CVE-2020-15590 - Private Internet Access VPN for Linux - Exposure of Sensitive Information to an Unauthorized Actor

by Sick Codes — August 31, 2020 - Updated on November 24, 2020 in Security, Tutorials 1

CVE-2020-15590

Private Internet Access VPN for Linux - Exposure of Sensitive Information to an Unauthorized Actor

CVSS Score

7.5

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N

Internal ID

SICK-2020-001

Vendor

London Trust Media, Inc

Product

Private Internet Access VPN Client for Linux

Product Version:

v1.5 thru 2.3.1 - Fixed in v2.4.0+

Vulnerability Details

A vulnerability in the Private Internet Access (PIA) VPN Client for Linux v1.5 through v2.3+ allows remote attackers to bypass an intended VPN kill switch mechanism and read sensitive information via intercepting network traffic. Since v1.5, PIA has supported a “split tunnel” OpenVPN bypass option. The PIA killswitch & associated iptables firewall is designed to protect you while using the Internet. When the kill switch is configured to block all inbound and outbound network traffic, privileged applications can continue sending & receiving network traffic if net.ipv4.ip_forward has been enabled in the system kernel parameters. For example, a Docker container running on a host with the VPN turned off, and the kill switch turned on, can continue using the internet, leaking the host IP (CWE 200). In PIA 2.4.0+, policy-based routing is enabled by default and is used to direct all forwarded packets to the VPN interface automatically.

Vendor Response

Vendor successfully patched CVE-2020-15590 in version 2.4.0.

Disclosure Timeline

2020-07-07 - Vendor notified via Twitter DM.
2020-07-07 - Vendor requests submission of disclosure to London Trust Media, Inc.
2020-07-07 - Vendor disclosure via email.
2020-07-08 - CVE Requested.
2020-07-08 - Vendor replied that they do not consider this to be a vulnerability as Docker requires privileged access.
2020-07-08 - Vulnerability assigned CVE-2020-15590.
2020-07-08 - Researcher sent evidence of 2020 distributions including Docker in their base OS.
2020-07-09 - Vendor responded, discussing each of the Researcher’s points.

2020-07-20 - Researcher responded, agreed on mistakes in previous email. Researcher expanded on particulars, namely the definition of a “kill-switch”, and reiterated the expected integrity of software.
2020-07-21 - Researcher engaged a coordinating organization from CERT/CC list of coordinating organizations.
2020-07-28 - Vendor confirms they are working on documentation to mitigate issues for users who use Docker. Vendor advises they are working on a fix.
2020-08-21 - Vendor releases new version without notifying affected users.
2020-08-23 - Researcher urgently re-raises the issue after finding 100,000,000 docker pulls of a relevant project.
2020-08-24 - Vendor responds stating new version is already underway is in the next release.
2020-08-25 - Researcher agrees to alpha test new client.
2020-08-30 - Vendor sends alpha test client.
2020-08-31 - Researcher confirms vulnerability mitigated in PIA Linux Client 2.4.0.
2020-09-09 - Vendor releases PIA Client for Linux 2.4.0.

Credits

@sickcodes - https://twitter.com/sickcodes/ - **raised the initial reported vulnerability.**

@cje - https://twitter.com/caseyjohnellis - **third-party coordinator via** Disclose.io

Links

https://sick.codes/cve-2020-15590/

https://github.com/sickcodes/security/blob/master/advisories/SICK-2020-001.md

https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-15590

https://twitter.com/sickcodes

https://www.privateinternetaccess.com/helpdesk/news/posts/announcement-release-desktop-version-2-4-0

https://github.com/pia-foss/desktop

https://privateinternetaccess.com

https://github.com/sickcodes

https://twitter.com/caseyjohnellis

https://disclose.io/

https://sick.codes/

Verify if your configuration is affected

Applications that enable IP forwarding for bridged networking.

E.g.

- Docker
- libvirtd/QEMU, when using bridged networking

```
# check if you have enabled ip forwarding
sysctl net.ipv4.ip_forward net.ipv6.conf.all.forwarding
```

Affected output

```
# AFFECTED, update to 2.4.0
# net.ipv4.ip_forward = 1
# net.ipv6.conf.all.forwarding = 0

# AFFECTED, update to 2.4.0
# net.ipv4.ip_forward = 1
# net.ipv6.conf.all.forwarding = 1

# AFFECTED, update to 2.4.0
# net.ipv4.ip_forward = 0
# net.ipv6.conf.all.forwarding = 1

# UNAFFECTED, update to 2.4.0
# net.ipv4.ip_forward = 0
# net.ipv6.conf.all.forwarding = 0

# UNAFFECTED (if no output), update to 2.4.0 anyway
```

PIA client for Linux users should immediately update to version 2.4.0

Manual Fix for those who don’t update

ONLY RUN THE FOLLOWING COMMANDS IF YOU DO NOT UPDATE PIA

```
# These rules must be reapplied each time the system is started.
# Mark all forwarded packets with a fwmark
iptables -w -A PREROUTING -j MARK --set-mark 0x6789 -t mangle
ip6tables -w -A PREROUTING -j MARK --set-mark 0x6789 -t mangle
# Create a routing table to direct forwarded packets to a specific interface, or blackhole if that interface disappears
# NOTE:
# - Interface route must be recreated if the interface is destroyed (i.e. a VPN tun device that is disconnected)
# - The interface name may differ from tun0
# - If directing to the physical interface instead of the VPN, also specify your gateway
ip route add table 26505 default dev tun0
ip route add table 26505 blackhole default metric 32000 # Leak protection route in case the route above is deleted
ip -6 route add table 26505 default dev tun0
ip -6 route add table 26505 blackhole default metric 32000 # Leak protection route in case the route above is deleted
# Create routing rules to direct forwarded packets to that routing table, while still permitting LAN routes from the main table
ip rule add from all fwmark 0x6789 lookup main suppress_prefixlen 1 prio 2000
ip rule add from all fwmark 0x6789 lookup 26505 prio 2001
ip -6 rule add from all fwmark 0x6789 lookup main suppress_prefixlen 1 prio 2000
ip -6 rule add from all fwmark 0x6789 lookup 26505 prio 2001
```

PIA Linux Client 2.3.2 Alpha Test Results

PIA Linux Client Version 2.3.2 Successfully Mitigates CVE-2020-15590
Noteworthy Result

Docker Container Killswitch ON

PIA VPN	ON	ON	OFF	OFF
Bypass	YES	NO	YES	NO
IP	Host	VPN	-	-

QEMU Bridged Networking Killswitch ON

PIA VPN	ON	ON	OFF	OFF
Bypass	YES	NO	YES	NO
IP	Host	VPN	-	-

QEMU Usermode Networking Killswitch ON

PIA VPN	ON	ON	OFF	OFF
Bypass	YES	NO	YES	NO
IP	VPN*	VPN	-	-

**as expected since using host nic*

LAN (Tested SSH) Killswitch ON

PIA VPN	ON	ON	OFF	OFF
Allow LAN	YES	NO	YES	NO
Host	Reachable	Refused	Reachable	Refused

Docker Container Killswitch OFF

PIA VPN	ON	ON	OFF	OFF
Bypass	YES	NO	YES	NO
IP	Host	VPN	Host	Host

QEMU Bridged Networking Killswitch OFF

PIA VPN	ON	ON	OFF	OFF
Bypass	YES	NO	YES	NO
IP	Host	VPN	Host	Host

QEMU Usermode Networking Killswitch OFF

PIA VPN	ON	ON	OFF	OFF
Bypass	YES	NO	YES	NO
IP	VPN*	VPN	Host	Host

**as expected due to using host nic*

LAN SSH Killswitch OFF

PIA VPN	ON	ON	OFF	OFF
Allow LAN	YES	NO	YES	NO
Host	Reachable	Reachable*	Reachable	Reachable

**as expected since killswitch off*

Tested by @sickcodes on 2020-08-31

PIA Linux Client 2.3.2 Alpha Test Results

PIA Linux Client Version 2.3.2 Successfully Mitigates CVE-2020-15590
Noteworthy Result

Docker Container Killswitch ON

PIA VPN	ON	ON	OFF	OFF
Bypass	YES	NO	YES	NO
IP	Host	VPN	-	-

QEMU Bridged Networking Killswitch ON

PIA VPN	ON	ON	OFF	OFF
Bypass	YES	NO	YES	NO
IP	Host	VPN	-	-

QEMU Usermode Networking Killswitch ON

PIA VPN	ON	ON	OFF	OFF
Bypass	YES	NO	YES	NO
IP	VPN*	VPN	-	-

**as expected since using host nic*

LAN (Tested SSH) Killswitch ON

PIA VPN	ON	ON	OFF	OFF
Allow LAN	YES	NO	YES	NO
Host	Reachable	Refused	Reachable	Refused

Docker Container Killswitch OFF

PIA VPN	ON	ON	OFF	OFF
Bypass	YES	NO	YES	NO
IP	Host	VPN	Host	Host

QEMU Bridged Networking Killswitch OFF

PIA VPN	ON	ON	OFF	OFF
Bypass	YES	NO	YES	NO
IP	Host	VPN	Host	Host

QEMU Usermode Networking Killswitch OFF

PIA VPN	ON	ON	OFF	OFF
Bypass	YES	NO	YES	NO
IP	VPN*	VPN	Host	Host

**as expected due to using host nic*

LAN SSH Killswitch OFF

PIA VPN	ON	ON	OFF	OFF
Allow LAN	YES	NO	YES	NO
Host	Reachable	Reachable*	Reachable	Reachable

**as expected since killswitch off*

Tested by @sickcodes on 2020-08-31

Comments 1

Pingback: OSX - Run Mac in a Docker container - Krypto Tech Lens

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment *

Name

Email

Website

Privacy - Terms

I am human

POST COMMENT

@sickcodes

@sickcodes

@sickcodes

Discord Server

sickcodes.slack.com

t.me/sickcodeschat

./contact_form