README - Sicherheitsanwendungen konfigurieren: aa_config.py

Übersicht

Das Skript aa_config.py automatisiert die Konfiguration mehrerer Sicherheitsanwendungen und die Einrichtung von Firewall-Regeln auf einem Linux-Server. Es richtet die **Uncomplicated Firewall (UFW)** ein und verwaltet die Dienste **Snort**, **Suricata**, **ClamAV** und **RKHunter**, um den Server zu schützen und den Netzwerkverkehr sowie Systemintegrität zu überwachen.

Voraussetzungen

- **Python 3.6+**: Das Skript erfordert mindestens Python 3.6.
- **Administratorrechte**: Root- oder sudo-Berechtigungen sind notwendig, um Firewall-Einstellungen zu ändern und Systemdienste zu verwalten.
- **Sicherheitsanwendungen**: Folgende Pakete müssen auf dem Server installiert sein:
  - **UFW**: Firewall
  - **Snort**: Intrusion Detection System
  - **Suricata**: Netzwerk-Sicherheitsüberwachung
  - **ClamAV**: Antivirus-Software
  - **RKHunter**: Rootkit-Scanner

Installationsbefehle:

```
sudo apt-get install ufw snort suricata clamav clamav-daemon rkhunter
```

Funktionsweise

1. Shell-Befehle ausführen (run_command)

Die Funktion run_command(command) übernimmt die Ausführung von Shell-Befehlen im Terminal. Sie prüft, ob der Befehl erfolgreich abgeschlossen wurde, und gibt eine Fehlermeldung aus, falls dies nicht der Fall ist.

```
def run_command(command):
    try:
        subprocess.run(command, check=True, shell=True)
        print(f"Erfolgreich ausgeführt: {command}")
    except subprocess.CalledProcessError as e:
        print(f"Fehler bei der Ausführung von {command}: {e}")
```

2. UFW konfigurieren (configure_ufw)

Diese Funktion richtet die Firewall ein, um den Zugriff auf bestimmte Dienste und Ports zu ermöglichen, während andere Verbindungen blockiert werden. Die Konfiguration umfasst:

- **Blockieren** eingehender Verbindungen
- **Erlauben** ausgehender Verbindungen
- Freigabe von Ports für:
    - **SSH** (22/tcp)
    - **LDAP** (389/tcp)
    - **LDAPS** (636/tcp)
    - **HTTP** (80/tcp)
    - **HTTPS** (443/tcp)
    - **ClamAV Daemon** (3310/tcp)
    - **Wazuh-Agent** (55000/tcp)

```
def configure_ufw():
run_command("sudo ufw default deny incoming")
run_command("sudo ufw default allow outgoing")
run_command("sudo ufw allow 22/tcp")
run_command("sudo ufw allow 389/tcp")
run_command("sudo ufw allow 636/tcp")
run_command("sudo ufw allow 80/tcp")
run_command("sudo ufw allow 443/tcp")
run_command("sudo ufw allow 3310/tcp")
run_command("sudo ufw allow 55000/tcp")
run_command("sudo ufw logging on")
run_command("sudo ufw enable")
run_command("sudo ufw status verbose")
```

3. Sicherheitsdienste konfigurieren

Das Skript konfiguriert und startet verschiedene Sicherheitsdienste:

Snort konfigurieren: Intrusion Detection System

```
def configure_snort():
```

```
        run_command("sudo systemctl enable snort")
        run_command("sudo systemctl start snort")
```

Suricata konfigurieren: Alternative zu Snort für Netzwerküberwachung

```
    def configure_suricata():
        run_command("sudo systemctl enable suricata")
        run_command("sudo systemctl start suricata")
```

ClamAV konfigurieren: Antivirus-Dienst zur Bedrohungserkennung

```
    def configure_clamav():
        run_command("sudo systemctl enable clamav-daemon")
        run_command("sudo systemctl start clamav-daemon")
```

RKHunter konfigurieren: Überprüfung des Systems auf Rootkits

```
        def configure_rkhunter():
            run_command("sudo rkhunter --update")
            run_command("sudo rkhunter –propupd")
```

Hauptfunktion (main())

Die Hauptfunktion führt alle oben beschriebenen Konfigurationen nacheinander aus.

```
    def main():
            print("Starte die Konfiguration von SSH, LDAP und
    Sicherheitsanwendungen...")
            configure_ufw()
           configure_snort()
            configure_suricata()
            configure_clamav()
            configure_rkhunter()
            print("Konfiguration abgeschlossen.")
```

Nutzung

1. **Skript ausführen**:

   sudo python3 aa_config.py

Das Skript muss mit sudo-Rechten ausgeführt werden, um Firewall-Einstellungen und Systemdienste zu verwalten.

- **Statusüberprüfung**:
- UFW-Status:

  sudo ufw status verbose

- ***Status der Sicherheitsdienste:***

  sudo systemctl status snort

  - ***sudo systemctl status suricata***
  - ***sudo systemctl status clamav-daemon***

Fehlerbehebung

- **Fehler bei der Ausführung von Befehlen**: Wenn Befehle fehlschlagen, stellt sicher, dass die entsprechenden Pakete installiert sind und das Skript mit Administratorrechten ausgeführt wird.
- **Überprüfen der Systemprotokolle**: Bei Problemen mit Diensten kannst du die Systemprotokolle einsehen:

  sudo journalctl -xe

Lizenz

Dieses Skript steht unter der MIT-Lizenz.

Die MIT-Lizenz ist eine sehr einfache und populäre Open-Source-Lizenz, die Entwicklern erlaubt, den Code frei zu verwenden, zu modifizieren und weiterzugeben. Dabei gelten folgende Bedingungen:

1. **Erlaubnisse**: Der Code darf kostenlos verwendet, kopiert, verändert und weitervertrieben werden, auch in kommerziellen Projekten.
2. **Pflicht**: In jeder Kopie oder jedem abgeleiteten Werk muss der ursprüngliche Copyright-Hinweis sowie die Lizenz erhalten bleiben.
3. **Haftungsausschluss**: Der ursprüngliche Autor übernimmt keine Haftung für Schäden, die durch die Verwendung des Codes entstehen.