

# Serwer, aplikacje klienckie i protokoły znakowych sesji terminalowych.

Patryk Kaniewski

# Intro: Urządzenia znakowe w systemach Unixopodobnych

# Wszystko jest plikiem

- W systemach Unix istnieje jedna struktura systemów plików:  
Drzewo
- Wszystkie zamontowane systemy plików znajdują się w tej hierarchii
- Wirtualne systemy plików (/proc/, /dev/mem) znajdują się w tej hierarchii
- Urządzenia (?!) znajdują się w tej hierarchii

# Urządzenia znakowe

- Pisanie większości sterowników w systemach Unix polega na implementacji operacji plikowych
- Przynajmniej `open()` `read()/write()` `close()` system call
- Jednym przykładem takiego sterownika jest urządzenie znakowe

# Demo 1:

## Urządzenie znakowe

<https://lwn.net/Kernel/LDD3/>

# Telnet

# Telnet

- Prosty protokół przesyłania głównie komend przez TCP/IP
- Brak uwierzytelniania hosta
- Brak szyfrowania połączenia
- Wyparty przez SSH

SSH



# Idea

- rsh, rlogin, telnet są niezabezpieczone
- Używanie różnych standardów szyfrowania (negocjacja)
- Bardziej uniwersalny

# Schemat SSH

- Transport Layer
- User authentication Layer
- Connection Layer

# Transport Layer: Wymiana Kluczy

- W większości przypadków nad TCP/IP
- Zapewnia poufność początkowego połączenia
- Wymienia klucze (Diffie-Hellman, ECDH)
- Jest interfejsem do przysyłania pakietów w plaintext
- Renegocjuje klucze co jakiś czas

# User authentication layer

- Umożliwia autoryzację połączenia
- Jest inicjowana przez klienta
  - Hasło
  - Klucz publiczny/prywatny
  - Interaktywna

# Demo 2: Autoryzacja SSH

# Connection layer

- shell
- direct-tcpip
- forwarded-tcpip

## Connection layer: Kanały

- Jedno połączenie SSH może zawierać wiele wirtualnych „połączeń”
- Te połączenia nazywane kanałami

# Connection layer: Forwarding

- Istnienie kanałów oraz uniwersalność pozwala na tunelowanie połączeń przez SSH
- Można łączyć porty
- Można nawet przekazywać klienta X window



# Demo 3: Tunel SSH

# Transfer plików przy użyciu SSH

# SCP

- SCP jest prostym rozwiązaniem pozwalającym na bezpieczny transfer plików
- Oparty na SSH
- Bardzo prymitywny i nie posiada opcji przydatnych dla użytkownika

# SFTP

- SFTP jest zastosowaniem protokołu SSH do tunelowania innego połączenia
- SFTP będąc wzorowany na FTP pozwala na większą ilość akcji niż sam transfer plików
- Pozwala on na przeszukiwanie katalogów, ponawianie transferu oraz usuwanie plików

# SSHFS

- Jest to rozwiązanie umożliwiające montowanie zewnętrznego systemu plików na maszynie lokalnej
- Jako prawdziwy system plików pozwala na całkowitą kontrolę nad ich używaniem za pomocą standardowych interfejsów

# Demo 4: SSH file transfer

[https://github.com/p7tryk/  
aplikacjesieciowe](https://github.com/p7tryk/aplikacjesieciowe)

# Bibliografia

- RFC 4251-54
- Jonathan Corbet, Alessandro Rubini, and Greg Kroah-Hartman.  
*Linux Device Drivers, Third Edition*. 2005
- <https://www.redhat.com/sysadmin/sshfs>
- <https://unix.stackexchange.com/questions/44818/how-can-i-connect-to-a-remote-x-server-without-ssh/257097#257097?s=00607d9d7b644635ae9e381731b7481e>
- <https://unix.stackexchange.com/questions/33557/using-an-already-established-ssh-channel>