Remote access to LIACS REL

This is a quick introduction on how to access servers in the LIACS Research and Education Lab (REL) remotely using ssh. More information can be found on the REL website, which is only accessible from within the campus network for security reasons. Using the information below under "I am at home and I want to access the REL website", you can access the REL website from outside of LIACS.

Information on how to setup a VPN connection can only be found on the REL website.

Some internal REL websites that use HTTPS, such as privgit.liacs.nl, docs.liacs.nl and ipsec.liacs.nl, require the CACert Root certificates to be installed in order for the secure connection to be verified successfully. At the bottom of this document, more information on CACert can be found.

Important: REL is maintained by various staff members at LIACS and not by ISSC. So for these services you cannot turn to ISSC for support.

SSH Access

All REL services are accessible through ssh.liacs.nl. You can login to ssh.liacs.nl using your ULCN account if you are affiliated with LIACS as staff member or student. From the shell you obtain on ssh.liacs.nl after successful login, you access other machines within REL using SSH or other means.

SSH actually offers more possibilities than just a shell on a remote server. Using SSH forwarding you can set up direct connections from your local machine to a machine in REL. Using SSH as SOCKS proxy server you can access various internal REL services directly from you local machine, relaying the traffic through ssh.liacs.nl. This way, internal websites such as privgit.liacs.nl and rel.liacs.nl can be access using the web browser running on your local computer (without going through the hassle of configuring VPN).

SSH forwarding

Let's say you want to access SSH (TCP port 22) on the data science lab server duranium.liacs.nl directly from your laptop, while at home. In this case, you can set up an SSH forward from a local port on your laptop to port 22 of duranium. This can be achieved with the following command:

```
ssh -L 1234:duranium:22 username@ssh.liacs.nl
```

This command will open port 1234 on your local laptop and forward any traffic to that port to duranium port 22 (SSH) through the SSH connection. After a successful login, it is now possible to access duranium by executing on your local laptop:

```
ssh -p 1234 username@localhost
```

Note that although you are connecting to localhost, because a connection is made to port 1234 all this traffic will be forwarded to duranium. So, you are connecting directly to duranium with all traffic being relayed through ssh.liacs.nl.

If you use such a forward often, you can also pre-configure it in your SSH configuration located at ~/.ssh/config:

```
Host ssh.liacs.nl
  User username  # replace "username" with your ULCN username
  LocalForward 1234 duranium:22
  # Note that multiple LocalForward lines may be specified
```

Now the command `ssh username@ssh.liacs.nl` will connect to the SSH server with the correct username (taken from the configuration file) and automatically create the port forward as specified in the configuration file.

Similarly, a configuration can be added to connecting to duranium through localhost:
Instead of ssh -p 1234 username@localhost it now suffices to write ssh localduranium.

Similar configurations can be made with Putty on Windows, see for instance https://www.akadia.com/services/ssh-putty.html.

**SSH as SOCKS server**

The SSH client can act as a SOCKS proxy server. Your local applications can be configured to use this SOCKS server as proxy server. This means that all traffic is first sent to ssh.liacs.nl through the SSH connection and then relayed to the destination from there. So, your local application will have the same network access as if it were running on ssh.liacs.nl. This means that the local application has access to the machines and services in REL.

In order to enable this, establish the SSH connection as follows:

```
ssh -D 7000 username@ssh.liacs.nl
```

After that, configure your local application to use a proxy server. This can be done per application, for instance web browsers often have Network Settings where a proxy server can be configured. Select “SOCKS proxy” as type, add localhost as hostname and 7000 as port number.

You can get ssh to connect to an internal computer hostname.liacs.nl through the proxy server as follows:

```
ssh -o ProxyCommand="nc -X 5 -x localhost:7000 %h %p" hostname.liacs.nl
```

You can also include this in your `~/.ssh/config` file:

```
Host hostname.liacs.nl
    ProxyCommand=nc -X 5 -x localhost:7000 %h %p
```

And simply use ssh hostname.liacs.nl on the command line.

Note that the SOCKS proxy server will only be available as long as the SSH disable the proxy server again!

**Frequent use cases that can be solved with SSH**

I am running a Jupyter notebook on tritanium port XYZ. How can I access this from my local laptop?

Set up an SSH forward through ssh.liacs.nl, see above. Pick a local port (for instance 6000) and connect it to port XYZ on tritanium:

```
ssh -L6000:tritanium:XYZ username@ssh.liacs.nl
```

If you use this regularly, you can pre-configure it in your SSH configuration file as described above.
I am at home and I want to access the REL website

Connect to ssh.liacs.nl with the SOCKS proxy server enabled:

`ssh -D 7000 username@ssh.liacs.nl`

and temporarily change the network settings in your web browser to use localhost port 7000 as SOCKS proxy server. After that enter rel.liacs.nl configured in your web browser, all websites that you access are accessed through ssh.liacs.nl.

I want to access privgit.liacs.nl without setting up VPN

This is possible using the SSH SOCKS proxy server. For accessing the privgit website, the instructions are equivalent to these for accessing rel.liacs.nl, see previous question.

For accessing the SSH port of privgit via the SSH SOCKS proxy, add the following to your SSH configuration ~/.ssh/config:

```
Host privgit.liacs.nl
  ProxyCommand=nc -X 5 -x localhost:7000 %h %p
```

This cryptic line tells SSH to use nc (netcat) as SOCKS client to connect to privgit.liacs.nl through the proxy server.

Installing CACert Root certificates

When connecting to certain internal REL websites (such as privgit.liacs.nl, docs.liacs.nl, ipsec.liacs.nl) over HTTPS your web browser might display a warning telling you that the secure connection could not be verified. This is caused by the fact that certificates of internal REL websites are signed using CACert and your browser does not contain the CACert Root Certificate.

This problem is solved by installing the CACert Root Certificates. These certificates can be found at http://www.cacert.org/index.php?id=3. You need to install both the Class 1 and Class 3 PKI keys. For information and instructions on installing these certificates refer to http://wiki.cacert.org/FAQ/ImportRootCert.

After a one-time installation of these certificates, your browser should now recognize that the connection is secure.