



Długa ręka Shella.

Dariusz.Puchalak@osec.pl

Dariusz Puchalak

- 19+ lat Linux/Unix Sysadmin
- 7+ lat trener
- 6+ m-cy w OSEC

OSEC

- 6+ lat na rynku
- doświadczona kadra (ACNI, RHCA)
- specjalizacja open-source

Shell

- Interfejs użytkownika (Powłoka systemowa)
 - Tekstowa
 - Graficzna

Program - Filtr

Program Uniksowy/Linuksowy jest filtrem. Akceptuje dane na wejściu, przetwarza je i wysyła na swoje wyjście.

Typowe użycie:

- `grep "regex" /etc/passwd | wc -l`¹
- `find /etc -name "*.conf" | wc -l`
- `find /etc/rsyslog.d/ -name "*.conf" -exec grep -o '/var/log' \; | sort -u | wc -l`
- ...

1. Bardziej formalnie poprawne byłoby użycie: `grep -c "regex" /etc/passwd`

Historia

- Stworzony przez Tatu Ylonen (1995)
- Bezpieczne logowanie do zdalnych komputerów
- Uwierzytelnianie
- Szyfrowanie przesyłanych danych
- Integralność przesyłanych danych

Ochrona przed:

- IP spoofing
- IP source routing
- DNS spoofing
- Password sniffing
- Manipulation of transferred data
- Ataki na X11 (podśluchanie uwierzytelnienia)

r-komendy

- SSH przedłuża działanie shella na zdalne komputery.

rexec

ssh host "cat /etc/passwd"

rlogin

ssh user@host

rcp:

scp file host.domena.pl:

Metody uwierzytelnienia

- password,
- hostbased,
- publickey,
- Keyboard-interactive
- PAM
- keyboard-interactive:pam

Uwierzytelnianie poprzez publickey.

- Nie wymaga centralnej infrastruktury (ad-hoc)
- Uwierzytelnianie dwuskładnikowe (co wiem, co posiadam)
- Single Sign On - dzięki agentowi podajemy hasło tylko raz podczas rozpoczęcia pracy
- Bardzo proste w użyciu:
 - ssh-keygen
 - ssh-copy-id `user@komputer`



Co nowego w OpenSSH.

Curve25519

- Daniel J. Bernstein's Curve25519

Nowy format przechowywania kluczy.

- Based on bcrypt

Nowy sposób szyfrowania.

- chacha20-poly1305@openssh.com
- Szyfr strumieniowy ChaCha20
- Kod uwierzytelniania wiadomości Poly1305

ssh-keygen

- Generowanie pary kluczy.
- Zalecane algorytmy (-t) : rsa, ed25519
- Zalecany nowy format klucza (-o) dla OpenSSH ≥ 6.5
- Łatwe zarządzanie know_hosts
 - `ssh-keygen -f "/home/puchalakd/.ssh/known_hosts" -R 192.168.100.179`

AuthorizedKeysCommand

- Pobieranie `authorized_keys` za pomocą komendy.
- `AuthorizedKeysFile`
- `AuthorizedKeysCommand`
- `AuthorizedKeysCommandUser` (dedykowane konto)

PKI

- OpenSSH CERTIFICATES
- AuthorizedPrincipalsFile
- TrustedUserCAKeys
- RevokedKeys

PKI

- KEY REVOCATION LISTS (format)
- CERTIFICATES format

Nowy format przechowywania kluczy.

- Based on bcrypt

Match

- User
- Group
- Host
- LocalAddress
- LocalPort
- Address
- All

Match

AcceptEnv, AllowAgentForwarding, AllowGroups, AllowTcpForwarding, AllowUsers, AuthenticationMethods, AuthorizedKeysCommand, AuthorizedKeysCommandUser, AuthorizedKeysFile, AuthorizedPrincipalsFile, Banner, ChrootDirectory, DenyGroups, DenyUsers, ForceCommand, GatewayPorts, GSSAPIAuthentication, HostbasedAuthentication, HostbasedUsesNameFromPacketOnly, KbdInteractiveAuthentication, KerberosAuthentication, MaxAuthTries, MaxSessions, PasswordAuthentication, PermitEmptyPasswords, PermitOpen, PermitRootLogin, PermitTTY, PermitTunnel, PermitUserRC, PubkeyAuthentication, RekeyLimit, RhostsRSAAuthentication, RSAAuthentication, X11DisplayOffset, X11Forwarding, X11UseLocalHost.

Tożsamości

Host s1.puchalak.net

IdentityFile ~/.ssh/s1.key

IdentitiesOnly yes

Wildcards

Host s??w.puchalak.net

IdentityFile ~/.ssh/show.key

IdentitiesOnly yes

User rootdp

Match po stronie klienta

- exec
- host
- originalhost
- user
- localuser
- All

Match po stronie klienta

Match host sun exec "ping -c 1 -W 1 192.168.1.1"

Hostname 192.168.0.45

Host sun

Hostname 1.2.3.4

Port 2024

ProxyCommand ssh ovh-https nc %h %p

