

# Kerberoasting

When a TGS is requested, an event log with ID 4769 is generated. However, AD also generates the same event ID whenever a user attempts to connect to a service, which means that the volume of this event is gigantic, and relying on it alone is virtually impossible to use as a detection method. If we happen to be in an environment where all applications support AES and only AES tickets are generated, then it would be an excellent indicator to alert on event ID 4769 . If the ticket options is set for RC4 , that is, if RC4 tickets are generated in the AD environment (which is not the default configuration), then we should alert and follow up on it. Here is what was logged when we requested the ticket to perform this attack:

Event 4769, Microsoft Windows security auditing.

General Details

A Kerberos service ticket was requested.

Account Information:

Account Name:	bob@EAGLE.LOCAL
Account Domain:	EAGLE.LOCAL
Logon GUID:	{82b8d5e6-2a99-e568-44f6-d78608bb86e5}

Service Information:

Service Name:	Administrator
Service ID:	EAGLE\Administrator

Network Information:

Client Address:	::ffff:172.16.18.25
Client Port:	60491

Additional Information:

Ticket Options:	0x40800000
Ticket Encryption Type:	0x17
Failure Code:	0x0
Transited Services:	-

This event is generated every time access is requested to a resource such as a computer or a Windows service. The service name indicates the resource to which access was requested.

This event can be correlated with Windows logon events by comparing the Logon GUID fields in each event. The logon event occurs on the machine that was accessed, which is often a different machine than the domain controller which issued the service ticket.

Ticket options, encryption types, and failure codes are defined in RFC 4120.

Relevant fields highlighted and annotated:

- Account Name: bob@EAGLE.LOCAL (Represents who requests the ticket)
- Service Name: Administrator (Represents whom the requested TGS is for)
- Client Address: ::ffff:172.16.18.25 (Represents from which machine the request originated)
- Ticket Encryption Type: 0x17 (Represents the encryption of the ticket, in this case RC4)

Security Number of events: 170,911 (!) New events available											
Keywords	Date and Time	Source	Event ID	Task Category							
Audit Success	12/12/2022 9:11:08 PM	Microsoft Windows secu...	4769	Kerberos Service Ticket ...							
Audit Success	12/12/2022 9:11:08 PM	Microsoft Windows secu...	4769	Kerberos Service Ticket ...							
Event 4769, Microsoft Windows security auditing.											
General	Details	Matching volume of events by source IP address or requester name within a short time									
A Kerberos service ticket was requested.											
Account Information:											
Account Name:	bob@EAGLE.LOCAL										
Account Domain:	EAGLE.LOCAL										
Logon GUID:	{82b8d5e6-2a99-e568-44f6-d78608bb86e5}										
Service Information:											
Service Name:	Administrator										
Service ID:	EAGLE\Administrator										
Network Information:											
Client Address:	::ffff:172.16.18.25										
Client Port:	60491										
Additional Information:											
Ticket Options:	0x40800000										
Ticket Encryption Type:	0x17										
Failure Code:	0x0										
Transited Services:	-										
This event is generated every time access is requested to a resource such as a computer or a Windows service. The service name indicates the resource to which access was requested.											
This event can be correlated with Windows logon events by comparing the Logon GUID fields in each event. The logon event occurs on the machine that was accessed, which is often a different machine than the domain controller which issued the service ticket.											

## AS-REProasting

When we executed Rubeus, an Event with ID 4768 was generated, signaling that a Kerberos Authentication ticket was generated:

Audit Success	12/6/2022 7:27:58 PM	Microsoft Windows secu...	4768	Kerberos Authentication...			
Event 4768, Microsoft Windows security auditing.							
General	Details						
A Kerberos authentication ticket (TGT) was requested.							
Account Information:							
Account Name:	anni						
Supplied Realm Name:	eagle.local						
User ID:	EAGLE\anni						
Service Information:							
Service Name:	krbtgt						
Service ID:	EAGLE\krbtgt						
Network Information:							
Client Address:	::ffff:172.16.18.25						
Client Port:	59266						
Additional Information:							
Ticket Options:	0x40800010						
Result Code:	0x0						
Ticket Encryption Type:	0x17						

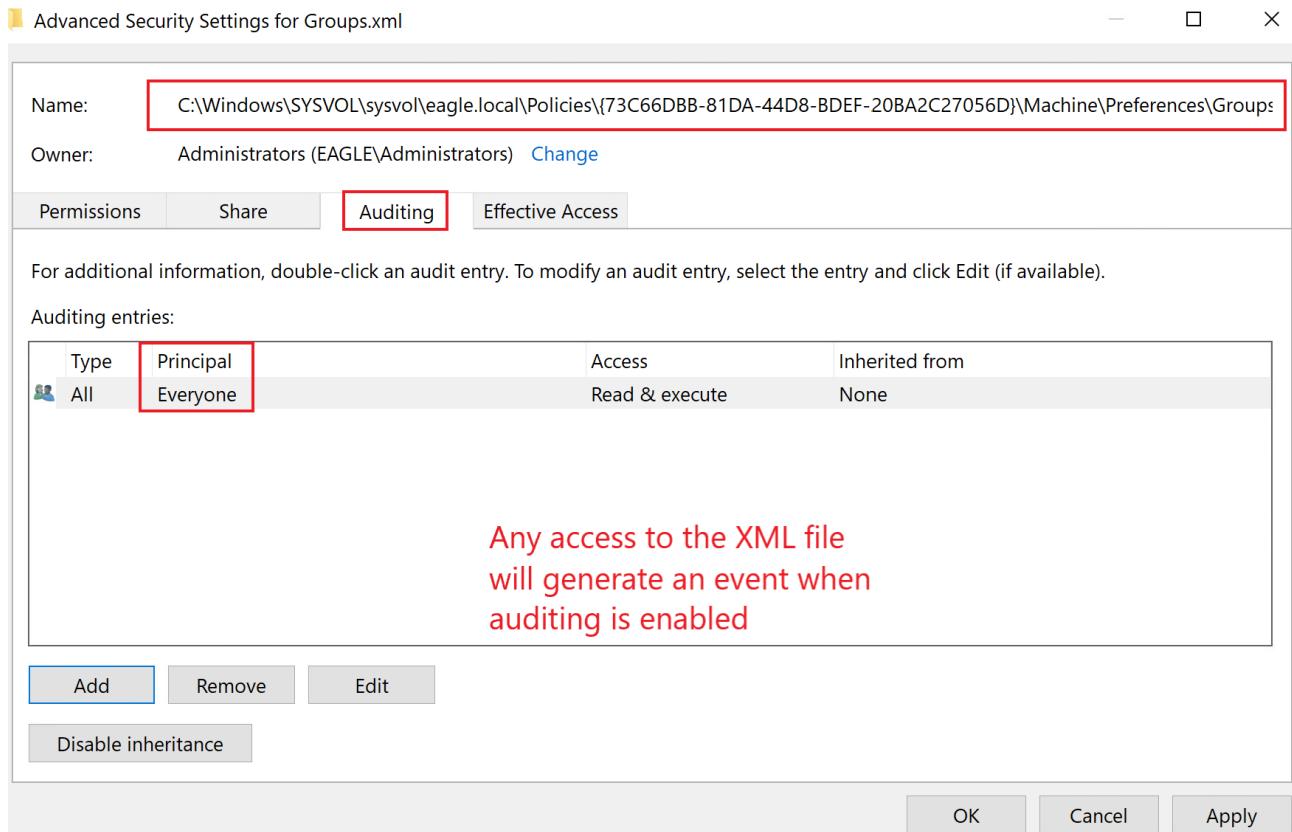
The caveat is that AD generates this event for every user that authenticates with Kerberos to any device; therefore, the presence of this event is very abundant. However, it is possible to know where the user authenticated from, which we can then use to correlate known good logins against potential malicious hash extractions. It may be hard to inspect specific IP addresses, especially if a user moves around office locations. However, it is possible to scrutinize the particular VLAN and alert on anything outside it.

---

## GPP Passwords

There are two detection techniques for this attack:

- Accessing the XML file containing the credentials should be a red flag if we are auditing file access; this is more realistic (due to volume otherwise) regarding detection if it is a dummy XML file, not associated with any GPO. In this case, there will be no reason for anyone to touch this file, and any attempt is likely suspicious. As demonstrated by `Get-GPPPasswords`, it parses all of the XML files in the Policies folder. For auditing, we can generate an event whenever a user reads the file:



Once auditing is enabled, any access to the file will generate an Event with the ID 4663 :

Event 4663, Microsoft Windows security auditing.

General Details

An attempt was made to access an object.

Subject:

Security ID: EAGLE\bob  
Account Name: bob  
Account Domain: EAGLE  
Logon ID: 0x6B7AE81

Object:

Object Server: Security  
Object Type: File  
Object Name: C:\Windows\SYSVOL\domain\Policies\{73C66DBB-81DA-44D8-BDEF-20BA2C27056D}\Machine  
\\Preferences\Groups\Groups.xml  
Handle ID: 0x934  
Resource Attributes: S:AI

Bob accessing the  
XML file with GPP  
credentials

Process Information:

Process ID: 0x4  
Process Name:

Logon attempts (failed or successful, depending on whether the password is up to date) of the user whose credentials are exposed is another way of detecting the abuse of this attack; this should generate one of the events 4624 (successful logon), 4625 (failed logon), or 4768 (TGT requested). A successful logon with the account from our attack scenario would generate the following event on the Domain Controller:

Event 4624, Microsoft Windows security auditing.

General Details

Logon Information:

Logon Type: 3  
Restricted Admin Mode: -  
Virtual Account: No  
Elevated Token: Yes

Impersonation Level: Impersonation

New Logon:

Security ID: EAGLE\svc-iis  
Account Name: svc-iis  
Account Domain: EAGLE.LOCAL  
Logon ID: 0x6BD373B  
Linked Logon ID: 0x0  
Network Account Name: -  
Network Account Domain: -  
Logon GUID: {e7da2965-e718-7a58-c857-92a787f1e23d}

Successful logon for svc-iis from a specific IP address - the IP can be correlated to discover if the event is legit or suspicious according to normal behavior in the environment

Process Information:

Process ID: 0x0  
Process Name: -

Network Information:

Workstation Name: -  
Source Network Address: 172.16.18.25  
Source Port: 60637

In the case of a service account, we may correlate logon attempts with the device from which the authentication attempt originates, as this should be easy to detect, assuming we know where certain accounts are used (primarily if the logon originated from a workstation, which is abnormal behavior for a service account).

## GPO Permissions/GPO Files

Fortunately, it is straightforward to detect when a GPO is modified. If Directory Service Changes auditing is enabled, then the event ID 5136 will be generated:

Security Number of events: 93,908 (!) New events available									
Keywords	Date and Time	Source	Event ID	Task Category					
Audit Success	12/8/2022 10:53:36 PM	Microsoft Windows security	5136	Directory Service Changes					
	12/8/2022 10:53:35 PM	Microsoft Windows security	4624	Directory Service Changes					
Event 5136, Microsoft Windows security auditing.									
General	Details								
A directory service object was modified.									
Subject:									
Security ID:	EAGLE\Administrator								
Account Name:	Administrator								
Account Domain:	EAGLE								
Logon ID:	0x347638								
User performing the modification action									
Directory Service:									
Name:	eagle.local								
Type:	Active Directory Domain Services								
Object:									
DN:	CN={31B2F340-016D-11D2-945F-00C04FB984F9},CN=POLICIES,CN=SYSTEM,DC=EAGLE,DC=LOCAL								
GUID:	CN={31B2F340-016D-11D2-945F-00C04FB984F9},CN=Policies,CN=System,DC=eagle,DC=local								
Class:	groupPolicyContainer								
Attribute:									
LDAP Display Name:	versionNumber								
Syntax (OID):	2.5.5.9								
Value:	9								
Shows the GUID value of the GPO modified									
Operation:									
Type:	Value Deleted								
Correlation ID:	{ba41cec4-2fa1-4109-b142-a74ef06ad569}								
Application Correlation ID:	-								
Log Name:	Security								
Source:	Microsoft Windows security	Logged:	12/8/2022 10:53:36 PM						
Event ID:	5136	Task Category:	Directory Service Changes						
Level:	Information	Keywords:	Audit Success						
User:	N/A	Computer:	DC1.eagle.local						
OpCode:	Info								
More Information:	<a href="#">Event Log Online Help</a>								

From a defensive point of view, if a user who is not expected to have the right to modify a GPO suddenly appears here, then a red flag should be raised.

## Credentials in Shares

Understanding and analyzing users' behavior is the best detection technique for abusing discovered credentials in shares. Suppose we know the time and location of users' login via data analysis. In that case, it will be effortless to alert on seemingly suspicious behaviors—for example, the discovered account 'Administrator' in the attack described above. If we were a mature organization that used Privileged Access Workstation, we would be alert to privileged users not authenticating from those machines. These would be alerts on event IDs 4624 / 4625 (failed and successful logon) and 4768 (Kerberos TGT requested).

Below is an example of a successful logon with event ID 4624 for the Administrator account:

Event 4624, Microsoft Windows security auditing.

General Details

Logon Information:

Logon Type: 3  
Restricted Admin Mode: -  
Virtual Account: No  
Elevated Token: Yes

Impersonation Level: Impersonation

New Logon:

Security ID: EAGLE\Administrator  
Account Name: Administrator  
Account Domain: EAGLE  
Logon ID: 0x31BA63  
Linked Logon ID: 0x0  
Network Account Name: -  
Network Account Domain: -  
Logon G UID: {00000000-0000-0000-0000-000000000000}

Correlate User with  
Source of  
authentication for  
abnormal activity

Process Information:

Process ID: 0x0  
Process Name: -

Network Information:

Workstation Name: -  
Source Network Address: 172.16.18.20  
Source Port: 48710

Similarly, if Kerberos were used for authentication, event ID 4768 would be generated:

Event Properties - Event 4768, Microsoft Windows security auditing.

General Details

A Kerberos authentication ticket (TGT) was requested.

Account Information:

Account Name:	Administrator
Supplied Realm Name:	eagle
User ID:	EAGLE\Administrator

Service Information:

Service Name:	krbtgt
Service ID:	EAGLE\krbtgt

Network Information:

Client Address:	::ffff:172.16.18.25
Client Port:	60755

Additional Information:

Ticket Options:	0x40810010
Result Code:	0x0
Ticket Encryption Type:	0x12
Pre-Authentication Type:	2

Certificate Information:

Certificate Issuer Name:	
Certificate Serial Number:	
Certificate Thumbprint:	

Certificate information is only provided if a certificate was used for pre-authentication.

Pre-authentication types, ticket options, encryption types and result codes are defined in RFC 4120.

Correlate User with Source of authentication for abnormal activity

Another detection technique is discovering the one-to-many connections, for example, when Invoke-ShareFinder scans every domain device to obtain a list of its network shares. It would be abnormal for a workstation to connect to 100s or even 1000s of other devices simultaneously.

## Credentials in Object Properties

Baselining users' behavior is the best technique for detecting abuse of exposed credentials in properties of objects. Although this can be tricky for regular user accounts, triggering an alert for

administrators/service accounts whose behavior can be understood and baselined is easier. Automated tools that monitor user behavior have shown increased success in detecting abnormal logons. In the example above, assuming that the provided credentials are up to date, we would expect events with event ID 4624 / 4625 (failed and successful logon) and 4768 (Kerberos TGT requested). Below is an example of event ID 4768 :

Event 4768, Microsoft Windows security auditing.

General Details

A Kerberos authentication ticket (TGT) was requested.

Account Information:

Account Name:	bonni
Supplied Realm Name:	eagle
User ID:	EAGLE\bonni

Service Information:

Service Name:	krbtgt
Service ID:	EAGLE\krbtgt

Network Information:

Client Address:	::ffff:172.16.18.25
Client Port:	60861

Additional Information:

Ticket Options:	0x40810010
Result Code:	0x0
Ticket Encryption Type:	0x12
Pre-Authentication Type:	2

Certificate Information:

Certificate Issuer Name:	
Certificate Serial Number:	
Certificate Thumbprint:	

Certificate information is only provided if a certificate was used for pre-authentication.

Unfortunately, the event ID 4738 generated when a user object is modified does not show the specific property that was altered, nor does it provide the new values of properties. Therefore, we cannot use this event to detect if administrators add credentials to the properties of objects.

## DCSync

Detecting DCSync is easy because each Domain Controller replication generates an event with the ID 4662 . We can pick up abnormal requests immediately by monitoring for this event ID and checking whether the initiator account is a Domain Controller. Here's the event generated

from earlier when we ran `Mimikatz`; it serves as a flag that a user account is performing this replication attempt:

Event 4662, Microsoft Windows security auditing.

General Details

An operation was performed on an object.

Subject :

Security ID:	EAGLE\rocky
Account Name:	rocky
Account Domain:	EAGLE
Logon ID:	0x1EB0C4C

Account name is not a Domain Controller

Object:

Object Server:	DS
Object Type:	domainDNS
Object Name:	DC=eagle,DC=local
Handle ID:	0x0

Operation:

Operation Type:	Object Access
Accesses:	Control Access
Access Mask:	0x100
Properties:	Control Access {1131f6ad-9c07-11d1-f79f-00c04fc2dcd2} domainDNS

Additional Information:

Parameter 1:	-
Parameter 2:	

Since replications occur constantly, we can avoid false positives by ensuring the followings:

- Either the property `1131f6aa-9c07-11d1-f79f-00c04fc2dcd2` or `1131f6ad-9c07-11d1-f79f-00c04fc2dcd2` is [present in the event](#).
- Whitelisting systems/accounts with a (valid) business reason for replicating, such as Azure AD Connect (this service constantly replicates Domain Controllers and sends the obtained password hashes to Azure AD).

# Golden Ticket

Correlating users' behavior is the best technique to detect abuse of forged tickets. Suppose we know the location and time a user regularly uses to log in. In that case, it will be easy to alert on other (suspicious) behaviors—for example, consider the account 'Administrator' in the attack described above. If a mature organization uses Privileged Access Workstations ( PAWs ), they should be alert to any privileged users not authenticating from those machines, proactively monitoring events with the ID 4624 and 4625 (successful and failed logon).

Domain Controllers will not log events when a threat agent forges a **Golden Ticket** from a compromised machine. However, when attempting to access another system(s), we will see events for successful logon originating from the compromised machine:

Event Properties - Event 4624, Microsoft Windows security auditing.

General	Details
An account was successfully logged on.	
<b>Subject:</b>	
Security ID:	NULL SID
Account Name:	-
Account Domain:	-
Logon ID:	0x0
<b>Logon Information:</b>	
Logon Type:	3
Restricted Admin Mode:	-
Virtual Account:	No
Elevated Token:	Yes
Impersonation Level:	Delegation
<b>New Logon:</b>	
Security ID:	EAGLE\Administrator
Account Name:	Administrator
Account Domain:	eagle.local
Logon ID:	0x1D4181
Linked Logon ID:	0x0
Network Account Name:	-
Network Account Domain:	-
Logon GUID:	{76f46441-2072-b710-591b-1ae0adc7a0c0}
<b>Process Information:</b>	
Process ID:	0x0
Process Name:	-
<b>Network Information:</b>	
Workstation Name:	-
Source Network Address:	172.16.18.25
Source Port:	56211

**Logon event generated by Golden Ticket appears normal.**

**Correlate to detect abnormal behavior**

Another detection point could be a TGS service requested for a user without a previous TGT. However, this can be a tedious task due to the sheer volume of tickets (and many other factors). If we go back to the attack scenario, by running `dir \\dc1\c$` at the end, we generated two TGS tickets on the Domain Controller:

Event 4769, Microsoft Windows security auditing.

General Details

A Kerberos service ticket was requested.

Account Information:

Account Name:	Administrator@eagle.local
Account Domain:	eagle.local
Logon GUID:	{3c6ed6ab-5fa8-6970-42fe-302018cc30a0}

Service Information:

Service Name:	DC1\$
Service ID:	EAGLE\DC1\$

Network Information:

Client Address:	::ffff:172.16.18.25
Client Port:	56212

Additional Information:

Ticket Options:	0x40810000
Ticket Encryption Type:	0x12
Failure Code:	0x0
Transited Services:	-

Correlate to detect abnormal behavior

Event 4769, Microsoft Windows security auditing.

General Details

A Kerberos service ticket was requested.

Account Information:

Account Name:	Administrator@eagle.local
Account Domain:	eagle.local
Logon GUID:	{3c6ed6ab-5fa8-6970-42fe-302018cc30a0}

Service Information:

Service Name:	krbtgt
Service ID:	EAGLE\krbtgt

Network Information:

Client Address:	::ffff:172.16.18.25
Client Port:	56213

Additional Information:

Ticket Options:	0x60810010
Ticket Encryption Type:	0x12
Failure Code:	0x0
Transited Services:	-

Correlate to detect abnormal behavior

The only difference between the tickets is the service. However, they are ordinary compared to the same events not associated with the `Golden Ticket`.

If `SID filtering` is enabled, we will get alerts with the event ID `4675` during cross-domain escalation.

---

## Kerberos Constrained Delegation

Correlating users' behavior is the best technique to detect constrained delegation abuse. Suppose we know the location and time a user regularly uses to log in. In that case, it will be easy to alert on other (suspicious) behaviors—for example, consider the account 'Administrator' in the attack described above. If a mature organization uses Privileged Access Workstations (PAWs), they should be alert to any privileged users not authenticating from those machines, proactively monitoring events with the ID `4624` (successful logon).

In some occasions, a successful logon attempt with a delegated ticket will contain information about the ticket's issuer under the `Transited Services` attribute in the events log. This attribute is normally populated if the logon resulted from an `S4U` (`Service For User`) logon process.

`S4U` is a Microsoft extension to the Kerberos protocol that allows an application service to obtain a Kerberos service ticket on behalf of a user; if we recall from the attack flow when utilizing `Rubeus`, we specified this `S4U` extension. Here is an example logon event by using the web service to generate a ticket for the user `Administrator`, which then was used to connect to the Domain Controller (precisely as the attack path above):

General Details

## Logon Information:

Logon Type:	3
Restricted Admin Mode:	-
Virtual Account:	No
Elevated Token:	Yes

Impersonation Level: Impersonation

## New Logon:

Security ID:	EAGLE\Administrator
Account Name:	Administrator
Account Domain:	EAGLE.LOCAL
Logon ID:	0x910D5C
Linked Logon ID:	0x0
Network Account Name:	-
Network Account Domain:	-
Logon GUID:	{c3ad4454-92fa-3a43-51ea-ace6e3d46411}

Correlate user and source IP

## Process Information:

Process ID:	0x0
Process Name:	-

## Network Information:

Workstation Name:	-
Source Network Address:	172.16.18.25
Source Port:	57637

## Detailed Authentication Information:

Logon Process:	Kerberos
Authentication Package:	Kerberos
Transited Services:	webservice@EAGLE.LOCAL
Package Name (NTLM only):	-
Key Length:	0

User who generated the ticket via S4U

## Print Spooler & NTLM Relaying

Exploiting the `PrinterBug` will leave traces of network connections toward the Domain Controller; however, they are too generic to be used as a detection mechanism.

In the case of using `NTLMRelayx` to perform DCSync, no event ID 4662 is generated (as mentioned in the DCSync section); however, to obtain the hashes as DC1 from DC2, there will be a successful logon event for DC1. This event originates from the IP address of the Kali machine, not the Domain Controller, as we can see below:

## Event Properties - Event 4624, Microsoft Windows security auditing.

General Details

### Logon Information:

Logon Type: 3  
Restricted Admin Mode: -  
Virtual Account: No  
Elevated Token: Yes

Impersonation Level: Impersonation

### New Logon:

Security ID: EAGLE\DC1\$  
**Account Name: DC1\$**  
Account Domain: EAGLE  
Logon ID: 0xA65298  
Linked Logon ID: 0x0  
Network Account Name: -  
Network Account Domain: -  
Logon GUID: {00000000-0000-0000-0000-000000000000}

### Process Information:

Process ID: 0x0  
Process Name: -

### Network Information:

Workstation Name: DC1  
**Source Network Address: 172.16.18.20**  
Source Port: 35854

Relayed connection  
for DC1\$ comes from  
a different IP Address

### Detailed Authentication Information:

Logon Process: NtLmSsp  
Authentication Package: NTLM  
Transited Services: -  
Package Name (NTLM only): NTLM V2  
Key Length: 128

A suitable detection mechanism always correlates all logon attempts from core infrastructure servers to their respective IP addresses (which should be static and known).

## Coercing Attacks & Unconstrained Delegation

As mentioned, Windows does not provide an out-of-the-box solution for monitoring RPC activity. The RPC Firewall from [zero networks](#) is an excellent method of detecting the abuse of these functions and can indicate immediate signs of compromise; however, if we follow the general

recommendations to not install third-party software on Domain Controllers then firewall logs are our best chance.

A successful coercing attack with Coercer will result in the following host firewall log, where the machine at `.128` is the attacker machine and the `.200` is the Domain Controller:

pfirewall - Notepad										
File	Edit	Format	View	Help						
2022-12-09	13:35:02	ALLOW	TCP	192.168.28.201	192.168.28.200	56460	49695	0	-	0
2022-12-09	13:35:06	ALLOW	TCP	192.168.28.128	192.168.28.200	60346	445	0	-	0
2022-12-09	13:35:06	ALLOW	TCP	192.168.28.128	192.168.28.200	60348	445	0	-	0
2022-12-09	13:35:06	ALLOW	TCP	192.168.28.128	192.168.28.200	60350	445	0	-	0
2022-12-09	13:35:06	ALLOW	TCP	192.168.28.128	192.168.28.200	60352	445	0	-	0
2022-12-09	13:35:06	ALLOW	TCP	192.168.28.128	192.168.28.200	60354	445	0	-	0
2022-12-09	13:35:06	ALLOW	TCP	192.168.28.200	192.168.28.128	52245	445	0	-	0
2022-12-09	13:35:06	ALLOW	TCP	192.168.28.200	192.168.28.128	52246	445	0	-	0
2022-12-09	13:35:06	ALLOW	TCP	192.168.28.200	192.168.28.128	52247	445	0	-	0
2022-12-09	13:35:07	ALLOW	TCP	192.168.28.200	192.168.28.128	52248	445	0	-	0
2022-12-09	13:35:07	ALLOW	TCP	192.168.28.200	192.168.28.128	52249	445	0	-	0
2022-12-09	13:35:07	ALLOW	TCP	192.168.28.200	192.168.28.128	52250	445	0	-	0
2022-12-09	13:35:07	ALLOW	TCP	192.168.28.200	192.168.28.128	52251	445	0	-	0
2022-12-09	13:35:07	ALLOW	TCP	192.168.28.200	192.168.28.128	52252	445	0	-	0
2022-12-09	13:35:07	ALLOW	TCP	192.168.28.200	192.168.28.128	52253	445	0	-	0
2022-12-09	13:35:07	ALLOW	TCP	192.168.28.128	192.168.28.200	60356	445	0	-	0
2022-12-09	13:35:07	ALLOW	TCP	192.168.28.128	192.168.28.200	60358	445	0	-	0
2022-12-09	13:35:07	ALLOW	TCP	192.168.28.128	192.168.28.200	60360	445	0	-	0
2022-12-09	13:35:07	ALLOW	TCP	192.168.28.200	192.168.28.128	52254	445	0	-	0
2022-12-09	13:35:07	ALLOW	TCP	192.168.28.200	192.168.28.128	52255	445	0	-	0
2022-12-09	13:35:07	ALLOW	TCP	192.168.28.200	192.168.28.128	52256	445	0	-	0
2022-12-09	13:35:07	ALLOW	TCP	192.168.28.200	192.168.28.128	52257	445	0	-	0
2022-12-09	13:35:07	ALLOW	TCP	192.168.28.128	192.168.28.200	60362	445	0	-	0
2022-12-09	13:35:07	ALLOW	TCP	192.168.28.128	192.168.28.200	60364	445	0	-	0
2022-12-09	13:35:07	ALLOW	TCP	192.168.28.128	192.168.28.200	60366	445	0	-	0
2022-12-09	13:35:07	ALLOW	TCP	192.168.28.200	192.168.28.128	52258	445	0	-	0
2022-12-09	13:35:07	ALLOW	TCP	192.168.28.200	192.168.28.128	52259	445	0	-	0
2022-12-09	13:35:07	ALLOW	TCP	192.168.28.200	192.168.28.128	52260	445	0	-	0
2022-12-09	13:35:07	ALLOW	TCP	192.168.28.200	192.168.28.128	52261	445	0	-	0
2022-12-09	13:35:14	ALLOW	UDP	192.168.28.130	192.168.28.200	62324	53	0	-	-

Inbound requests to port 445  
and outbound connections  
following towards port 445 too

We can see plenty of incoming connections to the DC, followed up by outbound connections from the DC to the attacker machine; this process repeats a few times as Coercer goes through several different functions. All of the outbound traffic is destined for port 445.

If we go forward and block outbound traffic to port 445, then we will observe the following behavior:

2022-12-09	13:44:35	ALLOW	TCP	192.168.28.128	192.168.28.200	60436	445	0	-	0
2022-12-09	13:44:35	ALLOW	TCP	192.168.28.128	192.168.28.200	60438	445	0	-	0
2022-12-09	13:44:35	ALLOW	TCP	192.168.28.128	192.168.28.200	60440	445	0	-	0
2022-12-09	13:44:35	ALLOW	TCP	192.168.28.128	192.168.28.200	60442	445	0	-	0
2022-12-09	13:44:35	ALLOW	TCP	192.168.28.128	192.168.28.200	60444	445	0	-	0
2022-12-09	13:44:35	DROP	TCP	192.168.28.200	192.168.28.128	52301	445	0	-	0

Outbound traffic to  
untrusted IPs/VLANs  
is blocked

Incoming packets  
from Coercer

Now we can see that even though the inbound connection is successful, the firewall drops the outbound one, and consequently, the attacker does not receive any coerced TGTs. Sometimes, when port 445 is blocked, the machine will attempt to connect to port 139 instead, so blocking both ports `139` and `445` is recommended.

The above can also be used for detection, as any unexpected dropped traffic to ports 139 or 445 is suspicious.

---

## Object ACLs

Fortunately, we have several ways to detect if AD objects are modified. Unfortunately, the events generated for modified objects are incomplete, as they do not provide granular visibility over what was changed. For example, in the first case described above, Bob modified Anni by adding an SPN value. By doing so, Bob will have the means to perform Kerberoasting against Anni. When the SPN value gets added, an event with the ID 4738 , "A user account was changed", is generated. However, this event does not demonstrate all modified user properties, including the SPN. Therefore, the event only notifies about the modification of a user but does not specify what exactly was changed ( although it does have a fair amount of fields that can be useful). Below is the event that will be generated if Bob adds any bogus SPN value to Anni's User Object:

## Event 4738, Microsoft Windows security auditing.

General Details

A user account was changed.

Subject:

Security ID:	EAGLE\bob
Account Name:	bob
Account Domain:	EAGLE
Logon ID:	0x1C057AC

Bob modified Anni

Target Account:

Security ID:	EAGLE\anni
Account Name:	anni
Account Domain:	EAGLE

Changed Attributes:

SAM Account Name:	-
Display Name:	-
User Principal Name:	-
Home Directory:	-
Home Drive:	-
Script Path:	-
Profile Path:	-
User Workstations:	-
Password Last Set:	-
Account Expires:	-
Primary Group ID:	-
AllowedToDelegateTo:	-

No details of what was changed

However, using this event, we can tell if a non-privileged user performs privileged actions on another user. If, for example, all privileged users have a naming convention that begins with "adminxxxx", then any change not associated with "adminxxxx" is suspicious. If an ACL abuse leads to a password reset, the event ID 4724 will be logged.

Similarly, if Bob were to perform the second scenario, an event with ID 4742 would be generated, which is also unfortunately limited in the information it can provide; however, it notifies about the action that the user account Bob is compromised and used maliciously. The following was the event ID 4742 generated when Bob modified Server01:

## Event 4742 Microsoft Windows security auditing.

General Details

A computer account was changed.

Subject:

Security ID: EAGLE\bob  
Account Name: bob  
Account Domain: EAGLE  
Logon ID: 0x1BC3119

Bob modified Server01

Computer Account That Was Changed:

Security ID: EAGLE\SERVER01\$  
Account Name: SERVER01\$  
Account Domain: EAGLE

Changed Attributes:

SAM Account Name:  
Display Name:  
User Principal Name:  
Home Directory:  
Home Drive:  
Script Path:  
Profile Path:  
User Workstations:  
Password Last Set:  
Account Expires:  
Primary Group ID: -  
AllowedToDelegateTo:  
Old UAC Value:  
New UAC Value:

-  
-  
-  
-  
-  
-  
-  
-  
-  
-  
-  
-  
-  
-

No details what was changed

## PKI - ESC1

When the CA generates the certificate, two events will be logged, one for the received request and one for the issued certificate, if it succeeds. Those events have the IDs of 4886 and 4887 as shown below:

## Event 4886, Microsoft Windows security auditing.

General Details

Certificate Services received a certificate request.

Request ID: 47  
Requester: EAGLE\bob  
Attributes:

ccm:WS001.eagle.local

## Event 4887, Microsoft Windows security auditing.

General Details

Certificate Services approved a certificate request and issued a certificate.

Request ID: 47

Requester: EAGLE\bob

Attributes:

ccm:WS001.eagle.local

Disposition: 3

SKI: f0 8b 25 f8 6b 5c 9b 91 d4 ff 4e 28 07 16 fb 8d e7 e0 dd f7

Subject: CN=bob, OU=EagleUsers, DC=eagle, DC=local

No details of SAN,  
only requester  
information

Unfortunately, we can only tell that Bob requested a certificate from WS001; we cannot know if the request specified the SAN.

The CA contains a list of all issued certificates, so if we look there, we will see the request for certificate ID 36 (the one from the attack scenario above):

Request ID	Requester Name	Binary Certificate	Certificate Template	Serial Number	Certificate Effective Date	Certificate Expiration Date
36	EAGLE\bob	-----BEGIN CERTIF...	UserCert (3.6.1.4.1.311.2...	1600000024cc5a...	10/16/2022 12:04 AM	10/16/2024 12:14 AM

The general overview of the GUI tool does not display the SAN either, but we can tell that a certificate was issued via the vulnerable template. If we want to find the SAN information, we'll need to open the certificate itself:

A screenshot of a Windows Certificate Details window. The window has tabs at the top: General, Details, and Certification Path. The Details tab is selected. Below the tabs is a dropdown menu labeled "Show: <All>". The main area contains a table with two columns: "Field" and "Value". The "Field" column lists various certificate fields, and the "Value" column shows their corresponding values. A red box highlights the "Subject Alternative Name" field, which has a value of "Other Name:Principal Name=A...". Another red box highlights the "Other Name" entry in the "Value" column, which is "Principal Name=Administrator".

Field	Value
Enhanced Key Usage	Smart Card Logon (1.3.6.1.4.1....)
Application Policies	[1]Application Certificate Policy:...
SMIME Capabilities	[1]SMIME Capability: Object ID...
Subject Key Identifier	f82701f6139a3512be297f754d...
Subject Alternative Name	Other Name:Principal Name=A...
Authority Key Identifier	KeyID=7c0995ebc086e3f1656...
CRL Distribution Points	[1]CRL Distribution Point: Distri...
Authority Information Access	[1]Authority Info Access: Acces...

Other Name:  
Principal Name=Administrator

There is also the possibility to view that programmatically: the command `certutil -view` will dump everything on the CA with all of the information about each certificate (this can be massive in a large environment):

```
Issued Country/Region: EMPTY
Issued Organization: EMPTY
Issued Organization Unit: "EagleUsers"
Issued Common Name: "bob"
Issued City: EMPTY
Issued State: EMPTY
Issued Title: EMPTY
Issued First Name: EMPTY
Issued Initials: EMPTY
Issued Last Name: EMPTY
Issued Domain Component: "local"
eagle"
Issued Email Address: EMPTY
Issued Street Address: EMPTY
Issued Unstructured Name: EMPTY
Issued Unstructured Address: EMPTY
Issued Device Serial Number: EMPTY

Request Attributes:
  RequestOSVersion: "6.2.9200.2"
  SAN: "upn=Administrator"
  RequestCSPProvider: "Microsoft Strong Cryptographic Provider"
  ccm: "WS001.eagle.local"

Certificate Extensions:
  1.3.6.1.4.1.311.21.7: Flags = 20000(Origin=Policy), Length = 31
  Certificate Template Information
    Template=UserCert(1.3.6.1.4.1.311.21.8.11545821.9410490.6243468.13526546.8366809.221.16588593.10220936)
      Major Version Number=100
      Minor Version Number=5
```

Bob requested certificate for the user Administrator from WS001 for the template UserCert

With some scripting, we can automate parsing and discovery of abused vulnerable templates by threat agents.

Finally, if you recall, in the attack, we used the obtained certificate for authentication and obtained a TGT; AD will log this request with the event ID 4768 , which will specifically have information about the logon attempt with a certificate:

Event 4768, Microsoft Windows security auditing.

General Details

A Kerberos authentication ticket (TGT) was requested.

Account Information:

Account Name:	Administrator
Supplied Realm Name:	eagle.local
User ID:	EAGLE\Administrator

Service Information:

Service Name:	krbtgt
Service ID:	EAGLE\krbtgt

Network Information:

Client Address:	::ffff:172.16.18.25
Client Port:	64869

Additional Information:

Ticket Options:	0x40800010
Result Code:	0x0
Ticket Encryption Type:	0x17
Pre-Authentication Type:	16

Certificate Information:

Certificate Issuer Name:	eagle-PKI-CA
Certificate Serial Number:	160000002C7ACD5E9B6DF3756500000000002C
Certificate Thumbprint:	7104BB8ACBEF5FD6438FC5F48BDC64DB6E6164A5

Correlate User / Client IP for suspicious behavior

Login with certificate

Note that events 4886 and 4887 will be generated on the machine issuing the certificate rather than the domain controller. If GUI access is not available, we can use PSSession to interact with the PKI machine, and the Get-WinEvent cmdlet to search for the events:

```
C:\Users\bob\Downloads>runas /user:eagle\htb-student powershell
```

```
Enter the password for eagle\htb-student:
```

```
Attempting to start powershell as user "eagle\htb-student" ...
```