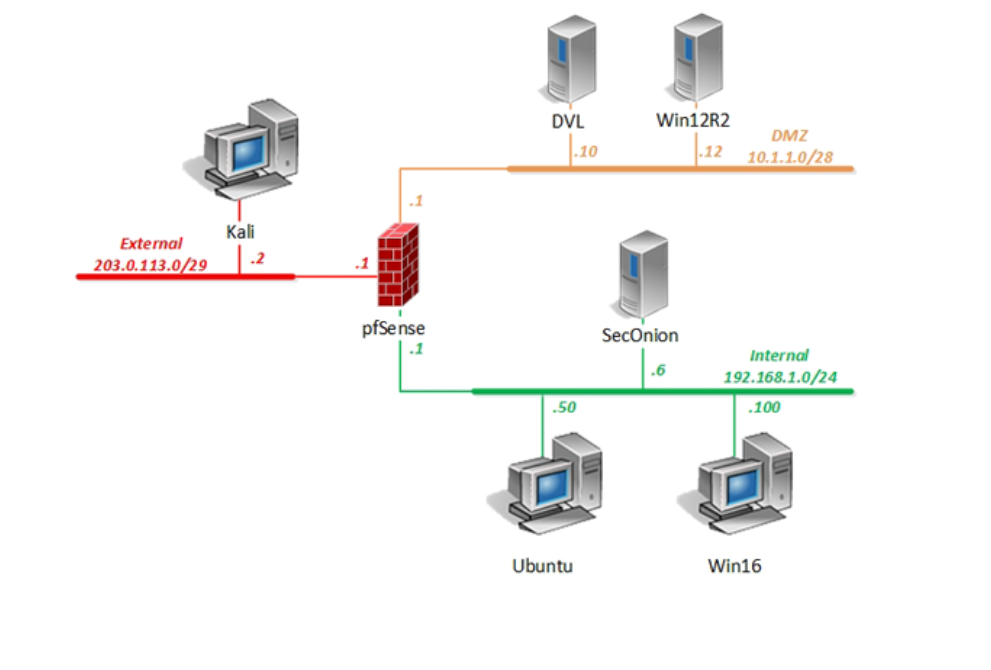
Lab 4-5: active reconnaissance with windows and linux



For windows

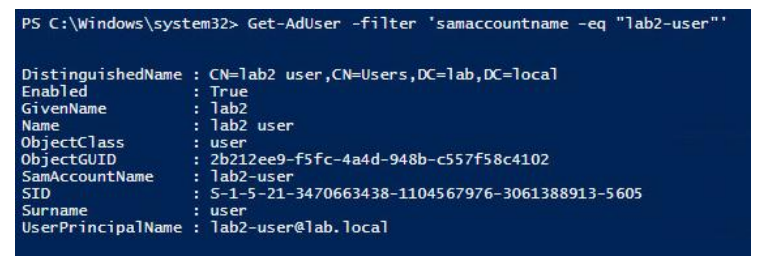
On a windows server virtual machine, I opened its powershell as administrator and ran *$cred=Get-Credential* on it.

$cred means Get-Credential without asking for prompts.

Then ran *Get-ADGroupMember -Credential $cred -server Win12R2 “Domain Users” | select samaccountname* to get a list of domain users on the system.   
get-adgroupmember gets the members of an active directory group (ADGroupmember).  
the command specifies to get the domain users under samaccountname category (SAM account names) in Win12R2 server.

Then the next command to run is *Get-ADGroupMember -Credential $cred -server Win12R2 “Domain Admins”* to get the admin members list.   
again, you could pipe with select samaccountname to filter the SAM account names. These are user logon names user in windows operating systems.   
ex: *Get-ADGroupMember -Credential $cred -server Win12R2 “Domain Admins” | select samaccountname*

Get-ADDomain is for domain itself, and Get-ADUser is for each user which can use -filter to see if a specific account is enabled.   
ex: *Get-ADUser -filter ‘samaccountname -eq “lab2-user”’*

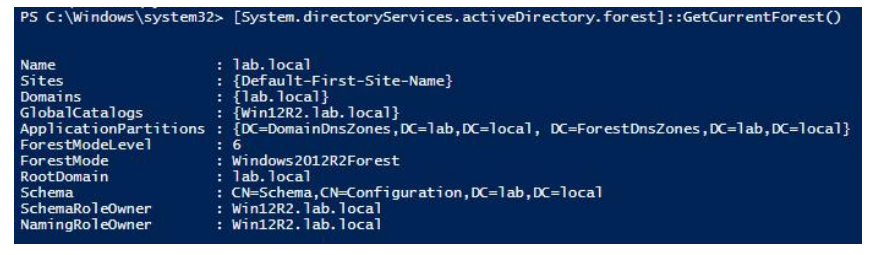


This shows SID which is the security identifier of an account.

Get-ADPrincipalGroupMembership <account> is to check which group <account> belongs to.

So far this has been for windows server.

For windows client, you need to use the .NET command.

For ex: *[System.directoryServices.activeDirectory.forest]::GetCurrentForest()*  
this shows the Active Directory that the account, lab-user, belongs to.   
  
forest in this context is the top most level of grouping or organization which eases deployment of access control.

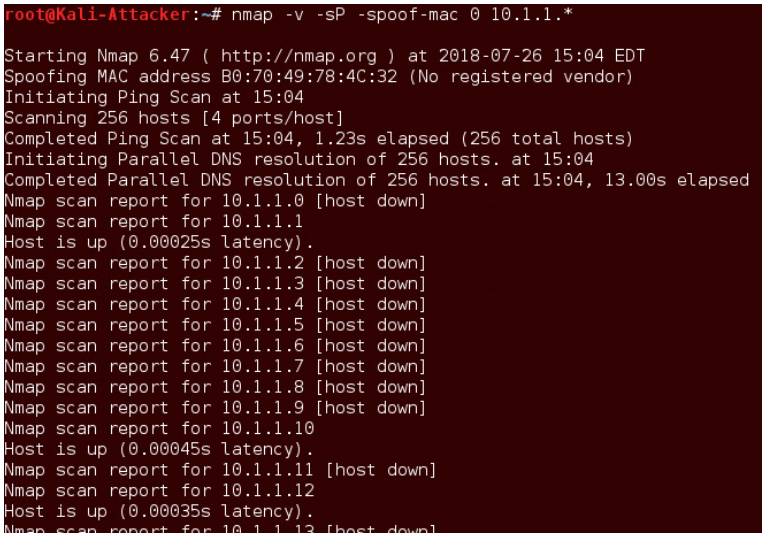
To identify which domain the user is associated with, *[System.DirectoryServices.ActiveDirectory.Domain]::GetCurrentDomain()* command shows the information you need.

For linux, I will be trying the nmap to do the active reconnaissance.

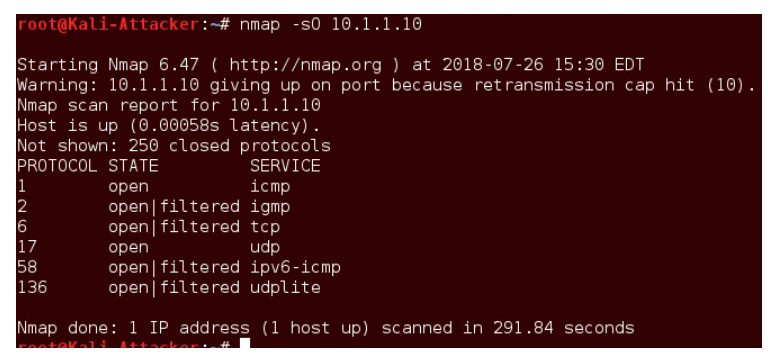
First, I tried -sP to quick ping scan through network ID of 10.1.1.\*



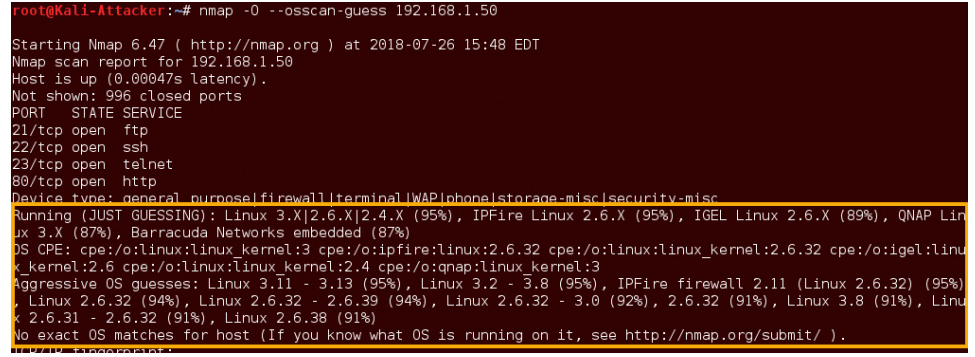
And I could ping scan while spoofing the source MAC address at the same time.



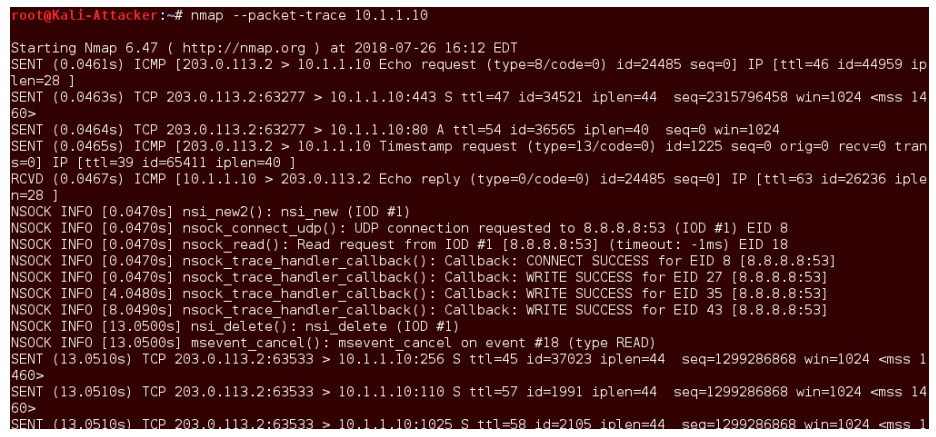
Based on the manual, -sO gives me IP protocols that are supported by the targeted host.



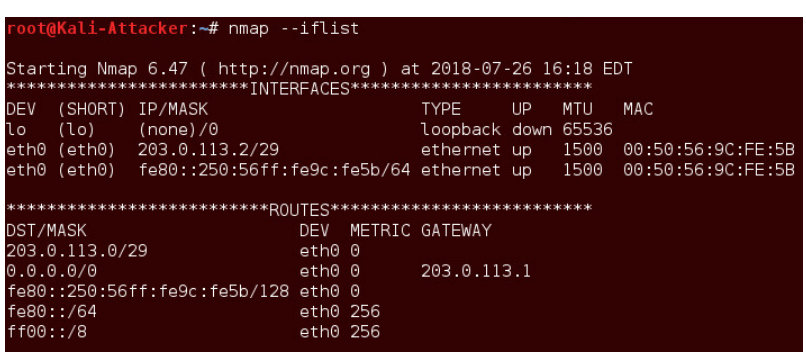
-sT is for TCP scan against a host and -O shows me the operating system the host runs on.   
but if -O fails, I can guess with -O –osscan-guess



To check the status of a specific port, -p <portnumber> <host ip> reveals the information.

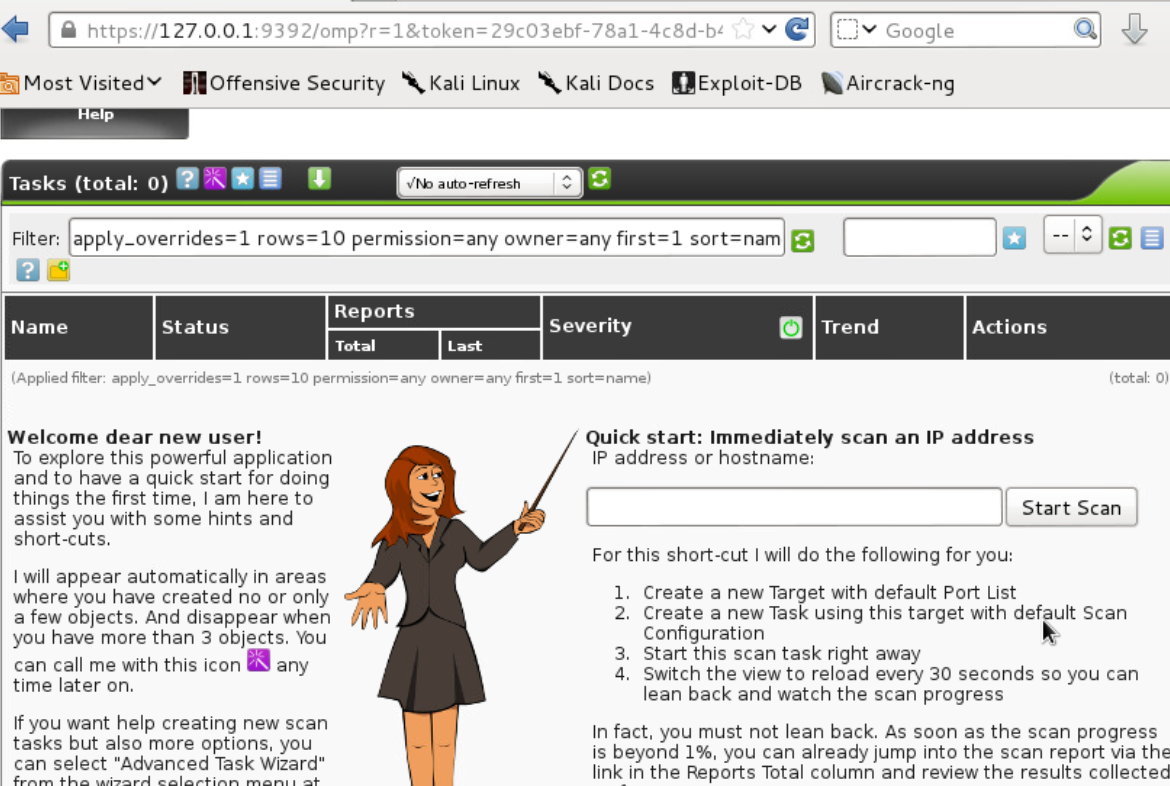
Nmap also has packet-trace function. –packet-trace. 

--iflist lets me see local host data about which interfaces are up and what the route table looks like.

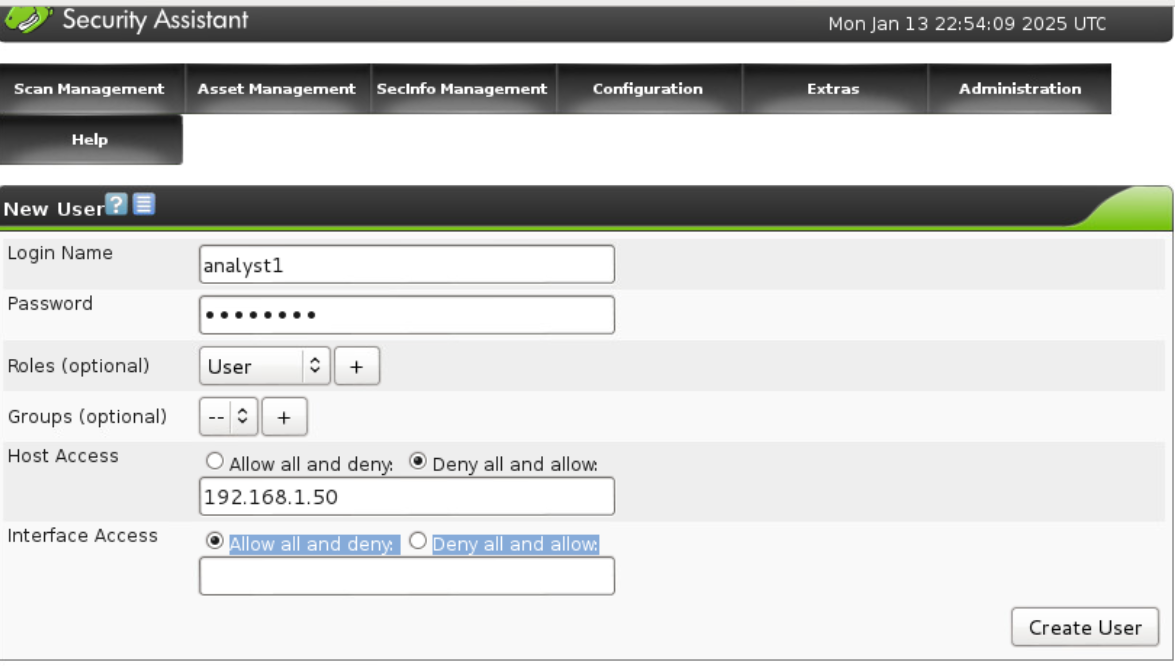
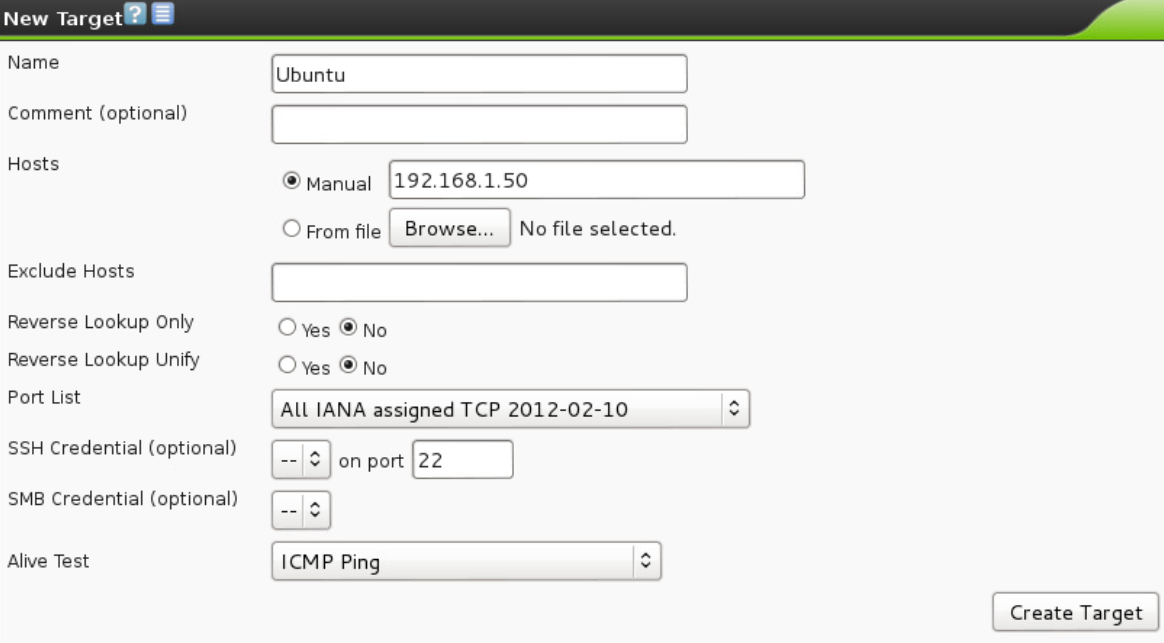


And final -sV helps to detect remote services and daemons.

Similar to nessus, there is a program called OpenVAS.



On here, I tried to make a new target to scan and a new account.

  
and finally scheduling tasks  
