Stronger Authentication Using Kerberos with Secure Shell (SSH)

Eric Raisters

Reflection Network Security
Technical Lead
WRQ, Inc.

HP WORLD 2003
Solutions and Technology Conference & Expo

Why Use Kerberos with SSH?

- Authentication in Secure Shell raises concerns
  - Username/password authentication sends known text across with each login, making it easier to crack
  - Certificates are difficult to manage in timely manner if implement user key authentication
- Provides the proven security of Kerberos authentication
- Maintains the simplicity and flexibility of Secure Shell

Kerberos authentication security

- Centralized user administration (Active Directory or UNIX)
- No passwords transmitted over the wire
- Near single sign-on in many heterogeneous networks
- Single sign-on from PCs with Windows Active directory
- Kerberos credentials forwarding – less hassles than administering and distributing user keys
- IETF draft standard for GSSAPI authentication with SSH

Secure Shell flexibility

- More (stronger and faster) cipher options
- Protocol independent with TCP port forwarding (X11)
- No modification of applications required
- Fewer ports available to attackers

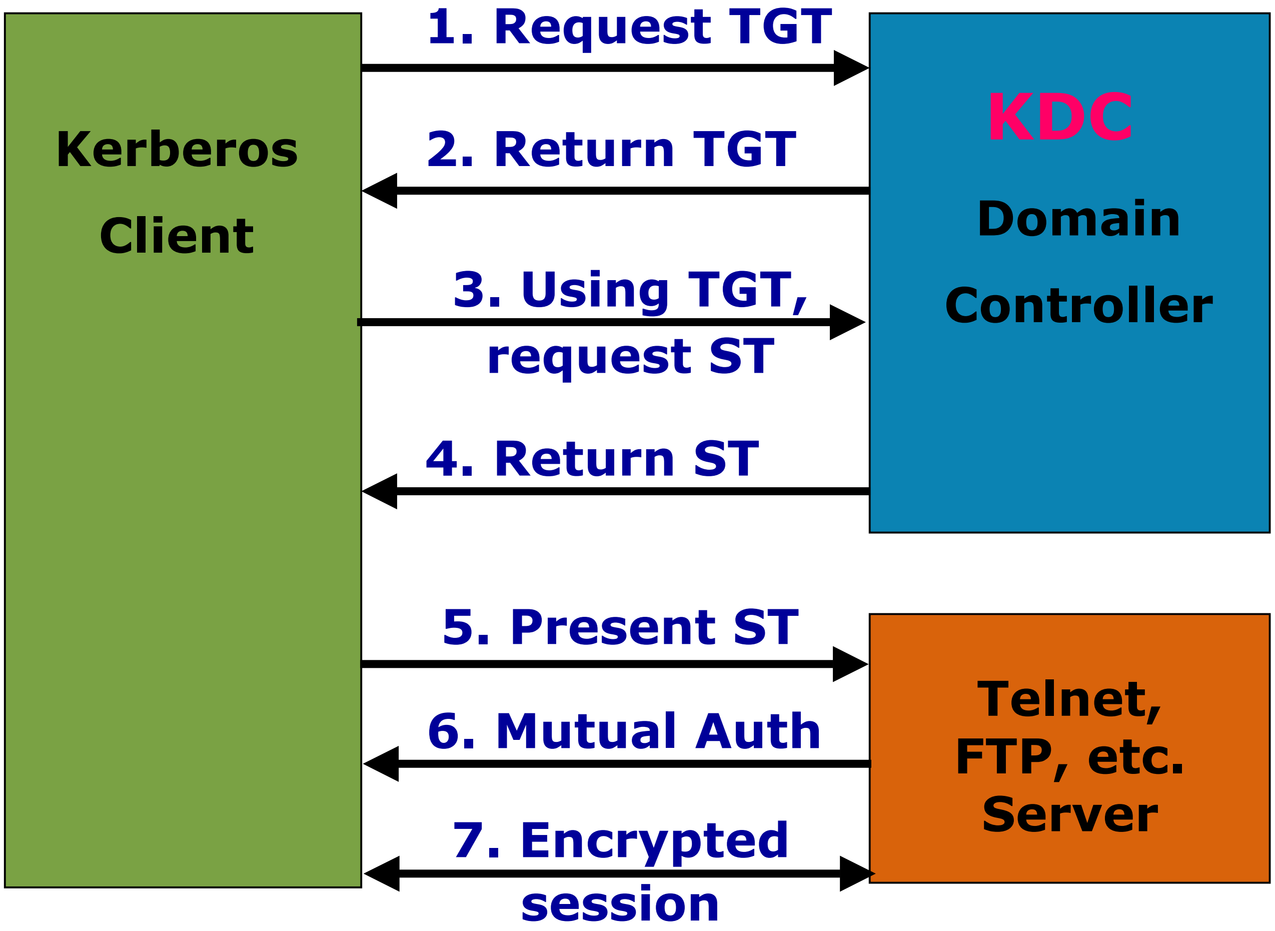
What is Kerberos

- Named for the mythical three-headed dog that guarded the entrance to Hades
- RFC standard protocol used for authentication, data integrity, and encryption
  - RFC 1510 – Kerberos authentication
  - RFC 2743 – Generic Security Services API (GSSAPI)
- Created at MIT in the early 1980s as project Athena
- Current open-standard version is 5.0
- Both commercial and open source versions available
- Implemented in Windows 2000 and XP
  - Active Directory servers use it for authenticating users

Kerberos - Features

- Secure authentication
  - Password never travels over the network
  - Memory-only credentials caches
- Data stream protections
  - Detection of data stream modification
  - 56-bit DES or 168-bit 3DES encryption

Kerberos Basics



Kerberos - Pluses

- Mature, open standard that’s never been broken
- Minimal administration and server overhead
- Programmatic access through GSSAPI
- Widely available for UNIX, Linux, Windows, OpenVMS, Unisys and IBM mainframes
- No patent or royalty encumbrances with all publicly available, standard algorithms

Kerberos - Minuses

- The Key Distribution Centers (KDCs) must be secured
- Prone to offline attacks on TGT; brute force attacks are now feasible on 56-bit keys
- Ciphers are “slow” – AES only now being added
- Significant cost of implementation
  - Requires applications be “kerberized”
  - Administrators require specialized training
  - Initial design and implementation may be difficult

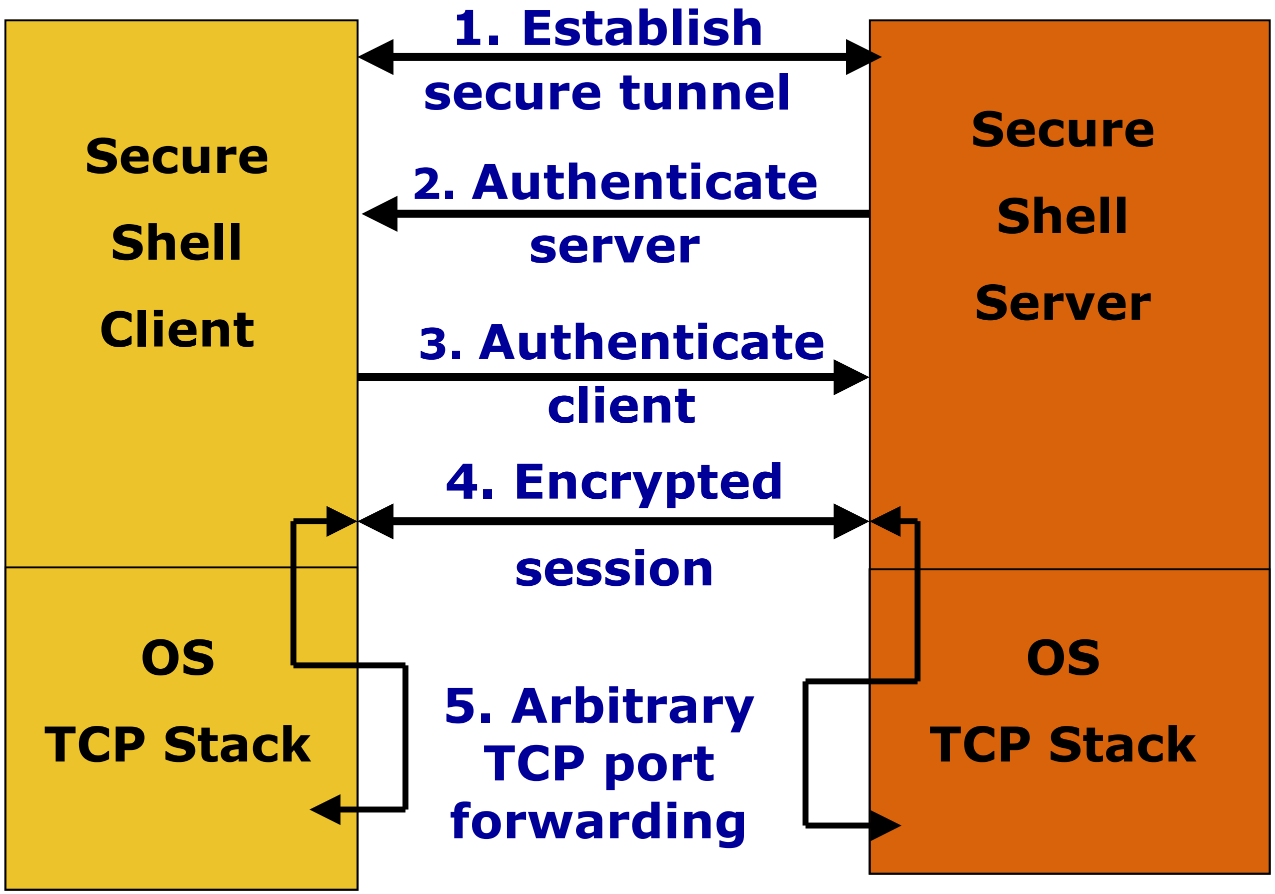
Secure Shell (SSH)

- Developed by grad student Tatu Ylönen of Finland in 1995
- First open source version released in 1999
- Both commercial and open source versions available
- SSH-1 (deprecated) and SSH-2 protocols
- Replaces Telnet, *rlogin*, *rsh*, and *rcp*

Secure Shell - Features

- Many more data encryption types: 56-bit DES, 168-bit 3DES, 128-bit Arcfour, 128-bit CAST, 128-bit Blowfish and AES algorithms up to 256-bits
- Secure forwarding of arbitrary TCP connections, including X-11 protocol
- FTP replacement *sftp* in SSH-2
- Secure file copy *scp* in SSH-1
- OpenSSL libraries used for SSH-1 compatibility and user key authentication

Secure Shell Basics



Secure Shell Client

OS TCP Stack

1. Establish secure tunnel

2. Authenticate server

3. Authenticate client

4. Encrypted session

5. Arbitrary TCP port forwarding

Secure Shell Server

OS TCP Stack

Secure Shell - Pluses

- IETF draft, open-source standard
- Many commercial implementations also available
- Only one firewall port need be opened (22)
- No patent or royalty encumbrances in OpenSSH
- Protocol-independent
- Available on UNIX, Linux, OpenVMS, Windows

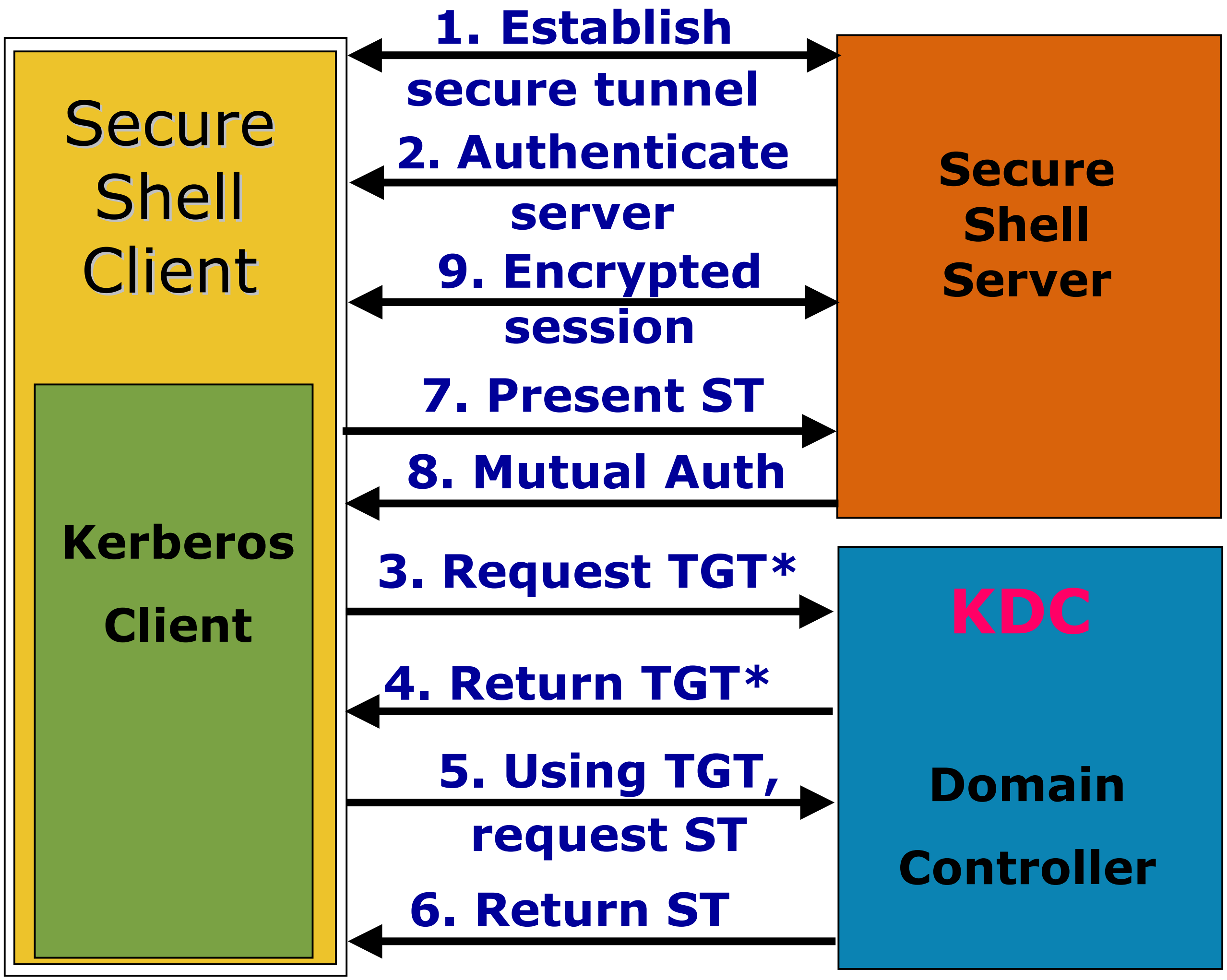
Secure Shell - Minuses

- Authentication concerns
- Administration problems
  - Each server is independently installed, configured and administered.
  - Specialized administration required if using Kerberos
- Requires regular security updates as bugs and holes are identified and fixed in the open-source implementation.

What do I need to implement SSH with Kerberos Authentication?

- OpenSSH clients and servers (included in HP-UX 11i)
- Kerberos KDC infrastructure - Open source (MIT) or commercial (Windows or others)
- Simon Wilkinson’s GSSAPI source code patches added to UNIX servers and clients (included in HP-UX 11i)
- Two lines added to sshd_config file
  - **GSSAPIAuthentication Yes**
  - **GSSAPIDelegateCredentials Yes**
- Allow transmission of 2 more ports:
  - 88 for Kerberos authentication
  - 749 for Kerberos password changing (optional)
- Clients which can acquire the initial TGT are not required unless single sign-on is desired

SSH with Kerberos Authentication

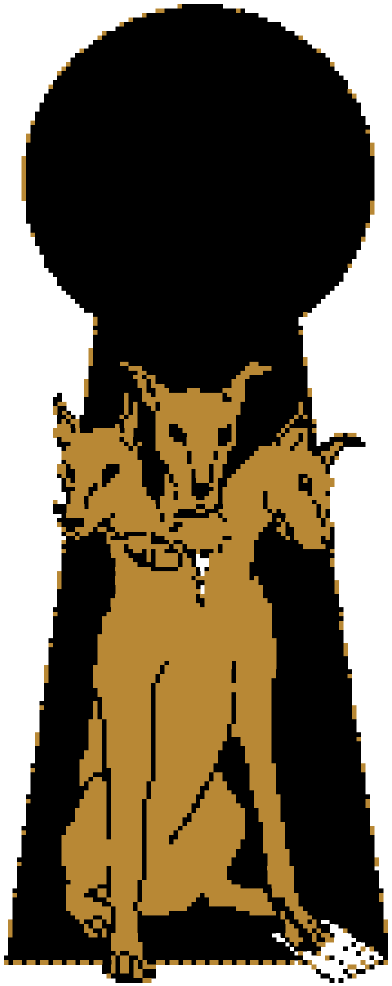


Windows single sign-on considerations

- Decide which style of KDC to use as primary
  - Microsoft Active Directory:
    - simpler to administer
    - more difficult initial setup
    - significant additional network traffic
  - UNIX-based
    - simpler to set up
    - more difficult to administer
- Host service keys need to be created and securely placed onto each target server
- Kerberos authentication software (kinit) needs to be installed on each client system

Where can I get the required open source distributions?

- Kerberos –
  - MIT web.mit.edu/kerberos/www/ (U.S. and Canada only)
  - www.crypto-publish.org (International)
- OpenSSH –
  - OpenSSH – www.openssh.org
  - Simon Wilkinson’s patches – www.sxw.org.uk/computing/patches/openssh.html



Thank you!

HP WORLD 2003

Solutions and Technology Conference & Expo

Interex, Encompass and HP bring you a powerful new HP World.

interex
shared knowledge • shared power

encompass
AN HP USER GROUP

hp
invent