mmu

VPN: Virtual Private Network

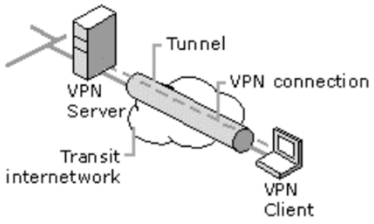


Figure 1Virtual Private Network Connection

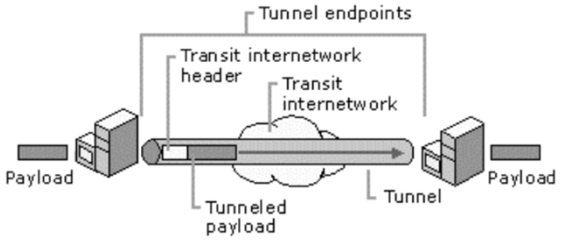


Figure 5 Tunneling Technique

IPSec is designed to provide interoperable, high quality, cryptographically based security for Internet protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6). The set of security services offered includes access control, connectionless integrity, data origin authentication, protection against replays (a form of partial sequence integrity), confidentiality (encryption), and limited traffic flow confidentiality. These services are provided at the IP layer, offering protection for IP and/ or upper layer protocols[1][5][7].

https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.503.7298&rep=rep1&type=pdf

Dr. Gaurav Varshney, IIT Jammu

IPsec and VPNs: Network Layer Security

The security services are provided through the use of two security protocols-the Authentication Header (AH) and the Encapsulating Security Payload (ESP)- and through the use of cryptographic key management procedures and protocols, including the Internet Security Association and Key Management Protocol (ISAKMP) and the Internet Key Exchange protocol (IKE).

When a source IPsec entity (typically a host or a router) sends secure datagrams to a destination entity (also a host or a router), it does so with either the AH protocol or the ESP protocol. The AH protocol provides source authentication and data integrity but *does not* provide confidentiality.

The ESP protocol provides source authentication, data integrity, *and* confidentiality. Because confidentiality is often critical for VPNs and other IPsec applications, the ESP protocol is much more widely used than the AH protocol.

Transport Mode provides a secure connection between two endpoints as it encapsulates IP's payload, while Tunnel Mode encapsulates the *entire* IP packet to provide a virtual "secure hop" between two gateways.

Dr. Gaurav Varshney, IIT Jammu

IPsec and VPNs: Network Layer Security

The concept of a security association (SA) is fundamental to the IP security architecture. An SA defines the kinds of security measures that should be applied to the packets based on who is sending the packets, where they are going, and what type of payload they are carrying.

SAs can be negotiated dynamically between two communicating peers when they wish to use one or more of IPsec's security services, based on the security policies given by the security administrator.

A SA is uniquely identified by three parameters: a destination IP address, a security protocol identifier, and a Security Parameter Index (SPI). The destination IP address is the IP address of the destination endpoint for the SA. The SPI is a 32-bit number usually chosen by the destination endpoint of the SA, and it has local significance only within that destination endpoint. The security protocol identifier is the protocol number for either AH (51) or ESP (50).

Note that the source IP address is not used to define a SA. This is because a SA is a security services agreement between two hosts or gateways for data sent in one direction. As a result, if two peers need to exchange information in both directions using IPSec, two SAs are required, one for each direction.

IPsec and VPNs: Network Layer Security

Dr. Gaur ammu

SAs operate in two modes: transport mode and tunnel mode. Transport mode is designed primarily to protect the higher-layer protocols (e.g., TCP and UDP). In tunnel mode, an IP packet becomes the payload for another IP packet. This allows the inner IP packet, including its IP header, to be subjected to encryption or other security measures, whereas the outside IP packet serves to steer the data through the network. Hosts can provide both transport mode and tunnel modes, whereas security gateways can provide only tunnel mode (unless the gateway is acting as a host, in which case it can provide either mode).

A SA can be viewed as a unidirectional channel, offering either AH or ESP security. Note that each SA can only offer one of these security services. If both AH and ESP protection is applied to a data stream, then two SAs must be established and maintained. Similarly, to secure bi-directional communications between two hosts or security gateways, two SAs (one in each direction) are required. The term SA bundle is applied to a sequence of SAs through which traffic must be processed to satisfy a specific security policy.

Two databases are associated within an IPSec node: the Security Policy Database (SPD) and the Security Association Database (SAD). A policy administrator composes a set of security policies to meet the security needs of all types of IP traffic both into and out of this node. These policies are kept in the SPDs to be used in directing the processing of IP packets and the construction of SAs as needed. All SAs are registered in the SAD, along with the SAs' parameters.

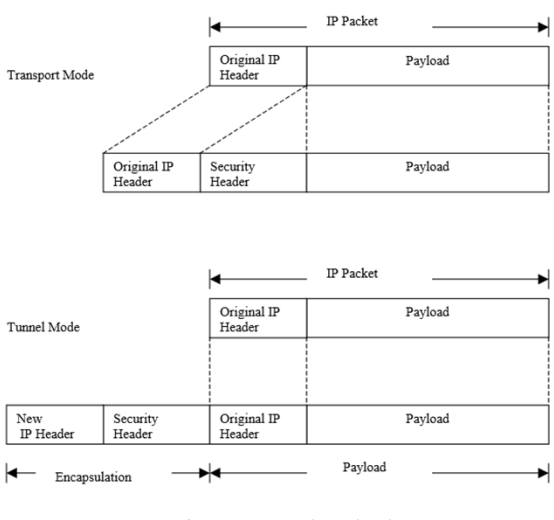
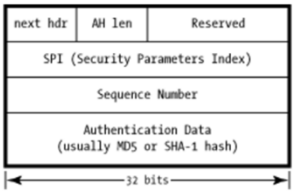


Figure 6 Transport and Tunnel Modes

IPsec and VPNs: Network Layer Security

Dr. Gaurav Varshney, IIT Jammu

IPSec AH Header



Next Header The Next Header field is an 8-bit field that identifies the type of protocol header immediately after the AH.

Payload length This 8-bit field specifies the length of AH in 32-bit words, minus 2.

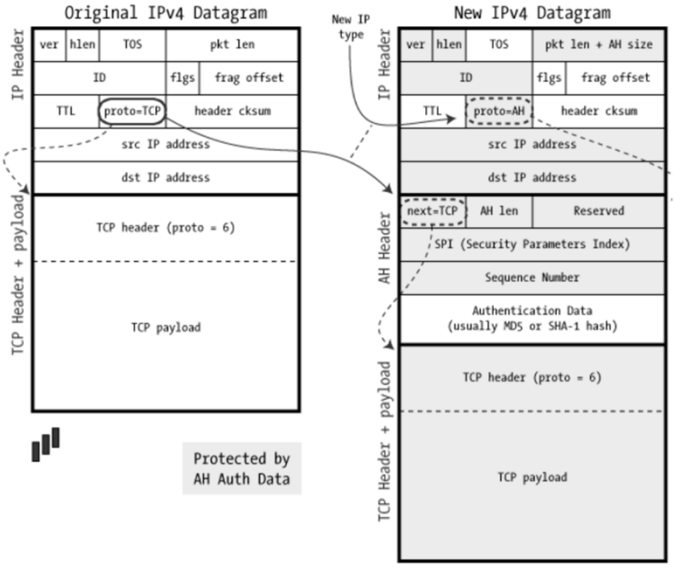
Reserved This 16-bit field is reserved for future use, and is set to 0.

Security Parameters Index This 32-bit field together with the IP destination address, identifies the Security association. The SPI is an arbitrary value chosen upon establishing the SA. The value 0 may be used to indicate that a SA has not yet been established, while the values 1-255 are reserved for future use.

Sequence Number This 32-bit field indicates the packet's position in sequence. The sequence number is initialized to 0 upon establishing a SA, and is incremented by one with each sent packet. If replay protection is enabled, this value must never be allowed to cycle. When all sequence number values have been used, a new SA and thus a new key must be established.

Authentication Data This field contains the Integrity Check Value (ICV) generated by using an authentication algorithm and a cryptographic key specified in the corresponding SA. The length of this field is not fixed, and depends on the algorithm specification. The sender computes the data prior to sending the IP packet, and the receiver verifies it upon reception.

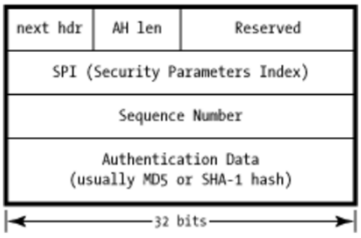
IPSec in AH Transport Mode



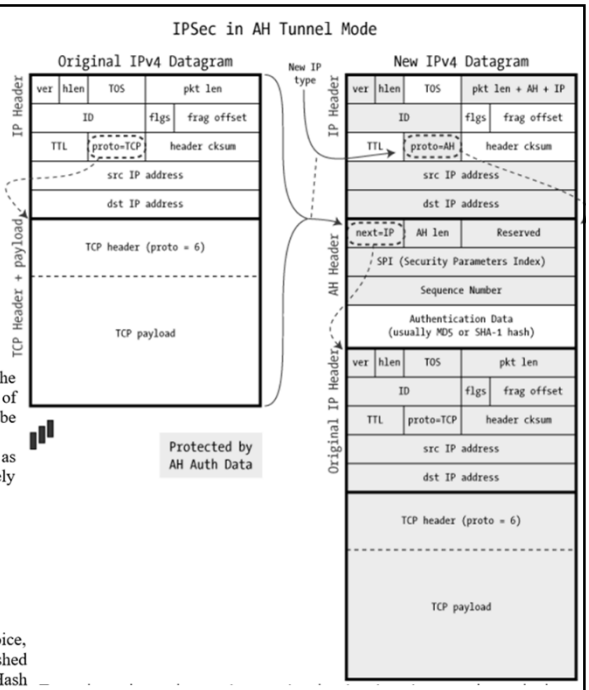
IPsec

Dr. Gaurav Varshney, IIT Jammu

IPSec AH Header



IPSec in AH Tunnel Mode



The original IP header is completely unaltered and contains the ultimate destination as well as the original source IP address. The new IP header contains the source and destination IP addresses of the IPSec devices between which the new packet will travel. Consequently, tunnel mode can be used whether the endpoints of the SA are hosts or security gateways.
If the SA is between hosts, the new source and destination IP addresses are usually the same as the original. The typical reason for using AH in tunnel mode between hosts is to completely authenticate the original packet.

When the next-header value is *IP,* it means that this packet encapsulates an entire IP datagram (including the independent source and destination IP addresses that allow separate routing after de-encapsulation). This is Tunnel mode.

AH is algorithm independent, which means that AH will operate with the algorithm of choice, depending on the level of security required. Currently the algorithm options are HMAC (hashed message authentication code), MD5 (message digest 5) or HMAC SHA1 (Secure Hash Algorithm).

IPsec

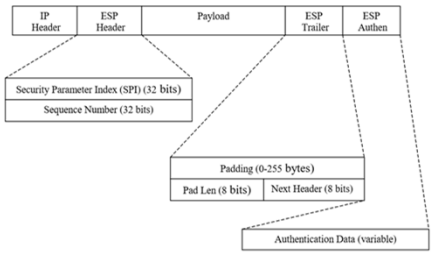
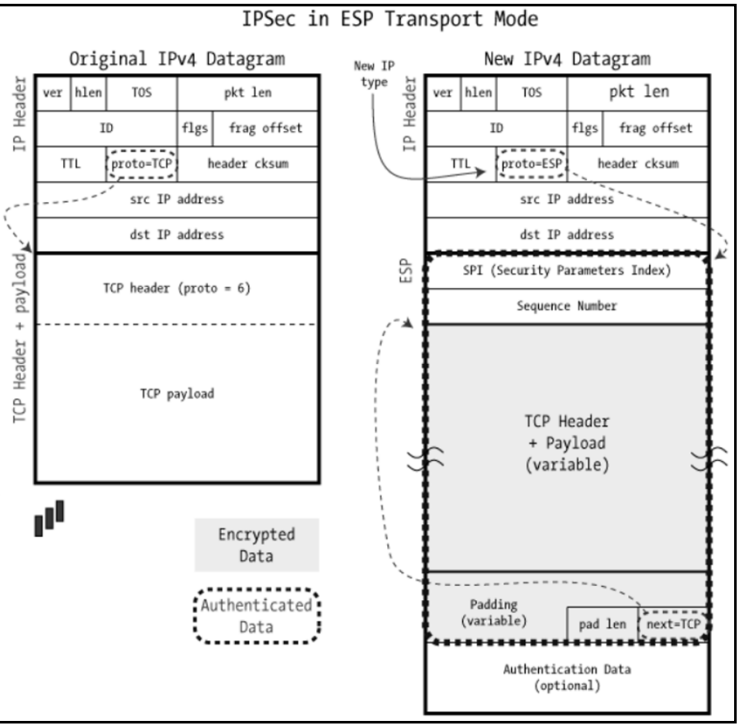


Figure 10 ESP header, trailer, and authentication segment formats

Padding The padding field contains 0-255 randomly generated types of data, and can be used for several purposes:

- Some encryption algorithms require the cleartext to be a multiple of some number of bytes, in which case the Padding field is used to fill out the cleartext (consisting of Payload Data Pad Length and Next Header Fields) to the size required by the algorithm.
- Padding may also be required, irrespective of encryption algorithm requirements, to ensure that the resulting encrypted payload terminates on a 4-byte/32-bit boundary as required by the ESP packet format.
- Padding may also be used to conceal the actual length of the payload, in support of (partial) traffic flow confidentiality.

Pad Length The Pad length field indicates the number of pad bytes immediately preceding it. The range of valid values 0-255, where a value of zero indicates that no Padding bytes are present. The Pad Length field is mandatory.



IPsec

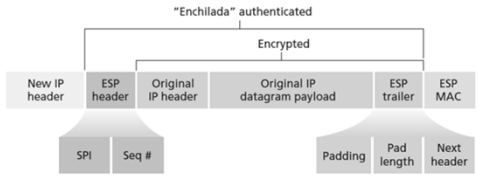


Figure 8.29 ♦ IPsec datagram format

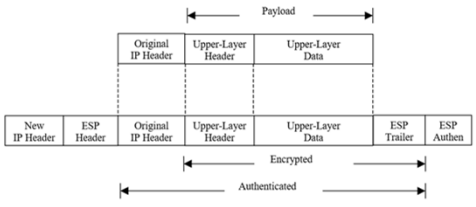
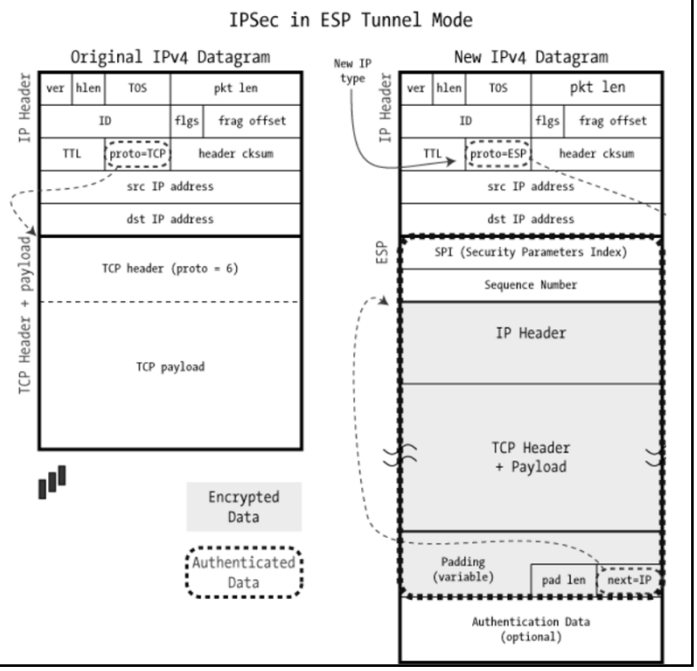


Figure 13 ESP Tunnel mode

ESP is also designed to be algorithm independent and the options are DES, 3DES, RC5, Blowfish, Idea, Cast, and others are being added.

	Data Origin Authentication	Data Integrity	Replay Protection	Data Confidentiality
AH	Yes	Yes	Yes	No
ESP	Yes	Yes	Yes	Yes

Table 1: Comparison of security services provided by AH and ESP Headers



Dr. Gaurav Varshney, IIT Jammu

IPTables and Netfilter Framework

The following hooks represent various well-defined points in the networking stack:

- `NF_IP_PRE_ROUTING`: This hook will be triggered by any incoming traffic very soon after entering the network stack. This hook is processed before any routing decisions have been made regarding where to send the packet.
- `NF_IP_LOCAL_IN`: This hook is triggered after an incoming packet has been routed if the packet is destined for the local system.
- `NF_IP_FORWARD`: This hook is triggered after an incoming packet has been routed if the packet is to be forwarded to another host.
- `NF_IP_LOCAL_OUT`: This hook is triggered by any locally created outbound traffic as soon it hits the network stack.
- `NF_IP_POST_ROUTING`: This hook is triggered by any outgoing or forwarded traffic after routing has taken place and just before being put out on the wire.

Kernel modules that wish to register at these hooks must provide a priority number to help determine the order in which they will be called when the hook is triggered.

Each module will be called in turn and will return a decision to the netfilter framework after processing that indicates what should be done with the packet.

Dr. Gaurav Varshney, IIT Jammu

IPTables and Netfilter Framework

IPTables Tables and Chains

The `iptables` firewall uses tables to organize its rules. These tables classify rules according to the type of decisions they are used to make. For instance, if a rule deals with network address translation, it will be put into the `nat` table. If the rule is used to decide whether to allow the packet to continue to its destination, it would probably be added to the `filter` table.

Within each `iptables` table, rules are further organized within separate "chains". While tables are defined by the general aim of the rules they hold, the built-in chains represent the `netfilter` hooks which trigger them. Chains basically determine *when* rules will be evaluated.

As you can see, the names of the built-in chains mirror the names of the `netfilter` hooks they are associated with:

- `PREROUTING`: Triggered by the `NF_IP_PRE_ROUTING` hook.
- `INPUT`: Triggered by the `NF_IP_LOCAL_IN` hook.
- `FORWARD`: Triggered by the `NF_IP_FORWARD` hook.
- `OUTPUT`: Triggered by the `NF_IP_LOCAL_OUT` hook.
- `POSTROUTING`: Triggered by the `NF_IP_POST_ROUTING` hook.

IPTables and Netfilter Framework

Dr. Gaurav Varshney, IIT Jammu

Tables↓/Chains→	PREROUTING	INPUT	FORWARD	OUTPUT	POSTROUTING
(routing decision)				✓	
raw	✓			✓	
(connection tracking enabled)	✓			✓	
mangle	✓	✓	✓	✓	✓
nat (DNAT)	✓			✓	
(routing decision)	✓			✓	
filter		✓	✓	✓	
security		✓	✓	✓	
nat (SNAT)		✓			✓

Chain Traversal Order

Assuming that the server knows how to route a packet and that the firewall rules permit its transmission, the following flows represent the paths that will be traversed in different situations:

- **Incoming packets destined for the local system**: `PREROUTING` → `INPUT`
- **Incoming packets destined to another host**: `PREROUTING` → `FORWARD` → `POSTROUTING`
- **Locally generated packets**: `OUTPUT` → `POSTROUTING`

IPTables and Netfilter Framework

Dr. Gaurav Varshney, IIT Jammu

Available States

Connections tracked by the connection tracking system will be in one of the following states:

- `NEW`: When a packet arrives that is not associated with an existing connection, but is not invalid as a first packet, a new connection will be added to the system with this label. This happens for both connection-aware protocols like TCP and for connectionless protocols like UDP.
- `ESTABLISHED`: A connection is changed from `NEW` to `ESTABLISHED` when it receives a valid response in the opposite direction. For TCP connections, this means a `SYN/ACK` and for UDP and ICMP traffic, this means a response where source and destination of the original packet are switched.
- `RELATED`: Packets that are not part of an existing connection, but are associated with a connection already in the system are labeled `RELATED`. This could mean a helper connection, as is the case with FTP data transmission connections, or it could be ICMP responses to connection attempts by other protocols.
- `INVALID`: Packets can be marked `INVALID` if they are not associated with an existing connection and aren't appropriate for opening a new connection, if they cannot be identified, or if they aren't routable among other reasons.

iptables firewall

- Linux inbuilt firewall provides an access to put network firewalling rules directly to a module in the kernel.
- Commands [default table is filter]

  sudo <iptables> -A <category> -p <protocol> -j <action>

  Iptables organizes firewall rules into different categories referred to as “chains”

  The basic three which are maintained by the Kernel and used are

  1. INPUT: Incoming or Ingress network traffic
  2. OUTPUT: Outgoing or egress network traffic
  3. FORWARD: Traffic forwarded from one interface to another [if a router]

iptables Firewall

- Iptables follows “match plus action” paradigm which means that it matches the packet and performs an actions
- Iptables accepts a number of actions also referred to as targets. One of these predefined actions are available
  - ACCEPT: allow the packet
  - REJECT: Deny the packet
  - DROP: Ignore or drop the packet
  - LOG: log information about the packet in the system log.

iptables firewall

Example rule to allow ICMP traffic

- sudo iptables –A INPUT –p icmp –j ACCEPT

To delete a rule we use –D

- sudo iptables –D INPUT –p icmp –j ACCEPT
- To delete all rules from a particular chain
- sudo iptables –F INPUT

Delete all rules from all chains

- sudo iptables –F

Picking up specific network ports

- sudo iptables –A INPUT –p tcp -–dport 22 –j ACCEPT

iptables firewall

Picking up specific IP addresses

- sudo iptables –A INPUT –p tcp –s 10.0.0.0/255.255.255.0 –dport 22 –j ACCEPT

- Handling egress traffic
  - sudo iptables –A OUTPUT –p tcp –d 10.0.0.0/255.255.255.0 –j ACCEPT
  - sudo iptables –A OUTPUT –p udp –d 10.0.0.0/255.255.255.0 –j ACCEPT
  - sudo iptables –A OUTPUT –p icmp –d 10.0.0.0/255.255.255.0 –j ACCEPT

Saving rules: sudo sh -c iptables-save > /etc/iptables.rules
sudo sh –c iptables-restore < /etc/iptables.rules

sudo gedit /etc/network/interfaces
auto eth0
iface eth0 inet dhcp
pre-up iptables-restore < /etc/iptables.rules
post-down iptables-restore < /etc/iptables.downrules

Dr. Gaurav Varshney, IIT Jammu

Iptables firewall

Logging traffic that doesn't match any rule in INPUT chain

- sudo iptables –A INPUT –j LOG –-log-prefix "iptables-rejected"

Deny all other ingress traffic that does not match a rule in the input chain

- sudo iptables –A INPUT –j REJECT –-reject-with icmp-host-prohibited
- sudo iptables –A INPUT –j REJECT –-reject-with tcp-reset
- Fore more reject-with options: https://www.linuxtopia.org/Linux_Firewall_iptables/x4550.html

Web servers can configure firewall to only listen to port 443

- sudo iptables –A INPUT–p tcp –dport 443 –j ACCEPT

DNS servers will listen on port 53 of both TCP and UDP

Jammu

Iptables Firewall

Additional match options are also available through modules loaded by the `iptables` command. To use a match option module, load the module by name using the `-m` option, such as `-m <module-name>` (replacing *<module-name>* with the name of the module).

A large number of modules are available by default. It is even possible to create modules that provide additional functionality.

The following is a partial list of the most commonly used modules:

- `limit` module — Places limits on how many packets are matched to a particular rule. This is especially beneficial when used in conjunction with the `LOG` target as it can prevent a flood of matching packets from filling up the system log with repetitive messages or using up system resources. Refer to Section 18.3.5 *Target Options* for more information about the `LOG` target.

  The `limit` module enables the following options:

  - `--limit` — Sets the number of matches for a particular range of time, specified with a number and time modifier arranged in a `<number>/<time>` format. For example, using `--limit 5/hour` only lets a rule match `5` times in a single hour.

    If a number and time modifier are not used, the default value of `3/hour` is assumed.

  - `--limit-burst` — Sets a limit on the number of packets able to match a rule at one time. This option should be used in conjunction with the `--limit` option, and it accepts a number to set the burst threshold.

    If no number is specified, only five packets are initially able to match the rule.

- `state` module — Enables state matching.

  The `state` module enables the following options:

  - `--state` — match a packet with the following connection states:
    - `ESTABLISHED` — The matching packet is associated with other packets in an established connection.
    - `INVALID` — The matching packet cannot be tied to a known connection.
    - `NEW` — The matching packet is either creating a new connection or is part of a two-way connection not previously seen.
    - `RELATED` — The matching packet is starting a new connection related in some way to an existing connection.

    These connection states can be used in combination with one another by separating them with commas, such as `-m state --state INVALID,NEW`.

- `mac` module — Enables hardware MAC address matching.

  The `mac` module enables the following option:

  - `--mac-source` — Matches a MAC address of the network interface card that sent the packet. To exclude a MAC address from a rule, place an exclamation point character (`!`) after the `--mac-source` match option.

https://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-en-4/s1-iptables-options.html

Dr. Gaurav Varshney, IIT Jammu

Iptables Firewall

- Egress rules
  - sudo iptables –A OUTPUT –i lo –j ACCEPT
  - sudo iptables –A OUTPUT –p icmp –j ACCEPT
  - sudo iptables –m state –-state ESTABLISHED,RELATED –j ACCEPT
  - sudo iptables –A OUTPUT –p tcp –d 10.0.0.0/255.255.255.0 –j ACCEPT
  - sudo iptables –A OUTPUT –p udp –d 10.0.0.0/255.255.255.0 –j ACCEPT
  - sudo iptables –A OUTPUT –j LOG –-log-prefix “iptables-reject”
  - sudo iptables –A OUTPUT –j REJECT –-reject-with icmp-host-prohibited

Dr. Gaurav Varshney, IIT Jammu

Iptables firewall

Syn-flood protection:

```
# iptables -A FORWARD -p tcp --syn -m limit --limit 1/s -j ACCEPT
```

Furtive port scanner:

```
# iptables -A FORWARD -p tcp --tcp-flags SYN,ACK,FIN,RST RST \
-m limit --limit 1/s -j ACCEPT
```

Ping of death:

```
# iptables -A FORWARD -p icmp --icmp-type echo-request -m limit \
--limit 1/s -j ACCEPT
```

- **Limit connections per source IP**
- -A INPUT -p tcp -m connlimit --connlimit-above 111 -j REJECT --reject-with tcp-reset
- **Limit RST packets**
- iptables -A INPUT -p tcp --tcp-flags RST RST -m limit --limit 2/s --limit-burst 2 -j ACCEPT
- **Limits new TCP connections that client can establish per second**
- iptables -A INPUT -p tcp -m conntrack --ctstate NEW -m limit --limit 60/s --limit-burst 20 -j ACCEPT
- **Allow established and related traffic**
- iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
- One can install arptables utility on linux to filter layer 2 packets
- FOR MORE OPTIONS LOOK AT https://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-en-4/s1-iptables-options.html