SSH Key Management: Status and Best Practice

Tatu Ylonen

SSH Communications Security

2012-11-08 (updated)

ssh®
Securing the path to your information assets

Where Is SSH Used?

- All Unix/Linux servers, most routers, many other devices (telecom equipment, xDSL, printers, aircraft entertainment systems, …)
- Underneath most virtualization platforms
- For managing >50% world's web server hosts (Apache on Linux/BSD; virtualization infrastructure)
- Zillions of automated scripts and management tools, file transfer solutions, security tools, backup tools, etc. in organizations

Findings

- SSH user keys mostly used for automated access
- Major bank A has over 1,000,000 SSH user keys on 10k servers (out of 40k), 10% grant privileged access, no knowledge of who can access what, no termination of key-based access, no key rotation
- Major bank B has so far found 400,000 SSH user keys
- Major bank C still has no idea how many keys they have on their 100,000 servers and no approval process or control of new key setups of any kind, never remove or rotate keys
- Major technology company D has >10000 hosts sharing the same host key (-> no man-in-the-middle protection), user keys in NFS
- Variation of the story repeats in most large organizations

Findings (cont'd)

- Many companies have SSH keys (private and authorized) on NFS volumes
- In most companies any admin can install SSH key to leave unaudited permanent backdoor to critical functional account
- Most companies do not know which keys are still in use and a large fraction of keys aren't
- Most companies have not documented which application needs each key
- Almost nobody regularly changes SSH keys
- Most keys do not have a "forced command"
- SSH user keys used for systems management, backups, disaster recovery, file transfers, etc

Findings (cont'd)

- SSH host keys sometimes shared, and many embedded boxes using a shared host key from the firmware image or creating key without having accumulated enough randomness (e.g., keys with one prime shared on many devices)
- The problems are not specific to SSH user keys but mostly also apply to certificate-based, Kerberos-based, and host based automated trust relationships

Implications

- A virus or cyberweapon using SSH keys can spread to nearly all servers quickly it gets to one server; combined with other vectors and destruction code causes existential threat to organizations and society
- Major risks from rogue administrators/staff (copying private keys, adding permanent key-based backdoors that bypass privileged access management systems)
- Major SOX/PCI/FISMA/HIPAA/NERC compliance issue: currently no proper control of who can access what servers and no proper termination of key-based access
- Inability to audit who can access what and that access is properly terminated

How Realistic Is the Virus Threat?

- The Morris Worm used automated trust relations (.rhosts based) as attack vector already in 1988 (and took down much of the Internet of that time)
- I have heard several stories on SSH user keys being used for actual attacks from forensics experts and from penetration testing consultants
- It would only take a few hundred lines of code/ script to write code that uses SSH keys to spread on an internal network
- My opinion is that it WILL happen, likely SOON

What Really Is the Problem?

- It is not a flaw of the SSH protocol
- It is not a flaw of existing implementations
- The problem is lack of awareness of the issue, lack of guidelines
- It is basically sloppy system administration, but you cannot really blame the sysadmins, because:
  - Each admin only sees a small part of the problem
  - IT operations and IT security management is scarcely aware of the problem
  - IT security auditors lack knowledge and tools to check for the problems and understand its scope

Elements for SSH Access Management Best Practice

- MUST be able to audit who can access what
- MUST ensure proper termination of key-based access when employees leave and change roles
- SHOULD use “forced command” for all keys to limit virus spread potential
- SHOULD be able to monitor key-based privileged access (including session logging)
- SHOULD enforce approval process for new key setups (including keys in root-owned location)
- SHOULD remove keys that are no longer needed
- SHOULD regularly change SSH user keys
- MUST also change private key when generating new host/user certificate
- SHOULD use distinct host key on each device and ensure proper randomness for their generation
- SHOULD regularly change SSH host keys/host certificates or use Kerberos for them
- MUST also similarly manage host-based and Kerberos-based trust relationships (details TBD)

Mostly Not a Technical Issue

- Need guidelines on how SSH keys should be managed
- Organizations need to take SSH keys into account in security policies and establish proper processes
- IT security auditors should take keys into account in audit checklists and checks should be implemented into audit and penetration testing tools
- Should liaise with regulators to ensure problem addressed without a disaster happens; security regulations should address automated SSH-based access more explicitly as SSH is a key protocol used for managing lots and lots of Unix/Linux servers and embedded devices (and also many Windows servers, especially for file transfers)

Next Steps

- Currently no good Best Practice document on how SSH keys (and other trust relationships) should be managed
- Could proceed with individual draft (goal to have it approved as BCP) or Working Group chartered to develop a BCP
- The problem appears real and high risk; would like to proceed quickly to BCP
- Proposing to proceed the Individual Draft to BCP route
  - Document best practice on how SSH user keys and other trust relationships granting automated access to servers should be managed
  - Document best practice on how SSH host keys should be managed
  - Document best practice on how automated SSH access and SSH keys should be audited.
- Comments?