Talos Vulnerability Report

TALOS-2021-1356

Garrett Metal Detectors iC Module CMA CLI setenv command directory traversal vulnerability

DECEMBER 20, 2021

CVE NUMBER

CVE-2021-21904

Summary

A directory traversal vulnerability exists in the CMA CLI `setenv` command of Garrett Metal Detectors’ iC Module CMA Version 5.0. A specially-crafted command line argument can lead to arbitrary file overwrite. An attacker can provide malicious input to trigger this vulnerability.

Tested Versions

Garrett Metal Detectors iC Module CMA Version 5.0

Product URLs

https://garrett.com/security/walk-through/accessories

CVSSv3 Score

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

CWE

CWE-22 - Improper Limitation of a Pathname to a Restricted Directory (‘Path Traversal’)

Details

The Garrett iC Module provides network connectivity to either the Garrett PD 6500i or Garrett MZ 6100 models of walk-through metal detectors. This module enables a remote user to monitor statistics such as alarm and visitor counts in real time as well as make configuration changes to metal detectors.

The Garrett iC Module exposes an authenticated CLI over TCP port 6877. This interface is used by a secondary GUI client, “CMA Connect”, to interact with the iC Module on behalf of the user. After a client successfully authenticates they may send plaintext commands to interact with the device. This CLI is how the remote software invokes the majority of its functionality when getting and setting various device configurations.

One of the commands exposed by this service allows an authenticated user to create or modify what the application refers to as “environment variables”. These “environment variables” are tracked as key/value pairs where the value is stored as the contents of a file named after the key. This command, `setenv [key] [value]`, will result in a file being created in the `/ltrx_user/env/` directory. This filename is crafted by concatenating the string “`/ltrx_user/env/`” with the user-supplied `key`.

For reference, an approximate decompilation of the `setEnv` handler function is included below. For brevity, functionality that was not relevant to the vulnerability (such as logging, error handling and remote client interaction) has been excluded.

```
void setEnv(char* key, char* val)
{
  size_t len;
  char filename[128];
  // Please note that the function that handles receiving CLI data from remote clients
  // limits the value in `key` to less than 256 bytes
  // Therefore, this `log_buf` stack variable protects the stack frame from corruption
  // if attempting to overflow `filename` by supplying a long `key` string
  char log_buf[256];
  FILE *fd;

  strcpy(filename, "/ltrx_user/env/");
  strcat(filename, key);
  fd = fopen(filename, "w");
  len = strlen(val);
  fwrite(val, 1, len, fd);
  fclose(fd);
}
```

The `setEnv` function does not attempt to sanitize or otherwise validate the contents of the `key` parameter, allowing an authenticated attacker to supply directory traversal primitives and overwrite or create arbitrary files. The attacker can control the contents of the file to a limited degree. The contents may not contain the space or newline characters, and the length of `val` is limited to `249 - strlen(key)`. In practice, this is enough to execute arbitrary code on the system as the root user.

Exploit Proof of Concept

```
    setenv ../../../tmp/poc example
```

Timeline

2021-08-17 - Vendor Disclosure

2021-11-10 - Talos granted disclosure extension

2021-12-13 - Vendor patched

2021-12-15 - Talos tested patch

2021-12-20 - Public Release

CREDIT

Discovered by Matt Wiseman of Cisco Talos.

VULNERABILITY REPORTS

PREVIOUS REPORT

TALOS-2021-1355

NEXT REPORT

TALOS-2021-1359