SSH agent: what is it and how to use it securely

Ondřej Caletka

29th May 2020

Public key client authentication

1. client holds a public/private key pair
2. client offers public key to the server
3. server asks client to sign a message
4. client performs the signature
5. client is authenticated

RIPE NCC

How to start with public key authentication

- generate a keypair using `ssh-keygen`
- **protect the private key** with a strong password
- choose the right algorithm
  - DSA – 1024bit, legacy, unsupported since OpenSSH 7.0 (2015)
  - RSA – variable length, secure default 2048 bits
  - ECDSA – 256, 384, or 521 bits, supported since OpenSSH 5.7 (2011)
  - Ed25519 – 256 bits, since OpenSSH 6.4 (2014)
  - ECDSA-SK, Ed25519-SK - Fido U2F, supported since OpenSSH 8.2 (2020)

RIPE NCC

SSH agent

- *optional* keychain for SSH keys
- allows you to **keep decrypted private keys** in memory
- communicate using a **unix-domain socket** (`$SSH_AUTH_SOCK`)
- **does not support** exporting of private keys
- usually run with the desktop environment
- ad-hoc run with `$ ssh-agent $SHELL`
- load/remove keys using `$ ssh-add`

RIPE NCC

SSH agent forwarding

- the agent's socket can be tunnelled to a server using `$ ssh -A` or `ForwardAgent yes` configuration option
- allows using keys loaded in local computer on the remote server

Security risk of forwarding SSH agent

Anybody with access to the agent's socket **can access the agent** – `root` user of the remote machine, for instance. They cannot obtain your private keys but they are **free to authenticate using them**.

RIPE NCC

SSH agent forwarding use cases

Jumphost access – not recommended

- `$ ssh -A <bastion> ssh <private-server>`
- hop-by-hop instead of end-to-end security
- `ProxyJump` is much better alternative

Work on remote workstation – valid reason

- copying files from server A to server B
- accessing private Git repositories from a remote server

RIPE NCC

Hardening SSH agent forwarding

Apply at least one of these:

- **do not enable universally**, but only to servers you *absolutely* trust
- load keys to the agent using `ssh-add -c` to get prompted before each key usage
- limit the lifetime of the key in the agent `ssh-add -c -t8h`

Getting key usage confirmation working on macOS

```
$ brew tap theseal/ssh-askpass
$ brew install ssh-askpass
```

RIPE NCC

Try it yourself

```
$ ssh whoami.filippo.io

                    ***** WARNING ***** WARNING *****

      You have SSH agent forwarding turned (universally?) on. That
     is a VERY BAD idea. For example right now I have access to your
     agent and I can use your keys however I want as long as you are
    connected. I'm a good guy and I won't do anything, but ANY SERVER
     YOU LOG IN TO AND ANYONE WITH ROOT ON THOSE SERVERS CAN LOGIN AS
                                YOU ANYWHERE.

                     Read more:  http://git.io/vO2A6

  +-------------------------------------------------------------------------+
  |                                                                         |
  |              _o/ Hello Ondřej Caletka!                                  |
  |                                                                         |
  |                                                                         |
  |  Did you know that ssh sends all your public keys to any server        |
  |  it tries to authenticate to?                                           |
  |                                                                         |
  |  That's how we know you are @oskar456 on GitHub!
```

RIPE NCC

Try it yourself

```
|  That's how we know you are @oskar456 on GitHub!                    |
|                                                                      |
|  Ah, maybe what you didn't know is that GitHub publishes all users'  |
|  ssh public keys and Ben (benjojo.co.uk) grabbed them all.           |
|                                                                      |
|  That's pretty handy at times :) for example your key is at          |
|  https://github.com/oskar456.keys                                    |
|                                                                      |
|                                                                      |
|  P.S. This whole thingy is Open Source! (And written in Go!)         |
|  https://github.com/FiloSottile/whoami.filippo.io                    |
|                                                                      |
|  -- @FiloSottile (https://twitter.com/FiloSottile)                   |
|                                                                      |
+----------------------------------------------------------------------+
```

```
$ host whoami.filippo.io
whoami.filippo.io is an alias for nyc.filippo.io.
nyc.filippo.io is an alias for p50212.probes.atlas.ripe.net.
p50212.probes.atlas.ripe.net has address 96.246.192.163
```

RIPE NCC

Alternative SSH agent implementations

- GnuPG agent
  - can use PGP keys/smartcards as SSH keys
  - cannot load foreign keys
  - certificates are not supported
- SeKey - macOS Secure Enclave SSH agent
  - uses the Apple security chip of touchbar MacBooks
  - keys generated inside the Secure Enclave
  - touch ID required for each key usage
  - keys cannot be imported
  - certificates are not supported
- Guardian Agent
  - a framework to securely forward SSH keys even to untrusted servers
  - asks the user verbosely on every key usage

RIPE NCC

Thank you!

Ondřej Caletka
Ondrej.Caletka@ripe.net
https://**Ondřej.Caletka.cz**

RIPE NCC
RIPE NETWORK COORDINATION CENTRE

The slides are already published on my website.

One more thing: mosh – Mobile Shell

- a new protocol for **better terminal sessions** on a lossy/laggy line
- uses custom UDP-based protocol and **inteligent local echo**
- reuses SSH for authentication and session establishment
- session survives extended interruptions
- client can **change IP address** (but not address family)
- synchronises only display state between terminals
- only UTF-8 terminal, ignoring *dangerous* control sequences
- cannot tunnel anything

RIPE NCC