SSH Doing It Right

Wouter D'Haeseleer - Nucleus

SSH Doing it right

What will I cover

- A small intro in the protocols
- Introduction in PKI
- A lot of practical tips on doing SSH better like:
    - Using SSH-keys
    - Logging / Audit
    - ssh certificates
    - jump hosts
    - harding tips
- Q/A

SSH Doing it right

And why do we have to tune SSH

SSH Doing it right

About me

Wouter D'Haeseleer

- Operations Engineer at Nucleus in a cool devops team
- Tech geek
- Father of 2 nice little boys

Who are you?

Who are you?

Want to follow on your PC ?

Want to follow on your PC ?

http://172.18.114.80:9090

Protocols , so much choise !

Protocols , so much choise !

Protocols , so much choise !

- SSH
  - A remote terminal session protocol like telnet, but encrypted and packed full of features
- FTP
  - Plain text protocol, everyone is able to see and alter your data
- FTPs
  - FTP with SSL, is the same as HTTP vs https
- sFTP
  - Like FTP but over an SSH connection
  - same features as FTP (Resume, Directory lists, ...)
- scp
  - is core in the SSH stack

But first some words about cryptography

But first some words about cryptography

But first some words about cryptography

Alice and bob

But first some words about cryptography

Symetric key encryption

But first some words about cryptography

Symetric key encryption

But first some words about cryptography

Symetric key encryption

But first some words about cryptography

Symetric key encryption

But first some words about cryptography

Symetric key encryption

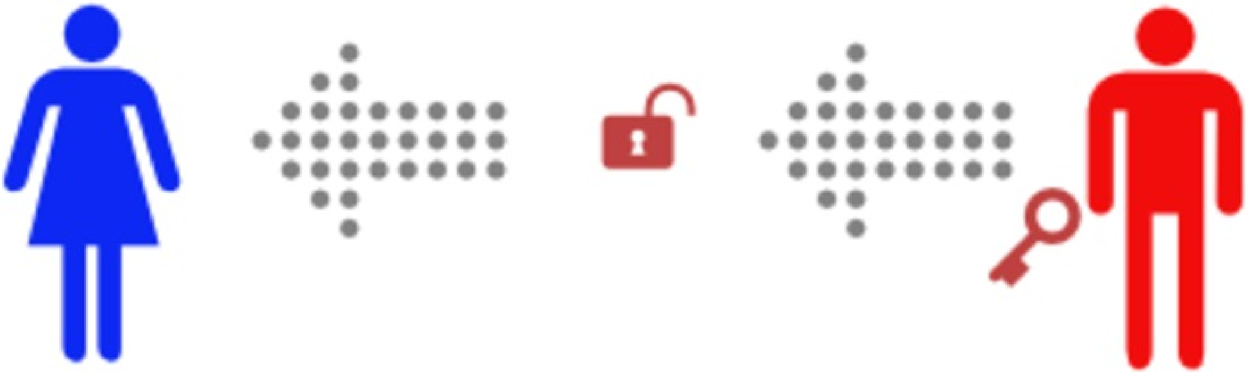
- In the internet world it's quite difficult to arrange a meeting and exchange a key
- This is where Asymmetric keys come into play

But first some words about cryptography

Asymmetric key encryption (PKI)

But first some words about cryptography

Asymmetric key encryption (PKI)



But first some words about cryptography

Asymmetric key encryption (PKI)

But first some words about cryptography

Asymmetric key encryption (PKI)

But first some words about cryptography

Asymmetric key encryption (PKI)

But first some words about cryptography

Asymmetric key encryption (PKI)

- Real world example:
  - Bob generates a key pair, consisting of his public key (red padlock) and private key (red key).
  - Bob then publishes his public key, and Alice fetches it (Bob mails his padlock to Alice).
  - Alice then generates a temporary symmetric key (the pair of orange keys) and uses Bob’s public key (red padlock) to securely send it to Bob.
  - Bob then uses his private key (red key) to unlock his copy of the symmetric key (orange key).
  - Bob and Alice can then use those symmetric keys to securely send messages back and forth.

But first some words about cryptography

symmetric vs asymmetric

Symmetrical	Asymmetrical
+ quick	+ no need to share THE encryption key
+ not resource intensive	+ Can be used for encryption an signing
+ Usefull small an big messages	– Very resource intesive
– need to sendover the key	– only useful for small messages

But first some words about cryptography

But....

- How can alice be sure the padlock received from bob and not from eve (The mailman)
    - or
- How can bob be sure the message was from alice and not from eve
- This is solved with certificates !

But first some words about cryptography

Certificates

The general idea

- Bob sends his key to a trusted party
- The trusted party verifies that bob is indeed bob and not eve (By doing manual / automated tests)
- The public key verified by the trusted party is a certificate
- If alice receives the certificate alice will see that the trusted party has confirmed that this is from bob
- Because alice also trusts the trusted party she can be really sure that it is bob

But first some words about cryptography

Certificates

- In the real world this is called a `server certificate`
    - Because it authenticates the server, so you as a user can be sure your talking to the right server it is verified by a trusted party.
- The same is possible for the client, this is then called a `client certificate`
    - A client certificate verifies the identity of the client

But first some words about cryptography

Enough please !

- This is all you have to know about PKI for now !

But first some words about cryptography

No passwords ! Use Keys

Guess this one is obvious

When using SSH as a user you should not use passwords We have SSH keys for that

Why ?

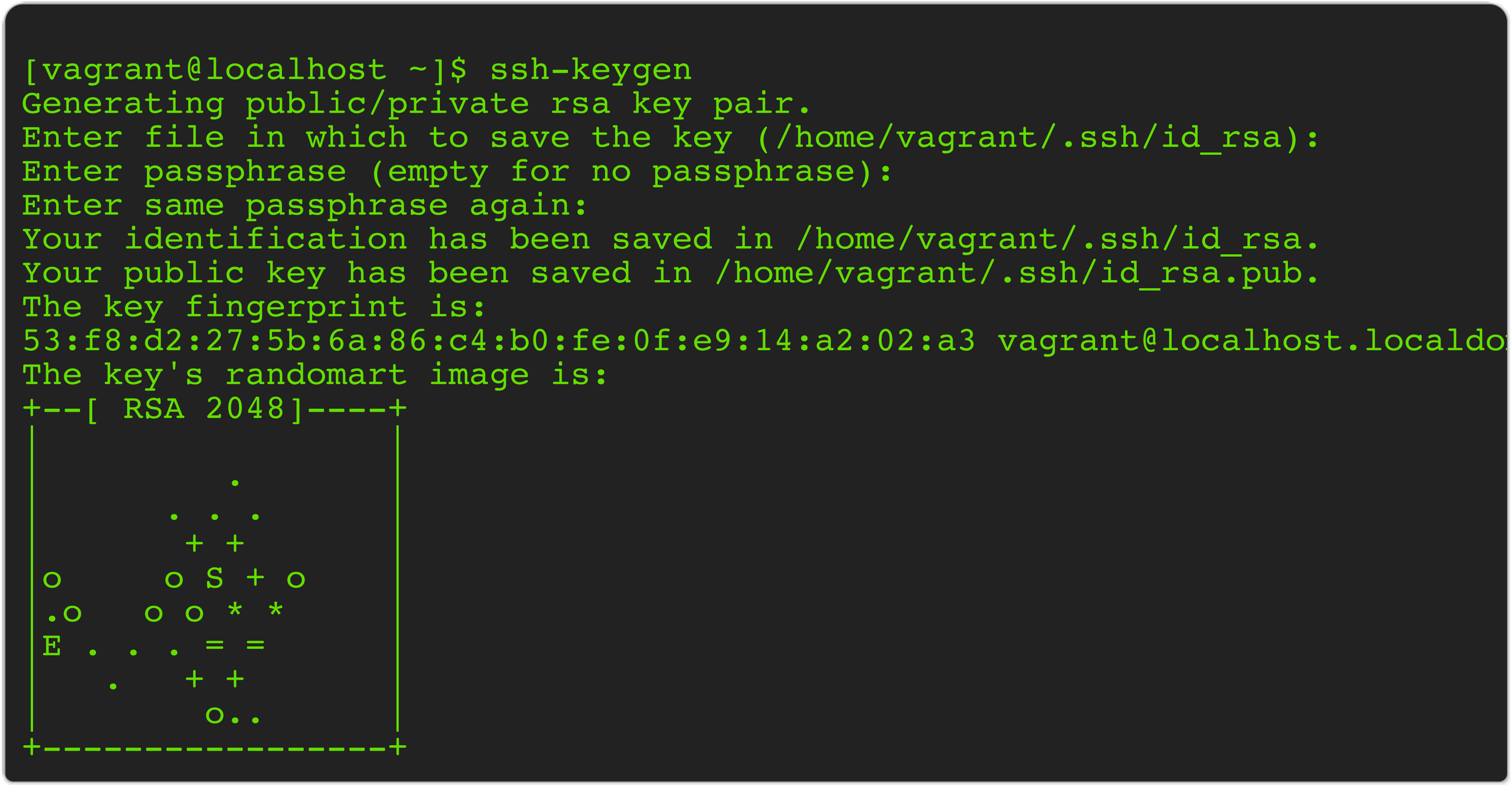
Just because it is a lot easier for you and a lot more secure Your not vulnerable for password attacks.

How ?

Let's have a look

But first some words about cryptography

First you have to create a key pair



```
[vagrant@localhost ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/vagrant/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/vagrant/.ssh/id_rsa.
Your public key has been saved in /home/vagrant/.ssh/id_rsa.pub.
The key fingerprint is:
53:f8:d2:27:5b:6a:86:c4:b0:fe:0f:e9:14:a2:02:a3 vagrant@localhost.localdo
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|         .       |
|      . . .      |
|       + +       |
|o     o S + o    |
|.o   o o * *     |
|E . . . = =      |
|   .   + +       |
|        o..      |
+-----------------+
```

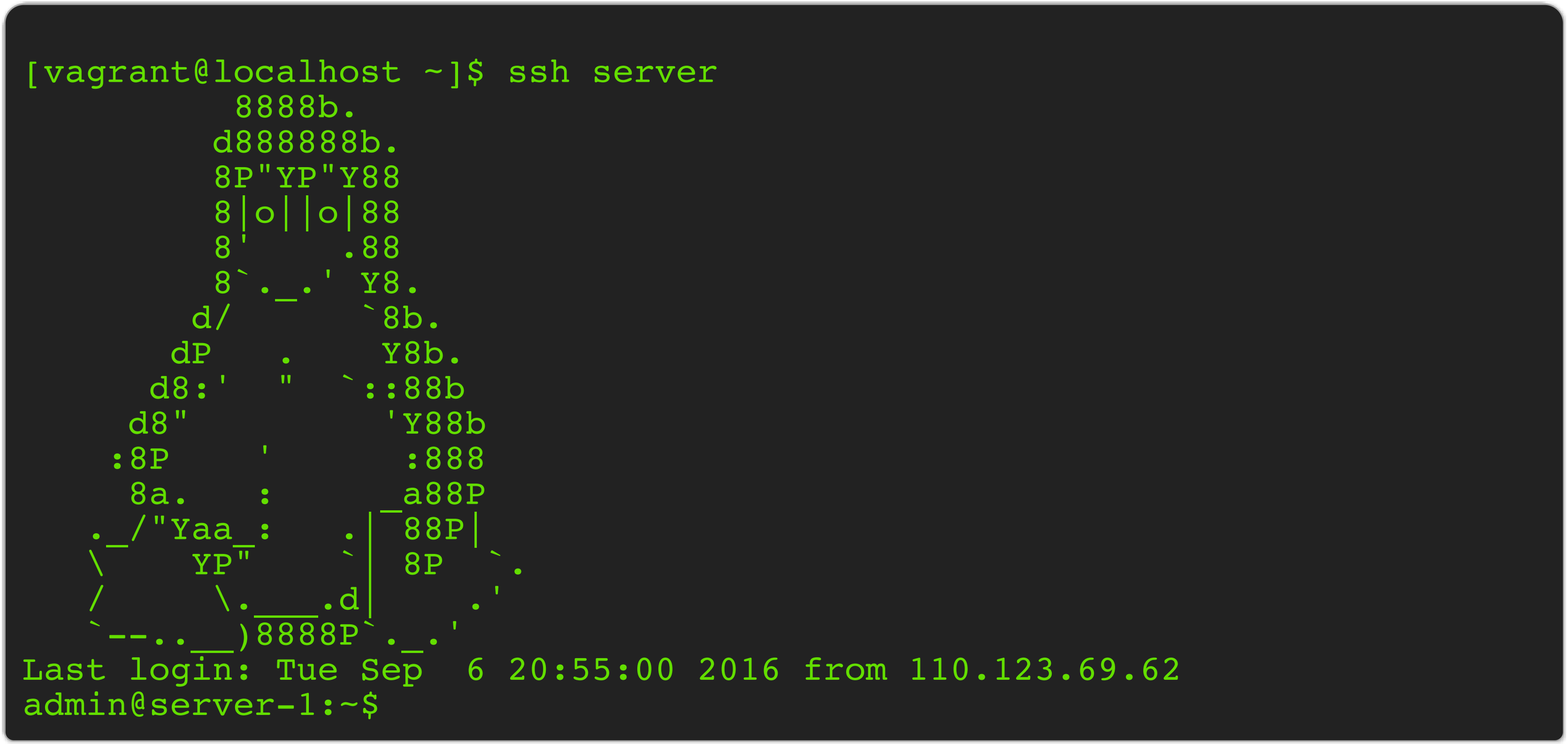
But first some words about cryptography

Install key on server

```
~:$ cat ~/.ssh/id_rsa.pub | ssh server cat >> ~/.ssh/authorized_keys

# Or of you are on linux

~:$ ssh-copy-id username@server.example.com
```

But first some words about cryptography

Now you are able to login



```
[vagrant@localhost ~]$ ssh server
          8888b.
         d888888b.
         8P"YP"Y88
         8|o||o|88
         8'    .88
         8`._.' Y8.
        d/      `8b.
       dP   .    Y8b.
      d8:'  "  `::88b
     d8"         'Y88b
    :8P    '      :888
     8a.   :     _a88P
   ._/"Yaa_:   .| 88P|
   \    YP"    `| 8P  `.
   /     \.___.d|    .'
   `--..__)8888P`._.'
Last login: Tue Sep  6 20:55:00 2016 from 110.123.69.62
admin@server-1:~$
```

But first some words about cryptography

So what's in it for you as drupal user

```
~:$ brew install homebrew/php/drush

~:$ cat ~/.drush/aliases.drushrc.php
   <?php
   $aliases['dev'] = array(
     'uri' => 'http://staging.reggaegeel.com',
     'root' => '/var/www/vhosts/staging.reggaegeel.com/htdocs',
     'remote-host' => 'staging.reggaegeel.com',
     'remote-user' => 'userforstaging'
   );
```

DEMO on drush

DEMO on drush

How to log / audit your users

How to log / audit your users

How to log / audit your users

Why the need, you have a history file

- History file is only written to disk when you logout
- You can change the history file in a current session to use `/dev/null` Thus disabling the history file
- By default a history file is read/write by the user itself. So a user is able to change / alter the history
- We can do it better then the default behaviour.

How to log / audit your users

Step 1 : Make sure files can only be append-only

```
chattr +a /home/user/.bash_history
chattr +a /home/user/.bash_profile
chattr +a /home/user/.bash_login
chattr +a /home/user/.profile
chattr +a /home/user/.bash_logout
chattr +a /home/user/.bashrc
```

How to log / audit your users

Step 2 : Set important variables read only

```
shopt -s histappend
readonly PROMPT_COMMAND="history -a"
readonly HISTFILE
readonly HISTFILESIZE
readonly HISTSIZE
readonly HISTCMD
readonly HISTCONTROL
readonly HISTIGNORE
```

How to log / audit your users

What about syslog ?

How to log / audit your users

Using a bash profile variable

```
readonly PROMPT_COMMAND='pwd=`pwd` && history 1 | \
                         /bin/sed "s:^  *[0-9]*  :$pwd :" | \
                         /usr/bin/logger \
                           -p local5.notice \
                           -t "$USER[$$] \
                           $SSH_CONNECTION"'
```

- This only works in the current bash session so in the following cases you loose this function:
    - When using SUDO
    - When the user starts an other shell and prefers not to use the default profile

How to log / audit your users

Using bash build in syslog option

Bash has support to sending the histfile also to syslog #

- Not a single distribution enables this option
- Thus you have to modify the source and compile bash
- So it also means you have to maintain the package yourself and put it in a repo

Only allow a certain commands

Only allow a certain commands

Only allow a certain commands

SSH_ORIGINAL_COMMAND

- Envioromnent variable set by ssh and contains the command a user wants to execute
- So we can create a script which checks if the command is allowed
- Let's have a look at an example

Only allow a certain commands

SSH_ORIGINAL_COMMAND – example code

```
#!/bin/sh
# Script: /usr/local/bin/wrapper.sh

case "$SSH_ORIGINAL_COMMAND" in
    "ps")
        ps -ef
        ;;
    "nodejs stop")
        /etc/init.d/nodejs stop
        ;;
    "nodejs start")
        /etc/init.d/nodejs start
        ;;
    *)
        echo "Sorry. Only these commands are available to you:"
        echo "ps, nodejs stop, nodejs start"
        exit 1
        ;;
esac
```

Only allow a certain commands

How to force this ?

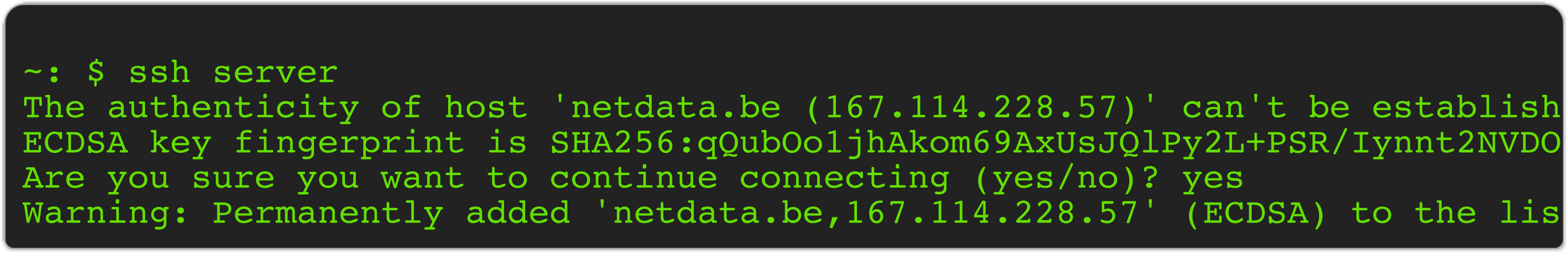
```
~: $ cat ~/.ssh/authorized_keys

command="/usr/local/bin/wrapper.sh",no-port-forwarding,
no-X11-forwarding,no-pty ssh-rs  AAAAB3NzaC1yc2EAAAABIw
p0KMipajKK468mfihpZHqmrMk8w+PmzTnJrZUFYZZNmLkRk+icn+m71
DdEHmza2cSf9WdiK7TGibGjZTE/Ez0IEhYRj5RM3dKkfYqitKTKlxVh
XNda7az6VqAJ/jtaBXAMTjHeD82xlFoghLZOMkScTdWmu47FyVkv/IM
1GjgX/I8s4307ds1M+sICyDUmgxUQyNF3UnAduPn1m8ux3V8/xAqPF+
bRuFlB0fbiAEsSu4+AkvfX7ggriBONBR6eFexOvRTBWtriHsCybvd6t
OpJHN8JYZLxCRYHOGX+sY+YGE4iIePKVf2H54kS5UlpC/fnWgaHbmu/
XsGYjYrAFnVw== Test key
```

SSH Server certificates

SSH Server certificates

SSH Server certificates

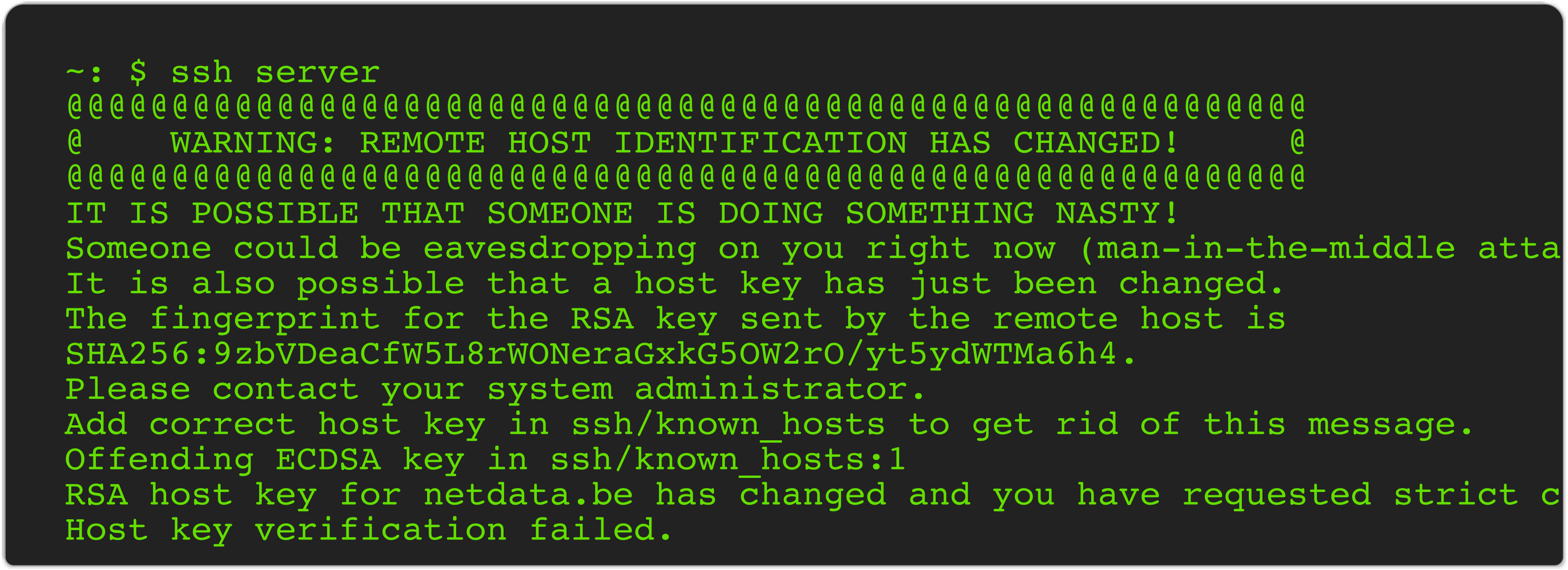
For every server you have to accept the host key



```
~: $ ssh server
The authenticity of host 'netdata.be (167.114.228.57)' can't be establish
ECDSA key fingerprint is SHA256:qQubOo1jhAkom69AxUsJQlPy2L+PSR/Iynnt2NVDO
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'netdata.be,167.114.228.57' (ECDSA) to the lis
```

SSH Server certificates

If the host is rebuild you will have troubles



```
~: $ ssh server
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle atta
It is also possible that a host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
SHA256:9zbVDeaCfW5L8rWONeraGxkG5OW2rO/yt5ydWTMa6h4.
Please contact your system administrator.
Add correct host key in ssh/known_hosts to get rid of this message.
Offending ECDSA key in ssh/known_hosts:1
RSA host key for netdata.be has changed and you have requested strict c
Host key verification failed.
```

SSH Server certificates

The solution to this is Server Certificates

- You sign every host key `/etc/ssh/ssh_host_rsa_key.pub` with a so called CA key (Our trusted party).
- You install the trusted party key inside your `known_hosts`

SSH Client certificates

SSH Client certificates

SSH Client certificates

Multiple server & multiple users

- You have to copy the keys to every server for each user.
    - This can become a nightmare to maintain if you don't use tools like puppet
- By using SSH client certificates is easy to do central mgmt

SSH Jump hosts

SSH Jump hosts

SSH Jump hosts

Consider the following

```
                                              +-------------+
                                        +---> |   Host X    |
                                        |     +-------------+
                                        |
+-------------+        +-------------+  |     +-------------+
|    You      +------->|   Host A    +------->|   Host Y    |
+-------------+        +-------------+  |     +-------------+
                                        |
                                        |     +-------------+
                                        +---> |   Host Z    |
                                              +-------------+
```

SSH Jump hosts

Several options are available

- Using a SSH Port forward
- Using an SSH agent
- Using a ProxyCommand option

SSH Jump hosts

Using a SSH Port forward

```
ssh -L 2222:hostx.example.org:22 hosta.example.org
ssh -p 2222 remoteuser@localhost
```

- This one is might be the most known option
- First we will open a Port forward by logging in to host A
- Second we can connect on localhost port 2222

SSH Jump hosts

Using a SSH agent forward

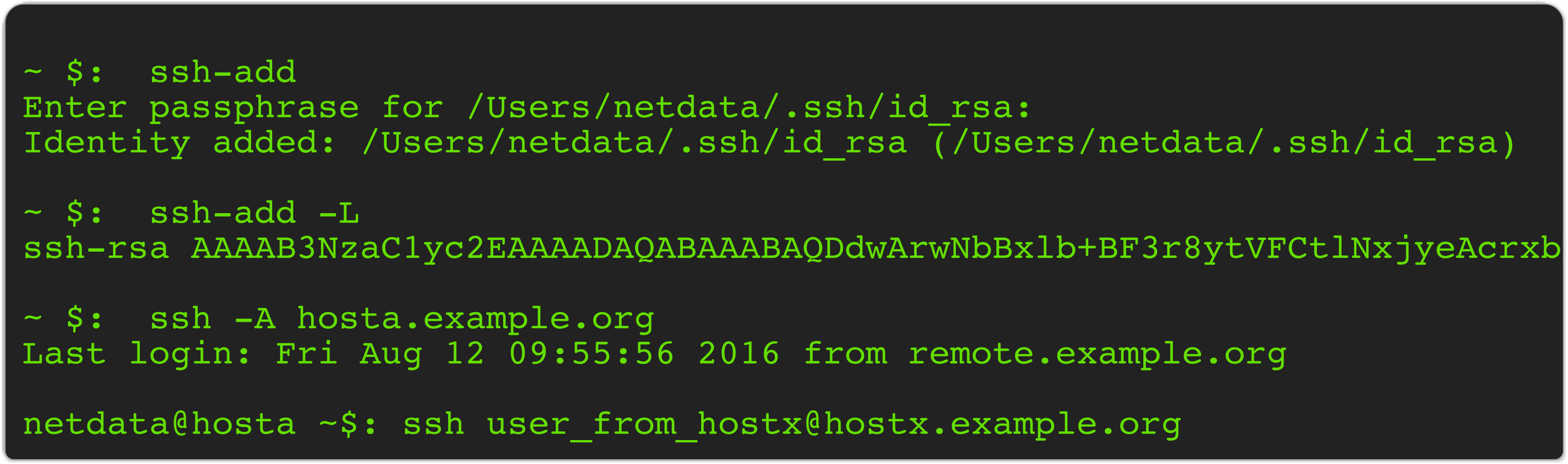
- Most of the linux distribution will have an SSH agent running when using a graphical env
- On OSX ssh-agent is also running. THis is handled by Keychain

> There are serious security issues with use ssh agent forwarding ! It is using a unix socket on disk, so anyone with root access is able to act on your behalf.

If you really want to use ssh agent forwarding I strongly advise you to make sure your SSH agent is configred to ask confirmation On linux this can be done like this `ssh-agent -c`

SSH Jump hosts

Using a SSH agent forward



```
~ $:  ssh-add
Enter passphrase for /Users/netdata/.ssh/id_rsa:
Identity added: /Users/netdata/.ssh/id_rsa (/Users/netdata/.ssh/id_rsa)

~ $:  ssh-add -L
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDdwArwNbBxlb+BF3r8ytVFCtlNxjyeAcrxb

~ $:  ssh -A hosta.example.org
Last login: Fri Aug 12 09:55:56 2016 from remote.example.org

netdata@hosta ~$: ssh user_from_hostx@hostx.example.org
```

SSH Jump hosts

Using the ProxyCommand

```
~ $: cat .ssh/config

Host host-a
  User your_username
  Hostname 10.0.0.5

Host host-x
  User your_username
  Hostname 192.168.0.1
  Port 22
  ProxyCommand ssh host-a nc %h %p
```

- A LOT easier then the previous methods.
- You can now directly ssh to `host-x` Your SSH client will authenticate to host-x as if it was directly reachable
- No security concerns here !

SSH hardening some easy tips

SSH hardening some easy tips

```
$ cat /etc/ssh/sshd_config

# Configure Idle Log Out Timeout Interval

ClientAliveInterval 300
ClientAliveCountMax 0

# Disable root Login via SSH

PermitRootLogin no

# Change SSH Port and Limit IP Binding

Port 9999
ListenAddress 192.168.1.5
ListenAddress 202.54.1.5
```

Q/A

Q/A

calibrate

IMALINK

corecrew
WE BUILD YOUR DIGITAL PRESENCE

nucleus
Hosting Solution Builder

Dropsolid
Makkelijk Digitaal Ondernemen

MEMBER OF SII GROUP
uniway
Business Critical Internet Solutions