Semestrální práce z předmětu KIV/PD

Firewall PF (Packet Filter) v BSD

Yaseen ALI – A11N0092P
yaseen@students.zcu.cz

Úvod

- Co je to filtrování paketů?

V podstatě jde o přidání další úrovně do analýzy paketů, která říká, zda danému paketu je povoleno projít dál přes rozhraní směrovače či ne.

- Proč PF?

Prostě protože, PF technologie je levná, flexibilní a mocná technologie, která nám garantuje vysokoúrovňové zabezpečení proti externím možným útokům.

- Jak mocný je PF?
- PF je „binární“ nástroj (povolit *či* nepovolit).

- PF nám explicitně umožňuje omezit nebo povolit paket na úrovni zařízení, portu či zařízení-portu:

Př.: blokovat port č. 80 na všech zařízeních v LAN až na zařízení A a zařízení B.

- PF poskytuje dva typy filtrování:
  - Statické: př.: zakázat telnet z venku na jedno zařízení v LAN.
  - Dynamické (více flexibilní): př.: zakázat telnet pro nějakou skupinu zařízení z venku a zároveň povolit ho pro jinou skupinu na to samé zařízení v LAN.
- PF má sadu pravidel:
  - Pravidla filtru specifikují kritéria, která leží na 3. vrstvě (network layer of IPv4 and IPv6) a 4. vrstvě (transport layer of TCP and UDP).
  - Často použitá kritéria jsou: zdrojové a cílové adresy, zdrojové a cílové porty a protokoly.

- Když paket dorazí na PF, všechna pravidla filtru se ohodnotí (až na jednu výjimku), a podle hodnoty posledního pravidla, rozhoduje se, zda paket může projít či ne.

- FA defaultní nastavení (deny all):

Když se nasazuje PF, tak je defaultně nastaveno, že všechny IP jsou automatické blokované, a to je z bezpečných důvodů. Potom, uživatel si může explicitně definovat své pravidla (exceptions) na filtru a vytvořit svou politiku.

- PF a směrování:
  - Celý IP provoz je na začátku blokován.
  - Spolupráce směrovače s PF nám garantuje vysokoúrovňové zabezpečení síťového provozu.
- PF je v jádře (\dev\pf):
  - Uživatelské procesy kontrolují chování filtru via rozhraní ioctl (control device).

- Jsou příkazy: enable/disable filter, load ruleset, add/remove rule, get statistics apod.
- Většina používaných funkcí jsou pokryté v pfctl (control the packet filter and network address translation (NAT) device).
- Tabulky pravidel a IP adres jsou uložené v tzn. anchor (později).
- Když se zpracovává ioctl požadavek, a anchor je prázdný, tak kernel používá defaultní anchor, což je defaultní sadu pravidel.
- Anchory jsou specifikované jmény a nebo posloupností komponent oddělených znakem „/“. Poslední komponent anchoru reprezentuje poslední pravidlo, podle kterého se rozhoduje o paketu.

Konfigurace PF

- 3 základní kroky (high-level):

1. Určit, které služby jsou povolené a které nejsou (definovat pravidla filtru):

Př. Všechna zařízení v LAN přijímají maily z Internetu a nebo jen jedno centrální zařízení.

2. Definovat pakety (povolený paket, blokovaný paket):

Formulace paketů:

Action	Source	Port	Destination	Port	Type
deny	xxx.xxx.xxx.xxx	####	xxx.xxx.xxx.xxx	####	(type)
allow	xxx.xxx.xxx.xxx	####	xxx.xxx.xxx.xxx	####	(type)

3. Přeložit formulaci paketů do syntaxe filtru (směrovače):

source/mask:destination/mask:action

Kde,

source: zdrojová stanice (network, subnet or host).

mask: netmask v bitech. 32bitů pro host zařízení a 24bitů pro síť třídy C.

destination: zdrojová stanice (network, subnet or host).

action: dvě akce, P (permit): povolit a D (deny): blokovat.

Příklad:

0.0.0.0/0:206.128.14.2/32:p - povolit přístup všem (0.0.0.0/0) do zařízení (206.128.14.2/32).

0.0.0.0/0:0.0.0.0/0:d - zakázat přístup všem do všech zařízení v síti.

- Block-Policy PF:
  - Block-policy reprezentuje akci, když se rozhoduje (podle hodnoty posledního pravidla) o blokování paketu.
  - Je defaultně nastaveno, že když se blokuje paket, tak se rovno maže (drop). Ale, dá se block-policy přeimplementovat, aby výsledná akce byla jiná než „drop“.
  - Možné akce v případě blokování paketu:
    - Return-rst: jenom u TCP paket, kde se končí připojení.
    - Return-icmp/icmp6: ICMP (Internet Control Message Protocol). Defaultní nastavení: ICMP UNREACHABLE message.
    - Return: TCP-RST a ICMP UNREACHABLE pro UDP.
    - Pass: když je vše OK, tak paket projde dál.

- Syntaxe pravedel PF:

Podrobně na příkladu:

action [direction] [log] [quick] [on interface] [af] [proto protocol] \[from src_addr [port src_port]] [to dst_addr [port dst_port]] \ [flags tcp_flags] [state]

Kde:

Action: if action = p then paket goto kernal then forward. Else viz nastavení block-policy.

Direction: směr paketu, tj. do/z (in/out) rozhraní (vchází/vychází).

Log: pakety by měly být registrované via pflogd (background daemon which reads packets logged by PF to a pflog interface). *Pflog is a pseudo-device which makes visible all packets logged by PF.*

Když pravidlo tvoří status (později), potom jen pakety, které patří k tomu statusu budou registrované.

Quick: výjimka, kterou jsem zmínil, když jsem mluvil o ohodnocování pravil. Když pravidlo je označeno klíčovým slovem *quick*, tak se považuje jako poslední (tj. ignoruje se další pravila, co přijdou potom), a podle něj se rozhoduje o paketu.

Interface: jméno a nebo skupina rozhraní, kterým prochází paket. Rozhraní může být definováno pomocí příkazů *ipconfig* , anebo automaticky vytvořeno jádrem (př. egress, ppp a carp).

Af: (address family) paketu. Tj. inet pro IPv4 a inet6 pro IPv6. PF je schopný na základě zdrojové/cílové adresy určit af paketu.

Protocol: protokol 4. vrstvy paketu, a to může být:

tcp, udp, icmp, icmp6, validní jméno z */etc/protocols*, číslo mezi 0-255 anebo sada protokolů (list {udp, tcp,…}).

src-addr, dst-addr: zdrojová/cílová adresa v IP záhlaví. Adresy můžou být specifikované jako:

- Klasicky, tj. IPv4 nebo IPv6 adresy.
- CIDR síťový blok.

- Doména.
- Jméno rozhraní.
- Jméno rozhraní, následováno maskou sítě (i.e. /24).
- Jméno rozhraní v závorkách ().
- Jméno rozhraní, následováno jedním z následujících modifikátorů:
  - :network, substituce CIDR bloku (i.e. 192.168.0.0./24).
  - :broadcast, substituce broadcastové síťové adresy (i.e. 192.168.0.255).
  - :peer, substituce peer‘s address na point-to-point linkách.
  - :0, která může reprezentovat jméno rozhraní nebo jakýkoliv předchozí identifikátor.
- Tabulka, obsahující skupinu IPv4 nebo IPv6 adresy.
- Kíčové sloco urpf_failed, u zdrojové adresy, to slovo říká, že adresy musí běžet via uRPF check (Unicast Reverse Path Forwarding), když paket běží via uRPF, najde se zdrojová adresa paketu v tabulce směrovače.

- Jakákoliv varianta z předchozích s negací ! .
- List {xxx.xxx.xxx.xxx. atd.}.
- Klíčové slovo any, znamená všechny adresy.
- Klíčové slovo all, from any to any.

Src-port, dst-port: Porty se můžou být specifikované takto:

- Číslo mezi 0 – 65535.
- Validní jméno služby z /etc/services.
- Seznam portů {port1, port2,…}.
- rozsah: !=, <, >, <=, >=, :, <> nebo ><. (poslední dva jsou binární operátory, potřebují dva argumenty. Výsledný rozsah neobsahuje použité argumenty). „:“ je také binární operátor a rozsah obsahuje použité argumenty.

Tcp-flag: specifikuje flag, které musí být v záhlaví TCP, když se používá *proto tcp*.

- State: stav(skupenství), určuje, zda se bude ukládat info. O paketech.
  - No state: nebude se ukládat info. O paketech (connection will not be statefully). TCP, UDP a ICMP.
  - Keep state: defaultní nastavení filtru.
  - Modulate state: jenom u TCP. PF vygeneruje INSs (Initial Sequence Number) pro pakety, pro které pravilo platí.
  - Snyproxy state: zahrnuje poslední dvě varianty a výhoda tohoto nastavení je, že chrání server před zfalšovanými TCP SYN.

Vlastnosti PF

- Default deny (defaultní blokování): to nám usnadní práci při psaní pravidel.

Obousměrné blokování: *block in all - block out all*

- Passing traffic (funkční provoz):

Pravidla musí být striktní, tj. projít přes rozhraní může jen povolené pakety, nic jiného.

Př.

```
# Pass traffic in on dc0 from the local network, 192.168.0.0/24,
# to the OpenBSD machine's IP address 192.168.0.1. Also, pass the
# return traffic out on dc0.
```

pass in on dc0 from 192.168.0.0/24 to 192.168.0.1
pass out on dc0 from 192.168.0.1 to 192.168.0.0/24

- Klíčové slovo quick:

Už jsme o tom mluvili.

- Keeping state (uchovávání stavu – něco jako anchor):

Hodně zajímavá a důležitá vlastnost PF, urychluje práci PF. PF nahrává info. o každém připojení do tzn. „State table“, poté, PF je schopný velmi rychle určit, zda aktuální IP (paket), který dorazil na rozhraní patří k již uloženému připojení či ne. Pokud ano, tak paket projde dál rovno (tj. bez ohodnocení pravidel filtru).

Další výhoda této vlastnosti je, že je obousměrná, tj. i pakety, které jdou opačně (tj. od přijímače do odesilatele), pokud patří do tabulky stavů, tak projdou rovno rozhraním.

další výhoda taky je, že odpovídající ICMP projde rovno rozhraním, patří-li k již nahranému připojení (i.e. TCP připojení).

- Jak je to s uchováváním stavů u UDP?

Sice UDP je „stateless protocol“, protože nemá explicitní začátek a konec připojení, ale to nemá žádný vliv na vlastnost PF. Čili, statefull connection funguje i u UDP.

- Možnosti nastavení vlastnosti uchovávání stavů u PF:

Ukážeme si několik možností nastavení na příkladu:

Př.:

```
pass in on $ext_if proto tcp to $web_server \
 port www keep state \
(max 200, source-track rule, max-src-nodes 100, max-src-states 3)
```

Předchozí pravidla definují následující chování:

- *Limit the absolute maximum number of states that this rule can create to 200.*
- *Enable source tracking; limit state creation based on states created by this rule only.*
- *Limit the maximum number of nodes that can simultaneously create state to 100.*
- *Limit the maximum number of simultaneous states per source IP to 3.*

- Blokování falešných paketů:

PF poskytuje pár ochran oproti falešným paketům pomocí klíčového slova *antispoof*.

antispoof [log] [quick] for *interface* [*af*]

- URPF:

Má vliv ho používat, jen když směrování v síti je symetrické. Když paket běží via uRPF, hledá se jeho zdrojová adresa, a pokus se nenajde, tak to může znamenat, že máme falešný paket => block!

block in quick from urpf-failed label uRPF

- TCP SYN Proxy:

pass in on $ext_if proto tcp to $web_server port www synproxy state

Tady, připojení k webovému serveru bude TCP „proxied“ pomocí PF.

Nevýhody PF

Nevýhody musí existovat -)

1. Není možno kontrolovat obsah dat, které obsahuje povolené pakety.
2. PF není schopný provádět ověřování uživatelů, ať to je na úrovni aplikaci nebo na úrovni uživatele.
3. Jakmile, je nějakému protokolu povoleno projít přes interface, tak může kdokoliv jiný host navázat spojení přes tento protokol k ostatním klientům v síti, což snižuje zabezpečení sítě.
4. Potenciální nebezpeční je, že IP adresy, které projdou přes PF můžou být falešné.

Užitečné odkazy

- http://www.openbsd.org/faq/pf/filter.html#pass
- http://www.openbsd.org/faq/pf/nat.html
- http://www.gsp.com/cgi-bin/man.cgi?section=5&topic=pf.conf
- http://www.support.psi.com/support/common/routers/files/Filter-Desc.html
- http://www.isaserver.org/images/ch10.pdf
- http://www.benzedrine.cx/pf-paper.html (test performance!!!)

Q&A