

---

# **An Introduction to ‘iptables’: Firewall Filtering Aspects**

Course: CS6315 Mobile Systems Security  
Lecturer: Simon Foley

## **William Fitzgerald**

Cork Constraint Computation Centre,  
Department of Computer Science,  
University College Cork,  
Ireland.

Web: [www.williamfitzgerald.net](http://www.williamfitzgerald.net)

Email: [wfitzgerald@4c.ucc.ie](mailto:wfitzgerald@4c.ucc.ie)

February, 2010

# What is it?

- ▷ Background
- Rule Components
- Active Rule-Set
- Case Study
- Summary

*iptables* is a front-end to Netfilter.

Netfilter is a framework that enables:

- Packet filtering (i.e. Firewalling).
- Network Address Translation (NAT).
- Packet mangling.

As a firewall, it is both a stateful and stateless packet filter that is characterised by a sequence of firewall rules against which all packets traversing the firewall are filtered.

Each firewall rule takes the form of a series of conditions representing packet attributes that must be met in order for that rule to be applicable, with a consequent action for the matching packet (accept, drop, log and so forth).

# iptables Rule Components

- Background
- ▷ Rule Components
- Active Rule-Set
- Case Study
- Summary

- iptables* configuration is defined by an ordered set of rules.
- Each *iptables* rule is applied to a chain within a table.
- Each *iptables* rule describes an action to be taken having inspected a packet that matched its filter conditions.

**[Table][Chain][Filter Conditions][Target Action]**

# [Table][Chain][Filter Conditions][Target Action]

- Background
- ▷ Rule Components
- Active Rule-Set
- Case Study
- Summary

A *table* is a classification of common packet handling functionality.

- `filter`: firewall rules.
- `nat`: Network Address Translation (*NAT*).
- `mangle`: specialised packet alteration, for example, QoS.
- `raw`: configure exceptions to connection tracking.

The table under consideration in this lecture is the `filter` table.

# [Table][Chain][Filter Conditions][Target Action]

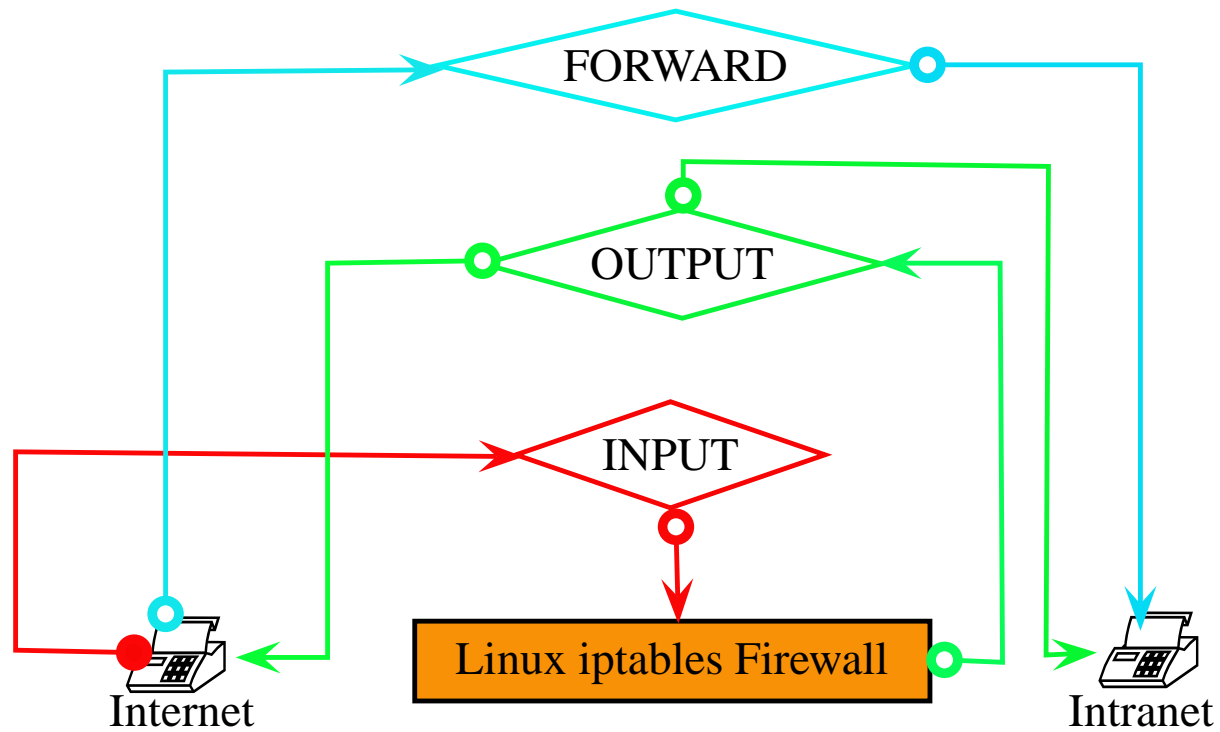
- Background
- ▷ Rule Components
- Active Rule-Set
- Case Study
- Summary

*iptables* provides a mechanism of three separate firewall or filtering (in-built) chains to police various kinds of network traffic:

- INPUT: packets being routed to the firewall device itself.
- OUTPUT: packets being routed from the firewall device itself.
- FORWARD: packets being routed beyond the firewall device.
- User-Defined: human friendly classification. Packets are bound to either INPUT, OUTPUT or FORWARD chains.

# [Table][Chain][Filter Conditions][Target Action]

## Linux iptables (filter table) Packet Traversal



- Background
- ▷ Rule Components
- Active Rule-Set
- Case Study
- Summary

# [Table][Chain][Filter Conditions][Target Action]

- Background
- ▷ Rule Components
- Active Rule-Set
- Case Study
- Summary

There are two approaches when applying a chain policy:

- Deny everything by default*, whereby packets that are not matched by a rule in a chain are then dropped. This approach is recommended as best practice.
- Accept everything by default*, whereby packets that have not been explicitly dropped by rules within a chain are then accepted as a result of the default policy.

# Chain Commands

Background  
▷ Rule Components  
Active Rule-Set  
Case Study  
Summary

- P, --policy
- F, --flush
- Z, --zero
- A, --append
- D, --delete
- I, --insert
- R, --replace
- N, --new-chain
- X, --delete-chain
- E, --rename-chain

Detailed descriptions can be found in the `iptables(8)` - Linux man page

# [Table][Chain][Filter Conditions][Target Action]

- Background
- ▷ Rule Components
- Active Rule-Set
- Case Study
- Summary

Packets are matched against a set of filter conditions (or packet criteria).

Each packet header that the firewall intercepts will be inspected according to the rule conditions specified.

# [Table][Chain][Filter Conditions][Target Action]

- Background
- ▷ Rule Components
- Active Rule-Set
- Case Study
- Summary

Example set of filter conditions:

- `-s, --source`: Source IP Address filtering.
- `-d, --destination`: Destination IP Address filtering.
- `-p, --protocol`: Protocol filtering.
- `-i, --in-interface`: Inbound interface filtering.
- `-o, --out-interface`: Outbound interface filtering.
- `--tcp-flags`: Flag attribute filtering.
- `-m limit --limit`: Rate of packet flow filtering.
- `-m state --state`: Stateful connection filtering.
- `-m string --string`: Application layer payload filtering.
- `-m layer7 --l7proto`: Pre-determined macro application layer payload filtering.

Consult the `iptables(8)` - Linux man page for additional information and filter conditions.

# [Table][Chain][Filter Conditions][Target Action]

- Background
- ▷ Rule Components
- Active Rule-Set
- Case Study
- Summary

*iptables* provides a mechanism of packet authorisations.

When a filter condition matches a packet traversing a particular chain, a firewall target action specifies the fate of that packet.

Example target actions:

- ACCEPT: permit the packet.
- DROP: block the packet.
- REJECT: block the packet but send an appropriate response packet.
- LOG: record the packet.
- RECORD: Continue processing packet within calling chain.

Consult the `iptables(8)` - Linux man page for additional target actions.

# View the Active Set of iptables Rules

- Background
- Rule Components
- ▷ Active Rule-Set
- Case Study
- Summary

*iptables* commands:

- L, --list
- v, --verbose
- n, --numeric
- x, --exact

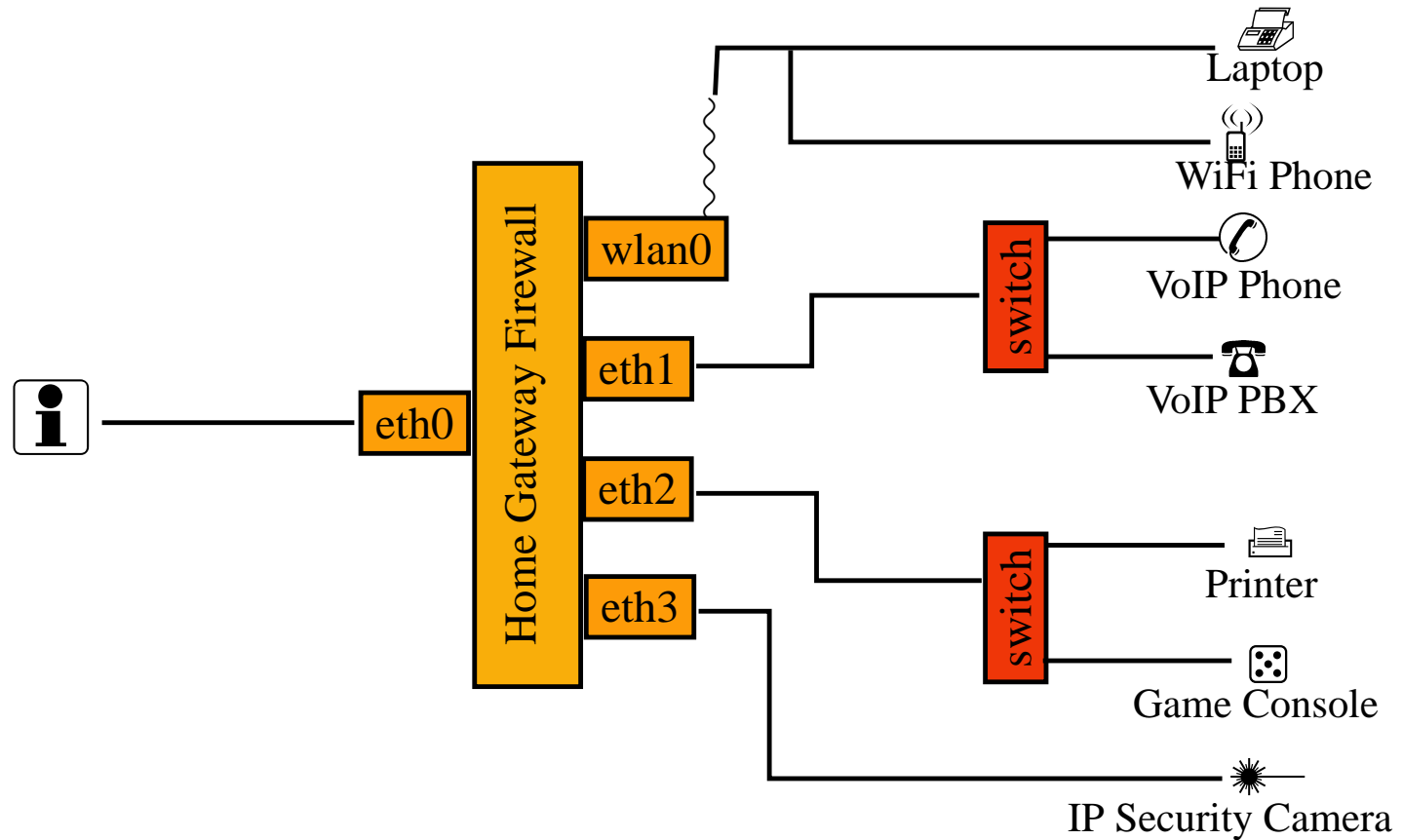
`sudo iptables -L -v`

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)								
pkts	bytes	target	prot	opt	in	out	source	destination
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)								
pkts	bytes	target	prot	opt	in	out	source	destination
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)								
pkts	bytes	target	prot	opt	in	out	source	destination

Note, the above iptables configuration has no active firewall rules and is configured with an *accept everything by default* policy!

# Example: Firewall Control for a Home Area Network

- Background
- Rule Components
- Active Rule-Set
- ▷ Case Study
- Summary



# Assumptions: Firewall Control for a Home Area Network

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

- NAT configured correctly.
- Port-forwarding configured correctly.
- VLAN's and firewall zones configured correctly.

Note, for the most part, you do not need to understand these concepts to understand the firewalling aspects.

# Example: Network Security Policy for Home Area Network

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

Policy ID	Description
nsp-1	Block RFC-based anti-bogon IP Spoofing attempts.
nsp-2	Block external network port scanning.
nsp-3	Block Denial of Service Attacks.
nsp-4	Permit administrator (laptop IP) SSH access to firewall.
nsp-5	Permit game console access (for example xbox).
nsp-6	Permit LAN users access to network printer.
nsp-7	Permit VoIP PBX and Phone access.
nsp-8	Permit LAN users access to HTTP(S),SSH,IMAP,SMTP.
nsp-9	Permit LAN users access to internal security camera.
nsp-10	Block all other traffic.

# Initial Set-Up & General House Keeping

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

```
#!/bin/sh
```

```
# Define variables
```

```
ipt=' '/sbin/iptables''
```

```
LAN=' '192.168.1.0/24''
```

```
WAN=' '1.2.3.4''
```

```
Laptop=' '192.168.1.10''
```

```
Phone=' '192.168.1.12''
```

```
PBX=' '192.168.1.13''
```

```
Printer=' '192.168.1.14''
```

```
XBox=' '192.168.1.15''
```

```
Cam=' '192.168.1.16''
```

```
# Flush Rules and Zero counters.
```

```
ipt --flush
```

```
ipt --zero
```

# Example NSP-10: Configure the Default Policy

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

Network security policy goal, nsp-10, requires a configuration that *denies everything by default*.

```
# Drop everything by default.  
$ipt -P INPUT DROP  
$ipt -P FORWARD DROP  
$ipt -P OUTPUT DROP
```

It is also advisable to assert this requirement within the rule-set itself. Note, these rules must be the last three rules of the iptables firewall configuration.

```
# Default drop rules.  
$ipt -A INPUT -j DROP  
$ipt -A OUTPUT -j DROP  
$ipt -A FORWARD -j DROP
```

# Example NSP-1: Anti-Bogon IP Spoofing Control

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

Example IP spoof using nmap:

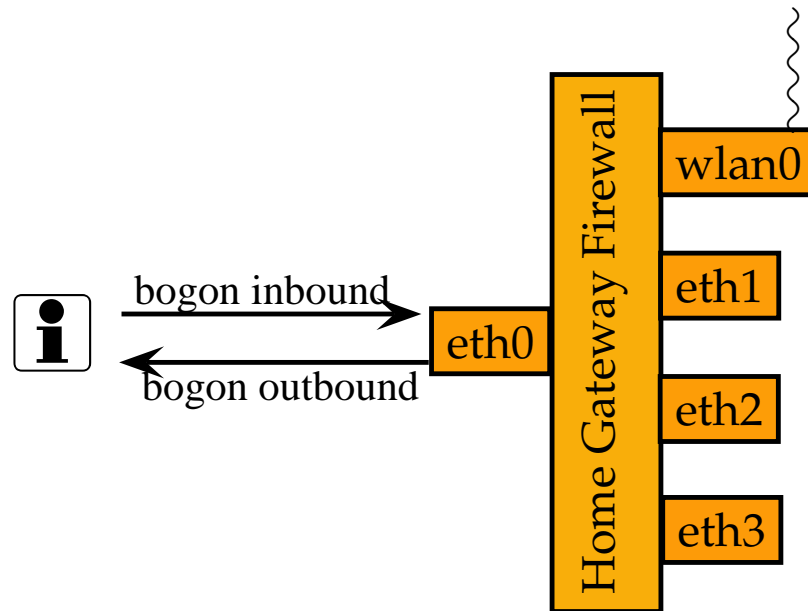
□ `nmap -S some.google.comIP some.microsoft.comIP -sN`

While these sort of scan do not provide the attacker useful information, they may have other detrimental implications!

# Example NSP-1: Anti-Bogon IP Spoofing Control

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

Bogon filtering can be applied to the External WAN interface.



# Example NSP-1: Anti-Bogon IP Spoofing Control

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

An IP range that should never cross networks (exception VPN's).  
RFC 3330 recommends filtering of 18 IP address ranges.

```
# Drop non-routable spoofed IP addresses.
```

```
0.0.0.0/8  
10.0.0.0/8  
14.0.0.0/8  
24.0.0.0/8  
39.0.0.0/8  
127.0.0.0/8  
128.0.0.0/16  
169.254.0.0/16  
172.16.0.0/12  
191.255.0.0/16  
192.0.0.0/24  
192.0.2.0/24  
192.88.99.0/24  
192.168.0.0/16  
198.18.0.0/15  
223.255.255.0/24  
224.0.0.0/4  
240.0.0.0/4
```

# Example NSP-1: Anti-Bogon IP Spoofing Control

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

- A bogon IP address range is one that should never cross networks (exception VPN's).
- RFC 3330 recommends filtering of 18 IP address ranges.
- **Should be applied to INPUT, OUTPUT and FORWARD (inbound and outbound) chains.**
- Should be applied to source and destination IP addresses.
- Total set of RFC3330 rules is 144.

```
# Drop non-routable spoofed IP addresses.  
$ipt -A INPUT -i eth0 -s 192.168.0.0/16 -j DROP  
$ipt -A INPUT -o eth0 -d 192.168.0.0/16 -j DROP  
$ipt -A OUTPUT -i eth0 -s 192.168.0.0/16 -j DROP  
$ipt -A OUTPUT -o eth0 -d 192.168.0.0/16 -j DROP  
$ipt -A FORWARD -i eth0 -s 192.168.0.0/16 -j DROP  
$ipt -A FORWARD -i eth0 -d 192.168.0.0/16 -j DROP  
$ipt -A FORWARD -o eth0 -s 192.168.0.0/16 -j DROP  
$ipt -A FORWARD -o eth0 -d 192.168.0.0/16 -j DROP
```

# Example NSP-1: Anti-Bogon IP Spoofing Control

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

- A bogon IP address range is one that should never cross networks (exception VPN's).
- RFC 3330 recommends proper filtering of 18 IP address ranges.
- Should be applied to INPUT, OUTPUT and FORWARD (inbound and outbound) chains.
- **Should be applied to source and destination IP addresses.**
- Total set of RFC3330 rules is 144.

```
# Drop non-routable spoofed IP addresses.  
$ipt -A INPUT -i eth0 -s 192.168.0.0/16 -j DROP  
$ipt -A INPUT -i eth0 -d 192.168.0.0/16 -j DROP  
$ipt -A OUTPUT -o eth0 -s 192.168.0.0/16 -j DROP  
$ipt -A OUTPUT -o eth0 -d 192.168.0.0/16 -j DROP  
$ipt -A FORWARD -i eth0 -s 192.168.0.0/16 -j DROP  
$ipt -A FORWARD -i eth0 -d 192.168.0.0/16 -j DROP  
$ipt -A FORWARD -o eth0 -s 192.168.0.0/16 -j DROP  
$ipt -A FORWARD -o eth0 -d 192.168.0.0/16 -j DROP
```

# Example NSP-2: Nmap Scan Probe Control

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

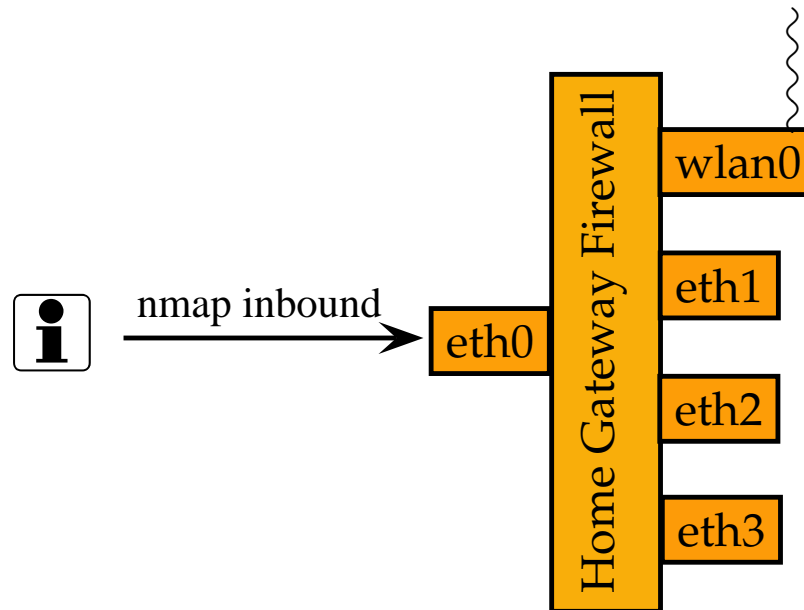
Example of (simple) nmap port scans:

- SYN Scan: 'nmap -sS'
- FIN Scan: 'nmap -sF \$WAN'
- XMAS Scan: 'nmap -sX \$WAN'
- NULL Scan: 'nmap -sN \$WAN'

# Example NSP-2: Nmap Scan Probe Control

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

Network port scan filtering can be applied to the external WAN interface.



# Example NSP-2: Nmap Scan Probe Control

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

```
# Excerpt Fragment of anti-Nmap port scan filters.  
$ipt -A -p tcp --tcp-flags ACK,FIN FIN -j NmapScan  
$ipt -A -p tcp --tcp-flags ACK,PSH PSH -j NmapScan  
$ipt -A -p tcp --tcp-flags ACK,URG URG -j NmapScan  
$ipt -A -p tcp --tcp-flags FIN,RST FIN,RST -j NmapScan  
$ipt -A -p tcp --tcp-flags SYN,FIN SYN,FIN -j NmapScan  
$ipt -A -p tcp --tcp-flags SYN,RST SYN,RST -j NmapScan  
$ipt -A -p tcp --tcp-flags ALL ALL -j NmapScan  
$ipt -A -p tcp --tcp-flags ALL NONE -j NmapScan  
$ipt -A -p tcp --tcp-flags ALL FIN,PSH,URG -j NmapScan  
$ipt -A -p tcp --tcp-flags ALL SYN,FIN,PSH,URG -j NmapScan  
$ipt -A -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG -j NmapScan  
$ipt -A NmapScan -j DROP
```

# Example NSP-2: Nmap Scan Probe Control

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

Example of a user-defined chain.

```
# Excerpt Fragment of anti-Nmap port scan filters.
$Ipt -A -p tcp --tcp-flags ACK,FIN FIN -j NmapScan
$Ipt -A -p tcp --tcp-flags ACK,PSH PSH -j NmapScan
$Ipt -A -p tcp --tcp-flags ACK,URG URG -j NmapScan
$Ipt -A -p tcp --tcp-flags FIN,RST FIN,RST -j NmapScan
$Ipt -A -p tcp --tcp-flags SYN,FIN SYN,FIN -j NmapScan
$Ipt -A -p tcp --tcp-flags SYN,RST SYN,RST -j NmapScan
$Ipt -A -p tcp --tcp-flags ALL ALL -j NmapScan
$Ipt -A -p tcp --tcp-flags ALL NONE -j NmapScan
$Ipt -A -p tcp --tcp-flags ALL FIN,PSH,URG -j NmapScan
$Ipt -A -p tcp --tcp-flags ALL SYN,FIN,PSH,URG -j NmapScan
$Ipt -A -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG -j NmapScan
$Ipt -A NmapScan -j DROP
```

# Example NSP-2: Nmap Scan Probe Control

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

How to interpret the TCP flag filter condition.

```
# Excerpt Fragment of anti-Nmap port scan filters.
$ipt -A -p tcp --tcp-flags ACK,FIN FIN -j NmapScan
$ipt -A -p tcp --tcp-flags ACK,PSH PSH -j NmapScan
$ipt -A -p tcp --tcp-flags ACK,URG URG -j NmapScan
$ipt -A -p tcp --tcp-flags FIN,RST FIN,RST -j NmapScan
$ipt -A -p tcp --tcp-flags SYN,FIN SYN,FIN -j NmapScan
$ipt -A -p tcp --tcp-flags SYN,RST SYN,RST -j NmapScan
$ipt -A -p tcp --tcp-flags ALL ALL -j NmapScan
$ipt -A -p tcp --tcp-flags ALL NONE -j NmapScan
$ipt -A -p tcp --tcp-flags ALL FIN,PSH,URG -j NmapScan
$ipt -A -p tcp --tcp-flags ALL SYN,FIN,PSH,URG -j NmapScan
$ipt -A -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG -j NmapScan
$ipt -A NmapScan -j DROP
```

# Example NSP-2: Nmap Scan Probe Control

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

The first string of flags is a list of flags you want to examine.

```
# Excerpt Fragment of anti-Nmap port scan filters.
$ipt -A -p tcp --tcp-flags ACK,FIN FIN -j NmapScan
$ipt -A -p tcp --tcp-flags ACK,PSH PSH -j NmapScan
$ipt -A -p tcp --tcp-flags ACK,URG URG -j NmapScan
$ipt -A -p tcp --tcp-flags FIN,RST FIN,RST -j NmapScan
$ipt -A -p tcp --tcp-flags SYN,FIN SYN,FIN -j NmapScan
$ipt -A -p tcp --tcp-flags SYN,RST SYN,RST -j NmapScan
$ipt -A -p tcp --tcp-flags ALL ALL -j NmapScan
$ipt -A -p tcp --tcp-flags ALL NONE -j NmapScan
$ipt -A -p tcp --tcp-flags ALL FIN,PSH,URG -j NmapScan
$ipt -A -p tcp --tcp-flags ALL SYN,FIN,PSH,URG -j NmapScan
$ipt -A -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG -j NmapScan
$ipt -A NmapScan -j DROP
```

# Example NSP-2: Nmap Scan Probe Control

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

The second string of flags tells which one(s) should be set.

```
# Excerpt Fragment of anti-Nmap port scan filters.
$Ipt -A -p tcp --tcp-flags ACK,FIN FIN -j NmapScan
$Ipt -A -p tcp --tcp-flags ACK,PSH PSH -j NmapScan
$Ipt -A -p tcp --tcp-flags ACK,URG URG -j NmapScan
$Ipt -A -p tcp --tcp-flags FIN,RST FIN,RST -j NmapScan
$Ipt -A -p tcp --tcp-flags SYN,FIN SYN,FIN -j NmapScan
$Ipt -A -p tcp --tcp-flags SYN,RST SYN,RST -j NmapScan
$Ipt -A -p tcp --tcp-flags ALL ALL -j NmapScan
$Ipt -A -p tcp --tcp-flags ALL NONE -j NmapScan
$Ipt -A -p tcp --tcp-flags ALL FIN,PSH,URG -j NmapScan
$Ipt -A -p tcp --tcp-flags ALL SYN,FIN,PSH,URG -j NmapScan
$Ipt -A -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG -j NmapScan
$Ipt -A NmapScan -j DROP
```

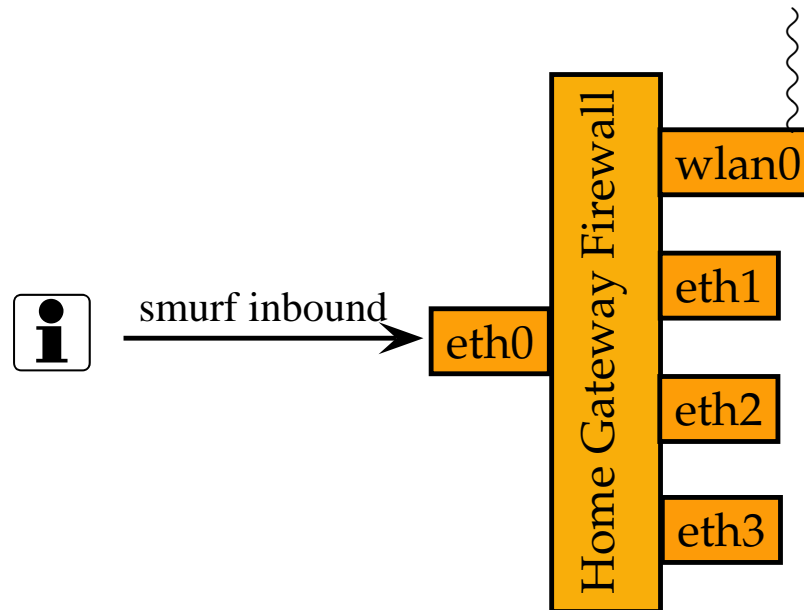
Notice the ALL and NONE operators.

```
$Ipt -A -p tcp --tcp-flags ALL NONE -j NmapScan
≡
$Ipt -A -p tcp --tcp-flags FIN,SYN,RST,PSH,ACK,URG NONE -j NmapScan
```

# Example NSP-3: Prevent Denial of Service Attacks

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

ICMP filtering can be applied to the external WAN interface.



# Example NSP-3: Prevent Denial of Service Attacks

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

A ICMP-flood attack (aka Smurf Attack) is a *denial of service attack that sends a host more ICMP echo request (“ping”) packets than the protocol implementation can handle* [rfc2828].

```
# Mitigate or minimise external Smurf attacks on the firewall.
```

```
$iptables -A INPUT -eth0 -p icmp -m limit --limit 1/s --limit-burst 1 -j ACCEPT
```

```
$iptables -A INPUT -p icmp -j DROP
```

# Example NSP-3: Prevent Denial of Service Attacks

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

A ICMP-flood attack (aka Smurf Attack) is a *denial of service attack that sends a host more ICMP echo request (“ping”) packets than the protocol implementation can handle* [rfc2828].

```
# Mitigate or minimise external Smurf attacks on the firewall.
```

```
$ipt -A INPUT -eth0 -p icmp -m limit --limit 1/s --limit-burst 1 -j ACCEPT
```

```
$ipt -A INPUT -p icmp -j DROP
```

- Filter ICMP protocol packets.

# Example NSP-3: Prevent Denial of Service Attacks

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

A ICMP-flood attack (aka Smurf Attack) is a *denial of service attack that sends a host more ICMP echo request (“ping”) packets than the protocol implementation can handle* [rfc2828].

```
# Mitigate or minimise external Smurf attacks on the firewall.
```

```
$ipt -A INPUT -eth0 -p icmp -m limit --limit 1/s --limit-burst 1 -j ACCEPT
```

```
$ipt -A INPUT -p icmp -j DROP
```

- `--limit`: Maximum average matching rate set for /second, /minute, /hour, or /day suffix; the default is 3/hour.
- `--limit-burst`: Maximum initial number of packets to match.

# Example NSP-3: Prevent Denial of Service Attacks

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

A ICMP-flood attack (aka Smurf Attack) is a *denial of service attack that sends a host more ICMP echo request (“ping”) packets than the protocol implementation can handle* [rfc2828].

```
# Mitigate or minimise external Smurf attacks on the firewall.
```

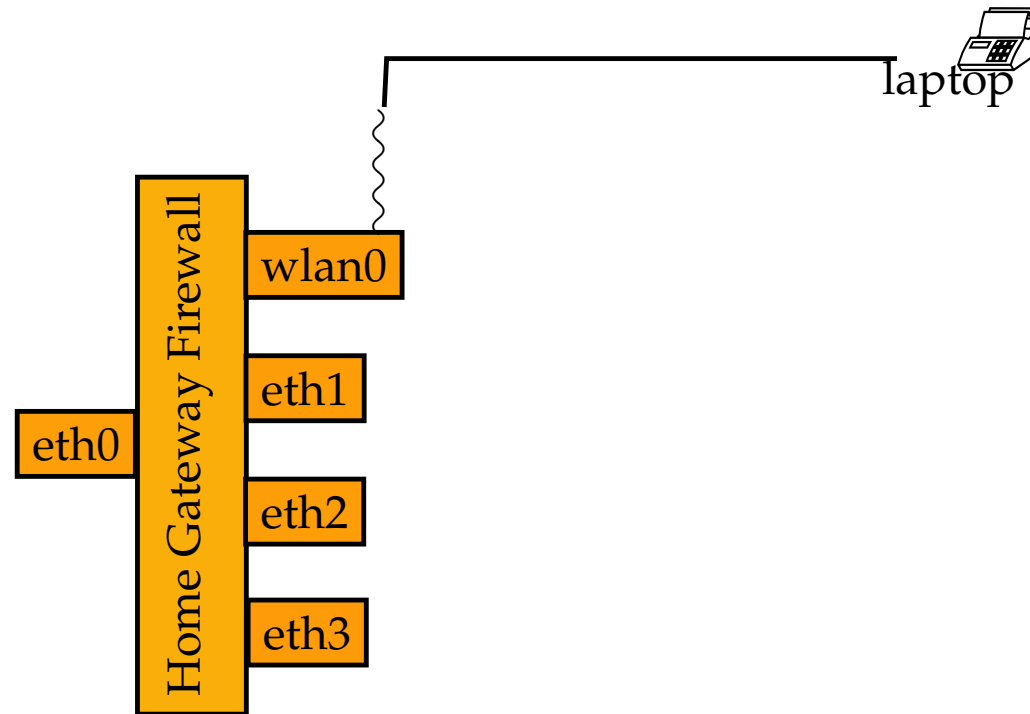
```
$ipt -A INPUT -eth0 -p icmp -m limit --limit 1/s --limit-burst 1 -j ACCEPT
```

```
$ipt -A INPUT -p icmp -j DROP
```

- Drop ICMP packets in breach of the rate limiting threshold.

# Example NSP-4: Permit Administrator SSH Access To Firewall

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary



# Example NSP-4: Permit Administrator SSH Access To Firewall

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

Internal remote firewall administration via SSH by administrator only.

```
# Permit secure configuration communication to firewall.  
$iptables -A INPUT -wlan0 -p tcp -s $Laptop -d 192.168.1.1 --dport 22 -j ACCEPT  
$iptables -A OUTPUT -wlan0 -p tcp -s 192.168.1.1 --sport 22 -d $Laptop -j ACCEPT
```

# Example NSP-4: Permit Administrator SSH Access To Firewall

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

Internal remote firewall administration via SSH by administrator only.

```
# Permit secure configuration communication to firewall.
```

```
$iptables -A INPUT -wlan0 -p tcp -s $Laptop -d 192.168.1.1 --dport 22 -j ACCEPT
```

```
$iptables -A OUTPUT -wlan0 -p tcp -s 192.168.1.1 --sport 22 -d $Laptop -j ACCEPT
```

- Remember traffic is bi-directional.
- An OUTPUT chain rule is required. Otherwise the default OUTPUT chain policy will take effect.

# Example NSP-4: Permit Administrator SSH Access To Firewall

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

Internal remote firewall administration via SSH by administrator only.

```
# Permit secure configuration communication to firewall.
```

```
$iptables -A INPUT -wlan0 -p tcp -s $Laptop -d 192.168.1.1 --dport 22 -j ACCEPT
```

```
$iptables -A OUTPUT -wlan0 -p tcp -s 192.168.1.1 --sport 22 -d $Laptop -j ACCEPT
```

- SSH traffic directed towards the firewall itself.
- Note, both inbound and outbound rule filters swap the source & destination IP addresses and source & destination ports.

# Example NSP-4: Permit Administrator SSH Access To Firewall

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

Internal remote firewall administration via SSH by administrator only.

```
# Permit secure configuration communication to firewall.  
$iptables -A INPUT -wlan0 -p tcp -s $Laptop -d 192.168.1.1 --dport 22 -j ACCEPT  
$iptables -A OUTPUT -wlan0 -p tcp -s 192.168.1.1 --sport 22 -d $Laptop -j ACCEPT
```

- ❑ Restrict the interface in conjunction with the source IP address. Helps reduce IP spoofing from other internal subnets.
- ❑ Remember defense in depth!
- ❑ Note, in practice, it is not advisable to perform a firmware upgrade via WiFi and should be done via Ethernet port.

# Example NSP-4: Permit Administrator SSH Access To Firewall

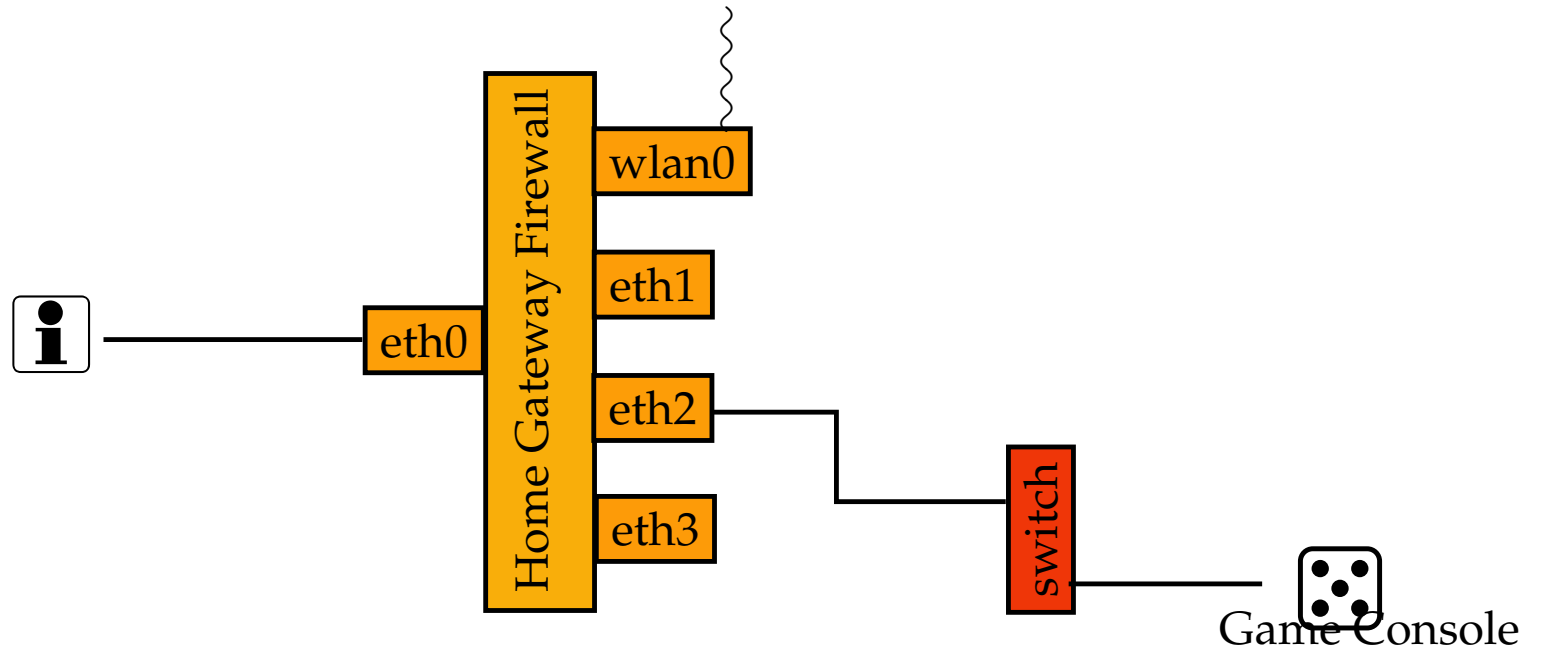
Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

Previous firewall SSH (stateless) rules can be replaced by the following **stateful** rules.

```
# Permit secure configuration communication to firewall.  
$iptables -A INPUT -wlan0 -p tcp -s $Laptop -d 192.168.1.1 --dport 22  
    -m state --state NEW,ESTABLISHED -j ACCEPT  
$iptables -A OUTPUT -wlan0 -p tcp -s 192.168.1.1 -m state --state ESTABLISHED -j ACCEPT
```

# Example NSP-5: Permit Xbox external access

- Background
- Rule Components
- Active Rule-Set
- ▷ Case Study
- Summary



# Example NSP-5: Permit Xbox external access

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

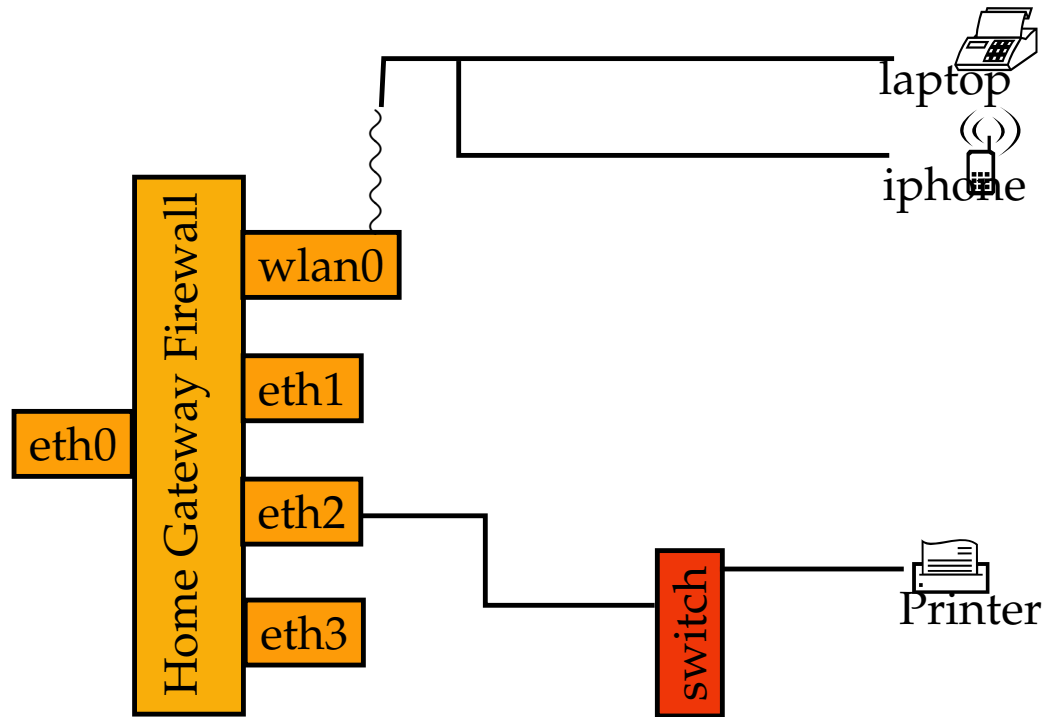
Xbox-live server communicates using TCP port 3074 and UDP ports 3074 and 88.

```
# Permit XBox to make External connections
$Ipt -A FORWARD -i eth2 -p tcp -s $XBox --dports 3074
    -m state --state NEW, ESTABLISHED -j ACCEPT
$Ipt -A FORWARD -i eth2 -p udp -s $XBox -m multiport --dports 88,3074
    -m state --state NEW, ESTABLISHED -j ACCEPT
```

- Note, only Intranet to Internet stateful rules have been defined.
- Later slides will show a generic set of stateful inbound rules to manage the correct return traffic flows.

# Example NSP-6: Provide Printer Access

- Background
- Rule Components
- Active Rule-Set
- ▷ Case Study
- Summary



# Example NSP-6: Provide Printer Access

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

```
# Permit LAN users access to network printer.
$Ipt -A FORWARD -i wlan0 -p tcp -s $Laptop -d $Printer --dports 9100
-m state --state NEW, ESTABLISHED -j ACCEPT
$Ipt -A FORWARD -i wlan0 -p tcp -s $Mobile -d $Printer --dports 9100
-m state --state NEW, ESTABLISHED -j ACCEPT
$Ipt -A FORWARD -i eth2 -p tcp -s --sports 9100 $Printer -d $Laptop
-m state --state ESTABLISHED -j ACCEPT
$Ipt -A FORWARD -i eth2 -p tcp -s $Printer --sports 9100 -d $Mobile
-m state --state ESTABLISHED -j ACCEPT
```

# Example NSP-6: Provide Printer Access

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

```
# Permit LAN users access to network printer.
$Ipt -A FORWARD -i wlan0 -p tcp -s $Laptop -d $Printer --dports 9100
-m state --state NEW, ESTABLISHED -j ACCEPT
$Ipt -A FORWARD -i wlan0 -p tcp -s $Mobile -d $Printer --dports 9100
-m state --state NEW, ESTABLISHED -j ACCEPT
$Ipt -A FORWARD -i eth2 -p tcp -s --sports 9100 $Printer -d $Laptop
-m state --state ESTABLISHED -j ACCEPT
$Ipt -A FORWARD -i eth2 -p tcp -s $Printer --sports 9100 -d $Mobile
-m state --state ESTABLISHED -j ACCEPT
```

- Access to the printer is restricted to the laptop and mobile phone. There is no reason why the PBS system or security camera for example to communicate with the printer.

# Example NSP-6: Provide Printer Access

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

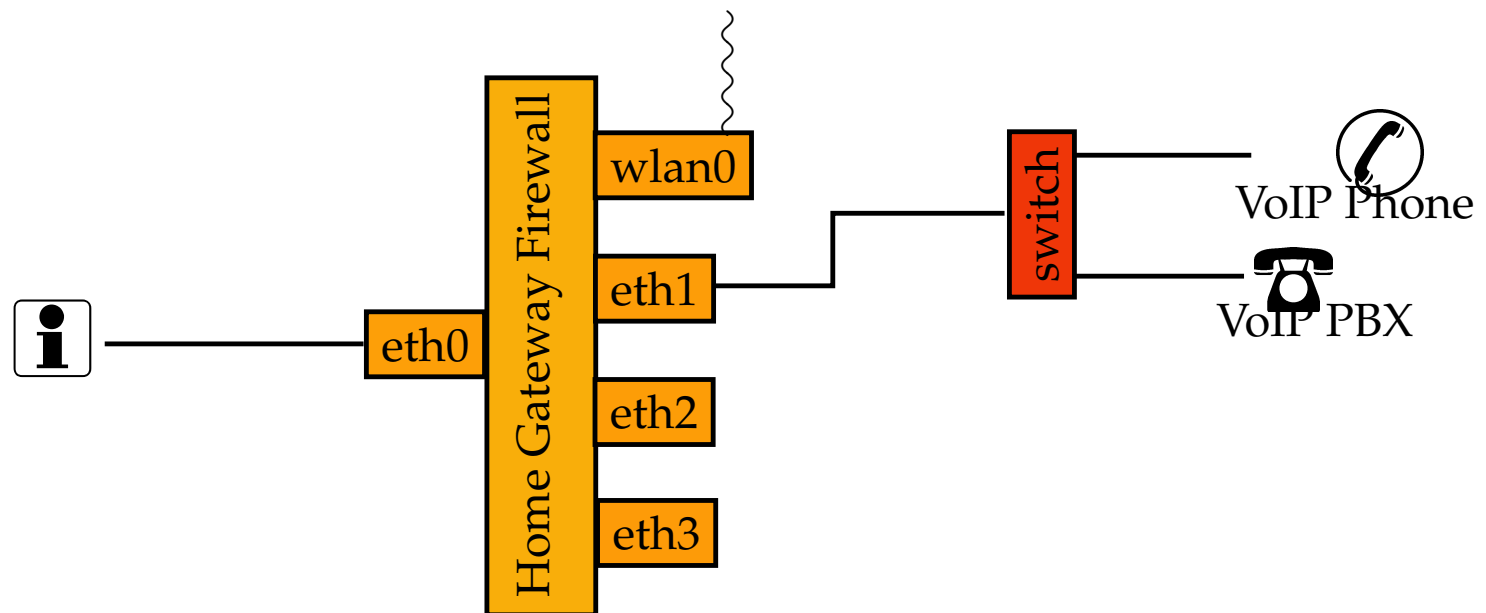
```
# Permit LAN users access to network printer.  
$ipt -A FORWARD -i wlan0 -p tcp -s $Laptop -d $Printer --dports 9100  
-m state --state NEW, ESTABLISHED -j ACCEPT  
$ipt -A FORWARD -i wlan0 -p tcp -s $Mobile -d $Printer --dports 9100  
-m state --state NEW, ESTABLISHED -j ACCEPT  
$ipt -A FORWARD -i eth2 -p tcp -s --sports 9100 $Printer -d $Laptop  
-m state --state ESTABLISHED -j ACCEPT  
$ipt -A FORWARD -i eth2 -p tcp -s $Printer --sports 9100 -d $Mobile  
-m state --state ESTABLISHED -j ACCEPT
```

- Printer should not be initiating SYN packets, therefore it is controlled with the **ESTABLISHED** state.

# Example NSP-7: Permit VoIP PBX and Phone access

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

The PBX server is similar to a proxy server: SIP clients (VoIP phones), register with the PBX server, and when they wish to make a call the PBX will establish the connection. The PBX has a directory of all phones and their corresponding SIP address and thus is able to connect an internal call or route an external call.



# Example NSP-7: Permit VoIP PBX and Phone access

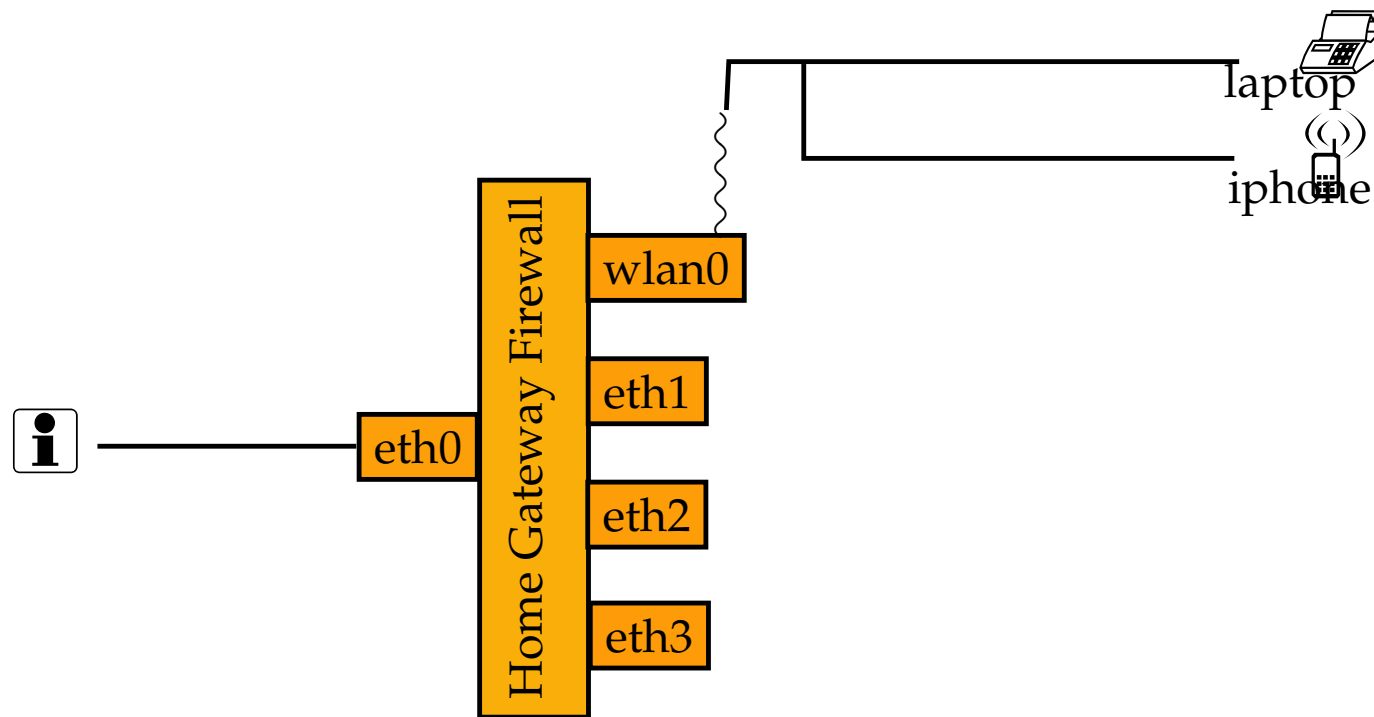
Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

```
# Permit VoIP traffic outbound.  
$ipt -A FORWARD -i eth1 -p tcp -s $PBX --dport 5060  
-m state --state NEW, ESTABLISHED -j ACCEPT  
$ipt -A FORWARD -i eth1 -p udp -s $PBX --dport 5060  
-m state --state NEW, ESTABLISHED -j ACCEPT  
$ipt -A FORWARD -i eth1 -p udp -s $Phone --dport 10000:20000  
-m state --state NEW, ESTABLISHED -j ACCEPT
```

- PBX communicates with a SIP provider (example sip.blueface.ie) typically over TCP port 5060 but can also use UDP port 5060.
- Phone to phone communication encapsulates RTP traffic over UDP port range 10000 to 20000.

# Example NSP-8: Provide LAN users typical Internet access

- Background
- Rule Components
- Active Rule-Set
- ▷ Case Study
- Summary



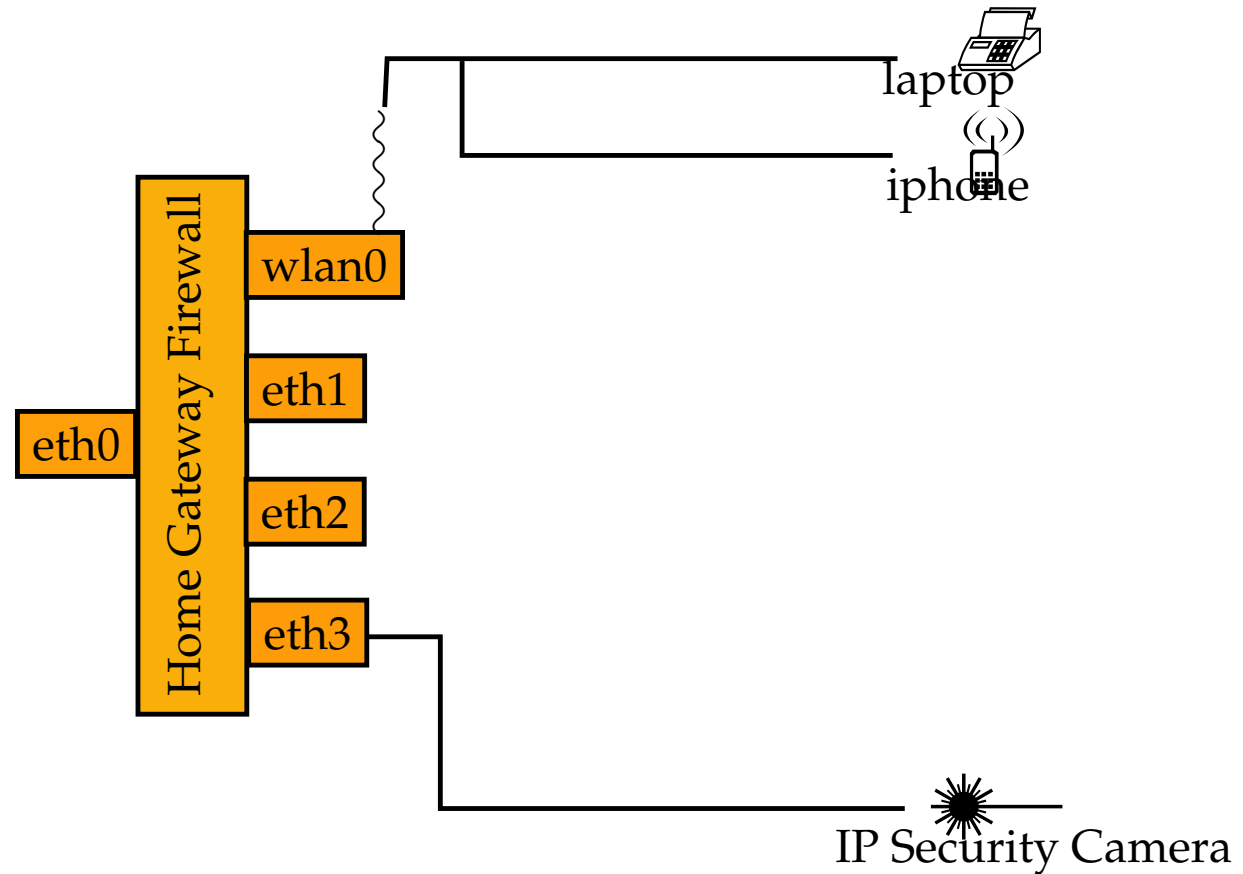
# Example NSP-8: Provide LAN users typical Internet access

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

```
# Permit LAN users access to HTTP(S),SSH,SMTP,IMAP.
$Ipt -A FORWARD -i wlan0 -p tcp -s $LAN --dport 80
-m state --state NEW, ESTABLISHED -j ACCEPT
$Ipt -A FORWARD -i wlan0 -p tcp -s $LAN --dport 443
-m state --state NEW, ESTABLISHED -j ACCEPT
$Ipt -A FORWARD -i wlan0 -p tcp -s $LAN --dport 22
-m state --state NEW, ESTABLISHED -j ACCEPT
$Ipt -A FORWARD -i wlan0 -p tcp -s $LAN --dport 25
-m state --state NEW, ESTABLISHED -j ACCEPT
$Ipt -A FORWARD -i wlan0 -p tcp -s $LAN --dport 587
-m state --state NEW, ESTABLISHED -j ACCEPT
$Ipt -A FORWARD -i wlan0 -p tcp -s $LAN --dport 993
-m state --state NEW, ESTABLISHED -j ACCEPT
```

# Example NSP-9: Provide access to Security Camera

- Background
- Rule Components
- Active Rule-Set
- ▷ Case Study
- Summary



# Example NSP-9: Provide access to Security Camera

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

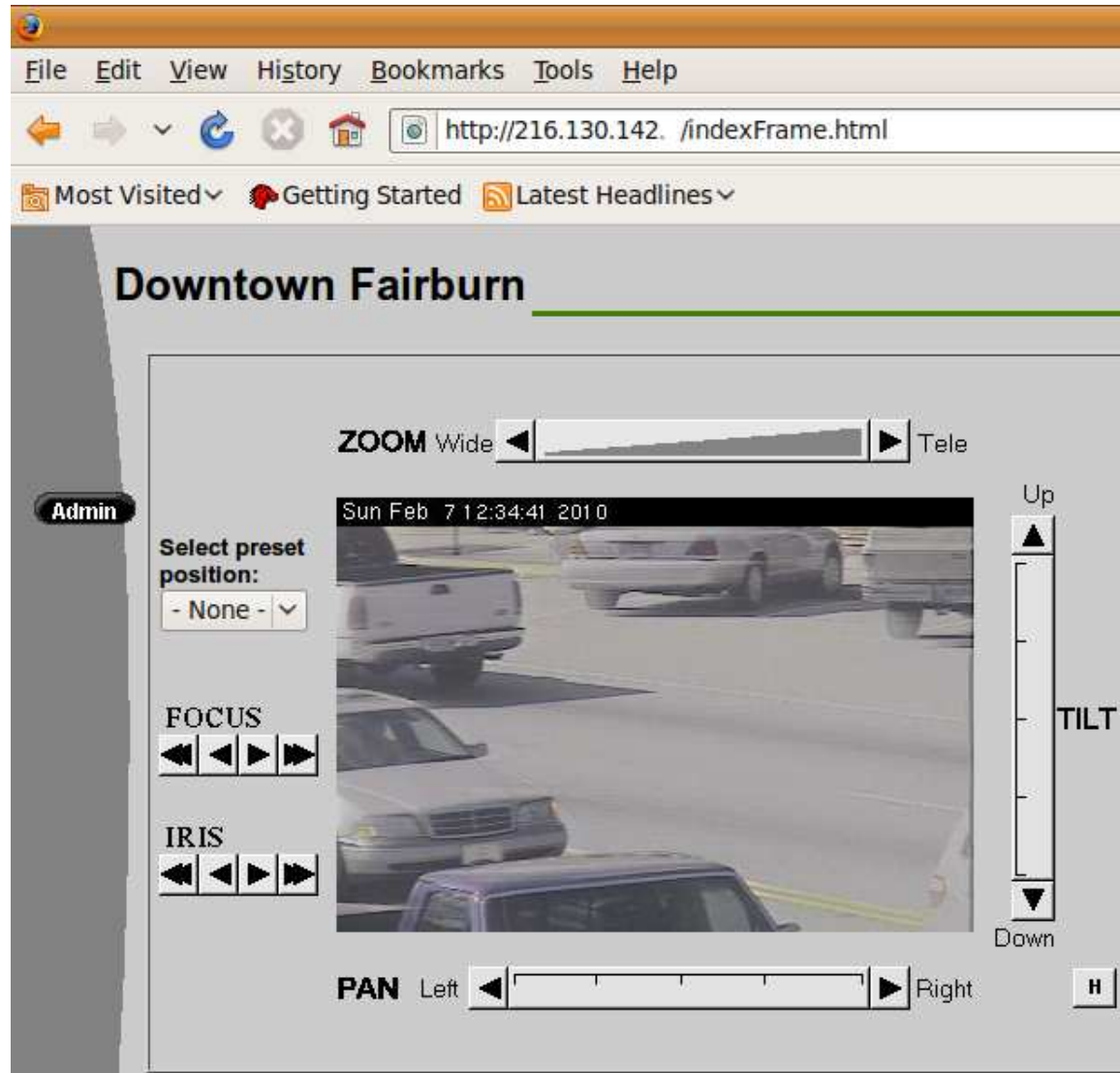
Why would one want to control access to security camera?

- Most IP-based camera's have little or no security.
- Prevent unauthorised users gaining knowledge of what you do and do not have in your backyard!
- Google URL spidering and indexing.
- Examples:
  - (1) "Live view - / - AXIS" ,
  - (2) indexFrame.html axis,
  - (3) /home/homeJ.html

IMPORTANT: I nor UCC advocates searching for and/or exploiting IP-based camera's. These examples are for educational purposes only.

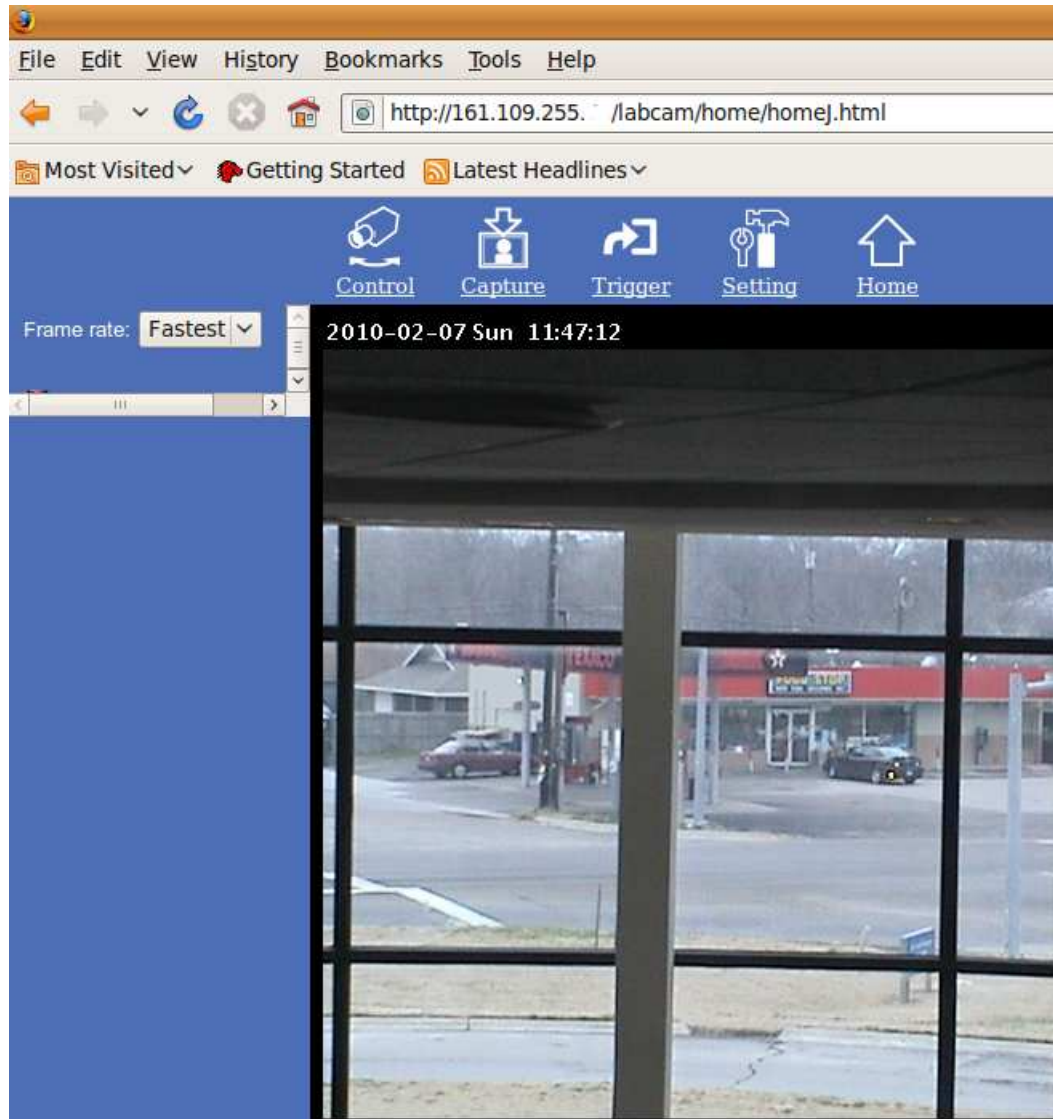
# Example NSP-9: Provide access to Security Camera

- Background
- Rule Components
- Active Rule-Set
- ▶ Case Study
- Summary



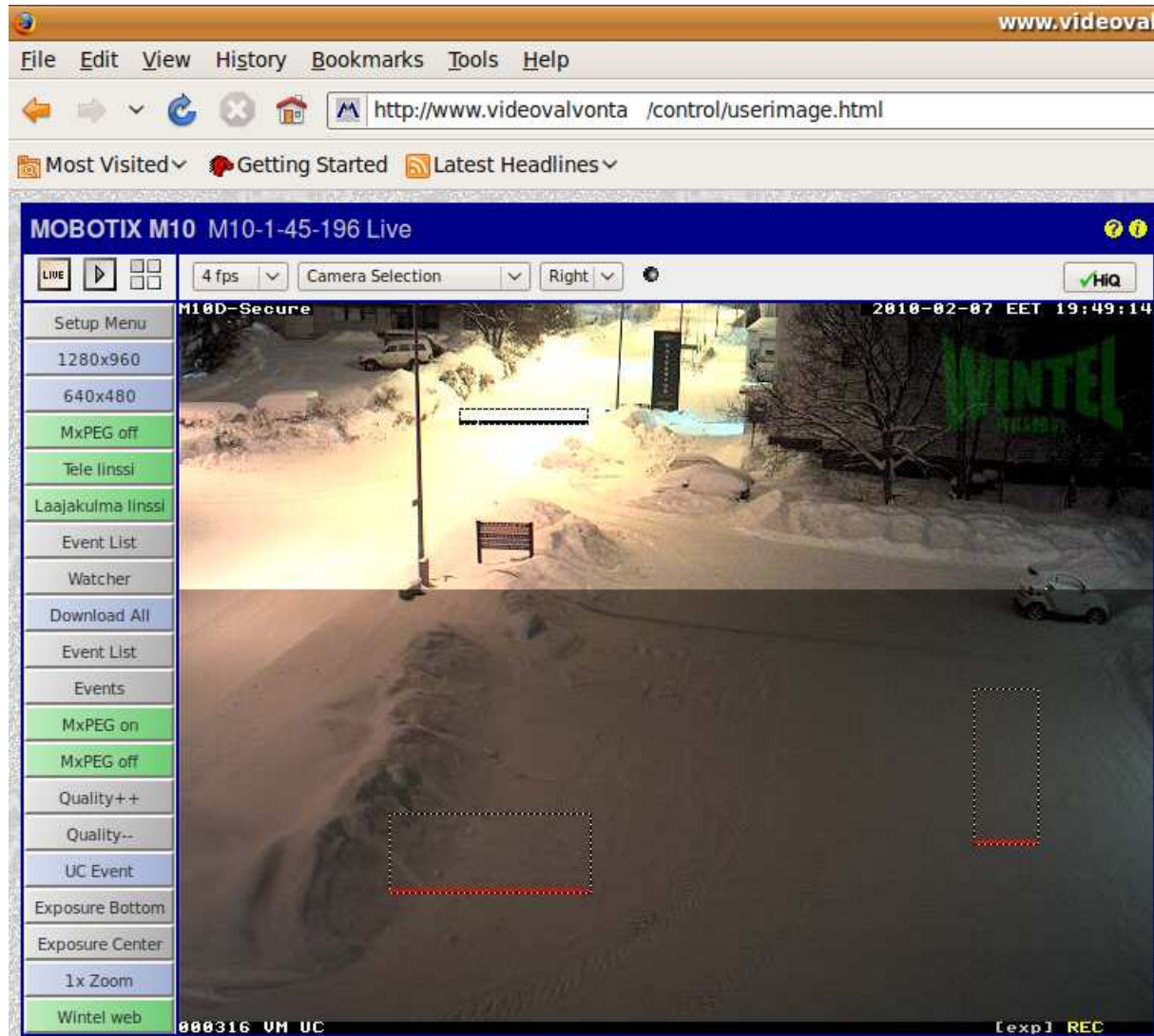
# Example NSP-9: Provide access to Security Camera

Background  
Rule Components  
Active Rule-Set  
▶ Case Study  
Summary



# Example NSP-9: Provide access to Security Camera

Background  
Rule Components  
Active Rule-Set  
▶ Case Study  
Summary



# Example NSP-9: Provide access to Security Camera

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

Typically IP-based security camera's provide access to its video stream via an internal Web server.

```
# Permit LAN users access to internal security camera.  
# $ipt -A FORWARD -i wlan0 -p tcp -s $LAN --dport 80  
# -m state --state NEW, ESTABLISHED -j ACCEPT  
$ipt -A FORWARD -i eth3 -p tcp -s $Cam --sport 80 -d $LAN  
-m state --state ESTABLISHED -j ACCEPT
```

# Example NSP-9: Provide access to Security Camera

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

```
# Permit LAN users access to internal security camera.  
# $ipt -A FORWARD -i wlan0 -p tcp -s $LAN --dport 80  
# -m state --state NEW, ESTABLISHED -j ACCEPT  
$ipt -A FORWARD -i eth3 -p tcp -s $Cam --sport 80 -d $LAN  
-m state --state ESTABLISHED -j ACCEPT
```

- Note, the first rule above is commented out (#) as this permission was generically provided (no destination IP explicitly provided) previously in provisioning for NSP-8.

# Another Example of NSP-3: Prevent DoS Attacks on Camera

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

Provide external access to the Camera (perhaps to be viewed remotely while on holiday) should tunneled over ssh.  
Connection attempts should be limited to prevent possible denial of service.

```
# Permit LAN users external access to internal security camera over SSH.
$Ipt -A FORWARD -i eth0 -p tcp -d $WAN --dport 22
-m state --state NEW, ESTABLISHED -m recent --set -j ACCEPT
$Ipt -A FORWARD -i eth0 -p tcp -d $WAN --dport 22
-m state --state NEW -m recent --update --seconds 600
--hitcount 11 -j DROP
```

# Another Example of NSP-3: Prevent DoS Attacks on Camera

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

```
# Permit LAN users external access to internal security camera over SSH.  
$ ipt -A FORWARD -i eth0 -p tcp -d $WAN --dport 22  
-m state --state NEW, ESTABLISHED -m recent --set -j ACCEPT  
$ ipt -A FORWARD -i eth0 -p tcp -d $WAN --dport 22  
-m state --state NEW -m recent --update --seconds 60  
--hitcount 11 -j DROP
```

- The firewall's SSH daemon can act as a secure tunnel.
- It is assumed that the correct *Port-Forwarding* via source NAT (SNAT) and destination NAT (DNAT) has been implemented.

# Another Example of NSP-3: Prevent DoS Attacks on Camera

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

```
# Permit LAN users external access to internal security camera over SSH.  
$iptables -A FORWARD -i eth0 -p tcp -d $WAN --dport 22  
-m state --state NEW, ESTABLISHED -m recent --set -j ACCEPT  
$iptables -A FORWARD -i eth0 -p tcp -d $WAN --dport 22  
-m state --state NEW -m recent --update --seconds 300  
--hitcount 1 -j DROP
```

- Recent module.
- Limit incoming connection to ssh server (port 22) (which is a front for the IP camera (port 80) to no more than 1 connection in a 5 minute interval.

# Example NSP-5,NSP-7,NSP-8: Permit Return Traffic

Background  
Rule Components  
Active Rule-Set  
▷ Case Study  
Summary

Previous rules permitted TCP/UDP connections outbound.  
Keep state so that corresponding return connections are allowed back in.

```
# Permit traffic into the Intranet if it has already been approved.  
$ipt -A FORWARD -i eth0 -m state --state ESTABLISHED -j ACCEPT
```

# Summary

Background  
Rule Components  
Active Rule-Set  
Case Study  
▷ Summary

This lecture/tutorial provided an overview of iptables regarding the firewall (only) aspects.