Review of

OpenVPN is Open to VPN Fingerprinting

By Diwen Xue, Reethika Ramesh, Arham Jain, Michalis Kallitsis, J.Alex halderman, Jedidiah R. Crandall, Roya Ensafi

What is the problem?

VPN adoption has seen steady growth over the past decade due to increased public awareness of privacy and surveillance threats. In response various ISPs , advertisers and governments are now seeking to track or block VPN traffic in order to maintain visibility and control over the traffic within their jurisdiction. Examples:

1. Binxing Fang, the designer of the great firewall of China(GFW) said there is an “eternal war” between the Firewall and VPNs and the country has ordered ISPs to report and block personal VPN usage
2. ISPs, even less-powerful ones, now have access to technologies such as carrier-grade deep packet inspection(DPI) with which they can implement more sophisticated modes of detection based on protocol semantics.

In this paper, they seek to answer 2 research problems:

a. Can ISPs and governments identify traffic flows as OpenVPN connections in real time?
b. Can they do so at scale without incurring significant collateral damage from false positives?

Summary

- The paper develops mechanisms to accurately fingerprint OpenVPN connection based on 3 protocol features:

1. Byte Pattern
2. Packet Size
3. Server Response

- Playing the role of an attacker, they develop a 2 phased framework which performs passive fingerprinting followed by active proving, in sequence. This is inspired by the architecture of the Great Firewall consisting of *Filter* and *Prober* components.

A *filter* performs passive filtering over passive network trafficking real time, exploiting protocol quirks identified in the OpenVPN’s handshake stage

After a flow is flagged by a *Filter,* the IP and port information is passed to a *Prober* that performs active probing as confirmation. The probers send a set of pre-defined probes specifically designed to fingerprint an OpenVPN server.

Finally, probed servers that are confirmed as Openvpn servers are logged for manual analysis

This 2 phase framework is capable of processing ISP-scale traffic at line-speed with an extremely low positive rate. They evaluate their framework in partnership with a million user ISP-MeritY and find that they identify 85% of OpenVPN connections with negligible false positives, suggesting that OpenVPN services can be effectively blocked with little collateral damage.They identify 39 out of 40 vanilla configurations. Additionally the framework identifies 34 out of 41 obfuscated connections as well

Obfuscated VPN:

Obfuscated VPN services, whose operators often tout them as “invisible” and “unblockable”, typically use OpenVPN with an additional obfuscation layer to avoid detection. Techniques:

1. OpenVPN XOR patch: Originally developed by ClayFace, the XOR patch scrambles a packet by either XOR-ing the bytes with a pre-shared key, reversing the order of bytes, xoring each byte with its position, or a combination of these steps.
2. OpenVPN encrypted tunnels:
   Some VPN services wrap OpenVPN traffic inside encrypted tunnels to prevent DPI fingerprinting. Some of the obfuscation tunnels are Obfsproxy, Stunnel, Websocket tunnel etc
3. Proprietary protocols

- They identify two-thirds of obfuscated OpenVPN flows. Most of the obfuscated traffic resemble the vanilla open VPN with an XOR patch
- Lack of random padding and co-location with vanilla Openvpn servers make the obfuscated servers more vulnerable to detection.

Key Insights

- Unlike circumvention tools like Tor and Refraction networking, which employ sophisticated techniques to avoid detection, robust obfuscation techniques have been absent from the VPN ecosystem. In the long term, a cat-and-mouse game similar to that between Great Firewall and Tor is imminent in the VPN ecosystem. Hence, there is a marked demand for an emerging class of services called “stealth” or “obfuscated” VPN, especially from users in countries with heavy censorship or laws against personal VPN usage.(project Idea??)
- **A packet field taking a fixed number of values can be easy to fingerprint**
- **For OpenVPN, the presence of explicit ACK packets, uniform in size and only seen in some parts of a session provides another fingerprintable feature. Solution: VPN providers should switch from static to random padding for their obfuscated services.**
- Although an application may not respond to probing, an attacker may still be able to fingerprint application-specific thresholds at TCP level, such as timeouts or RST thresholds. Solution: server specific random delay as in obfs4
- Sharing infrastructure between VNn services and other services such as TLS or obfuscation services, leaves the VPN server guilty by association and leaves it

vulnerable to identification. Solution: VPN providers offering both vanilla and obfuscated VPN services should avoid co-locating them.

Strengths

- They evaluated the practicality of their framework in partnership with a mid-tier ISP.
- They put a lot of stress on Ethics, privacy and responsible disclosure since raw networ traffic contains real user's data and is highly sensitive:

a. They cleared their research plan with university counsel ans IRB. Although IRB determined that the work is not regulated, they take measure to minimise potential risk for end user
b. *Fiter* analysed only the first payload byte, completely ignoring the remainder of the payload. The raw snapshot was never inspected by humans.
c. The logs were stored and analysed on server that is securely maintained by Merit

Limitations/Weaknesses

- Source code was not published or could not be published due to the fear of malicious agents

Summary of Key Results

- Tracking and blocking the use of Open-VPN, even with the most current obfuscation methods, is straightforward and within reach of any ISP or network operator, as well as any nation state adversary. Overall they identified 1718 out o 2000 vanilla flows corresponding to 39 out of 40 unique configurations
- 4 out of 5 VPN providers use XOR-based obfuscation which is easily fingerprintable. They identify over two thirds of all obfuscated flows, corresponding to 34 out of 41 obfuscated configurations.
- UDP and obfuscated servers often share infrastructure with vanilla TCP servers leaving them "guilty by association" and hence giving away their identify

Open Questions?

- Can VPN providers develop more standardised obfuscation solutions, such as Pluggable transports
- Will the VPN ecosystem see the same cat-and-mouse game?