Linux firewall skript

Semestrální práce Y36SPS

Vlastimil Vagner

vagnev1@fel.cvut.cz

Zadání:

Skript pro „automatické zabezpečení linuxové stanice“. Skript, který prohledá operační systém, a podle zvoleného typu systému doporučí postup pro zabezpečení stroje proti síťovým útokům. Cíle se hodlá dosáhnout konfigurací iptables na základě zadání několika jednoduchých parametrů.

Příprava:

Tvorba skriptu probíhala na stolnímu PC se dvěma síťovými kartami, jenž je pomocí jedné síťové karty připojen k internetu a pomocí druhé je připojen k notebooku. Tento stolní počítač tedy funguje jako Router-PC a zajišťuje notebooku konektivitu pomocí SNAT překladu adres. Modelová situace na routeru se 2 síťovými kartami, kde na jedné straně je internet a na straně druhé lokální síť je v prostředí malých firem velmi častá.

Na obou počítačích je nainstalován operační systém linux (distribuce Ubuntu 6.06)

Ačkoli byla předtím zmíněna modelová situace PC routeru, jež funguje v malých firmách jako brána mezi internetem a lokální sítí, přesto také bývá na takovém routeru provozován i POP3 server a další služby. Pro demonstrativní účel byly nasazeny v této práci níže uvedené služby.

Na Router-PC běží tyto služby:

- FTP
- HTTP(s)
- SMTP
- SSH
- POP3
- IMAP
- RSYNC
- DNS

Veškerá konfigurace probíhá tedy právě na zmíněném router-PC.

K ověřování PINGu a zásahů zvenku byl pověřen soused.

IPTABLES:

Iptables je nástroj, který umožňuje linuxovému nebo unixovému systému plně pracovat se síťovou komunikací. Pomocí něj si můžeme snadno postavit různé druhy firewallů (stavový, transparentní ...) nebo sdílení internetu, zkrátka snadno řídit velkou síťovou dopravní křižovatku na serveru.

Základní syntaxe iptables:

```
iptables [tabulka] [akce] [chain] [ip_část] [match] [target] [target_info]
```

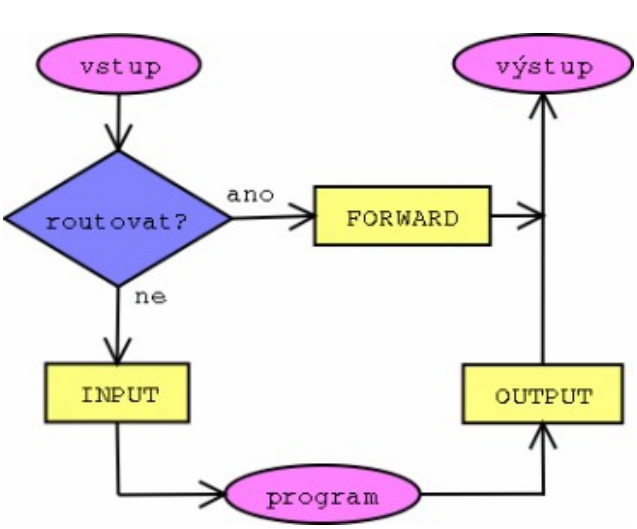
Jeho použití je vázáno na jádra verzí 2.4. Ačkoliv je možné na těchto verzích používat také starší program ipchains.

Existují 3 základní tabulky: filter (výchozí), nat a mangle. Poznáme je podle přepínače `-t` v příkazu. Tabulky si můžeme představit jako 3 katalogy, každý z nich má svoje položky – řetězce

Stavový firewall:

Každý IP datagram s sebou nese vyjma vlastních užitečných dat také hlavičku, obsahující zejména IP adresu původce i adresáta, zdrojový a cílový port specifikující program, kterému je datagram určen, a další informace popisující komunikaci, ke které datagram náleží. Paketový firewall je pak jakýmsi filtrem, který na základě těchto informací rozhoduje o tom, které pakety mohou být připuštěny až k programům, nebo které naopak smějí opustit počítač.

Každý paket, ať už je jeho původ jakýkoli, prochází systémem řetězců, které tvoří filtrovací tabulku.



1. Jádro se rozhoduje, zda-li je příchozí paket určen pro tento počítač, nebo jestli je potřeba jej routovat jinam.
2. Je-li adresátem router, předá paket k dalšímu zpracování do vstupního (INPUT) řetězce
3. Pokud je datagram určen někomu jinému v síti, bude postoupen do řetezce FORWARD a router se jej pak pokusí podle svých možností doručit příjemci
4. Pokud datagram vytvořil některý z lokálních programů, je nutné, aby paketový filtr jej opustil přes řetezec OUTPUT.

Stavový firewall při filtrování bere v úvahu nejen informace obsažené v záhlaví zkoumaného datagramu, ale dokáže na něj nahlížet komplexně, v kontextu spojení, do kterého patří. Stavový firewall rozezná paket, který otevírá nové spojení, od paketů, které tuto komunikaci realizují, a díky tomu lze precizněji filtrovat datové toky.

Každý zkoumaný datagram (a to nejen TCP segment, ale i UDP paket) je pak zařazen do některé z těchto kategorií:

- NEW - datagram otevírá novou komunikaci
- ESTABLISHED, RELATED - datagram je součástí již navázaného spojení nebo s ním nějakým způsobem souvisí.
- INVALID - datagram není součástí žádného spojení nebo se jej nepodařilo identifikovat.

Na základě stavové informace můžeme tedy pakety třídit.

Cíle firewallu:

Optimalizovat datový tok skrz router.

Nastavit dynamické směrování v síti a zajistit bezproblémový chod.

Zakázat vše co není povoleno.

Logovat chyby.

Zajistit ochranu proti základním typům útoků:

- syn flood
- ip spoofing
- zneužití služeb
- zahlcení icmp zprávami

Postup:

Nastavení zakladní konfigurace:

```
IPTABLES=/sbin/iptables
LO_IFACE=lo
INET_IFACE=eth0
LAN_IFACE=eth1
INET_IP="1.2.3.4"
LAN_IP="192.168.0.1/32"
LAN_BCAST="192.168.0.255/32"

#------------------------------------------------------------------------#
#                    Moduly a inicializace
#------------------------------------------------------------------------#
```

Zavedení modulů pro nestandardní cíle:

```
/sbin/modprobe ipt_REJECT
/sbin/modprobe ipt_MASQUERADE
```

Moduly pro FTP prenosy

```
/sbin/modprobe ip_conntrack_ftp
/sbin/modprobe ip_nat_ftp
```

Cesta k programu iptables

```
IPTABLES="/sbin/iptables"
```

Zapnuti routování paketu přes ip_forward a tcp_syncookies:

```
echo "1" > /proc/sys/net/ipv4/ip_forward
echo "1" > /proc/sys/net/ipv4/tcp_syncookies
```

Nastavení rp_filteru na zamezení IP spoofovani. IP spoofování znamená metodu podvržení IP adresy:

```
for interface in /proc/sys/net/ipv4/conf/*/rp_filter; do
  echo "1" > ${interface}
done
```

```
#--------------------------------------------------------------------------#
#                    Výchozí politika iptables
#--------------------------------------------------------------------------#
```

Nastaví se defaultní politika pro INPUT, OUTPUT, FORWARD. Výstup přímo z routeru ven je implicitně povolen. FORWARD a INPUT implicitně zákázán.:

```
$IPTABLES -P INPUT      DROP
$IPTABLES -P OUTPUT   ACCEPT
$IPTABLES -P FORWARD  DROP
```

Zde se nastavuje limit pro příchozí SYN spojení. Tento postup chrání PC před SYN FLOODS (přetečení SYN požadavky) tím, že propustí pouze 4 SYN segmenty za sekundu:

```
$IPTABLES -N syn-flood
$IPTABLES -A syn-flood -m limit --limit 20/s --limit-burst 5 -j RETURN
$IPTABLES -A syn-flood -j DROP
```

```
#--------------------------------------------------------------------------#
#                    INPUT
#--------------------------------------------------------------------------#
```

Pakety od již navázaných spojení se mají povolit. Tímto se z paketového filtru stává stavový firewall. :

```
$IPTABLES -A INPUT -i $INET_IFACE -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Aktivity na Loopback rozhraní povolit pro všechny případy. Není třeba omezovat samého sebe:

```
$IPTABLES -A INPUT -i $LO_IFACE -j ACCEPT
```

Aktivity na místní síti LAN povolit pro všechny případy. Jedná se totiž o důvěryhodnou síť :

```
$IPTABLES -A INPUT -i $LAN_IFACE -j ACCEPT
```

Povolit propouštění ICMP ping. Následuje filtrování ostatních ICMP útoků:

```
$IPTABLES -A INPUT -i $INET_IFACE -p ICMP --icmp-type echo-request -j ACCEPT
```

Filtrovani pokusu o syn-flooding (zahlcení navazovacími požadavky):

```
$IPTABLES -A INPUT -i $INET_IFACE -p tcp --syn -j syn-flood
```

Filtrování pokusů o zahlceni icmp (zahlcení datových cest icmp zprávami):

```
$IPTABLES -A INPUT -i $INET_IFACE -p icmp -j syn-flood
```

Autorizační službu AUTH není vhodné pomoci DROP kvůli prodlevám při navazování spojení. Zamítneme jej, ale aby nedošlo k nežádoucím prodlevám.

```
$IPTABLES -A INPUT -i $INET_IFACE -p TCP --dport 113 -m limit --limit 12/h -j LOG
$IPTABLES -A INPUT -i $INET_IFACE -p TCP --dport 113 -j REJECT --reject-with tcp-reset #AUTH server
```

Povolení všech zmíněných služeb na routeru:

```
$IPTABLES -A INPUT -i $INET_IFACE -p TCP --dport 21    -j ACCEPT        #  FTP server
$IPTABLES -A INPUT -i $INET_IFACE -p TCP --dport 22    -j ACCEPT        #  SSH server
$IPTABLES -A INPUT -i $INET_IFACE -p TCP --dport 25    -j ACCEPT        #  SMTP server
$IPTABLES -A INPUT -i $INET_IFACE -p UDP --dport 53    -j ACCEPT        #  DNS server UDP
$IPTABLES -A INPUT -i $INET_IFACE -p TCP --dport 53    -j ACCEPT        #  DNS server TCP
$IPTABLES -A INPUT -i $INET_IFACE -p TCP --dport 80    -j ACCEPT        #  WWW server
$IPTABLES -A INPUT -i $INET_IFACE -p TCP --dport 110   -j ACCEPT        #  POP3 server
$IPTABLES -A INPUT -i $INET_IFACE -p TCP --dport 143   -j ACCEPT        #  IMAP server
$IPTABLES -A INPUT -i $INET_IFACE -p TCP --dport 443   -j ACCEPT        #  HTTPS server
$IPTABLES -A INPUT -i $INET_IFACE -p TCP --dport 873   -j ACCEPT        #  rsync server
```

Zakázání služby DC++:

```
$IPTABLES -A INPUT -p udp --dport 9176 -j DROP
```

Všechno ostatní je třeba zakázat. A následně i logovat (max. 12x5 záz./hod)

```
$IPTABLES -A INPUT -m limit --limit 12/h -j LOG --log-prefix "INPUT drop: "
```

#---#
 # OUTPUT
#---#

Jak již bylo dříve uvedeno. Vše na output z routeru je defaultně povoleno

Pro SSH a FTP je třeba zajistit minimální zpoždění a pro FTP data maximální propustnost. K této optimalizaci datových cest jsou použity TOS flagy. Jsou použity flagy Minimize a Maximize delay. Použití blokovacích flagů je zde zbytečné:

```
$IPTABLES -t mangle -A OUTPUT -o $INET_IFACE -p tcp  --sport ssh -j TOS --set-tos Minimize-Delay
$IPTABLES -t mangle -A OUTPUT -o $INET_IFACE -p tcp  --dport ssh -j TOS --set-tos Minimize-Delay
$IPTABLES -t mangle -A OUTPUT -o $INET_IFACE -p tcp  --sport ftp -j TOS --set-tos Minimize-Delay
$IPTABLES -t mangle -A OUTPUT -o $INET_IFACE -p tcp  --dport ftp -j TOS --set-tos Minimize-Delay
$IPTABLES -t mangle -A OUTPUT -o $INET_IFACE -p tcp  --dport telnet -j TOS --set-tos Minimize-Delay
$IPTABLES -t mangle -A OUTPUT -o $INET_IFACE -p tcp  --sport ftp-data -j TOS --set-tos Maximize-Throughput
```

```
#--------------------------------------------------------------------------#
#                    FORWARD
#--------------------------------------------------------------------------#
```

Je nejprve potřeba nastavit překlad adres NAT. Pro domácí účely srovnatelné s použitím v malé firmě je zvolen konkrétně SNAT:

```
$IPTABLES -t nat -A POSTROUTING -o $INET_IFACE -j SNAT --to $INET_IP
```

Na NAT je nyní potřeba spustit maškarádu. Překlad adres sám o sobě působí jako vlastní způsob ochrany :

```
echo "1" > /proc/sys/net/ipv4/ip_forward
$IPTABLES -t nat -A POSTROUTING -o $INET_IFACE -j MASQUERADE
```

Zablokování problematické IP z místní sítě. Pro ukázku byla zvolena fiktivní IP 192.168.1.99 . V praxi se může jednat o uživatele který porušil pracovní kázeň a je třeba ho odpojit:

```
$IPTABLES -A FORWARD -i $LAN_IFACE -s 192.168.1.99 -j REJECT
```

Obecné povoleni provozu pro forward pro navázaná i nová spojení :

```
$IPTABLES -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A FORWARD -i $LAN_IFACE -p all -m state --state NEW -j ACCEPT
```

Musíme umožnit přesměrování portů na stanici dovnitř sítě. Připojeny notebook má IP 192.168.1.2:

```
$IPTABLES -A FORWARD -i $INET_IFACE -o $LAN_IFACE -p tcp -d 192.168.1.2 --dport ssh -j ACCEPT
```

Routování zevnitř sítě ven není z bezpečnostních důvodů potřeba filtrovat:

```
$IPTABLES -A FORWARD -i $LAN_IFACE -j ACCEPT
```

Routování zvenku je vhodné povolit pouze pro již navázaná spojení (opět princip stavového firewallu):

```
$IPTABLES -A FORWARD -i $INET_IFACE -o $LAN_IFACE -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Ostatní pakety se zahodí. Poté se logují (12 x 5 záz/hod)

$IPTABLES -A FORWARD -m limit --limit 12/h -j LOG --log-prefix "forward drop: "

Závěr:

- Nebylo možné důkladněji testovat komunikaci mezi více uzly v jedné síti z důvodu nedostatku prostředků
- Není ještě vyřešené řízení MS klientů
- Nebylo naplněno doslovné zadání. Tento skript je určen pouze pro PC, který působí jako router a zároveň jako server pro služby. Zabezpečení stanic v síti nebylo třeba řešit, jelikož
- Parametry (vnější/vnitřní IP,…) se pro přehlednost zadávají přímo do úvodní sekce skriptu, nikoli jako argument samotnému skriptu
- Implicitním povolením veškerých již navázaných spojení (established, related) se z původního paketového filtru podařilo vytvořit stavový firewall
- Překlad adres NAT a masquerade samy o sobě mohou fungovat jako způsob ochrany stanic v vnitřní síti za routerem
- Vytvořením BASH skriptu je jednoduché
- Úloha navazuje na první laboratorní cvičení SPS
- Tato semestrální práce mě obohatila o praktické poznatky z síťové bezpečnosti a principů fungování filtru IPTABLES, stejně jako o teoretické poznatky o metodách firewallingu.

Zdroje:

http://www.root.cz/clanky/stavime-firewall-2/

http://www.root.cz/clanky/vse-o-iptables-uvod/

http://shell.sh.cvut.cz/~oskar/stah/iptables.pdf

http://jk.myserver.cz/hack/iptables/

http://www.avc-cvut.cz/avc.php?id=3241