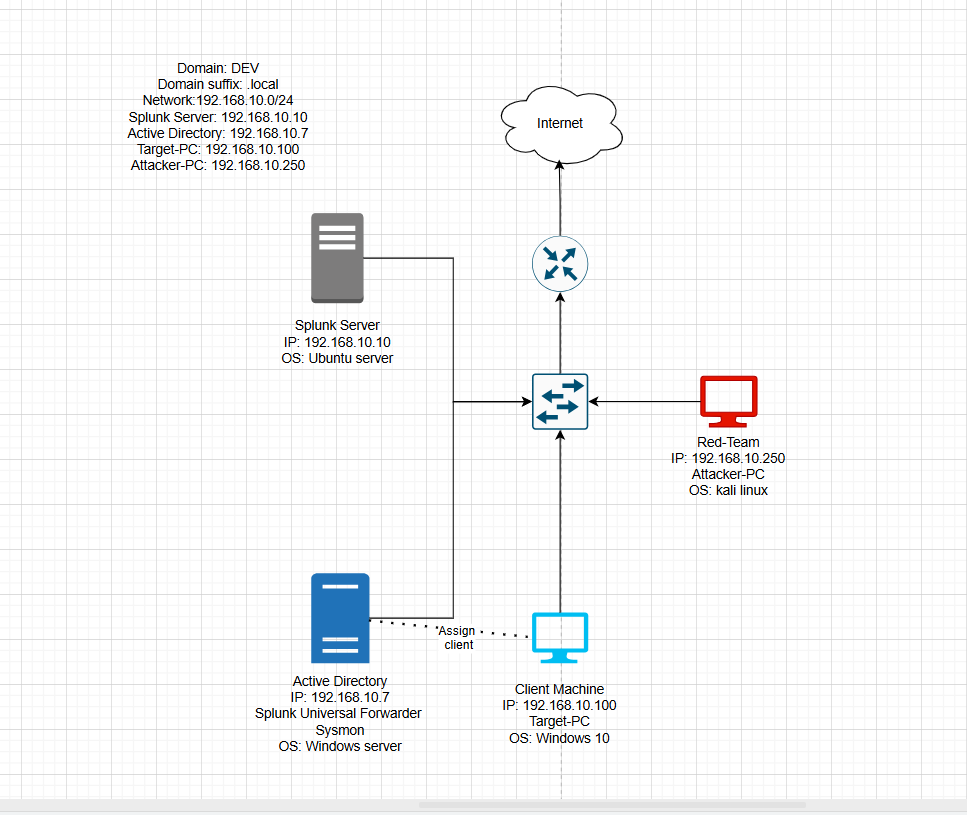
**Active Directory SOC Analysis Home Lab Setup Guide**

This comprehensive guide provides detailed step-by-step instructions for creating a complete Active Directory home lab environment with integrated security monitoring capabilities using Oracle VirtualBox. The lab simulates a real-world enterprise network environment suitable for cybersecurity learning, SOC analysis training, and penetration testing practice.



Active Directory SOC Analysis Home Lab Network Topology - DEV.local Domain

The lab environment consists of four virtual machines connected via an internal network, providing a contained environment for Active Directory administration, security monitoring with Splunk, and red team/blue team exercises.[[1]](#fn1)[[2]](#fn2)[[3]](#fn3)

**Prerequisites and Requirements**

**Hardware Specifications**

**Minimum System Requirements:**

* **RAM:** 20GB minimum (24GB recommended)
* **CPU:** Quad-core processor with virtualization extensions (Intel VT-x/AMD-V)
* **Storage:** 250GB available disk space (SSD recommended)
* **Network:** Stable internet connection for downloads and updates

**Virtual Machine Resource Allocation:**

* **Active Directory Server:** 4GB RAM, 100GB storage
* **Splunk Server:** 8GB RAM, 100GB storage
* **Windows 10 Client:** 4GB RAM, 80GB storage
* **Kali Linux Attacker:** 4GB RAM, 50GB storage[[4]](#fn4)[[3]](#fn3)[[5]](#fn5)

**Required Software Downloads**

**Operating System ISO Files:**

1. **Windows Server 2019:** Microsoft Evaluation Center (180-day evaluation)
2. **Ubuntu Server 22.04 LTS:** Official Ubuntu website
3. **Windows 10 Enterprise:** Microsoft Evaluation Center (90-day evaluation)
4. **Kali Linux:** Official Kali Linux website (latest version)

**Additional Software Components:**

* Oracle VirtualBox (latest version)
* VirtualBox Extension Pack
* Splunk Enterprise (free trial license)
* Splunk Universal Forwarder
* Sysmon (Microsoft Sysinternals)
* SwiftOnSecurity Sysmon configuration[[1]](#fn1)[[5]](#fn5)[[6]](#fn6)

**Phase 1: VirtualBox Environment Configuration**

**Installing VirtualBox**

1. **Download and Install VirtualBox:**

# Download from https://www.virtualbox.org/  
# Install with administrator privileges  
# Accept default installation options

1. **Install VirtualBox Extension Pack:**
   * Download the Extension Pack matching your VirtualBox version
   * Open VirtualBox → File → Preferences → Extensions
   * Click the package icon and select the downloaded extension pack
   * Accept the license agreement[[4]](#fn4)[[7]](#fn7)

**Creating Internal Network Infrastructure**

1. **Configure NAT Network:**
   * Open VirtualBox Manager
   * Navigate to **File → Preferences → Network**
   * Click the **"+"** icon to create new NAT Network
   * **Network Configuration:**
     + **Name:** DEV-Lab-Network
     + **Network CIDR:** 192.168.10.0/24
     + **Enable DHCP:** Unchecked (we'll use static IPs)
     + **IPv6:** Disabled
2. **Network Verification:**
   * Gateway IP: 192.168.10.1 (automatically assigned by VirtualBox)
   * Available IP range: 192.168.10.2 - 192.168.10.254
   * Subnet mask: 255.255.255.0[[8]](#fn8)[[9]](#fn9)[[10]](#fn10)

**Phase 2: Active Directory Domain Controller Setup**

**Creating the Domain Controller VM**

1. **Virtual Machine Creation:**

VM Name: DEV-DC01  
Type: Microsoft Windows  
Version: Windows Server 2019 (64-bit)  
Memory: 4096 MB (4GB)  
Hard Disk: 100 GB (Dynamically allocated VDI)

1. **Network Configuration:**
   * **Adapter 1:** Enabled, attached to NAT Network
   * **Name:** DEV-Lab-Network
   * **Advanced:** Allow All, Cable Connected[[11]](#fn11)[[7]](#fn7)

**Windows Server 2019 Installation**

1. **Installation Process:**
   * Mount Windows Server 2019 ISO to the VM
   * Start VM and boot from ISO
   * Select **"Windows Server 2019 Standard Evaluation (Desktop Experience)"**
   * Choose **Custom Installation**
   * Create Administrator password: **P@ssw0rd123!**
2. **Initial Configuration:**
   * Install VirtualBox Guest Additions for better performance
   * Enable Remote Desktop if needed
   * Update Windows to latest patches[[11]](#fn11)[[12]](#fn12)

**Network Configuration for Domain Controller**

1. **Configure Static IP Address:**

# Open Network and Sharing Center  
# Change adapter settings  
# Right-click network adapter → Properties  
# Select Internet Protocol Version 4 (TCP/IPv4)

1. **Static IP Configuration:**
   * **IP Address:** 192.168.10.7
   * **Subnet Mask:** 255.255.255.0
   * **Default Gateway:** 192.168.10.1
   * **Preferred DNS Server:** 192.168.10.7 (self-referencing)
   * **Alternate DNS Server:** 8.8.8.8[[4]](#fn4)[[11]](#fn11)
2. **Server Hostname Configuration:**
   * Rename computer to **DEV-DC01**
   * Restart server to apply changes

**Active Directory Domain Services Installation**

1. **Install AD DS Role:**

# Using Server Manager  
Add-WindowsFeature -Name AD-Domain-Services -IncludeManagementTools

**Via Server Manager GUI:**

* + Open Server Manager
  + Click **"Add roles and features"**
  + Select **"Role-based or feature-based installation"**
  + Choose **"Active Directory Domain Services"**
  + Include DNS Server when prompted
  + Complete installation[[11]](#fn11)[[7]](#fn7)

1. **Promote to Domain Controller:**

# Using PowerShell (alternative to GUI)  
Install-ADDSForest -DomainName "DEV.local" -SafeModeAdministratorPassword (ConvertTo-SecureString "P@ssw0rd123!" -AsPlainText -Force)

**Via GUI (Recommended for beginners):**

* + Click yellow flag in Server Manager
  + Select **"Promote this server to a domain controller"**
  + Choose **"Add a new forest"**
  + **Root domain name:** DEV.local
  + **Forest/Domain functional level:** Windows Server 2016 or higher
  + **DNS Options:** Install DNS Server (checked)
  + **DSRM Password:** P@ssw0rd123!
  + Complete configuration and restart[[7]](#fn7)[[11]](#fn11)

**DHCP Server Configuration**

1. **Install DHCP Role:**

Add-WindowsFeature -Name DHCP -IncludeManagementTools

1. **Configure DHCP Scope:**
   * Open DHCP Management Console
   * Create new scope: **DEV-Lab-Scope**
   * **IP Range:** 192.168.10.100 - 192.168.10.199
   * **Subnet Mask:** 255.255.255.0
   * **Gateway:** 192.168.10.1
   * **DNS Servers:** 192.168.10.7
   * **Lease Duration:** 8 hours
   * **Exclusions:** 192.168.10.7 (DC), 192.168.10.10 (Splunk), 192.168.10.250 (Kali)[[13]](#fn13)[[14]](#fn14)
2. **Authorize DHCP Server:**

Add-DhcpServerInDC -DnsName "DEV-DC01.DEV.local"

**Phase 3: Splunk Server Setup (Ubuntu)**

**Creating Ubuntu Server VM**

1. **VM Configuration:**

VM Name: SPLUNK-01  
Type: Linux  
Version: Ubuntu (64-bit)  
Memory: 8192 MB (8GB)  
Hard Disk: 100 GB  
Network: NAT Network (DEV-Lab-Network)

1. **Ubuntu Server Installation:**
   * Mount Ubuntu Server 22.04 LTS ISO
   * Follow installation wizard
   * Create user: **splunkadmin**
   * Install OpenSSH server for remote access[[1]](#fn1)[[5]](#fn5)

**Network Configuration for Splunk Server**

1. **Configure Static IP:**

sudo nano /etc/netplan/00-installer-config.yaml

**Configuration:**

network:  
 version: 2  
 ethernets:  
 enp0s3:  
 addresses:  
 - 192.168.10.10/24  
 routes:  
 - to: default  
 via: 192.168.10.1  
 nameservers:  
 addresses: [192.168.10.7, 8.8.8.8]  
 search: [DEV.local]

1. **Apply Network Configuration:**

sudo netplan apply  
sudo systemctl restart systemd-networkd

**Splunk Enterprise Installation and Configuration**

1. **Download and Install Splunk:**

# Download Splunk Enterprise  
wget -O splunk-9.1.2-linux.tgz "https://download.splunk.com/products/splunk/releases/9.1.2/linux/splunk-9.1.2-b6b9c8185839-Linux-x86\_64.tgz"  
  
# Extract to /opt  
sudo tar -xzf splunk-9.1.2-linux.tgz -C /opt/  
sudo chown -R splunkadmin:splunkadmin /opt/splunk

1. **Initial Splunk Configuration:**

cd /opt/splunk/bin  
./splunk start --accept-license  
# Create admin credentials when prompted  
# Username: admin  
# Password: P@ssw0rd123!  
  
# Enable boot startup  
sudo ./splunk enable boot-start -user splunkadmin

1. **Configure Data Receiving:**

# Enable receiving on port 9997  
./splunk enable listen 9997  
  
# Configure receiving from Universal Forwarders  
./splunk add forward-server 192.168.10.10:9997

1. **Web Interface Verification:**
   * Access Splunk at [**http://192.168.10.10:8000**](http://192.168.10.10:8000)
   * Login with admin credentials
   * Verify receiving configuration in Settings → Forwarding and receiving

**Phase 4: Windows 10 Client Setup**

**Creating Windows 10 Target VM**

1. **VM Creation:**

VM Name: TARGET-PC  
Type: Microsoft Windows   
Version: Windows 10 (64-bit)  
Memory: 4096 MB (4GB)  
Hard Disk: 80 GB  
Network: NAT Network (DEV-Lab-Network)

1. **Windows 10 Installation:**
   * Use Windows 10 Enterprise evaluation ISO
   * Create local user account initially: **localuser**
   * Complete Windows setup and install Guest Additions[[15]](#fn15)[[16]](#fn16)

**Network Configuration and Domain Join**

1. **Network Configuration:**

# Configure to obtain IP automatically from DHCP  
# Verify DNS points to domain controller (192.168.10.7)  
# IP should be assigned in range 192.168.10.100-199

1. **Join Domain:**

# Method 1: GUI  
# Right-click "This PC" → Properties  
# Change settings → Change  
# Select "Domain" → Enter "DEV.local"  
# Use credentials: DEV\Administrator

**PowerShell Method:**

Add-Computer -DomainName "DEV.local" -Credential (Get-Credential) -Restart

1. **Post-Domain Join Verification:**

whoami # Should show DEV\username  
echo %USERDOMAIN% # Should show DEV  
nslookup DEV.local # Should resolve to 192.168.10.7

**Installing Security Monitoring Tools**

**Sysmon Installation and Configuration:**

1. **Download Sysmon and Configuration:**

# Download from https://docs.microsoft.com/sysinternals/downloads/sysmon  
# Download SwiftOnSecurity configuration from GitHub  
curl -o sysmonconfig-export.xml https://raw.githubusercontent.com/SwiftOnSecurity/sysmon-config/master/sysmonconfig-export.xml

1. **Install Sysmon:**

# Run as Administrator  
sysmon64.exe -accepteula -i sysmonconfig-export.xml

1. **Verify Sysmon Installation:**

# Check Event Viewer  
# Navigate to: Applications and Services Logs → Microsoft → Windows → Sysmon → Operational  
# Should see Sysmon events being logged

**Splunk Universal Forwarder Installation:**

1. **Download and Install:**

# Download from Splunk website  
# Install via MSI with forwarding configuration  
msiexec.exe /i splunkforwarder-9.1.2-x64-release.msi RECEIVING\_INDEXER="192.168.10.10:9997" AGREETOLICENSE=yes /quiet

1. **Configure Data Inputs:**

cd "C:\Program Files\SplunkUniversalForwarder\bin"  
  
# Add Windows Event Logs  
splunk add monitor "Application"  
splunk add monitor "Security"   
splunk add monitor "System"  
  
# Add Sysmon logs  
splunk add monitor "Microsoft-Windows-Sysmon/Operational"

1. **Configure outputs.conf:**

# Edit: C:\Program Files\SplunkUniversalForwarder\etc\system\local\outputs.conf  
[tcpout]  
defaultGroup = default-autolb-group  
  
[tcpout:default-autolb-group]  
server = 192.168.10.10:9997

1. **Restart Universal Forwarder:**

net stop SplunkForwarder  
net start SplunkForwarder

**Phase 5: Kali Linux Attacker Setup**

**Creating Kali Linux VM**

1. **VM Configuration:**

VM Name: ATTACKER-PC  
Type: Linux  
Version: Debian (64-bit)  
Memory: 4096 MB (4GB)  
Hard Disk: 50 GB  
Network: NAT Network (DEV-Lab-Network)

1. **Kali Linux Installation:**
   * Use latest Kali Linux ISO
   * Create user: **kali**
   * Install with default desktop environment[[17]](#fn17)

**Network Configuration for Kali Linux**

1. **Configure Static IP:**

sudo nano /etc/network/interfaces

**Configuration:**

auto eth0  
iface eth0 inet static  
 address 192.168.10.250  
 netmask 255.255.255.0  
 gateway 192.168.10.1  
 dns-nameservers 192.168.10.7 8.8.8.8  
 dns-search DEV.local

1. **Apply Network Configuration:**

sudo systemctl restart networking  
ip addr show eth0 # Verify IP configuration

**Installing Additional Security Tools**

1. **Update System and Install Tools:**

sudo apt update && sudo apt upgrade -y  
  
# Install additional penetration testing tools  
sudo apt install -y \  
 nmap \  
 enum4linux \  
 smbclient \  
 bloodhound \  
 neo4j \  
 crackmapexec \  
 impacket-scripts \  
 responder \  
 john \  
 hashcat

1. **Active Directory Testing Tools:**

# Install BloodHound for AD enumeration  
sudo apt install bloodhound neo4j  
  
# Install PowerShell for Windows PowerShell scripts  
sudo apt install powershell  
  
# Install additional Python-based AD tools  
pip3 install ldap3 dnspython

**Phase 6: Testing and Validation**

**Network Connectivity Testing**

1. **Domain Controller Testing:**

# Test from DEV-DC01  
ping 192.168.10.1 # Gateway  
ping 192.168.10.10 # Splunk Server  
ping 192.168.10.100 # Windows 10 Client (if static)  
ping 192.168.10.250 # Kali Linux  
nslookup DEV.local # DNS resolution

1. **Windows 10 Client Testing:**

# Test from TARGET-PC  
ping DEV-DC01.DEV.local  
nslookup DEV.local  
gpupdate /force  
whoami /all

1. **Kali Linux Testing:**

# Test from ATTACKER-PC  
nmap -sn 192.168.10.0/24 # Network discovery  
enum4linux 192.168.10.7 # AD enumeration  
dig @192.168.10.7 DEV.local # DNS query

**Active Directory Validation**

1. **Domain Controller Health Check:**

# Run on DEV-DC01  
dcdiag /test:dns /v  
dcdiag /c /comprehensive  
repadmin /showrepl  
netdom query fsmo

1. **DHCP Verification:**

# Check DHCP leases  
netsh dhcp server show scope  
Get-DhcpServerv4Scope  
Get-DhcpServerv4Lease -ScopeId 192.168.10.0

1. **DNS Testing:**

nslookup DEV-DC01.DEV.local  
nslookup \_ldap.\_tcp.DEV.local

**Splunk Data Validation**

1. **Verify Data Ingestion:**
   * Access Splunk Web Interface: [**http://192.168.10.10:8000**](http://192.168.10.10:8000)
   * Run search queries:

# Basic connectivity test  
index=main earliest=-15m  
  
# Windows Event Logs  
index=main sourcetype=WinEventLog  
  
# Sysmon Events  
index=main source="WinEventLog:Microsoft-Windows-Sysmon/Operational"  
  
# Authentication Events  
index=main EventCode=4624 OR EventCode=4625

1. **Create Basic Dashboards:**
   * Failed Login Attempts
   * Process Creation Events
   * Network Connections
   * DNS Queries[[1]](#fn1)[[5]](#fn5)

**Phase 7: Security Testing Scenarios**

**Basic Red Team Exercises**

1. **Network Reconnaissance:**

# From Kali Linux  
nmap -sC -sV 192.168.10.0/24  
enum4linux -a 192.168.10.7  
crackmapexec smb 192.168.10.0/24

1. **Active Directory Enumeration:**

# Using BloodHound  
bloodhound-python -u 'guest' -p '' -d DEV.local -ns 192.168.10.7 -c all  
  
# Using PowerShell from Kali  
pwsh -c "Get-ADUser -Filter \* -Server 192.168.10.7"

1. **Monitor in Splunk:**

# Monitor reconnaissance attempts  
index=main sourcetype=WinEventLog EventCode=4625  
| stats count by src\_ip, user  
| where count > 5

**Troubleshooting Guide**

**Common Network Issues**

**Problem: VMs Cannot Communicate**

# Solution steps:  
1. Verify all VMs on same NAT Network (DEV-Lab-Network)  
2. Check firewall settings on each VM  
3. Verify IP configuration with: ip addr show (Linux) or ipconfig (Windows)  
4. Test with ping between VMs

**Problem: DNS Resolution Failures**

# Solution steps:  
1. Verify DNS server IP (192.168.10.7) in network configuration  
2. Check DNS service on Domain Controller: Get-Service DNS  
3. Flush DNS cache: ipconfig /flushdns  
4. Test with: nslookup DEV.local 192.168.10.7

**Active Directory Issues**

**Problem: Domain Join Failures**

# Solutions:  
1. Verify network connectivity to DC: ping DEV-DC01.DEV.local  
2. Check time synchronization: w32tm /query /status  
3. Verify DNS resolution: nslookup DEV.local  
4. Use domain admin credentials  
5. Ensure Windows 10 is Pro/Enterprise edition

**Problem: DHCP Not Working**

# Solutions:  
1. Check DHCP service: Get-Service DHCPServer  
2. Verify DHCP authorization: Get-DhcpServerInDC  
3. Check scope status: Get-DhcpServerv4Scope  
4. Review exclusions and reservations

**Splunk Universal Forwarder Issues**

**Problem: No Data in Splunk**

# Diagnosis steps:  
1. Check forwarder service: net start SplunkForwarder  
2. Test connectivity: telnet 192.168.10.10 9997  
3. Review logs: C:\Program Files\SplunkUniversalForwarder\var\log\splunk\splunkd.log  
4. Verify outputs.conf configuration  
5. Check Splunk receiving configuration

**Problem: Sysmon Events Missing**

# Solutions:  
1. Verify Sysmon service: sc query sysmon64  
2. Check Event Viewer for Sysmon events  
3. Verify configuration: sysmon64 -c  
4. Restart Sysmon: net stop sysmon64 && net start sysmon64

**Performance Optimization**

**VM Performance Tips:**

1. **Resource Allocation:**
   * Monitor VM resource usage in VirtualBox
   * Adjust RAM allocation based on usage patterns
   * Use thin-provisioned disks to save host storage
2. **Network Optimization:**
   * Use NAT Network instead of individual NAT adapters
   * Configure appropriate MTU sizes (1500 bytes default)
   * Consider paravirtualized network adapters for better performance
3. **Security Considerations:**
   * Take snapshots before major changes
   * Regularly update all VMs with security patches
   * Monitor logs for suspicious activities
   * Keep VMs isolated from production networks[[4]](#fn4)[[3]](#fn3)

**Advanced Configuration Options**

**Group Policy Configuration**

1. **Create Security-Focused GPOs:**

# Enable advanced auditing  
New-GPO -Name "Enhanced Security Auditing" -Comment "Enable detailed logging"  
  
# Configure password policies  
New-GPO -Name "Domain Password Policy" -Comment "Enforce strong passwords"

1. **Deploy Security Baselines:**
   * Download Microsoft Security Compliance Toolkit
   * Import security baselines for Windows 10 and Server 2019
   * Link GPOs to appropriate OUs[[18]](#fn18)

**Advanced Splunk Configuration**

1. **Install Splunk Apps:**

# Install Splunk App for Windows  
./splunk install app splunk-add-on-for-microsoft-windows\_8.8.0.tgz  
  
# Install Sysmon App  
./splunk install app splunk-add-on-for-sysmon\_3.2.2.tgz

1. **Configure Index Separation:**

# Create separate indexes for different data types  
./splunk add index windows  
./splunk add index sysmon  
./splunk add index security

This comprehensive Active Directory home lab provides an excellent foundation for learning enterprise Windows administration, security monitoring, and penetration testing in a controlled environment. The integration of Splunk SIEM capabilities enables hands-on experience with security operations center (SOC) workflows, while the inclusion of Kali Linux supports red team exercises and vulnerability assessments.[[1]](#fn1)[[2]](#fn2)[[3]](#fn3)[[19]](#fn19)

The lab environment closely mirrors real-world enterprise networks, making it ideal for developing practical cybersecurity skills, testing incident response procedures, and understanding the relationship between Active Directory infrastructure and security monitoring systems. Regular snapshots of clean VM states enable repeated testing scenarios and quick recovery from configuration changes.

⁂