Secure Delegation

Dima Kogan, Henri Stern, Ashley Tolbert, David Mazières, Keith Winstein

Stanford University
Delegation today

```
[alice@laptop]$ ssh aws
[alice@aws]$ git clone \ git@github.com:alice/private-repo
    Authentication required...
```
Inadequate Solutions

• Unrestricted credentials (e.g., copying private keys)
• Finer-grained credentials
• Tunneling
OpenSSH Authentication Passphrase Request

Allow use of key /home/alice/.ssh/id_rsa?
Key fingerprint SHA256:qwLY8d0kKayuxPNR7HDa8M43elZ65U
TKJyZvMICYQ.

Cancel          OK
Delegation today

The user cannot verify
- the identity of the **delegate**, 
- the **command** the delegate will run, or 
- the **server** it will run the command on

[alice@aws]$ ssh alice-home rm -rf /

Delegation today = signing blank checks

Allow use of key /home/alice/.ssh/id_rsa?
Key fingerprint SHA256:qwLY8d0kKayuxPNR7HDa8M43e1Z65l/TKJyzVvMICYQ.

Cancel  OK

=  

Pay to the order of DIMA

For  $1000
Instead of:

We should have:

Thesis: secure protocols should support Secure Delegation: Restrict who, can do what, to whom
Guardian Agent

• **Secure Delegation** scheme for SSH

• Consists of:
  • On the user’s machine – an authentication agent
  • On the delegate – a custom SSH client
  • On the server – **no changes**

Available at [https://github.com/StanfordSNR/guardian-agent](https://github.com/StanfordSNR/guardian-agent)
Guardian Agent

1. Delegate connects to agent and **issues a connection request**
2. Agent connects to the server and **authenticates** using the private key
3. Agent **hands off** the established connection to the delegate

---

**Local Machine** (running Guardian Agent)

**Delegate** (running our SSH client)

**Server** (unmodified OpenSSH)

---

1. Please run `git-fetch-pack` on `git@github.com`

2. Establish SSH connection
   - Authenticate using private key
   - Issue command
   - Issue `no-more-sessions`

3. Handoff SSH connection

---

**Key Re-Exchange**

Resume SSH connection
Transport layer

• **Constraints:**
  • Agent might not have direct connectivity with the server
  • Handoff must be transparent to server

• **Solution:**
  • Client relays a TCP connection between the agent and the server

```
[bob@aws]$ git clone \
git@github.com:bob/private-rep
```
TLS has an analogous issue

• No easy way for a publisher to **securely** delegate HTTPS resources to a CDN

• Recent workaround – KeylessTLS [SS15]
  • Similarly to ssh-agent, essentially exposes a signing oracle
Secure delegation for apps, payments, the web

$ sudo apt-get update
[sudo] password for alice:

Restricted who, can do what, to whom
Conclusion

• There are more parties than just the client and the server
  • Semi-trusted hosts, firewalls, CDNs,…
• Existing protocols mostly focus on the client and the server
• The void has been filled with insecure workarounds
• Protocols should support Secure Delegation
  • User should control who, can do what, to whom

https://github.com/StanfordSNR/guardian-agent