Talos Vulnerability Report

TALOS-2022-1554

Abode Systems, Inc. iota All-In-One Security Kit web interface /action/factory* authentication bypass vulnerability

OCTOBER 20, 2022

CVE NUMBER

CVE-2022-29477

SUMMARY

An authentication bypass vulnerability exists in the web interface /action/factory* functionality of Abode Systems, Inc. iota All-In-One Security Kit 6.9X and 6.9Z. A specially-crafted HTTP header can lead to authentication bypass. An attacker can send an 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

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

CWE

CWE-798 - Use of Hard-coded Credentials

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.

The majority of these endpoints are protected with a username and password. However, there are a handful of endpoints—the `factory` endpoints—which allow a different form of authentication that relies on a hard-coded HTTP header.

When an HTTP request is received, a function named `web_auth_check` is called to determine whether the request is authenticated and authorized to proceed. The function does not differ from 6.9X to 6.9Z. In 6.9Z this function can be found at offset `0x19F714` of the `/root/hpgw` binary. The relevant portion of the decompilation of this function is included below.

```
int __fastcall web_auth_check(mg_connection *conn, mg_request_info *ri, int
check_setup)
{
  int session_id;
  char *user_agent;
  size_t agent_len;
  const char *uri;
  int result;
  char *x_climax_tag;
  char *auth_header;
  char *basic_auth_content;
  int buffer_len;
  _BYTE *strBase64;
  const char *error_str;
  char admin_username[192];
  char admin_password[192];
  char* buffer;

  do_trace(1, "web_auth_check", 527);
  ...
  if ( !in_setup_mode() )
    goto LABEL_6;
  uri = ri->uri;
  if ( !strcmp(uri, "/action/welcomeGet") || !strcmp(uri, "/action/logout") ||
!strcmp(uri, "/action/devStatusGet") )
  {
    // These endpoints are available to all users, without authentication
    do_trace(1, "web_auth_check", 557);
    return 0;
  }
  if ( !in_setup_mode()
    || (uri = ri->uri, (result = strcmp(uri, "/action/wlSiteSurveyList")) != 0)
    && (result = strcmp(uri, "/action/wirelessConnect")) != 0
    && (result = strcmp(uri, "/action/wirelessPost")) != 0
    && (result = strcmp(uri, "/action/xmppGet")) != 0
    && (result = strcmp(uri, "/action/reset")) != 0 )
  {
    // If the device is being set up through the direct-connect access point mode
then
    // the endpoints needed to configure the device's network connection do not need
to be authenticated

// [0] For vulnerability to work, we must arrive at LABEL_6 from the goto above
LABEL_6:
    do_trace(1, "web_auth_check", 574);

    // [1] IF the URI starts with /action/factory
    if ( startswith(ri->uri, "/action/factory") {
      x_climax_tag = mg_get_header(conn, "X-Climax-Tag")

      // [2] AND IF the 'X-Climax-Tag' header exists
      if ( x_climax_tag ) {
        vsnprintf(buffer, 0x1Fu, "%s-%d-%d", "Factory", 27940001, 21245121);

        // [3] check if the header is "Factory-27940001-21245121"
        if ( strcmp(x_climax_tag, buffer) == 0) {

          // [4] return a successful authentication if the header matches
```

```
            return SUCCESS;
          }
        }
      }
      else
      {
        // [5] Otherwise navigate the standard auth path, with username and password
        ...
      }
    }
    return result;
  }
```

At `[1]` a check is conducted to identify whether the request's URI begins with `/action/factory`, as these endpoints allow a different form of authentication. At `[2]` the `X-Climax-Tag` header value is extracted from the request. At `[3]` this value is compared to a fixed value of `Factory-27940001-21245121`. If the `X-Climax-Tag` value is correct, then the request is authenticated and allowed to proceed.

The endpoints associated with `/action/factory` are:

- factoryTestWkList
- factoryTestPost
- factoryTestGet
- factorySerialMacPost
- factorySerialMacGet
- factoryRstPost
- factoryRstGet

Submitting a request to these endpoints with the appropriate `X-Climax-Tag` will allow a remote user to access pretty significant functionality related to factory testing and configuration, including factory reset.

TIMELINE

2022-07-13 - Initial Vendor Contact
2022-07-14 - Vendor Disclosure
2022-10-20 - Public Release

CREDIT

Discovered by Matt Wiseman of Cisco Talos.
