

Using public key authentication with SecSH

Michael Richardson
<mcr@xelerance.com>

Xelerance Corporation
<http://www.xelerance.com/>

me:

<http://www.sandelman.ca/mcr/>

talk: <http://www.xelerance.com/talks/ssh2006/>



Overview of talk

- Introduction to speaker
- Introduction to problem to be solved
- Public keys and certificates
- Demo of doing it
- What usually goes wrong



Introduction to speaker

- Michael Richardson <mcr@xelerance.com>
- Long time Unix, BSD, Linux user.
- I write network protocol stuff: IPsec, radius, telnet, ssh, etc.
- history of security stuff: milkyway.com (firewalls), solidum.com (L2-L7 policy-based classification), SSH (IPsec-Express), other stuff.
- VP R&D at Xelerance Corporation
- Xelerance.com is providing 3rd level defect support for Openswan.
- AKA Sandelman Software Works, also tcpdump.org maintainer.
- This talk at <http://www.xelerance.com/talks/ssh2006/>



Introduction to problem to be solved

- **Goal:** login to a remote system without using a password --- something you have as well as something you know.
- Ancillary: be able to have single-sign-on
- **WHY:** passwords can be stolen, guessed, or **brute forced**.
- This is happening regularly now that SSH is commonly used.



Why this talk?

- OCLUG's tux.oclug.on.ca server uses PK-only logins
- we need more tech volunteers
- getting ssh working seems a difficult hurdle to overcome.



Public keys and certificates

- public keys systems like RSA and DSA. See wikipedia.
- certificate systems **use** public keys, to provide a **third party** trust anchor.
 - certificates are largely the reason why public key systems have been slow to be adopted.
- PKI = Public Key Infrastructure



Secure Shell

- SecSH – (RFC4251) uses **public keys** but does not mandate the use of an **Infrastructure**. (Does not preclude one though!)
- products include
 - ssh.com's ssh1
 - openbsd's openssh (derivative of ssh1)
 - ssh.com's ssh2
 - putty
 - kermit
 - dropbear



Basic overview

- ssh-keygen to make the keys.
- copy&paste or email to distribute them (pgp helps!)
- ssh-keygen to convert the keys
- .authorized_keys to bind them
- ssh-agent to control them



DEMO



SSH-KEYGEN

- puttygen on windows



DISTRIBUTE KEYS



KEY FORMATS



.ssh/authorized_keys



ssh-agent – single sign on

- run `eval `ssh-agent`` in parent of window manager
 - all modern gnome/kde do this.
 - you can do this in `xsession` if you need to.
- `ssh-add`
- `ssh-add -l`
- `ssh -X` and `ssh -A` options



What usually goes wrong

- home dir is writeable by group
- .ssh dir is writeable by group
- .ssh/authorized_keys file is writeable by group
- some dir is not searchable (openssh runs as non-root!)
- NFS is involved and above



Some other ways to mitigate password attacks

- ipt_recent
- hosts.allow
- log scanning program(s) + iptables.



Questions?

