustworthiness of the progress display.  
  
In conclusion, the medium severity of the OpenSSH Progress Display Manipulation vulnerability underscores the importance of ensuring the integrity and reliability of the progress display during file transfers. While it may not directly result in unauthorized access, it can still lead to confusion and potential manipulation of information, requiring attention to prevent potential exploitation by malicious actors.

#### 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 related to missing character encoding in the progress display in OpenSSH 7.9, it is recommended to update OpenSSH to a version beyond 7.9. This update should address the issue where a malicious server or Man-in-The-Middle attacker could manipulate the client output by employing crafted object names to hide additional files being transferred using ANSI control codes. Additionally, it is important to ensure that character encoding is properly handled in the progress display to prevent manipulation of the client output. Implementing strict input validation and output encoding mechanisms can help mitigate the risk of unauthorized actions and data manipulation. Regularly updating OpenSSH to the latest version is crucial to address known vulnerabilities and maintain a secure communication environment.

### Finding 9

* Affected IP: **127.0.0.1**
* 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: OpenSSH SCP Access Restriction Bypass (Medium)  
  
#### Severity Rationale:  
The severity of the OpenSSH SCP Access Restriction Bypass vulnerability is classified as medium due to the potential risk it poses to the integrity and security of file transfers using the SCP (Secure Copy Protocol) client in OpenSSH. This vulnerability allows remote SSH servers to bypass intended access restrictions by using the filename of "." or an empty filename, potentially leading to modifications of permissions on the target directory on the client side.  
  
The medium severity indicates that while this vulnerability can lead to unauthorized modifications of permissions on the client-side target directory, it may not directly result in complete system compromise or unauthorized access to sensitive data. However, it still presents a significant risk to the confidentiality and integrity of the client-side file system.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by attackers, it could lead to unauthorized modifications of permissions on the target directory on the client side, potentially impacting the confidentiality and integrity of files stored in that directory. This could result in unauthorized access to files, manipulation of permissions, and potential disruption of operations on the client system.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH SCP Access Restriction Bypass vulnerability, attackers can use the filename of "." or an empty filename to bypass intended access restrictions and modify permissions on the target directory on the client side. This could allow them to manipulate file permissions, potentially granting unauthorized access to files or disrupting the normal operation of the client system. While this vulnerability may not directly lead to complete system compromise, it can impact the security of the client-side file system.  
  
In conclusion, the medium severity of the OpenSSH SCP Access Restriction Bypass vulnerability highlights the importance of ensuring proper access restrictions and permissions during file transfers using the SCP client. While it may not directly result in unauthorized access to sensitive data, it can still lead to unauthorized modifications of permissions, requiring attention to prevent potential exploitation by malicious actors.

#### 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 related to bypassing access restrictions in OpenSSH 7.9's scp client, it is recommended to update OpenSSH to a version beyond 7.9. This update should address the issue where remote SSH servers could bypass intended access restrictions by using the filename of '.' or an empty filename. This could lead to modifying the permissions of the target directory on the client side. Additionally, it is important to validate and sanitize filenames received from remote servers to prevent unauthorized access and modifications. Implementing strict access controls and avoiding interactions with untrusted servers can help reduce the risk of exploitation. Regularly updating OpenSSH to the latest version is essential to address known vulnerabilities and maintain a secure file transfer environment.

### Finding 10

* Affected IP: **127.0.0.1**
* 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: OpenSSH User Enumeration (Medium)  
  
#### Severity Rationale:  
The severity of the OpenSSH User Enumeration vulnerability is classified as medium due to the potential risk it poses to the confidentiality and security of user authentication in OpenSSH. This vulnerability allows for user enumeration by not delaying bailout for an invalid authenticating user until after the packet containing the request has been fully parsed, affecting modules such as auth2-gss.c, auth2-hostbased.c, and auth2-pubkey.c.  
  
The medium severity indicates that while this vulnerability can lead to user enumeration and potentially disclose valid user accounts, it may not directly result in unauthorized access or compromise of sensitive data. However, it still presents a significant risk to the confidentiality and privacy of user information.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by attackers, it could lead to the enumeration of valid user accounts within the OpenSSH environment, potentially providing valuable information for targeted attacks or unauthorized access attempts. The disclosure of valid user accounts could aid attackers in further reconnaissance and exploitation of the organization's systems.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH User Enumeration vulnerability, attackers can enumerate valid user accounts by exploiting the timing issue in the authentication process. This could allow them to identify valid user accounts within the OpenSSH environment, potentially aiding in targeted attacks or unauthorized access attempts. While this vulnerability may not directly lead to unauthorized access, it can provide valuable information for attackers.  
  
In conclusion, the medium severity of the OpenSSH User Enumeration vulnerability underscores the importance of protecting user authentication information and preventing unauthorized disclosure of valid user accounts. While it may not directly result in unauthorized access, it can still lead to user enumeration and aid attackers in further exploitation, requiring attention to prevent potential exploitation by malicious actors.

#### 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 related to user enumeration in OpenSSH through version 7.7, it is recommended to update OpenSSH to a version beyond 7.7. This update should address the issue where a user enumeration vulnerability exists due to not delaying bailout for an invalid authenticating user until after the packet containing the request has been fully parsed. Additionally, it is important to implement proper error handling mechanisms to prevent leaking information that could aid in user enumeration. By updating to the latest version of OpenSSH and ensuring robust error handling practices, the risk of user enumeration can be mitigated. Regularly monitoring for security updates and patches for OpenSSH is essential to address known vulnerabilities and maintain a secure authentication process.

### Finding 11

* Affected IP: **127.0.0.1**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2017-15906**
* Severity score: **0.0** (**MEDIUM** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: OpenSSH SFTP Server Zero-Length File Creation (Medium)  
  
#### Severity Rationale:  
The severity of the OpenSSH SFTP Server Zero-Length File Creation vulnerability is classified as medium due to the potential risk it poses to the integrity and security of file operations in OpenSSH. This vulnerability in the process\_open function in sftp-server.c allows attackers to create zero-length files even in readonly mode, potentially leading to unauthorized file creation.  
  
The medium severity indicates that while this vulnerability can lead to the creation of zero-length files, it may not directly result in complete system compromise or unauthorized access to sensitive data. However, it still presents a significant risk to the integrity and availability of the file system.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by attackers, it could lead to the unauthorized creation of zero-length files, potentially impacting the availability and organization of files within the system. While the creation of zero-length files may not seem critical, it could disrupt normal operations and lead to confusion or potential misuse of the file system.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH SFTP Server Zero-Length File Creation vulnerability, attackers can create zero-length files even in readonly mode, bypassing intended restrictions. This could allow them to manipulate the file system, potentially disrupting operations, causing confusion, or creating a foothold for further attacks. While this vulnerability may not directly lead to unauthorized access, it can impact the integrity and availability of the file system.  
  
In conclusion, the medium severity of the OpenSSH SFTP Server Zero-Length File Creation vulnerability highlights the importance of enforcing proper file operation restrictions and preventing unauthorized file creation. While it may not directly result in unauthorized access to sensitive data, it can still lead to unauthorized file creation and disruption of normal operations, requiring attention to prevent potential exploitation by malicious actors.

#### 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 related to allowing write operations in readonly mode in OpenSSH before version 7.6, it is recommended to update OpenSSH to a version beyond 7.6. This update should address the issue where the process\_open function in sftp-server.c does not properly prevent write operations in readonly mode, leading to the creation of zero-length files by attackers. Additionally, it is important to enforce proper permissions and access controls to restrict write operations in readonly mode effectively. By updating to the latest version of OpenSSH and configuring the server to enforce readonly mode restrictions correctly, the risk of unauthorized file creation can be mitigated. Regularly updating OpenSSH and implementing secure configuration practices are essential to prevent exploitation of known vulnerabilities and maintain a secure file transfer environment.

### Finding 12

* Affected IP: **127.0.0.1**
* 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: OpenSSH Password Length DoS (High)  
  
#### Severity Rationale:  
The severity of the OpenSSH Password Length DoS vulnerability is classified as high due to the significant risk it poses to the availability and performance of the SSH service in OpenSSH. This vulnerability in the auth\_password function in auth-passwd.c allows remote attackers to cause a denial of service (DoS) by sending a long string as part of the password authentication process, leading to excessive CPU consumption for password hashing.  
  
The high severity indicates that this vulnerability can be easily exploited by remote attackers to disrupt the availability of the SSH service, potentially leading to service unavailability and performance degradation. The impact of this vulnerability on the availability of the system is critical.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by attackers, it could lead to a denial of service condition on the SSH service, rendering it unavailable for legitimate users. The excessive CPU consumption caused by processing long password strings could result in performance degradation, system unresponsiveness, and disruption of critical SSH services.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH Password Length DoS vulnerability, attackers can send long strings as part of the password authentication process, causing excessive CPU consumption for password hashing and leading to a denial of service condition. This could render the SSH service unavailable, impacting legitimate users' ability to access the system and disrupting critical operations. The attacker could leverage this vulnerability to disrupt services and potentially cause system instability.  
  
In conclusion, the high severity of the OpenSSH Password Length DoS vulnerability highlights the urgent need for immediate remediation to prevent potential exploitation by remote attackers and protect the availability and performance of the SSH service. It is crucial to enforce password length limitations to mitigate the risk of denial of service attacks and ensure the reliability of the SSH service.

#### 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 related to unlimited password lengths for password authentication in OpenSSH before version 7.3, it is recommended to update OpenSSH to a version beyond 7.3. This update should address the issue where the auth\_password function in auth-passwd.c does not limit password lengths, leading to a denial of service (crypt CPU consumption) via a long string. Additionally, it is important to enforce password length restrictions and implement proper input validation mechanisms to prevent excessive CPU consumption due to long passwords. By updating to the latest version of OpenSSH and configuring password length limits, the risk of denial of service attacks can be mitigated. Regularly updating OpenSSH and following secure authentication practices are essential to prevent exploitation of known vulnerabilities and maintain a secure SSH environment.

### Finding 13

* Affected IP: **127.0.0.1**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2016-6210**
* Severity score: **0.0** (**MEDIUM** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: OpenSSH User Enumeration via Timing Attack (Medium)  
  
#### Severity Rationale:  
The severity of the OpenSSH User Enumeration via Timing Attack vulnerability is classified as medium due to the potential risk it poses to the confidentiality and security of user authentication in OpenSSH. This vulnerability in sshd before version 7.3 allows remote attackers to enumerate users by leveraging the timing difference between responses when a large password is provided, specifically when using SHA256 or SHA512 for user password hashing.  
  
The medium severity indicates that while this vulnerability can lead to user enumeration and potentially disclose valid user accounts, it may not directly result in unauthorized access or compromise of sensitive data. However, it still presents a significant risk to the confidentiality and privacy of user information.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by attackers, it could lead to the enumeration of valid user accounts within the OpenSSH environment, potentially providing valuable information for targeted attacks or unauthorized access attempts. The disclosure of valid user accounts could aid attackers in further reconnaissance and exploitation of the organization's systems.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH User Enumeration via Timing Attack vulnerability, attackers can leverage the timing difference between responses when a large password is provided to enumerate users in the OpenSSH environment. This could allow them to identify valid user accounts, potentially aiding in targeted attacks or unauthorized access attempts. While this vulnerability may not directly lead to unauthorized access, it can provide valuable information for attackers.  
  
In conclusion, the medium severity of the OpenSSH User Enumeration via Timing Attack vulnerability underscores the importance of protecting user authentication information and preventing unauthorized disclosure of valid user accounts. While it may not directly result in unauthorized access, it can still lead to user enumeration and aid attackers in further exploitation, requiring attention to prevent potential exploitation by malicious actors.

#### 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 related to user enumeration in OpenSSH before version 7.3 when using SHA256 or SHA512 for user password hashing, it is recommended to update OpenSSH to a version beyond 7.3. This update should address the issue where sshd uses BLOWFISH hashing on a static password when the username does not exist, allowing remote attackers to enumerate users by exploiting timing differences in responses. Additionally, it is important to implement consistent response times for authentication attempts to prevent timing-based attacks for user enumeration. By updating to the latest version of OpenSSH and ensuring uniform response times, the risk of user enumeration can be mitigated. Regularly monitoring for security updates and patches for OpenSSH is essential to address known vulnerabilities and maintain a secure authentication process.

### Finding 14

* Affected IP: **127.0.0.1**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2016-3115**
* Severity score: **0.0** (**MEDIUM** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: OpenSSH CRLF Injection in session.c (Medium)  
  
#### Severity Rationale:  
The severity of the OpenSSH CRLF Injection vulnerability in session.c is classified as medium due to the potential risk it poses to the security and integrity of the SSH service in OpenSSH. This vulnerability allows remote authenticated users to bypass intended shell-command restrictions by exploiting CRLF injection vulnerabilities in the do\_authenticated1 and session\_x11\_req functions, specifically related to crafted X11 forwarding data.  
  
The medium severity indicates that while this vulnerability can lead to bypassing shell-command restrictions, it may not directly result in complete system compromise or unauthorized access to sensitive data. However, it still presents a significant risk to the security of the SSH service and the potential for unauthorized actions.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by authenticated users, it could lead to the bypassing of intended shell-command restrictions, potentially allowing them to execute unauthorized commands or actions within the SSH session. This could result in unauthorized access to resources, manipulation of data, or disruption of services within the organization.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH CRLF Injection vulnerability in session.c, authenticated users can bypass intended shell-command restrictions by injecting CRLF sequences in crafted X11 forwarding data. This could allow them to execute unauthorized commands or actions within the SSH session, potentially leading to unauthorized access or manipulation of resources. While this vulnerability may not directly lead to complete system compromise, it can impact the security and integrity of the SSH service.  
  
In conclusion, the medium severity of the OpenSSH CRLF Injection vulnerability in session.c highlights the importance of enforcing proper input validation and preventing unauthorized actions within SSH sessions. While it may not directly result in unauthorized access to sensitive data, it can still lead to unauthorized actions and require attention to prevent potential exploitation by authenticated users.

#### 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 related to CRLF injection vulnerabilities in sshd in OpenSSH before version 7.2p2, it is recommended to update OpenSSH to a version beyond 7.2p2. This update should address the issue where remote authenticated users can bypass intended shell-command restrictions via crafted X11 forwarding data, specifically in the do\_authenticated1 and session\_x11\_req functions in session.c. Additionally, it is important to sanitize and validate X11 forwarding data to prevent CRLF injection attacks and unauthorized shell-command execution. By updating to the latest version of OpenSSH and implementing secure data handling practices, the risk of exploitation through CRLF injection vulnerabilities can be mitigated. Regularly updating OpenSSH and following secure configuration guidelines are essential to prevent unauthorized access and maintain a secure SSH environment.

### Finding 15

* Affected IP: **127.0.0.1**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2016-20012**
* Severity score: **0.0** (**MEDIUM** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: OpenSSH User Enumeration (Medium)  
  
#### Severity Rationale:  
The severity of the OpenSSH User Enumeration vulnerability is classified as medium due to the potential risk it poses to the confidentiality and security of user authentication in OpenSSH. This vulnerability allows remote attackers to test whether a certain combination of username and public key is known to an SSH server, potentially confirming the validity of the credentials.  
  
The medium severity indicates that while this vulnerability can lead to user enumeration and potentially disclose valid user accounts, it may not directly result in unauthorized access or compromise of sensitive data. However, it still presents a significant risk to the confidentiality and privacy of user information.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by attackers, it could lead to the enumeration of valid user accounts within the OpenSSH environment, potentially providing valuable information for targeted attacks or unauthorized access attempts. The disclosure of valid user accounts could aid attackers in further reconnaissance and exploitation of the organization's systems.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH User Enumeration vulnerability, attackers can test whether a certain combination of username and public key is known to an SSH server, potentially confirming the validity of the credentials. This could allow them to identify valid user accounts, potentially aiding in targeted attacks or unauthorized access attempts. While this vulnerability may not directly lead to unauthorized access, it can provide valuable information for attackers.  
  
In conclusion, the medium severity of the OpenSSH User Enumeration vulnerability underscores the importance of protecting user authentication information and preventing unauthorized disclosure of valid user accounts. While it may not directly result in unauthorized access, it can still lead to user enumeration and aid attackers in further exploitation, requiring attention to prevent potential exploitation by malicious actors.

#### 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 related to user enumeration in OpenSSH through version 8.7, it is recommended to update OpenSSH to a version beyond 8.7. This update should address the issue where remote attackers can test whether a certain combination of username and public key is known to an SSH server, potentially confirming the validity of login credentials. While the vendor does not recognize user enumeration as a vulnerability, it is still important to implement security best practices to protect against unauthorized access attempts. By updating to the latest version of OpenSSH and following secure authentication practices, the risk of user enumeration can be minimized. Regularly monitoring for security updates and patches for OpenSSH is essential to address known vulnerabilities and maintain a secure SSH environment.

### Finding 16

* Affected IP: **127.0.0.1**
* 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: OpenSSH X11 Forwarding Privilege Escalation (Critical)  
  
#### Severity Rationale:  
The severity of the OpenSSH X11 Forwarding Privilege Escalation vulnerability is classified as critical due to the significant risk it poses to the security and integrity of X11 forwarding in OpenSSH. This vulnerability allows remote X11 clients to trigger a fallback and obtain trusted X11 forwarding privileges by leveraging configuration issues on the local X11 server, as demonstrated by lack of the SECURITY extension on the X11 server.  
  
The critical severity indicates that this vulnerability can be easily exploited by remote attackers to escalate their privileges and gain unauthorized access to trusted X11 forwarding privileges. The impact of this vulnerability on the security of the system is severe, potentially leading to unauthorized access and compromise of sensitive data.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by attackers, it could lead to the unauthorized escalation of X11 forwarding privileges, allowing them to access sensitive X11 resources and potentially compromise the confidentiality and integrity of the system. The lack of proper access control decisions on the local X11 server could result in unauthorized access and manipulation of X11 forwarding privileges.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH X11 Forwarding Privilege Escalation vulnerability, remote X11 clients can trigger a fallback and obtain trusted X11 forwarding privileges by leveraging configuration issues on the local X11 server. This could allow attackers to escalate their privileges, access sensitive X11 resources, and potentially compromise the security of the system. Lack of the SECURITY extension on the X11 server could facilitate unauthorized access and manipulation of X11 forwarding privileges.  
  
In conclusion, the critical severity of the OpenSSH X11 Forwarding Privilege Escalation vulnerability highlights the urgent need for immediate remediation to prevent potential exploitation by remote attackers and protect the integrity and security of X11 forwarding in OpenSSH. It is crucial to address configuration issues on the local X11 server to mitigate the risk of unauthorized privilege escalation and ensure the secure operation of X11 forwarding.

#### 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 related to mishandling of X11 forwarding in OpenSSH before version 7.2, it is recommended to update OpenSSH to a version beyond 7.2. This update should address the issue where the client mishandles failed cookie generation for untrusted X11 forwarding, potentially allowing remote X11 clients to trigger a fallback and obtain trusted X11 forwarding privileges by exploiting configuration issues on the local X11 server. Additionally, it is important to configure the X11 server properly, including enabling the SECURITY extension, to prevent unauthorized access and privilege escalation through X11 forwarding. By updating to the latest version of OpenSSH and securing the X11 server configuration, the risk of exploitation through X11 forwarding vulnerabilities can be mitigated. Regularly monitoring for security updates and patches for OpenSSH is essential to address known vulnerabilities and maintain a secure SSH environment.

### Finding 17

* Affected IP: **127.0.0.1**
* 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: OpenSSH NEWKEYS Message Denial of Service (High)  
  
#### Severity Rationale:  
The severity of the OpenSSH NEWKEYS Message Denial of Service vulnerability is classified as high due to the significant risk it poses to the availability and stability of the SSH service in OpenSSH. This vulnerability in sshd before version 7.4 allows remote attackers to cause a denial of service by sending an out-of-sequence NEWKEYS message, resulting in a NULL pointer dereference and daemon crash, as demonstrated by Honggfuzz, related to kex.c and packet.c.  
  
The high severity indicates that this vulnerability can be easily exploited by remote attackers to disrupt the availability of the SSH service, potentially leading to service unavailability and system instability. The impact of this vulnerability on the availability of the system is critical.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by attackers, it could lead to a denial of service condition on the SSH service, rendering it unavailable for legitimate users. The NULL pointer dereference and daemon crash caused by the out-of-sequence NEWKEYS message could result in service disruption, system unresponsiveness, and potential downtime for critical SSH services.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH NEWKEYS Message Denial of Service vulnerability, attackers can send an out-of-sequence NEWKEYS message to trigger a NULL pointer dereference and daemon crash, leading to a denial of service condition. This could render the SSH service unavailable, impacting legitimate users' ability to access the system and disrupting critical operations. The attacker could leverage this vulnerability to disrupt services and potentially cause system instability.  
  
In conclusion, the high severity of the OpenSSH NEWKEYS Message Denial of Service vulnerability highlights the urgent need for immediate remediation to prevent potential exploitation by remote attackers and protect the availability and stability of the SSH service. It is crucial to address this vulnerability to ensure the reliability and availability of SSH services and prevent service disruptions caused by denial of service attacks.

#### 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 related to a denial of service in OpenSSH before version 7.4, it is recommended to update OpenSSH to a version beyond 7.4. This update should address the issue where remote attackers can cause a denial of service by triggering a NULL pointer dereference and crashing the daemon through an out-of-sequence NEWKEYS message. By updating to the latest version of OpenSSH, the vulnerability related to kex.c and packet.c can be patched, reducing the risk of a denial of service attack. Regularly monitoring for security updates and patches for OpenSSH is essential to address known vulnerabilities and maintain a secure SSH environment.

### Finding 18

* Affected IP: **127.0.0.1**
* 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: OpenSSH Shared Memory Manager Privilege Escalation (High)  
  
#### Severity Rationale:  
The severity of the OpenSSH Shared Memory Manager Privilege Escalation vulnerability is classified as high due to the significant risk it poses to the security and privilege separation mechanisms in OpenSSH. This vulnerability in sshd before version 7.4 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.  
  
The high severity indicates that this vulnerability can be exploited by local users to escalate their privileges and potentially gain unauthorized access to sensitive resources or compromise the integrity of the system. The impact of this vulnerability on the security of the system is critical.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by local users, it could lead to the escalation of privileges within the OpenSSH environment, potentially allowing them to gain unauthorized access to sensitive resources or compromise the security of the system. Leveraging access to a sandboxed privilege-separation process could provide attackers with the means to bypass security controls and gain elevated privileges.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH Shared Memory Manager Privilege Escalation vulnerability, local users can gain privileges by leveraging access to a sandboxed privilege-separation process and manipulating the m\_zback and m\_zlib data structures. This could allow attackers to escalate their privileges, potentially gaining unauthorized access to sensitive resources, compromising the integrity of the system, or bypassing security controls. The vulnerability could lead to unauthorized privilege escalation and compromise of the system.  
  
In conclusion, the high severity of the OpenSSH Shared Memory Manager Privilege Escalation vulnerability underscores the urgent need for immediate remediation to prevent potential exploitation by local users and protect the security and integrity of the privilege separation mechanisms in OpenSSH. It is crucial to address this vulnerability to ensure the secure operation of OpenSSH and prevent unauthorized privilege escalation within the system.

#### 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 related to the shared memory manager in sshd in OpenSSH before version 7.4, it is recommended to update OpenSSH to a version beyond 7.4. This update should address the issue where local users could potentially gain privileges by exploiting a lack of bounds check enforcement in the shared memory manager associated with pre-authentication compression. By updating to the latest version of OpenSSH, the vulnerability related to the m\_zback and m\_zlib data structures can be fixed, reducing the risk of privilege escalation attacks. Regularly monitoring for security updates and patches for OpenSSH is essential to address known vulnerabilities and maintain a secure SSH environment.

### Finding 19

* Affected IP: **127.0.0.1**
* 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: OpenSSH Private-Key Information Disclosure (Medium)  
  
#### Severity Rationale:  
The severity of the OpenSSH Private-Key Information Disclosure vulnerability is classified as medium due to the potential risk it poses to the confidentiality and security of private-key information in OpenSSH. This vulnerability in authfile.c in sshd before version 7.4 allows local users to obtain sensitive private-key information by leveraging access to a privilege-separated child process, as the effects of realloc on buffer contents are not properly considered.  
  
The medium severity indicates that while this vulnerability can lead to the disclosure of sensitive private-key information, it may not directly result in complete system compromise or unauthorized access to critical resources. However, it still presents a significant risk to the confidentiality and integrity of private-key data.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by local users, it could lead to the disclosure of sensitive private-key information, potentially compromising the confidentiality and security of the organization's cryptographic keys. Unauthorized access to private-key data could result in unauthorized access to encrypted data, manipulation of cryptographic operations, or compromise of secure communications.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH Private-Key Information Disclosure vulnerability, local users can obtain sensitive private-key information by leveraging access to a privilege-separated child process and manipulating buffer contents affected by realloc. This could allow attackers to access private keys, compromising the confidentiality and security of cryptographic operations and potentially leading to unauthorized access to encrypted data or secure communications. While this vulnerability may not directly lead to complete system compromise, it can impact the security of private-key information.  
  
In conclusion, the medium severity of the OpenSSH Private-Key Information Disclosure vulnerability highlights the importance of protecting sensitive private-key information and preventing unauthorized access to cryptographic keys. It is crucial to address this vulnerability to ensure the confidentiality and integrity of private-key data and prevent potential exploitation by local users to access sensitive cryptographic information.

#### 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 related to sensitive private-key information exposure in OpenSSH before version 7.4, it is recommended to update OpenSSH to a version beyond 7.4. This update should address the issue where authfile.c in sshd does not properly consider the effects of realloc on buffer contents, potentially allowing local users to obtain sensitive private-key information by exploiting access to a privilege-separated child process. By updating to the latest version of OpenSSH, the vulnerability can be fixed, reducing the risk of private-key exposure. Regularly monitoring for security updates and patches for OpenSSH is essential to address known vulnerabilities and maintain a secure SSH environment.

### Finding 20

* Affected IP: **127.0.0.1**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2016-10010**
* Severity score: **0.0** (**HIGH** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: OpenSSH Privilege Escalation via Unix-Domain Sockets (High)  
  
#### Severity Rationale:  
The severity of the OpenSSH Privilege Escalation via Unix-Domain Sockets vulnerability is classified as high due to the significant risk it poses to the security and privilege separation mechanisms in OpenSSH. This vulnerability in sshd before version 7.4, when privilege separation is not used, creates forwarded Unix-domain sockets as root, potentially allowing local users to gain privileges via unspecified vectors, related to serverloop.c.  
  
The high severity indicates that this vulnerability can be exploited by local users to escalate their privileges and potentially gain unauthorized access to sensitive resources or compromise the integrity of the system. The impact of this vulnerability on the security of the system is critical.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by local users, it could lead to the escalation of privileges within the OpenSSH environment, potentially allowing them to gain unauthorized access to sensitive resources or compromise the security of the system. Creating forwarded Unix-domain sockets as root could provide attackers with the means to bypass security controls and gain elevated privileges.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH Privilege Escalation via Unix-Domain Sockets vulnerability, local users can gain privileges by leveraging the creation of forwarded Unix-domain sockets as root when privilege separation is not used. This could allow attackers to escalate their privileges, potentially gaining unauthorized access to sensitive resources, compromising the integrity of the system, or bypassing security controls. The vulnerability could lead to unauthorized privilege escalation and compromise of the system.  
  
In conclusion, the high severity of the OpenSSH Privilege Escalation via Unix-Domain Sockets vulnerability underscores the urgent need for immediate remediation to prevent potential exploitation by local users and protect the security and integrity of the privilege separation mechanisms in OpenSSH. It is crucial to address this vulnerability to ensure the secure operation of OpenSSH and prevent unauthorized privilege escalation within the system.

#### 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 related to privilege escalation in OpenSSH before version 7.4 when privilege separation is not used, it is recommended to update OpenSSH to a version beyond 7.4. This update should address the issue where sshd creates forwarded Unix-domain sockets as root, potentially allowing local users to gain privileges via unspecified vectors related to serverloop.c. By updating to the latest version of OpenSSH, the vulnerability can be fixed, reducing the risk of privilege escalation attacks. It is also important to enable privilege separation in OpenSSH to enhance security and restrict the capabilities of potential attackers. Regularly monitoring for security updates and patches for OpenSSH is essential to address known vulnerabilities and maintain a secure SSH environment.

### Finding 21

* Affected IP: **127.0.0.1**
* 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: OpenSSH Untrusted Search Path for PKCS#11 Modules (High)  
  
#### Severity Rationale:  
The severity of the OpenSSH Untrusted Search Path for PKCS#11 Modules vulnerability is classified as high due to the significant risk it poses to the security and integrity of the PKCS#11 modules in OpenSSH. This vulnerability in ssh-agent.c in ssh-agent before version 7.4 allows remote attackers to execute arbitrary local PKCS#11 modules by leveraging control over a forwarded agent-socket.  
  
The high severity indicates that this vulnerability can be exploited by remote attackers to execute arbitrary local PKCS#11 modules, potentially leading to unauthorized access, compromise of cryptographic operations, or manipulation of security mechanisms. The impact of this vulnerability on the security of the system is critical.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by remote attackers, it could lead to the execution of arbitrary local PKCS#11 modules, potentially compromising the security and integrity of cryptographic operations within the organization. Unauthorized access to PKCS#11 modules could result in unauthorized cryptographic operations, manipulation of security mechanisms, or compromise of sensitive data.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH Untrusted Search Path for PKCS#11 Modules vulnerability, remote attackers can execute arbitrary local PKCS#11 modules by leveraging control over a forwarded agent-socket. This could allow attackers to compromise the security of cryptographic operations, manipulate security mechanisms, or gain unauthorized access to sensitive data. The vulnerability could lead to unauthorized execution of PKCS#11 modules and compromise the integrity of cryptographic operations.  
  
In conclusion, the high severity of the OpenSSH Untrusted Search Path for PKCS#11 Modules vulnerability highlights the urgent need for immediate remediation to prevent potential exploitation by remote attackers and protect the security and integrity of PKCS#11 modules in OpenSSH. It is crucial to address this vulnerability to ensure the secure operation of cryptographic operations and prevent unauthorized access to sensitive cryptographic mechanisms.

#### 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 related to arbitrary local PKCS#11 module execution in OpenSSH before version 7.4, it is recommended to update OpenSSH to a version beyond 7.4. This update should address the untrusted search path vulnerability in ssh-agent.c, which allows remote attackers to execute arbitrary local PKCS#11 modules by exploiting control over a forwarded agent-socket. By updating to the latest version of OpenSSH, the vulnerability can be fixed, reducing the risk of arbitrary library loading attacks. Additionally, it is important to restrict access to forwarded agent-sockets and implement proper access controls to prevent unauthorized execution of PKCS#11 modules. Regularly monitoring for security updates and patches for OpenSSH is essential to address known vulnerabilities and maintain a secure SSH environment.

### Finding 22

* Affected IP: **127.0.0.1**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2015-8325**
* Severity score: **0.0** (**HIGH** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: OpenSSH Privilege Escalation via LD\_PRELOAD (High)  
  
#### Severity Rationale:  
The severity of the OpenSSH Privilege Escalation via LD\_PRELOAD vulnerability is classified as high due to the significant risk it poses to the security and privilege separation mechanisms in OpenSSH. This vulnerability in sshd through version 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.  
  
The high severity indicates that this vulnerability can be exploited by local users to escalate their privileges and potentially gain unauthorized access to sensitive resources or compromise the integrity of the system. The impact of this vulnerability on the security of the system is critical.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by local users, it could lead to the escalation of privileges within the OpenSSH environment, potentially allowing them to gain unauthorized access to sensitive resources or compromise the security of the system. Triggering a crafted environment for the /bin/login program via the LD\_PRELOAD environment variable could provide attackers with the means to bypass security controls and gain elevated privileges.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH Privilege Escalation via LD\_PRELOAD vulnerability, local users can gain privileges by triggering a crafted environment for the /bin/login program, leveraging the LD\_PRELOAD environment variable. This could allow attackers to escalate their privileges, potentially gaining unauthorized access to sensitive resources, compromising the integrity of the system, or bypassing security controls. The vulnerability could lead to unauthorized privilege escalation and compromise of the system.  
  
In conclusion, the high severity of the OpenSSH Privilege Escalation via LD\_PRELOAD vulnerability underscores the urgent need for immediate remediation to prevent potential exploitation by local users and protect the security and integrity of the privilege separation mechanisms in OpenSSH. It is crucial to address this vulnerability to ensure the secure operation of OpenSSH and prevent unauthorized privilege escalation within the system.

#### 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 related to privilege escalation in OpenSSH through version 7.2p2 when the UseLogin feature is enabled and PAM is configured to read .pam\_environment files in user home directories, it is recommended to update OpenSSH to a version beyond 7.2p2. This update should address the issue where the do\_setup\_env function in session.c allows local users to gain privileges by manipulating the environment for the /bin/login program, such as by setting a crafted LD\_PRELOAD environment variable. By updating to the latest version of OpenSSH, the vulnerability can be fixed, reducing the risk of privilege escalation attacks. Additionally, it is important to review and restrict PAM configurations and environment settings to prevent unauthorized privilege escalation. Regularly monitoring for security updates and patches for OpenSSH is essential to address known vulnerabilities and maintain a secure SSH environment.

### Finding 23

* Affected IP: **127.0.0.1**
* 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: OpenSSH Use-After-Free Privilege Escalation (Medium)  
  
#### Severity Rationale:  
The severity of the OpenSSH Use-After-Free Privilege Escalation vulnerability is classified as medium due to the potential risk it poses to the security and privilege separation mechanisms in OpenSSH. This vulnerability in the mm\_answer\_pam\_free\_ctx function in monitor.c in sshd before version 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.  
  
The medium severity indicates that while this vulnerability can lead to privilege escalation and potentially gain unauthorized access to sensitive resources, it may not directly result in complete system compromise or widespread impact. However, it still presents a significant risk to the confidentiality and integrity of the system.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by local users, it could lead to the escalation of privileges within the OpenSSH environment, potentially allowing them to gain unauthorized access to sensitive resources or compromise the security of the system. Leveraging control of the sshd uid to send an unexpectedly early MONITOR\_REQ\_PAM\_FREE\_CTX request could provide attackers with the means to bypass security controls and gain elevated privileges.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH Use-After-Free Privilege Escalation vulnerability, local users can gain privileges by leveraging control of the sshd uid to send an unexpectedly early MONITOR\_REQ\_PAM\_FREE\_CTX request, triggering a use-after-free condition. This could allow attackers to escalate their privileges, potentially gaining unauthorized access to sensitive resources, compromising the integrity of the system, or bypassing security controls. The vulnerability could lead to unauthorized privilege escalation and compromise of the system.  
  
In conclusion, the medium severity of the OpenSSH Use-After-Free Privilege Escalation vulnerability highlights the importance of protecting the privilege separation mechanisms in OpenSSH and preventing unauthorized privilege escalation by local users. It is crucial to address this vulnerability to ensure the secure operation of OpenSSH and prevent unauthorized access to sensitive resources within 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 related to a use-after-free vulnerability in OpenSSH before version 7.0 on non-OpenBSD platforms, it is recommended to update OpenSSH to a version beyond 7.0. This update should address the issue where local users could potentially gain privileges by exploiting a use-after-free vulnerability in the mm\_answer\_pam\_free\_ctx function in monitor.c. By updating to the latest version of OpenSSH, the vulnerability can be fixed, reducing the risk of privilege escalation attacks. Additionally, it is important to restrict access to the sshd uid and monitor and control the requests sent to the sshd process to prevent unauthorized actions. Regularly monitoring for security updates and patches for OpenSSH is essential to address known vulnerabilities and maintain a secure SSH environment.

### Finding 24

* Affected IP: **127.0.0.1**
* 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: OpenSSH Impersonation Attack via MONITOR\_REQ\_PAM\_INIT\_CTX (Low)  
  
#### Severity Rationale:  
The severity of the OpenSSH Impersonation Attack via MONITOR\_REQ\_PAM\_INIT\_CTX vulnerability is classified as low due to the limited impact it has on the security and integrity of the SSH service in OpenSSH. This vulnerability in the monitor component in sshd before version 7.0 on non-OpenBSD platforms allows local users to conduct impersonation attacks by sending extraneous username data in MONITOR\_REQ\_PAM\_INIT\_CTX requests and leveraging any SSH login access in conjunction with control of the sshd uid to send a crafted MONITOR\_REQ\_PWNAM request.  
  
The low severity indicates that while this vulnerability can lead to impersonation attacks and potential unauthorized access, it may not directly result in widespread compromise or significant impact on the system. However, it still presents a risk to the confidentiality and integrity of the system.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by local users, it could lead to impersonation attacks and potential unauthorized access by leveraging extraneous username data in MONITOR\_REQ\_PAM\_INIT\_CTX requests. Conducting impersonation attacks by sending crafted MONITOR\_REQ\_PWNAM requests could allow attackers to gain unauthorized access to resources or impersonate other users within the system.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH Impersonation Attack via MONITOR\_REQ\_PAM\_INIT\_CTX vulnerability, local users can conduct impersonation attacks by sending extraneous username data in MONITOR\_REQ\_PAM\_INIT\_CTX requests and leveraging control of the sshd uid to send crafted MONITOR\_REQ\_PWNAM requests. This could allow attackers to impersonate other users, potentially gaining unauthorized access to resources or conducting unauthorized actions within the system. While the impact may be limited, the vulnerability could still lead to unauthorized access and compromise.  
  
In conclusion, the low severity of the OpenSSH Impersonation Attack via MONITOR\_REQ\_PAM\_INIT\_CTX vulnerability indicates a lower risk level compared to other critical vulnerabilities. It is important to address this vulnerability to prevent impersonation attacks and unauthorized access within the system, even though the impact may be limited.

#### 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 related to impersonation attacks in OpenSSH before version 7.0 on non-OpenBSD platforms, it is recommended to update OpenSSH to a version beyond 7.0. This update should address the issue where the monitor component in sshd accepts extraneous username data in MONITOR\_REQ\_PAM\_INIT\_CTX requests, potentially allowing local users to conduct impersonation attacks. By updating to the latest version of OpenSSH, the vulnerability can be fixed, reducing the risk of impersonation attacks. Additionally, it is important to monitor and control the requests sent to the sshd process to prevent unauthorized actions and impersonation attempts. Regularly monitoring for security updates and patches for OpenSSH is essential to address known vulnerabilities and maintain a secure SSH environment.

### Finding 25

* Affected IP: **127.0.0.1**
* 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: OpenSSH Keyboard-Interactive Device Processing (High)  
  
#### Severity Rationale:  
The severity of the OpenSSH Keyboard-Interactive Device Processing vulnerability is classified as high due to the significant risk it poses to the security and availability of the SSH service in OpenSSH. This vulnerability in the kbdint\_next\_device function in auth2-chall.c in sshd through version 6.9 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) by providing a long and duplicative list in the ssh -oKbdInteractiveDevices option.  
  
The high severity indicates that this vulnerability can be exploited by remote attackers to conduct brute-force attacks, potentially leading to unauthorized access or denial of service due to excessive CPU consumption. The impact of this vulnerability on the security and availability of the system is critical.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by remote attackers, it could lead to the compromise of user passwords through brute-force attacks or cause a denial of service by consuming excessive CPU resources. Unauthorized access to user accounts or denial of service could disrupt operations, compromise sensitive data, and impact the availability of the SSH service within the organization.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH Keyboard-Interactive Device Processing vulnerability, remote attackers can conduct brute-force attacks or cause a denial of service by providing a long and duplicative list in the ssh -oKbdInteractiveDevices option. This could lead to unauthorized access to user accounts through password guessing or denial of service due to excessive CPU consumption. The vulnerability could result in compromised user passwords or service disruption.  
  
In conclusion, the high severity of the OpenSSH Keyboard-Interactive Device Processing vulnerability highlights the urgent need for immediate remediation to prevent potential exploitation by remote attackers and protect the security and availability of the SSH service. It is crucial to address this vulnerability to mitigate the risk of unauthorized access and denial of service attacks within the organization.

#### 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 related to keyboard-interactive device processing in OpenSSH through version 6.9, it is recommended to update OpenSSH to a version beyond 6.9. This update should address the issue where the kbdint\_next\_device function in auth2-chall.c does not properly restrict the processing of keyboard-interactive devices within a single connection, potentially enabling remote attackers to conduct brute-force attacks or cause denial of service through excessive CPU consumption. By updating to the latest version of OpenSSH, the vulnerability can be fixed, reducing the risk of brute-force attacks and resource exhaustion. Additionally, it is important to configure and limit the use of keyboard-interactive devices to prevent abuse and unauthorized access attempts. Regularly monitoring for security updates and patches for OpenSSH is essential to address known vulnerabilities and maintain a secure SSH environment.

### Finding 26

* Affected IP: **127.0.0.1**
* 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: OpenSSH X11 Connection Time Window Bypass (Medium)  
  
#### Severity Rationale:  
The severity of the OpenSSH X11 Connection Time Window Bypass vulnerability is classified as medium due to the potential risk it poses to the integrity and security of X11 forwarding in OpenSSH. This vulnerability in the x11\_open\_helper function in channels.c in ssh before version 6.9, when ForwardX11Trusted mode is not used, lacks a check of the refusal deadline for X connections, making it easier for remote attackers to bypass intended access restrictions by establishing a connection outside of the permitted time window.  
  
The medium severity indicates that while this vulnerability can lead to bypassing access restrictions and potential unauthorized access, it may not directly result in complete system compromise or widespread impact. However, it still presents a significant risk to the confidentiality and integrity of the X11 forwarding mechanism.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by remote attackers, it could lead to the bypassing of intended access restrictions for X11 forwarding, potentially allowing unauthorized access to X11 resources or compromising the security of the system. Establishing a connection outside of the permitted time window could result in unauthorized access to graphical applications or sensitive data.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH X11 Connection Time Window Bypass vulnerability, remote attackers can bypass intended access restrictions for X11 forwarding by establishing a connection outside of the permitted time window. This could allow attackers to gain unauthorized access to X11 resources, compromise the security of the system, or access graphical applications outside of the intended time frame. While the impact may be limited, the vulnerability could still lead to unauthorized access and compromise.  
  
In conclusion, the medium severity of the OpenSSH X11 Connection Time Window Bypass vulnerability underscores the importance of enforcing access restrictions and preventing unauthorized access to X11 resources within the system. It is crucial to address this vulnerability to ensure the secure operation of X11 forwarding and prevent potential exploitation by remote attackers to bypass access restrictions.

#### 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 related to X11 connection restrictions in OpenSSH before version 6.9, it is recommended to update OpenSSH to a version beyond 6.9. This update should address the issue where the x11\_open\_helper function in channels.c lacks a check of the refusal deadline for X connections, potentially allowing remote attackers to bypass intended access restrictions by connecting outside of the permitted time window. By updating to the latest version of OpenSSH, the vulnerability can be fixed, reducing the risk of unauthorized X11 connections. Additionally, it is important to configure and enforce access restrictions, including the use of ForwardX11Trusted mode, to prevent unauthorized access and abuse of X11 forwarding. Regularly monitoring for security updates and patches for OpenSSH is essential to address known vulnerabilities and maintain a secure SSH environment.

### Finding 27

* Affected IP: **127.0.0.1**
* 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: OpenSSH SSHFP DNS RR Checking Bypass (Medium)  
  
#### Severity Rationale:  
The severity of the OpenSSH SSHFP DNS RR Checking Bypass vulnerability is classified as medium due to the potential risk it poses to the integrity and security of SSH connections in OpenSSH. This vulnerability in 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.  
  
The medium severity indicates that while this vulnerability can lead to the bypassing of SSHFP DNS RR checking and potential security implications, it may not directly result in complete system compromise or unauthorized access. However, it still presents a significant risk to the authenticity and trustworthiness of SSH connections.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by remote servers, it could lead to the skipping of SSHFP DNS RR checking, potentially allowing for the acceptance of invalid or compromised host certificates. This could compromise the authenticity and trustworthiness of SSH connections, potentially leading to unauthorized access or manipulation of data during SSH sessions.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH SSHFP DNS RR Checking Bypass vulnerability, remote servers can trigger the skipping of SSHFP DNS RR checking by presenting an unacceptable HostCertificate. This could allow attackers to bypass a critical security check and potentially present invalid or compromised host certificates, compromising the authenticity and trustworthiness of SSH connections. While the impact may not be immediate, the vulnerability could lead to security implications during SSH sessions.  
  
In conclusion, the medium severity of the OpenSSH SSHFP DNS RR Checking Bypass vulnerability highlights the importance of enforcing proper security checks and ensuring the authenticity of SSH connections in OpenSSH. It is crucial to address this vulnerability to prevent the bypassing of critical security checks and maintain the integrity of SSH connections to prevent potential exploitation by remote servers.

#### 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 related to SSHFP DNS RR checking in OpenSSH 6.6 and earlier, it is recommended to update OpenSSH to a version beyond 6.6. This update should address the issue where the verify\_host\_key function in sshconnect.c allows remote servers to bypass SSHFP DNS RR checking by presenting an unacceptable HostCertificate. By updating to the latest version of OpenSSH, the vulnerability can be fixed, reducing the risk of skipping SSHFP DNS RR checking. Additionally, it is important to configure SSH clients to enforce strict host key verification and not skip DNS RR checking to prevent potential security bypasses. Regularly monitoring for security updates and patches for OpenSSH is essential to address known vulnerabilities and maintain a secure SSH environment.

### Finding 28

* Affected IP: **127.0.0.1**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2014-2532**
* Severity score: **0.0** (**MEDIUM** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: OpenSSH AcceptEnv Wildcard Bypass (Medium)  
  
#### Severity Rationale:  
The severity of the OpenSSH AcceptEnv Wildcard Bypass vulnerability is classified as medium due to the potential risk it poses to the security and integrity of environment restrictions in OpenSSH. This vulnerability in sshd before version 6.6 does not properly support wildcards on AcceptEnv lines in sshd\_config, allowing remote attackers to bypass intended environment restrictions by using a substring located before a wildcard character.  
  
The medium severity indicates that while this vulnerability can lead to the bypassing of environment restrictions and potential security implications, it may not directly result in complete system compromise or unauthorized access. However, it still presents a significant risk to the confidentiality and integrity of the environment configuration.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by remote attackers, it could lead to the bypassing of intended environment restrictions in OpenSSH, potentially allowing unauthorized access or manipulation of environment variables. Bypassing environment restrictions could lead to security implications, unauthorized access, or manipulation of system configurations.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH AcceptEnv Wildcard Bypass vulnerability, remote attackers can bypass intended environment restrictions by using a substring located before a wildcard character in AcceptEnv lines in sshd\_config. This could allow attackers to manipulate environment variables, potentially bypassing security controls or gaining unauthorized access to resources. While the impact may not be immediate, the vulnerability could lead to security implications within the system.  
  
In conclusion, the medium severity of the OpenSSH AcceptEnv Wildcard Bypass vulnerability underscores the importance of enforcing proper environment restrictions and configurations in OpenSSH. It is crucial to address this vulnerability to prevent the bypassing of environment restrictions and maintain the integrity of system configurations to prevent potential exploitation by remote attackers.

#### 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 related to wildcard support in AcceptEnv lines in sshd\_config in OpenSSH before version 6.6, it is recommended to update OpenSSH to a version beyond 6.6. This update should address the issue where sshd does not properly support wildcards on AcceptEnv lines, potentially allowing remote attackers to bypass intended environment restrictions by manipulating substrings located before wildcard characters. By updating to the latest version of OpenSSH, the vulnerability can be fixed, reducing the risk of environment restriction bypasses. Additionally, it is important to review and configure AcceptEnv lines carefully to avoid unintended exposure of sensitive information. Regularly monitoring for security updates and patches for OpenSSH is essential to address known vulnerabilities and maintain a secure SSH environment.

### Finding 29

* Affected IP: **127.0.0.1**
* 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: OpenSSH J-PAKE Protocol Memory Corruption (High)  
  
#### Severity Rationale:  
The severity of the OpenSSH J-PAKE Protocol Memory Corruption vulnerability is classified as high due to the significant risk it poses to the security and stability of the SSH service in OpenSSH. This vulnerability in the hash\_buffer function in schnorr.c in OpenSSH through version 6.4, when the J-PAKE protocol is enabled, does not initialize certain data structures, potentially allowing remote attackers to cause a denial of service (memory corruption) or have unspecified other impacts via vectors that trigger an error condition.  
  
The high severity indicates that this vulnerability can be exploited by remote attackers to cause memory corruption, potentially leading to denial of service or other unspecified impacts that could compromise the security and stability of the SSH service. The impact of this vulnerability on the availability and integrity of the system is critical.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by remote attackers, it could lead to memory corruption, denial of service, or other unspecified impacts that could disrupt the availability and stability of the SSH service in OpenSSH. Memory corruption could lead to system instability, service unavailability, or potential security vulnerabilities within the organization.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH J-PAKE Protocol Memory Corruption vulnerability, remote attackers can trigger memory corruption in the hash\_buffer function in OpenSSH, potentially leading to denial of service or other unspecified impacts. This could disrupt the availability and stability of the SSH service, compromise the security of the system, or lead to unexpected behavior that could impact the organization's operations. The vulnerability could result in memory corruption and potential system instability.  
  
In conclusion, the high severity of the OpenSSH J-PAKE Protocol Memory Corruption vulnerability highlights the urgent need for immediate remediation to prevent potential exploitation by remote attackers and protect the security and stability of the SSH service. It is crucial to address this vulnerability to mitigate the risk of memory corruption, denial of service, or other unspecified impacts that could compromise the integrity of the 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 related to memory corruption in OpenSSH through version 6.4 when the J-PAKE protocol is enabled, it is recommended to update OpenSSH to a version beyond 6.4. This update should address the issue where the hash\_buffer function in schnorr.c does not properly initialize certain data structures, potentially allowing remote attackers to cause a denial of service (memory corruption) or other unspecified impacts by triggering error conditions. By updating to the latest version of OpenSSH, the vulnerability can be fixed, reducing the risk of memory corruption and potential denial of service attacks. Additionally, it is important to review and configure the J-PAKE protocol settings carefully to prevent exploitation of vulnerabilities. Regularly monitoring for security updates and patches for OpenSSH is essential to address known vulnerabilities and maintain a secure SSH environment.

### Finding 30

* Affected IP: **127.0.0.1**
* 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: OpenSSH Debug Message Information Disclosure (Low)  
  
#### Severity Rationale:  
The severity of the OpenSSH Debug Message Information Disclosure vulnerability is classified as low due to the limited impact it has on the security and confidentiality of the SSH service in OpenSSH. This vulnerability in the auth\_parse\_options function in auth-options.c in sshd before version 5.7 provides debug messages containing authorized\_keys command options, allowing remote authenticated users to obtain potentially sensitive information by reading these messages.  
  
The low severity indicates that while this vulnerability can lead to the disclosure of potentially sensitive information, it may not directly result in unauthorized access or compromise of critical data. However, it still presents a risk to the confidentiality and privacy of authorized\_keys command options.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by remote authenticated users, it could lead to the disclosure of potentially sensitive information contained in authorized\_keys command options. While the impact may be limited, unauthorized access to this information could potentially aid attackers in further reconnaissance or exploitation of the SSH service.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH Debug Message Information Disclosure vulnerability, remote authenticated users can obtain potentially sensitive information by reading debug messages containing authorized\_keys command options. This could allow attackers to gather information that may aid in understanding the configuration of the SSH service, potentially leading to further exploitation or unauthorized access. While the impact may be low, the vulnerability could still lead to information disclosure.  
  
In conclusion, the low severity of the OpenSSH Debug Message Information Disclosure vulnerability indicates a lower risk level compared to critical vulnerabilities. It is important to address this vulnerability to prevent potential information disclosure and protect the confidentiality of authorized\_keys command options within the SSH service.

#### 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 related to debug messages containing authorized\_keys command options in OpenSSH before version 5.7, it is recommended to update OpenSSH to a version beyond 5.7. This update should address the issue where the auth\_parse\_options function in auth-options.c provides debug messages that may expose potentially sensitive information from authorized\_keys command options. By updating to the latest version of OpenSSH, the vulnerability can be fixed, reducing the risk of unauthorized access to sensitive information. Additionally, it is important to review and restrict debug message settings to prevent the exposure of sensitive data. Regularly monitoring for security updates and patches for OpenSSH is essential to address known vulnerabilities and maintain a secure SSH environment.

### Finding 31

* Affected IP: **127.0.0.1**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2011-5000**
* Severity score: **0.0** (**LOW** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: OpenSSH GSSAPI Memory Consumption (Low)  
  
#### Severity Rationale:  
The severity of the OpenSSH GSSAPI Memory Consumption vulnerability is classified as low due to the limited impact it has on the security and availability of the SSH service in OpenSSH. This 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, allows remote authenticated users to cause a denial of service (memory consumption) by providing a large value in a certain length field.  
  
The low severity indicates that while this vulnerability can lead to memory consumption and potential denial of service, it may not directly result in unauthorized access or compromise of critical data. However, it still presents a risk to the availability and stability of the SSH service.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by remote authenticated users, it could lead to memory consumption and potential denial of service within the SSH service. While the impact may be limited, denial of service could disrupt SSH connections and impact the availability of the service for legitimate users.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH GSSAPI Memory Consumption vulnerability, remote authenticated users can cause memory consumption by providing a large value in a certain length field during gssapi-with-mic authentication. This could lead to denial of service, potentially disrupting SSH connections and impacting the availability of the service. While the impact may be low, the vulnerability could still lead to service disruption.  
  
In conclusion, the low severity of the OpenSSH GSSAPI Memory Consumption vulnerability indicates a lower risk level compared to critical vulnerabilities. It is important to address this vulnerability to prevent potential denial of service and protect the availability and stability of the SSH service within limited scenarios where this issue may be relevant.

#### 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 related to denial of service due to memory consumption in OpenSSH 5.8 and earlier when gssapi-with-mic authentication is enabled, it is recommended to update OpenSSH to a version beyond 5.8. This update should address the issue where the ssh\_gssapi\_parse\_ename function in gss-serv.c allows remote authenticated users to cause memory consumption by providing a large value in a specific length field. By updating to the latest version of OpenSSH, the vulnerability can be fixed, reducing the risk of denial of service attacks. Additionally, it is important to review and configure authentication settings to limit exposure to potential vulnerabilities. Regularly monitoring for security updates and patches for OpenSSH is essential to address known vulnerabilities and maintain a secure SSH environment.

### Finding 32

* Affected IP: **127.0.0.1**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2011-4327**
* Severity score: **0.0** (**LOW** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: OpenSSH ssh-rand-helper Information Disclosure (Low)  
  
#### Severity Rationale:  
The severity of the OpenSSH ssh-rand-helper Information Disclosure vulnerability is classified as low due to the limited impact it has on the security and confidentiality of sensitive key information in OpenSSH. This vulnerability in ssh-keysign.c in ssh-keysign in OpenSSH before version 5.8p2 on certain platforms executes ssh-rand-helper with unintended open file descriptors, allowing local users to obtain sensitive key information via the ptrace system call.  
  
The low severity indicates that while this vulnerability can lead to the disclosure of sensitive key information, it may not directly result in unauthorized access or compromise of critical data. However, it still presents a risk to the confidentiality and privacy of key information.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by local users, it could lead to the disclosure of sensitive key information via the ptrace system call. While the impact may be limited, unauthorized access to key information could potentially aid attackers in further exploitation or unauthorized access to cryptographic keys.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH ssh-rand-helper Information Disclosure vulnerability, local users can obtain sensitive key information by leveraging unintended open file descriptors in ssh-rand-helper execution. This could lead to the disclosure of key information via the ptrace system call, potentially compromising the confidentiality and security of cryptographic keys. While the impact may be low, the vulnerability could still lead to information disclosure.  
  
In conclusion, the low severity of the OpenSSH ssh-rand-helper Information Disclosure vulnerability indicates a lower risk level compared to critical vulnerabilities. It is important to address this vulnerability to prevent potential information disclosure and protect the confidentiality of sensitive key information within the OpenSSH environment.

#### 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 related to sensitive key information exposure in OpenSSH before version 5.8p2 on certain platforms, it is recommended to update OpenSSH to a version beyond 5.8p2. This update should address the issue where ssh-keysign.c in ssh-keysign executes ssh-rand-helper with unintended open file descriptors, potentially allowing local users to obtain sensitive key information via the ptrace system call. By updating to the latest version of OpenSSH, the vulnerability can be fixed, reducing the risk of sensitive key exposure. Additionally, it is important to restrict access to sensitive key information and review the execution environment of ssh-rand-helper to prevent unauthorized access. Regularly monitoring for security updates and patches for OpenSSH is essential to address known vulnerabilities and maintain a secure SSH environment.

### Finding 33

* Affected IP: **127.0.0.1**
* 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: OpenSSH Connection-Slot Exhaustion (Medium)  
  
#### Severity Rationale:  
The severity of the OpenSSH Connection-Slot Exhaustion vulnerability is classified as medium due to the potential risk it poses to the availability and performance of the SSH service in OpenSSH. This vulnerability in the default configuration of OpenSSH through version 6.1 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.  
  
The medium severity indicates that while this vulnerability can lead to denial of service and potential exhaustion of connection slots, it may not directly result in unauthorized access or compromise of critical data. However, it still presents a significant risk to the availability and performance of the SSH service.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by remote attackers, it could lead to denial of service by exhausting connection slots in the SSH service. This could disrupt legitimate users' ability to establish SSH connections, impact system availability, and degrade the performance of the SSH service within the organization.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH Connection-Slot Exhaustion vulnerability, remote attackers can cause denial of service by periodically making many new TCP connections and exhausting connection slots in the SSH service. This could lead to service disruption, prevent legitimate users from establishing SSH connections, and impact the availability and performance of the SSH service. While the impact may be limited to denial of service, the vulnerability could still lead to service degradation.  
  
In conclusion, the medium severity of the OpenSSH Connection-Slot Exhaustion vulnerability underscores the importance of mitigating denial of service risks and ensuring the availability and performance of the SSH service. It is crucial to address this vulnerability to prevent potential exploitation by remote attackers and protect the availability of SSH services within the organization.

#### 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 related to denial of service due to connection-slot exhaustion in OpenSSH through version 6.1, it is recommended to update OpenSSH to a version beyond 6.1. This update should address the issue where the default configuration enforces a fixed time limit between establishing a TCP connection and completing a login, potentially making it easier for remote attackers to cause denial of service by exhausting connection slots with periodic new TCP connections. By updating to the latest version of OpenSSH, the vulnerability can be mitigated, reducing the risk of connection-slot exhaustion attacks. Additionally, it is important to review and adjust the configuration settings to optimize resource utilization and prevent abuse of connection slots. Regularly monitoring for security updates and patches for OpenSSH is essential to address known vulnerabilities and maintain a secure SSH environment.

### Finding 34

* Affected IP: **127.0.0.1**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2010-4755**
* Severity score: **0.0** (**MEDIUM** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: OpenSSH SFTP Denial of Service (Medium)  
  
#### Severity Rationale:  
The severity of the OpenSSH SFTP Denial of Service vulnerability is classified as medium due to the significant risk it poses to the availability and performance of the SFTP service in OpenSSH. This vulnerability in the remote\_glob function in sftp-glob.c and the process\_put function in sftp.c in OpenSSH 5.8 and earlier allows remote authenticated users to cause a denial of service (CPU and memory consumption) by using crafted glob expressions that do not match any pathnames.  
  
The medium severity indicates that while this vulnerability can lead to denial of service and resource consumption, it may not directly result in unauthorized access or compromise of critical data. However, it still presents a significant risk to the availability and stability of the SFTP service.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by remote authenticated users, it could lead to denial of service by causing excessive CPU and memory consumption in the SFTP service. This could disrupt SFTP operations, impact system performance, and degrade the availability of file transfer services within the organization.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH SFTP Denial of Service vulnerability, remote authenticated users can cause denial of service by using crafted glob expressions that do not match any pathnames, leading to excessive CPU and memory consumption in the SFTP service. This could disrupt file transfer operations, consume system resources, and impact the availability and performance of the SFTP service. While the impact may be limited to denial of service, the vulnerability could still lead to service disruption.  
  
In conclusion, the medium severity of the OpenSSH SFTP Denial of Service vulnerability highlights the importance of mitigating denial of service risks and ensuring the availability and performance of the SFTP service. It is crucial to address this vulnerability to prevent potential exploitation by remote authenticated users and protect the availability of file transfer services within the organization.

#### 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 related to denial of service due to CPU and memory consumption in OpenSSH 5.8 and earlier, it is recommended to update OpenSSH to a version beyond 5.8. This update should address the issue where the remote\_glob function in sftp-glob.c and the process\_put function in sftp.c allow remote authenticated users to cause denial of service by using crafted glob expressions that do not match any pathnames. By updating to the latest version of OpenSSH, the vulnerability can be fixed, reducing the risk of CPU and memory consumption attacks. Additionally, it is important to review and restrict the use of glob expressions to prevent abuse and resource exhaustion. Regularly monitoring for security updates and patches for OpenSSH is essential to address known vulnerabilities and maintain a secure SSH environment.

### Finding 35

* Affected IP: **127.0.0.1**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2010-4478**
* Severity score: **0.0** (**HIGH** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: OpenSSH J-PAKE Protocol Validation Bypass (High)  
  
#### Severity Rationale:  
The severity of the OpenSSH J-PAKE Protocol Validation Bypass vulnerability is classified as high due to the significant risk it poses to the security and integrity of the authentication process in OpenSSH. This vulnerability in OpenSSH 5.6 and earlier, when J-PAKE is enabled, 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 J-PAKE protocol, bypassing the proper validation of public parameters.  
  
The high severity indicates that this vulnerability can lead to unauthorized authentication and potential compromise of the authentication process, allowing attackers to gain unauthorized access to the system. The impact of this vulnerability on the security of the system is critical.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by remote attackers, it could lead to unauthorized authentication and successful bypass of the authentication process in OpenSSH when J-PAKE is enabled. This could result in unauthorized access to sensitive resources, compromise of user accounts, and potential security breaches within the organization.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH J-PAKE Protocol Validation Bypass vulnerability, remote attackers can bypass the need for knowledge of the shared secret and successfully authenticate by sending crafted values in each round of the J-PAKE protocol. This allows attackers to bypass the proper validation of public parameters, leading to unauthorized authentication and potential compromise of the authentication process. The vulnerability could result in unauthorized access and security breaches.  
  
In conclusion, the high severity of the OpenSSH J-PAKE Protocol Validation Bypass vulnerability underscores the urgent need for immediate remediation to prevent potential exploitation by remote attackers and protect the authentication process in OpenSSH. It is crucial to address this vulnerability to ensure the secure authentication of users and prevent unauthorized access to the system.

#### 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 protocol validation in OpenSSH 5.6 and earlier, it is recommended to update OpenSSH to a version beyond 5.6. This update should address the issue where OpenSSH does not properly validate public parameters in the J-PAKE protocol, potentially allowing 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. By updating to the latest version of OpenSSH, the vulnerability can be fixed, reducing the risk of unauthorized authentication. Additionally, it is important to review and configure J-PAKE settings carefully to prevent exploitation of vulnerabilities. Regularly monitoring for security updates and patches for OpenSSH is essential to address known vulnerabilities and maintain a secure SSH environment.

### Finding 36

* Affected IP: **127.0.0.1**
* 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: OpenSSH X11 Forwarding Port Hijacking (Low)  
  
#### Severity Rationale:  
The severity of the OpenSSH X11 Forwarding Port Hijacking vulnerability is classified as low due to the limited impact it has on the security and integrity of the X11 forwarding feature in OpenSSH. This vulnerability in OpenSSH before version 5.1, when the X11UseLocalhost configuration setting is disabled, sets the SO\_REUSEADDR socket option, 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.  
  
The low severity indicates that while this vulnerability can lead to X11 forwarding port hijacking, it may not directly result in unauthorized access or compromise of critical data. However, it still presents a risk to the availability and integrity of the X11 forwarding feature.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by local users on affected platforms, it could lead to the hijacking of the X11 forwarding port via a bind to a single IP address. While the impact may be limited, unauthorized access to the X11 forwarding port could potentially disrupt X11 forwarding operations or impact the availability of X11 services within the organization.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH X11 Forwarding Port Hijacking vulnerability, local users on certain platforms can hijack the X11 forwarding port via a bind to a single IP address when the X11UseLocalhost configuration setting is disabled. This could potentially disrupt X11 forwarding operations or allow unauthorized access to the X11 forwarding port. While the impact may be low, the vulnerability could still lead to service disruption.  
  
In conclusion, the low severity of the OpenSSH X11 Forwarding Port Hijacking vulnerability indicates a lower risk level compared to critical vulnerabilities. It is important to address this vulnerability to prevent potential X11 forwarding port hijacking and protect the availability and integrity of X11 forwarding services within the organization.

#### 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 related to X11 forwarding port hijacking in OpenSSH before version 5.1, it is recommended to update OpenSSH to a version beyond 5.1. This update should address the issue where OpenSSH sets the SO\_REUSEADDR socket option when the X11UseLocalhost configuration setting is disabled, potentially allowing local users on certain platforms to hijack the X11 forwarding port via a bind to a single IP address. By updating to the latest version of OpenSSH, the vulnerability can be fixed, reducing the risk of X11 forwarding port hijacking. Additionally, it is important to review and configure X11 forwarding settings securely to prevent unauthorized access and port hijacking. Regularly monitoring for security updates and patches for OpenSSH is essential to address known vulnerabilities and maintain a secure SSH environment.

### Finding 37

* Affected IP: **127.0.0.1**
* Affected port and service: **22** (service **ssh**)
* Related CVE: **CVE-2008-1657**
* Severity score: **0.0** (**MEDIUM** severity)

## Finding severity rationale

### Vulnerability Severity Rationale  
  
#### Vulnerability: OpenSSH ForceCommand Bypass via .ssh/rc (Medium)  
  
#### Severity Rationale:  
The severity of the OpenSSH ForceCommand Bypass vulnerability is classified as medium due to the potential risk it poses to the security and access control mechanisms in OpenSSH. This vulnerability in OpenSSH versions 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.  
  
The medium severity indicates that while this vulnerability can lead to bypassing access controls and potentially unauthorized actions, it may not directly result in complete system compromise or critical data exposure. However, it still presents a significant risk to the integrity and security of the ForceCommand directive.  
  
#### Impact on the Organization:  
If this vulnerability is exploited by remote authenticated users, it could lead to the bypassing of the ForceCommand directive in OpenSSH, potentially allowing unauthorized actions or commands to be executed. Bypassing access controls could lead to unauthorized access to resources or compromise the security of the system.  
  
#### Attacker's Result by Exploiting the Vulnerability:  
By exploiting the OpenSSH ForceCommand Bypass vulnerability via the .ssh/rc session file, remote authenticated users can bypass the ForceCommand directive in OpenSSH, potentially executing unauthorized commands or actions. This could lead to unauthorized access to resources, compromise of security controls, or unauthorized actions within the system. While the impact may be limited, the vulnerability could still lead to unauthorized actions.  
  
In conclusion, the medium severity of the OpenSSH ForceCommand Bypass vulnerability highlights the importance of enforcing access controls and preventing unauthorized actions within OpenSSH. It is crucial to address this vulnerability to mitigate the risk of bypassing access controls and protect the integrity of the ForceCommand directive within the organization.

#### 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 related to bypassing the ForceCommand directive in OpenSSH versions 4.4 up to versions before 4.9, it is recommended to update OpenSSH to a version beyond 4.9. This update should address the issue where remote authenticated users can bypass the ForceCommand directive in sshd\_config by modifying the .ssh/rc session file. By updating to the latest version of OpenSSH, the vulnerability can be fixed, reducing the risk of unauthorized command execution. Additionally, it is important to review and secure the configuration settings, including the use of ForceCommand, to prevent unauthorized access and command bypass. Regularly monitoring for security updates and patches for OpenSSH is essential to address known vulnerabilities and maintain a secure SSH environment.