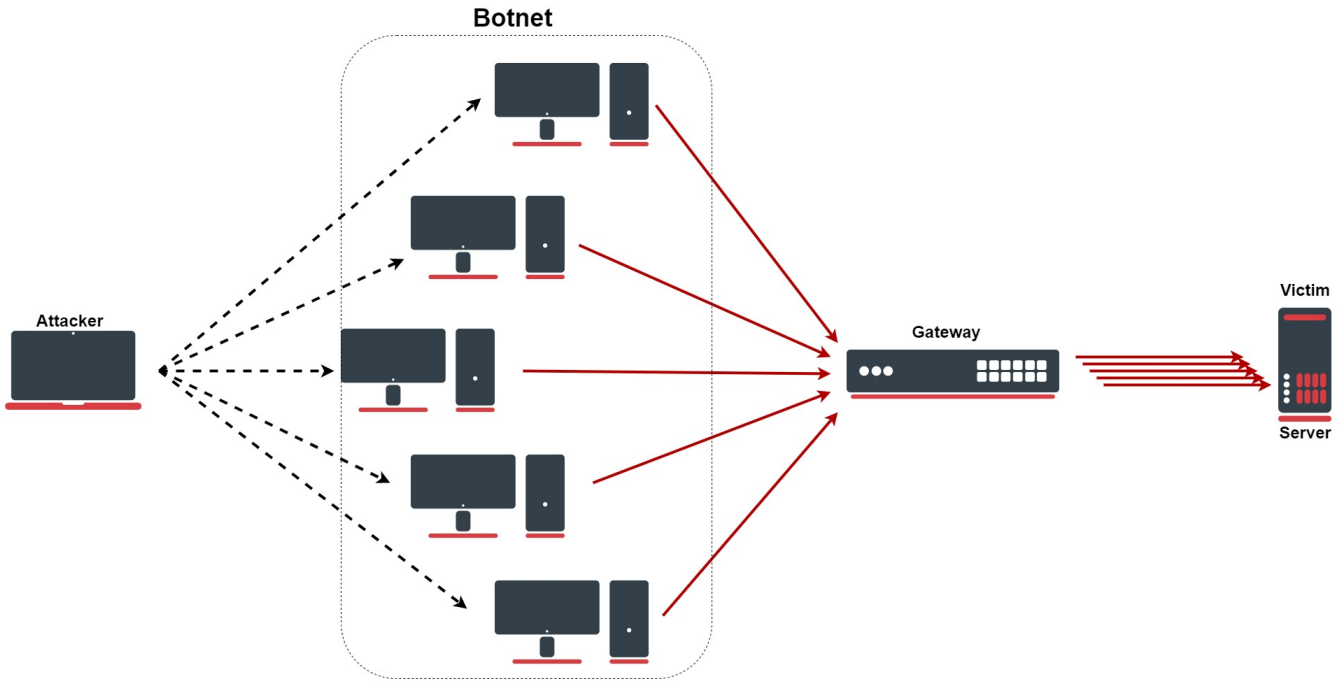
A denial-of-service (DoS) or distributed denial-of-service (DDoS) attack is a malicious attempt to disrupt normal traffic of a targeted server, service, or network by overwhelming the target or its surrounding infrastructure with a flood of Internet traffic. There are several types of DDoS attacks, for example, HTTP flood, SYN flood, DNS amplification, etc.



Configuration lines

> ⚠ These rules are only an improvement for firewall, do not forget to properly secure your device: Building Your First Firewall!

```
/ip firewall address-list
add list=ddos-attackers
add list=ddos-target
/ip firewall filter
add action=return chain=detect-ddos dst-limit=32,32,src-and-dst-addresses/10s
add action=add-dst-to-address-list address-list=ddos-target address-list-timeout=10m chain=detect-ddos
add action=add-src-to-address-list address-list=ddos-attackers address-list-timeout=10m chain=detect-ddos
/ip firewall raw
add action=drop chain=prerouting dst-address-list=ddos-target src-address-list=ddos-attackers
```

Configuration explained

First, we will send every new connection to the specific firewall chain where we will detect DDoS:

```
/ip/firewall/filter/add chain=forward connection-state=new action=jump jump-target=detect-ddos
```

In the newly created chain, we will add the following rule with the "dst-limit" parameter. This parameter is written in the following format: **dst-limit=count[/time],burst, mode[/expire]**. We will match 32 packets with 32 packet burst based on destination and source address flow, which renews every 10 seconds. The rule will work until a given rate is exceeded.

```
/ip/firewall/filter/add chain=detect-ddos dst-limit=32,32,src-and-dst-addresses/10s action=return
```

So far all the legitimate traffic should go through the "action=return", but in the case of DoS/DDoS "dst-limit" buffer will be fulfilled and a rule will not "catch" any new traffic. Here come the next rules, which will deal with the attack. Let`s start with creating a list for attackers and victims which we will drop:

```
ip/firewall/address-list/add list=ddos-attackers
ip/firewall/address-list/add list=ddos-targets
ip/firewall/raw/add chain=prerouting action=drop src-address-list=ddos-attackers dst-address-list=dddos-targets
```

With the firewall filter section, we will add attackers in the "DDoS-attackers" and victims in list "ddos-targets" list:

```
/ip/firewall/filter/
add action=add-dst-to-address-list address-list=ddos-target address-list-timeout=10m chain=detect-ddos
add action=add-src-to-address-list address-list=ddos-attackers address-list-timeout=10m chain=detect-ddos
```

SYN Flood

An SYN flood is a form of DoS attack in which an attacker sends a succession of SYN requests to a target's system in an attempt to consume enough server resources to make the system unresponsive to legitimate traffic. Fortunately, in RouterOS we have specific feature for such an attack:

```
/ip/settings/set tcp-syncookies=yes
```

The feature works with sending back ACK packets that contain a little cryptographic hash, which the responding client will echo back with as part of its SYN-ACK packet. If the kernel doesn't see this "cookie" in the reply packet, it will assume the connection is bogus and drop it.

SYN-ACK Flood

An SYN-ACK flood is an attack method that involves sending a target server spoofed SYN-ACK packet at a high rate. The server requires significant resources to process such packets out-of-order (not in accordance with the normal SYN, SYN-ACK, ACK TCP three-way handshake mechanism), it can become so busy handling the attack traffic, that it cannot handle legitimate traffic and hence the attackers achieve a DoS/DDoS condition. In RouterOS, we can configure similar rules from the previously mentioned example, but more specifically for SYN-ACK flood:

```
/ip/firewall/filter add action=return chain=detect-ddos dst-limit=32,32,src-and-dst-addresses/10s protocol=tcp tcp-
flags=syn,ack
```