What is a IPIP tunnel?

Much like a proxy, a IPIP tunnel allows you to pass traffic from your BuyVM VPS including DDoS filtering to another remote destination.

IPIP tunnels allow **all traffic** through, not just HTTP. With a IPIP tunnel you can serve, and deliver any type of content from any type of server (audio, FTP, SSH, SCP, video, etc.).

What can your use a IPIP tunnel for?

IPIP tunneling is very handy when you want to use our DDoS filtering services to protect services that are too large to host with us (I.e. game servers, Java applications, large database driven applications, etc.).

IPIP tunneling is also the only tunneling method that OVH supports in their included kernels.

Don't have root access for your destination server or are running a huge Windows deployment? Check out our alternative method to redirect traffic to your remote server.

IPIP Tunnel How-to Tutorial Begins Here

Our how-to tutorial to setup a IPIP tunnel between BuyVM DDoS filtered VPS IP and your remote server starts here.

Following the simple instructions below you should be able to create a IPIP tunnel in under 20 minutes.

Supported Operating Systems

It is possible to use Windows to create, and forward your IPIP tunnel. If you need to protect a Windows server please consider purchasing a KVM plan.

In this document we'll only be covering a Linux IPIP tunnel configuration.

This guide will work 100% on both our KVM, and OpenVZ based plans.

Prerequisites

- iptables installed on your BuyVM VPS (included already in most cases)
- iproute2 (included with pretty much every recent Linux distribution)
- A kernel with IPIP support (Linux includes this by default - ipip kernel module)
- A list of ports you need forwarded to your destination
- A BuyVM VPS (starting as low as $15/yr for our 128MB OpenVZ VPS or $25/yr for our 128MB KVM VPS)

- A BuyVM DDoS filtered IP ($3.00/m per IP. 209.141.38.x & 209.141.39.x are the current filtered subnets)

Tunnel Setup

First we need to set our tunnel up.

On your BuyVM VPS please execute the following commands:

```
  echo 'net.ipv4.ip_forward=1' >> /etc/sysctl.conf  sysctl -p  iptunnel add ipip1 mode ipip local
YOUR_FILTERED_IP remote DESTINATION_SERVER_IP ttl 255  ip addr add
192.168.168.1/30 dev ipip1  ip link set ipip1 up
```

On the remote server you wish to protect run the following:

```
  iptunnel add ipip1 mode ipip local DESTINATION_SERVER_IP remote YOUR_FILTERED_IP
ttl 255  ip addr add 192.168.168.2/30 dev ipip1  ip link set ipip1 up
```

Please note the first line of each changes to mark what IP to use locally and which remotely. The 2nd line documents each end point. In a /30, 2 IP's are usable: .1 and .2.

Test your New IPIP Tunnel with Ping

On your BuyVM VPS, you should now be able to ping 192.168.168.2.

For the sake of completeness, test pinging 192.168.168.1 from your destination server.

Setup Source Route Tables

Source route entries are required to make sure data that came in via the IPIP tunnel is sent back out the IPIP tunnel.

Please execute the following commands on the **destination** server.

```
  echo '100 BUYVM' >> /etc/iproute2/rt_tables  ip rule add from 192.168.168.0/30 table BUYVM
ip route add default via 192.168.168.1 table BUYVM
```

Please note that the echo command only needs to be ran once. The entry will be saved into /etc/iproute2/rt_tables until you remove it manually.

Initial NAT Entries to Move Data over IPIP Tunnel

NAT is used to pass data over our IPIP and out the other end.

While it would be possible to use a KVM based VPS with a purchased /29 allocation, this guide doesn't cover that.

On your BuyVM VPS run the following command:

```
  iptables -t nat -A POSTROUTING -s 192.168.168.0/30 -j SNAT --to-source
YOUR_FILTERED_IP
```

Test Outbound Connections

On your destination server you can run either of the following commands to see if the tunnel is passing traffic properly:

```
  curl http://www.cpanel.net/showip.cgi --interface 192.168.168.2  wget
http://www.cpanel.net/showip.cgi --bind-address=192.168.168.2 -q -O -
```

The IP dumped should be your BuyVM filtered IP.

Forwarding Ports Over your IPIP Tunnel

To make things easy, we'll forward **all** ports from our filtered IP to the backend server. You can change this rule to only forward certain ports if you like.

Please adjust, and run the following commands on your BuyVM VPS:

```
  iptables -t nat -A PREROUTING -d YOUR_FILTERED_IP -j DNAT --to-destination
192.168.168.2  iptables -A FORWARD -d 192.168.168.2 -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT
```

The first rule sets up the actual port forwarding and the second rule makes sure that connections get NAT'd, and matched back properly.

At this point you should be able to connect to YOUR_FILTERED_IP and the destination port with your application and get passed through the IPIP tunnel without issue.

Restarting your IPIP Tunnel After Rebooting

You can edit /etc/rc.local with your favourite editor of choice (or WINSCP even) and place all the commands we just ran before the exit 0 at the bottom.

Your distribution of choice (like Debian) may have hooks in /etc/network/interfaces to bring your IPIP tunnels up at boot time but that's outside the scope of this guide.