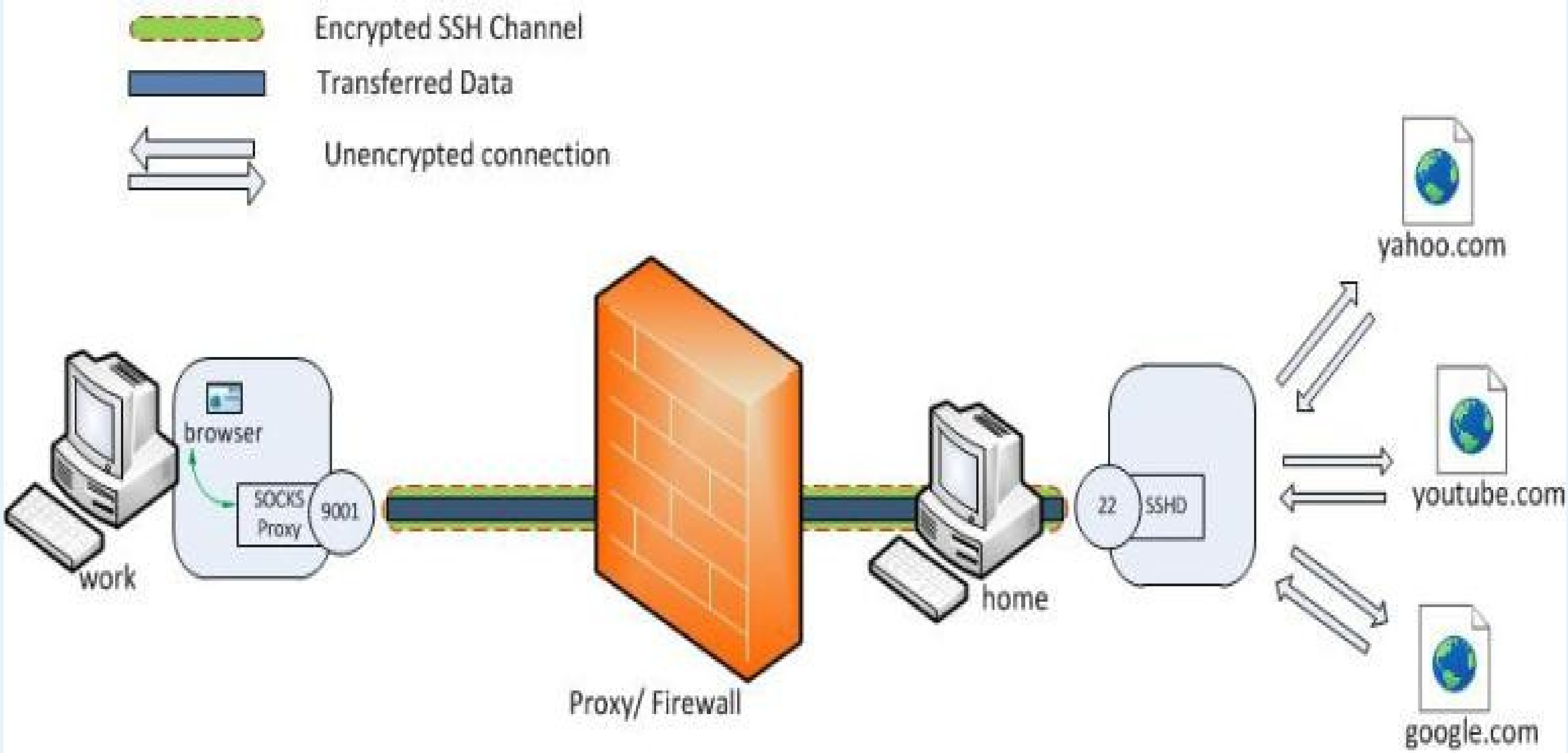
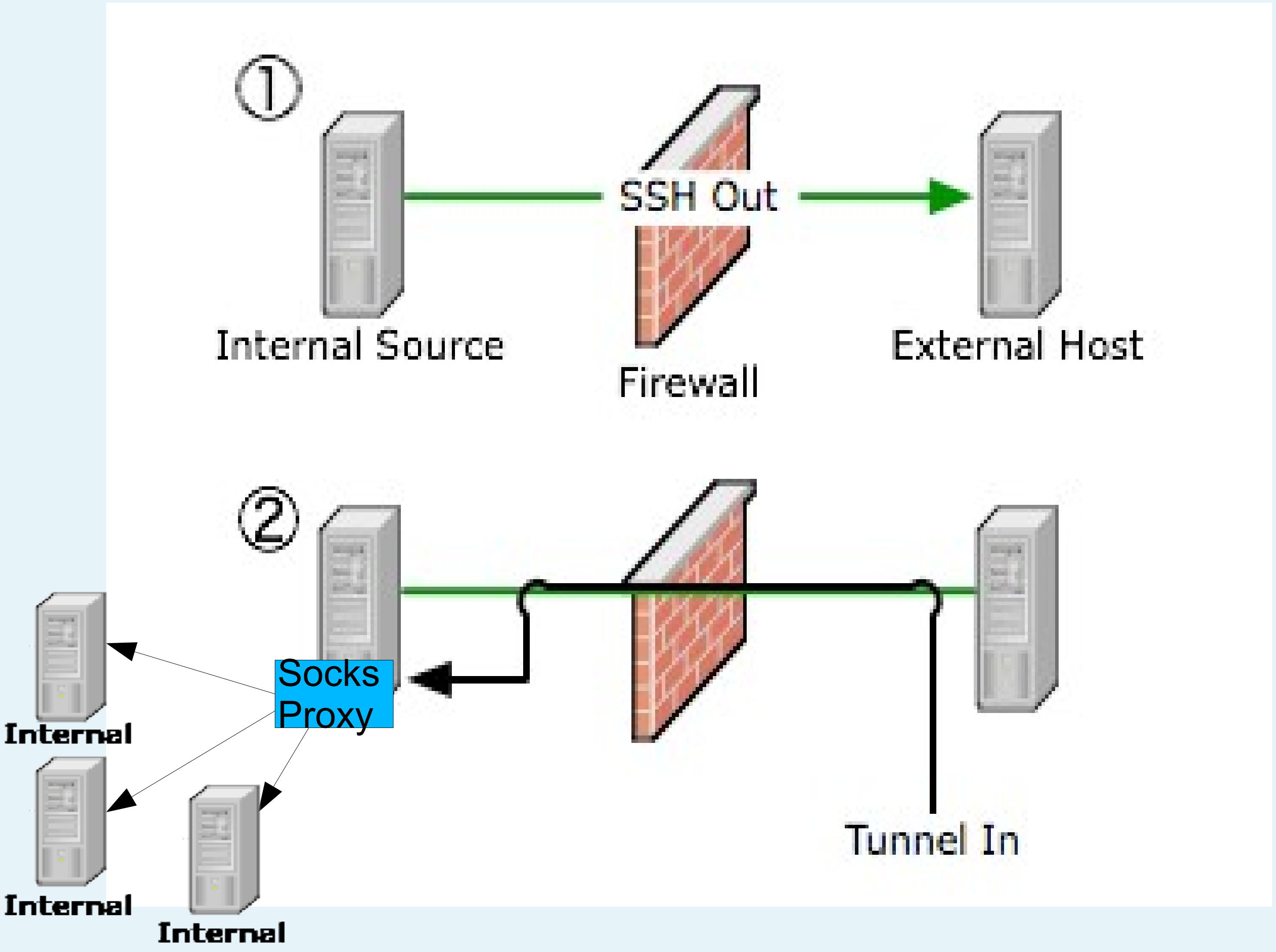
Reverse SSH Tunnels

St. Louis Unix Users Group

SSH Forward Tunnels



Reverse SSH Tunnels



Reverse SSH Tunnels

Firewalls may deny incoming SSH connections, but allow outgoing SSH. Using forwarding, an incoming path may be setup by tunneling access to the internal sshd server over a persistent outbound connection. Warning! This use of SSH may impose a security risk to the site in question, or be a security policy violation!

Connect from the internal system to the external host. Use the -R option to open port 2222 on the external host back to port 22 on the internal system.

```
    internal$> ssh -R 2222:127.0.0.1:22 external.example.org
```

On the external host, connect back in from the external host.

```
    external$> ssh -p 2222 127.0.0.1
```

Optionally use dynamic port forward to set up a socks proxy to the entire internal network.

```
    external$> ssh -N -D 127.0.0.1:2223 -p 2222 internaluser@127.0.0.1 &
```

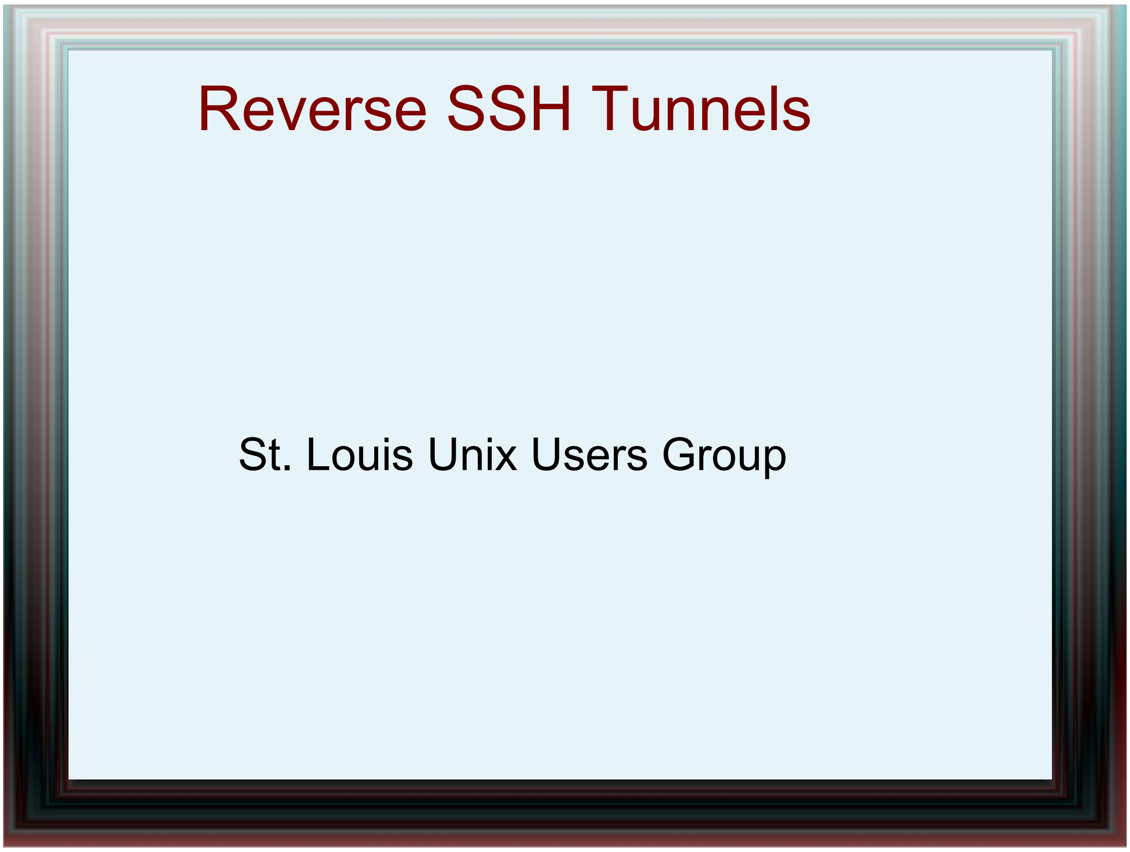
SSH to any host on the network

```
    external$>  ssh -o ProxyCommand='nc -x 127.0.0.1:2223 %h %p' user@internalip
```

Setting the NoHostAuthenticationForLocalhost yes option might be required to avoid key conflicts, if multiple connections are done using the localhost address. For more information, see ssh_config(5).

Good and Bad

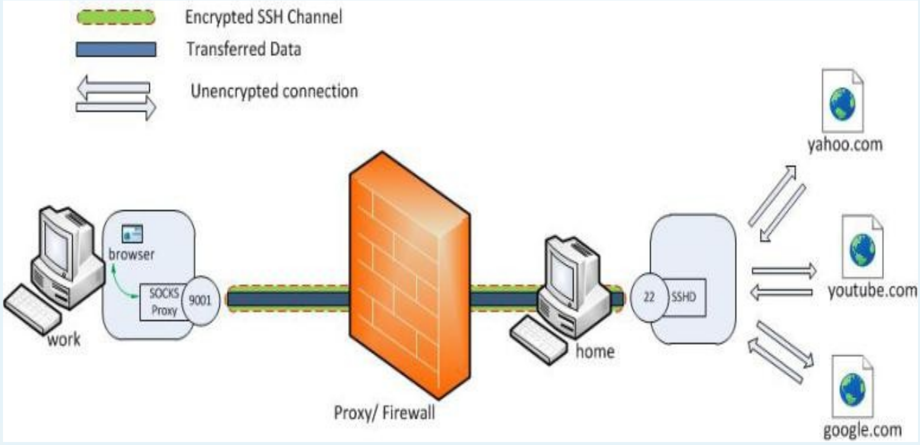
- Bad
  - Trojan Horses to allow access from outside the network bypassing proxy
  - “Inside Jobs” to breach security firewalls
- Good
  - Allow hosts behind a NAT firewall to be administered remotely
  - Reverse connections to a central server
  - Access to a private network through an exposed secured server preventing detection of the real address



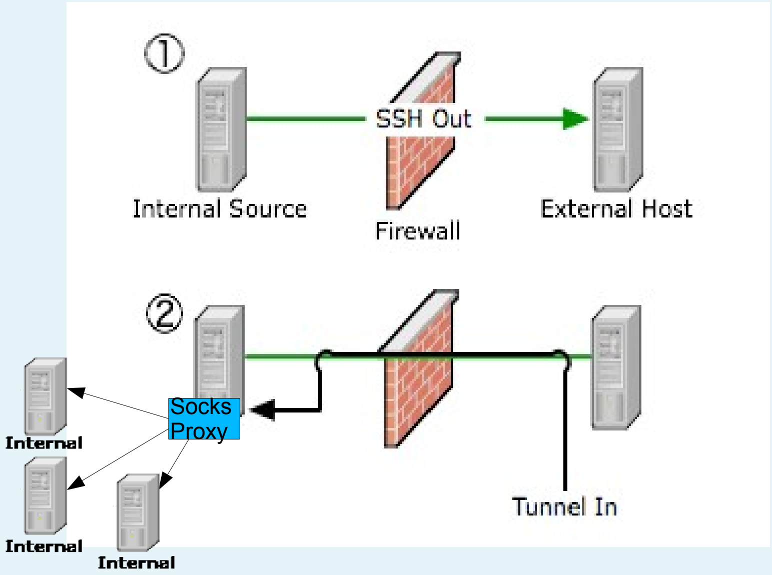
Reverse SSH Tunnels

St. Louis Unix Users Group

SSH Forward Tunnels



Reverse SSH Tunnels



Reverse SSH Tunnels

Firewalls may deny incoming SSH connections, but allow outgoing SSH. Using forwarding, an incoming path may be setup by tunneling access to the internal sshd server over a persistent outbound connection. Warning! This use of SSH may impose a security risk to the site in question, or be a security policy violation!

Connect from the internal system to the external host. Use the -R option to open port 2222 on the external host back to port 22 on the internal system.

```
internal$> ssh -R 2222:127.0.0.1:22 external.example.org
```

On the external host, connect back in from the external host.

```
external$> ssh -p 2222 127.0.0.1
```

Optionally use dynamic port forward to set up a socks proxy to the entire internal network.

```
external$> ssh -N -D 127.0.0.1:2223 -p 2222 internaluser@127.0.0.1 &
```

SSH to any host on the network

```
external$>  ssh -o ProxyCommand='nc -x 127.0.0.1:2223 %h %p' user@internalip
```

Setting the NoHostAuthenticationForLocalhost yes option might be required to avoid key conflicts, if multiple connections are done using the localhost address. For more information, see ssh_config(5).

Good and Bad

- Bad
  - Trojan Horses to allow access from outside the network bypassing proxy
  - “Inside Jobs” to breach security firewalls
- Good
  - Allow hosts behind a NAT firewall to be administered remotely
  - Reverse connections to a central server
  - Access to a private network through an exposed secured server preventing detection of the real address