Enhance OpenSSH for Fun and Security

Julien Pivotto

LinuxCon Europe
October 5, 2015

Match User roidelapluie

Julien Pivotto

- Sysadmin at inuits.eu
- FLOSS user since 2004
- DevOps believer
- *@roidelapluie* on irc/twitter/github

inuits.eu

World, 2015

Connected devices

- Mainframes
- Servers
- Virtual machines
- Containers
- IoT

Entrance Doors

- Physical Access
- Telnet
- RSH
- SSH
- HTTPS
- ...

SSH

- Dozens of implementations
- OpenSSH
- Dropbear (embedded)
- Closed-source
- ...

SSH

- Dozens of usecases
- Shell access and TCP Tunelling
- Code (git)
- File transfert (sftp)
- X terminal (x2go)
- Automation (ansible)
- ...

OpenSSH

OpenSSH

- Developed by the OpenBSD project
- Released first in 1995
- Server/Client implementation
- Included in BSD, Linux, Cygwin, Mac OS X, ...
- Available in many other platforms

Out of scope

- Firewalling, OS, …
- Basic tips: RootLogin, Pubkeys, …
- Crypto/Encryption/Key Exchanges https://stribika.github.io/2015/01/04/secure-secure-shell.html

Security

Common sense

- Do you need SSH? (immutable infra, containers...)
- KISS
- Chose what will get public IP and then exposition.. hypervisors vs vms?
- Port 22 is not Evil

Server-side

"Server config"

- /etc/ssh/sshd_config
- Restart of the service does not kill current ssh sessions

Allow/Deny rules

AllowUsers

```
AllowUsers jenkins
AllowUsers jenkins nagios@172.31.29.5
AllowUsers jenkins nagios@172.31.29.0/12
```

AllowUsers is exclusive

AllowGroups

```
AllowGroups staff jenkins
```

AllowGroups is exclusive

Allow* ordering

- DenyUsers
- AllowUsers
- DenyGroups
- AllowGroups

Match

- Match + conditions
- reads until next Match or EOF

Match

```
AllowGroups staff
Match Address 172.31.16.8
AllowGroups staff jenkins
```

Trust On First Use

TOFU

```
The authenticity of host ’example.com (93.184.216.34)’
   can’t be established.
ED25519 key fingerprint is SHA256:eIvxpj9aMSS/+
   Ed7NQZ9er/vyV17mabfiUxtgF2Q1X0.
Are you sure you want to continue connecting (yes/no)?
```

Trust on first use

- Who checks the key on the server?
- Who says no?
- Security fatigue

Alternative to TOFU (1/2)

- Automation
- Export keys from hosts
- Collect them from hosts
- Apply then to /etc/ssh/known_hosts

```
# saz/puppet-ssh - ASL 2.0
if $::sshrsakey {
  @@sshkey { "${::fqdn}_rsa”:
    ensure       => present,
    host_aliases => $host_aliases,
    type         => rsa,
    key          => $::sshrsakey,
  }
} else {
  @@sshkey { "${::fqdn}_rsa”:
    ensure       => absent,
  }
}
```

```
Sshkey <<| |>>
```

Alternative to TOFU (2/2)

- DNS
- Export keys in SSHFP DNS records
- Can be secured by DNSSEC
- https://github.com/jpmens/facts2sshfp

```
$ dig +short SSHFP example.com
1 1 F00A55CEA3B8E15528665A6781CA7C35190CF0
2 1 CC1F004DA60CF38E809FE58B10D0F22680D59D
```

```
ssh —o VerifyHostKeyDNS=yes example.com
```

```
The authenticity of host ’example.com (93.184.216.34)’
   can’t be established.
ED25519 key fingerprint is SHA256:eIvxpj9aMSS/+
   Ed7NQZ9er/vyV17mabfiUxtgF2Q1X0.
Matching host key fingerprint found in DNS
Are you sure you want to continue connecting (yes/no)?
```

Authorized keys

Licensed under a Creative Commons Attribution 2.0 License
https://www.flickr.com/photos/brenda-starr/4498078166

```
ssh-rsa AAsafgrewgBzhfadgthgfpoDtGlUBIYhzf user@desktop
```

- One key, one user
- Always with a password
- Distribute them in an automated way

```
from="172.21.32.4" ssh-rsa AAspoDtGlUBIYhzf ansible
no-port-forwarding,no-x11-forwarding,no-agent-forwarding ssh-rsa
   AAspDjeFJwFRf jenkins
```

```
ssh_authorized_key {
  ’jenkins’:
    type      => ’ssh–rsa’,
      key     => ’AAAAKZ6TwZl3ikhY42clyY/De7J’,
      user    => ’jenkins’,
}
```

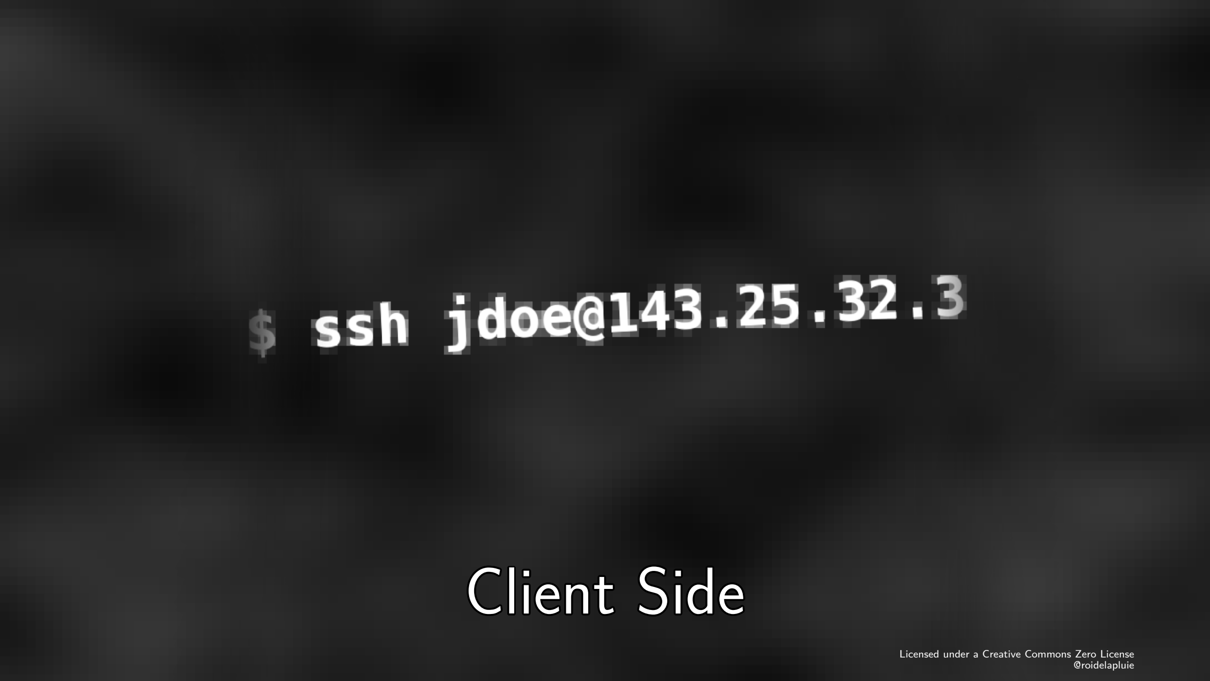
```
ssh_authorized_key {
  ’jenkins’:
    type      => ’ssh-rsa’,
      key     => ’AAAAKZ6TwZl3ikhY42clyY/De7J’,
      user    => ’jenkins’,
      options => ’from="192.168.10.1"’
}
```

Purge undefined keys!

```
user {
  ’jenkins’:
    purge_ssh_keys => true,
}
```

AuthorizedKeysCommand

- Script that takes username as arguments and returns authorized_keys
- Exemple reference: openssh-ldap RPM



```
$ ssh jdoe@143.25.32.3
```

Client Side

Client configuration

- $HOME/.ssh/config
- /etc/ssh/ssh_config

```
Host web1
Hostname web1.example.com
User roidelapluie
```

SSH Hops

```
Host web1
Proxycommand ssh proxy nc %h %p
Host proxy
Proxycommand ssh out nc %h %p
```

SSH Hops

- Acces restricted areas
- Keeps your private keys in your machine
- No need for agent forwarding

STOP! attach aerial before the power cord!
FUSED
FUSED
FUSED
SURGE PROTECTED
WITH SAFETY THERMAL FUSES
NEON 'ON' INDICATES SURGE PROTECTION OPERATIONAL
battery charger

Sockets

```
Host git.example.com
ControlMaster auto
ControlPath /tmp/ssh–%r@%h:%p
ControlPersist 5
```

SSH Sockets

- Speed up reconnection time
- Do not renegotiate each time
- Useful for git

Stopping OpenSSH

Send to background

```
<enter> ~ &
```

Pause

```
<enter> ~ <ctrl+z>
```

Kill the session

```
<enter> ~ .
```

Tunnels

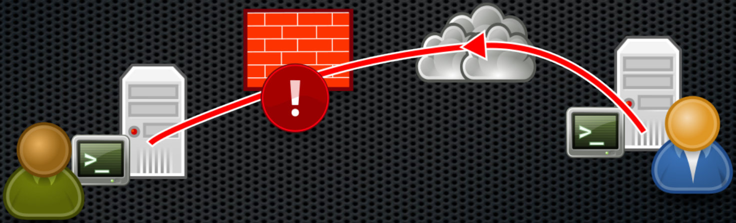
Tunnels

- TCP Tunnels
- SOCKS proxy

Tunnels

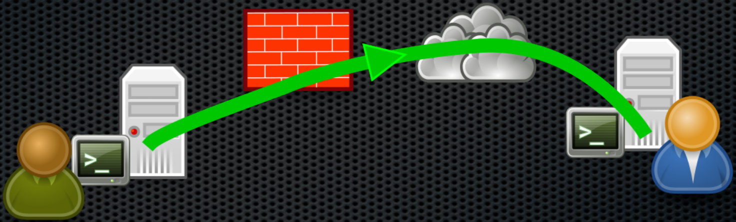
- Local TCP Port Forwarding: give remote acces to local port
- Remote TCP Port Forwarding: get access to remote ports

Local TCP Port Forwarding



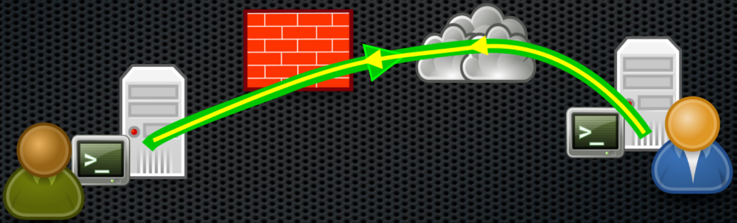
Icons from http://www.opensecurityarchitecture.org/cms/library/icon-library
and the Tango Icons project

Local TCP Port Forwarding



Icons from http://www.opensecurityarchitecture.org/cms/library/icon-library
and the Tango Icons project

Local TCP Port Forwarding



Icons from http://www.opensecurityarchitecture.org/cms/library/icon-library
and the Tango Icons project

Local TCP Tunnel example

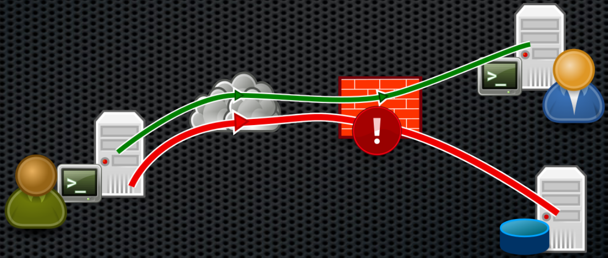
- User A is natted behind a firewall
- He wants to give User B access to local SSH daemon

```
userA@hostA> ssh —NR 22222:localhost:22 userA@hostB
```

```
userB@hostB> ssh —p 22222 localhost
```

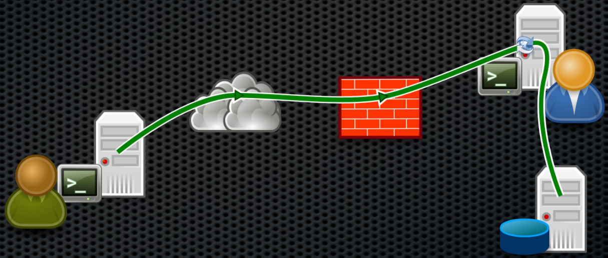
-N is for No Shell

Remote TCP Port Forwarding



Icons from http://www.opensecurityarchitecture.org/cms/library/icon-library
and the Tango Icons project

Remote TCP Port Forwarding



Icons from http://www.opensecurityarchitecture.org/cms/library/icon-library
and the Tango Icons project

Remote Port Forwarding example

- User A is behind a firewall that blocks VNC port
- He wants to access User B local VNC daemon

```
userA@hostA> ssh -NL 5900:localhost:5900 userA@hostB
userA@hostA> vncviewer localhost
```

SOCKS Proxy

- "Dynamic" port forwarding
- Enable UDP, TCP, …
- Creates a SOCKS5 proxy

```
userA@hostA> ssh –ND 9500 userA@hostB
userA@hostA> proxychains wget http://example.com
```

Tools

ssh-agent

- Stores your private key in memory
- eval $(ssh-agent)
- ssh-add; ssh-add -t 1h foo.key
- ssh-add -x (lock)
- ssh-add -X (unlock)
- Part of OpenSSH

screen

- Keep session accross ssh connection
- Have multiple shell `windows'
- Run long command and keep them running
- screen (launch new session)
- Ctrl+a d (detach)
- screen -dx (detach and reattach)
- ssh host -t screen -dx
- Alternative: tmux

reptyr

- Attach a long running process to the current terminal
- Idea: launch a screen and rattach another process inside
- Useful when you forgot to launch your screen before
- reptyr -p PID

vim

- Edit files remotely with scp
- vim scp://web//etc/hosts

Conclusion

Conclusion

- SSH is still part of modern infrastructures
- It should be part of what you automate/control
- Lots of other projects rely on it
- You can harden it in a lot of ways
- There is a lot of things to discover!

Homework

- SSH certificate authority
- command= permitopen=
- Match blocks
- sshfs
- ...

Any Question?

Contact

Julien Pivotto
julien@inuits.eu
@roidelapluie

inuits
https://inuits.eu
info@inuits.eu
+32 473 441 636