Talos Vulnerability Report

TALOS-2022-1555

# Abode Systems, Inc. iota All-In-One Security Kit XCMD doDebug denial of service vulnerability

OCTOBER 20, 2022

## CVE NUMBER

CVE-2022-32760

## SUMMARY

A denial of service vulnerability exists in the XCMD doDebug functionality of Abode Systems, Inc. iota All-In-One Security Kit 6.9X and 6.9Z. A specially-crafted XCMD can lead to denial of service. An attacker can send a malicious XML payload 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:C/C:N/I:N/A:H

## CWE

CWE-489 - Leftover Debug Code

## 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 receives command and control messages (referred to in the application as XCMDs) via an XMPP connection established during the initialization of the `hpgw` application. As of version 6.9Z there are 222 XCMDs registered within the application. Each XCMD is associated with a function intended to handle it. As discussed in TALOS-2022-1552 there is a service running on UDP/55050 that allows an unauthenticated attacker access to execute these XCMDs.

An XCMD, by virtue of being commonly transmitted over XMPP, is an XML payload structured in a specific format. Each XCMD must contain a root node `<p>`, which must contain a child element, `<mac>` with an attribute `v` containing the target device MAC Address. There must also be a child element `<cmd>` which must contain an attribute `a` naming the XCMD to be executed. From there, various XCMDs require various child elements that contain information relevent only to that handler.

For example, one of the simplest XCMDs that can be executed is `getDev`.

```
<?xml version="1.0" encoding="UTF-8"?>
<p>
  <mac v="B0:C5:CA:00:00:00"/>
  <cmds>
    <cmd a="getDev"/>
  </cmds>
</p>
```

One of the XCMDs, `doDebug`, appears to be intended for diagnostic access by developers and technical support. The `doDebug` XCMD can react several different ways, based on the children of the `<cmd>` element. For this vulnerability we focus on the `crash` child element, which results in a Denial of Service condition for the device.

The XCMD handler responsible for `doDebug` is located at offset `0x1182D0` in the `/root/hpgw` binary included in version 6.9Z. Included below are only the relevant portions of the decompilation of this function:

```
int __fastcall doDebug(xml_related_t *xcmd, int *a2)
{
  char *crash;
  ...
  // [1] Attempt to extract the value of the `crash` tag
  crash = get_xcmd_param(xcmd, "crash");
  ...
  // [2] If a `crash` parameter was extracted
  if ( crash )
  {
    log(DEBUG, XCMD, "\n===do crash===\n");
    // [3] Force a fault by writing to 0x00000000
    MEMORY[0] = 1;
  }
  ...
}
```

At `[1]` the function attempts to extract any value associated with the `crash` tag. At `[2]` it checks to see if a value was successfully extracted, and if so, then at `[3]` it will force a fault by writing to address 0, resulting in a crash of the `/root/hpgw` process. When the `hpgw` process stops, the system's watchdog is no longer maintained. After a period of time the entire system will reboot, resulting in a denial of service.

### Exploit Proof of Concept

```
<?xml version="1.0" encoding="UTF-8"?>
<p>
  <mac v="B0:C5:CA:00:00:00"/>
  <cmds>
    <cmd a="doDebug">
      <crash v="1"/>
    </cmd>
  </cmds>
</p>
```

### 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.