Bypassing Tunnels: Leaking VPN Client Traffic by Abusing Routing Tables

Nian Xue, Yashaswi Malla, Zihang Xia, Christina Pöpper, and Mathy Vanhoef
Contributions

We make VPN clients leak traffic
› By manipulating the client’s routing table
› Attacks are independent of the crypto protocol

Tested 67+ VPN clients
› >248 experiments → 66% attack success
› Every VPN is vulnerable on at least one OS

→ Widespread design issues!
Usage of VPNs: watch videos from other country
Usage of VPNs: protect your traffic

› Identify website visits: IP address, plaintext DNS, SNI,…
› Attack TLS: no cert check, sslstrip, academic attacks,…
Usage of VPNs: protect your traffic

› Defend against untrusted Wi-Fi & compromised core routers

› Research goal: can we trick the client into leaking packets?
  » Yes, by manipulating the client’s routing table → ~66% vulnerable!
  » Attacks are independent of the crypto protocol
Background: VPN client routing table

```
$ ip route  # Detailed output
1  default via 10.0.0.1 dev tun0
2  192.168.1.0/24 dev eth0 proto kernel scope link src 192.168.1.2 metric 100
3  2.2.2.2 via 192.168.1.2 dev eth0
```
1. By default, send packets over tun0 = over the VPN tunnel

```bash
$ ip route       # Simplified output
1 default via tun0
```
1. By default, send packets over tun0 = over the VPN tunnel

2. **LocalNet exception**: local network is directly accessible
# Background: VPN client routing table

```
$ ip route  # Simplified output
1  default via tun0
2  192.168.1.0/24 via eth0
3  2.2.2.2 via eth0
```

1. By default, send packets over tun0 = over the VPN tunnel
2. **LocalNet exception**: local network is directly accessible
3. **ServerIP exception**: avoid re-encryption of VPN packets
We assume secure DNS behavior

Can’t trust the network’s DNS server

```bash
$ cat /etc/resolv.conf
nameserver 6.6.6.6
```
We assume secure DNS behavior

Can’t trust the network’s DNS server

1. Once connected, VPN client sets a trusted DNS server
2. DNS is sent through the VPN tunnel
   + we assume other routing-based attacks are prevented

$ cat /etc/resolv.conf
nameserver 2.2.2.3
LocalNet attack
LocalNet attack

Local network is 1.2.3.0/24

Create VPN tunnel with 2.2.2.2

Set trusted DNS server

default via tun0
1.2.3.0/24 via eth0

Target.com
1.2.3.4

2.2.2.2
LocalNet attack

default via tun0
1.2.3.0/24 via eth0
LocalNet attack

Visit random.com

Visit target.com

Send to 1.2.3.4

Intercept traffic!

default via tun0
1.2.3.0/24 via eth0

Target.com
1.2.3.4

2.2.2.2

Leak
LocalNet attack: 195 experiments

<table>
<thead>
<tr>
<th>VPN Provider</th>
<th>Class</th>
<th>OS</th>
<th>Version Number</th>
<th>LAN Setting</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Free</td>
<td>Windows</td>
<td>Windows 10 Pro</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Free</td>
<td>Windows</td>
<td>Windows 11 Pro</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Free</td>
<td>macOS</td>
<td>Ventura 13.0.1</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Free</td>
<td>iOS</td>
<td>iOS 16.1.1</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Free</td>
<td>Android</td>
<td>Android 8.1.0</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Free</td>
<td>Android</td>
<td>Android 12</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td>OS Built-in VPN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Free</td>
<td>Windows</td>
<td>2022.10.106.0</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Free</td>
<td>Linux</td>
<td>2022.9.591</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Free</td>
<td>macOS</td>
<td>2022.10.107.0</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Free</td>
<td>iOS</td>
<td>6.16</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Free</td>
<td>Android</td>
<td>6.17</td>
<td>No</td>
<td>N/A</td>
</tr>
</tbody>
</table>

1.1.1.1
### LocalNet attack: 195 experiments

<table>
<thead>
<tr>
<th>VPN Name</th>
<th>Version</th>
<th>OS</th>
<th>Result</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExpressVPN</td>
<td></td>
<td>Windows</td>
<td>12.37.0</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Linux</td>
<td>3.36</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>macOS</td>
<td>11.12.0</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>iOS</td>
<td>11.70.0</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Android</td>
<td>10.63.2</td>
<td>Yes</td>
</tr>
<tr>
<td>VPN Proxy Master</td>
<td>Free</td>
<td>iOS</td>
<td>2.1.5</td>
<td>No</td>
</tr>
</tbody>
</table>

### ExpressVPN
- Paid
- Windows
- 12.37.0
- Yes

### ExpressVPN
- Paid
- macOS
- 11.12.0
- Yes

### ExpressVPN
- Paid
- iOS
- 11.70.0
- Yes

### ExpressVPN
- Paid
- Android
- 10.63.2
- Yes

### VPN Proxy Master for iPhone
- Free
- iOS
- 2.1.5
- No
LocalNet attack: summary

- **Android**: 21.4% Vulnerable, some Blocks non-RFC1918 local traffic, 78.6% Secure
- **Linux**: 35.7% Vulnerable, 64.3% Secure
- **Windows**: 66.7% Vulnerable, 33.3% Secure
- **macOS**: 87.5% Vulnerable, 12.5% Secure
- **iOS**: 100% Vulnerable, 0% Secure
DEMO
Selected special cases

Some clients block traffic to local network
  › Problem when local network uses public IPs
  › Traffic to these public IPs gets blocked!

VPN Proxy Master for iPhone (and others)
  › DNS server returns private-use IP addresses
  › VPN server forwards traffic to real IP address
The iOS case

Prevent attacks by setting includeAllNetworks=True

› And excludeLocalNetworks=False on iOS ≥ 14.2
› Causes reliability issues, vendors hesitant to enable this

Result is that **iOS remains less secure**

› Context: VPNs on iOS were already known to leak traffic in certain scenarios.
› E.g., OS traffic may leak, leaks when switching networks,…
We were warned in the past…

Andrew Ayer: Hardening OpenVPN for DEF CON (2015)
› Guide for OpenVPN on Linux
› Essentially suggested the risk of LocalNet attacks!

Unclear how widespread this issue (already) was at the time
› VPN clients were not systematically tested → vendors were not warned, so clients never were not audited either
› Using domain names would still enable ServerIP attacks…
ServerIP attack

DNS request for vpn.com

Spoof DNS reply: 1.2.3.4

Create VPN tunnel with 1.2.3.4

Redirect to 2.2.2.2

Set trusted DNS server

default via tun0
1.2.3.4 via eth0

Target.com
1.2.3.4

2.2.2.2
ServerIP attack

default via tun0
1.2.3.4 via eth0

Target.com
1.2.3.4

2.2.2.2
ServerIP attack

- Default via tun0
- 1.2.3.4 via eth0

- Visit random.com
- Visit target.com
- Send to 1.2.3.4

Intercept traffic!
ServerIP attack: 53 experiments

› Many **built-in clients** are affected (Windows, macOS, Linux)
› Legacy built-in VPN on **Android 11 and below** was affected
› Most iOS/Android apps not vulnerable

Impact: can leak traffic to single IP address

› Can target the DNS server set by the VPN client 😊
› Or repeat the attack for different IPs…
DEMO

```
mathy@mathy-VirtualBox:~/vpn_tester$
mathy@mathy-VirtualBox:~/vpn_tester$
mathy@mathy-VirtualBox:~/vpn_tester$
mathy@mathy-VirtualBox:~/vpn_tester$ I
```
Defenses

**LocalNet attack**: disable local network access when it’s using public IP addresses.
  › Or allow local network access when using 192.168.* or alike

**ServerIP Attack**: send all traffic over VPN, except packets generated by VPN process
  › On Linux, you can use fwmark (policy-based routing)
  › Or quick fix: use secure DNS to get VPN server’s IP address
Disclosure

› Reported to CERT/CC on May 10, 2023

› Reported to selected vendors that had a security contact:
  › Some had no e-mail contact, only a bug bounty program
  › In report say we deviate from T&Cs and reserve right to disclose
Disclosure: special cases

Dubai-based ClarioVPN
› Initially: “MitM attacks are out of scope”
› Later: “Clario isn’t interested in participating in this multi-party disclosure on VPN security”

Ivanti Pulse Secure
› Provided a test server! But at first didn’t work
› Kept asking for time-consuming recordings
› Seems like they didn’t try our PoC script…
Conclusion

› Two wide-spread flaws in VPN clients
› In hindsight easy attack, but ~66% vulnerable
› Bad integration of protocols into real systems

› Defense: more carefully configure routing tables
› OS should have API to create VPN tunnels
Questions?

› Two wide-spread flaws in VPN clients
› In hindsight easy attack, but ~66% vulnerable
› Bad integration of protocols into real systems

› Defense: more carefully configure routing tables
› OS should have API to create VPN tunnels