Written by BiRU
Monday, 14 November 2016 11:42 -

What is a GRE tunnel?

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

GRE tunnels allow **all traffic** through, not just HTTP. With a GRE 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 GRE tunnel for?

GRE 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.).

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.

Note: If you are tunneling to an OVH server, you most likely don't have GRE support in your kernel. You'll need to use a IPIP tunnel instead.

GRE Tunnel How-to Tutorial Begins Here

Our how-to tutorial to setup a GRE 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 GRE tunnel in under 20 minutes.

Supported Operating Systems

It is possible to use Windows to create, and forward your GRE 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 GRE 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 GRE support (Linux includes this by default - ip_gre 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 gre1 mode gre local
YOUR_UNFILTERED_IP remote DESTINATION_SERVER_IP ttl 255  ip addr add
192.168.168.1/30 dev gre1  ip link set gre1 up
```

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

```
  iptunnel add gre1 mode gre local DESTINATION_SERVER_IP remote
YOUR_UNFILTERED_IP ttl 255  ip addr add 192.168.168.2/30 dev gre1  ip link set gre1 up
```

You will always want to form your GRE with your **unfiltered** IP address for all GRE tunnels to make sure you don't run into any sort of MTU issues or trigger the DDOS protection.

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 GRE 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 GRE tunnel is sent back out the GRE 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 GRE Tunnel

NAT is used to pass data over our GRE 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 ! -o gre+ -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 GRE Tunnel

To make things easier, we'll forward all ports to the backend server.

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
```

If you're wanting to get more specific, you could add:

```
  -p tcp --dport 25565
```

If you just wanted to protect a minecraft server for instance.

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 GRE tunnel without issue.

Restarting your GRE 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 GRE tunnels up at boot time but that's outside the scope of this guide.