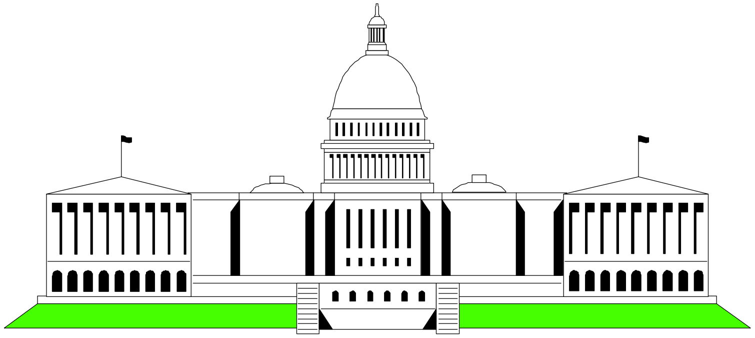
Installing OS/390 Firewall Technologies



Washington System Center

Mary Sweat
E - Mail: sweatm@us.ibm.com

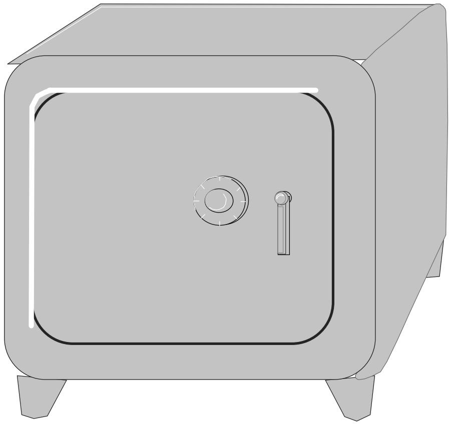
Agenda

- Guidelines for firewalls
- Security Considerations
- Firewall Configuration
- IP Filters
- Virtual Private Networks

Guidelines

- Define a policy of how your firewall will function
  - what type of traffic is allowed through the firewall and under what conditions
  - what functions will run under the firewall
    - what users/groups will be allowed access
- Disable everything when configuring the firewall
  - enable only those services defined in the security policy
  - everything not explicitly allowed is disabled
- Implement the same level of security for ALL gateways between the internal system and the Internet
- Log both successful and rejected access events
  - use daily admin procedures to analyze and react to the information from these logs

Security Considerations



- Isolate the firewall on its own system or logical partition
  - remove any services that are not required by the firewall
- Direct all incoming traffic (from the Internet) through the firewall stacks
  - allows Telnet or FTP applications to be active
  - subject applications to filtering rules defined in the firewall
- Ensure default passwords associated with program products are changed to non-trivial passwords
- Limit or disallow, when possible, amount of access from the Internet to the secure network
- Monitor log records stored in HFS, and ensure the HFS does not become full
  - records could be lost

Firewall Requirements for Implementation

- SYS1.PARMLIB updates
- SYS1.PROCLIB updates
- Security Requirements
- Hardware Cryptography
- TCP/IP Updates
- Logging
- Firewall stacks
- Configuration Files
- Adapters
- GUI Configuration

SYS1.PARMLIB Updates

- **BPXPRMxx**
  - MAXPROCSYS
  - MAXPROCUSER
  - MAXFILEPROC
  - MAXTHREADTASKS
  - MAXTHREADS
  - MAXSOCKETS
  - Define AF_UNIX and AF_INET file systems

```
NETWORK DOMAINNAME(AF_UNIX)
        DOMAINNUMBER(1)
        MAXSOCKETS(100)
        TYPE(UDS)
NETWORK DOMAINNAME(AF_INET)
        DOMAINNUMBER(2)
        MAXSOCKETS(n)
        TYPE(CINET)
```

- **PROGxx** - add SYS1.SICALMOD (APF authorizations)

- **LNKLSTxx** - add SYS1.SICALMOD

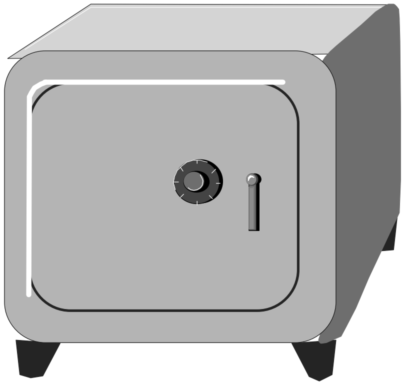
- **IKJTSOxx** - add AUTHPGMs (authorized commands and programs)

SYS1.PROCLIB

- Add the JCL for the FW daemons or concatenate the FW procedure library (SYS1.SICAPROC)
  - FWKERN
  - ICAPCFGS
  - ICAPKERN
  - ICAPPFTP
  - ICAPSLOG
  - ICAPSOCK
  - ICAPSTAK

Security Requirements

- Groups and User ID
  - add Firewall group
    - Example:
      ```
      au  fwgrp  SUP(SYS1)  OW(SYS1)  OMVS(GID(100))
      ```
  - add user FWKERN
    - Example:
      ```
      mkdir'/u/fwkern' mode(7,5,5)
      au fwkern  DFLTGRP(fwgrp)  auth(create) uacc(alter)
      password(xxxx) ow(sys1) omvs(home(/u/fwkern/)
      uid(0))
      ```
  - add firewall start up program as a started task
    - **Example:**
      ```
      setr raclist(started) refresh
      rdef started fwkern stdata(user(fwkern))
      ```

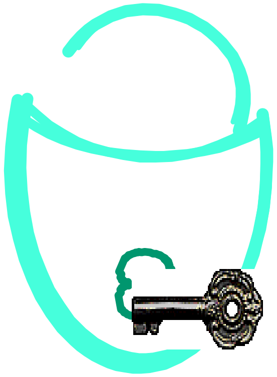


- Control the start of the firewall
  - Example**:**
    ```
    setr classact(facility)
    def  facility  fwkern.start.request  uacc(none)
    pe  fwkern.start.request  cl(facility)  id(fwkern) ac(update)
    ```

Grant Authority to Firewall Objects

- Define all FW daemons to the STARTED class and grant access to the FW user ID
  - Example:

```
rdef started fwkern.** stdata(user(fwkern) group(fwgrp))
             icapslog.**
             icapsock.**
             icappftp.**
             icapcfgs.**
             icapstak.**
setr raclist(started) refresh
```



- Grant Firewall kernel access to TCP datasets
  - Example: **pe tcpip.** id(fwkern) ac(read)**

- Allow Firewall logging to SMF access is needed to BPX.SMF
  - Example:

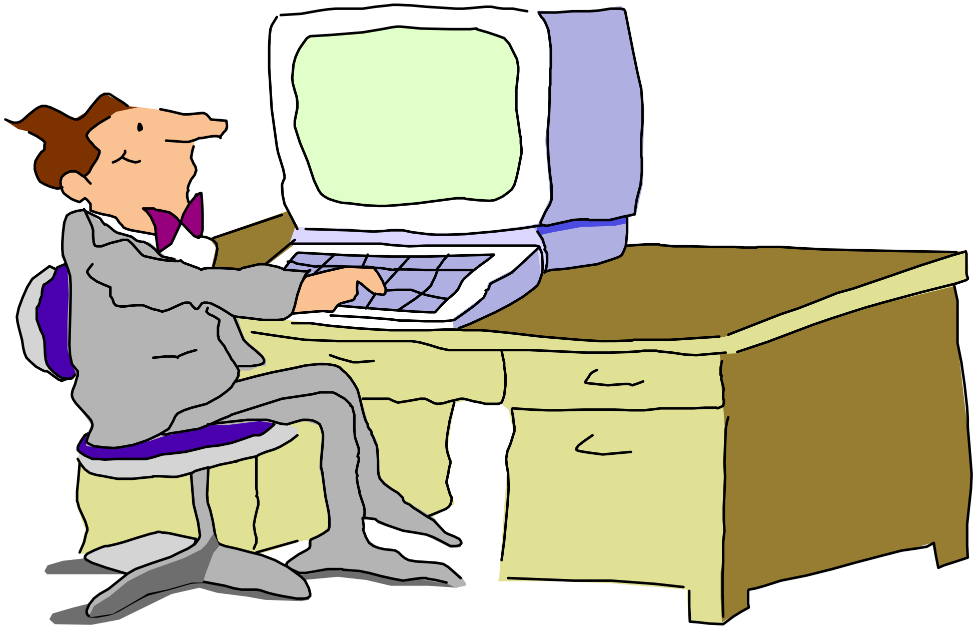
```
rdef  facility  bpx.smf  uacc(none)
pe  bpx.smf  cl(facility)  id(fwkern)  ac(read)
```

- Allow FTP daemon to change identity to another UID
  - Example:

```
rl facility bpx.daemon all
rdef  facility bpx.daemon  uacc(none)
pe  bpx.daemon  cl(facility)  id(fwkern)  ac(read)
```

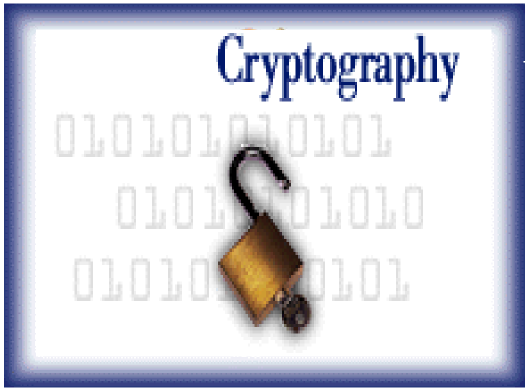
Configuration Server Access

- Control access to the configuration server
  - Example: **rdef facility ica.cfgsrv uacc(none)**
    pe ica.cfgsrv cl(facility) id(userid) ac(update)
    setr raclist(facility) refresh



Integrated Cryptographic Service Facility/MVS

- Encryption hardware service firewall can use;
  - CSFCKI - clear key import callable service
  - CSFDEC1 - decipher (with ALET) callable service
  - CSFENC1 - encipher (with ALET) callable service
  - CSFRNG - random number generate callable service
  - CSFCKM - clear key multiple import callable service
  - CSFOWH1 - one way hash (with ALET) callable service



- Examples;

ACTIVATE CLASS: setr cl(CSFSERV)
DEFINE SERVICES; rdef CSFSERV *service-name* uacc(none)
PERMIT USER ACCESS: pe *service-name* cl(CSFSERV) id(*yourid*) ac(read)

REFRESH IN STORAGE PROFILES: setr raclist(CSFSERV) refresh

TCP/IP Firewall Updates

- Define Firewall adapters in TCP/IP profile
  - add DEVICE and LINK statements for the system adapters
    - Example:

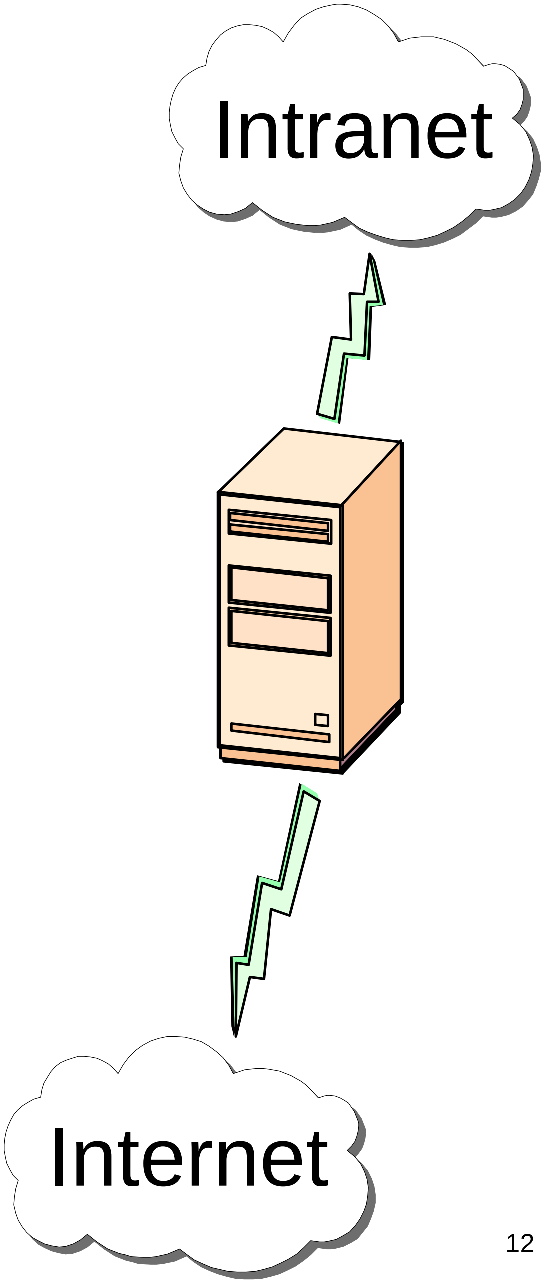
```
DEVICE  OSA5510  LCS  5510
LINK  OSTR5510  IBMTR    0  OSA5510
;
DEVICE  CTC1  CTC  5530
LINK  LINKMVS CTC   1  CTC1
```

- Internet (IP) addresses of each link in the host
  - Example:

```
HOME
9.81.10.5        OSTR5510
192.168.16.5   LINKMVS
```

- Start all the defined devices
  - Example: START---device_name

```
START OSA5510
START LINKMVS
```



TCP/IP Firewall Updates

- Add AUTOLOG statements for the firewall kernel
  - Example:

```
AUTOLOG
 FWKERN  ; OS/390 Firewall Kernel
ENDAUTOLOG
```

- Define port reserves for Firewall Technologies daemons
  - Example: PORT

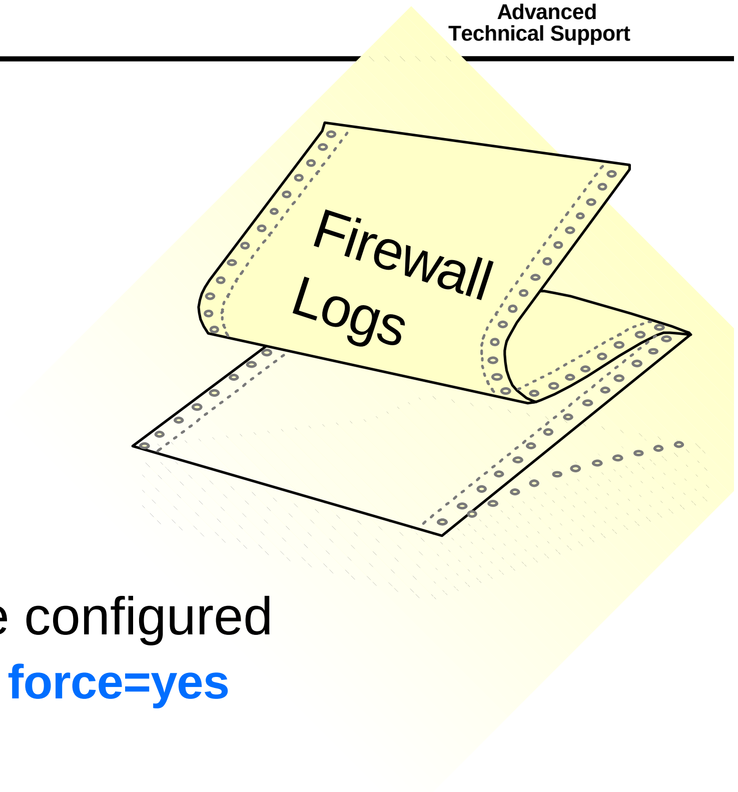
```
  20  TCP OMVS   NOAUTOLOG  ; Firewall FTP Proxy server
  21  TCP OMVS              ; Firewall FTP Proxy server
  53  TCP OMVS              ; OS/390 Firewall Domain Name Server
  53  UDP OMVS              ; OS/390 Firewall Domain Name Server
 514  UDP OMVS              ; OS/390 Firewall SYSLOGD
1080  TCP OMVS              ; Firewall Socks Server
1014  TCP OMVS              ; Config Server
```

- Identify TCP/IP stack as a firewall and enable transfer of data between networks
  - IPCONFIG FIREWALL DATAGRAMFWD

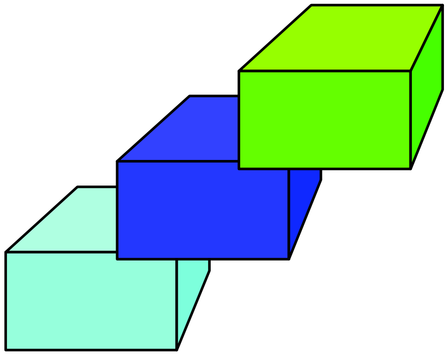
SYSLOG & Firewall Stack

- Create /etc/services under Unix Services
  - add definition for the *SYSLOG* server

```
 -----      /etc/services  --------
syslog         514/udp
```



- Firewall stack
  - FWSTACK - define firewall stacks for each one configured
    - Example: **fwstack cmd=add stack=stackname force=yes**



Configuration Files

- **/usr/lpp/fw/etc** =====> **/etc**
  - syslog.conf - logging server configuration
  - fwftp.data - FTP proxy configuration
  - fwftp.deniedusers - FTP proxy configuration which list users that are denied access to the FTP proxy
- **/usr/lpp/fw/etc/security** =====> /etc/security
  - fwaudio.cfg - real audio
  - fwdaemon.cfg - firewall daemons
  - fwobjects.cfg - object definitions
  - fwservices.cfg - services
  - fwsocks.cfg - socks rules
  - fwrules.cfg - default filter rules
  - logmgmt.cfg - log management
  - fwguicmds.En_US or fwguicmds.Ja_JP (if Japanese version is installed)

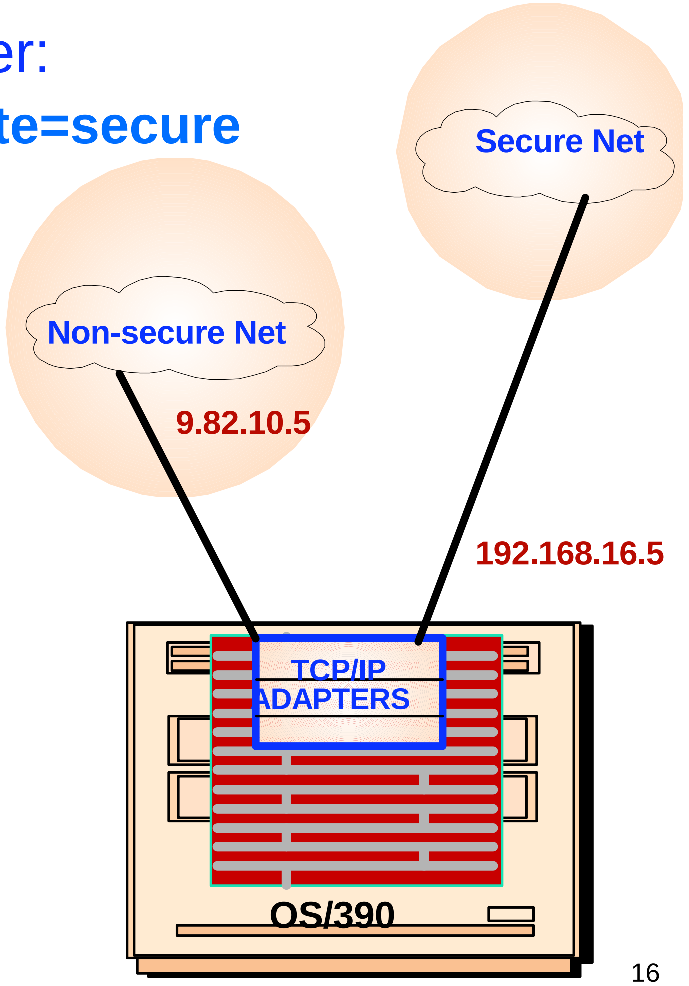
Identification of Secure Adapters

- To list the adapters attached to the Firewall machine
  - **fwadapter cmd=list [addr=x.x.x.x]**

```
          9.82.10.5      Non-Secure Interface    OSTR5510
        192.168.16.5      Non-Secure Interface    LINKMVS
```

- To set the secure/nonsecure state of the adapter:
  - **fwadapter cmd=change addr=192.168.16.5 state=secure**
  - **fwadapter cmd=list**

```
          9.82.10.5     Non-Secure Interface   OSTR5510
        192.168.16.5    Secure Interface           LINKMVS
```



Server Configuration File

- **fwdaemon cmd=list**
  - used to list and change server configuration attributes

```
SYSLOGD    Yes  300  300  1
SOCKD       No  300  300  300
PFTPD       No  300  300  300
CFGSRV      No  300  300  1

FWSTACKD   Yes  300  300  1
```

  - query server status
  - start and stop individual servers

- **S FWKERN** - start Firewall
  - View started servers

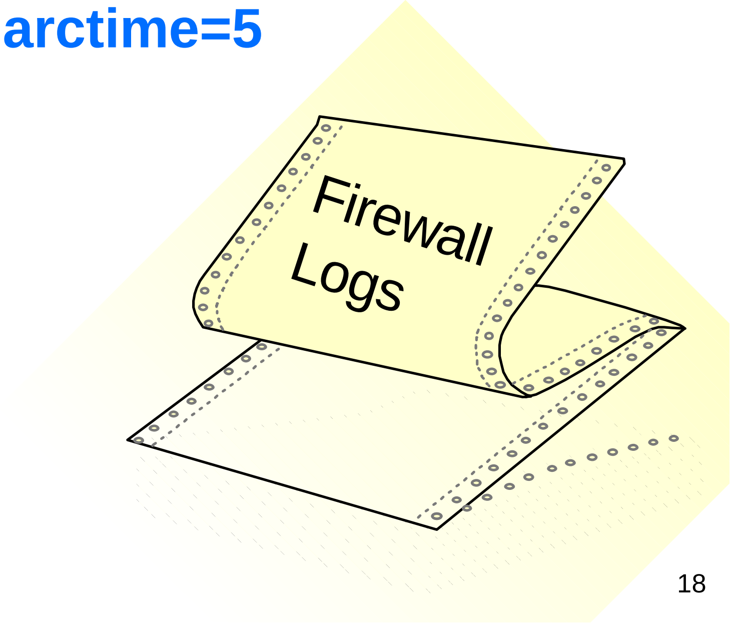
f fwkern,query all

```
FIR1 STC00298  ICAM1001i Firewall daemon SYSLOGD status is READY and   process id is 50331659
FIR1 STC00298  ICAM1001i Firewall daemon FWSTACKD status is READY and process id is 6710887
```

Syslog.conf Default Log

- **SYSLOG.CONF**
  - specifies logging defaults
  - located in /usr/lpp/fw/etc.syslog.conf
  - messages of all priorities from all facilities are logged in;
    - /var/fw/fwdata/syslogd.local0
    - /var/fw/fwdata/syslogd.local4
    - /var/fw/fwdata/*

- Define a log
  - Example: **fwlog cmd=add facility=firewall priority=info logfile=/var/fw/fwdata/admin.info logtime=3 arcfile=/var/fw/fwdata/arcfile.a arctime=5 workspace=/tmp**

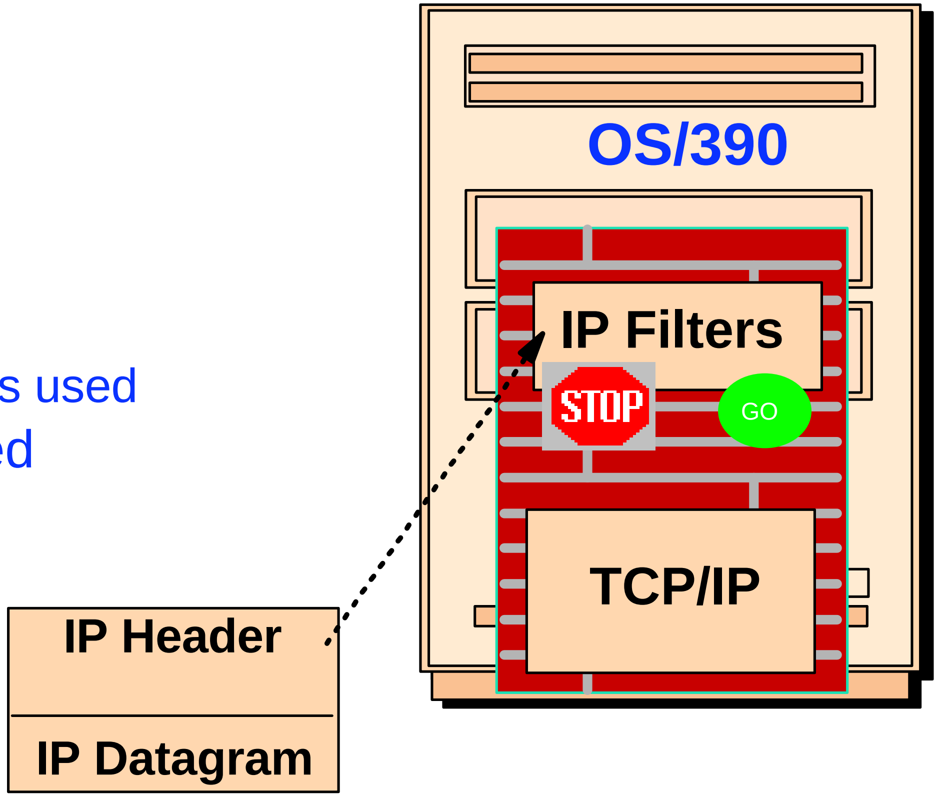


GUI Configuration

- Configuration server uses Secure Sockets Layer (SSL) protocol for communication
  - Configure SSL
    - run gskkyman SSL command
    - use option Create a self-signed certificate
    - use option Store encrypted database password
  - Configure Configuration Server (CFGSV)
    - define name of encrypted password file to CFGRSV
      - Example: **fwdaemon cmd=change daemon=CFSSRV daemonopts="-f /dir/key.kdp -p 1014"**
    - allow Firewall user ID to start the server
      - Example: **fwdaemon cmd=change started=yes daemon=CFGSRV**
  - Setup up the Configuration Client code on AIX or Windows

IP Packet Filtering

- IP level technology for controlling access through the firewall
  - allows or stops packets based on information in IP header
- Each packet is filtered separately
  - packets are either passed or ignored
- Filters Internet packets
  - controlled by filter rules
    - allow/deny packets
    - searched from top down
    - last rule should deny everything
    - first rule that matches a packet is used
  - unwanted packets are discarded



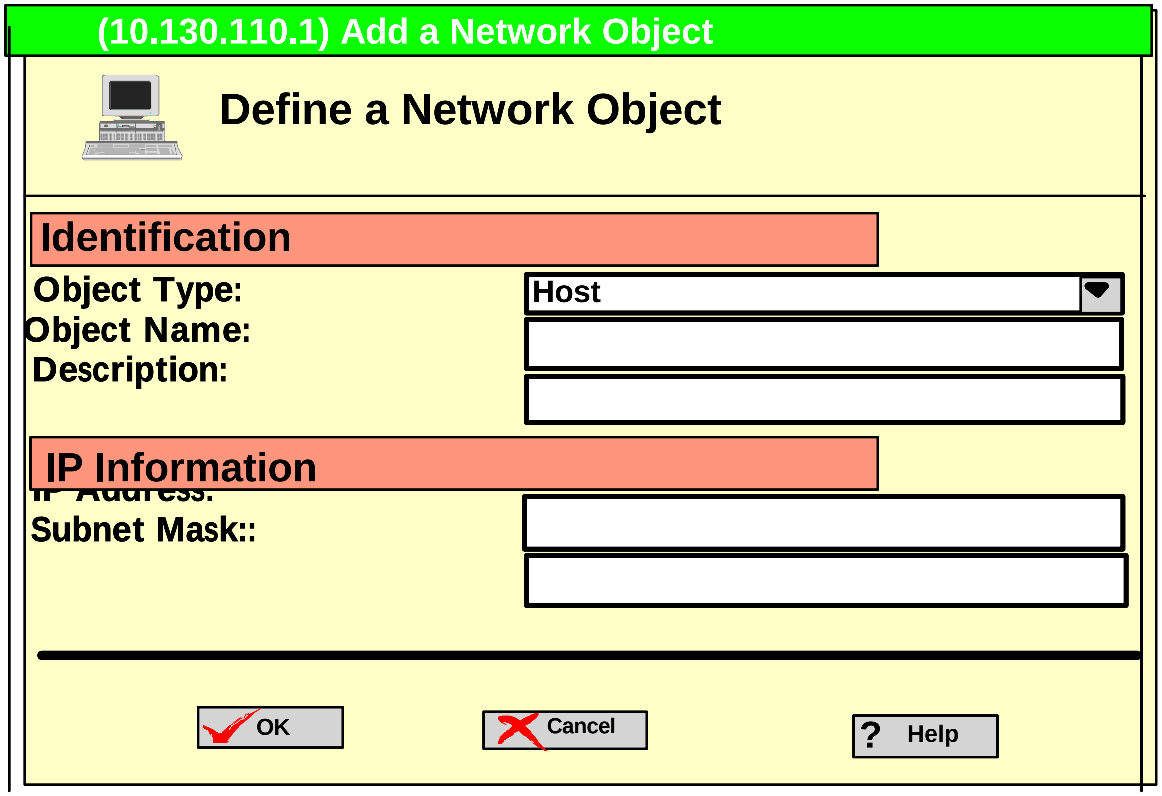
Components of Filtering

- Network Objects
- Network Object Groups
- Rules
- Services
- Connections

Network Objects

- Represent various hosts and entities
- Defined with "**fwnwobj**" command

```
fwnwobj cmd=add name=LAN_1A type= Network
             desc="1 LAN' addr=10.130.10.0 mask=255.255.255.0
```



(10.130.110.1) Add a Network Object

Define a Network Object

Identification

Object Type: Host

Object Name:

Description:

IP Information

IP Address:

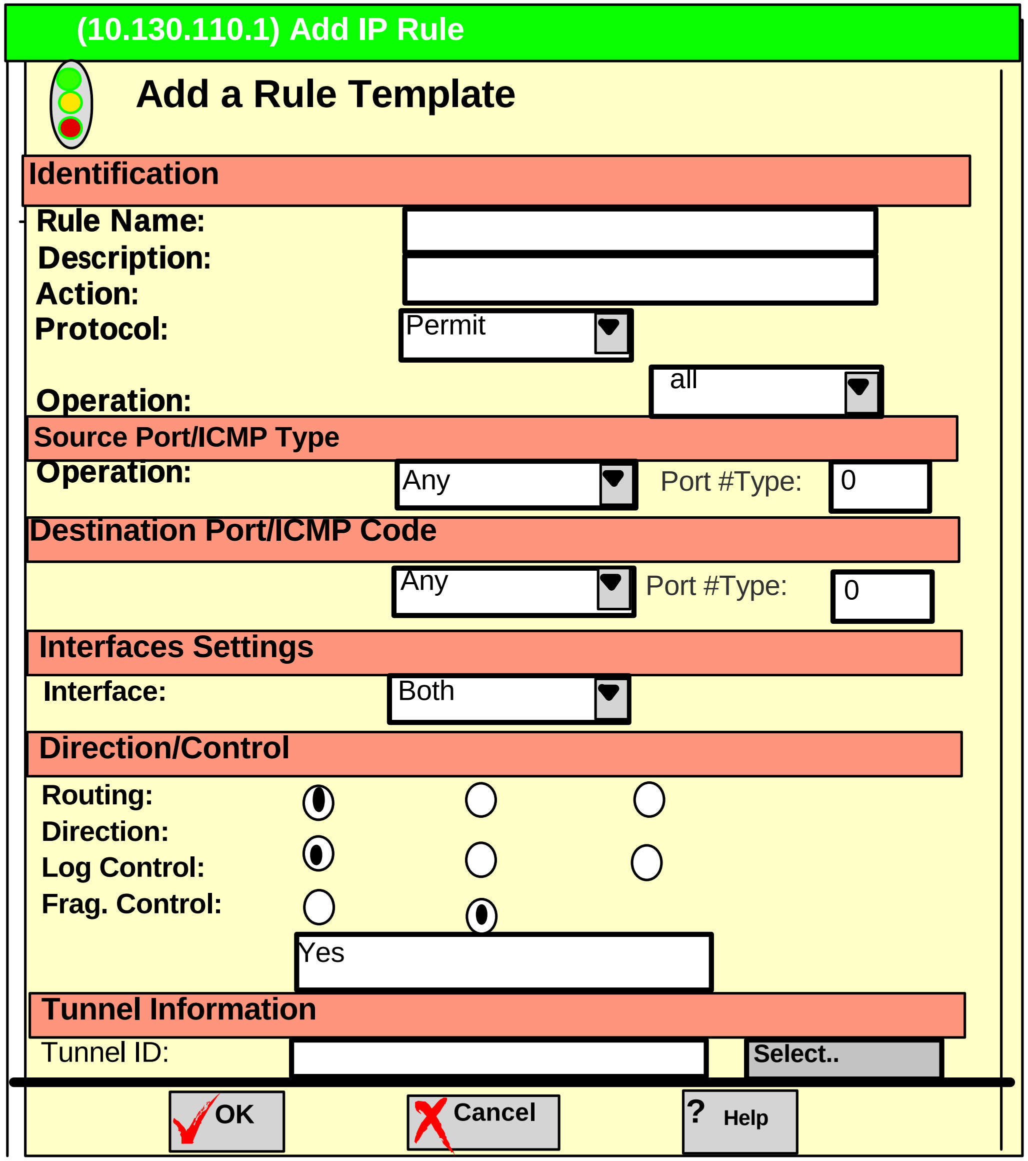
Subnet Mask::

Rules

- Instructions to permit or deny packets
- Defined with "**fwfrule**" command or via the GUI

```
fwfrule cmd=add name="Ping"
        desc="ICMP port 8"
        type=permit protocol=icmp
        srcopcode=eq srcport=8
        destopcode=eq destport=0
        interface=both routing=both
        direction=both log=no

fwfrule cmd=add name="Ping Response"
        desc="ICMP port 0"
        type=permit protocol=icmp
        srcopcode=eq srcport=0
        destopcode=eq destport=0
        interface=both routing=both
        direction=both log=no
```



(10.130.110.1) Add IP Rule

Add a Rule Template

Identification

Rule Name:
Description:
Action:
Protocol: Permit
Operation: all

Source Port/ICMP Type

Operation: Any   Port #Type: 0

Destination Port/ICMP Code

Any   Port #Type: 0

Interfaces Settings

Interface: Both

Direction/Control

Routing: ● ○ ○
Direction:
Log Control: ● ○ ○
Frag. Control: ○ ●

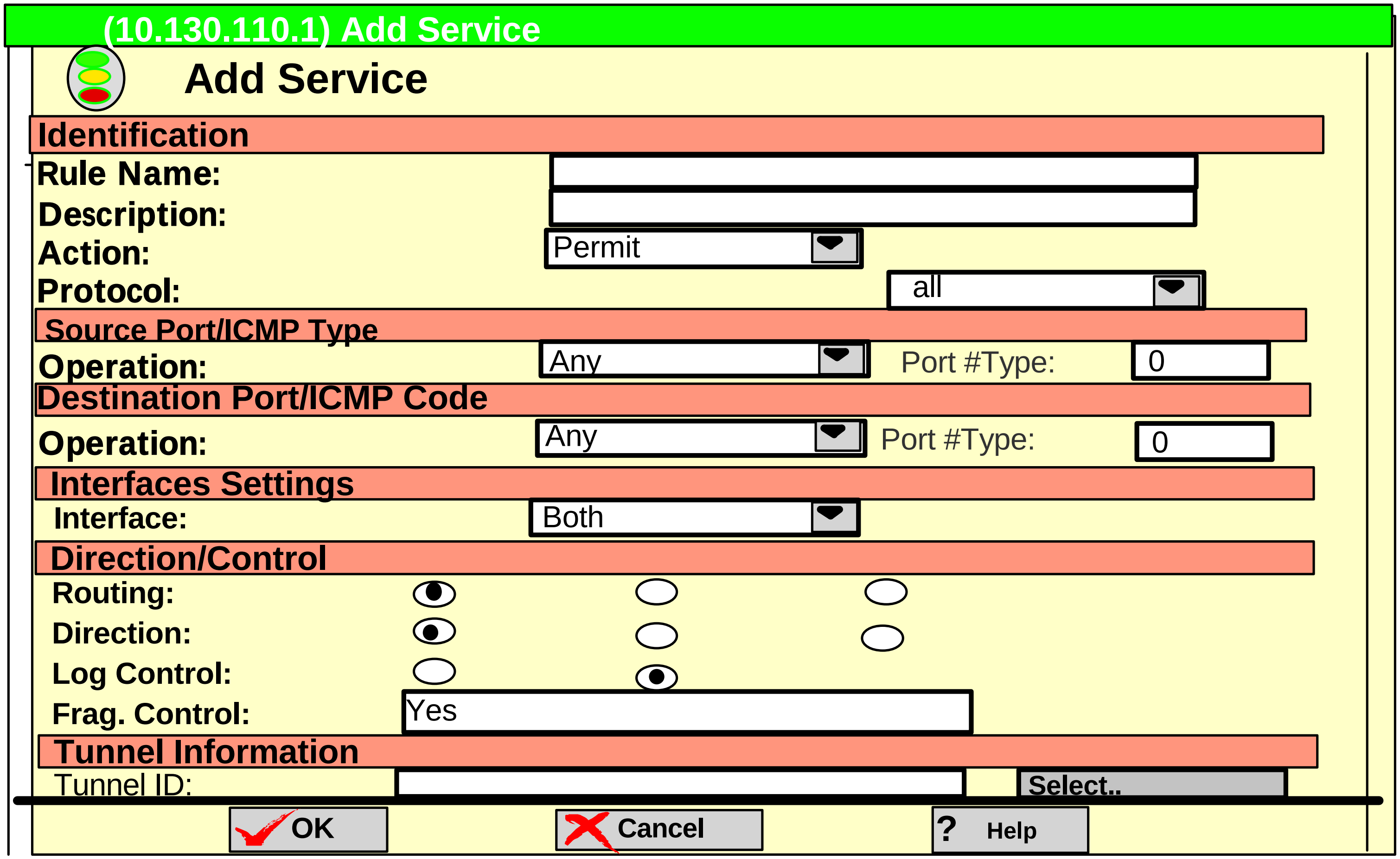
Yes

Tunnel Information

Tunnel ID:

Services

- Groups of rules which instruct the firewall to permit or deny access
  - Defined with "**fwservice**" command or GUI
    - **fwservice cmd=create name=Ping**
      desc="Allow outbound Ping to anywhere"
      rulelist=13/f,12/b



(10.130.110.1) Add Service

Add Service

Identification

Rule Name:	
Description:	
Action:	Permit
Protocol:	all

Source Port/ICMP Type

Operation: Any Port #Type: 0

Destination Port/ICMP Code

Operation: Any Port #Type: 0

Interfaces Settings

Interface: Both

Direction/Control

Routing: ● ○ ○

Direction: ● ○ ○

Log Control: ○ ●

Frag. Control: Yes

Tunnel Information

Tunnel ID:

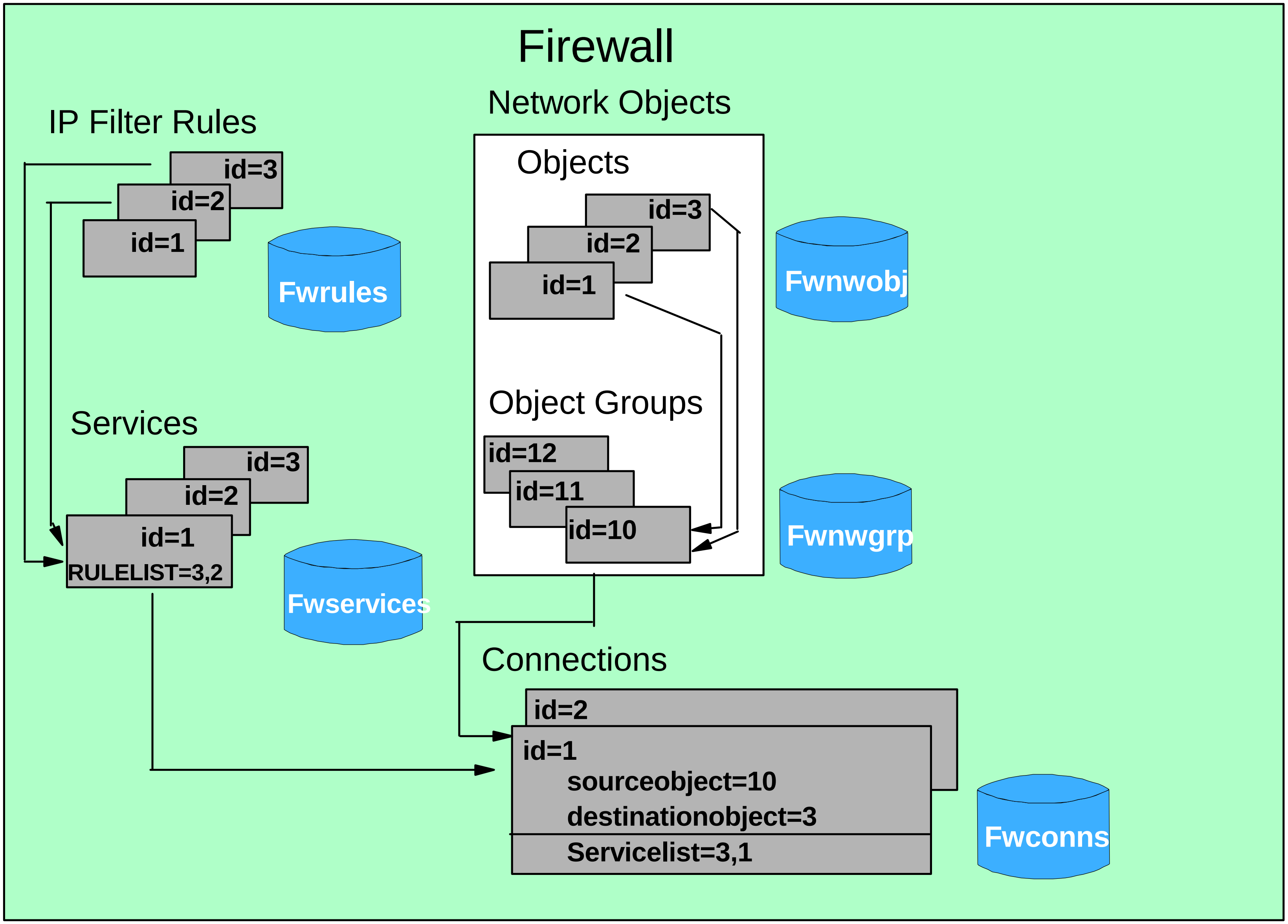
Connections

- Associate network objects with services to define types of communications allowed between endpoints
  - defined with "**fwconns**" command

    -
      ```
      fwconns cmd=create name="Allow Internet Ping"
                 desc="Allow Pings from Lan_1A to Internet"
                 source=Lan_1A destination="The World"
                 servicelist=18
      ```

name	name you assign to this connection
desc	description that you give to this connection
source	ID of source network object
destination	ID of destination network object
servicelist	ID's of service rules that apply to this connection

Configuration Overview



Refresh

```
fwfrule cmd=add name="Ping Response"
        desc="ICMP port 8" type=permit
        protocol=icmp srcopcode=eq
        srcport=8 destopcode=eq
        destport=0 interface=both routing=both
        direction=both log=no
```

```
fwnwobj cmd=add name=LAN_1A type= Network
        type=network addr=10.130.10.0
        mask=255.255.255.0
```

```
fwconns cmd=create name="Allow Internet
        desc=" Pings from LAN_1A to Internet"
        source=Lan_1A destination="The World"
        servicelist=18
```

```
fwfrule cmd=add name="Ping" desc="ICMP port 0
        type=permit protocol=icmp srcopcode=eq
        srcport=0 destopcode=eq destport=0
        interface=both routing=both direction=both
        log=no
```

```
fwnwobj cmd=add name=the world  type= Network
        desc="1 LAN' addr=0.0.0.0
        mask=0.0.0.0
```

```
fwservice cmd=create name=Ping
        desc="Permit Ping outbound"
        rulelist=13/f,12/b
```

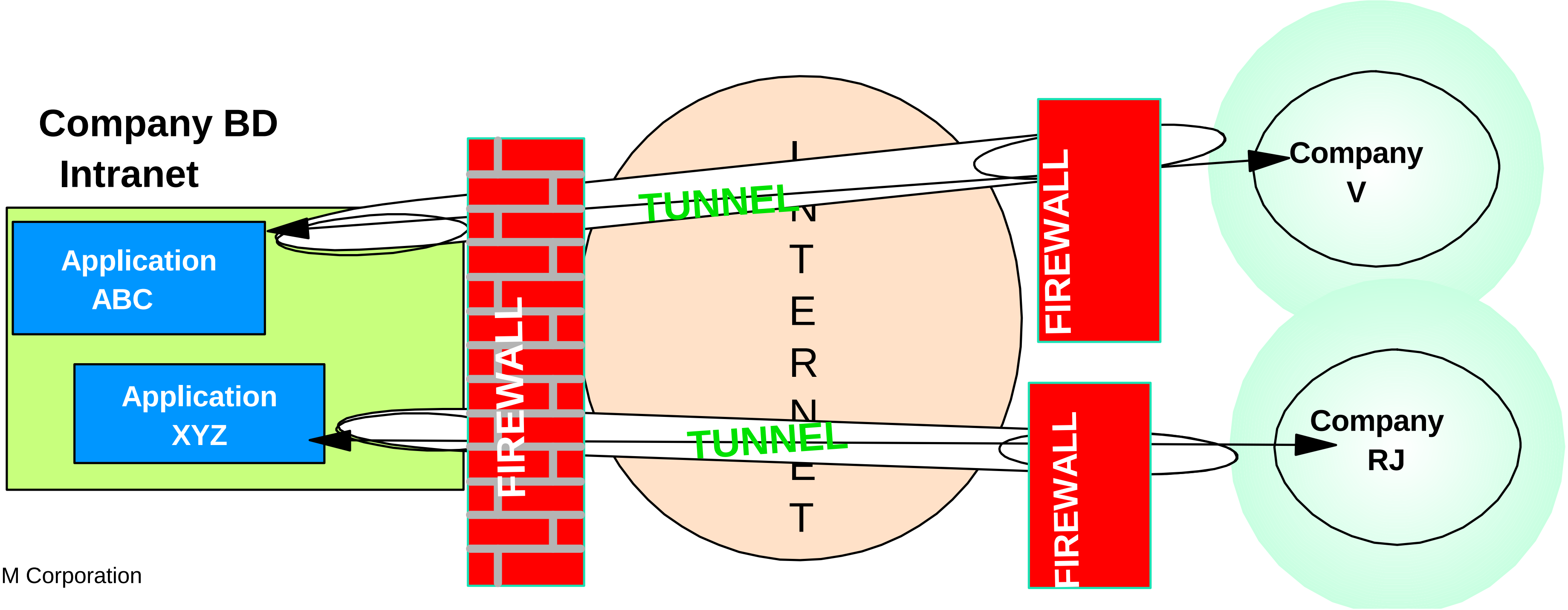
FWFILTER cmd=update

RESULTS: fwfilter cmd=list

```
#Service: Ping
#Description: Permit Ping Outbound
permit 10.130.10.0  255.255.255.0  0.0.0.0  0.0.0.0  icmp eq 8 eq 0 both both both l=n f=y
permit 0.0.0.0  0.0.0.0  10.130.10.0  255.255.255.0  icmp eq 0 eq 0 both both both l=n f=y
```

Virtual Private Networks

- Virtual Private Networking allows secure communications between remote sites over a public network like the Internet
  - Communications over VPN can be authenticated and encrypted
  - Virtual Private Network is comprised of one or more IP tunnels between two networks
  - VPN is included with OS/390 Firewall Technologies
  - Packets sent through a tunnel can be;
    - encrypted and/or authenticated
    - sent in a new IP packet to the destination firewall
    - sent using IPSec protocol, not TCP or UDP



VPN Configuration

- To configure tunnels;
  - Local Host
    1. create firewall network objects
    2. add tunnel definition
    3. export the tunnel definition to a set of files
    4. transfer the tunnel definition files to the partner tunnel
    5. define filter rules and services for VPN
    6. add connection definitions
  - Remote Host
    7. import the tunnel definition in the remote firewall
    8. repeat 1,5,6 at remote firewall
  - Both Hosts
    9. activate rulesets at both ends
    10. activate tunnel at both ends
    11. refresh the tunnel when session key has expired

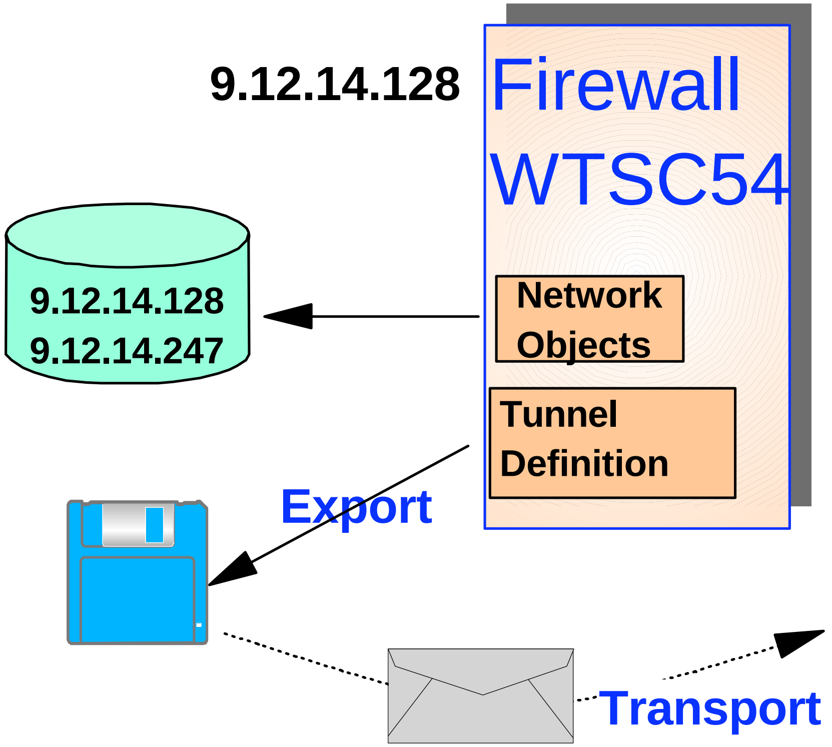
Tunnel Setup

- Network Objec
  - fwnwobj c Host "
    WTSC5 255
  - fwnw TSC57"
    ty

```
fwtunnl cmd=add tunnel=391 type=manual addr=9.12.14.128
          remaddr=9.12.14.247 policy=auth algorithm=KEYED_MD5
          spi=500 timeout=480
```

Tunnel Setup (Export)

- Tunnel definition must be exported to a format that may be transported to the remote site for importing into the partner firewall
  - **fwtunnl cmd=export directory /anyexportdirectoryname tunnel=391**
- Command creates two files in /anyexportdirectoryname called **fwexppolicy** and **fwexpmctx.manual**
- Transport files to remote host



Tunnel Setup (rule, service & connection)

- Rule (WTSC54, 9.12.14.128)

  -
    ```
    fwfrule cmd=add type=permit name=tunneltraffic
                 desc="route all traffic"  protocol=all srcopcode=any
                 srcport=0 destopcode=any destport=0 interface=nonsecure
                 routing=local direction=both log=no  tunnel=391
    ```

  -
    ```
    fwfrule cmd=add type=permit name=ahtraffic
                 desc="authenticated traffic" protocol=ah srcopcode=any
                 srcport=0 destopcode=any destport=0 interface=nonsecure
                 routing=local direction=both log=yes
    ```

- Service (WTSC54, 9.12.14.128)

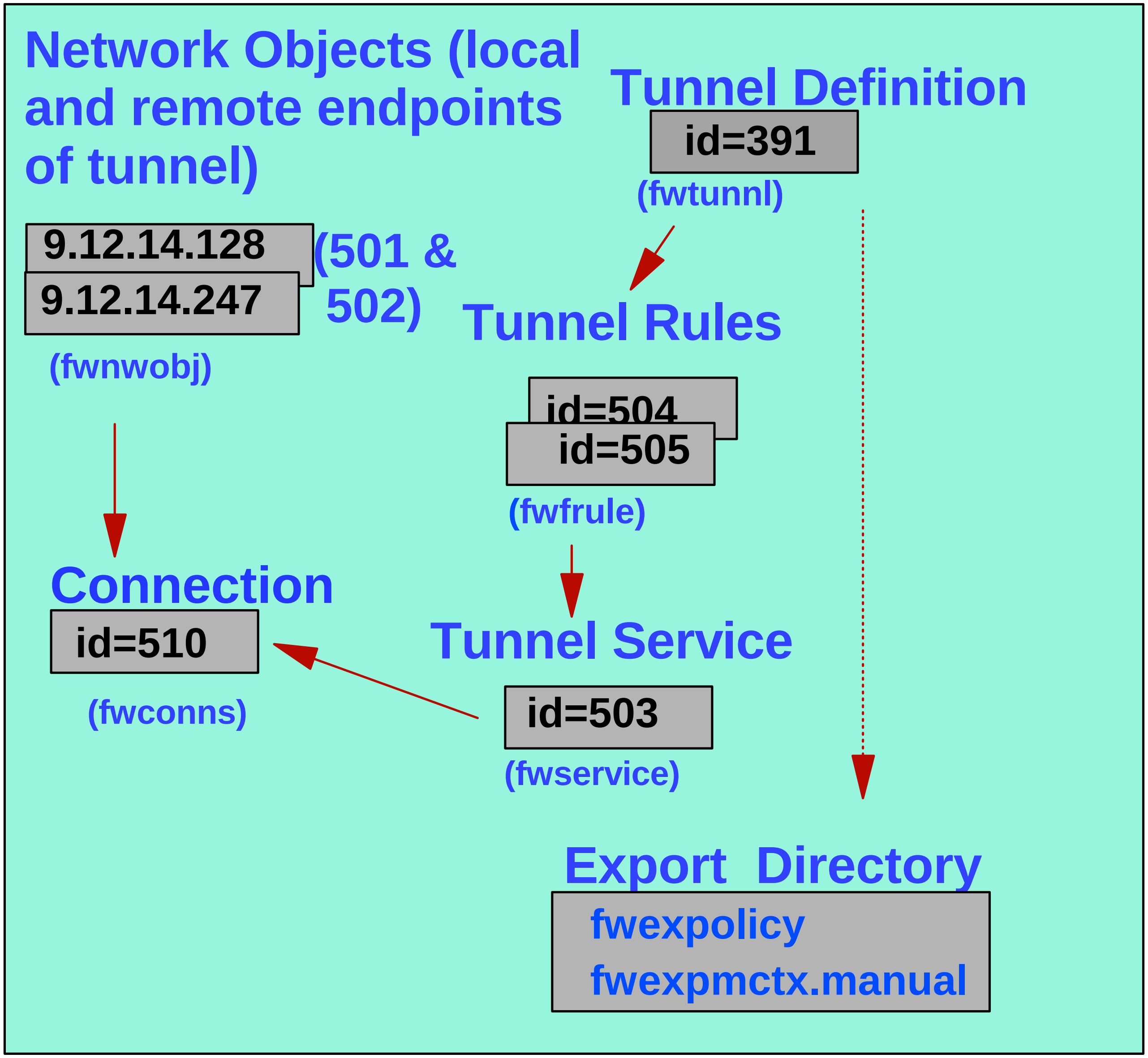
  -
    ```
    fwservice cmd=create name=alltrafficservice desc="all traffic"
                 rulelist=505/f,505/b,504/f,504/b
    ```

- Connection

  -
    ```
    fwconns cmd=create name=alltrafficconnection source=501
                 destination=502 servicelist=503
    ```

Tunnel Review

G'Burg Firewall WTSC54 9.12.14.128



Network Objects (local and remote endpoints of tunnel)

9.12.14.128
9.12.14.247
(501 & 502)
(fwnwobj)

Tunnel Definition
id=391
(fwtunnl)

Tunnel Rules
id=504
id=505
(fwfrule)

Connection
id=510
(fwconns)

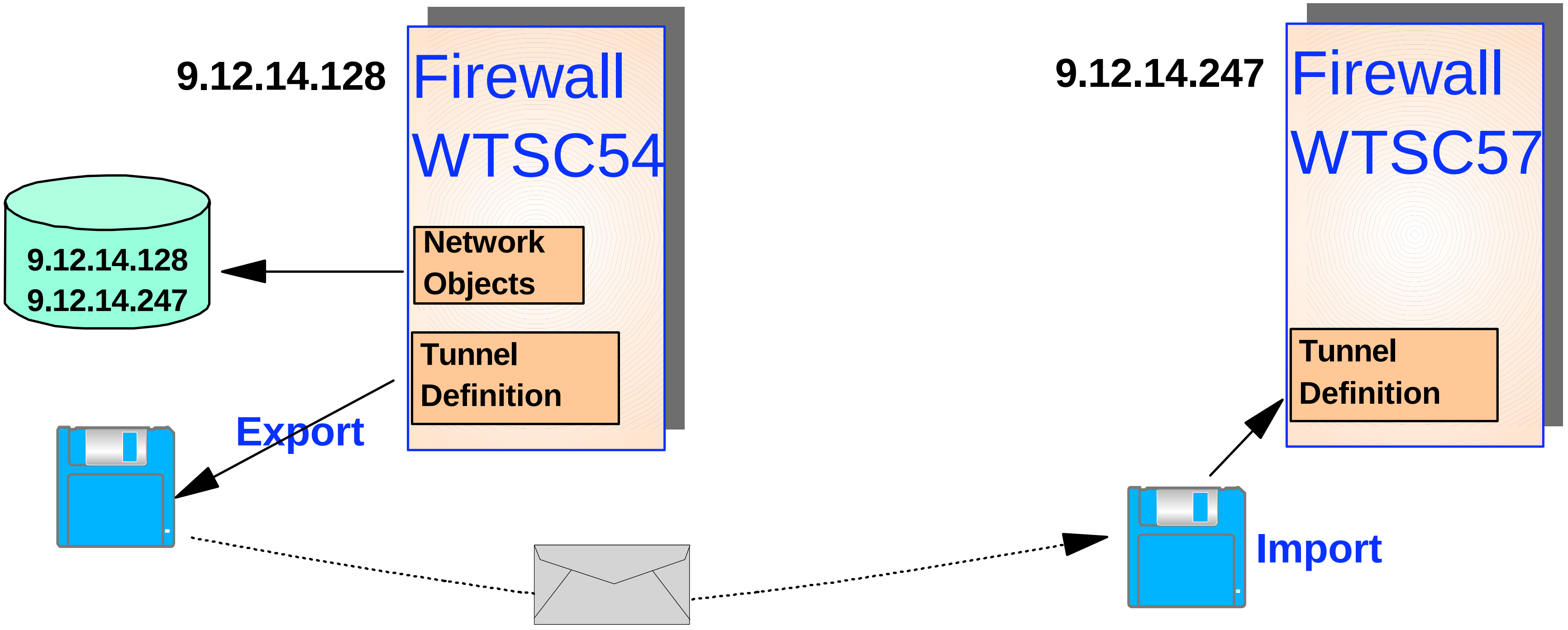
Tunnel Service
id=503
(fwservice)

Export Directory
fwexpolicy
fwexpmctx.manual

Receive and Import

- Receive files sent from tunnel partner

- After tunnel partner receives the exported files, place them in a directory and import the definitions;
  - **fwtunnl cmd=import directory=/importdirectoryname tunnel=391**



Tunnel Setup (Objects, & Rules)

- Network Objects (WTSC57, remote host)

  -
    ```
    fwnwobj cmd=add name=wtsc57 desc="wtsc57 system"
                   type=host addr=9.12.14.247 mask=255.255.255.255
    ```

  -
    ```
    fwnwobj cmd=add name=wtsc54 desc="wtsc54 host"
                   type=host addr=9.12.14.128 mask=255.255.255.255
    ```

- Filter Rule (WTSC57 9.12.14.247)

  -
    ```
    fwfrule cmd=add type=permit name=tunneltraffic desc="route all"
                 protocol=all srcopcode=any srcport=0 destopcode=any
                destport=0 interface=nonsecure routing=local direction=both
                 log=yes tunnel=391
    ```

  -
    ```
    fwfrule cmd=add type=permit name=vpn desc="authenticated traffic"
                 protocol=ah  srcopcode=any srcport=0 destopcode=any
                destport=0 interface=nonsecure routing=local direction=both
    ```

Tunnel Setup (Service & Connections)

- Service (WTSC57, 9.12.14.247)

  -
    ```
    fwservice cmd=create name=alltrafficservice desc="all traffic"
                      rulelist=501/f,501/b
    ```

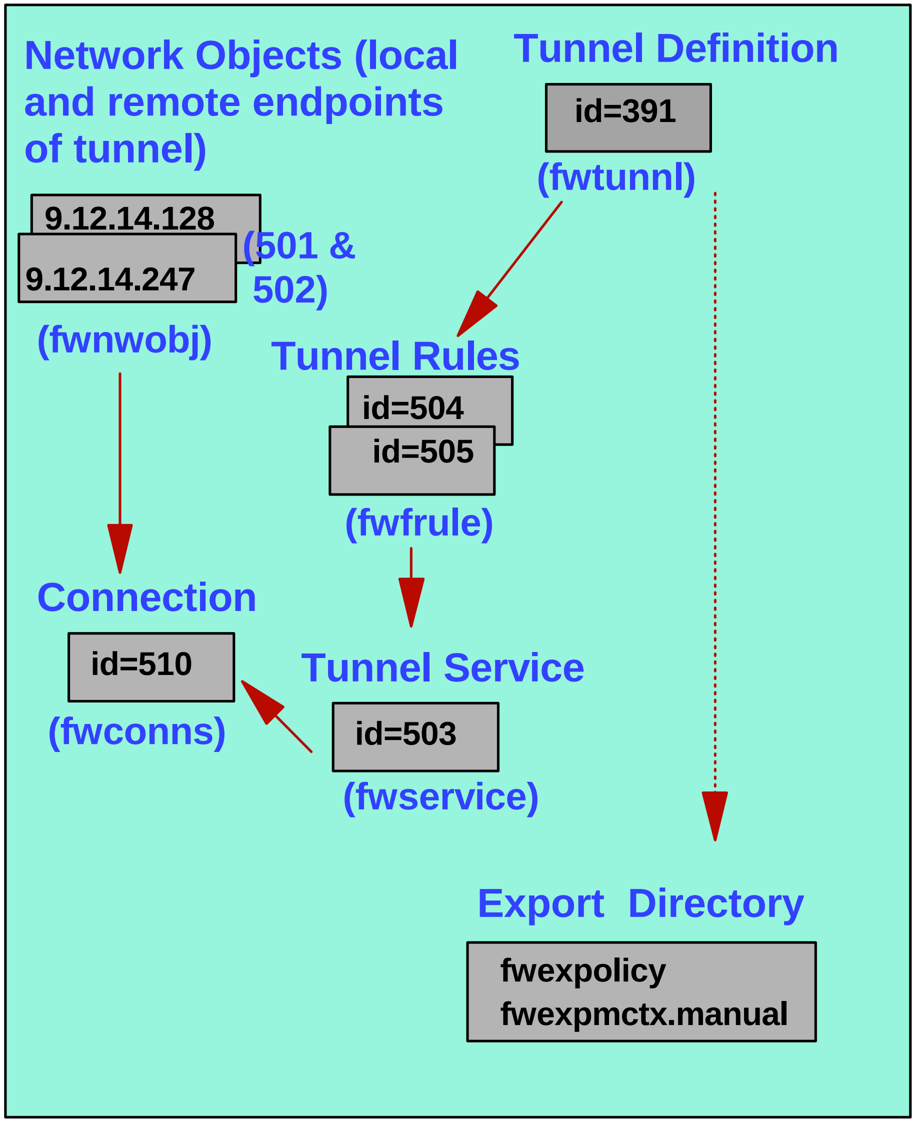
  -
    ```
    fwservice cmd=create name=vpnauth desc="authenticate traffic"
                      rulelist=503/f,503/b
    ```
- Connections

  -
    ```
    fwconns cmd=create name=alltrafficconnection
                  desc="connect all traffic" source=521 destination=524
                  servicelist=551,552
    ```

Tunnel Review

G'Burg WTSC54
Firewall 9.12.14.128

N. Y. WTSC57
Firewall 9.12.14.247



Network Objects (local and remote endpoints of tunnel)

9.12.14.128
9.12.14.247
(501 & 502)
(fwnwobj)

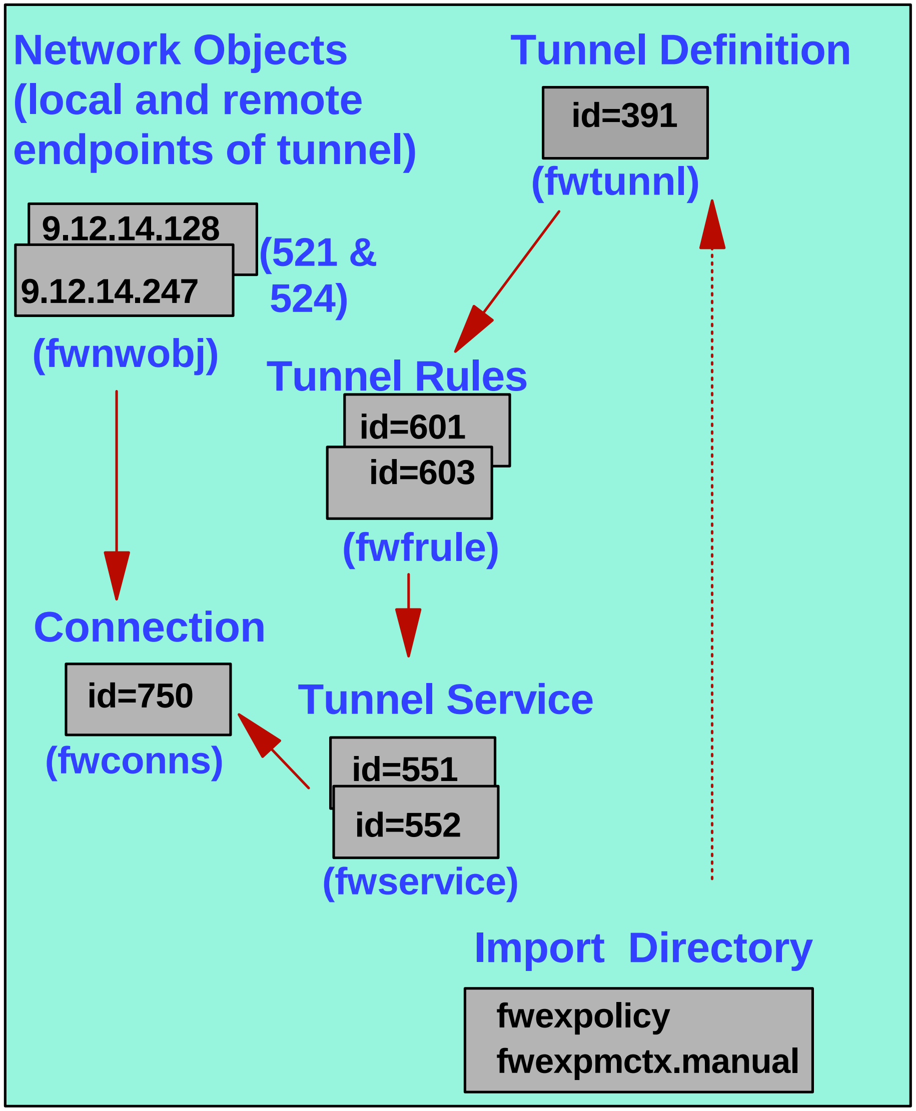
Tunnel Definition
id=391
(fwtunnl)

Tunnel Rules
id=504
id=505
(fwfrule)

Connection
id=510
(fwconns)

Tunnel Service
id=503
(fwservice)

Export Directory
fwexpolicy
fwexpmctx.manual



Network Objects (local and remote endpoints of tunnel)

9.12.14.128
9.12.14.247
(521 & 524)
(fwnwobj)

Tunnel Definition
id=391
(fwtunnl)

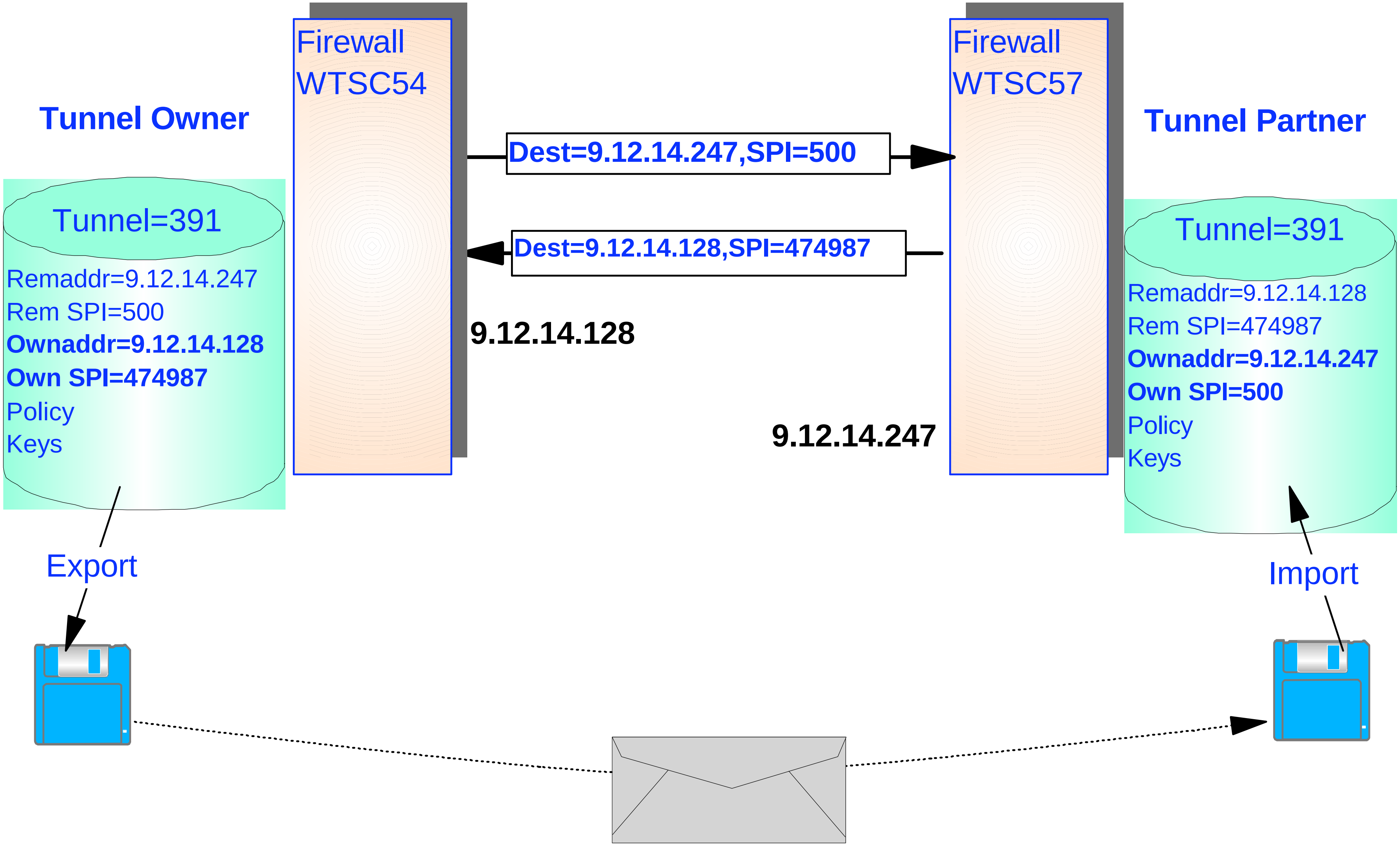
Tunnel Rules
id=601
id=603
(fwfrule)

Connection
id=750
(fwconns)

Tunnel Service
id=551
id=552
(fwservice)

Import Directory
fwexpolicy
fwexpmctx.manual

Tunnel Security Association



Activate Rulelist and Tunnels

- **fwfilter cmd=update**

- When both tunnel partners have the correct definitions, activate the tunnel
  - **fwtunnl cmd=activate tunnel=391**

  Tunnel activation enables the code and will be marked active even
  if the other end is not running or connected