Talos Vulnerability Report

TALOS-2020-1140

# Schneider Electric EcoStruxure Control Expert PLC Simulator Modbus message processing remote code execution vulnerability

DECEMBER 8, 2020

CVE NUMBER

CVE-2020-7559

## Summary

A code execution vulnerability exists in the Modbus message-processing functionality of Schneider Electric EcoStruxure Control Expert PLC Simulator 14.1. A specially crafted network request can lead to remote code execution. An attacker can send a large Modbus request to trigger this vulnerability.

## Tested Versions

Schneider Electric EcoStruxure Control Expert PLC Simulator 14.1

## Product URLs

https://www.se.com/ww/en/product-range-presentation/548-ecostruxure%E2%84%A2-control-expert/

## CVSSv3 Score

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

## CWE

CWE-121 - Stack-based Buffer Overflow

## Details

EcoStruxure Control Expert (formerly UnityPro) is Schneider Electric's flagship software for program development, maintenance, and monitoring of industrial networks. To aid in the testing process of developed programs, a device simulator is included with Control Expert. When the simulator mode is started, it opens 0.0.0.0:502 on the host, allowing for communication with the simulator via Modbus.

When a large Modbus message is sent to this simulator, it is possible to write outside the bounds of a stack buffer, allowing for remote code execution.

In function `sub_1b48b00` a stack buffer of 0x8000 bytes is reserved to hold the Modbus message data. This buffer will be used as the `dst` argument of a subsequent `memcpy` call.

```
//
// sub_1b48b00
//
...
01b48be1  lea     edx, [ebp-0x8004]       # allocate a stack buffer
01b48be7  push    edx                     # arg1
01b48be8  call    sub_1b495c0             # call to vulnerable function
...
```

Execution continues into `sub_1b495c0` where the Modbus message to be processed is copied into the previously reserved stack buffer. This is done via a `memcpy` call with the following arguments:

`dst`: pointer to the stack buffer described above `src`: pointer to the head of the current Modbus message `n` : value pulled from the current Modbus message MBAP header length field

```
//
// sub_1b495c0
//
...
01b4962a  mov     ecx, dword [ebp-0x4]
01b4962d  movzx   edx, word [ecx+0x11]
01b49631  push    edx                     # n (mbap_len) (user controlled)
01b49632  mov     eax, dword [ebp-0x4]
01b49635  mov     ecx, dword [eax+0x9]
01b49638  push    ecx                     # src (Modbus_msg_p) (user controlled)
01b49639  mov     edx, dword [ebp+0x8]
01b4963c  push    edx                     # dst (arg1)
01b4963d  call    memcpy                  # memcpy(dst, Modbus_msg_p, mbap_len)
...
```

Since both the `src` and `n` arguments are determined by the user controlled Modbus message, and since there is no check to ensure that the message length is smaller than the buffer size, it is possible to use the `memcpy` call to write outside of the reserved buffer and gain execution through corruption of the program's exception handler.

```
0:003> !exchain
0354fefc: 43434343
Invalid exception stack at 42424242
0:003> !msec.exploitable
!exploitable 1.6.0.0
Exploitability Classification: EXPLOITABLE
Recommended Bug Title: Exploitable - Exception Handler Chain Corrupted starting at MSVCR110!wcscpy_s+0x00000000000000b1
(Hash=0x996b7f71.0xf27769f2)
Corruption of the exception handler chain is considered exploitable
```

Crash Information

```
0:006> g
COMM Server Thread(1760.7c4): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=002af561 ebx=002981f8 ecx=00000001 edx=0000815d esi=002af560 edi=03550000
eip=6a4edf22 esp=0353fe20 ebp=0353fe40 iopl=0         nv up ei pl nz na po cy
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00010203
MSVCR110!wcscpy_s+0xb1:
6a4edf22 f3a4            rep movs byte ptr es:[edi],byte ptr [esi]
```

Timeline

2020-08-13 - Vendor Disclosure

2020-11-04 - CVE assigned

2020-11-09 - Vendor released

2020-12-08 - Public Release

CREDIT

Discovered by Alexander Perez-Palma and Jared Rittle of Cisco Talos.

---

VULNERABILITY REPORTS

PREVIOUS REPORT TALOS-2020-1126

NEXT REPORT TALOS-2020-1144