Risks of Unmanaged Encrypted Environments to Critical National Infrastructures

Tatu Ylönen

Founder and CEO

SSH Communications Security

www.ssh.com

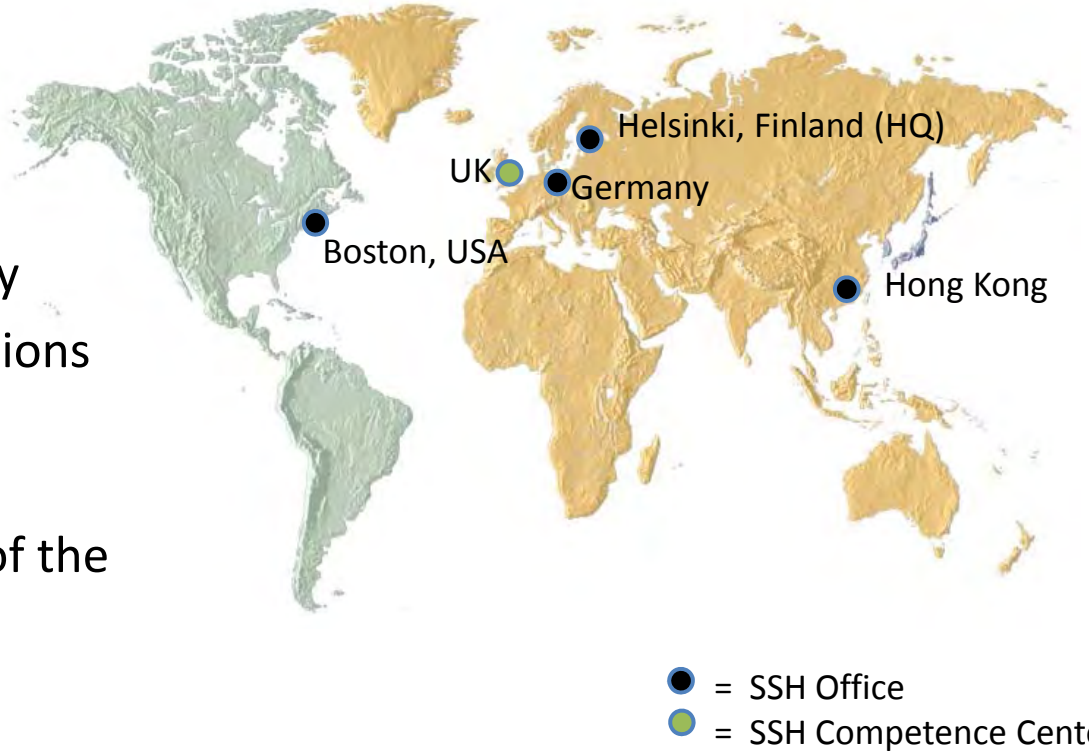
Who am I?

ssh® is a registered trademark of SSH Communications Security

- Developed and published the original Secure Shell as free software in 1995
- Founded SSH Communications Security Corp in 1995
- CEO and controlling shareholder for SSH Communications Security
- Long-term entrepreneur
- Deeply involved in development of solutions for large SSH environments for both commercial SSH products and OpenSSH

SSH Communications Security

- Founded in 1995
- Listed on NASDAQ OMX Helsinki (SSH1V)
- 50+ patents in various countries in cybersecurity
- Leading provider of solutions based on SSH protocol
- Over 3,000 customers worldwide - including 7 of the Fortune 10 and 40% of Fortune 500



Birth of Secure Shell (SSH)

- I was a researcher at Helsinki University of Technology, when the university network was hacked in early 1995, and passwords were stolen with a password-sniffing attack
- To prevent such attacks, I developed the **Secure Shell** protocol suite to replace the unsecured telnet, ftp, rlogin and rcp tools
- The first version of the SSH protocol was released to the Internet community in July 1995
- By the end of 1995, SSH had 20,000 users in 50 countries
- By year 2000, SSH had estimated 2 million users
- Today, it is widely used globally by enterprises, government agencies, research institutions and universities to secure sensitive network traffic both within and between networks.

Where is Secure Shell Used Today?

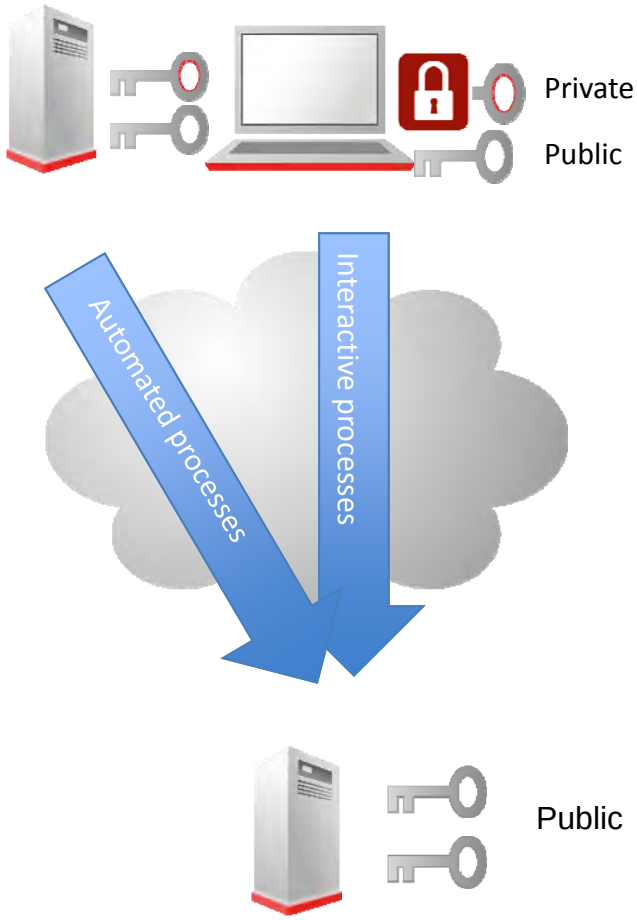
- SSH protocol is widely used
  - by system administrators
  - in automated machine-to-machine processes
- SSH protocol is used
  - on every Unix/Linux computer
  - in most cloud computing environments
  - in more than half of the world's web sites
  - in most xDSL modems, routers, telecom exchanges and other network equipment.
- SSH protocol has been in use since 1995 and is used widely today
- SSH protocol is still secure
- But the way organizations are handling keys for user authentication is not

Secure Shell Authentication Keys

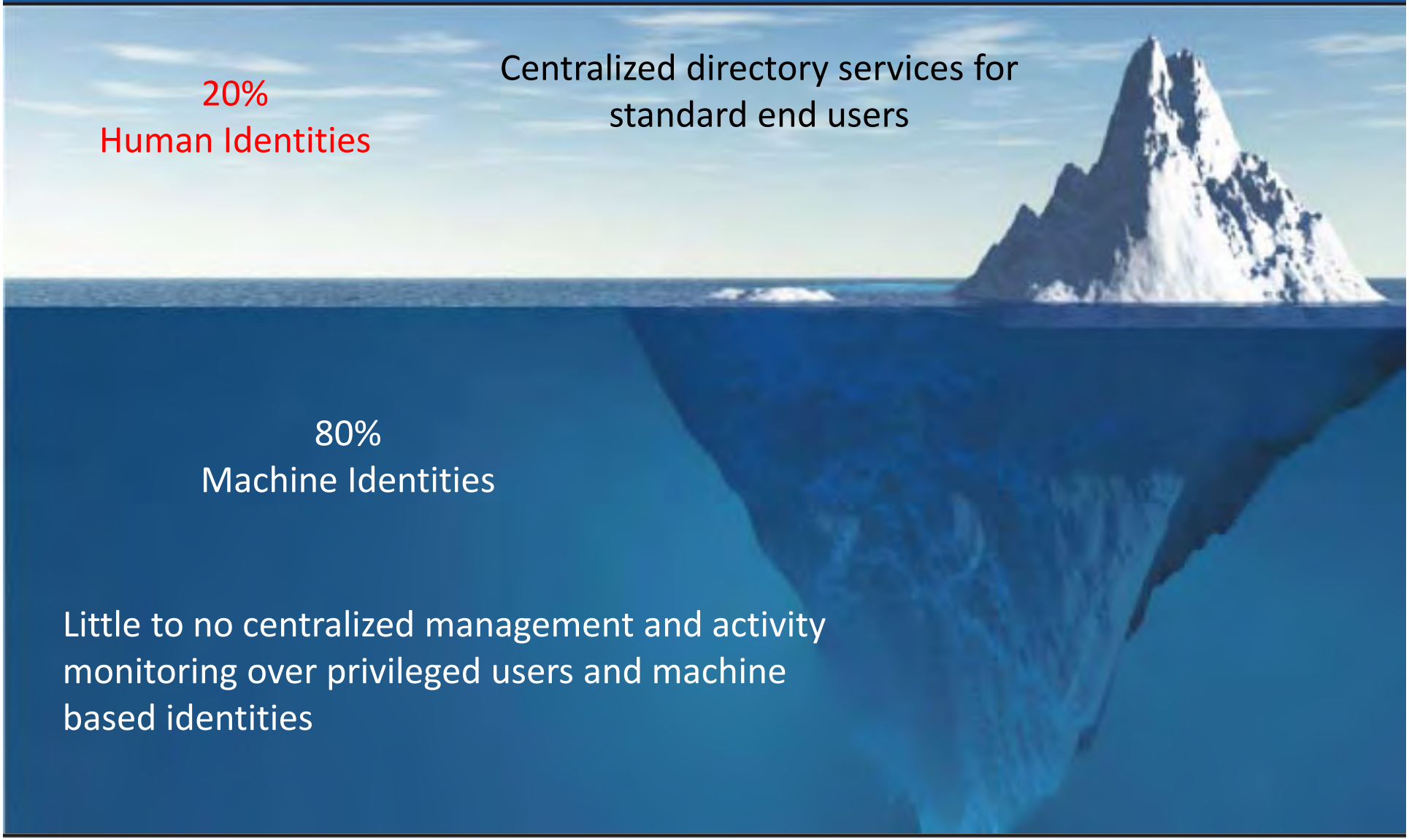
- A Key pair, consisting of a private and public key
  - Private Key is an “Identity Key”. It validates identity of the person or process requesting login.
  - Public Key is an “Authorized Key”. Establishes what level of access is granted to the holder(s) of the Private/Identity Key.

- Keys are used for automated processes
  - Enables authentication for scheduled and automated file transfers and other tasks where user interaction is not possible

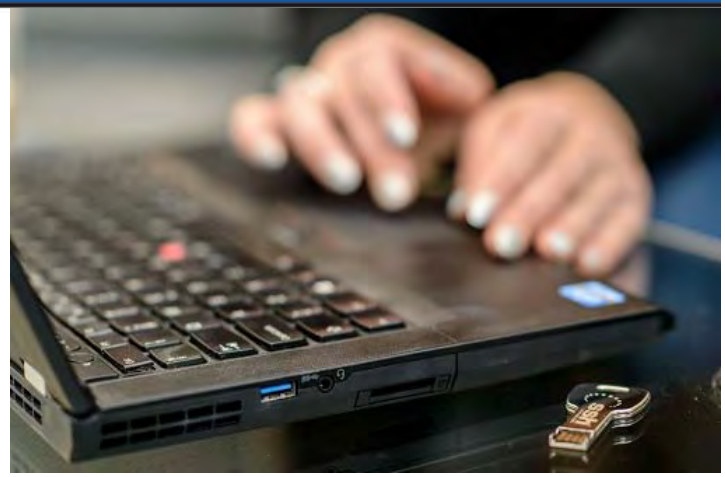
- Keys are used for interactive connections
  - Private key can be protected with a passphrase



Explosion of Machine Identities

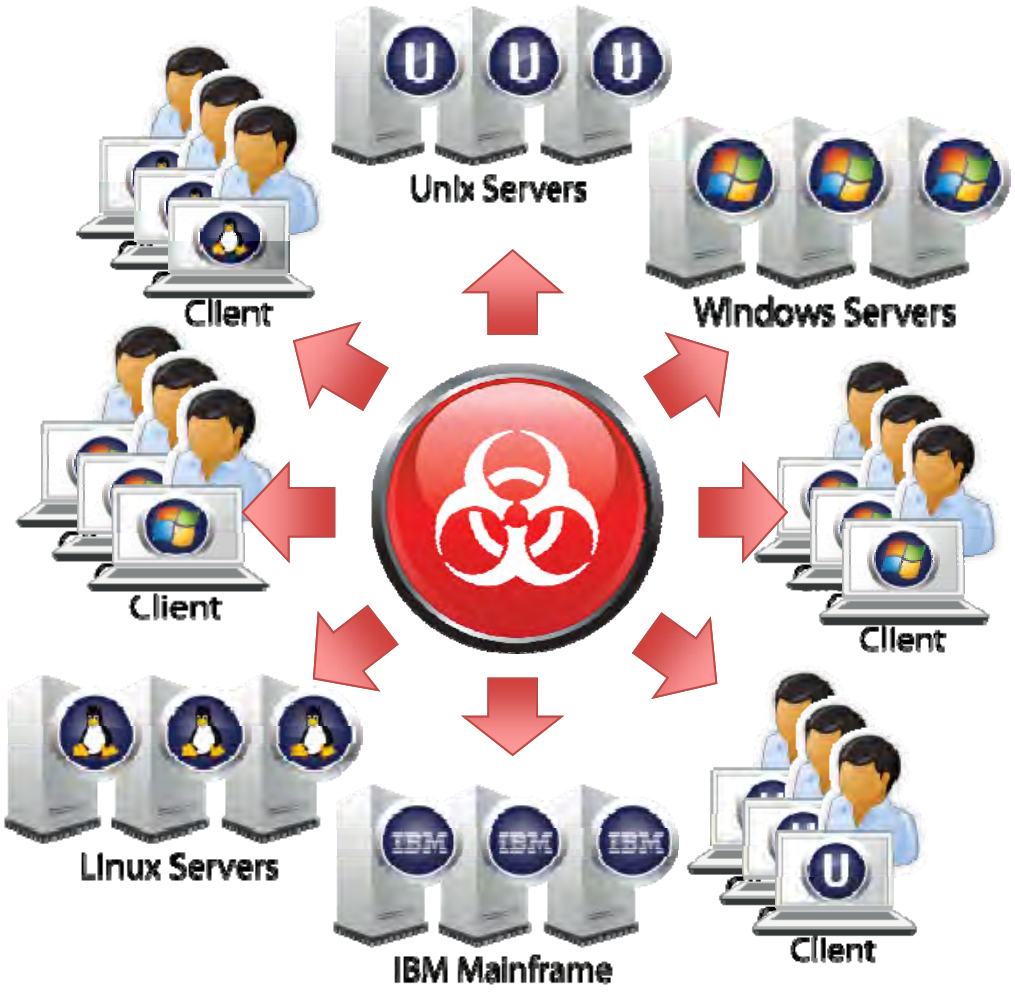


Vulnerabilities due to Mismanaged Authorized Keys



- Not knowing who can access what
- Not knowing who is accessing what
- Bypassing privileged access management systems
- Adding an authorized key to hide a backdoor
- Unintended non-file-transfer access by business partner (missing command restriction)
- Undocumented Cross-Environment Connections:
  - Development/Testing -> Production, low impact->high impact, non-PCI->PCI
- Lacking effective termination of access when employee leaves
- Quick attack spread once inside perimeter (malware, APT) – SERIOUS!
- Many organizations cannot practically change keys even after breach
- Lack of compliance with government/industry regulations (e.g., PCI)

Cyber Weapon or Advanced Persistent Threat



- Most organizations have on the average 8 to 100+ SSH keys configured granting access to each Unix/Linux server
- These keys often grant high level administrative access
- The mesh of key-based access is so dense that it is highly likely that an attack can spread to nearly all servers in an organization
- Risks increase substantially if the APT also utilizes other attack vectors to escalate privileges to “root” (high-level administrator) after penetrating a server

Real World Example

- A Major Financial Institution
  - Over 10,000 servers on their network
  - 1.5 million keys identified
  - 10% or 150,000 keys HAD ROOT ACCESS
  - Failed Monetary Authority of Singapore and SOX Audits
  - Now working with SSH on key remediation project

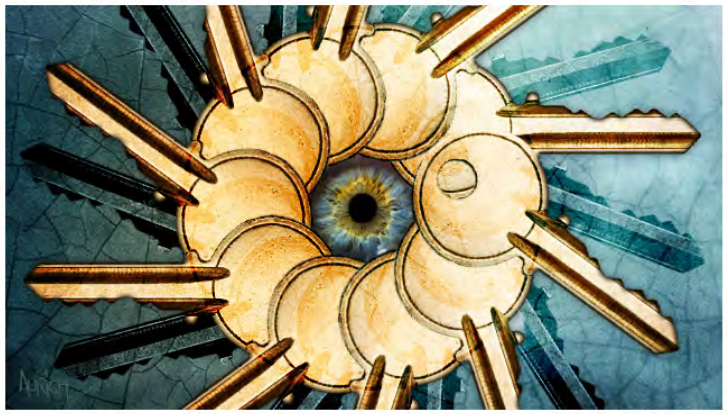
Real World Breaches and Exposures

Former Hostgator employee arrested, charged with rooting 2,700 servers

Prosecutors: Backdoor and digital key gave him near unfettered access.

by Dan Goodin - Apr 19 2013, 7:51pm FLEDT

BLACK HAT INTERNET CRIME 69



theguardian

News | Sport | Comment | Culture | Business | Money | Life & style

News › Technology › Software

GitHub users warned over security risk

Search tool on programming site turns up SSH keys, which could allow attackers to hack sites or alter programs silently

COMPUTERWORLD

Topics | News | In Depth | Reviews | Blogs | Opinion

Security | Application Security | Cybercrime and Hacking | Cyberwarfare | Malware and Vulnerabilities | Mobile Security | Privacy

Home > Security

News

Hackers break into two FreeBSD Project servers using stolen SSH keys

Users who installed third-party software packages distributed by FreeBSD.org are advised to reinstall their machines

By Lucian Constantin

November 19, 2012 06:29 AM ET 3 Comments

EXPOSED ROOT SSH KEY WAS SHIPPING WITH EMERGENCY ALERT SYSTEM DEVICES

by Michael Mimoso Follow @mike_mimoso

July 8, 2013 , 5:18 pm

Standardization Initiatives

- **PCI DSS 3.0** requires addressing SSH key based access in many ways (e.g., scope includes any system having security impact on cardholder data) – Compliance Kit at http://pages.ssh.com/pci3.html
- **Monetary Authority of Singapore Technology Risk Management Guidelines** require managing SSH key based access (e.g. Development/Testing -> Production) – see whitepaper at http://pages.ssh.com/mas.html
- **US Sarbanes-Oxley (SOX)** requires controlling who can access financial data
- **US FISMA** and **NIST** SP 800-53 require managing SSH keys in numerous ways (NIST Interagency Report coming out very soon!)
- **US FERC/NERC CIP** rules require controlling who can configure critical infrastructure
- **US HIPAA** requires controlling who can access patient data
- **IETF** draft-ylonen-sshkeybcp-01.txt provides best practice guidelines (pending update!)
- Most regulatory standards are written technology-agnostic, and don't mention SSH by name – but keys provide system-level access

Compliance Roadmap

1. Establish controlled key provisioning process
2. Remediate existing legacy keys
3. Establish continuous monitoring and management of keys, for both automated and interactive access

1. Establish Controlled Key Provisioning Process

- Enforce approvals for all new key creation
  - Access provisioning should be on need basis and auditable
- Move authorized keys to root-owned location
  - Prevent easy bypass of approval process and arbitrary access delegation
- Automate provisioning of approved key requests
  - Cost savings, security, less errors
- Document purpose and owner of each authorized key
  - Cannot audit and terminate keys if purpose is not known; need to know which person to ask about key
- Enforce command restrictions on keys
  - Limit attack spread by malware / APT

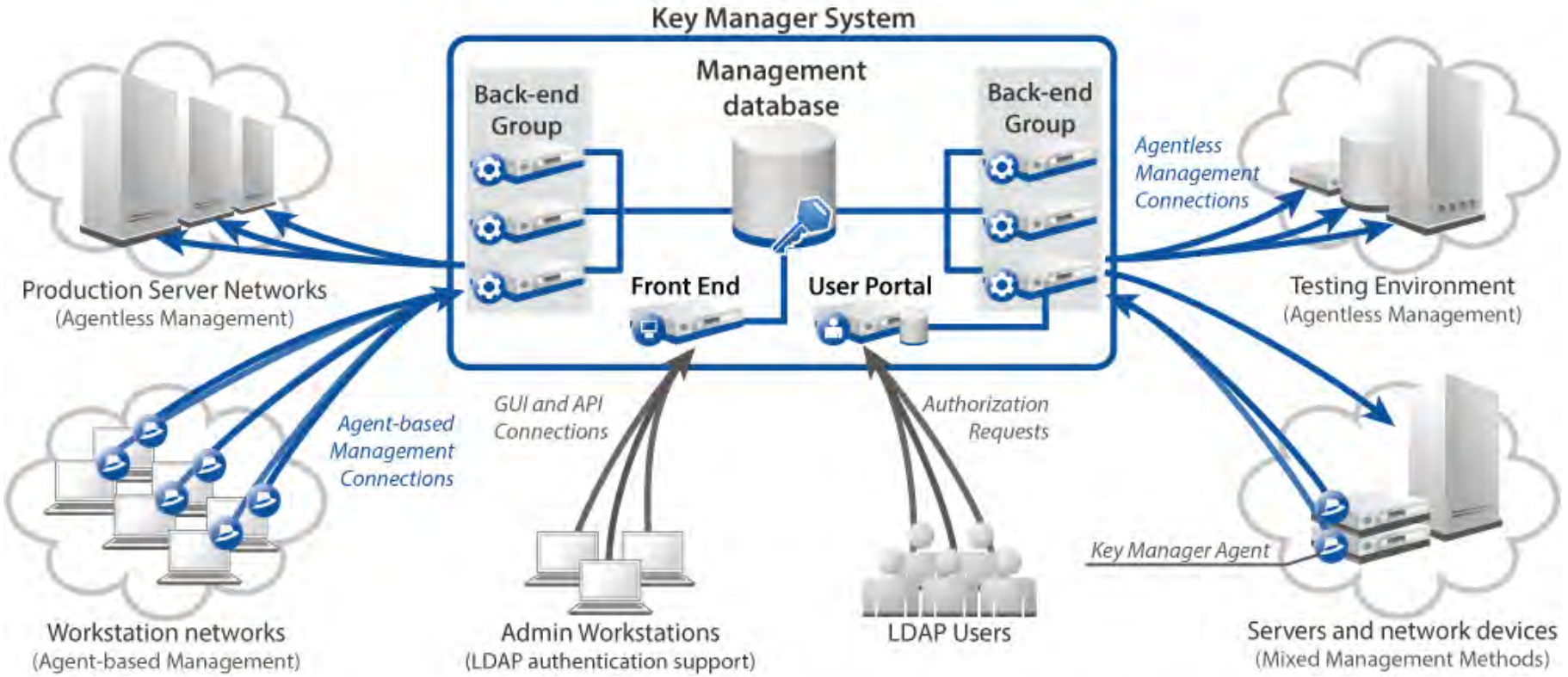
2. Remediate Existing Legacy Keys

- **Discover keys and monitor use**
  - Monitor key usage to determine which keys are actually being used and from where
- **Reduce number of keys**
  - Eliminate keys that are never used
  - Eliminate policy or boundary violating keys (e.g. Development -> Production)
  - Must always have a back-off option in case a key is critical for business!
- **Identify business purpose and owner for remaining keys**
  - Collect approvals from application teams
  - Record purpose, owner, and approval for compliance and maintenance
  - Eliminate keys for which no business purpose can be found
- **Add restrictions to remaining keys**
  - Prevent malware or Advance Persistent Threat (APT) spread

3. Monitor and Manage Keys Continuously

- **Monitor syslog for key access activity and scan hosts to find keys**
  - Immediate alerts on unauthorized keys and unauthorized key-based access
- **Implement periodic key rotation**
  - Ensures eventual termination of access by copied keys
    - Very important after breach!
- **Implement privileged access auditing also for key-based access**
  - Keys intended for automation may be used by people!
  - Prevent bypass of privileged access auditing
  - Detect attacks using keys early

Solution Approach: Centralized SSH Access Management



Centralized database:
-Repository of all configuration and key information

Backend:
-Host connectivity
- Job execution

Management Agent:
- Connectivity in Agent-based deployment

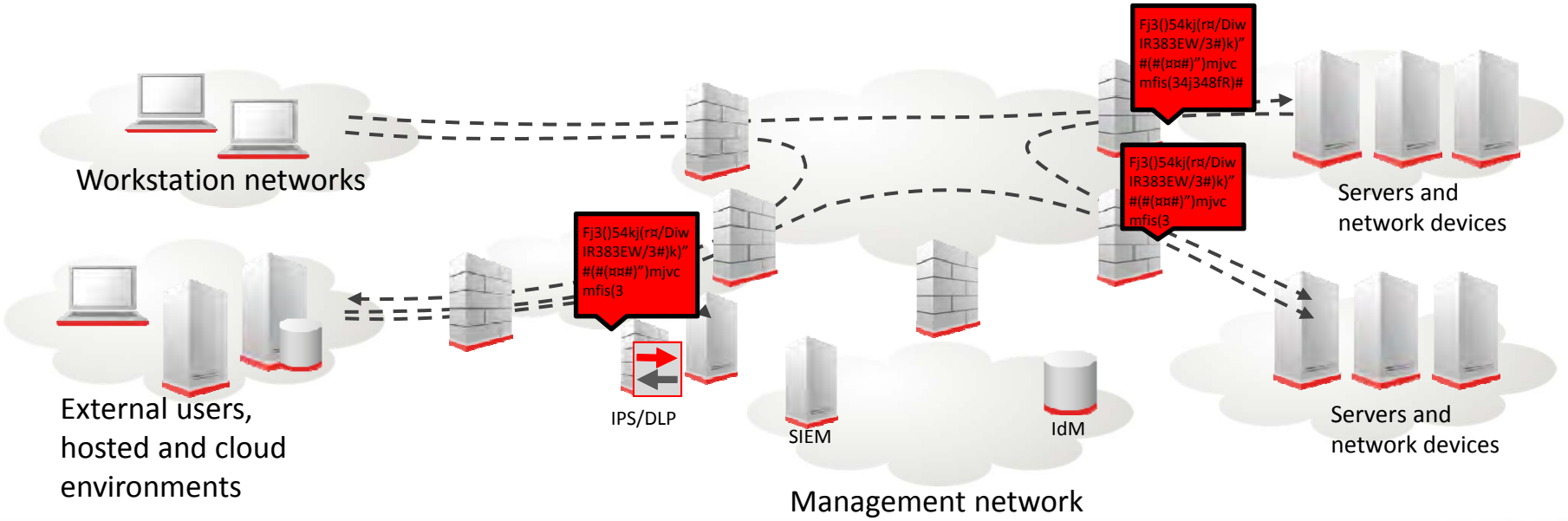
Frontend:
-Web-based graphical user Interface for configuration and reporting
- API integration point for configuration and reporting

UserPortal:
-Web-based portal for end users, application owners etc.
-External key import for external keys/ users
-Access request/removal/restriction enforcement signoffs and approval processes

Security Paradox: Security vs. Visibility

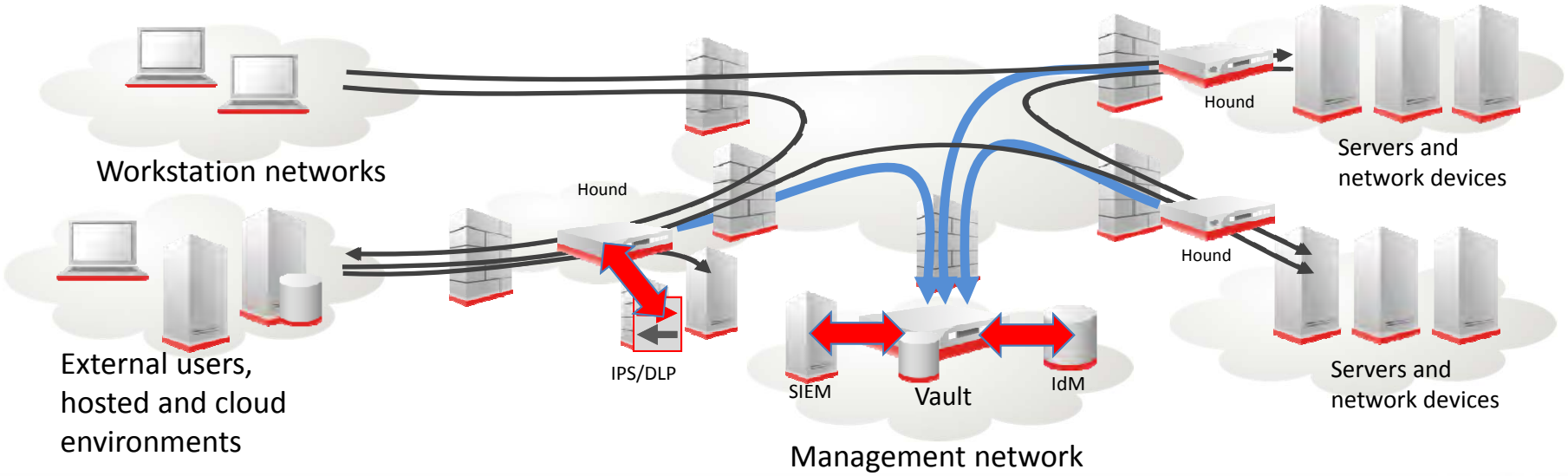
- Encrypted remote system access and data transfers of an enterprise:
  - Internal to Internal
  - Internal to External
  - External to Internal
- Encryption means NO visibility to the actual content of the traffic

- How to trace and audit user commands and activities?
- How to inspect and analyze incoming and outgoing data flows?



Solution Approach: Trusted and Transparent Audit Point

- Inline, agentless and invisible on-the-fly capture
- Centralized management, reporting and encrypted audit trail storage
- Audit point can be deployed as virtual or hardware appliance
- Audit and replay connections exactly as they happened
- Real-time content based indexing, searches and reporting
- Preventative channel and content controls
- Integration to DLP, IPS, SIEM
- Minimally invasive: No changes to user experience, processes or environment



Conclusion

- Most organizations with large Unix/Linux environments have a serious security and compliance problem with unmanaged authorized keys
- The scope and impact of the issue is not yet very widely understood
- Knowing who can access what systems and information is critical for information security – without it you don't have confidentiality, integrity or continuity
- It is about access, not cryptographic algorithms or sizes
- For more information:
  - http://tools.ietf.org/id/draft-ylonen-sshkeybcp-01.txt

Questions?

www.ssh.com

Tatu Ylönen

Founder and CEO

SSH Communications Security