Apache Guacamole (Clientless Remote Desktop)

In diesem Kapitel wird die Architektur und Einrichtung von **Apache Guacamole** dokumentiert. Das System fungiert als “Clientless Remote Desktop Gateway” und ermöglicht SSH-, RDP- und VNC-Verbindungen direkt über den Webbrowser, ohne dass zusätzliche Software auf dem Endgerät installiert werden muss.

1. Systemarchitektur

Guacamole besteht nicht aus einem einzelnen Programm, sondern aus mehreren miteinander verbundenen Schichten:

- **Webbrowser:** Die Benutzeroberfläche für den Anwender.
- **Nginx (Reverse Proxy):** Übernimmt das Routing und leitet den Datenverkehr sicher weiter.
- **Apache Tomcat:** Der Webserver, der die Java-basierte Webanwendung von Guacamole ausführt.
- **guacd:** Das Herzstück des Systems. Ein Proxy-Dienst im Hintergrund, der den Web-Traffic in echte SSH- oder RDP-Protokolle übersetzt.
- **MariaDB:** Die Datenbank zur Speicherung von Benutzerdaten, Passwörtern und Verbindungseinstellungen.

2. Installation & Konfiguration

Schritt 1: Datenbank-Vorbereitung (MariaDB)
Für die Verwaltung der Benutzer wurde eine dedizierte Datenbank (`guacamole_db`) erstellt. Anschließend wurden die erforderlichen Guacamole-Tabellen (Schemas) importiert und Benutzerrechte definiert.

Schritt 2: Tomcat & Java-Anwendung
Da das aktuelle Betriebssystem (Debian 13) Kompatibilitätsprobleme mit der neuesten Tomcat-Version (Tomcat 10) verursachte, wurde eine alternative Lösung implementiert:

- Die ältere, aber stabile Version **Tomcat 9** wurde manuell im Verzeichnis `/opt/tomcat9` installiert.
- Die Guacamole-Webanwendung (die `.war`-Datei) wurde erfolgreich in dieses Verzeichnis bereitgestellt.

Schritt 3: Der Übersetzungsdienst (guacd)
Der Dienst `guacd` wurde installiert, um die Verbindungen zu übersetzen.

- **Kritische Anpassung:** Anfänglich versuchte der Dienst nur über IPv6 zu kommunizieren. In der Konfigurationsdatei (`guacd.conf`) wurde er explizit so konfiguriert, dass er nur über das lokale IPv4-Interface (`127.0.0.1`) lauscht.

3. Nginx Reverse Proxy Integration

Um das System professionell und ohne Eingabe von speziellen Ports (wie Port 8080) über den Standard-Webport erreichbar zu machen, wurde Nginx konfiguriert:

- **WebSocket Support:** Diese Funktion wurde im Reverse Proxy zwingend aktiviert. Ohne WebSockets würde die ständige Echtzeit-Verbindung zwischen Browser und Server abbrechen.

4. Fehlerbehebung & Tests (Troubleshooting)

Während der Einrichtung traten technische Herausforderungen auf, die erfolgreich gelöst wurden:

- **Zertifikats-Ungenauigkeiten:** Bei der Verbindung zu Ziel-Servern wurden die SSH-Schlüssel als “unbekannte Quelle” eingestuft und blockiert.
  - **Lösung:** In den Guacamole-Verbindungseinstellungen wurden Parameter wie “Ignore Server Certificate” aktiviert und die “Host Keys” manuell hinterlegt.
- **Falsche Limit-Fehler:** Wenn eine Verbindung aufgrund von Zertifikatsfehlern fehlschlug, meldete das System fälschlicherweise ein “erreichtes Verbindungslimit” (False-Positive). Dies wurde bei der Fehlersuche berücksichtigt.

Erfolgreiche Tests:
Nach der Fehlerbehebung wurden lokale SSH-Verbindungen auf den eigenen Server sowie externe SSH- und RDP-Verbindungen auf weitere Ziel-Server im internen Netzwerk erfolgreich über den Webbrowser durchgeführt.

From:
https://wiki.berkayhasip.com/ - **BerkayWiki**

Permanent link:
https://wiki.berkayhasip.com/doku.php?id=project:local:guacamole

Last update: **2026/03/18 14:10**