

---

**guardhub**

**Jason R. Rokeach**

**Aug 04, 2019**



**CONTENTS:**

<b>1</b>	<b>guardhub package</b>	<b>1</b>
1.1	Submodules	1
1.2	guardhub.push_peers module	1
1.2.1	Command Line Usage	1
1.3	Module contents	2
<b>2</b>	<b>Indices and tables</b>	<b>3</b>
	<b>Python Module Index</b>	<b>5</b>
	<b>Index</b>	<b>7</b>



## GUARDHUB PACKAGE

### 1.1 Submodules

### 1.2 guardhub.push\_peers module

Configure servers to permit new Wireguard peers

#### 1.2.1 Command Line Usage

```
push_peers.py [-h] --pubkey PUBKEY -i IPADDRESS --desc DESCRIPTION [-y] -s PARAMETER_
→ [PARAMETER ...]
```

optional arguments:

- h, --help** show this help message and exit
- pubkey PUBKEY** Peer's public key
- i IPADDRESS, --ip IPADDRESS** Peer's WG IPv4/IPv6 address(es) Multiple instances accepted.
- desc DESCRIPTION** Peer's description
- y, --yes-all** Bypass confirmation prompts
- s PARAMETERS** Parameters for each server to which configuration should be pushed. Supported connectors are:
  - *vyos*

-s supports two formats:

  - **inline:**            -s HOSTNAME CONNECTOR INTERFACE  
                         USERNAME PASSWORD [PORT]
  - **file:** -s @FILENAME

Specify a newline delimited file with the same arguments. Multiple instances of -s may be used.

**class** guardhub.push\_peers.**Server** (*hostname, connector, interface, username, password, port=None*)

Bases: object

Server to which configuration should be pushed.

**Parameters**

- **hostname** (*str*) – Hostname/IP of device to push configuration to.
- **connector** (*str*) – Connector to use to configure server
- **interface** (*str*) – Interface name
- **username** (*str*) – Username for configuration session
- **password** (*str*) – Password for configuration session
- **port** (*int*) – Port for configuration session (Default: None)

`guardhub.push_peers.push_peer_vyos(server, pubkey, ips_with_prefixlen, peer_description, yes_all=False)`

Pushes configuration for a peer to a VyOS router.

**Parameters**

- **server** (*Server*) – Server to configure
- **pubkey** (*str*) – Public key of peer
- **ips\_with\_prefixlen** (*list*) – List of allowed IPs (v4 or v6) for peer in CIDR notation
- **peer\_description** (*str*) – Description of peer
- **yes\_all** (*bool*) – Bypass all prompts (Default: False)

`guardhub.push_peers.push_peers(pubkey, ipnets, description, servers, yes_all=False)`

Pushes configuration to multiple servers

**Parameters**

- **pubkey** (*str*) – Public key of peer
- **ipnets** (*list*) – List of allowed IPs as `IPv4Network` or `IPv6Network`
- **description** (*str*) – Description of peer
- **servers** (*list*) – List of *Server* objects to push configuration
- **yes\_all** (*bool*) – Bypass all prompts (Default: False)

## 1.3 Module contents

## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`





## PYTHON MODULE INDEX

### g

guardhub, [2](#)

guardhub.push\_peers, [1](#)



## INDEX

### G

`guardhub` (*module*), 2

`guardhub.push_peers` (*module*), 1

### P

`push_peer_vyos()` (*in module guardhub.push\_peers*), 2

`push_peers()` (*in module guardhub.push\_peers*), 2

### S

`Server` (*class in guardhub.push\_peers*), 1