Critical Vulnerabilities Report

Prepared by: Nicolas Dauria
Date: March 3, 2025

1. VNC Server 'password' Password

Vulnerability Identification:

- **Name:** VNC Server 'password' Password
- **Plugin ID:** 61708
- **Severity:** Critical
- **Type:** Remote
- **Published Date:** August 29, 2012
- **Modified Date:** September 24, 2015
- **Risk Factor:** Critical
- **CVSS v2.0 Base Score:** 10.0
- **CVSS v2.0 Vector:** AV:N/AC:L/Au:N/C:C/I:C/A:C

Description:

The VNC server running on the remote host is secured with a weak password. Specifically, Nessus was able to log in using VNC authentication with the password "password." This vulnerability allows a remote, unauthenticated attacker to exploit this weakness and gain full control of the system. The use of a default or easily guessable password like "password" on the VNC server poses a significant security risk, as it can be exploited without any prior authentication, potentially leading to unauthorized access and complete system compromise.

Affected Systems:

- **Port:** 5900/tcp/vnc
- **Host(s):** 192.168.56.105
- **Vulnerability Information:**
    - Default Account: True
    - Exploited by Nessus: True

Potential Impact:

This vulnerability is classified as critical due to its CVSS v2.0 base score of 10.0, indicating the highest level of severity. A remote, unauthenticated attacker could exploit this flaw to gain full control of the affected system. Potential consequences include data breaches, unauthorized access, data manipulation, or the deployment of malware. The use of the default password "password" significantly increases the likelihood of exploitation, as it is a widely known and easily guessable credential.

Solution:

To mitigate this vulnerability, take the following steps:

- Secure the VNC server with a strong, unique password adhering to best practices (e.g., minimum length of 12 characters, including a mix of uppercase, lowercase, numbers, and special characters).
- If supported, implement multi-factor authentication (MFA) to add an additional layer of security.
- Restrict access to the VNC server using network firewalls or access control lists (ACLs) to limit exposure.
- Regularly audit and update passwords across the system to prevent similar vulnerabilities.

2. SSL Version 2 and 3 Protocol Detection

Vulnerability Identification:

- **Vulnerability Title:** SSL Version 2 and 3 Protocol Detection
- **Plugin ID:** 20007
- **Severity:** Critical
- **Risk Factor:** Critical
- **CVSS Base Score:** 9.8
- **CVSS Vector:** CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
- **Published:** October 12, 2005
- **Modified:** April 4, 2022

Description:

The remote service accepts connections using SSL 2.0 and/or SSL 3.0, both of which are outdated and cryptographically flawed protocols. These versions are vulnerable to exploits such as insecure padding schemes and CBC ciphers, enabling attackers to conduct man-in-the-middle attacks or decrypt communications between the service and clients. An attacker could downgrade a connection to use SSL 2.0 or 3.0, exploiting these weaknesses (e.g., via the POODLE attack). As per PCI DSS v3.1, SSL versions no longer meet the definition of strong cryptography and should be disabled entirely.

Affected Systems:

- **Host:** 192.168.56.105
- **Ports:** 443 (https), 8443 (https)
- **Details:** The server supports at least one cipher suite for SSL 2.0 and SSL 3.0, including medium-strength ciphers (e.g., DES with 56-bit encryption).

Potential Impact:

With a CVSS v3.0 base score of 9.8, this vulnerability presents a critical risk. An attacker could exploit the cryptographic weaknesses to intercept and decrypt sensitive data, impersonate the server, or manipulate communications. This could lead to data breaches, loss of trust, and non-compliance with security standards like PCI DSS. The widespread documentation of these flaws increases the likelihood of exploitation.

Solution:

To address this vulnerability:

- Disable SSL 2.0 and SSL 3.0 protocols on the affected server by updating the application's configuration.
- Upgrade to TLS 1.2 or higher, using only approved, strong cipher suites (e.g., AES with 128-bit or higher encryption).
- Review and align with PCI DSS requirements for secure communication protocols.
- Apply necessary patches or updates as recommended in the referenced resources.

3. Blind Shell Backdoor Detection

Vulnerability Identification:

- **Vulnerability Title:** Blind Shell Backdoor Detection
- **Plugin ID:** 51988
- **Severity:** Critical
- **Risk Factor:** Critical
- **CVSS v3.0 Base Score:** 9.8
- **CVSS v3.0 Vector:** CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
- **CVSS v2.0 Base Score:** 10.0
- **CVSS v2.0 Vector:** CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C
- **Published Date:** February 15, 2011
- **Modified Date:** April 11, 2022
- **Type:** Backdoors
- **Family:** Backdoors
- **Version:** 1.10

Description:

The "Blind Shell Backdoor Detection" vulnerability indicates a critical security issue on the scanned system. A shell is listening on port 1524/tcp without requiring any authentication. This backdoor allows an attacker to connect to the remote port and send commands directly, potentially gaining unauthorized access to the system. The absence of authentication mechanisms makes this vulnerability highly exploitable, posing a severe risk to the confidentiality, integrity, and availability of the system.

Affected Systems:

- **Host:** 192.168.56.105
- **Port:** 1524/tcp
- **Service:** Wild shell

Potential Impact:

This vulnerability carries a CVSS v3.0 base score of 9.8 and a CVSS v2.0 score of 10.0, reflecting its near-maximum severity. An attacker could exploit this backdoor to execute arbitrary commands with root privileges, as demonstrated by the Nessus scan output: uid=0(root) gid=0(root) groups=0(root). This level of access could lead to complete system compromise, data breaches, installation of malicious software, theft of sensitive data, or

use of the system as a launch point for further network attacks. The ease of exploitation and potential impact make this a top-priority issue.

Solution:

To mitigate this vulnerability, implement the following steps:

1. Investigate the system for signs of compromise by reviewing logs, network traffic, and unauthorized activities.
2. If compromise is suspected, reinstall the system to remove all malicious components.
3. Harden the system by disabling unnecessary services, closing unused ports, and implementing strong authentication mechanisms.
4. Apply regular updates and patches to address known vulnerabilities and backdoors.
5. Deploy intrusion detection/prevention systems to monitor network traffic and block unauthorized access attempts.