Current research at LAS-IC-Unicamp

Paulo Lício de Geus

Laboratory of Systems Administration and Security
Institute of Computing
University of Campinas
Campinas, SP, Brazil

September 28, 2008

Agenda

MoBaSeC

Management of security services configuration

Imuno Project

Immune-based computer security

MoBaSeC

- Configuration management of security services
- Dealing with security policies
  - Hierarchical, model-based approach
  - Interactive policy refinement via graphical tool
- Helped by abstract subsystem viewing
- Correctness of config files through object-oriented design
- Being extended to the operating system level

MoBaSeC

- Configuration management of security services
- Dealing with security policies
  - Hierarchical, model-based approach
  - Interactive policy refinement via graphical tool
- Helped by abstract subsystem viewing
- Correctness of config files through object-oriented design
- Being extended to the operating system level

MoBaSeC

- Configuration management of security services
- Dealing with security policies
  - Hierarchical, model-based approach
  - Interactive policy refinement via graphical tool
- Helped by abstract subsystem viewing
- Correctness of config files through object-oriented design
- Being extended to the operating system level

Typical security administrator design activity

Task

Given a security policy like this:

"Allow Internet surfing"

- Generate config files for all security systems involved.
- Then, repeat that for all policies established by management.

Problems

Large variety of security functionalities and implementations

- Too many paradigms and syntaxes to familiarize

Scalability, understandability, need for integrated management

Typical security administrator design activity

Task

Given a security policy like this:

"Allow Internet surfing"

- Generate config files for all security systems involved.
- Then, repeat that for all policies established by management.

Problems

Large variety of security functionalities and implementations

- Too many paradigms and syntaxes to familiarize

Scalability, understandability, need for integrated management

An iptables set of rules, inside a much larger file

```
## SSH - external <-> dmz
for host in $BASTION ; do
  iptables -A FORWARD -i $EXT_IF -o $DMZ_IF -p tcp -s $ANY -d $host --dport ssh -m state --state NEW,ESTABLISHED -j ACCEPT
  iptables -A FORWARD -i $DMZ_IF -o $EXT_IF -p tcp -s $host --sport ssh -d $ANY -m state --state ESTABLISHED -j ACCEPT
  iptables -A FORWARD -i $DMZ_IF -o $EXT_IF -p tcp -s $host -d $ANY --dport ssh -m state --state NEW,ESTABLISHED -j ACCEPT
  iptables -A FORWARD -i $EXT_IF -o $DMZ_IF -p tcp -s $ANY --sport ssh -d $host -m state --state ESTABLISHED -j ACCEPT
done
unset host
```

Other filter language examples

Example: permission for an external host to all internal machines
172.16.51.50 $\longleftrightarrow$ 192.168.10.0–255

Cisco's IOS

```
access-list 101 permit ip 172.16.51.50 0.0.0.0 192.168.10.0 0.0.0.255
access-list 101 deny ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
interface serial 0
access group 101 in
access-list 102 permit ip 192.168.10.0 0.0.0.255 172.16.51.50 0.0.0.0
access-list 102 deny ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
interface serial 0
access group 102 out
```

Other filter language examples

Example: permission for an external host to all internal machines
172.16.51.50 $\longleftrightarrow$ 192.168.10.0–255

IP Filter

```
# block everything by default
block in from any to any
block out from any to any
# allow something
pass in from 172.16.51.50 to 192.168.10.0/24
pass out from 192.168.10.0/24 to 172.16.51.50
```

Other filter language examples

Example: permission for an external host to all internal machines
172.16.51.50 $\longleftrightarrow$ 192.168.10.0–255

flc

```
#define aquele_host 172.16.51.50
#define minha_rede 192.168.10.0
#if defined(__cisco__)
interface ethernet0;
access-list 101;
#endif
#if defined(__ipfilter__) || defined(__ipfirewall__)
interface le0;
#endif
#if defined(__ipfw__) || defined(__ipfwadm__)
interface 192.168.11.1
#endif
policy block all;
if ( from host aquele_host to minha_rede ) {
log and pass;
}
if ( from minha_rede to host aquele_host ) {
log && pass;
}
block .
end-policy
```

Other examples of security services to be configured

- Other packet filter functions
  NAT/PAT, reverse NAT, transparent proxy
- Proxy servers
- VPNs
- traffic shapers (since they are closely tied to packet filters)
- ad-hoc proxies for common Internet services
- IDSs
- IPSs
- Anti-virus and anti-spam boxes
- tcpwrappers
- DoS detectors and mitigators
- Logging/accounting/auditing structure

OpenVPN, squid . . .

```
remote 143.106.1.75 2222
dev tun
ifconfig 10.0.0.1 10.0.0.2
secret /etc/openvpn/lasitautec-key.txt
cipher AES-128-CBC   # AES
persist-key
persist-tun
port 2222
proto udp
user nobody
group nobody
route 10.2.1.0 255.255.255.0
verb 3
mssfix
```

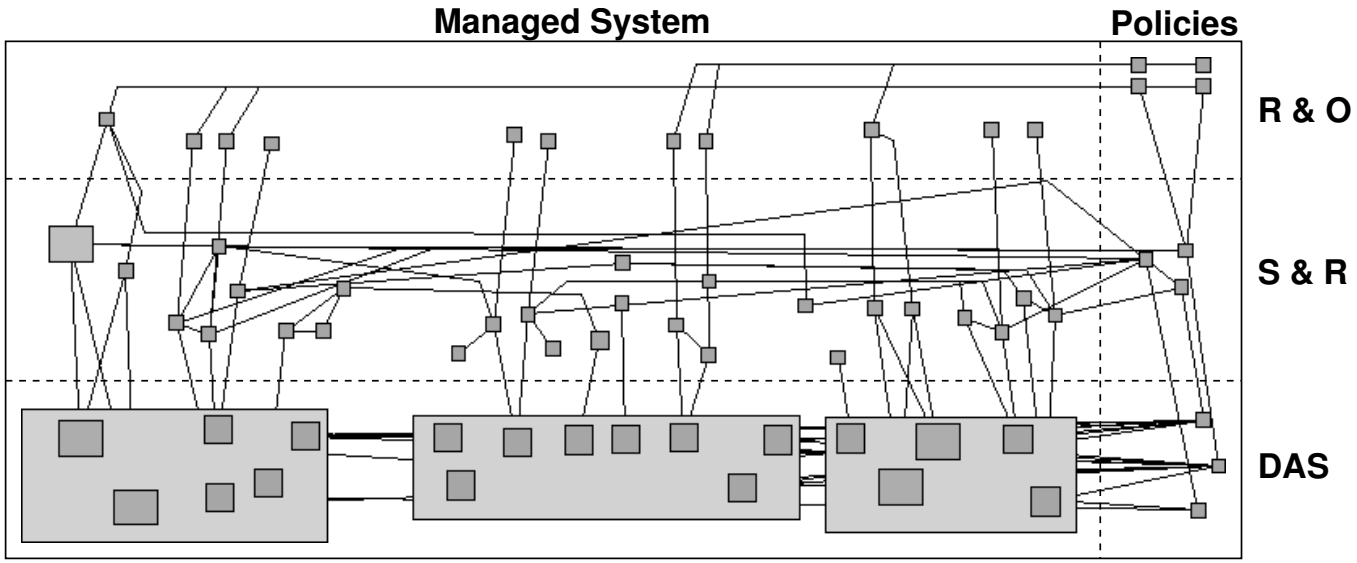
```
acl all src 0.0.0.0/0.0.0.0
acl interno src 10.1.1.0/255.255.255.0
acl itautec src 10.0.0.0/255.255.255.0
acl localhost src 127.0.0.1/255.255.255.255
acl to_localhost dst 127.0.0.0/8
acl SSL_ports port 443 563 1863 4443 5222
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443 563     # https
http_access allow manager localhost
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_port 10.1.1.1:8080
cache_peer 10.0.0.2 sibling 8080 3130
cache_peer proxy.unicamp.br sibling 3128 3130
...
```

OpenVPN, squid . . .

```
remote 143.106.1.75 2222
dev tun
ifconfig 10.0.0.1 10.0.0.2
secret /etc/openvpn/lasitautec-key.txt
cipher AES-128-CBC   # AES
persist-key
persist-tun
port 2222
proto udp
user nobody
group nobody
route 10.2.1.0 255.255.255.0
verb 3
mssfix
```

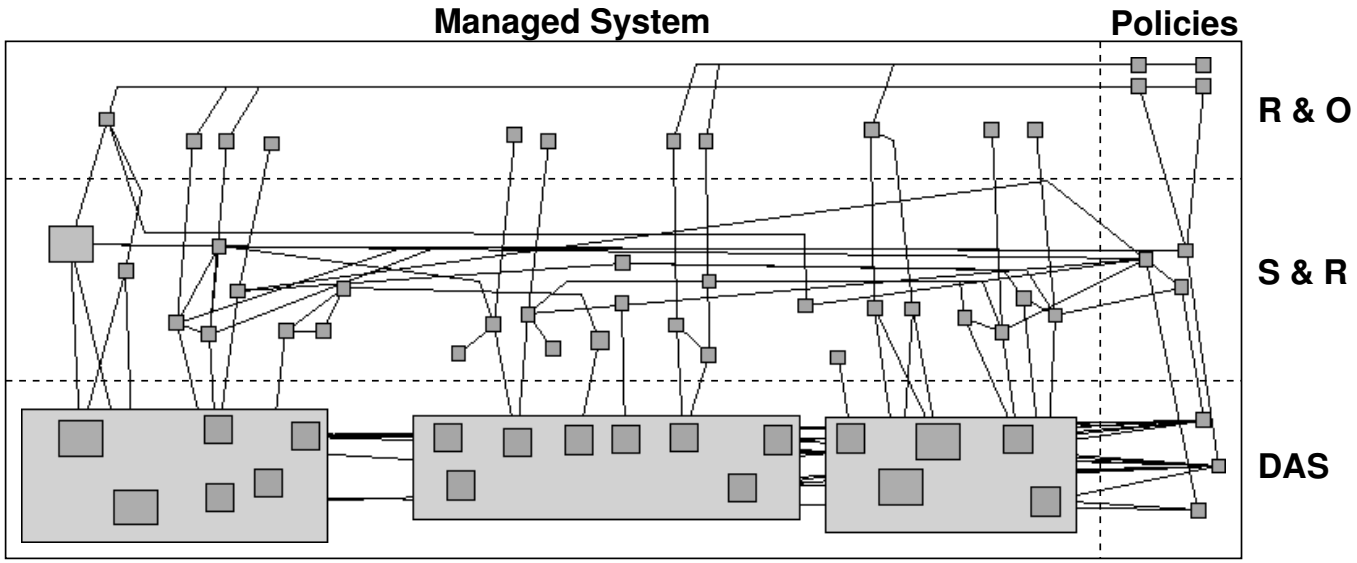
```
acl all src 0.0.0.0/0.0.0.0
acl interno src 10.1.1.0/255.255.255.0
acl itautec src 10.0.0.0/255.255.255.0
acl localhost src 127.0.0.1/255.255.255.255
acl to_localhost dst 127.0.0.0/8
acl SSL_ports port 443 563 1863 4443 5222
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443 563     # https
http_access allow manager localhost
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_port 10.1.1.1:8080
cache_peer 10.0.0.2 sibling 8080 3130
cache_peer proxy.unicamp.br sibling 3128 3130
...
```

Model-Based Management Approach



- Three-layered model
  - R & O Roles and Objects (RBAC concepts)
  - S & R Subjects and Resources
  - DAS Diagram of Abstract Subsystems (process/hosts)
- Automated building of a policy hierarchy
  - Configuration parameter generation
  - assisted by supporting tool

Model-Based Management Approach



- Three-layered model
  - R & O Roles and Objects (RBAC concepts)
  - S & R Subjects and Resources
  - DAS Diagram of Abstract Subsystems (process/hosts)
- Automated building of a policy hierarchy
  - Configuration parameter generation
  - assisted by supporting tool

Diagram of Abstract Subsystems (DAS)

Modular representation of a system's architecture

- Abstract Subsystems (ASs) as *building blocks*

Types of components

Actors	active elements Initiate communication Execute mandatory operations
Mediators	inspect, filter and/or transform data flows According to the policies
Targets	store relevant information
Connectors	communication interfaces between ASs Inert; help with graphing

Diagram of Abstract Subsystems (DAS)

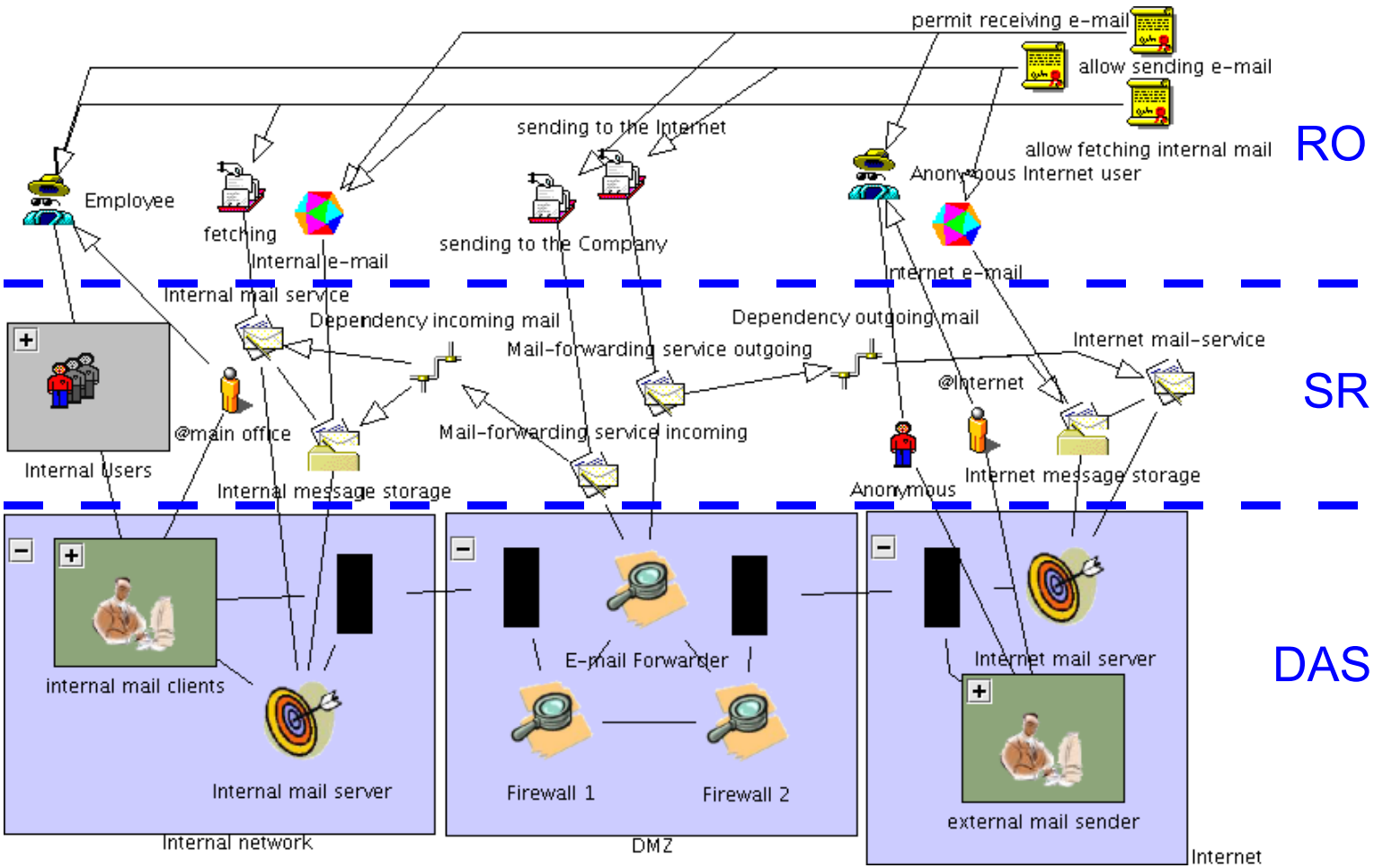
Modular representation of a system's architecture

- Abstract Subsystems (ASs) as *building blocks*

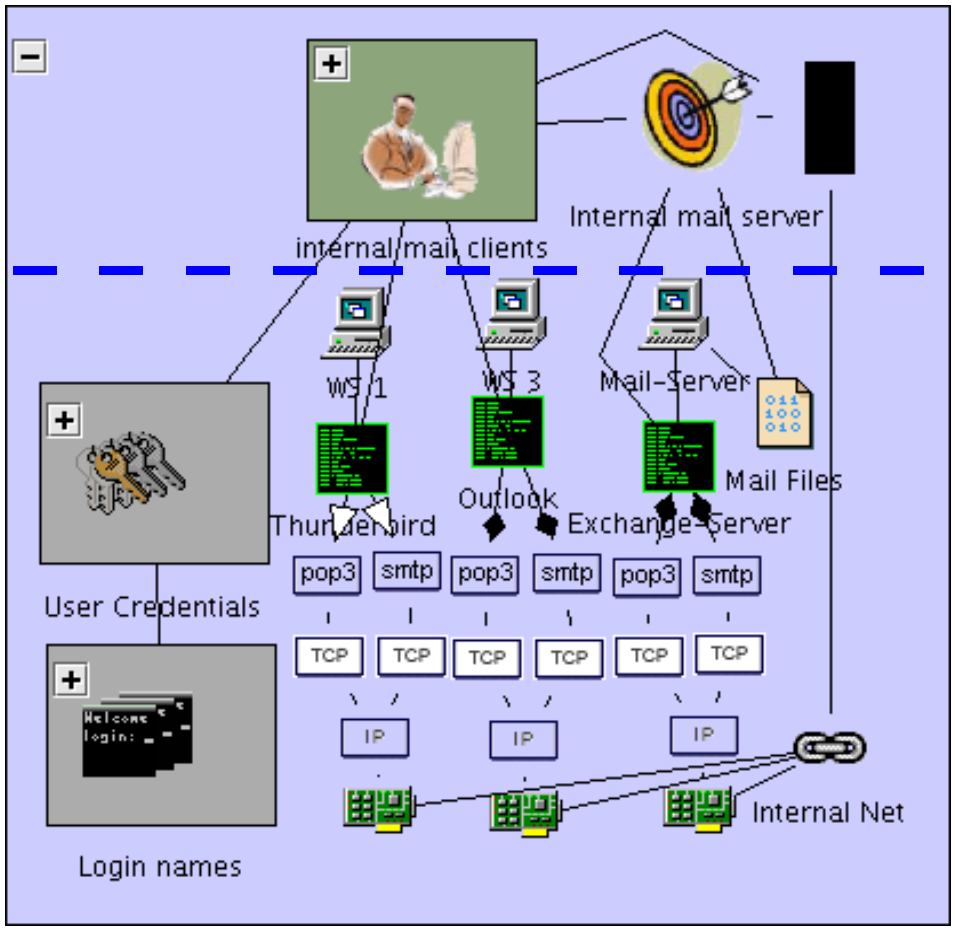
Types of components

Actors	active elements Initiate communication Execute mandatory operations
Mediators	inspect, filter and/or transform data flows According to the policies
Targets	store relevant information
Connectors	communication interfaces between ASs Inert; help with graphing

Simple model example



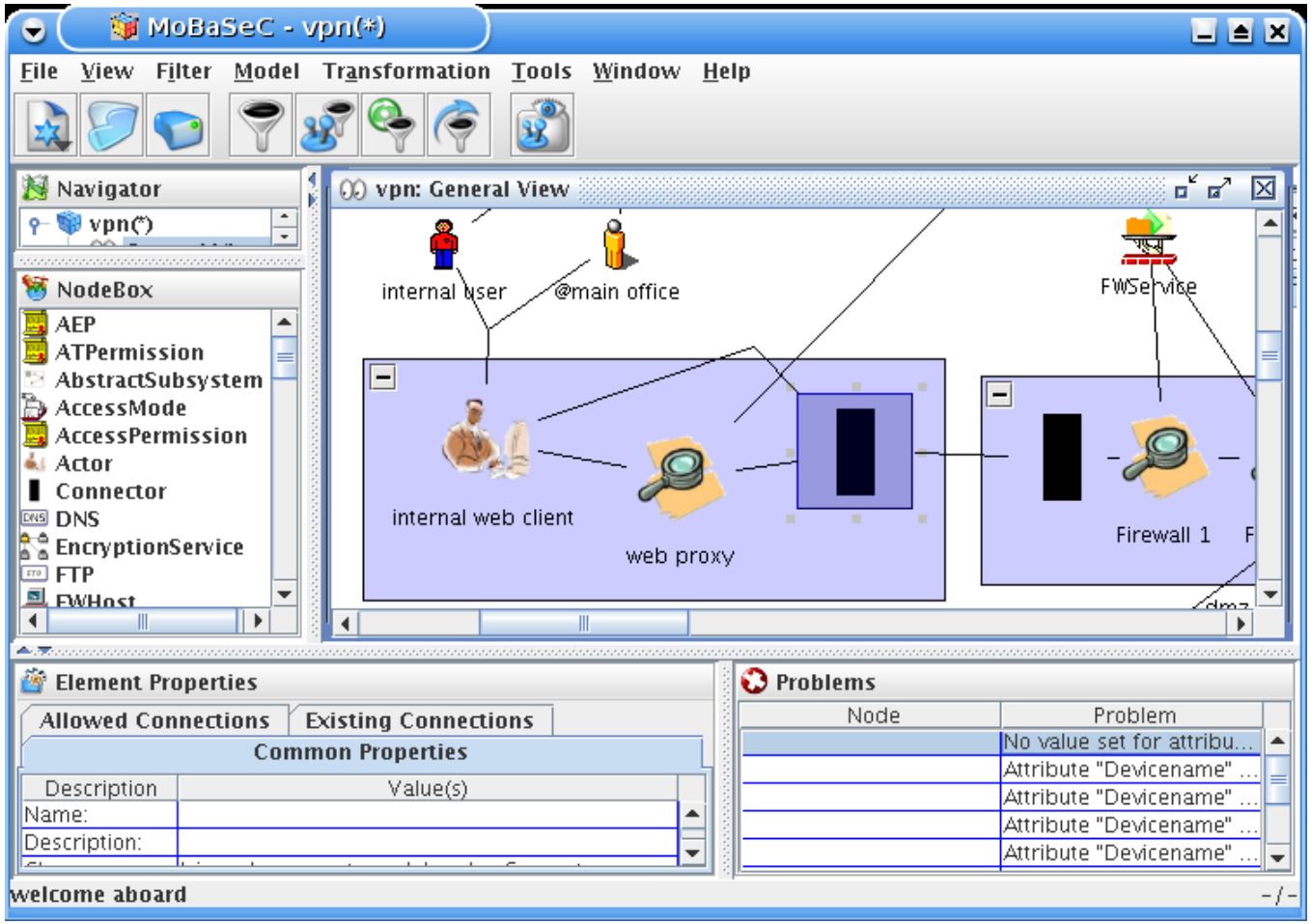
Detailed view of the lowest level



Abstract view

Expanded view (down to the processes and hosts level)

MoBaSeC interface



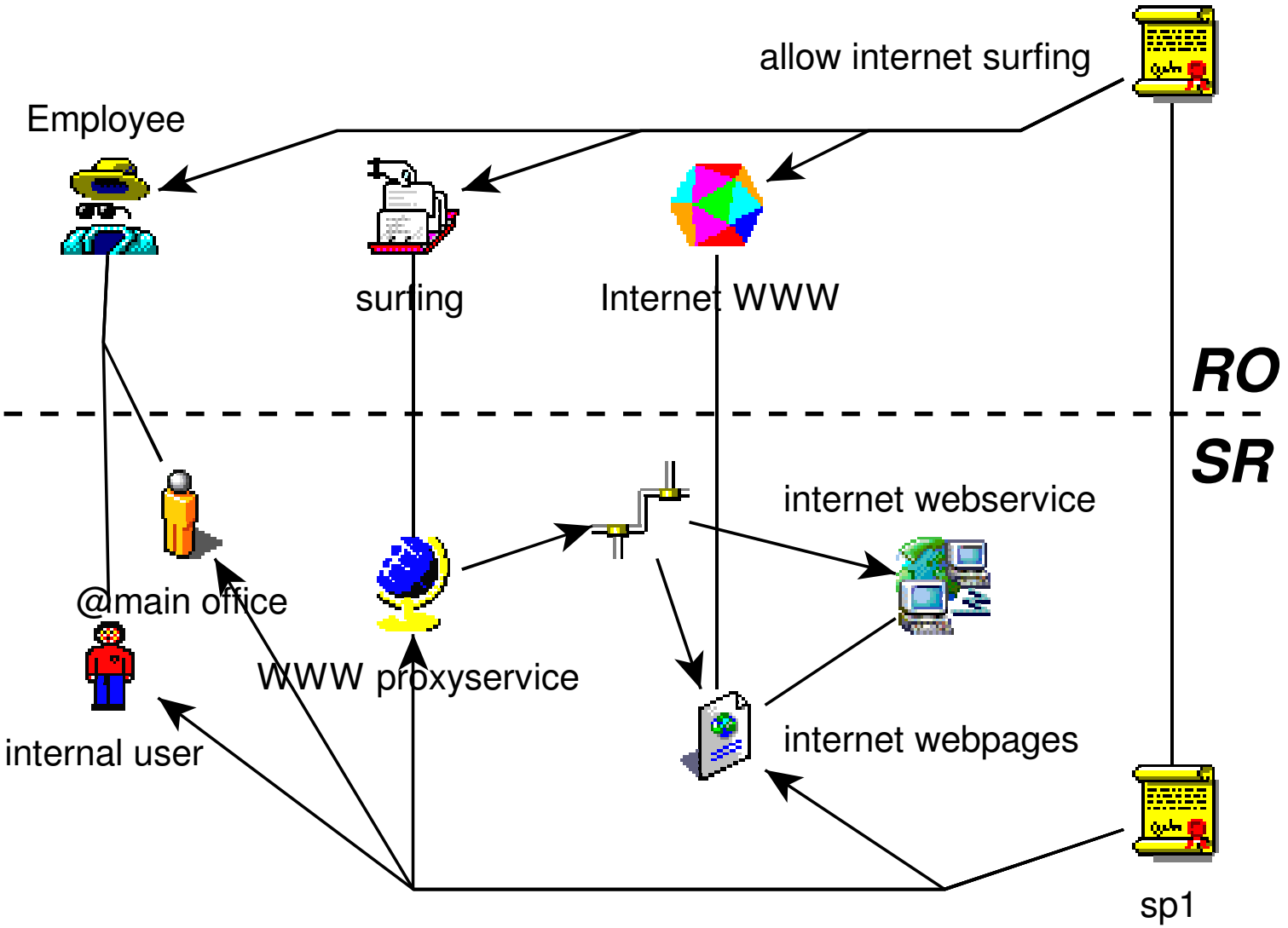
Supporting tool

- Diagram editor to draw models
- Execution of global and local consistency checks
- Automated generation of service configuration parameters
  - From *AccessPermissions*, highest to lower levels
    $\longrightarrow$ authorization policies automatically derived
  - *Back-end* functions translate PH model
    $\longrightarrow$ specific config files (iptables, Kerberos. . .)
  - Checks and formal validation
    $\longrightarrow$ assure compliance with policies

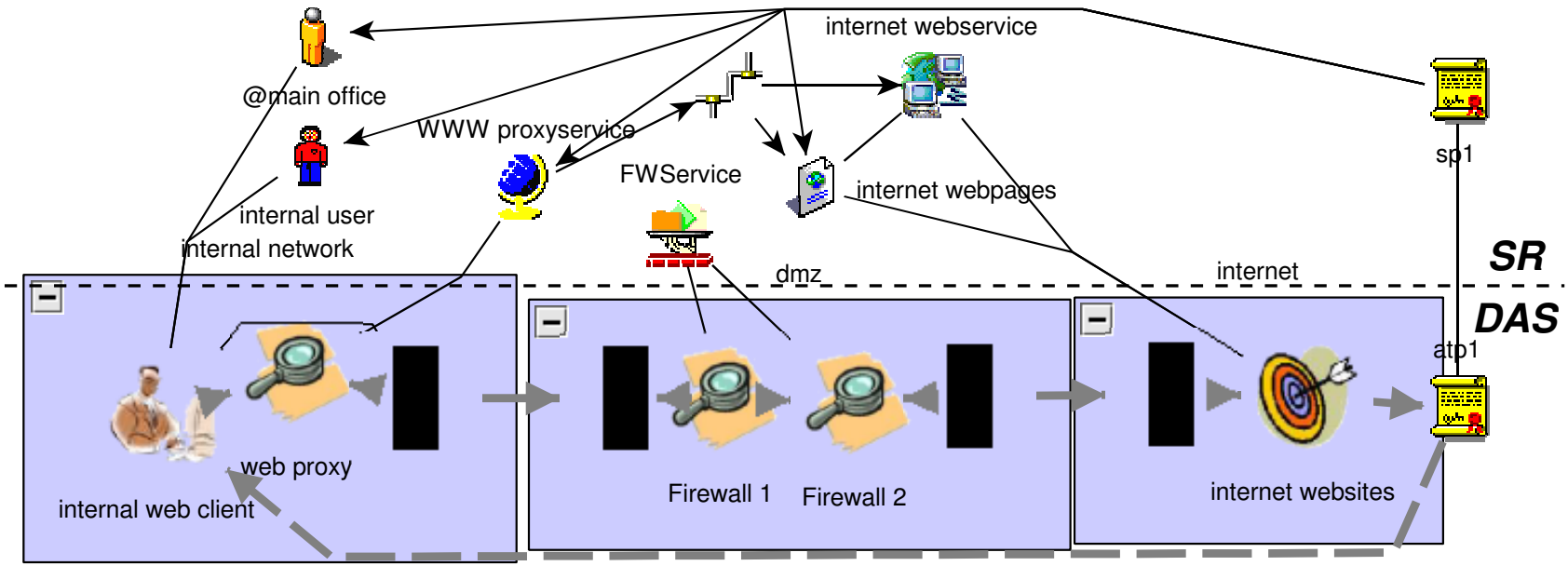
Supporting tool

- Diagram editor to draw models
- Execution of global and local consistency checks
- Automated generation of service configuration parameters
  - From *AccessPermissions*, highest to lower levels
    $\longrightarrow$ authorization policies automatically derived
  - *Back-end* functions translate PH model
    $\longrightarrow$ specific config files (iptables, Kerberos. . .)
  - Checks and formal validation
    $\longrightarrow$ assure compliance with policies

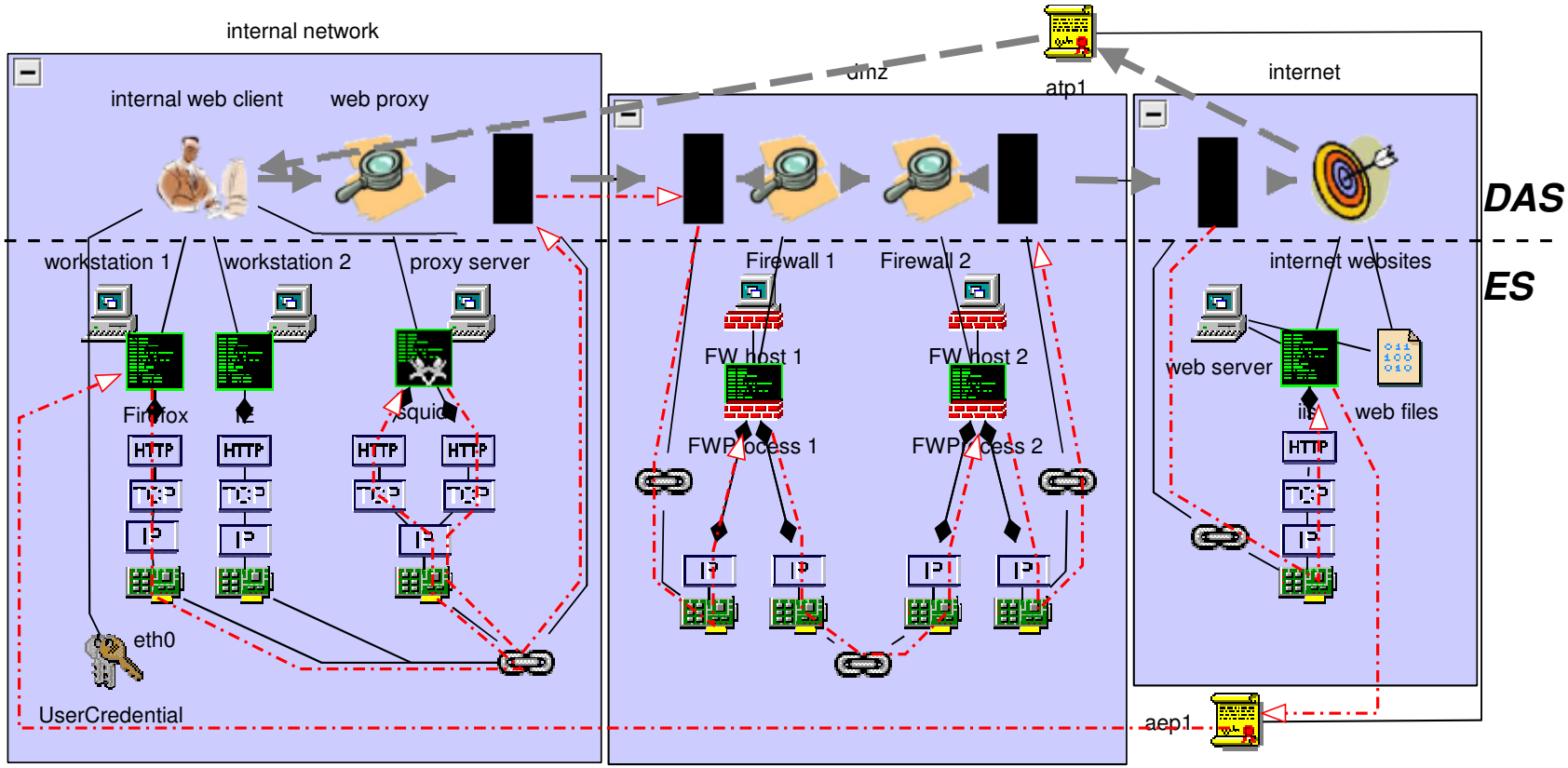
MoBaSeC refinement process - I



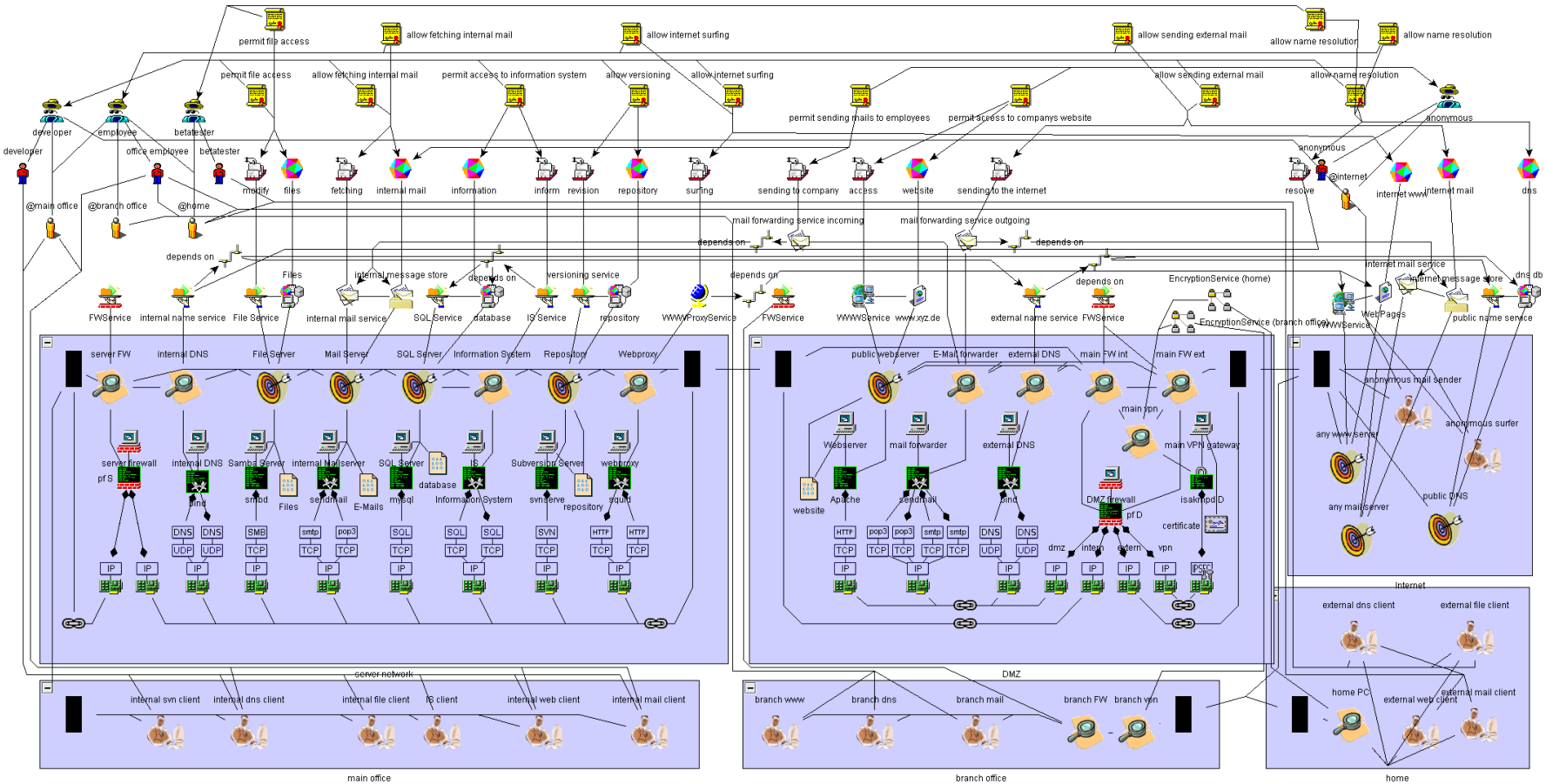
MoBaSeC refinement process - II



MoBaSeC refinement process - III



A larger example



Figures from case studies

scenario	n° of policies at the RO level	n° of network elements (PH/ES)	n° of DAS elements
simple network example	5	95	19
larger network, similar policy	5	540	32
medium-sized network	15	264	49

- Scalability regarding the number of elements
- Policies are the real measure of complexity. . .

Figures from case studies

scenario	n° of policies at the RO level	n° of network elements (PH/ES)	n° of DAS elements
simple network example	5	95	19
larger network, similar policy	5	540	32
medium-sized network	15	264	49

- Scalability regarding the number of elements
- Policies are the real measure of complexity. . .