The Global Leader in Resilient **PNT**

# SecureSync® App Note

## Advanced Security, IP Tab


## 1. Introduction

The SecureSync is the only Time and Synchronization Device approved by the Defense Information Systems Agency (DISA) for cybersecurity and interoperability for use in the US Department of Defense networks. Along with the industry's leading security settings the SecureSync also has the additional security capabilities for advanced security settings within the Linux based platform. IP Tables allows the user to add additional layers of security via packet filtering and Hardware resource protection. IP Tables is included in a basic form and functionality starting at firmware version 5.4.1. An enhanced version of IP tables which includes Kernel updates and not only the ability to set Pre and Post routing but advanced routing rules with Mangled Tables.

## 2. Configuration

Must have Firmware version 5.7.1 for advanced IP Tables, 5.4.1 for basic IP Tables, installed on the SecureSync. Confirm by going to the TOOLS/SYSTEM menu of the SecureSync and select Upgrade/Backup. IP Tables comes enabled by default. Access the SecureSync serial [Management] port in order to make modifications to the Iptables.

To check what version of IP Tables is on the SecureSync go to the /home/Spectracom directory type Ip tables -V for the latest version. IP tables -LS will give the end user a list of the iptables chain rules both default and end user created.

## 3. Advanced Settings and Test]

Using a Serial interface port connection, CLI into the SecureSync.
Create/Add/Modify the Iptables chain rules.

**EXAMPLE 1 Input/Output chain rules**

Iptables -A INPUT -s 192.168.5.90 -j DROP
Iptables -A INPUT -s 192.168.5.85 -j DROP
*anywhere Chain Rule example from Wikipedia*

```
Chain INPUT (policy DROP 0 packets, 0 bytes)
pkts bytes target      prot opt   in     out         source      destination
    0      0 ACCEPT    all—any    any    anywhere    anywhere    state ESTABLISHED
    0      0 ACCEPT    tcp—any    any    anywhere    anywhere    tcp dpt:smtp
    0      0 LOG       all—any    any    anywhere    anywhere    LOG level warning
    0      0 DROP      all—any    any    anywhere    anywhere
```

Iptables -A OUTPUT -s 192.168.5.90 -j DROP
Iptables -A OUTPUT -s 192.1685.85 -j DROP
*Output Chain Rule example from Wikipedia.*

```
Chain OUTPUT (policy ACCEPT)
pkts bytes target      prot opt   in     out        source       destination
    0      0 DROP      tcp—any    any    server     anywhere     tcp dpt:smtp
```

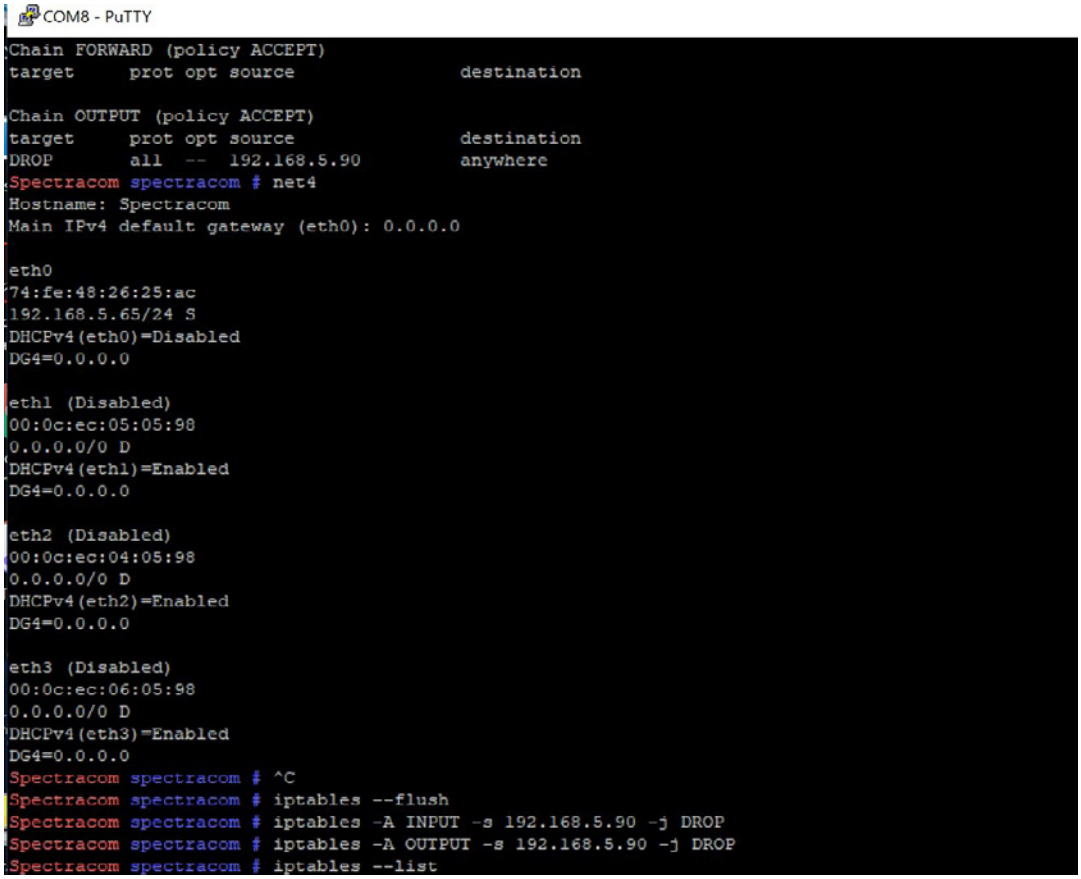


*Figure 1 INPUT/OUTPUT Chain Rule*

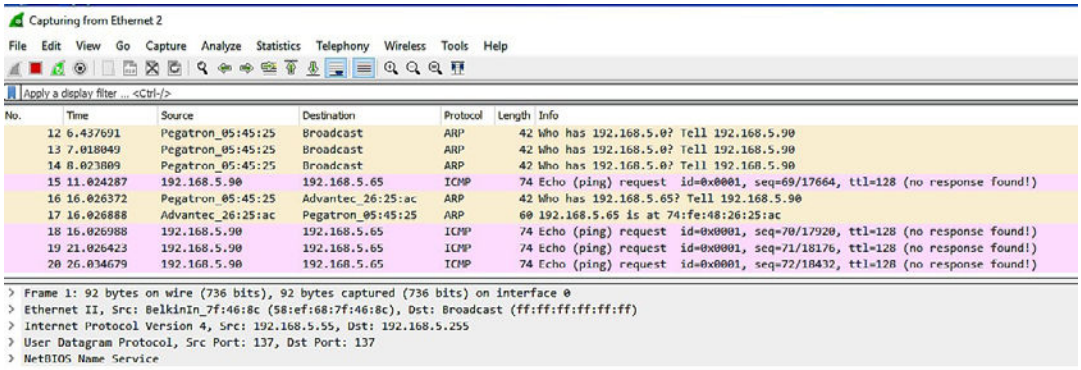


*Figure 2 Wireshark confirmation*

```
Try `iptables -h' or 'iptables --help' for more information.
Spectracom spectracom # iptables -A INPUT -s 192.168.5.90 -j DROP
Spectracom spectracom # iptables -A OUTPUT -s 192.168.5.90 -j DROP
Spectracom spectracom # iptables -A OUTPUT -s 192.168.5.55 -J DROP
iptables v1.4.21: unknown option "-J"
Try `iptables -h' or 'iptables --help' for more information.
Spectracom spectracom # iptables -A INPUT -s 192.168.5.55 -j DROP
Spectracom spectracom # iptables -A OUTPUT -s 192.168.5.55 -j DROP
Spectracom spectracom # iptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  192.168.5.90         anywhere
DROP       all  --  192.168.5.55         anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  192.168.5.90         anywhere
DROP       all  --  192.168.5.55         anywhere
Spectracom spectracom #
```

*Figure 3 Iptables List*

**Example 2 Port blocking Chain Rule**

Iptables -A INPUT -p tcp –destination-port 80 -j DROP

```
Spectracom spectracom # iptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
Spectracom spectracom # iptables -A INPUT -p tcp --destination-port 80 -j DROP
Spectracom spectracom #
```

*Figure 4 iptables Port Blocking Chain Rule*

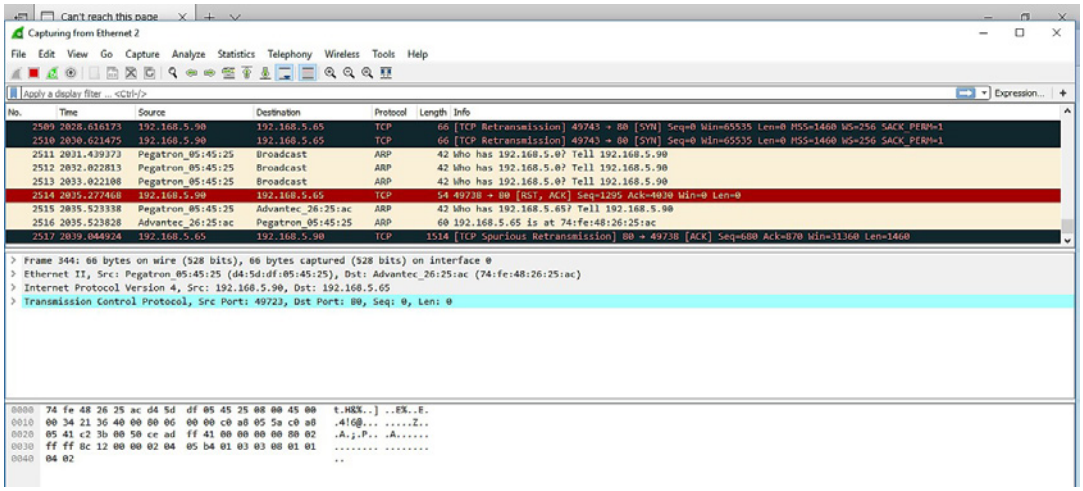


*Figure 5 Wireshark confirmation of TCP Port Blocking*

**EXAMPLE 3 Iptables Firewall Rules**

To trace the packets through the firewall, run the following to zero out the packet counters:

```
# iptables -Z
# iptables -nvL
```

Displays the trusted management rules in a more succinct form

```
# disp_mgmt_rules
```

Disable firewall

```
iptables -F
```

NAT table

```
# cat /proc/net/ip_conntrack
# iptables -nvL -t nat
```

**EXAMPLE 4 Iptables DDos Protection Rules**

The goal when using iptables should not be to detect and reject large DOS attacks. However, the goal should be to limit the amount TCP or UDP attacks using a combination of methods, to include iptables. With the use of iptables and the end-user can configure packet filtering at the Kernel level using the CLI.

Which table to use? Filter, Nat, Mangle, and Raw. The best answer is a combination of both dependent upon where your iptables are going to reside and work. More importantly is the performance of the hardware iptables reside on. It must be able to handle DOS attempts at a very high packet-rate.

Iptables implementation for TCP-based DOS attacks.

Mangle table will allow the end user take advantage of filtering unwanted packets in the Prerouting Chain.

```
# Iptables -t mangle -A PREROUTING -m conntrack –ctstate INVALID -j
```

This example will drop all ICMP packets that are used to ping to see if a host is kept alive.

```
# Iptable -t mangle -A PREROUTING -p icmp -j DROP
```

## 4. Help and additional references

**Commands:**

**By Typing iptables -h for additional CLI HELP**

```
Usage: iptables -[ACD] chain rule-specification [options]
 iptables -D chain rulenum [options]
 iptables -[LS] [chain [rulenum]] [options]
 iptables -[FZ] [chain] [options]
 iptables -[NX] chain
 iptables -E old-chain-name new-chain-name
 iptables -P chain target [options]
 iptables -h (print this help information
```

**Either long or short options are allowed.**

```
 --append -A chain Append to chain
 --check -C chain Check for the existence of a rule
 --delete -D chain Delete matching rule from chain
 --delete -D chain rulenum Delete rule rulenum (1 = first) from chain
 --insert -I chain [rulenum] Insert in chain as rulenum (default 1=first)
 --replace -R chain rulenum Replace rule rulenum (1 = first) in chain
 --list -L [chain [rulenum]] List the rules in a chain or all chains
 --list-rules -S [chain [rulenum]] Print the rules in a chain or all chains
 --flush -F [chain] Delete all rules in  chain or all chains
 --zero -Z [chain [rulenum]] Zero counters in chain or all chains
 --new -N chain Create a new user-defined chain
 --delete-chain -X [chain] Delete a user-defined chain
 --policy -P chain target Change policy on chain to target
 --rename-chain -E old-chain new-chain Change chain name, (moving any references)
```

**Options:**

```
    --ipv4 -4 Nothing (line is ignored by iptables-restore)
    --ipv6 -6 Error (line is ignored by ip6tables-restore)
[!] --protocol -p proto protocol: by number or name, eg. `tcp'
[!] --source -s address[/mask][...] Source specification
[!] --destination -d address[/mask][...] Destination specification
[!] --in-interface -i input name[+] Network interface name ([+] for wildcard)
    --jump -j target target for rule (may load target extension)
```

```
    --goto -g chain jump to chain with no return
    --match -m match Extended match (may load extension)
    --numeric -n numeric output of addresses and ports
[!] --out-interface -o output name[+] Network interface name ([+] for wildcard)
    --table -t table Table to manipulate (default: `filter')
    --verbose -v verbose mode
    --wait -w wait for the xtables lock
    --line-numbers print line numbers when listing
    --exact -x expand numbers (display exact values)
[!] --fragment -f match second or further fragments only
    --modprobe=<command> try to insert modules using this command
    --set-counters PKTS BYTES set the counter during insert/append
[!] --version -V print package version.
Usage: iptables -[ACD] chain rule-specification [options]
       iptables -D chain rulenum [options]
       iptables -[LS] [chain [rulenum]][options]
       iptables -[FZ] [chain][options]
       iptables -[NX] chain
       iptables -E old-chain-name new-chain-name
       iptables -P chain target [options]
       iptables -h (print this help information)
Commands: Either long or short options are allowed.
    --append -A chain Append to chain
    --check -C chain Check for the existence of a rule
    --delete -D chain Delete matching rule from chain
    --delete -D chain rulenum Delete rule rulenum (1 = first) from chain
    --insert -I chain [rulenum] Insert in chain as rulenum (default 1=first)
    --replace -R chain rulenum Replace rule rulenum (1 = first) in chain
    --list -L [chain [rulenum]] Lists the rules in a chain or all chains
    --list-rules -S [chain [rulenum]] Prints the rules in a chain or all chains
    --flush -F [chain] Delete all rules in chain or all chains
    --zero -Z [chain [rulenum]] Zero counters in chain or all chains
    --new -N chain Create a new user-defined chain
    --delete-chain -X [chain] Delete a user-defined chain
    --policy -P chain target Change policy on chain to target
    --rename-chain -E old-chain new-chain Change chain name, (moving any references)
    --ipv4 -4 Nothing (line is ignored by ip6tables-restore)
    --ipv6 -6 Error (line is ignored by iptables-restore)
[!] --protocol -p proto protocol: by number or name, eg. `tcp'
[!] --source -s address[/mask][...] Source specification
[!] --destination -d address[/mask][...] Destination specification
[!] --in-interface -i input name[+] Network interface name ([+] for wildcard)
    --jump -j target Target for rule (may load target extension)
    --goto -g chain Jump to chain with no return
    --match -m match Extended match(may load extension)
    --numeric -n numeric output of addresses and ports
[!] --out-interface -o output name[+] Network interface name ([+] for wildcard)
    --table -t table table to manipulate (default: `filter')
    --verbose -v verbose mode
    --wait -w wait for the xtables lock
    --line-numbers  print line numbers when listing
    --exact -x expand numbers (display exact values)
[!] --fragment -f match second or further fragments only
    --modprobe=<command> try to insert modules using this command
    --set-counters PKTS BYTES set the counter during insert/append
[!] --version -V print package version.
```

**References**

https://www.orolia.com/products-services/precision-timing
https://www.orolia.com/knowledge-center/resilient-pnt-cybersecurity
https://en.wikipedia.org/wiki/Ipfirewall
https://netfilter.org/