



Advanced firewalling

13 martie 2014

Objective

- ▶ Advanced Firewall features
- ▶ De ce avem nevoie de Application Inspection?
 - ❑ Studiu de caz: Active FTP vs Passive FTP
- ▶ ASA
 - ❑ Modular policy framework
 - ❑ Granular connection setting
 - ❑ Advanced Application Inspection (HTTP, FTP)
 - ❑ Traffic policing
- ▶ FortiGate
 - ❑ Session helpers
 - ❑ Application Control
 - ❑ Traffic shaping
 - ❑ Fortinet configuration converter

Advanced Firewall features



Application
Inspection



Traffic
shaping/policing



Intrusion
Prevention



VPN



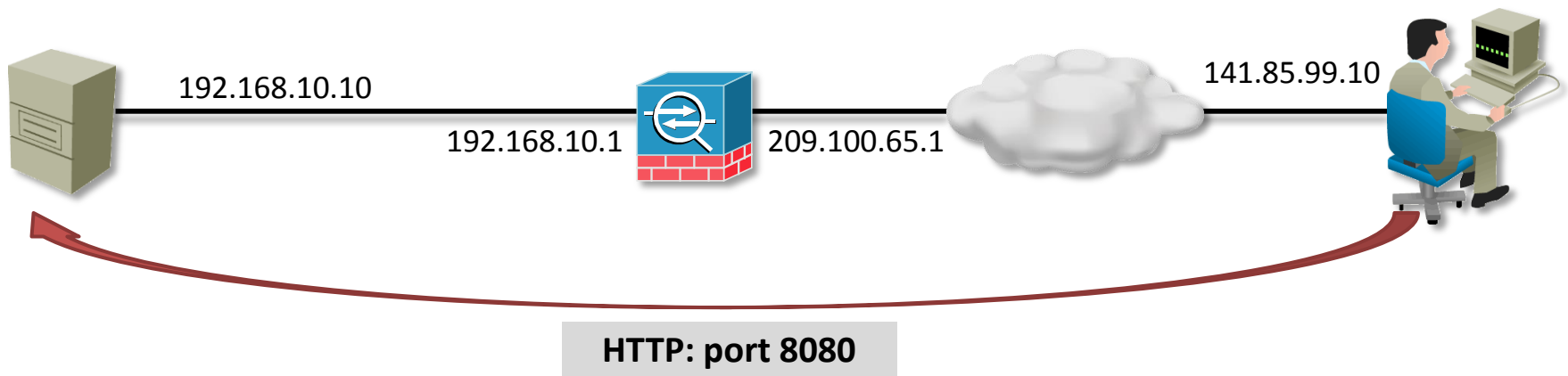
Anti {virus |
spam | spyware}



Granular
connection
limiting

Application Inspection(1)

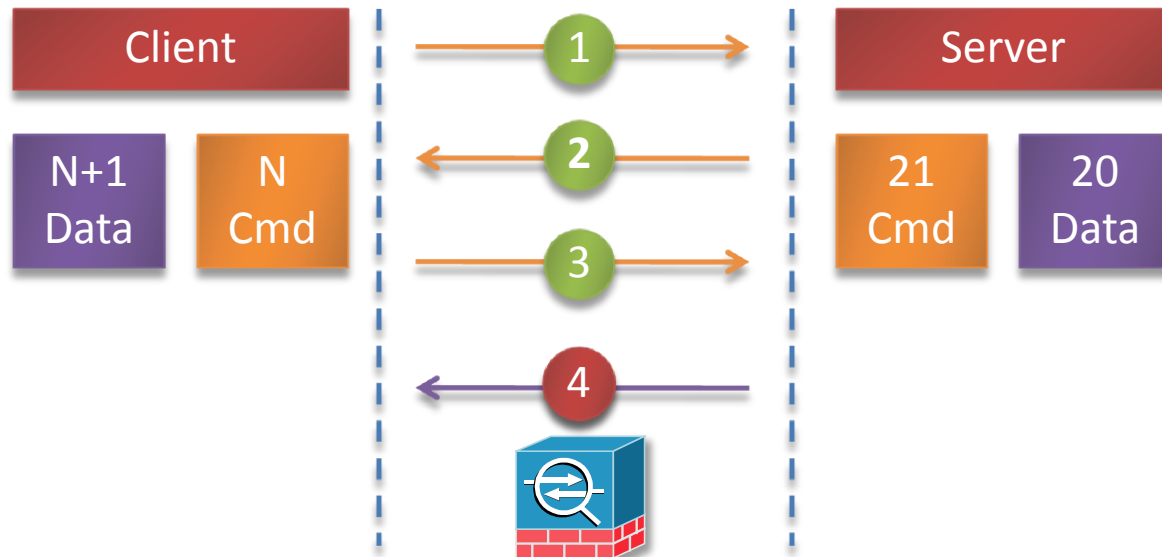
- ▶ De ce este nevoie de application inspection?
 - ❑ **(1)** Scenarii în care se rulează aplicații pe porturi ne-standard
- ▶ În mod implicit orice firewall identifică aplicația după portul destinație well-known
- ▶ Ex: dacă HTTP rulează pe portul 8080, orice firewall va face în mod implicit *drop*



Application Inspection(2)

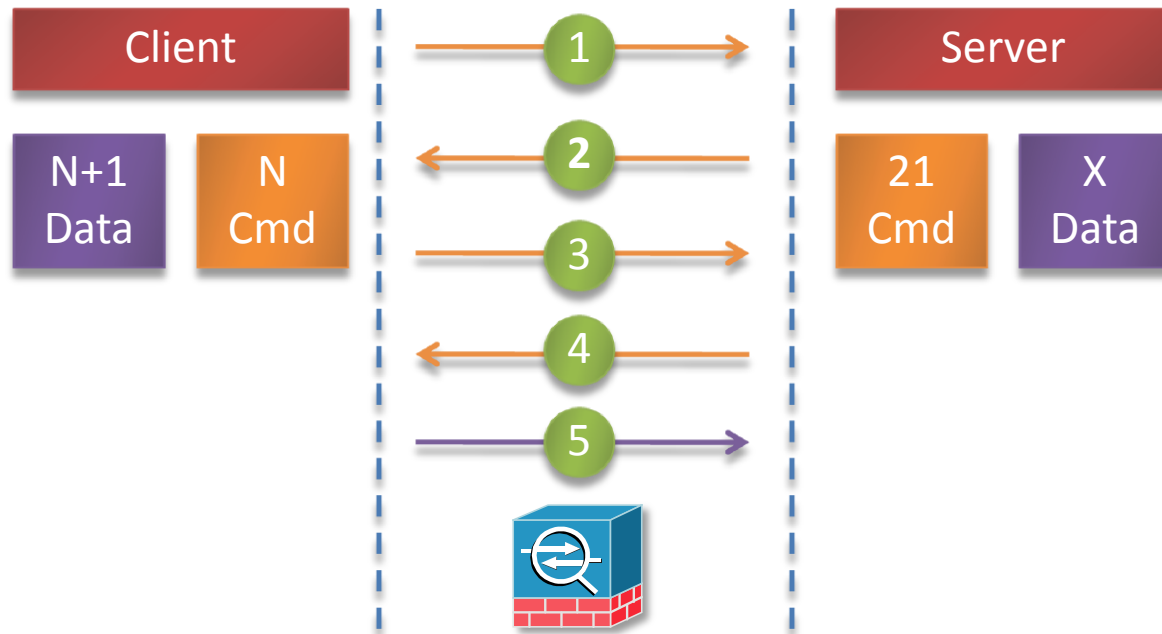
- ▶ De ce este nevoie de application inspection?
 - ❑ **(2)** Aplicații ce au nevoie să deschidă porturi în mod dinamic pentru a funcționa
 - ❑ Porturile deschise sunt negociate de aplicație peste canalul de control
 - ❑ Exemplu: Active FTP, aplicații multimedia, VoIP
- ▶ Studiu de caz:
 - ❑ Active FTP vs. Passive FTP

Active FTP



- ▶ 1. Clientul inițiază o conexiune către portul 21 al serverului de pe un port sursă N, ales aleator > 1023 .
- ▶ 2. Serverul răspunde cu ACK de pe portul sursă 21 către portul N al clientului.
- ▶ 3. Client transmite comanda "*PORT N+1*" peste canalul de control și își deschide portul N+1. Comanda PORT indica serverului care este portul clientului deschis pentru a primi date.
- ▶ 4. Serverul încearcă să realizeze o conexiune de pe portul 20 către portul N+1 al clientului.
- ▶ 5. Firewall-ul blochează conexiunea 4 din cauză că nu găsește obiectul de stare.

Passive FTP



- ▶ 1. Clientul inițiază o conexiune către portul 21 al serverului de pe un port sursă N, ales aleator > 1023 . Clientul deschide portul N+1 pentru date.
- ▶ 2. Serverul răspunde cu ACK de pe portul sursă 21 către portul N al clientului.
- ▶ 3. Clientul trimite comanda *PASV* către server.
- ▶ 4. Serverul își deschide un port X(aleator) > 1024 pentru conexiunea de date și trimite clientului comanda *Port X*.
- ▶ 5. Clientul inițiază conexiunea de date de pe portul N+1 către portul X al serverului.

Active FTP vs Passive FTP

► Concluzii:

- ❑ Active FTP nu funcționează dacă clientul este în spatele unui firewall
 - Din cauza inspecției stateful
 - Din cauza NAT
- ❑ Passive FTP ar trebui să funcționeze mereu
 - Cel puțin dacă administratorul a configurat serverul FTP în DMZ (e.g. configurat politici de acces din exterior)

► De ce dorim să folosim Active FTP?

- ❑ Pentru că deschide mai puțini socketi pe server:
<http://www.faqs.org/rfcs/rfc1579.html>

► Cu ajutorul Application Inspection:

- ❑ Firewall-ul poate citi comenzile trimise pe canalul de control FTP
- ❑ Când firewall-ul vede comanda PORT N, deschide portul N+1 pentru IP-ul serverului, ca acesta să poată contacta clientul în Active FTP

Application Inspection(3)

- ▶ De ce este nevoie de application inspection?
 - ❑ **(3)** Aplicații care fac embed la adrese IP în canalul de control și intră în conflict cu NAT
 - ❑ Adresa IP de la nivelul 3 (trecută prin NAT) nu coincide cu adresa primită la nivel aplicație
 - ❑ Aplicația încearcă să deschidă socketi către adresa IP privată și nu reușește
- ▶ Application Inspection to the rescue!
 - ❑ Firewall-ul inspectează IP-ul din canalul de control și îl înlocuiește cu cel din tabela xlate

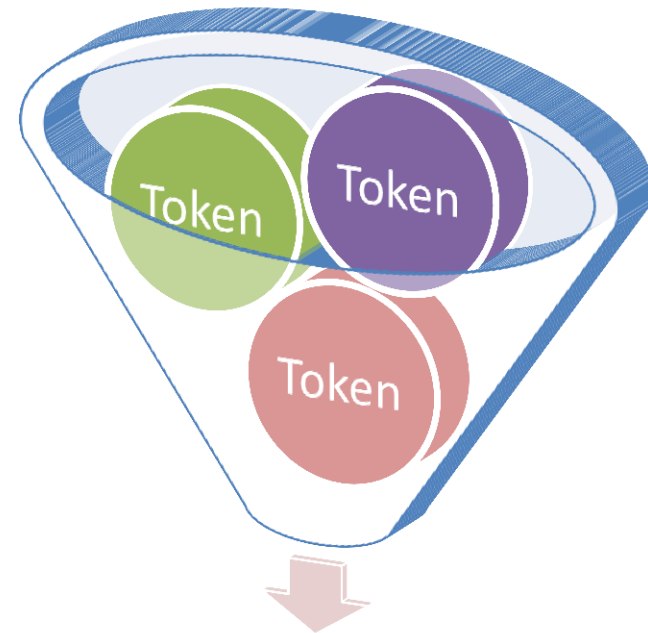


Traffic shaping vs Traffic policing

- ▶ Conform rate
 - ❑ Rata de trafic la care se așteaptă să fie transmis trafic printr-o interfață, într-o anumită direcție, conform SLA-ului
- ▶ Exceeding rate
 - ❑ Rata de trafic peste care se consideră că traficul nu mai este în conformitate cu SLA-ul sau poate congestiona interfața
- ▶ TS și TP sunt tehnici de a trata trafic ce depășește una dintre cele două limite de mai sus
- ▶ Traffic-shaping alocă buffere interne în RAM pentru a stoca traficul în exces și a îl transmite mai târziu
- ▶ Traffic-policing poate:
 - ❑ Face drop traficului în exces
 - ❑ Re-marca traficul cu o prioritate mai mică la nivel 2 sau 3

How it works

- ▶ Majoritatea implementărilor folosesc **token-uri** și un **token-bucket**
- ▶ Un **token** reprezintă permisiunea de a putea trimite un număr de X biți în rețea
- ▶ **Token-bucket**-ul grupează token-urile firewall-ului și definește de fapt conform rate-ul și excess-rate-ul funcție de numărul de token-uri
- ▶ Când un pachet trebuie trimis se verifică token-bucket-ul pentru a vedea dacă există îndeajuns de multe token-uri pentru a îl trimite
- ▶ Odată trimis, pachetul scoate un număr de token-uri din token-bucket
- ▶ Dacă nu mai există token-uri, se poate aștepta adăugarea lor (traffic shaping) sau se poate face drop la pachetul care dorește token-uri (traffic policing)



Există îndeajuns de multe token-uri pentru transmiterea pachetului?

Să încercăm o analogie

- ▶ Avem o pușculiță



- ▶ În fiecare zi pușculița primește 1\$ (resursele sistemului)
- ▶ Dacă se consumă 1\$/zi, atunci se consumă la **committed rate**, pentru că nu se consumă mai mult decât se face
- ▶ Dacă se dorește consumarea a 3\$ pe o înghețată, dar în pușculiță există doar 2\$, există 2 posibilități:
 - ❑ Se așteaptă o zi mai însorită și se „ține minte” dorința de a cumpăra înghețată (**traffic shaping**)
 - ❑ Se face drop la dorința de a cumpăra înghețată (**traffic policing**)
- ▶ Dacă se economisesc \$, se poate cumpăra ceva mai scump într-o anumită zi – **excess rate**

Back to QoS – traffic policing

- ▶ Nu consumă memorie pe firewall (nu are nevoie de buffere)
- ▶ Se poate aplică atât inbound cât și outbound pe o interfață
- ▶ Produce multe retransmisii TCP
 - ❑ Ar trebui folosit pe o interfață de viteză mare
- ▶ Nu produce jitter sau latență pachetelor în conform rate
- ▶ Acțiunile posibile sunt **drop** și **re-marcare** pachetului
- ▶ Un scenariu posibil ar fi:
 - ❑ Pachetele peste conform rate să fie re-marcate cu o prioritate mai mică
 - ❑ Pachetele peste excess rate să fie dropped

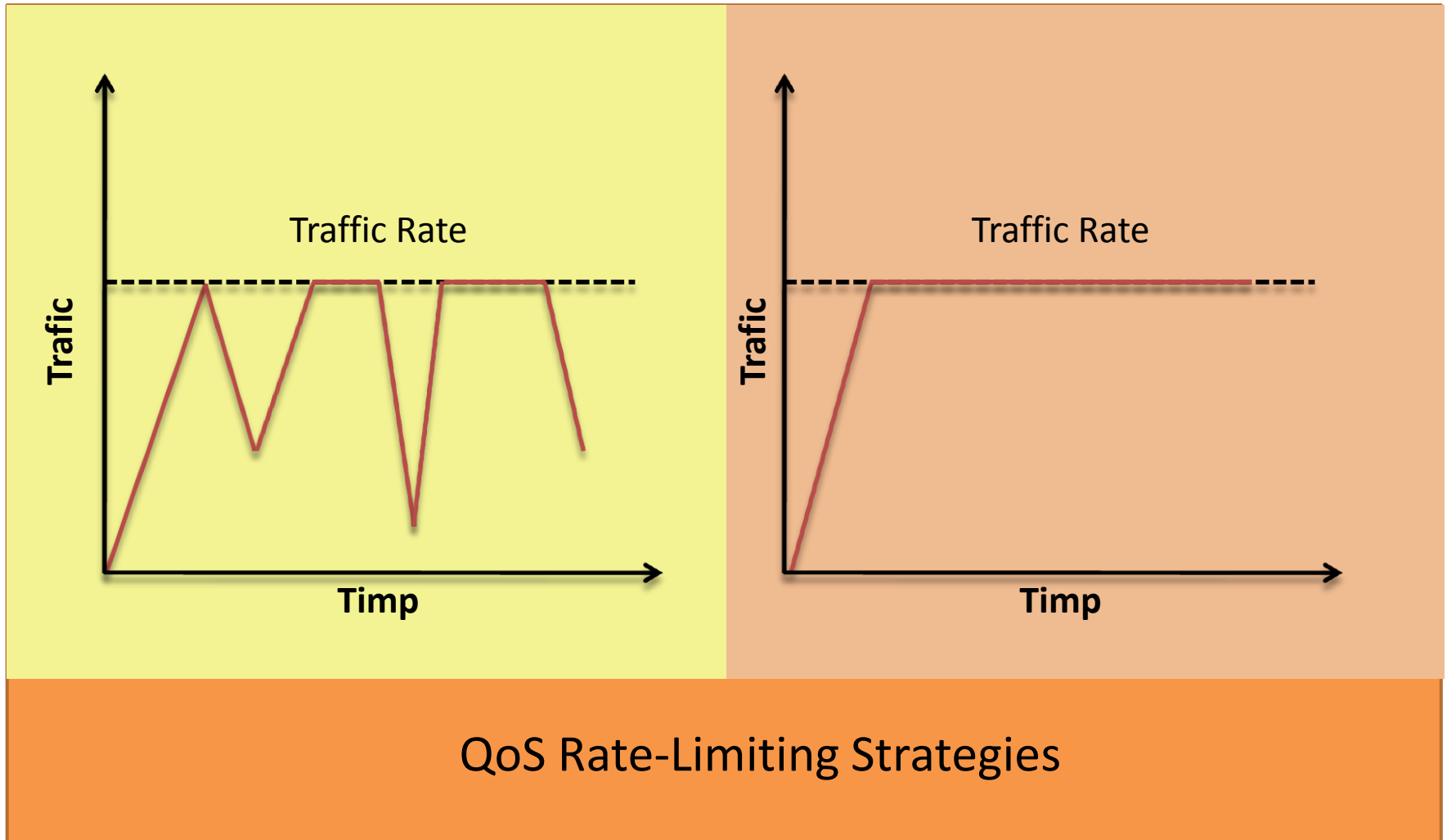


Back to QoS – traffic shaping

- ▶ Folosește memorie pentru a face buffering
- ▶ Nu cauzează retransmiteri TCP
 - ❑ Poate fi folosit pe interfețe de viteză mică
- ▶ Se poate aplica decât outbound
- ▶ Poate cauza jitter din cauza normalizării traficului
- ▶ Nu poate re-eticheta trafic
- ▶ Foarte folosit în Frame Relay unde se poate coordona cu biții de marcarea a congestiei din protocolul WAN



Traffic policing vs. traffic shaping





ASA – Modular Policy Framework

Modular Policy Framework

► Ce este MPF?

- ❑ Un set de structuri și comenzi în ASA OS
- ❑ Un mod de a gândi legăturile dintre multiple concepte teoretice și aplicarea lor

► Ce oferă MPF?

- ❑ Posibilitatea de a configura și controla cu aceleași structuri (comenzi):
 - Application Inspection
 - IPS (AIP-SSM)
 - Anti {virus | spam | spyware} (CSC –SSM)
 - Setarea limitelor pentru conexiuni
 - Traffic policing

MPF – Structuri de comenzi

► MPF este definită prin 3 structuri de bază

❑ Class-map

- Folosite pentru a identifica traffic-flow prin diferite moduri
- Există **class-map** generice care identifică la nivel 3 și 4 și **class-map** de inspecție care realizează identificare la nivel 7

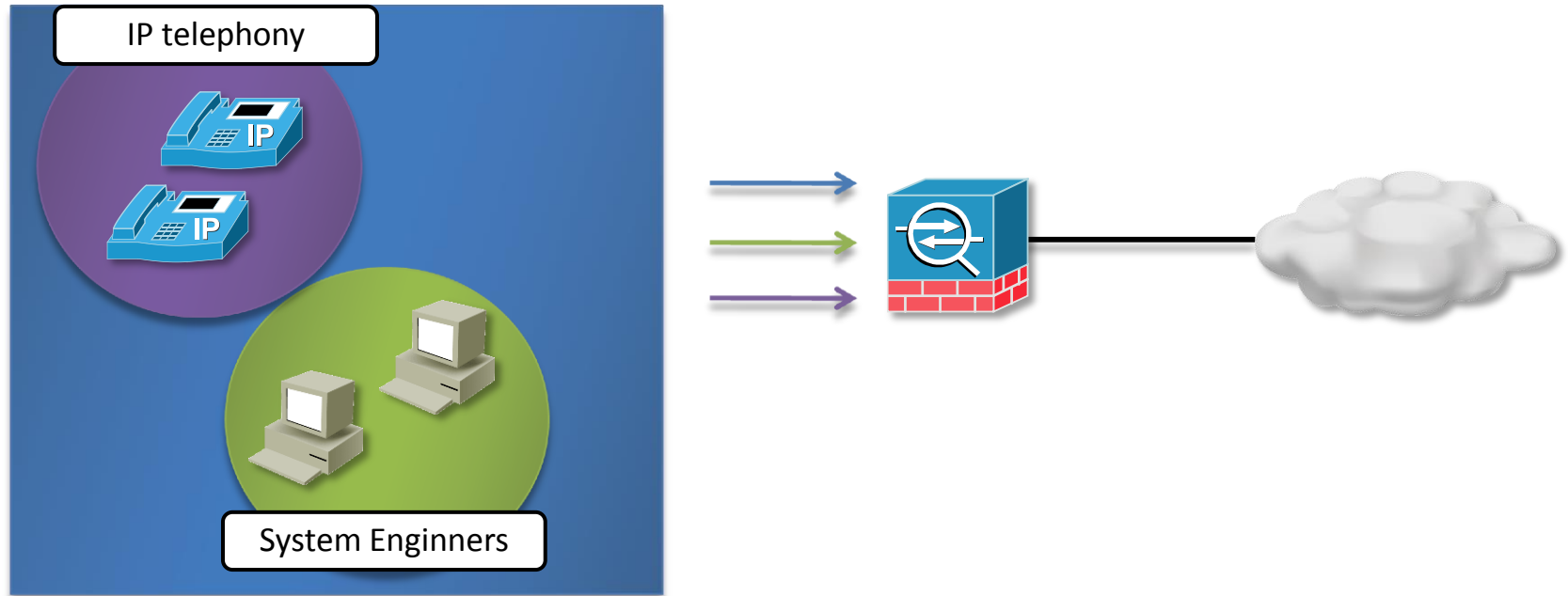
❑ Policy-map

- Folosite pentru a asocia una sau mai multe acțiuni pentru traficul identificat într-un class-map
- Există **policy-map** generice care aplică acțiunile standard (inspect, police, set connection etc) și **policy-map** de inspecție ce analizează în detaliu câmpurile și mesajele unor anumite protocoale de nivel aplicație

❑ Service-policy (comandă)

- Folosită pentru a aplica **policy-map** global sau pe anumite interfețe

MPF – Relația între structuri



MPF – Class-map (1)

► Definirea unui class-map (L3/L4)

```
ciscoasa(config)# class-map identify_by_L3_or_L4  
ciscoasa(config-cmap)# ?
```

MPF class-map configuration commands:

<code>description</code>	Specify class-map description
<code>exit</code>	Exit from MPF class-map configuration mode
<code>help</code>	Help for MPF class-map configuration commands
<code>match</code>	Configure classification criteria
<code>no</code>	Negate or set default values of a command
<code>rename</code>	Rename this class-map

► Comanda match este folosită pentru a identifica trafic

MPF – Class-map (2)

- ▶ După ce criterii poate comanda **match** identifica trafic?
 - ❑ **access-list**: folosește un ACL pentru identificare
 - ❑ **any**
 - ❑ **dscp**: match pe câmpul ToS conform standardului IETF DSCP
 - ❑ **precedence**: match pe câmpul ToS conform standardului IP Precedence
 - ❑ **tunnel-group**: match pe trafic trimis printr-un tunel. Acest criteriu poate fi folosit doar cu acțiuni ce țin de politici QoS
 - ❑ **flow ip destination-address**: identificarea adresei IP destinație înăuntrul unui tunnel-group. Se poate folosi doar împreună cu **tunnel-group**.
 - ❑ **port**: identifică un port TCP sau UDP
 - ❑ **default-inspection-traffic**: match pe o serie de protocoale ce rulează peste porturile configurate standard

MPF – Class-map (3)

- ▶ Un class-map suportă o singură comandă match
 - ❑ excepția o reprezintă parametrii **tunnel-group** și **default-inspection-traffic** care oferă posibilitatea de a da încă o comandă match
 - ❑ când există 2 comenzi match, se face **ȘI** logic între ele
- ▶ În mod implicit este activat class-map-ul inspection_default

```
ciscoasa# sh run
....
class-map inspection_default
  match default-inspection-traffic
....
```

MPF – Class-map (4)

► Ce reprezintă **default-inspection-traffic**?

```
ciscoasa(config-cmap)# match ?
```

```
mpf-class-map mode commands/options:
```

access-list	Match an Access List
any	Match any packet
default-inspection-traffic	Match default inspection traffic:
ctiqbe----tcp--2748	dns-----udp--53
ftp-----tcp--21	gtp-----udp--2123,3386
h323-h225-tcp--1720	h323-ras--udp--1718-1719
http-----tcp--80	icmp-----icmp
ils-----tcp--389	mgcp-----udp--2427,2727
netbios---udp--137-138	radius-acct---udp--1646
rpc-----udp--111	rsh-----tcp--514
rtsp-----tcp--554	sip-----tcp--5060
sip-----udp--5060	skinny----tcp--2000
smtp-----tcp--25	sqlnet----tcp--1521
tftp-----udp--69	waas-----tcp--1-65535

MPF – Policy-map

- ▶ Policy-map-ul determină acțiunea pe care firewall-ul să o ia
- ▶ Pasul 1: se dă un nume policy-map-ului
- ▶ Pasul 2: se asociază un class-map
- ▶ Pasul 3: se aplică o acțiune (sau mai multe acțiuni) traficului identificat

```
ciscoasa(config)# policy-map test_policy  
ciscoasa(config-pmap)# class major_protocols  
ciscoasa(config-pmap-c)# inspect ftp  
ciscoasa(config-pmap-c)# inspect icmp
```


MPF – Acțiuni posibile

- Un singur policy-map poate avea mai multe acțiuni și de mai multe tipuri

```
ciscoasa(config-pmap-c)# ?
```

MPF policy-map class configuration commands:

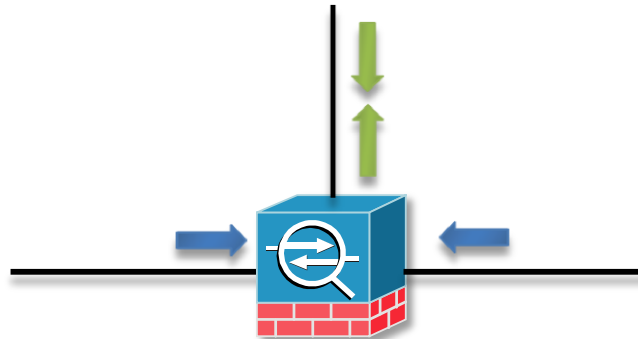
csc	Content Security and Control service module
exit	Exit from MPF class action configuration mode
flow-export	Configure filters for NetFlow events
help	Help for MPF policy-map class/match submode commands
inspect	Protocol inspection services
ips	Intrusion prevention services
no	Negate or set default values of a command
police	Rate limit traffic for this class
priority	Strict scheduling priority for this class
quit	Exit from MPF class action configuration mode
service-policy	Configure QoS Service Policy
set	Set connection values
shape	Traffic Shaping

MPF – Exemple de acțiuni

```
ciscoasa# sh run
.....
!
class-map http_map
  match port tcp eq www
!
policy-map http_policy
  class http_map
    inspect http
    police input 1000000
    set connection conn-max 1000 per-client-embryonic-max 50
!
.....
```

MPF – Aplicarea unei politici

- ▶ O politică se poate aplica pe interfață sau la nivel global
- ▶ **Politica la nivel global** afectează tot traficul ce trece prin orice interfață a ASA, dar doar în direcția **ingress**
- ▶ **Politica la nivel de interfață** afectează tot traficul ce trece prin acea interfață, **ingress** și **egress**



```
# aplicare la nivel global
```

```
ciscoasa(config)# service-policy inspect_http global
```

```
# aplicare pe interfață
```

```
ciscoasa(config)# service-policy inspect_http interface inside
```

MPF – Procesarea unei politici

- ▶ Dacă acțiunea politicii este diferită, pachetele pot face match de mai multe ori într-un policy-map, cât timp class-map-urile identifică acel pachet

```
class-map example
  match port tcp eq www
policy-map http_policy
  class example
    police input 1000000
    set connection conn-max 1000 per-client-embryonic-max 50
  class inspection_default
    inspect http
```

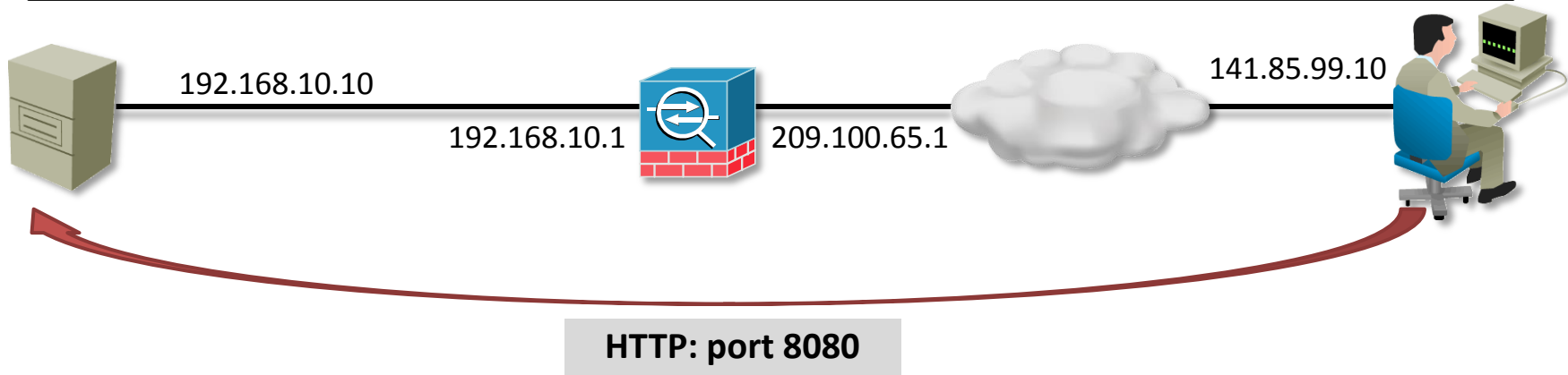
- ▶ Dacă acțiunea este aceeași, se face match pe un singur class-map



ASA – Advanced MPF

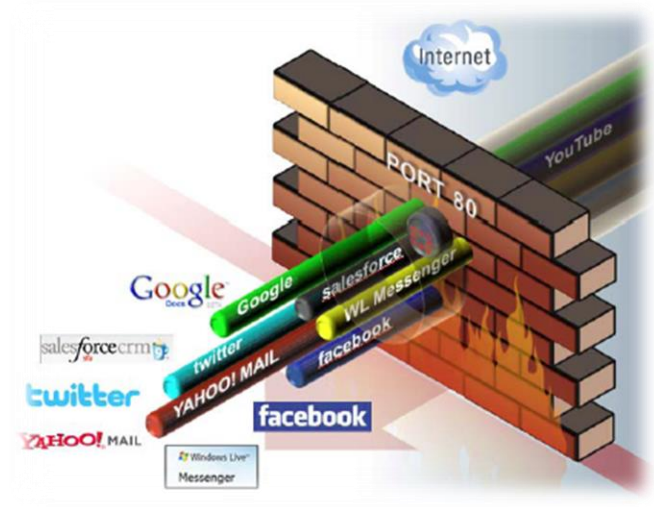
Inspecția pe un port ne-standard

```
asa1(config)# class-map 8080_INSPECT_TRAFFIC
asa1(config-cmap)# match port tcp eq 8080
asa1(config-cmap)# exit
asa1(config)# policy-map global_policy
asa1(config-pmap)# class 8080_inspect_traffic
asa1(config-pmap-c)# inspect http
asa1(config-pmap-c)# exit
asa1(config-pmap)# exit
```



Advanced Application Inspection

- ▶ Permite controlul granular asupra aplicațiilor
 - ❑ Blocarea fișierelor cu extensia .exe
 - ❑ Blocarea Kazaa și alte protocoale p2p
 - ❑ Setarea de limite pe câmpuri interne ale protocolului (ex: lungimea unui URL)
 - ❑ Protejarea serviciilor web prin validarea XML
 - ❑ Resetarea unei conexiuni TCP care în câmpul de date conține un anumit string



Configurarea advanced protocol inspection

- ▶ Se folosesc structuri adiționale de tip *inspect*

- ❑ (Optional) class-map type inspect – folosite pentru a face match după criterii specifice unei anumite aplicații

- ❑ Policy-map type inspect – folosite pentru a defini acțiuni speciale pentru inspecția unei anumite aplicații

- ❑ Class-map (L3/L4) – folosite pentru a identifica trafic la nivel 3 și 4

- ❑ Policy-map (L3/L4) – folosite pentru a aplica acțiuni traficului identificat de un class-map L3/L4

- ❑ Service-policy – folosite pentru a aplica o politică la nivel de interfață sau global

- ▶ Un policy-map type inspect este aplicat într-un policy-map L3/L4

Advanced MPF – Exemplu

- ▶ Exemplu: drop la conexiuni HTTP ce folosesc metoda POST

```
asa1(config)# class-map type inspect http POST_METHOD
asa1(config-cmap)# match request method post
asa1(config-cmap)# exit
asa1(config)# policy-map type inspect http MY_HTTP_MAP
asa1(config-pmap)# class POST_METHOD
asa1(config-pmap-c)# drop-connection log
asa1(config-pmap-c)# exit
asa1(config-pmap)# exit
asa1(config)# policy-map WEB_POLICY
asa1(config-pmap)# class inspection_default
asa1(config-pmap-c)# inspect http MY_HTTP_MAP
asa1(config-pmap-c)# exit
asa1(config-pmap)# exit
asa1(config)# service-policy WEB_POLICY interface inside
```

Expresii regulate

- ▶ Practic oferă posibilități infinite de identificare a traficului
- ▶ Folosește obiecte de tip “regex” care se adaugă la un class-map type regex
- ▶ Class-map-ul de tip regex poate fi folosit într-un class-map type inspect

```
asa1(config)# regex P2P_custom Kazaa2.1_custom
asa1(config)# regex ANYGIF ".*\.[Gg][Ii][Ff]"
asa1# test regex file.gif ".*\.[Gg][Ii][Ff]"
INFO: Regular expression match succeeded.
asa1(config)#class-map type regex match-any NEW_P2P
asa1(config-cmap)#match regex P2P_custom
asa1(config-cmap)#match regex ANYGIF
```

Exemplu – expresii regulate

```
asa1(config)#regex COMPANY_CONFIDENTIAL
    "[Cc][Oo][Nn][Ff][Ii][Dd][Ee][Nn][Tt][Ii][Aa][Ll]"
asa1(config)#regex CLASSIFIED "[Cc][Ll][Aa][Ss][Ss][Ii][Ff][Ii][Ee][Dd]"
asa1(config)#class-map type regex match-any CLASSIFIED_DOCUMENTS
asa1(config-cmap)#match regex COMPANY_CONFIDENTIAL
asa1(config-cmap)#match regex CLASSIFIED
. . .
asa1(config)#class-map type inspect http match-all CLASSIFIED_TRAFFIC
asa1(config-cmap)#match request header user-agent regex class CLASSIFIED_DOCUMENTS
asa1(config-cmap)#match request method post
. . .
asa1(config)#policy-map type inspect http CONFIDENTIAL_POLICY
asa1(config-pmap)#class CLASSIFIED_TRAFFIC
asa1(config-pmap-c)#drop-connection log
. . .
asa1(config)#policy-map DOCUMENT_SECURITY
asa1(config-pmap)#class inspection_default
asa1(config-pmap-c)#inspect http CONFIDENTIAL_POLICY
. . .
asa1(config)#service-policy DOCUMENT_SECURITY interface inside
```



FortiGate – Application Control

Session helpers

- ▶ Folosiți pentru a configura inspecția la nivel aplicație pentru diferite protocoale
- ▶ Se pot configura doar din linie de comandă

```
Fortigate51B # show system session-helper
config system session-helper
    edit 1
        set name pptp
        set port 1723
        set protocol 6
    next
    .....
    edit 9
        set name ftp
        set port 21
        set protocol 6
    next
```

Session helpers

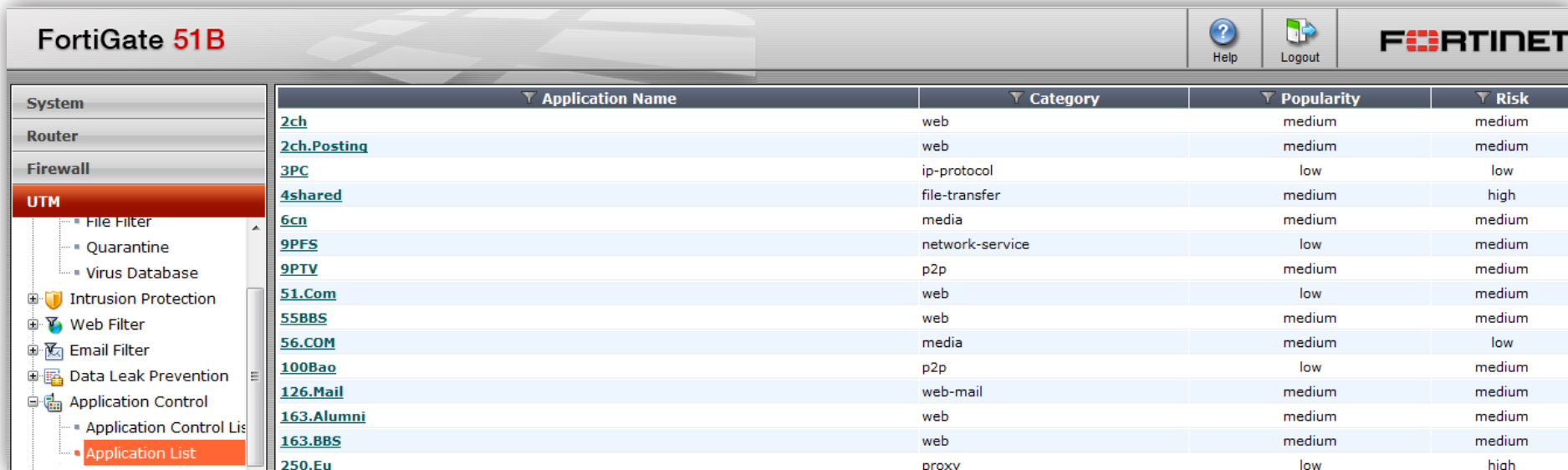
- ▶ Cu ajutorul session helpers se poate modifica portul pe care se realizează inspecția pentru o aplicație

```
Fortigate51B # config system session-helper
Fortigate51B (session-helper) # edit 1
Fortigate51B (1) # set
*name          helper name
*port          protocol port
*protocol      protocol number
Fortigate51B (1) # set name ftp
Fortigate51B (1) # set protocol 6
Fortigate51B (1) # set port 55555
```

- ▶ Câmpul “protocol” este numărul de protocol din antetul IP

Application Control

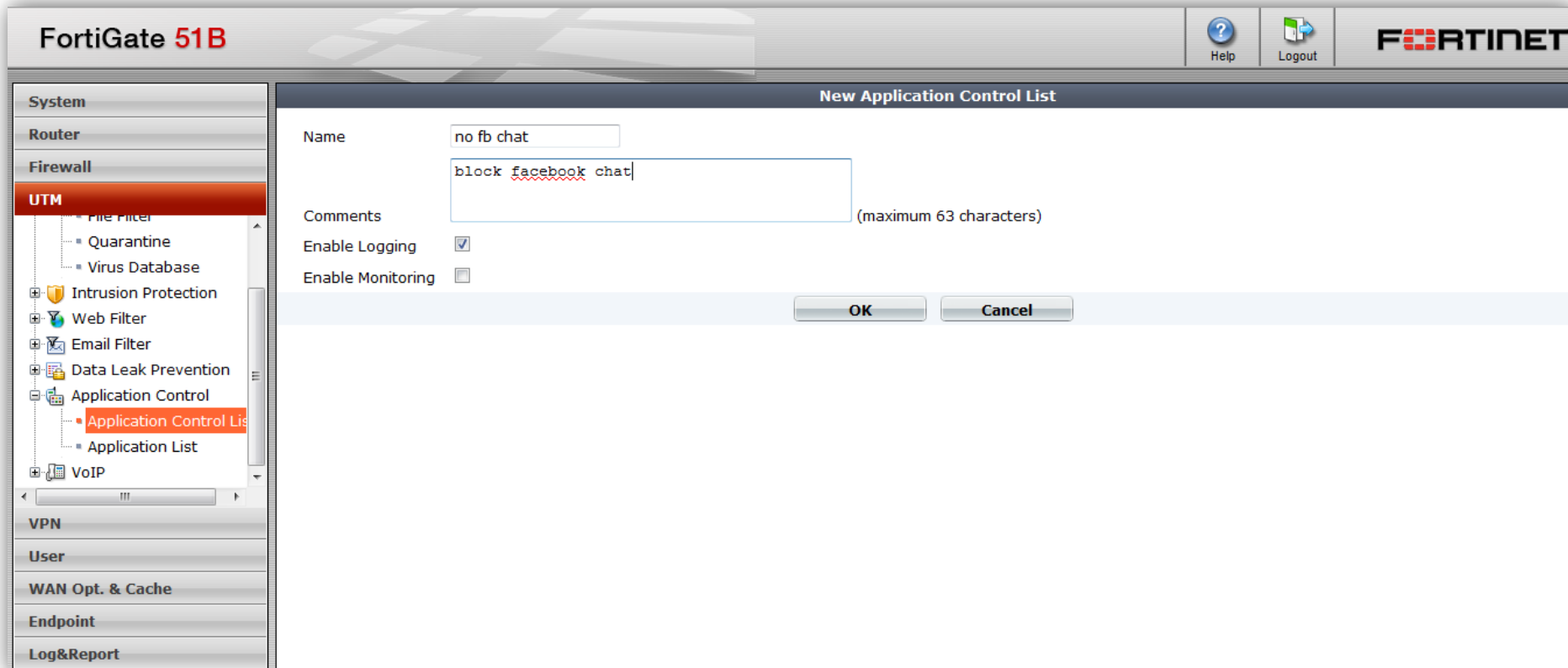
- ▶ Permite controlul granular al aplicațiilor
- ▶ FortiGate permite fine-tuning pe 1500+ aplicații
- ▶ Asemănător ASA, FortiGate are cod intern scris pentru a analiza nivelul 7 și a manipula mesajele din canalul de control



Application Name	Category	Popularity	Risk
2ch	web	medium	medium
2ch.Posting	web	medium	medium
3PC	ip-protocol	low	low
4shared	file-transfer	medium	high
6cn	media	medium	medium
9PFS	network-service	low	medium
9PTV	p2p	medium	medium
51.Com	web	low	medium
55BBS	web	medium	medium
56.COM	media	medium	low
1008ao	p2p	low	medium
126.Mail	web-mail	medium	medium
163.Alumni	web	medium	medium
163.BBS	web	medium	medium
250.Eu	proxy	low	high

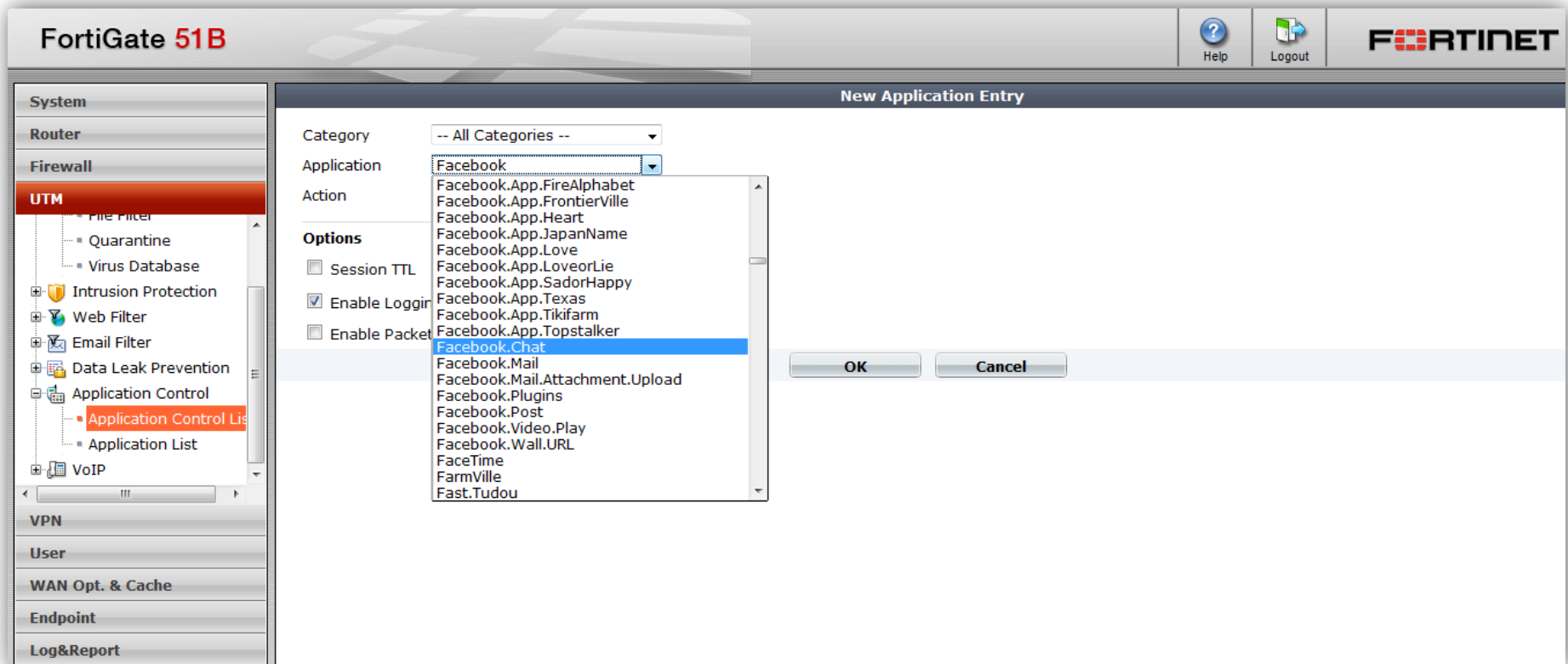
Application Control – Exemplu

- ▶ Ne propunem să blocăm facebook chat
- ▶ Pasul 1: definirea politicii de application control



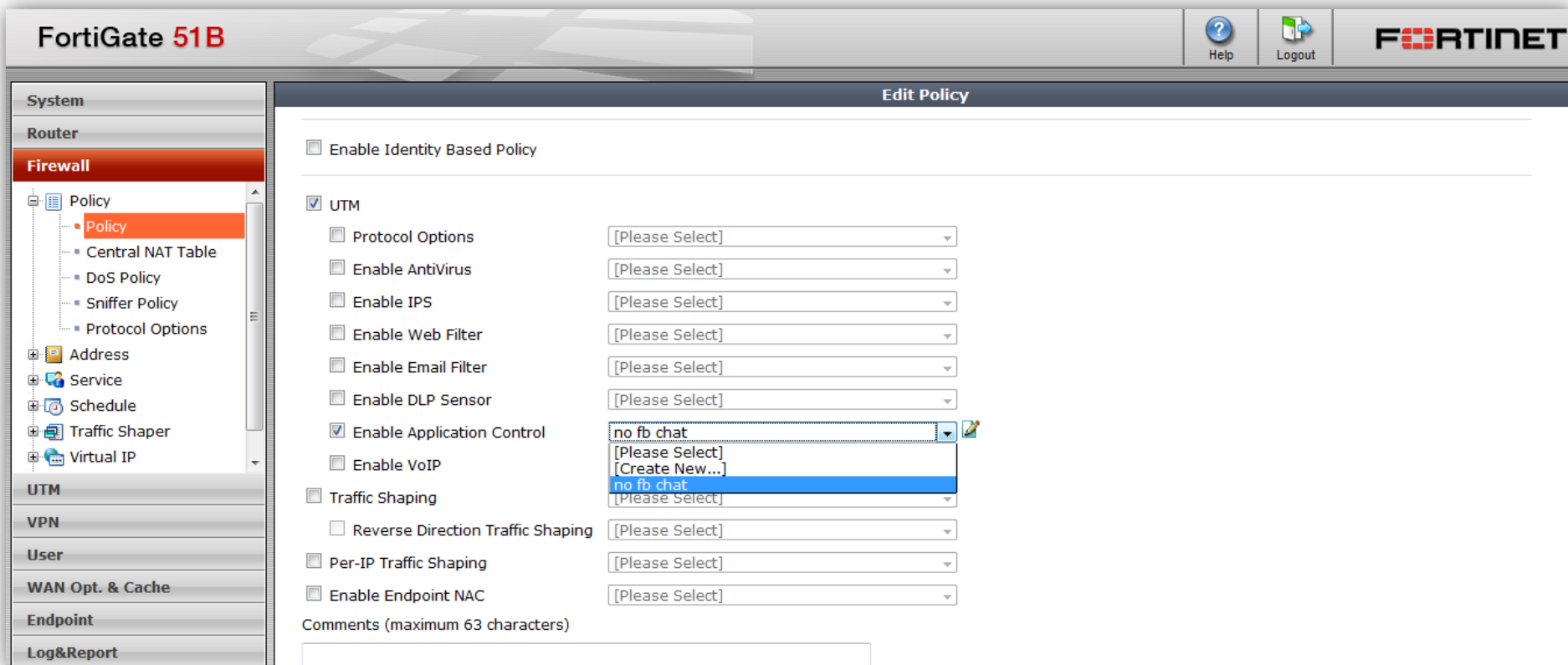
Application Control – Exemplu

- Pasul 2: creare unei intrări noi în politică de App Control și alegerea aplicației dorite



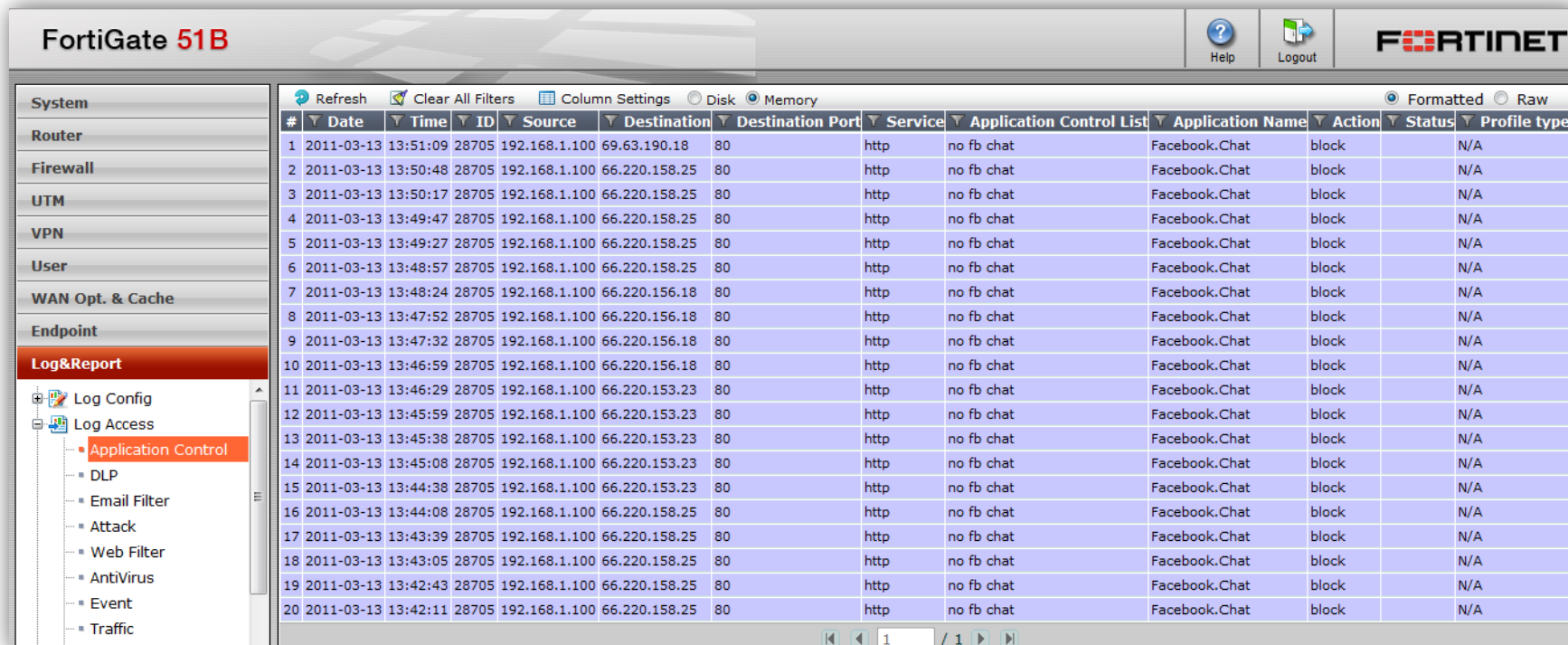
Application Control – Exemplu

- ▶ Pasul 3: aplicarea în politica de firewall
- ▶ Application Control e o funcționalitate de UTM



Application Control – Exemplu

- ▶ FortiGate oferă logging sau monitorizare în timp real pentru traficul afectat de Application Control



FortiGate 51B

Help Logout FORTINET

System Router Firewall UTM VPN User WAN Opt. & Cache Endpoint

Log & Report

- Log Config
- Log Access
 - Application Control
 - DLP
 - Email Filter
 - Attack
 - Web Filter
 - AntiVirus
 - Event
 - Traffic

Refresh Clear All Filters Column Settings Disk Memory Formatted Raw

#	Date	Time	ID	Source	Destination	Destination Port	Service	Application Control List	Application Name	Action	Status	Profile type
1	2011-03-13	13:51:09	28705	192.168.1.100	69.63.190.18	80	http	no fb chat	Facebook.Chat	block		N/A
2	2011-03-13	13:50:48	28705	192.168.1.100	66.220.158.25	80	http	no fb chat	Facebook.Chat	block		N/A
3	2011-03-13	13:50:17	28705	192.168.1.100	66.220.158.25	80	http	no fb chat	Facebook.Chat	block		N/A
4	2011-03-13	13:49:47	28705	192.168.1.100	66.220.158.25	80	http	no fb chat	Facebook.Chat	block		N/A
5	2011-03-13	13:49:27	28705	192.168.1.100	66.220.158.25	80	http	no fb chat	Facebook.Chat	block		N/A
6	2011-03-13	13:48:57	28705	192.168.1.100	66.220.158.25	80	http	no fb chat	Facebook.Chat	block		N/A
7	2011-03-13	13:48:24	28705	192.168.1.100	66.220.156.18	80	http	no fb chat	Facebook.Chat	block		N/A
8	2011-03-13	13:47:52	28705	192.168.1.100	66.220.156.18	80	http	no fb chat	Facebook.Chat	block		N/A
9	2011-03-13	13:47:32	28705	192.168.1.100	66.220.156.18	80	http	no fb chat	Facebook.Chat	block		N/A
10	2011-03-13	13:46:59	28705	192.168.1.100	66.220.156.18	80	http	no fb chat	Facebook.Chat	block		N/A
11	2011-03-13	13:46:29	28705	192.168.1.100	66.220.153.23	80	http	no fb chat	Facebook.Chat	block		N/A
12	2011-03-13	13:45:59	28705	192.168.1.100	66.220.153.23	80	http	no fb chat	Facebook.Chat	block		N/A
13	2011-03-13	13:45:38	28705	192.168.1.100	66.220.153.23	80	http	no fb chat	Facebook.Chat	block		N/A
14	2011-03-13	13:45:08	28705	192.168.1.100	66.220.153.23	80	http	no fb chat	Facebook.Chat	block		N/A
15	2011-03-13	13:44:38	28705	192.168.1.100	66.220.153.23	80	http	no fb chat	Facebook.Chat	block		N/A
16	2011-03-13	13:44:08	28705	192.168.1.100	66.220.158.25	80	http	no fb chat	Facebook.Chat	block		N/A
17	2011-03-13	13:43:39	28705	192.168.1.100	66.220.158.25	80	http	no fb chat	Facebook.Chat	block		N/A
18	2011-03-13	13:43:05	28705	192.168.1.100	66.220.158.25	80	http	no fb chat	Facebook.Chat	block		N/A
19	2011-03-13	13:42:43	28705	192.168.1.100	66.220.158.25	80	http	no fb chat	Facebook.Chat	block		N/A
20	2011-03-13	13:42:11	28705	192.168.1.100	66.220.158.25	80	http	no fb chat	Facebook.Chat	block		N/A

1 / 1

- ▶ Logging vs monitorizare
 - ❑ Logging se face implicit în RAM
 - ❑ Monitorizarea se face implicit într-o bază de date SQL salvată pe HDD

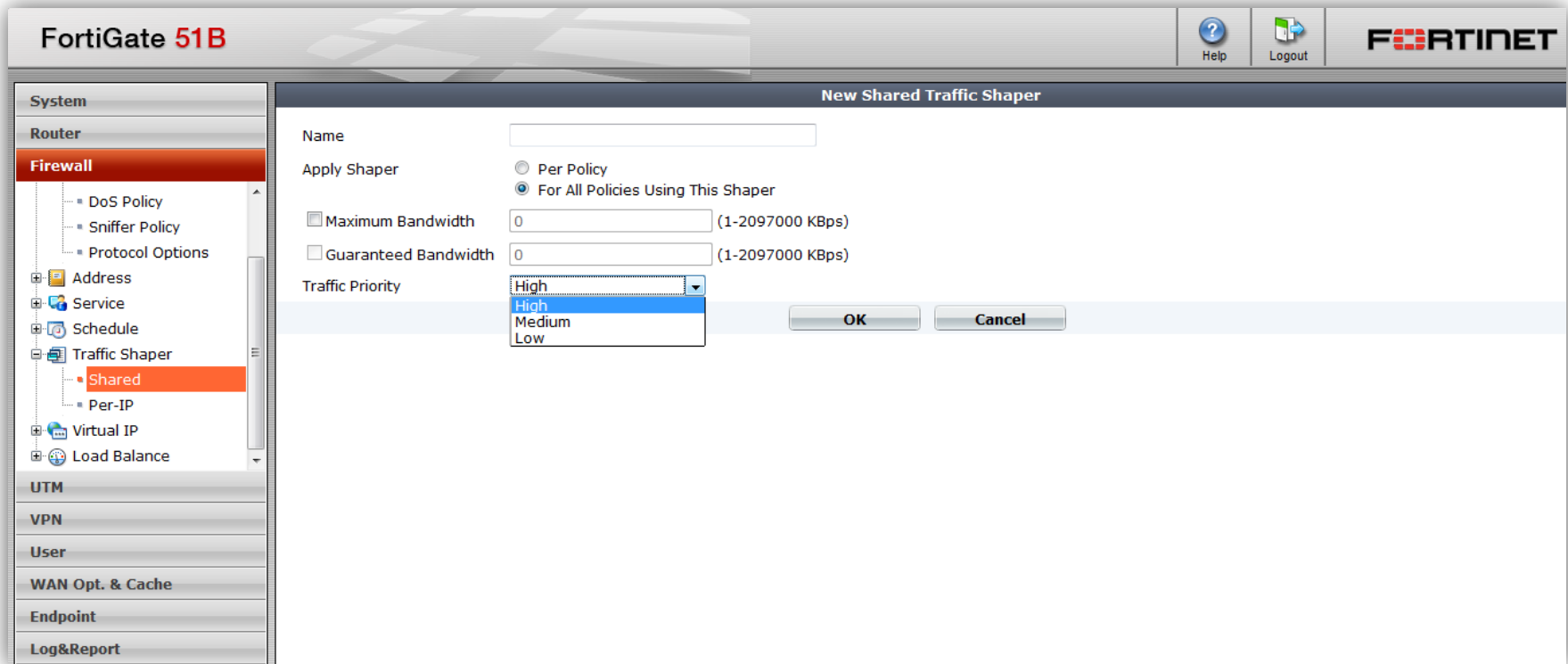
Traffic shaping

- ▶ Pe FortiGate, funcționalitatea de traffic shaping include:
 - ❑ Traffic policing
 - ❑ Traffic shaping
 - ❑ QoS prin cozi prioritare
- ▶ FortiGate suportă 3 tipuri de shaping:
 - ❑ Shared – suportă controlul lățimii de bandă la nivelul politicii de firewall
 - ❑ Per-IP – suportă controlul lățimii de bandă funcție de IP sursă
 - ❑ Application Control Shaping – suportă controlul lățimii de bandă folosind application inspection

Shared shapers

- ▶ Prin metoda **shared** se poate defini un shaper pentru:
 - ❑ O politică
 - ❑ Toate politicile
- ▶ Fiecare shaper trebuie să definească:
 - ❑ Maximum bandwidth – lățimea de bandă maximă ce poate fi folosită de traficul dintr-o anumită politică
 - ❑ Guaranteed bandwidth – lățimea de bandă garantată unei conexiuni
 - Atenție: această valoare trebuie să fie aleasă cu mult mai mică decât valoarea reală a lățimii de bandă a interfeței. Altfel există riscul de a rămâne fără bandă pentru trafic ce nu face match pe politică
 - ❑ Prioritarea traficului – definește cum este tratat traficul relativ la alte shapere
 - high/medium/low

Shared shapers



- ▶ **Atenție:** trebuie setat un shaper pentru toate politicile.
 - ❑ Orice politică fără shaper are în mod implicit prioritatea high și poate satura lățimea de bandă

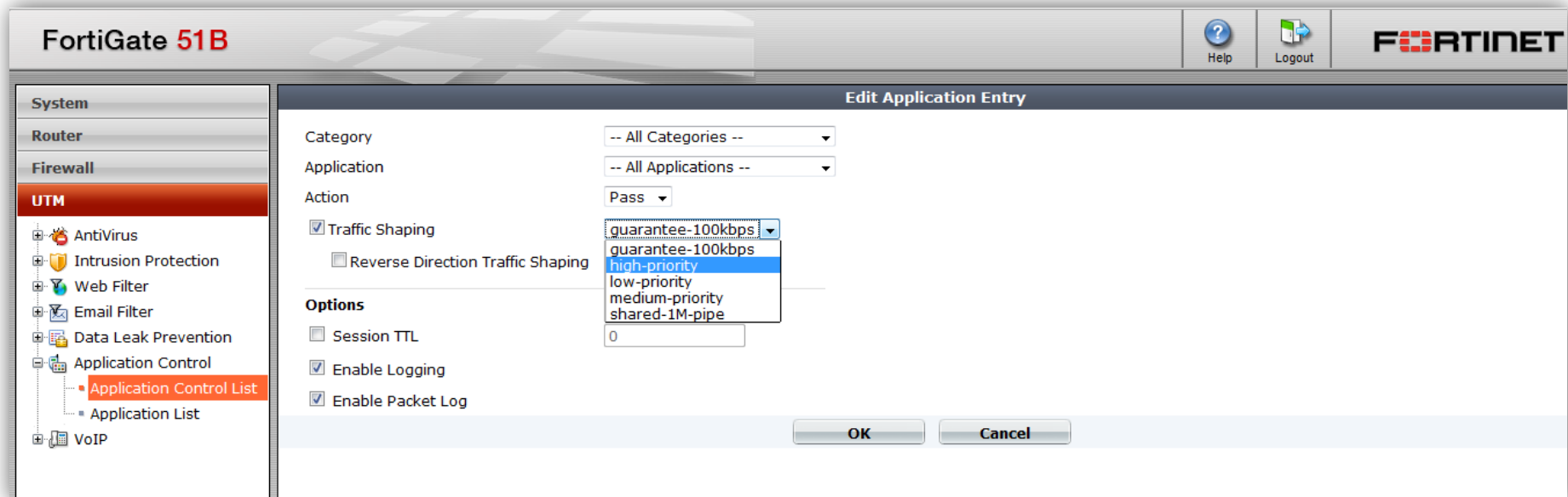
Per-IP shaper

- ▶ Definește un shaper ce se aplică tuturor IP-urilor dintr-o politică de firewall
- ▶ Diferența față de shared este că împiedică un utilizator să consume toată banda
 - ☐ Toți userii sunt egali
- ▶ Suportă definirea:
 - ☐ Lățimii de bandă maxime
 - ☐ Numărul maxim de conexiuni permise

The screenshot displays the FortiGate 51B web management interface. On the left, a sidebar menu shows the navigation tree: System > Router > Firewall > Traffic Shaper > Per-IP (highlighted). The main content area is titled 'New Per-IP Traffic Shaper'. It contains a 'Name' field, two checkboxes for 'Maximum Bandwidth' and 'Maximum Concurrent Connections' (both set to 0), and their respective units '(1-2097000 KBps)' and '(1-2097000)'. At the bottom right of the form are 'OK' and 'Cancel' buttons. The top of the interface shows the 'FortiGate 51B' logo, 'Help' and 'Logout' links, and the 'FORTINET' logo.

Application Control Shaper

- ▶ Permite definirea lăţimii de bandă şi priorităţii pentru o anumită aplicaţie



- ▶ Acest tip de shaper are prioritate în faţa oricărui alt shaper

Configuration converter

- ▶ Fortinet oferă un serviciu prin care se pot converti fișiere de configurare Cisco, Juniper, Checkpoint în fișiere de configurare Fortinet
- ▶ Sunt convertite feature-uri de policy, object, static route, NAT, VPN
- ▶ Cisco
 - ❑ Router: IOS 10.x,11.x,12.x
 - ❑ PIX/ASA: Pix 4.x, Pix 5.x, Pix 6.x, Pix 7.x, Pix 8.x
- ▶ <https://convert.fortinet.com/forticonverter/>

Configuration converter

FORTI
CONVERTER
V3.0 BETA



Support Platform

Platform	Version
Cisco router	IOS 10.x, IOS 11.x, IOS 12.x
Cisco PIX/ASA	Pix 4.x, Pix 5.x, Pix 6.x, Pix 7.x, Pix 8.x
Checkpoint	Smart Center, Provider-1 (excluding VPN-1 Edge, Safe@Office, SMP), with OS NG FP1 (4.0) to NGX R65 (6.5)
Juniper	ssg with OS 5.x

Support Feature

	Cisco router	Cisco PIX	Checkpoint	Juniper
Policy	✓	✓	✓	✓
Object	✓	✓	✓	✓
Static route	✓	✓	✗	✓
Service	✓	✓	✓	✓
NAT	✓	✓	✓	✓
VPN	✓	✓	✓	✓

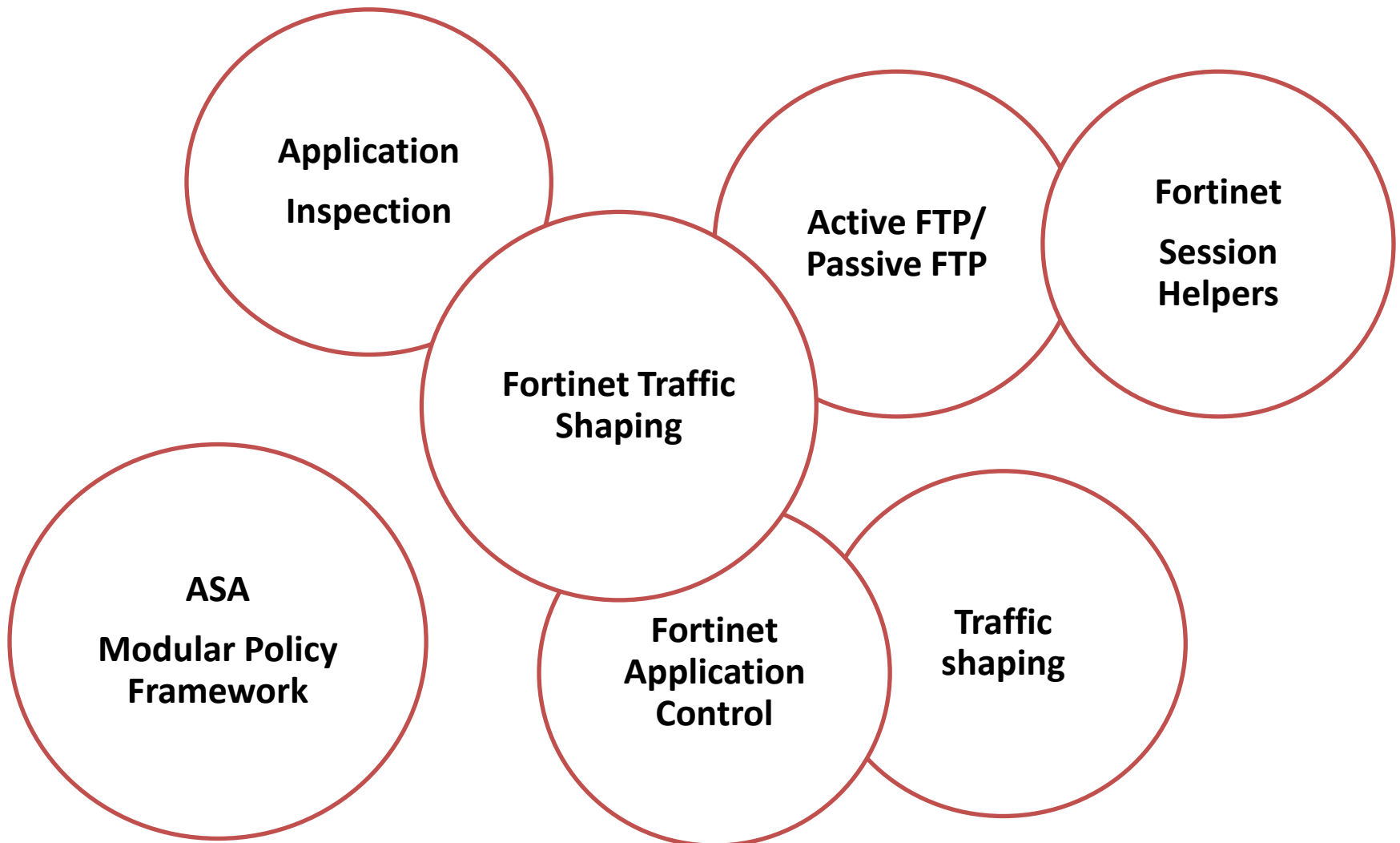
Known Issues

Platforms	Service object	Issue
Cisco router /FWSM /PIX	User DB	Will be available in version 3.1
Checkpoint	User DB	Will be available in version 3.1
Juniper	User DB	Will be available in version 3.1

START

FeedBack

Overview



Cursul viitor...

- ▶ ACL Object grouping



- ▶ Routing and switching

- ☐ Rute statice
- ☐ Protocoale de rutare dinamice
- ☐ PBR
- ☐ BGP
- ☐ VLAN-uri

