OpenVPN Technologies Inc.

OpenVPN Access Server, Explanation of “post_auth” MAC address checking script

Authentication and role of post_auth

The post_auth programming hook in Access Server was put in to extend the possibilities of the Access Server to authenticate against a source of credentials. By default without using post_auth the following sources can be authenticated against in Access Server:

- LOCAL
    - This method relies on using an SQLite3 database that is stored on the Access Server's file system itself. It contains both the user properties and the user credentials. Management of credentials is handled in the 'user permissions' area of the admin web services of the Access Server.
- PAM
    - This is the system that is present on Linux systems, and is the default choice for the Access Server. This uses the credentials from the Linux system itself to authenticate to. Management of credentials is done in the console/SSH session in the Linux OS itself using standard tools like adduser, deluser, and passwd.
- LDAP
    - To connect to an OpenLDAP or Windows AD, the LDAP option is fairly easy to use. For security a proxy account can be made on the LDAP server through which the Access Server can talk to the LDAP server to perform authentication against the credentials stored in the LDAP server. Management of credentials is handled by using LDAP browsing and management tools, or in Windows AD by using the "Active Directory Users and Computers" program.
- RADIUS
    - Like LDAP, the credentials are stored in the RADIUS server, and management of credentials is handled by using the tools available for browsing and manipulating the contents of the RADIUS credentials database. Windows Server Active Directory supports the ability to allow the Access Server to query the AD over the RADIUS protocol, as is the case with LDAP.

The post_auth script is run during the authentication session where a user tries to log in at the Access Server from a compatible OpenVPN client. The script runs just after a user's credentials are checked and found to be working. Hence, post_auth - after authentication.

The script itself is a text file in the programming language Python. With it you can do all sorts of interesting things like implement a custom authentication system or in this case to add an extra filter to the login process. This MAC address checking script is designed to allow only users with a registered and correct MAC address to connect to the VPN server. The registration process is automated but can also be forced to be completely controlled by the administrator.

How it works

Whenever a user logs on and has no MAC address registered yet in the user properties database of the Access Server, then the MAC address will be stored in that database. Any following attempt to log in under that user account will require that the connection is made from the same MAC address. The MAC address is taken from the first ethernet network connection the VPN client finds. Only the OpenVPN Connect Client latest version is supported, but open source clients may also work (some have been tested to work).

How to install/update the script

The script itself is available from our website at the URL mentioned below, the script file itself has to be saved somewhere on the Access Server, even if just temporarily. The filename is not important, but let's say it is mac.py and that it is stored in the following folder: /root/mac.py

For your information, the script file is available from our website here:
http://swupdate.openvpn.net/scripts/post_auth_mac_address_checking.py

It could for example be retrieved with the program wget like so;

```
sudo su
wget http://swupdate.openvpn.net/scripts/post_auth_mac_address_checking.py -O /root/mac.py
```

Once you have the file and it is named and saved as mentioned then use the following command lines as a guide to load this script into the Access Server, and reload with the new settings:

```
sudo su
cd /usr/local/openvpn_as/scripts
./sacli -k auth.module.post_auth_script --value_file=/root/mac.py ConfigPut
./sacli start
```

If you make alterations to the mac.py file then you will need to use the above commands again to load the new version of the script into the configuration database, and to refresh the configuration of the Access Server again.

Restricting automatic registration of MAC addresses for new accounts

By default a user account without a registered MAC address, like for example a newly created account, will allow any computer to connect. Once the first connection is made, the MAC address of that system is stored and locked. The server will now only accept a connection for this user account from that MAC address.

If you do not want this process to occur automatically, and to for example limit registration only from a specific network, like a corporate network, you can use the first_login_ip_addr variable in the script. If the first_login_ip_addr variable is set to a specific IP address, then only first time login attempts will be accepted from that address. For example if you only want to allow MAC address registrations from one computer or network, enter that network's public IP address in this script. You can find this parameter in the script. For example:

first_login_ip_addr="123.45.67.89"

If it is left empty, then first time logins will be accepted from any IP address. The choice of adding this extra safety requirement is up to you. It is also a way of denying users to register themselves, forcing you as an administrator to register all accounts yourself. If you set it to an impossible value like "NONE" then all MAC address registrations must be done through the command line of the Access Server by an administrator.

Notes regarding connection profile types

This script works with all 3 types of connection profiles. That is, user-locked, server-locked, and autologin. All will adhere to the restrictions in this post_auth script. The Connect Client may end up looping endlessly when it is being denied access, in which case you need to stop the connection by opening the menu and clicking 'disconnect', after which the solution to this problem is to correct the MAC address registration.

Notes regarding client software

OpenVPN Connect Client for Windows and Macintosh is supported. The following clients have also been tested to work but are not under any guarantee of support by us:

- The open source client called 'openvpn' on Linux.
- The open source client called 'OpenVPN GUI' on Windows.
- The open source client called 'Tunnelblick' on Macintosh.
- Most open source clients based on 'openvpn' binary should work.

Please note that for all clients it is assumed that you are using a recent version. If you make a connection with your client software of choice and it turns out that it is not reporting a MAC address to the Access Server then the first thing to check is that you have the latest version of that software with the most recent version of the OpenVPN binary core that you can get.

How to uninstall the script

The script is loaded into the configuration database, and can be removed from there with the following commands. Note that this action does not remove the MAC addresses already saved in the database, but these will simply be ignored and not cause further issues.

```
sudo su
cd /usr/local/openvpn_as/scripts
./sacli -k auth.module.post_auth_script ConfigDel
./sacli start
```

How to unregister/reset a MAC address

If for some reason the user account is being used on another system or for some reason the MAC address has changed, you will need to remove the MAC address that is currently stored and locked for that particular user. To do this you can use the following commands to remove the saved MAC address for the user "exampleuser" and reload the server:

```
sudo su
cd /usr/local/openvpn_as/scripts
./sacli -u "exampleuser" -k "pvt_hw_addr" UserPropDel
./sacli start
```

How to manually register/update a MAC address

If you want to manually specify the address for an account, you can do so with the following commands, where "exampleuser" is the user account name and where "00:01:02:ab:cd:ef" is the MAC address. Please stick to lower case as upper case is not what is used for the MAC address check. Please keep in mind that this lower/upper case sensitivity is a factor especially when using LDAP, in which the LDAP server reports the name back during the login phase. In other words, LDAP case is leading.

```
sudo su
cd /usr/local/openvpn_as/scripts
./sacli -u "exampleuser" -k "pvt_hw_addr" -v "00:01:02:ab:cd:ef" UserPropPut
./sacli start
```

How to debug any problems

If you use the restriction to allow only MAC address regisrations from a specific IP address as explained in this file, then please undo that particular restriction and install/update the post_auth script again, and reload the settings again, as explained earlier in this file.

Also make sure you use the very latest software, as older software may not support the feature of reporting MAC address to the server.

Any information that the post_auth script outputs to the log file can be found in the /var/log/openvpnas.log (by default) on most Access Server setups. You can easily filter for specific messages from post_auth with these commands:

```
sudo su
cat /var/log/openvpnas.log | grep "POST_AUTH"
```

Conclusion

We hope you will find this guide useful. If you have any comments please leave them at our support ticket system at www.openvpn.net by signing in and clicking the support link.