Ubuntu Server/Apache Guacamole

< Ubuntu Server

Guacamole [1] is an HTML5 web application that provides access to desktop environments using remote desktop protocols (such as VNC or RDP).

Contents

Installation

Building from source is normally discouraged but due to licensing reasons, the RADIUS plugin is not available with the Docker installation method.

guacamole-server (guacd)

Pre-requisites

```
# apt install build-essential
# apt install libcairo2-dev libjpeg-turbo8-dev libpng-dev libtool-bin uuid-dev \
              libavcodec-dev libavformat-dev libavutil-dev libswscale-dev freerdp2-dev libpango1.0-dev \
              libssh2-1-dev libtelnet-dev libvncserver-dev libwebsockets-dev libpulse-dev libssl-dev \
              libvorbis-dev libwebp-dev
```

Build and install

Date	Version	Notes
05 Apr 2024	1.5.4	systemd unit file has the path to the daemon at: /usr/local/sbin/guacd, rather than following --prefix. The User is set to daemon which can't write to its own home directory /usr/sbin

Download the guacamole-server-x.y.z.tar.gz and guacamole-client [2] and untar.gz to /usr/src.

```
# cd /usr/src/guacamole-server-1.5.5
# ./configure --with-systemd-dir=/etc/systemd/system --prefix=/opt/guacamole-server
# make
# make install
```

References

Edit the stock systemd unit file:

```
# systemctl edit guacd
```

Add the three lines between the comments and exit the editor. The two ExecStart= clears the current value and then sets it. User most be a system user that has correct file permissions.

```
### Anything between here and the comment below will become the new contents of the file

[Service]
ExecStart=
ExecStart=/opt/guacamole-server/sbin/guacd -f
User=local-admin

### Lines below this comment will be discarded
```

Create a configuration file /etc/guacamole/guacd.conf [3]

```
# guacd configuration file
# https://guacamole.apache.org/doc/gug/configuring-guacamole.html#configuring-guacd
# JG 09 Apr 2024

[daemon]
log_level = info

[server]
bind_host = localhost
bind_port = 4822
```

Enable and start the service:

```
# systemctl enable guacd
# systemctl start guacd
```

Create */etc/ld.so.conf.d/guacamole.conf*:

```
# Guacamole libs
/opt/guacamole-server/lib/
```

and run # *ldconfig -v* Failing to carry out this step will result in guacd logging entries like this:

```
guacd[23305]: WARNING:#011Support for protocol "rdp" is not installed
```

guacamole-client

Pre-requisits

```
# apt install default-jdk
# apt install maven
# apt install tomcat9
```

Ensure Maven can use a webproxy. Edit */etc/maven/settings.xml*:

```
  <proxies>
    <!-- proxy
     | Specification for one proxy, to be used in connecting to the network.
     |-->
    <proxy>
      <id>Blueloop Proxy</id>
      <active>true</active>
      <protocol>http</protocol>
      <username>webproxy</username>
      <password>s187dtt</password>
      <host>proxy.blueloop.net</host>
      <port>8080</port>
      <nonProxyHosts>local.net</nonProxyHosts>
    </proxy>

  </proxies>
```

Fix the Maven environment. Without this step you will get errors from mvn about being unable to run javadoc or find a $JAVA_HOME: Create /etc/profile.d/maven.sh:

```
export JAVA_HOME=/usr/lib/jvm/default-java
export M2_HOME=/opt/maven
export MAVEN_HOME=/opt/maven
export PATH=${M2_HOME}/bin:${PATH}
```

... make it executable and source it for this session:

```
# chmod +x /etc/profile.d/maven.sh
# source /etc/profile.d/maven.sh
```

Build and install

```
$ cd /usr/src/gaucamole-client-1.5.4
$ mvn package -Plgpl-extensions
```

Copy the .war file from */usr/src/guacamole-client-1.4.4/guacamole/target* to */var/lib/tomcat9/webapps/guacamole.war* Create the local configuration directories:

```
# mkdir -p /etc/guacamole/{extensions,lib}
```

Database authentication

The Postgres extension will be in the source tree after Maven has built the package somewhere similar to this:

/usr/src/guacamole-client-1.5.5/extensions/guacamole-auth-jdbc/modules/guacamole-auth-jdbc-postgresql/target/guacamole-auth-jdbc-postgresql-1.5.5.jar

Copy that .jar file /etc/guacamole/extensions

There are also some .sql files at: ./extensions/guacamole-auth-jdbc/modules/guacamole-auth-jdbc-postgresql/schema/

Download the Postgres JDBC driver [4] and put it in /etc/guacamole/lib

Create a configuration file at */etc/guacamole/guacamole.properties*:

```
# guacamole client web app configuration
# JG 09 Apr 2024

# post upgrade to ensure permissions are applied to any new tables in the database:
# su postgres
# psql -d guacamole_db
# GRANT SELECT,INSERT,UPDATE,DELETE ON ALL TABLES IN SCHEMA public TO guacamole_user;
# GRANT SELECT,USAGE ON ALL SEQUENCES IN SCHEMA public TO guacamole_user;

guacd-hostname: localhost
guacd-port:     4822

postgresql-hostname: localhost
postgresql-database: guacamole_db
postgresql-username: guacamole_user
postgresql-password: guacamole

postgresql-user-password-min-length: 8
postgresql-user-password-require-multiple-case: true
postgresql-user-password-require-symbol: true
postgresql-user-password-require-digit: true
postgresql-user-password-prohibit-username: true

postgresql-auto-create-accounts: true
```

Install Postgres

```
# apt install postgresql
```

Make sure the .sql files from the source are available. Create the database:

```
# su postgres
$ createdb guacamole_db
$ cat *.sql | psql -d guacamole_db -f -
```

Create a user and grant permissions:

```
$ psql -d guacamole_db
guacamole_db=# CREATE USER guacamole_user WITH PASSWORD 'guacamole';
guacamole_db=# GRANT SELECT,USAGE ON ALL SEQUENCES IN SCHEMA public TO guacamole_user;
guacamole_db=# GRANT SELECT,INSERT,UPDATE,DELETE ON ALL TABLES IN SCHEMA public TO guacamole_user;
```

RADIUS authentication [5]

As with postgresql suport, you need to install the RADIUS extension. For example:

```
# find /usr/src/guacamole-client-1.5.5/ -name "*.jar" | grep radius
# cp /usr/src/guacamole-client-1.5.5/extensions/guacamole-auth-radius/target/guacamole-auth-radius-1.5.5.jar /etc/guacamole/extensions
```

Add config similar to this to */etc/guacamole/guacamole.conf*:

```
radius-hostname: ### ip or name of RADIUS ###
radius-shared-secret: ### preshared key ###
radius-auth-protocol: pap
radius-timeout: 120
```

Postgres database support already has auto create users set. Either use the GUI to pre-populate usernames

Logging [6]

Create */etc/guacamole/logback.xml*. This example causes Guacamole to log at the debug level. Rename the file to something like logback.xml.DISABLED to go back to the defaults:

```
<configuration>

    <!-- Appender for debugging -->
    <appender name="GUAC-DEBUG" class="ch.qos.logback.core.ConsoleAppender">
        <encoder>
            <pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36} - %msg%n</pattern>
        </encoder>
    </appender>

    <!-- Log at DEBUG level -->
    <root level="debug">
        <appender-ref ref="GUAC-DEBUG"/>
    </root>

</configuration>
```

Initial testing

Open port 8080 temporarily and restart tomcat:

```
# firewall-cmd --zone=public --add-port=8080/tcp
# systemctl restart tomcat9
```

Login to the web GUI:

http://example.co.uk:8080/guacamole (http://remote.blueloop.net:8080/guacamole) guacadmin, guacadmin

If you can't login check the output of this as tomcat starts up:

```
# journalctl -f -u tomcat9
```

Reverse proxy with nginx [7]

Configure Tomcat

/etc/tomcat9/server.xml: Edit the connector, so that it listens on localhost only, set URIEncoding and change the reported server name:

```
<Connector address="127.0.0.1" port="8080" protocol="HTTP/1.1"
               connectionTimeout="20000"
               URIEncoding="UTF-8"
               redirectPort="8443"
               server="WWW Server"/>
```

In the same file add two valves (towards the bottom, inside *<host>*) The second one stops tomcat from giving away too much information when exposed to the web:

```
<Valve className="org.apache.catalina.valves.RemoteIpValve"
   internalProxies="127.0.0.1"
   remoteIpHeader="x-forwarded-for"
   remoteIpProxiesHeader="x-forwarded-by"
   protocolHeader="x-forwarded-proto" />

<Valve className="org.apache.catalina.valves.ErrorReportValve"
   showReport="false"
   showServerInfo="false"/>
```

Amend the *DefaultServlet* definition in */etc/tomcat9/web.xml*:

```
    <servlet>
        <servlet-name>default</servlet-name>
        <servlet-class>org.apache.catalina.servlets.DefaultServlet</servlet-class>
... other init-param ...
        <!-- JG 01 May 2024 -->
        <init-param>
            <param-name>showServerinfo</param-name>
            <param-value>false</param-value>
        </init-param>
        <!-- JG 01 May 2024 -->
        <load-on-startup>1</load-on-startup>
    </servlet>
```

Install and configure

Install nginx

```
# apt install nginx
```

Configure nginx - create */etc/nginx/sites-available/local*. Remove the default symlink in */etc/nginx/sites-enabled* and symlink *local.conf* into *sites-enabled*

```
# nginx reverse proxy for Guaamole

server {

        listen 80  default_server;
        listen [::]:80 default_server;

        server_name remote.example.co.uk;

        root /var/www/html;
        index index.html index.htm index.nginx-debian.html;
        server_name _;

        location / {
                proxy_pass http://localhost:8080/guacamole/;

                proxy_buffering    off;
                proxy_http_version 1.1;
                proxy_set_header   X-Forwarded-For $proxy_add_x_forwarded_for;
                proxy_set_header   Upgrade $http_upgrade;
                proxy_set_header   Connection $http_connection;
                client_max_body_size 1g;
                access_log           off;
        }
}
```

Set *servertokens* to *off* in */etc/nginx/nginx.conf*:

```
http {
    ...
    server_tokens off;
    ...
}
```

Restart the web proxy:

```
# systemctl restart nginx
```

https://remote.example.co.uk (https://remote.blueloop.net)

Branding [8][9]

Download the Guacamole supplied branding example [8]. Full instructions are supplied. The guacamole-ext documentation [9] explains how to add additional HTML to Guacamole, including disclaimers and other text.

Recreate the Postgres database

Make sure the .sql files are available in */etc/guacamole*.

```
# cd /etc/guacamole
# su postgres
$ dropdb guacamole_db
$ createdb guacamole_db
$ cat *.sql | psql -d guacamole_db -f -
$ psql -d guacamole_db
# GRANT SELECT,USAGE ON ALL SEQUENCES IN SCHEMA public TO guacamole_user;
# GRANT SELECT,INSERT,UPDATE,DELETE ON ALL TABLES IN SCHEMA public TO guacamole_user;
```

Login as *guacadmin*/*guacadmin*. Create a new user and assign all permissions. Logout and in again as the new user and delete *guacadmin*.

Making a user an administrator

This is how to edit the database to add administrative roles to an existing user account. Run psql as the postgres user:

```
# su - postgres
# psql guacamole_db
```

List the entity table to find the entity_id, then list the current system permissions (local-admin is a full admin already). Insert additional permissions (here I add all permissions to a-gerdesj):

```
guacamole_db=# select * from guacamole_entity;
 entity_id |    name     | type
-----------+-------------+------
         4 | local-admin | USER
         5 | gerdesj     | USER
         6 | a-gerdesj   | USER

guacamole_db=# select * from guacamole_system_permission;
 entity_id |       permission
-----------+-------------------------
         4 | CREATE_CONNECTION
         4 | CREATE_CONNECTION_GROUP
         4 | CREATE_SHARING_PROFILE
         4 | CREATE_USER
         4 | CREATE_USER_GROUP
         4 | ADMINISTER

guacamole_db=# INSERT INTO guacamole_system_permission VALUES (6, 'CREATE_CONNECTION');
guacamole_db=# INSERT INTO guacamole_system_permission VALUES (6, 'CREATE_CONNECTION_GROUP');
guacamole_db=# INSERT INTO guacamole_system_permission VALUES (6, 'CREATE_SHARING_PROFILE');
guacamole_db=# INSERT INTO guacamole_system_permission VALUES (6, 'CREATE_USER');
guacamole_db=# INSERT INTO guacamole_system_permission VALUES (6, 'CREATE_USER_GROUP');
guacamole_db=# INSERT INTO guacamole_system_permission VALUES (6, 'ADMINISTER');
```

References

1. https://guacamole.apache.org/doc/gug/ - Apache Guacamole manual
2. https://guacamole.apache.org/releases/ - Guacamole releases for download

3. https://guacamole.apache.org/doc/gug/configuring-guacamole.html#configuring-guacd - guacd configuration file
4. https://jdbc.postgresql.org/ - Postgres JDBC driver
5. https://guacamole.apache.org/doc/gug/radius-auth.html - Guacamole documentation on RADIUS
6. https://logback.qos.ch/manual/configuration.html - Logback documentation
7. https://guacamole.apache.org/doc/gug/reverse-proxy.html - Proxying Guacamole - official docs
8. https://github.com/apache/guacamole-client/tree/main/doc/guacamole-branding-example - Guacamole branding example
9. https://guacamole.apache.org/doc/gug/guacamole-ext.html - guacamole-ext docs

Retrieved from ‘https://wiki.blueloop.net/w/index.php?title=Ubuntu_Server/Apache_Guacamole&oldid=13113’

This page was last modified on 28 June 2024, at 14:23.