SHELL SCRIPTING

Scriptek elejére:

```
#!/bin/zsh (vagy #!/bin/bash)
```

Változók deklarálása

```
kali@kali:~$ somevariable="ABCD"
kali@kali:~$ echo $somevariable
```

```
ABCD
```

Változók kezelése kapcsos zárójellel:

```
kali@kali:~$ teststring="Hello"
kali@kali:~$ echo ${teststring}World
```

```
HelloWorld
```

Beépített parancsok kezelése "hagyományos" zárójellel (vagy ún. backtick-el: ‘. . . ‘):

```
kali@kali:~$ echo $date (nem a kívánt eredmény)
kali@kali:~$ echo ${date} (szintén nem jó)
kali@kali:~$ echo $(date)
```

```
Sun Feb 6 08:49:42 AM EST 2022
```

Matematikai műveletek dupla zárójellel:

```
kali@kali:~$ a=13
kali@kali:~$ b=5.9 (megj.: zsh-ban támogatott a tizedestört, bash-ban nem)
kali@kali:~$ echo $((a+b))
```

```
18.899999999999999 (a kerekítéssel gondok vannak azért)
```

Teljes escape egyszeres idézőjellel:

```
kali@kali:~$ echo '$(date)'
```

```
$(date)
```

Részleges escape dupla idézőjellel:

```
kali@kali:~$ "echo Teszt"
kali@kali:~$ echo "$(date)"
```

```
echo Teszt: command not found
Sun Feb 6 09:39:33 AM EST 2022
```

Feltételek (néhány példa):

-eq	egyenlő	-lt	kisebb	-z	üres string
-ne	nem egyenlő	-le	kisebb-egyenlő	-n	nem üres string
-gt	nagyobb	&&	és	-e	létezik a fájl
-ge	nagyobb-egyenlő	||	vagy	-r/w/x	olvasható/írható/futtatható

If-elif-else:

```
kali@kali:~$ a=6
kali@kali:~$ if [[ $a -gt 7 ]] then
then> echo "7-nél nagyobb"
then> elif [[ $a -lt 10 ]] then
elif-then> echo "10-nél kisebb"
elif-then> else
else> echo "Egyik sem"
else> fi
```

Két "életszerűbb" példa:

```
kali@kali:~$ echo "Ez egy tesztszöveg" > teszt.txt
kali@kali:~$ du teszt.txt
4  teszt.txt
kali@kali:~$ if [[ "$(du teszt.txt | cut -f 1)" -gt 3 ]] then
then> echo "A méret nagyobb, mint 3KB!"
then> fi
```

```
kali@kali:~$ ls -l teszt.txt
rw-r-r- 1 kali kali 20 Feb 6 12:23 teszt.txt
kali@kali:~$ if [[ -w teszt.txt ]] then
then> echo "A fajlt a tulajdonos írhatja!"
```

Case:

```
kali@kali:~$ a=5
kali@kali:~$ case $a in
case> 4)
case> echo "Négy"
case> ;;
case> 5)
case> echo "Öt"
case> ;;
case> *)
case> echo "Ki tudja?"
case> esac
```

Ciklusok:

while:

```
kali@kali:~$ a=5
kali@kali:~$ while [[ $a -lt 10 ]] do
while> echo "a értéke: $a"
while> a=$((a+1))
while> done
```

for:

```
kali@kali:~$ for i in {1..20..2}; do
for> echo $i
for> done
```

Tömbök (zsh indexelés 1-től, bash 0-tól):

```
kali@kali:~$ a=(5 8 13 'teszt')
kali@kali:~$ a+=17 (nem matematikai művelet)
kali@kali:~$ for i in $a; do echo $i; done
```

Függvények (argumentumok száma: $#, az argumentumok listája $@):

```
kali@kali:~$ tesztfuggveny(){
function> echo "Az argumentumok száma: $#"
function> for i in $@;do echo $i;done
function> }
```

"Return"-re célszerű echo-t használni:

```
kali@kali:~$ returnfuggveny(){
function> a=$1
function> a+=$2 (csak összefűzés, nem összeadás)
function> echo $a
function> }
kali@kali:~$ eredmeny=$(returnfuggveny elso masodik)
kali@kali:~$ echo $eredmeny
```

```
elsomasodik
```

Nmap alapok

Host discovery PING scannel:

```
kali@kali:~$ sudo nmap -sn -PE <host>
```

"Hagyományos" webcímek esetén ne felejtsük el a `-R` kapcsolóval bekapcsolni a reverse DNS feloldást.
Ha a hagyományos ICMP echo nem működik, `-PE` helyett próbálkozhatunk `-PP` vagy `-PM`-el is (szintén ICMP, de echo helyett timestamp és address requestek). Ha ezek sem mennek akkor UDP pinggel (`-PU <port>`), TCP SYN, vagy ACK pinggel (`-PS <port>, -PA <port>`) talán sikeresek lehetünk.

Több IP (web) címet (és később portot) vagy egyszerűen szóközzel elválasztva (pl. `abc.com def.com xyz.com`, vagy intervallumok esetén kötőjellel adhatunk meg (`192.168.1.1-255`, vagy akár `192.168.1-100.1-255`).

Ha rájöttünk a célpont címére, akkor elkezdhetünk alaposabb vizsgálatokat folytatni (megj.: a fenti `-sn` kapcsoló azért felelős, hogy kikapcsoljuk a port scant. Ne felejtsük el, mert egy olyan hálózaton ahol akár több száz gép is lehet nagyon sokáig fog tartani ha a felderítés mellett még a portokat és vizsgáljuk).

Néhány hasznos kapcsoló:

-v verbose (beszédes) mód	-p portlista - (T)CP, (U)DP	-O OS felderítés
-sn nincs port scan (régen -sP)	-F leggyakoribb portok	-sV verzió meghatározás
-PE,PP,PM ICMP host disc.	-sS TCP SYN scan (alap)	-sC alap szkriptek használata
-PU UDP ping	-sA TCP ACK scan	–script=<lista>megadott
-PS,PA TCP SYN/ACK ping	-sU UDP scan	szkriptek használata

Megjegyzés: Kali Linuxon az Nmap scriptek alapból a `/usr/share/nmap/scripts` mappában vannak.

SSH alapok

Eszközhöz való csatlakozás SSH segítségével:

```
kali@kali:~$ ssh -p <portszám> <felh. név>@<cél>
kali@kali:~$ ssh -p 1234 user@192.168.1.10
```

A `-p` kapcsoló nélkül alapból a 22-es porton csatlakozik. A felh. név megadása nélkül egy "login as" promptot dob fel (ekkor persze a @ sem kell).

Eszközhöz való csatlakozás privát kulcs felhasználásával:

```
kali@kali:~$ ssh -p <portszám> -i <kulcs> <felh. név>@<cél>
```

Megjegyzések:

- a privát kulcs publikus párjának szerepelnie kell a felhasználó `~/.ssh/authorized_keys` fájljában (ez az alapbeállítás, természetesen a hely megváltoztatható).
- a privát kulcson megfelelő jogosultságnak kell lennie (pl. 400).

SSH Tunneling belső hálózatba (local port forwarding):

```
kali@kali:~$ ssh -L <saját IP>:<saját port>:<cél>:<cél port> <felh.
név>@<köztes>
```

SSH Tunneling külső hálózatba (remote port forwarding):

```
kali@kali:~$ ssh -R <távoli IP>:<távoli port>:<cél>:<cél port> <felh.
név>@<köztes>
```

Megjegyzések:

- a "belső" hálózat azt jelenti, hogy a saját számítógépünkről szeretnénk belső hálózaton lévő eszközhöz csatlakozni egy mindkét gép által elérhető köztes eszközről.
- a "külső" hálózat a fentiekhez hasonló, de itt "mi vagyunk" a belső hálózat és egy külső eszköznek adunk jogot a csatlakozáshoz.
- ha a "saját IP" alapbeállítása a localhost, míg ha a "távoli IP" üres akkor bármilyen IP-ről engedélyezzük a csatlakozást.

Privát kulcs készítésére két lehetőség:

```
kali@kali:~$ openssl genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:2048
> privat.pem
kali@kali:~$ ssh-keygen -t rsa -b 2048
```

Publikus - SSH által elfogadott - kulcs készítése privátból:

```
kali@kali:~$ ssh-keygen -f privat.pem -y > publikus.pub
```

DIRB, Dirbuster

Weboldal szerkezetének feltérképezése `DIRB` segítségével:

```
kali@kali:~$ dirb <cím> <szólista> (-r)
```

Megjegyzések:

- a `-r` kapcsolóval a rekurzív keresést **kapcsolhatjuk ki**.

- a dirb helyett előnyösebb a `gobuster` használata, mert ez képes párhuzamosan futtatni a keresést
- Kali Linuxon a szólisták alapból a `/usr/share/wordlists` mappában vannak

Szólista generálása:

```
kali@kali:~$ dirb-gendict -h
Usage: dirb-gendict -type pattern
type: -n numeric [0-9]
-c character [a-z]
-C uppercase character [A-Z]
-h hexa [0-f]
-a alfanumeric [0-9a-z]
-s case sensitive alfanumeric [0-9a-zA-Z]
pattern: Must be an ascii string in which every 'X' character wildcard
will be replaced with the incremental value.

Example: dirb-gendict -n thisword_X
 thisword_0
 thisword_1
 [...]
 thisword_9
```

Dirbuster használata: