
PhanTap (Phantom Tap)



PhanTap is an ‘invisible’ network tap aimed at red teams. With limited physical access to a target building, this tap can be installed inline between a network device and the corporate network. PhanTap is silent in the network and does not affect the victim’s traffic, even in networks having NAC (Network Access Control 802.1X - 2004). PhanTap will analyze traffic on the network and mask its traffic as the victim device. It can mount a tunnel back to a remote server, giving the user a foothold in the network for further analysis and pivoting. PhanTap is an OpenWrt package and should be compatible with any device. The physical device used for our testing is currently a small, inexpensive router, the GL.iNet GL-AR150. You can find a detailed blogpost describing PhanTap [here](#)

Features:

- Transparent network bridge.
- Silent : no arp, multicast, broadcast.
- 802.1x passthrough.
- Automatic configuration:
 - capture traffic entering the network (the source is non RFC1918 and the destination is RFC1918), destination IP and MAC is our victim, source MAC is our gateway,
 - SNAT bridge traffic to the victim MAC and IP address,
 - set the router default gateway to the MAC of the gateway detected just before.
- Introspects ARP, multicast and broadcast traffic and adds a route to the machine IP address and adds the machine MAC address to the neighbour list, hence giving the possibility of talking to all the machines in the local network.
- Learns the DNS server from traffic and modifies the one on the router so that it’s the same.
- Introspects DHCP packets for dynamic reconfiguration.
- Can run commands (ex: /etc/init.d/openvpn restart) when a new IP or DNS is configured.
- Lets you choose any VPN software, for example OpenVPN tcp port 443 so it goes through most firewalls.
- You can talk to the victim machine (using the gateway IP).

Setup

PhanTap has been tested with the GL.iNet GL-AR150. This device has two separate network interfaces in OpenWrt (eth0, eth1). If your device is using an internal switch(swconfig based) with interfaces like eth0.1, eth0.2, some special traffic might be blocked, e.g. 802.1Q(tagged vlan), but PhanTap should work.

- Install a snapshot build, for the GL.iNet GL-AR150
- Update the OpenWrt package list

```
1 opkg update
```

- Install PhanTap package:

```
1 opkg install phantap
```

- Configure the Wifi and start administering the router through it.
- Either reboot the device, or run `/etc/init.d/phantap setup`.
- Get the interface names from that device:

```
1 # uci show network | grep -E 'device=|ports='
2 network.loopback.device='lo'
3 network.@device[0].ports='eth0'
4 network.lan.device='br-lan'
5 network.wan.device='eth1'
6 network.wan6.device='eth1'
7 network.phantap.device='br-phantap'
```

In this example we are using a GL-AR150, which only has 2 interfaces.

- Remove the interfaces from any network interface they might be used by, if that's the case, via the following commands in the cli (assuming we are using a GL-AR150):

```
1 uci delete network.@device[0].ports
2 uci delete network.wan.device
3 uci delete network.wan6.device
```

- Add the interfaces to the phantap bridge and restart the network service via the following commands in the cli (assuming we are using a GL-AR150):

```
1 uci add_list network.br_phantap.ports='eth0'
2 uci add_list network.br_phantap.ports='eth1'
3 uci commit network
4 /etc/init.d/network reload
```

-
- Phantap is now configured, as soon as you plug it between a victim and their switch, it will automatically configure the router and give it Internet access.
 - You can add your favorite VPN to have a remote connection back. We've tested PhanTap with OpenVpn, port TCP 443, to avoid some detection methods.
 - You can also add a command to be ran when a new IP or DNS is configured, in `/etc/config/phantap`, e.g. `/etc/init.d/openvpn restart` (restart OpenVpn service).
 - You can also look at disabling the wifi by default and using hardware buttons to start it (<https://openwrt.org/docs/guide-user/hardware/hardware.button>).

Limitations or how it can be detected :

- The GL.iNet GL-AR150 and most inexpensive devices only support 100Mbps, meanwhile modern network traffic will be 1Gbps.
- The network port will stay up, switch side, when the victim device is disconnected/shutdown.
- Some traffic is blocked by the Linux bridge (STP/Pause frames/LACP).
- OpenWrt failsafe mode sends IPv4 and IPv6 packets as described in the documentation: https://openwrt.org/docs/guide-user/troubleshooting/failsafe_and_factory_reset. This happens during early boot and can get the device detected. There is no easy solution to disable it at runtime, as this configuration is in U-Boot. The U-Boot partition is mounted as read-only and it's configuration can only be accessed and modified from the U-Boot shell (via UART on the GL-AR150 for example). The easier solution is to compile OpenWrt with the `TARGET_PREINIT_DISABLE_FAILSAFE` option enabled.

Roadmap :

- Add logic to restart the detection when the links go up/down.
- Add IPv6 support.
- Test limitations of devices that have switches(`swconfig`) instead of separate interfaces.