



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-add
 - ssh-copy-id `user@komputer`

Zdalne wykonywanie komend

Wykonanie backupu systemu plików / na zdalnej maszynie:

```
ssh remote "tar -cSzv --one-file-system -C / -f - ." |  
cat > remote-backup-root.tar.gz
```

Skopiowanie katalogu 2BACKUP ze zdalnej maszynie (tylko przesyłamy przez sieć różnice, ssh wykorzystane „niejawnie”):

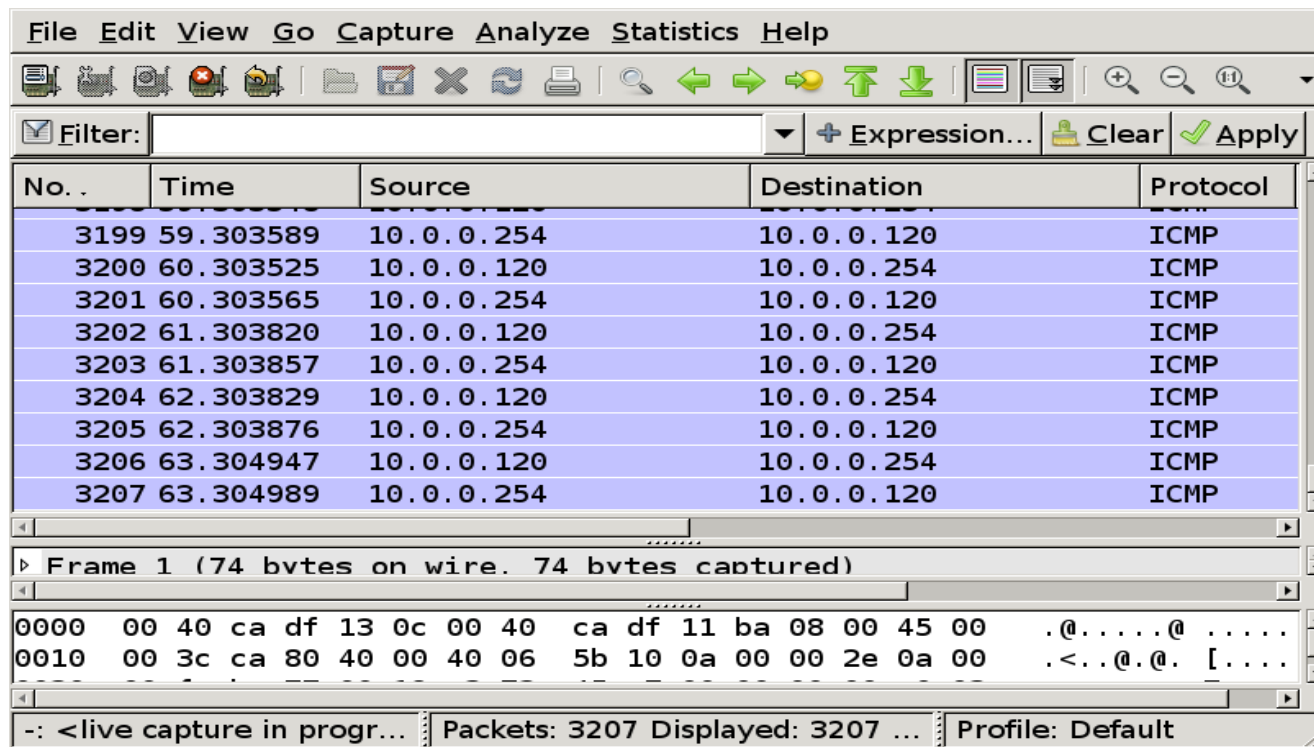
```
rsync -avH remote:2BACKUP .
```

SSH oraz wykorzystanie potęgi filtrów

```
ssh rootdp@hostA "tar -cSzv -C / -f -  
/u02/_installs/9iAS/" | ssh rootdp@192.168  
.1.44 "tar -xpSzv -C / -f -"
```

Zdalne przechwytywanie ruchu sieciowego.

```
ssh root@10.0.0.254 "tcpdump -l -n -s 0 -w - not  
port 22" | wireshark -i -
```



The image shows the Wireshark network protocol analyzer interface. The main pane displays a list of 10 captured packets, all of which are ICMP Echo (ping) requests. The packets alternate between source IP 10.0.0.254 and destination IP 10.0.0.120. The bottom pane shows the details of the selected packet (Frame 1), displaying the raw packet data in hexadecimal and ASCII format.

No.	Time	Source	Destination	Protocol
3199	59.303589	10.0.0.254	10.0.0.120	ICMP
3200	60.303525	10.0.0.120	10.0.0.254	ICMP
3201	60.303565	10.0.0.254	10.0.0.120	ICMP
3202	61.303820	10.0.0.120	10.0.0.254	ICMP
3203	61.303857	10.0.0.254	10.0.0.120	ICMP
3204	62.303829	10.0.0.120	10.0.0.254	ICMP
3205	62.303876	10.0.0.254	10.0.0.120	ICMP
3206	63.304947	10.0.0.120	10.0.0.254	ICMP
3207	63.304989	10.0.0.254	10.0.0.120	ICMP

Frame 1 (74 bytes on wire (588 bits) captured (74 bytes captured on interface (588 bits) on eth0)

0000 00 40 ca df 13 0c 00 40 ca df 11 ba 08 00 45 00 .@.....@
0010 00 3c ca 80 40 00 40 06 5b 10 0a 00 00 2e 0a 00 .<...@.@. [...

-: <live capture in progr... | Packets: 3207 Displayed: 3207 ... | Profile: Default

Zdalne

SSH to przedłużenie naszego shella na zdalne maszyny.

Ale do tego potrzeba posiadać system POSIX'owy który nie zabroni nam takich możliwości.

Polecany dowolny Linux. :)

Polecane OpenSSH (standardowo występuję w dowolnej dystrybucji Linuksa i nie tylko tam).

Opcje

OpenSSH w wersji 6.7 posiada 42 przełączniki dostępne z linii poleceń.

Jak się w tym wszystkim połąpać?

\$HOME/.ssh/config

Host router

Hostname 192.168.1.1

Port 2022

User root

Ciphers aes256-cbc

MACs hmac-sha1

Globalne

Host *

StrictHostKeyChecking ask

GSSAPIAuthentication no

ForwardAgent no

ForwardX11 no

ForwardX11Trusted no

GatewayPorts no

Protocol 2

CheckHostIP yes

Ciphers aes256-ctr,aes128-ctr

MACs hmac-sha2-512,hmac-sha2-256,hmac-ripemd160

KexAlgorithms diffie-hellman-group-exchange-sha256,diffie-hellman-group14-sha1,diffie-hellman-group-exchange-sha1

ServerAliveInterval 30

TCPKeepAlive yes

Lokalne zmiany

Host *.puchalak.???

IdentityFile ~/.ssh/id_puchalak

IdentitiesOnly yes

PreferredAuthentications publickey,password

LocalForward

```
$HOME/.ssh/config
```

```
Host corp-remote
```

```
    HostName XXX.corp.pl
```

```
    LocalForward 1100 mail.int.corp:110
```

```
    LocalForward 1025 mail.int.corp:25
```

```
    LocalForward 1143 mail.int.corp:143
```

```
$ ssh corp-remote
```

Klient poczty skonfigurowany na:

POP3 localhost:1100

IMAP localhost:1143

SMTP localhost:1025

RemoteForward

RemoteForward 65020 127.0.0.1:22

Własny serwer proxy

DynamicForward 1080

Socks4/Socks5 proxy on localhost:1080

ssh remote.site.pl

Go to any site with „your address is” and you are connected from:

remote.site.pl

OpenSSH dla kolegów/koleżanek/...

GatewayPorts yes

GatewayPorts clientspecified

...

RemoteForward przecieki.pl:2080 internal.corp.pl:80

...

GatewayPorts no

RemoteForward [localhost]:2080

LocalForward [localhost]:1080

DynamicForward [localhost]:1080

X11 over SSH

```
ssh -X user@host netscape
```

Trusted X11 forwarding:

```
ssh -Y user@host
```

Host gui

Hostname 192.168.1.99

User demo

ForwardX11 yes



O jeden host za daleko...

Połączenie przez komputer pośredni

```
.ssh/config
```

```
...
```

```
Host hostB
```

```
    ProxyCommand ssh hostA nc %h %p
```

```
    HostName 10.1.8.31
```

```
Host hostA
```

```
    HostName 172.16.48.10
```

```
...
```

```
bash$ ssh hostB
```

Połączenie przez proxy WWW

Gdy trzeba się uwierzytelnić do serwera proxy:

```
ProxyCommand nc -X connect -x  
192.168.1.1:8080 %h %p
```

```
netcat -X proxy_protocol
```

Supported protocols are ... “connect” (HTTPS proxy).

Nowa jakość

Host Live

Hostname 192.168.100.250

User puchalakd

LocalForward 3137 localhost:3389

PermitLocalCommand yes

LocalCommand rdesktop -c -xm -D -K -z -u puchalakd
-p TAJNEHASŁO -k pl -g 1024x768 localhost:3137 &

ProxyCommand ssh posrednik nc %h %p sleep 1

ConnectTimeout 10



A može VPN?

OpenSSH VPN

Host sshgateway

Tunnel yes

TunnelDevice 0:any

PermitLocalCommand yes

LocalCommand sh /etc/netstart tun0



Non-interactive?

SSH and cron

command="cat /etc/passwd" ssh-rsa AAAA[.....]sagSH klucz123

from="serverA.net"

idle-timeout=5m

no-agent-forwarding

no-port-forwarding

no-X11-forwarding

no-pty

permitopen="hostB.domain:12345"

tunnel="n"

SSH and cron

```
rsync -e 'ssh -i /home/puchalakd/key1' \  
--bwlimit=200 -avhHP --partial \  
root@x.y.z.net:/content-web/ /mnt/cont/
```

SSH and cron

```
from="A.B.C.D",command=" rsync --server  
--sender -vIHogDtpre.iLsf --bwlimit=200 . /content-  
web/",no-agent-forwarding,no-port-forwarding,no-  
pty,no-user-rc,no-X11-forwarding ssh-rsa
```

```
Afy5fJlOojCZhI1CZJ7qfsdnks.....+hK7Ksw==  
puchalakd@mirror rsync content
```



Moja aplikacja nie obsługuje SSH :(



SSHFS

Network filesystem using SSH
(Needs FUSE)

SSHFS

```
sshfs mirror:/mnt ~/mnt/
```

```
mirror:/mnt 6801418 4945780 1855639 73% /home/.../mnt
```



Reusing Control Connection

Tylko jedno połączenie

Host *

ControlMaster auto

ControlPath /tmp/%r@%h:%p



Mniej typowe zastosowania.



SSH w pakietach DNS

```
cat $HOME/.dns2tcp
```

```
domain = your.domain.tld
```

```
ressources = ssh
```

```
local_port = 4430
```

```
server = 10.12.23.45
```

```
cat $HOME/.ssh/config
```

```
Host remote-via-dns
```

```
    HostName localhost
```

```
    DynamicForward 1080
```

```
    Port 4430
```

SSH w pakietach DNS

dns2tcpc -r ssh

Listenning on port : 4430

ssh remote-via-dns

16:30:04.642528 IP 10.10.10.37.49062 > 10.12.23.45.53: 47518+ KEY? AACBgCTBA3NzaA==.your.domain.tld. (50)

16:30:04.657650 IP 10.12.23.45.53 > 10.10.10.37.49062: 47518 1/0/0 (74)

16:30:04.658668 IP 10.10.10.37.49062 > 10.12.23.45.53: 61644+ TXT? 2vIAAABBA==.your.domain.tld. (46)

16:30:04.713632 IP 10.12.23.45.53 > 10.10.10.37.49062: 61644 1/0/0 (111)

16:30:04.713952 IP 10.10.10.37.49062 > 10.12.23.45.53: 10728+ TXT? 2vIAAQACBA==.your.domain.tld. (46)

16:30:04.715062 IP 10.10.10.37.49062 > 10.12.23.45.53: 22382+[[domain]

16:30:04.715115 IP 10.10.10.37.49062 > 10.12.23.45.53: 2755+ TXT? 2vIAAAEBA==.your.domain.tld. (46)

16:30:04.716214 IP 10.10.10.37.49062 > 10.12.23.45.53: 12437+[[domain]

16:30:04.716319 IP 10.10.10.37.49062 > 10.12.23.45.53: 61821+ TXT? 2vIAAAAGBA==.your.domain.tld. (46)

16:30:04.716370 IP 10.10.10.37.49062 > 10.12.23.45.53: 22220+[[domain]

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 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

