Case study: Combating MAC address spoofing in access networks

DEFENDING WHEN 802.1X ISN’T AN OPTION

STEVEN MCNUTT | DENSEMODE.COM

PUBLISHED: FEBRUARY 8, 2020 | UPDATED: FEBRUARY 8, 2020

Contents

Introduction

Media Access Control (MAC) addresses commonly are used to identify endpoints for purposes of access control and authorization on access layer networks that have yet to implement 802.1x (dot1x) device authentication. The problem with this approach is MAC address spoofing is trivial to implement. However, with a defense in depth approach using basic tools and techniques, the risk and impact can be largely mitigated.

To explore the issues, we are going to evaluate the case of an organization that had recently implemented a network access control solution. A network penetration tester easily bypassed their access controls by cloning a mac address from an IP phone to a Linux laptop computer,

Background

The organization had recently implemented Cisco Systems Identity Services Engine (ISE) and had hired a pen-testing firm to evaluate its efficacy in preventing unwanted access to the network. In general, Network Access Control (NAC) implementations take a phased approach to control risk and get immediate value from the tool, and this was the case here.

At the time the penetration test was performed, some of the network was using 802.1x authentication with digital certificates, and some of the network was using MAC Authentication Bypass (MAB) combined with device profiling to determine the correct level of authorization for a connecting device.

Additionally, the Authorization policies weren't fully implemented, so effectively authorization was a simple 'yes/no' result where full access is granted based on a device profile match.

When the Pentester did her work, she grabbed the MAC address off the back of an IP phone in a common area, applied it to her Linux laptop computer, and used the network cable from the IP phone to connect to the network. ISE recognized her computer as the IP phone, and she was granted unrestricted access to the network.

The information security team had the impression that ISE was able to handle a basic access layer attack like MAC address spoofing, and wanted some answers regarding how this happened and what could be done to mitigate it until they were able to roll out dot1x authentication

Analysis

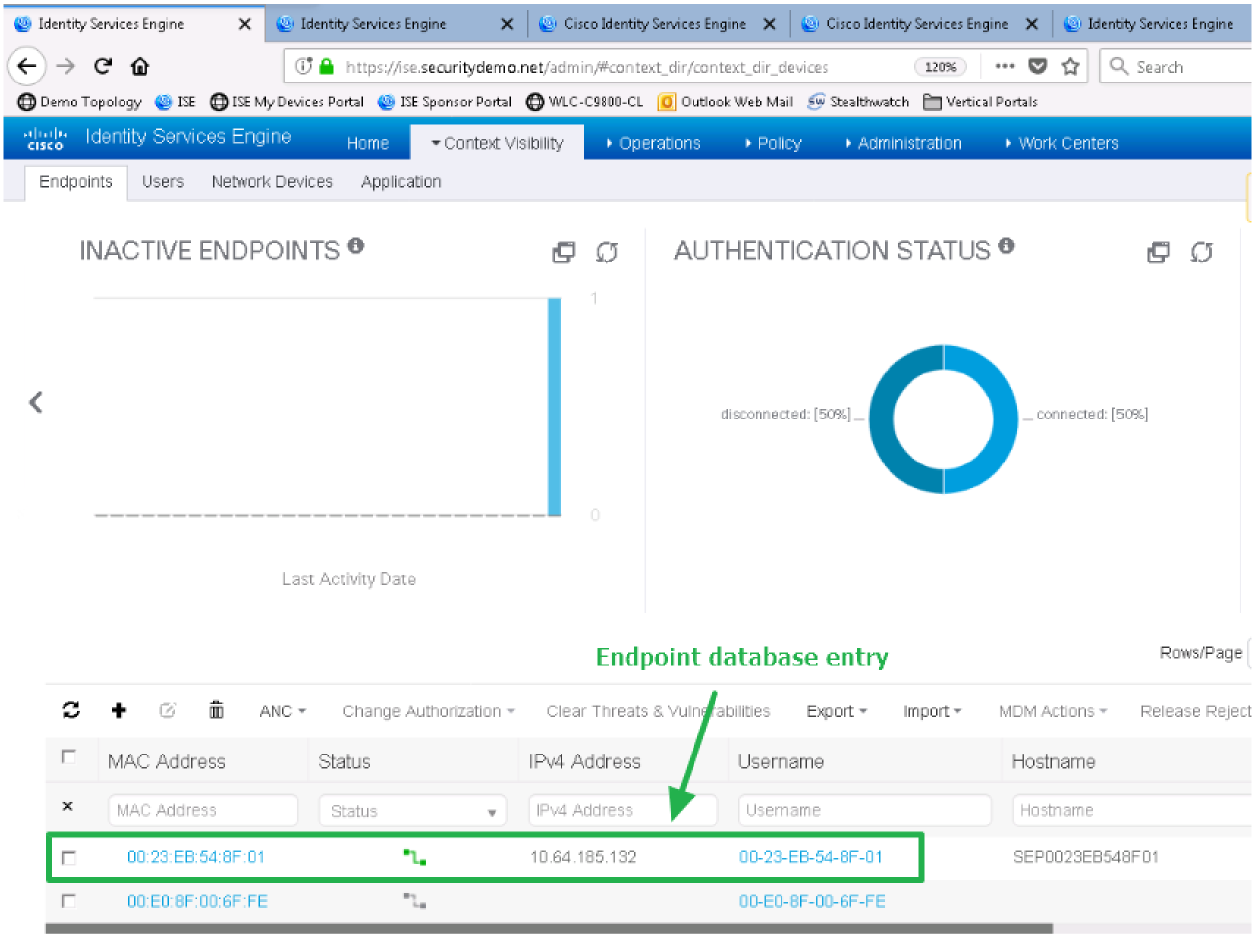
Before getting into the details, we will set the stage by briefly reviewing how the ISE profiler works. Then we will:

1. Dive into what happens when Windows and Ubuntu Linux devices connect to the network with the same MAC address as a test IP phone
2. Review the ISE Anomalous Endpoint Detection (AED) feature and explain why it is ineffective in this case

ISE profiler primer

The ISE profiler has 11 modules that ingest information from a variety of sources to build a database of endpoints and endpoint attributes. The primary key for this data structure is the MAC address of the endpoint. The ISE user interface provides an interface to view the endpoint database and inspect individual endpoints through the Context Visibility→Endpoints Menu.

Endpoint Database and Endpoint Attributes

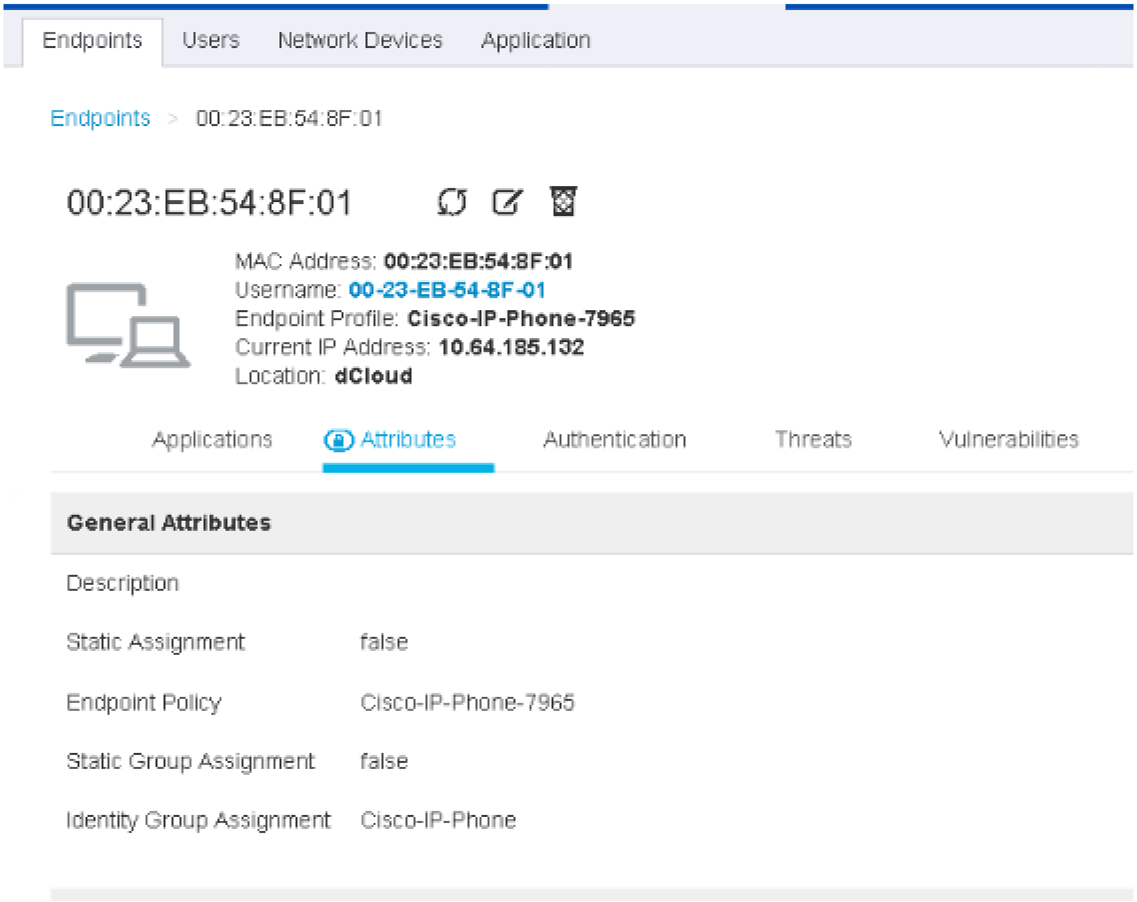


Endpoint database entry

MAC Address	Status	IPv4 Address	Username	Hostname
00:23:EB:54:8F:01		10.64.185.132	00-23-EB-54-8F-01	SEP0023EB548F01
00:E0:8F:00:6F:FE			00-E0-8F-00-6F-FE	

Figure 1 - endpoint database

By clicking on the MAC address of the endpoint, we can view detailed information from the database



Endpoints > 00:23:EB:54:8F:01

00:23:EB:54:8F:01

MAC Address: **00:23:EB:54:8F:01**
Username: **00-23-EB-54-8F-01**
Endpoint Profile: **Cisco-IP-Phone-7965**
Current IP Address: **10.64.185.132**
Location: **dCloud**

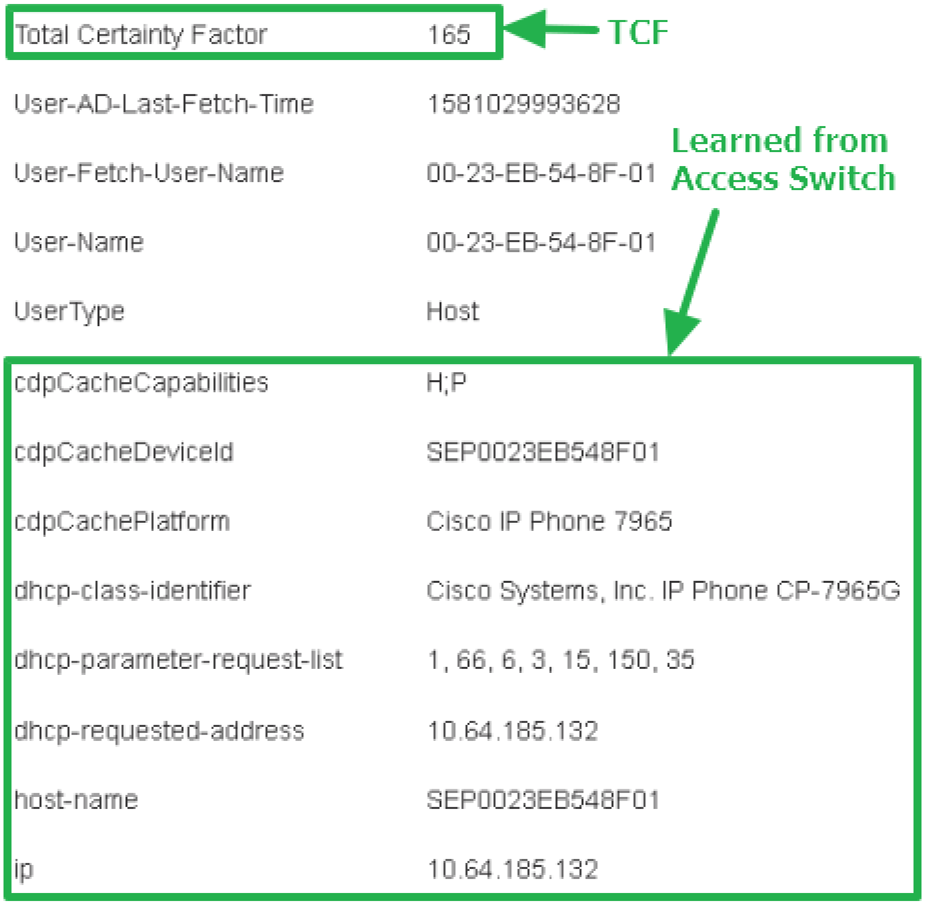
General Attributes

Description	
Static Assignment	false
Endpoint Policy	Cisco-IP-Phone-7965
Static Group Assignment	false
Identity Group Assignment	Cisco-IP-Phone

Figure 2 - Summary information about the endpoint

In figure 2, we can conclude that Media Access Control (MAC) Authentication bypass is being employed because the username is the same value as the MAC address. Ultimately this means we are not doing authentication. However, we can use the profiling information to authorize a specific level of access, which we will review later.

In the following image, we can see some attributes that ISE learned, as well as a value called Total Certainty Factor (TCF)



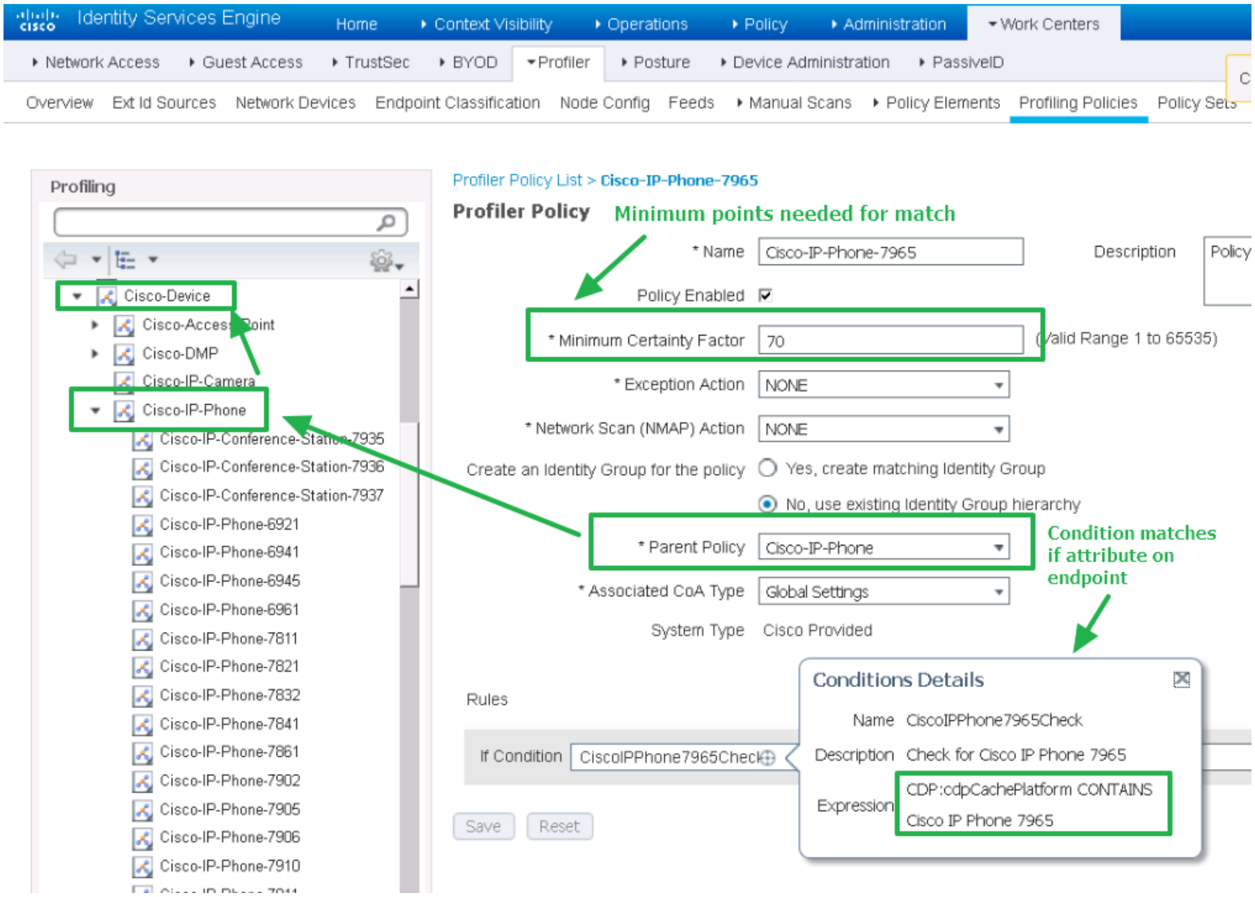
Total Certainty Factor	165
User-AD-Last-Fetch-Time	1581029993628
User-Fetch-User-Name	00-23-EB-54-8F-01
User-Name	00-23-EB-54-8F-01
UserType	Host
cdpCacheCapabilities	H;P
cdpCacheDeviceId	SEP0023EB548F01
cdpCachePlatform	Cisco IP Phone 7965
dhcp-class-identifier	Cisco Systems, Inc. IP Phone CP-7965G
dhcp-parameter-request-list	1, 66, 6, 3, 15, 150, 35
dhcp-requested-address	10.64.185.132
host-name	SEP0023EB548F01
ip	10.64.185.132

Figure 3 TCF and attributes learned from Device Sensor

So how is this information employed? ISE evaluates the attributes against a set of profiling policies, the policy with the highest TCF is assigned as the endpoint policy for that device. We then use that endpoint policy to decide how much access (if any) to authorize.

ISE profiling policies

Profiler polices assign point values to matching attributes. The highest Total Certainty Factor (TCF) score wins. The policies are arranged in a tree-like structure from coarse to finer-grained. The minimum score at each level of the tree has to be met before the child nodes will be evaluated.



Cisco-Device
Cisco-Access-Point
Cisco-DMP
Cisco-IP-Camera
Cisco-IP-Phone
Cisco-IP-Conference-Station-7935
Cisco-IP-Conference-Station-7936
Cisco-IP-Conference-Station-7937
Cisco-IP-Phone-6921
Cisco-IP-Phone-6941
Cisco-IP-Phone-6945
Cisco-IP-Phone-6961
Cisco-IP-Phone-7811
Cisco-IP-Phone-7821
Cisco-IP-Phone-7832
Cisco-IP-Phone-7841
Cisco-IP-Phone-7861
Cisco-IP-Phone-7902
Cisco-IP-Phone-7905
Cisco-IP-Phone-7906
Cisco-IP-Phone-7910

Profiler Policy List > Cisco-IP-Phone-7965

Profiler Policy

Minimum points needed for match

* Name: Cisco-IP-Phone-7965
Description: Policy
Policy Enabled ☑
* Minimum Certainty Factor: 70 (Valid Range 1 to 65535)
* Exception Action: NONE
* Network Scan (NMAP) Action: NONE
Create an Identity Group for the policy ○ Yes, create matching Identity Group
● No, use existing Identity Group hierarchy
* Parent Policy: Cisco-IP-Phone
* Associated CoA Type: Global Settings
System Type: Cisco Provided

Condition matches if attribute on endpoint

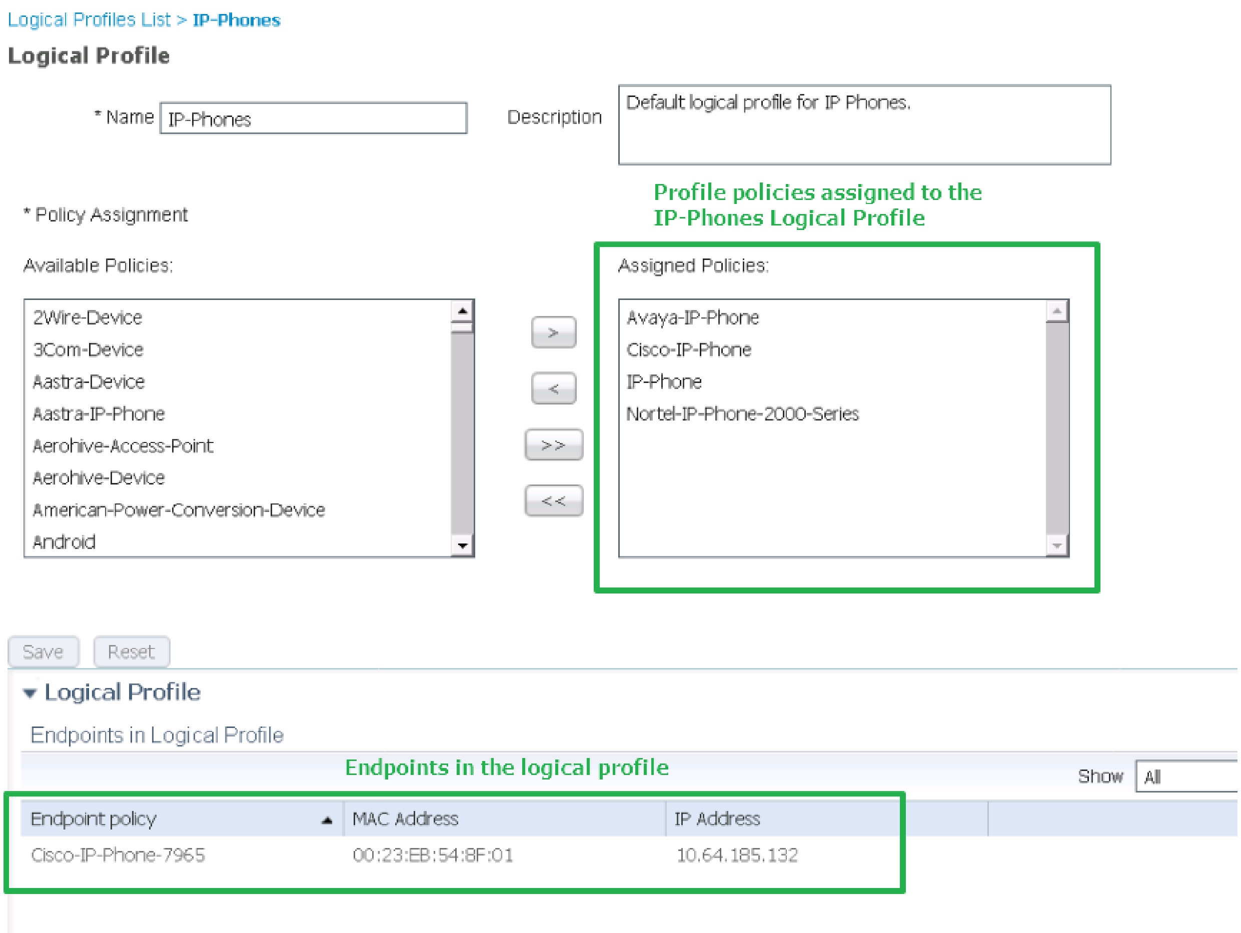
Rules
If Condition: CiscoIPPhone7965Check

Conditions Details
Name: CiscoIPPhone7965Check
Description: Check for Cisco IP Phone 7965
Expression: CDP:cdpCachePlatform CONTAINS Cisco IP Phone 7965

Figure 4 - Profiler policy for a 7965 IP phone

Logical polices

Logical policies are used to group like devices together where a collective policy decision would be made for them. It is the functional equivalent of putting users into groups for granting access to files and folders on a computer.



Logical Profiles List > **IP-Phones**

Logical Profile

* Name: IP-Phones

Description: Default logical profile for IP Phones.

Profile policies assigned to the IP-Phones Logical Profile

* Policy Assignment

Available Policies:

- 2Wire-Device
- 3Com-Device
- Aastra-Device
- Aastra-IP-Phone
- Aerohive-Access-Point
- Aerohive-Device
- American-Power-Conversion-Device
- Android

Assigned Policies:

- Avaya-IP-Phone
- Cisco-IP-Phone
- IP-Phone
- Nortel-IP-Phone-2000-Series

Logical Profile

Endpoints in Logical Profile

Endpoints in the logical profile

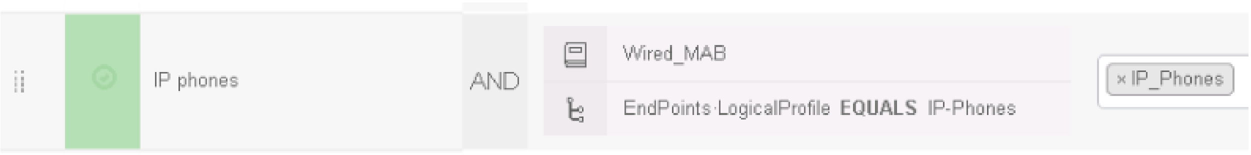
Show: All

Endpoint policy	MAC Address	IP Address
Cisco-IP-Phone-7965	00:23:EB:54:8F:01	10.64.185.132

Figure 5 - Logical profile

Policy Set Authorization (AuthZ) rule

Finally, we use the logical profile in an authorization rule, which then directs the network device to apply the authorization we have defined.



IP phones AND Wired_MAB

EndPoints·LogicalProfile **EQUALS** IP-Phones

× IP_Phones

Figure 6 - Authorization rule for IP Phones

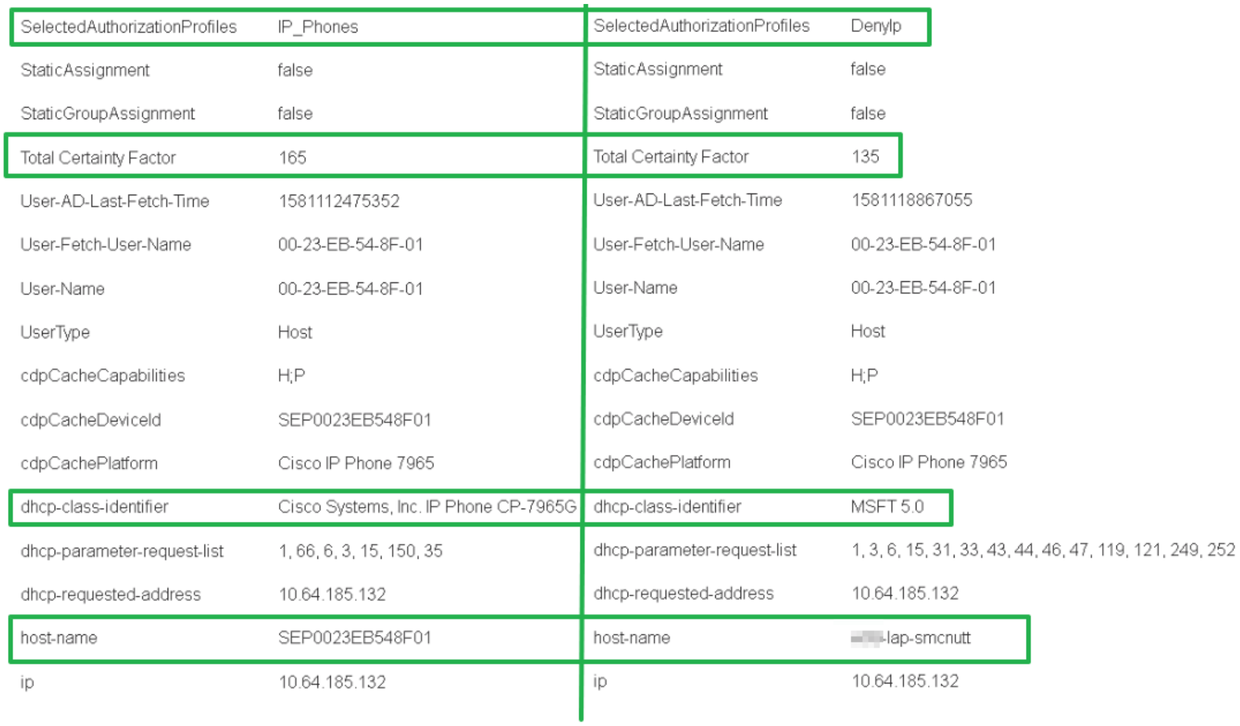
So how does mac address spoofing bypass this system? Now that we have set the stage, we can start to talk about that.

Effect of MAC address spoofing on the profiler

Now we will see what happens when we try a Windows and then a Linux Computer using the MAC address of our previously profiled phone

Windows laptop

For our first, we will connect a Windows 10 computer to the switch, with the same mac address as the test phone, and we will see what changes.



SelectedAuthorizationProfiles	IP_Phones	SelectedAuthorizationProfiles	DenyIp
StaticAssignment	false	StaticAssignment	false
StaticGroupAssignment	false	StaticGroupAssignment	false
Total Certainty Factor	165	Total Certainty Factor	135
User-AD-Last-Fetch-Time	1581112475352	User-AD-Last-Fetch-Time	1581118867055
User-Fetch-User-Name	00-23-EB-54-8F-01	User-Fetch-User-Name	00-23-EB-54-8F-01
User-Name	00-23-EB-54-8F-01	User-Name	00-23-EB-54-8F-01
UserType	Host	UserType	Host
cdpCacheCapabilities	H;P	cdpCacheCapabilities	H;P
cdpCacheDeviceId	SEP0023EB548F01	cdpCacheDeviceId	SEP0023EB548F01
cdpCachePlatform	Cisco IP Phone 7965	cdpCachePlatform	Cisco IP Phone 7965
dhcp-class-identifier	Cisco Systems, Inc. IP Phone CP-7965G	dhcp-class-identifier	MSFT 5.0
dhcp-parameter-request-list	1, 66, 6, 3, 15, 150, 35	dhcp-parameter-request-list	1, 3, 6, 15, 31, 33, 43, 44, 46, 47, 119, 121, 249, 252
dhcp-requested-address	10.64.185.132	dhcp-requested-address	10.64.185.132
host-name	SEP0023EB548F01	host-name	[redacted]-lap-smcnutt
ip	10.64.185.132	ip	10.64.185.132

Figure 7 – Windows 10 laptop

Reviewing figure 7, There are 4 items highlighted:

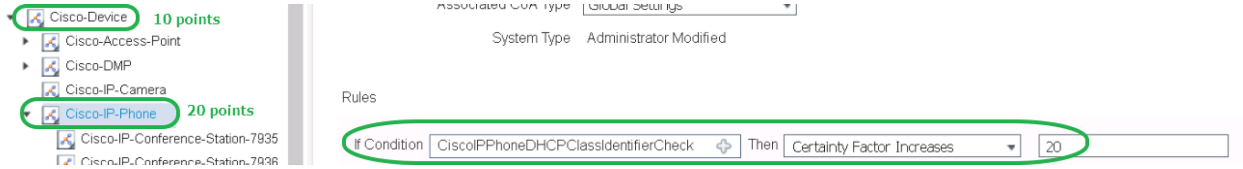
1. Authorization profile
2. Total Certainty Factor
3. dhcp-class-identifier
4. host-name

The reason why we have an authorization result of DenyIP is that I had configured Anomalous endpoint detection. The dhcp-class-identifier change triggered the Anomalous Endpoint flag on the endpoint to true. I then used this as a condition in a rule to return the DenyIp authorization result. We will dive into the details in the AED section.

There are two main takeaways here:

1. ISE was able to respond to the MAC address spoofing attempt and flagged the endpoint.
2. The attributes from when the phone was profiled are still present, even though we can be reasonably sure the laptop did not send them.

It is the second point that’s important to understand. The absence of a value being sent (that was sent prior) does not equal a change as far as the profiler is concerned. The TCF changed by -30 because the dhcp-class-identifier value is scored in two locations for 10 and 20 points, respectively.



Associated CoA Type Global Settings

System Type Administrator Modified

Rules

If Condition CiscoIPPhoneDHCPClassIdentifierCheck Then Certainty Factor Increases 20

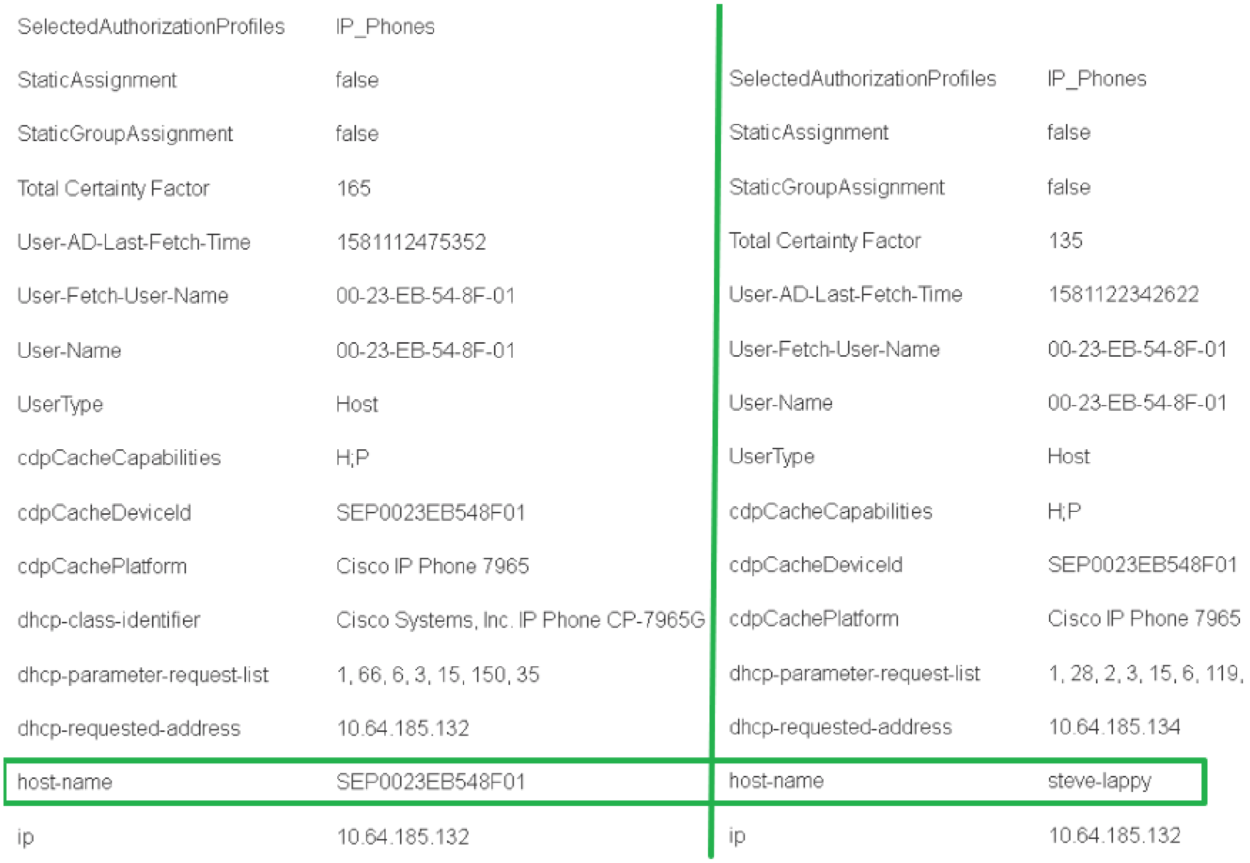
Figure 8 - how attribute matches are scored

It is essential to grasp that except for DHCP, when endpoint attributes accumulate, they remain until there is a change. Usually, this is not too much of a problem….

Once a device has had the AED flag set, the only way to clear the condition is to delete the endpoint. I am going to unplug the laptop, delete the endpoint, and next, we will try the Ubuntu Laptop.

Linux laptop

The endpoint was deleted, the phone plugged back in, and our endpoint has been recreated with the phone accurately identified. Now let us plug in the Linux laptop and take a look.



SelectedAuthorizationProfiles	IP_Phones
StaticAssignment	false
StaticGroupAssignment	false
Total Certainty Factor	165
User-AD-Last-Fetch-Time	1581112475352
User-Fetch-User-Name	00-23-EB-54-8F-01
User-Name	00-23-EB-54-8F-01
UserType	Host
cdpCacheCapabilities	H;P
cdpCacheDeviceId	SEP0023EB548F01
cdpCachePlatform	Cisco IP Phone 7965
dhcp-class-identifier	Cisco Systems, Inc. IP Phone CP-7965G
dhcp-parameter-request-list	1, 66, 6, 3, 15, 150, 35
dhcp-requested-address	10.64.185.132
host-name	SEP0023EB548F01
ip	10.64.185.132

SelectedAuthorizationProfiles	IP_Phones
StaticAssignment	false
StaticGroupAssignment	false
Total Certainty Factor	135
User-AD-Last-Fetch-Time	1581122342622
User-Fetch-User-Name	00-23-EB-54-8F-01
User-Name	00-23-EB-54-8F-01
UserType	Host
cdpCacheCapabilities	H;P
cdpCacheDeviceId	SEP0023EB548F01
cdpCachePlatform	Cisco IP Phone 7965
dhcp-parameter-request-list	1, 28, 2, 3, 15, 6, 119,
dhcp-requested-address	10.64.185.134
host-name	steve-lappy
ip	10.64.185.132

Figure 9 - Linux laptop

The only informational DHCP attribute Ubuntu sends in its DHCP discovery request is host-name. The result is that the laptop received the IP_phones Authorization Profile. Why didn't AED fire and block the endpoint?

Because we are not getting anything actionable, there is nothing to trigger ISE. If the residual attributes left from when the phone was plugged in were not persistent, this would trigger a reprofile and we would be able to do something. The reasons the attributes are cached are understandable, but it presents a difficulty here.

The main takeaways here are:

1. In default DHCP configuration, the Ubuntu laptop doesn't give up any useful information.
2. We just got p0wned.

Anomalous Endpoint Detection (AED)

We saw that AED worked in the case of a Windows machine but not in the case of the Ubuntu Linux machine. Let us a closer look at AED and how it works.

According to the Documentation (Cisco, 2018), AED can trigger on a change in one of three things.

1. NAS-Port-Type (i.e. wired or wireless)
2. DHCP-class-identifier
3. Endpoint policy change

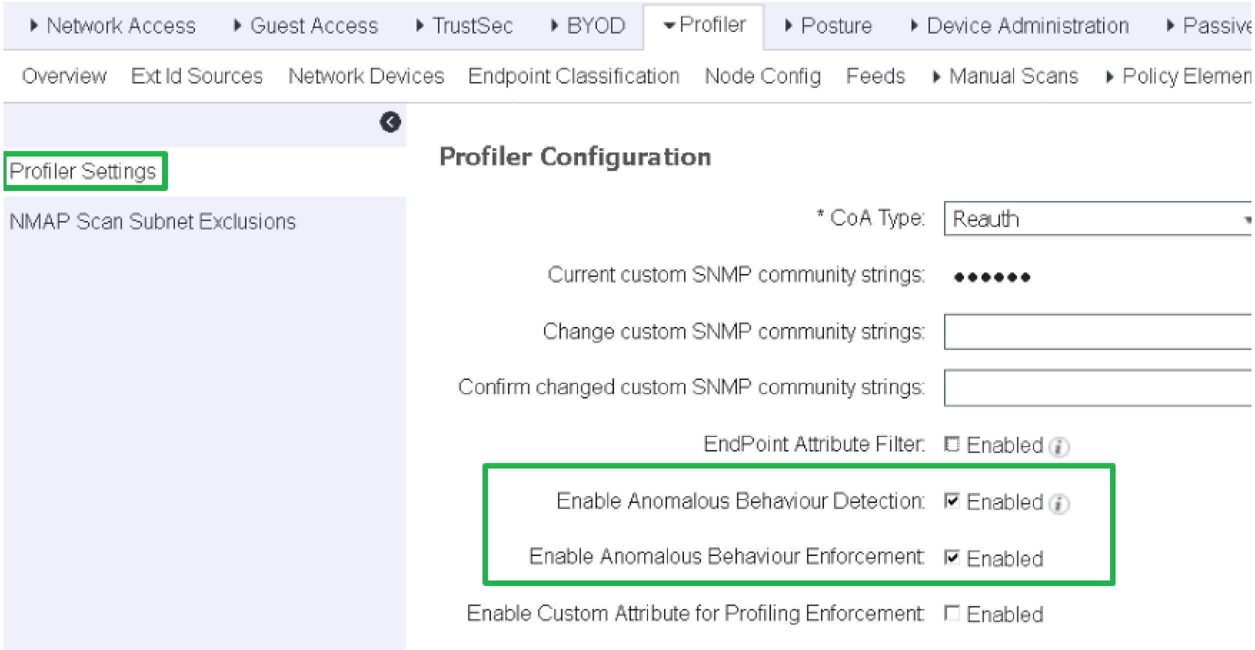
If any of these conditions occur with AED enabled, the endpoint will be flagged. It should be noted that not getting the class-id when it was received prior will not trigger AED. And this is why we got p0wned by Linux.

While we are here, let us walk through how to configure AED.

1. Enable it
2. Create an Authorization exception rule that matches on the AnomalousBehaviour endpoint flag
3. Apply an authorization profile that allows the device to receive an address (optional, I'll explain)

Enabling AED

In the profiler configuration settings, check off the AED boxes.



Profiler Configuration

* CoA Type: Reauth

Current custom SNMP community strings: ●●●●●●

Change custom SNMP community strings:

Confirm changed custom SNMP community strings:

EndPoint Attribute Filter: □ Enabled

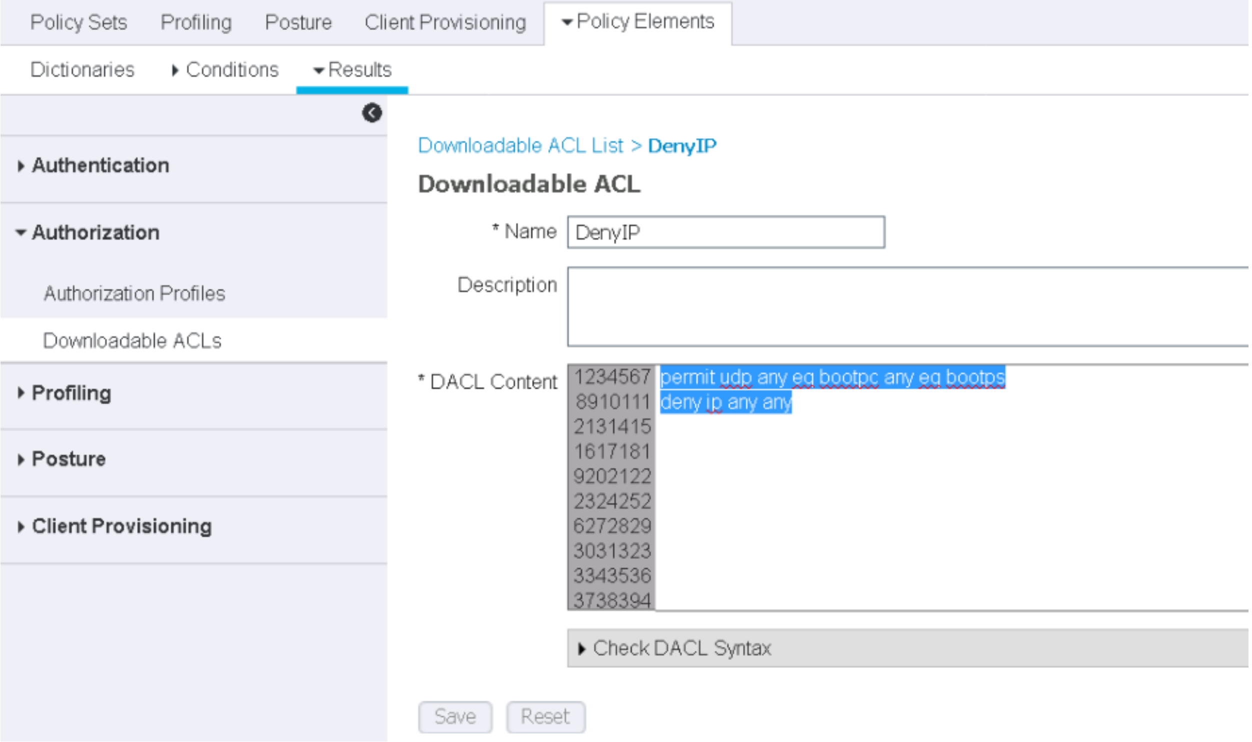
Enable Anomalous Behaviour Detection: ☑ Enabled

Enable Anomalous Behaviour Enforcement: ☑ Enabled

Enable Custom Attribute for Profiling Enforcement: □ Enabled

Figure 10- Enable AED

Define a DACL that only allows DHCP/Bootp



Downloadable ACL List > DenyIP

Downloadable ACL

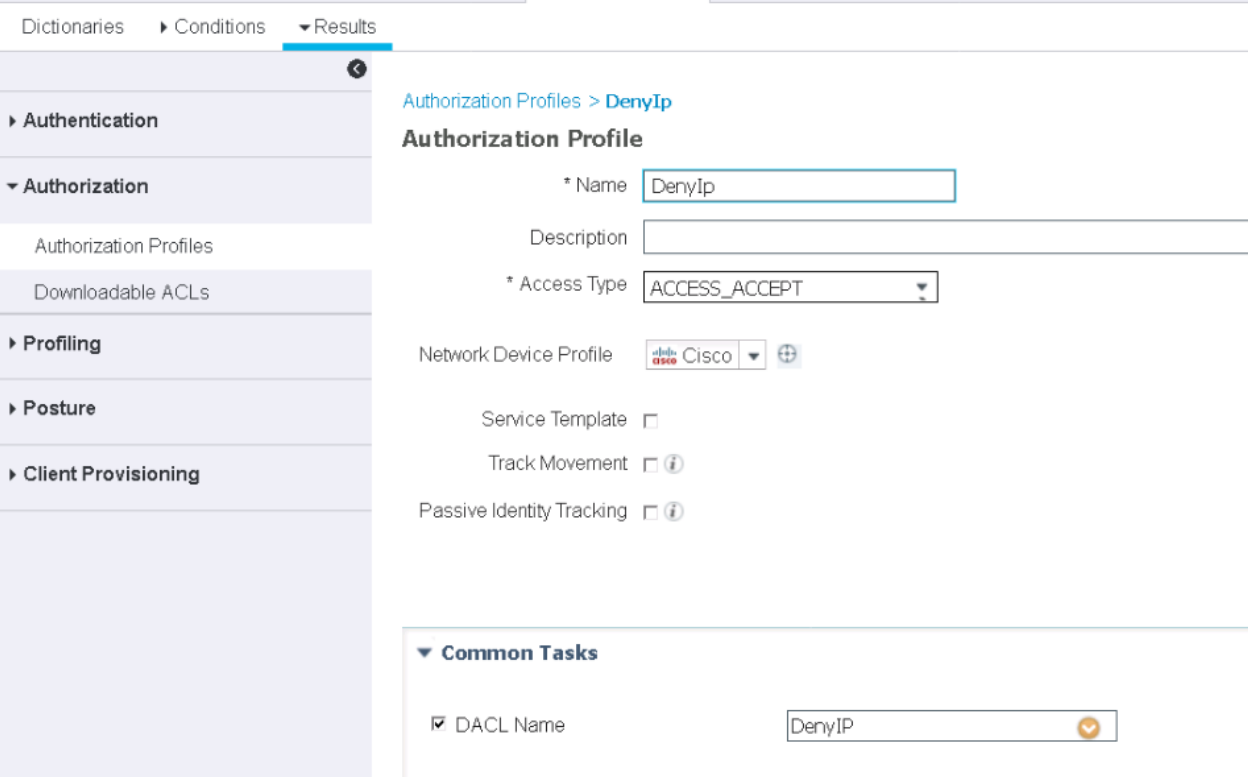
* Name: DenyIP

Description:

* DACL Content:

```
permit udp any eq bootpc any eq bootps
deny ip any any
```

Reference the DACL in an authorization profile:



Authorization Profiles > DenyIp

Authorization Profile

* Name: DenyIp

Description:

* Access Type: ACCESS_ACCEPT

Network Device Profile: Cisco

Service Template □

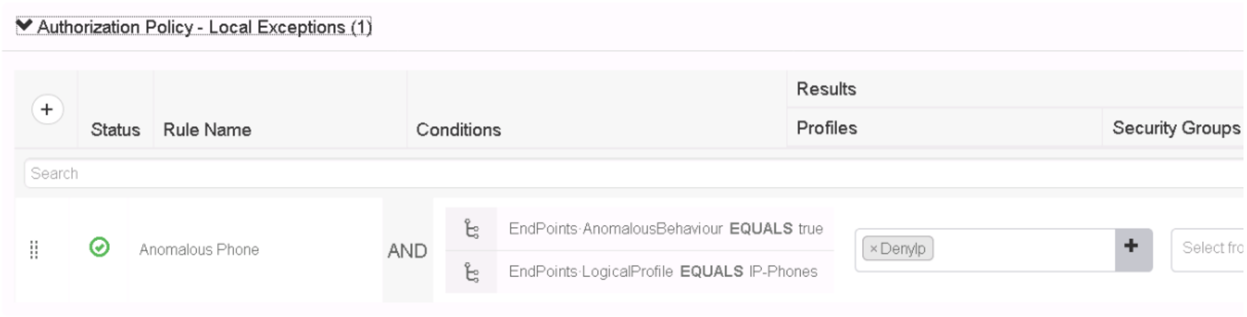
Track Movement □

Passive Identity Tracking □

Common Tasks

☑ DACL Name: DenyIP

In the policy set, create an Exception rule that conditions on AnomalousBehaviour Endpoint Attribute. In this case, I also conditioned on IP phones because I only want the scope to cover the specific assets of interest. Assign the AuthZ profile.



Authorization Policy - Local Exceptions (1)

+	Status	Rule Name	Conditions		Results	
					Profiles	Security Groups
Search						
⁞⁞	✓	Anomalous Phone	AND	EndPoints·AnomalousBehaviour EQUALS true EndPoints·LogicalProfile EQUALS IP-Phones	× DenyIp +	Select fro

Why the access-accept, and why give out an IP address? The simple answer is visibility. If we give the device an IP address, ISE can ingest the DHCP attributes and we will be able to tell from the ISE UI that a different device has been plugged in even though the endpoint was flagged by AED. If we were to prevent all access, we might not receive any new information until the endpoint was deleted and recreated.

Proposed Solution

Now we have discussed the limitations of AED and the inability to act with the Linux laptop because ISE is not receiving any actionable information.   Even if we rolled out 802.1x, there might still be devices that require using MAB, so we cannot sidestep the issue.

There is a straightforward solution: Implement first-hop security and require the client to send a class-identifier in its DHCP requests. If the class-identifier does not match, or there is not one sent, then the device will not obtain an IP Address.

There are three tools we will employ to accomplish this:

1. DHCP Snooping (which you have already enabled on your network, right?)
2. IP source guard (depends on DHCP snooping, prevents user-assigned static addresses)
3. DHCP policy on the DCHP Server

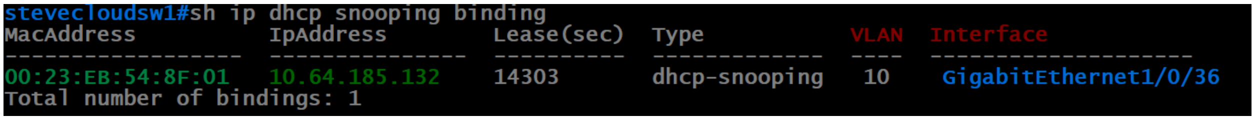
DHCP Snooping

DHCP snooping listens to DHCP traffic and creates a table of IP MAC and Interface bindings. Configuring DHCP snooping is very straightforward. However, consult the documentation and make to understand how to scope it to specific VLANS and trust the uplinks leading towards the DHCP server. (Cisco Security Configuration Guide, no.date.)

Sample configuration:

```
ip dhcp snooping vlan 10,100
no ip dhcp snooping information option
ip dhcp snooping
!
interface GigabitEthernet1/0/48
!
!-----------Trust the northbound link towards the DHCP server
ip dhcp snooping trust
!
```

Binding example:



```
stevecloudsw1#sh ip dhcp snooping binding
MacAddress          IpAddress        Lease(sec)  Type           VLAN  Interface
------------------  ---------------  ----------  -------------  ----  --------------------
00:23:EB:54:8F:01   10.64.185.132    14303       dhcp-snooping   10    GigabitEthernet1/0/36
Total number of bindings: 1
```

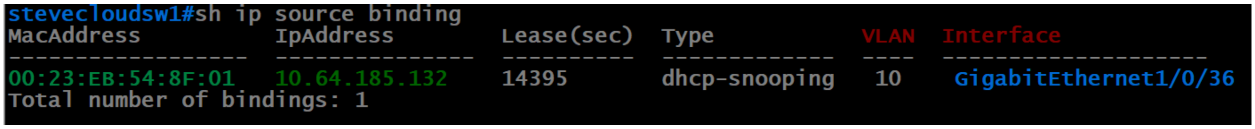
IP Source Guard

IP source guard will check the DHCP snooping binding table as well as the static binding table for a matching entry when IP packets are received on the switchport. If there no match, the traffic will be dropped. This prevents an attacker from statically applying an IP address.

Ip source guard is a one-line command:

```
interface GigabitEthernet1/0/36
!
ip verify source
```

Verification output:



```
stevecloudsw1#sh ip source binding
MacAddress          IpAddress        Lease(sec)  Type           VLAN  Interface
------------------  ---------------  ----------  -------------  ----  --------------------
00:23:EB:54:8F:01   10.64.185.132    14395       dhcp-snooping   10    GigabitEthernet1/0/36
Total number of bindings: 1
```

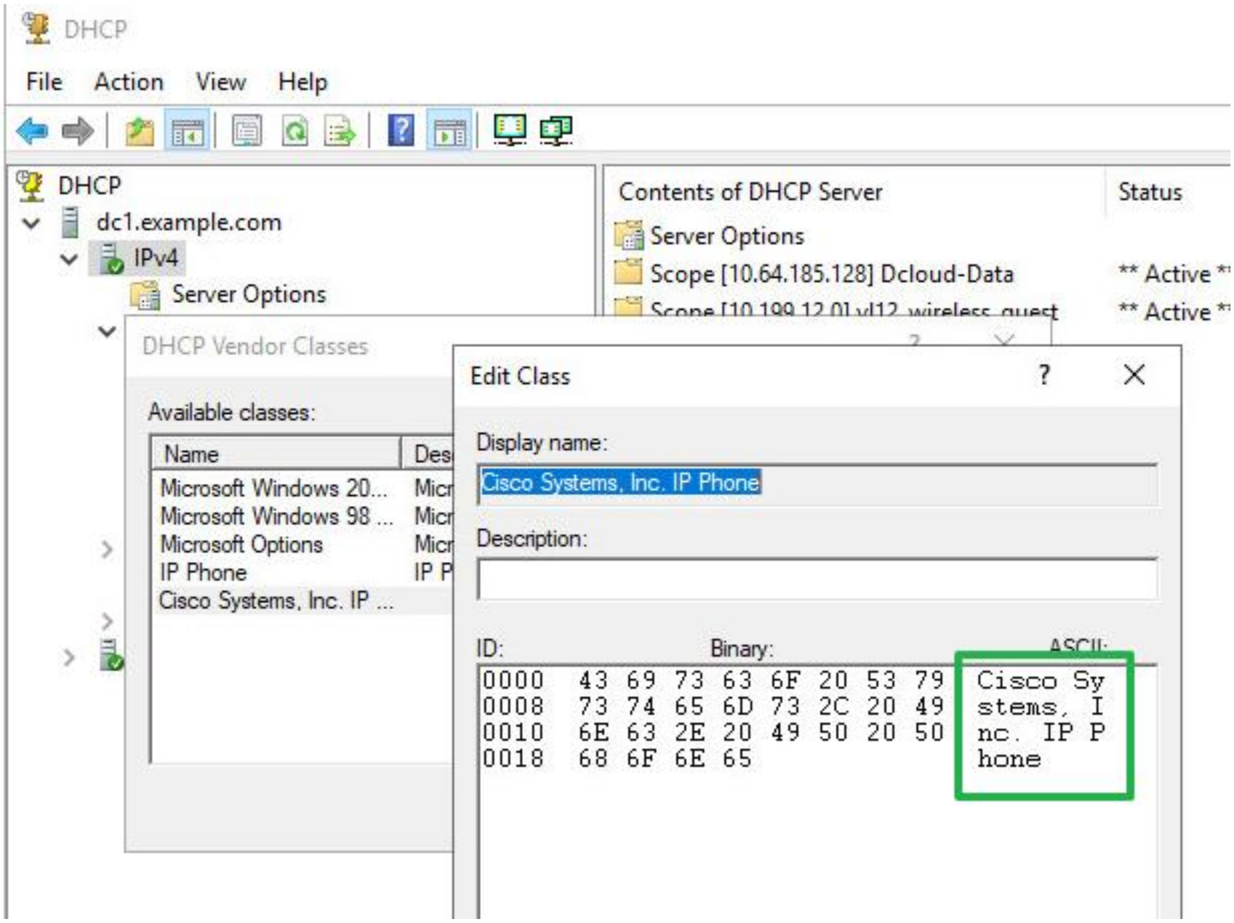
DHCP Policy

To close the loop, we will create a DHCP policy that will only offer addresses to devices that send the correct class-id in the DHCP Discover or Request packets. (Microsoft Docs, 2016). Creating the policy consists of the following steps:

1. Create a vendor Class
2. Create a policy that references the vendor class

Creating the Vendor Class.

In the DHCP management tool, right-click on the IP Address family icon and select define vendor classes. When defining the class, the beginning or the end of the string can be used for wildcard matches, but the characters must be exact. An easy way to get this is to copy it from the endpoint attribute in ISE.

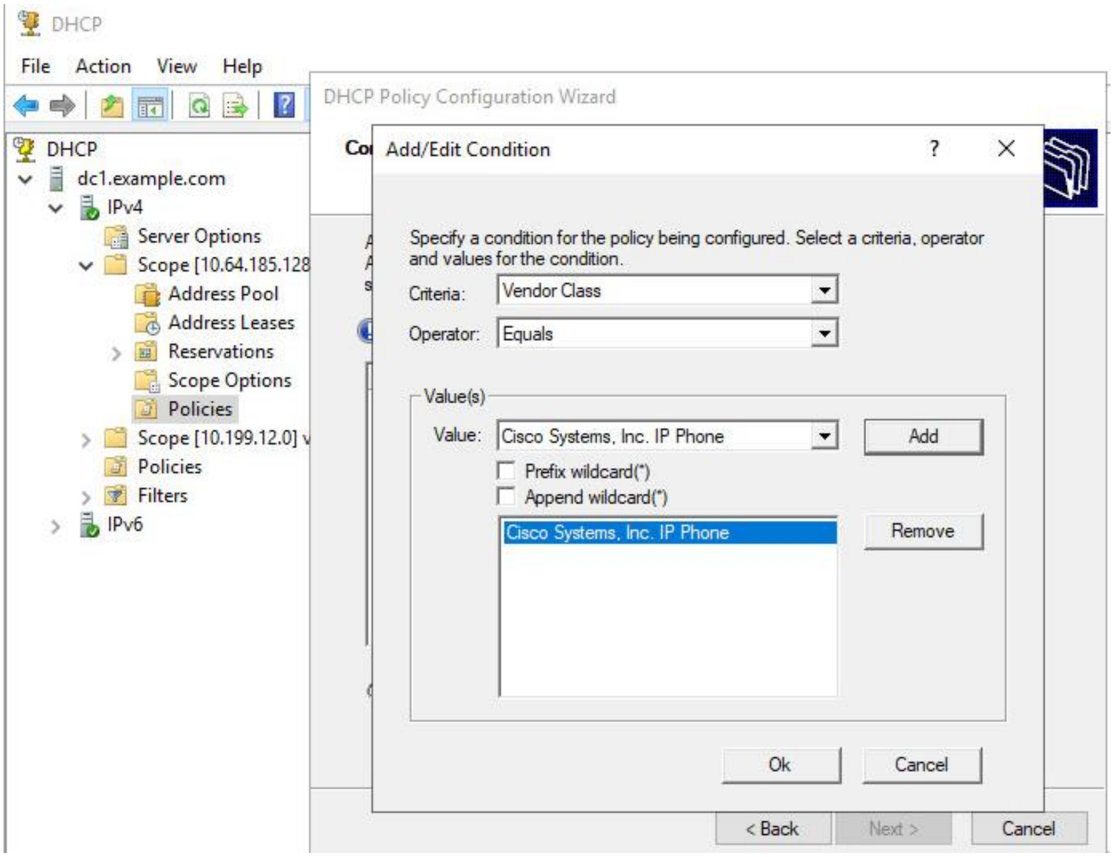


Display name: Cisco Systems, Inc. IP Phone

```
0000   43 69 73 63 6F 20 53 79   Cisco Sy
0008   73 74 65 6D 73 2C 20 49   stems, I
0010   6E 63 2E 20 49 50 20 50   nc. IP P
0018   68 6F 6E 65               hone
```

Creating the policy that references the vendor class

1. In the policies folder, right-click and select new policy. This will launch the new policy wizard.
2. On the configure conditions window, click add to add a condition.
3. Select vendor class, operator, the vendor class defined earlier, then wildcard if desired. Select add, and click ok.



Specify a condition for the policy being configured. Select a criteria, operator and values for the condition.

Criteria: Vendor Class

Operator: Equals

Value(s)

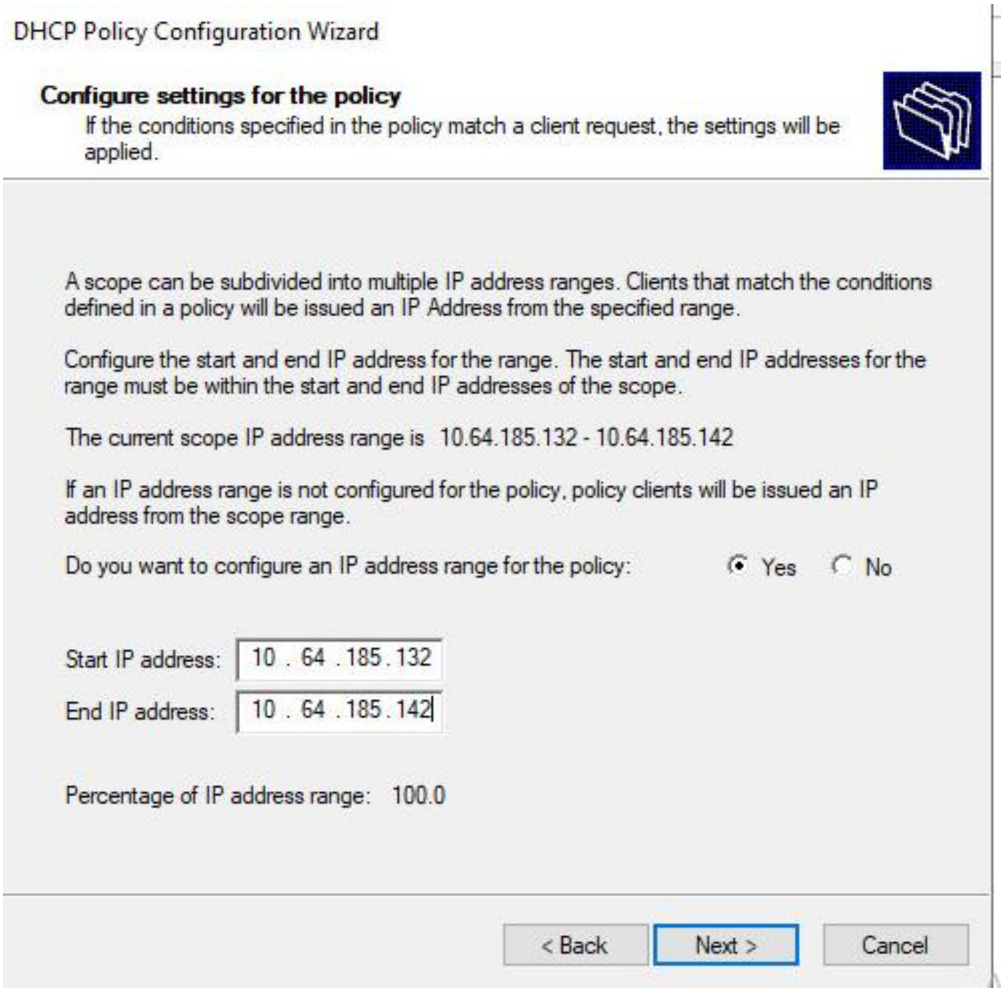
Value: Cisco Systems, Inc. IP Phone Add

□ Prefix wildcard(*)

□ Append wildcard(*)

Cisco Systems, Inc. IP Phone Remove

4. Use all of the addresses in the scope. NOTE: once done, the policy will reserve 100% of the addresses even if disabled. The policy must be deleted to release the reservation.



DHCP Policy Configuration Wizard

Configure settings for the policy

If the conditions specified in the policy match a client request, the settings will be applied.

A scope can be subdivided into multiple IP address ranges. Clients that match the conditions defined in a policy will be issued an IP Address from the specified range.

Configure the start and end IP address for the range. The start and end IP addresses for the range must be within the start and end IP addresses of the scope.

The current scope IP address range is 10.64.185.132 - 10.64.185.142

If an IP address range is not configured for the policy, policy clients will be issued an IP address from the scope range.

Do you want to configure an IP address range for the policy: ● Yes ○ No

Start IP address: 10 . 64 . 185 . 132

End IP address: 10 . 64 . 185 . 142

Percentage of IP address range: 100.0

5. Click through the rest of the wizard and select finish.

Conclusion

Strong authentication using 802.1x will protect against identity spoofing, but it is a significant undertaking for existing networks, and it takes time to roll out. Additionally, there may be devices that cannot perform 802.1x authentication, but they require network access. Therefore, we have to deploy a defense-in-depth strategy to secure the access network when device authentication is not possible.

Recommendations

- Use 802.1x with certificate authentication whenever possible.
- If you have a NAC such as ISE, employ least privilege authorization to mitigate impact.
- Deploy First Hop Security (FHS), primarily DHCP snooping and IP Source Guard.
- DHCP servers are a good control point to perform policy enforcement. Take advantage.
- Take operations security practices seriously and follow the cycle.

Bibliography

Cisco. (2018, May). *Configure Anomalous Endpoint Detection and Enforcement on ISE 2.2*. Retrieved from Cisco.com: https://www.cisco.com/c/en/us/support/docs/security/identity-services-engine-22/200973-configure-anomalous-endpoint-detection-a.html

Cisco Security Configuration Guide. (no.date.). *Security Configuration Guide, Cisco IOS XE Fuji 16.9.x, Configuring DHCP*. Retrieved from Cisco.com: https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst9300/software/release/16-9/configuration_guide/sec/b_169_sec_9300_cg/configuring_dhcp.html?bookSearch=true&arrowback=true

Microsoft Docs. (2016, August 31). *Scenario: Secure a subnet to a specific set of clients*. Retrieved from https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/dn425039(v%3Dws.11)