Talos Vulnerability Report

TALOS-2020-1006

Allen-Bradley Flex IO 1794-AENT/B ENIP Request Path Logical Segment Denial of Service Vulnerability

OCTOBER 13, 2020

CVE NUMBER

CVE-2020-6084, CVE-2020-6085

Summary

An exploitable denial of service vulnerability exists in the ENIP Request Path Logical Segment functionality of Allen-Bradley Flex IO 1794-AENT/B. A specially crafted network request can cause a loss of communications with the device resulting in denial-of-service. An attacker can send a malicious packet to trigger this vulnerability.

Tested Versions

Allen-Bradley Flex IO 1794-AENT/B 4.003

Product URLs

http://ab.rockwellautomation.com/IO/In-Cabinet-Modular/1794-FLEX-IO-Modules

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-120 - Buffer Copy without Checking Size of Input (Classic Buffer Overflow)

Details

The 1794-AENT FLEX I/O is a modular I/O platform produced by Allen-Bradley. It is designed to provide a wide range of I/O operations while keeping a smaller form factor. Communication with the device is primarily possible via EtherNet/IP (ENIP) and HTTP.

When using ENIP to communicate with the device, the SendRRData command can be used to send an encapsulated unconnected message. One field necessary for unconnected packets is the Request Path, also referred to as the EPATH or IOI. This value contains pairs of bytes, referred to as segments, that reference different parts of a CIP entity. Through use of a combination of segments, a description of the device can be represented.

Segments are structured as a bitfield, using the high three bits to indicate the segment type and the remaining to indicate the segment format. This can be seen in the table below:

```
+=====+=====+=====+=====+=====+=====+=====+=====+
|  7  |  6  |  5  |  4  |  3  |  2  |  1  |  0  |
+=====+=====+=====+=====+=====+=====+=====+=====+
|  Segment Type   |       Segment Format        |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

Of the eight possible Segment Types, seven are defined and one is reserved for future use. The breakdown for this field can be seen in the table below:

```
                        +===+===+===+
                        | 7 | 6 | 5 |
+-------------------------+===+===+===+
| Port Segment            | 0 | 0 | 0 |
+-------------------------+---+---+---+
| Logical Segment         | 0 | 0 | 1 |
+-------------------------+---+---+---+
| Network Segment         | 0 | 1 | 0 |
+-------------------------+---+---+---+
| Symbolic Segment        | 0 | 1 | 1 |
+-------------------------+---+---+---+
| Data Segment            | 1 | 0 | 0 |
+-------------------------+---+---+---+
| Data Type (constructed) | 1 | 0 | 1 |
+-------------------------+---+---+---+
| Data Type (elementary)  | 1 | 1 | 0 |
+-------------------------+---+---+---+
| Reserved                | 1 | 1 | 1 |
+-------------------------+---+---+---+
```

Each of the Segment Types then implements its own fields for the remaining bits in the field.

When a Logical Segment is chosen, the remaining bits get further broken up as shown below:

```
+=========+=========+=========+=========+=========+
|    4    |    3    |    2    |    1    |    0    |
+=========+=========+=========+=========+=========+
|        Logical Type         |  Logical Format   |
+---------+---------+---------+---------+---------+
```

Of the eight possible Logical Types, seven are defined and one is reserved for future use. The breakdown for this field can be seen in the table below:

	4	3	2
Class ID	0	0	0
Instance ID	0	0	1
Member ID	0	1	0
Connection Point	0	1	1
Attribute ID	1	0	0
Special*	1	0	1
Service ID*	1	1	0
Reserved	1	1	1

For most of the Logical Types, the Logical Format field uses a Logical Addressing Definition. If the Special or Service ID types are indicated, the Logical Format uses the following breakdown:

	1	0
Electronic Key Segment	0	0
Reserved	0	1
Reserved	1	0
Reserved	1	1

When the Electronic Key Segment is used, the byte following it is used to indicate the format of the key data. Of the possible formats, only one is not reserved for future use. The breakdown for this field can be seen in the table below:

	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0
Reserved	...							
	0	0	0	0	0	0	1	1
Key Format Table	0	0	0	0	0	1	0	0
	0	0	0	0	0	1	0	1
Reserved	...							
	1	1	1	1	1	1	1	1

CVE-2020-6084: Electronic Key Segment - Key Format Table

By sending an Electronic Key Segment with less bytes than required by the Key Format Table following the Key Format field, it is possible to make the device enter a fault state. In this state, all remote communications with the device are stopped and a physical power cycle is required to regain functionality.

CVE-2020-6085: Electronic Key Segment - Reserved Format Types 0x05-0xff

By sending an Electronic Key Segment with less than 0x18 bytes following the Key Format field, it is possible to make the device enter a fault state. In this state, all remote communications with the device are stopped and a physical power cycle is required to regain functionality.

Timeline

2020-02-11 - Vendor Disclosure
2020-04-15 - Disclosure extension provided
2020-06-30 - Vendor follow up
2020-07-24 - Talos provided 2nd disclosure extension per vendor request
2020-09-10 - Vendor request additional time; Talos provided final disclosure deadline of 2020-10-12
2020-10-12 - Public Release

CREDIT

Discovered by Jared Rittle of Cisco Talos.
