reckner

Index

The Rpcgssd Service Should Be Disabled.

Description:

The `rpcgssd` service is related to the NFS (Network File System),

which often involves Kerberos authentication.

Impact:

- If your system does not require NFS or you are not using Kerberos for authentication, leaving `rpcgssd` enabled could pose unnecessary security risks. Unnecessary services can be potential entry points for attackers.
- Disabling it could slightly reduce the system's resource usage.

Solution:

- `rpcgssd` is considered a static service that always starts during boot and cannot be simply disabled.
- If you want to prevent `rpcgssd` from starting during boot, you should mask the service.

```
systemctl mask rpcgssd.service
```

- For unmasking it

```
systemctl unmask rpcgssd.service
```

Ensure SSH Idle Timeout Interval is Configured.

Description:

SSH (Secure Shell) idle timeout intervals are a security feature that automatically terminates an SSH session after a specified period of inactivity.

Impact:

Without a timeout, an unauthorized user could potentially access another user's SSH session if they leave their computer unlocked.

Solution:

The solution is to configure the SSH daemon to automatically disconnect idle sessions after a certain period.

You can configure the SSH Idle Timeout Interval by setting the `ClientAliveInterval` and `ClientAliveCountMax` parameters in your SSH daemon configuration file (`/etc/ssh/sshd_config`).

```
sudo nano /etc/ssh/sshd_config
```

```
ClientAliveInterval 500  # No greater than 900 seconds
ClientAliveCountMax 0
```

```
sudo systemctl restart sshd
```

Ensure only Approved MAC Algorithms Are Used

Description:

- When data is transmitted over an SSH connection, the MAC algorithm uses a secret key and the message to generate a MAC, which is then sent along with the message to the receiver.
- To ensure data integrity and authenticity between the SSH client and server.

Impact:

Leads to security vulnerabilities

Unapproved or weak MAC algorithms can be exploited in SSH attacks, potentially allowing an attacker to capture sensitive data or credentials.

Solution:

To ensure only approved MAC algorithms are used, you can configure your SSH server by editing the `sshd_config` file.

Open the SSH configuration file:

```
sudo nano /etc/ssh/sshd_config
```

Specify approved MAC algorithms (replace with your approved list):

```
MACs hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,umac-128-etm@openssh.com
```

These are the list of algorithms that are allowed by default.

- It includes both secure options (like `hmac-sha2-512-etm@openssh.com`) and less secure ones (like `hmac-sha1`).
  - You should remove less secure algorithms from the list, such as those using SHA-1.

Save the file and restart the SSH service:

```
sudo systemctl restart sshd
```

This configuration will enforce the use of strong, approved MAC algorithms, enhancing the security of your SSH communications.

Ensure at/cron is Restricted to Authorized Users

Description:

The **at** and **cron** are both scheduling utilities in Linux systems:

- *at*: Executes commands at a specified time once.
- *cron*: Runs jobs automatically at regular intervals,

Impact:

- Unprivileged users may gain unauthorized access to schedule and modify at and cron jobs.
- Allowing them to execute commands or scripts with elevated privileges.

Solution:

To restrict at and cron usage to authorized users, follow these steps:

```
sudo rm /etc/cron.deny /etc/at.deny
```

```
sudo vi /etc/cron.allow
```

Add authorized users to cron and at allow lists

```
Echo "user1\nuser2" | sudo tee /etc/cron. Allow /etc/at. Allow
```

Replace user1, user2 with user names

Remove any existing deny lists

```
Sudo rm -f /etc/cron. Deny /etc/at. Deny
```

The Default Setting for Accepting Source Routed Packets Should Be Disabled for Network Interfaces.

Description:

- About disabling the acceptance of source-routed packets.
- Source routing is a feature that allows the sender to specify the route the packet takes through the network.
- Useful for troubleshooting like (path analysis, Network mapping, Performance testing), it can also pose security risks.

Impact:

- Attackers could exploit this to map your network, or redirect traffic for malicious purposes.

Solution:

- Open the sysctl configuration file in a text editor:

```
sudo nano /etc/sysctl.conf
```

- Add the following line to disable source routing:

```
net.ipv4.conf.default.accept_source_route = 0
```

- Load the new sysctl settings to apply the changes:

```
sudo sysctl -p
```

Windows Firewall: Private: Outbound Connections' is Set to 'Allow (default)

Description:

- Controls the behavior of outbound connections on a private network.

- By default, Windows Firewall allows all outbound connections unless there is a specific rule to block them.

Impact:

- If you change this setting to block outbound connections,
  - Applications might not be able to connect to the internet, which could affect their functionality.
  - you might encounter numerous prompts asking for permission to allow applications to connect to the network, which can be cumbersome for users.

Solution:

Press Win + R, type gpedit. Msc, and press Enter.

Go to Computer Configuration > Policies > Windows Settings > Security Settings > Windows Firewall with Advanced Security.

In the left pane, click on Windows Firewall with Advanced Security.
Double-click on Windows Firewall Properties.

Switch to the Private Profile tab.
Locate Outbound connections and set it to Allow (default)