let's run some targeted Nmap scans from your Xubuntu VM and observe the logs.

Below are the most commonly used by **attackers, penetration testers, and security researchers** during the reconnaissance phase of an operation.

# Scan 1: Basic SYN Scan (-sS -Pn)

This is a stealthy, half-open TCP scan. It's designed to provoke firewall drops without completing a full connection.

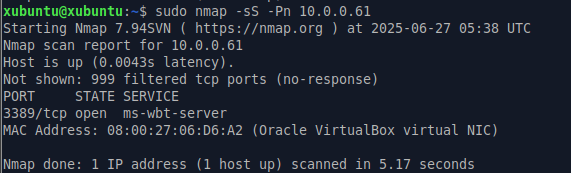
## Why attackers use it:

This is often the **first choice** for many attackers because it's considered "stealthy" or "half-open." It doesn't complete the full TCP 3-way handshake, making it less likely to be logged by the application layer services on the target system (though modern firewalls, like Windows Firewall when configured, will definitely log the initial SYN packet). It's fast and efficient.

From Xubuntu:

Sudo nmap -sS -Pn 10.0.0.61

(This will scan the top 1000 common ports by default.)



## Verify pfirewall.log (Windows VM):

**Expected Entries:** You will see many DROP TCP entries.

**action:** Look for DROP.

**protocol:** Look for TCP.

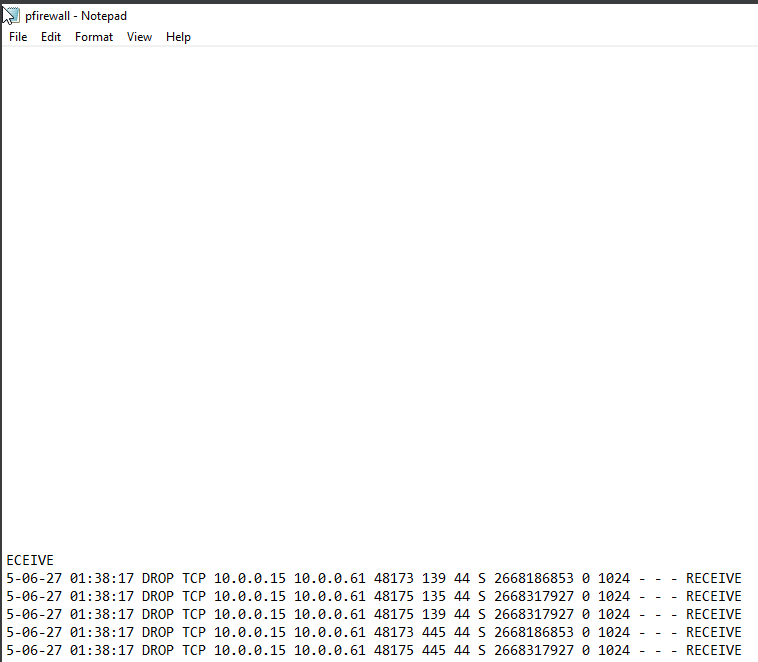
**src-ip:** Your Xubuntu VM's IP address. 10.0.0.15

**dst-ip:** Your Windows VM's IP address. 10.0.0.61

**tcpflags:** You *must* see the flag S (for SYN) in the tcpflags field. This indicates the incoming SYN packet was dropped.

**dst-port:** These will be the various ports Nmap scanned. 139, 135, 445.

**Timestamps:** Check that the date and time match when you ran the scan.

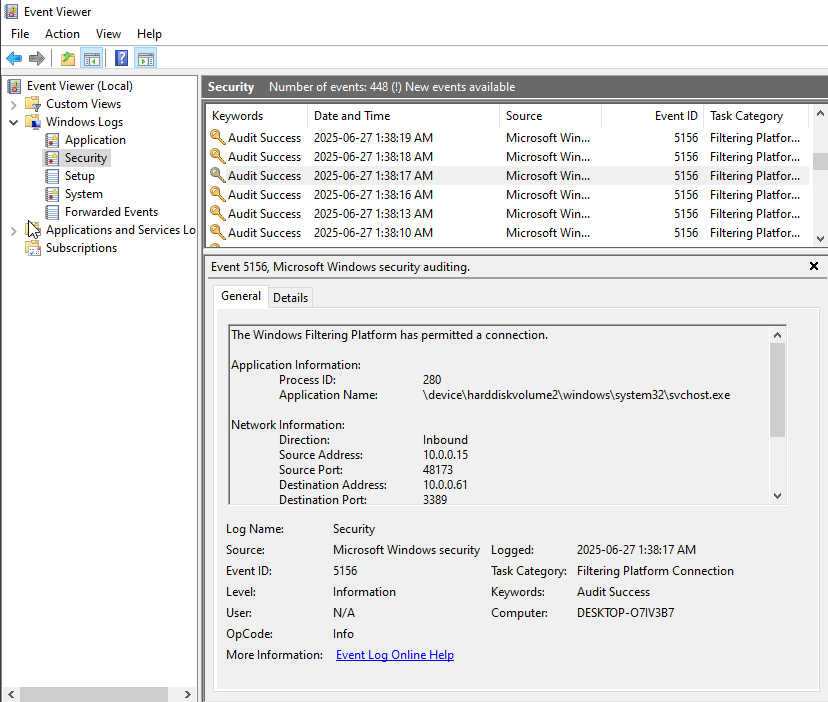


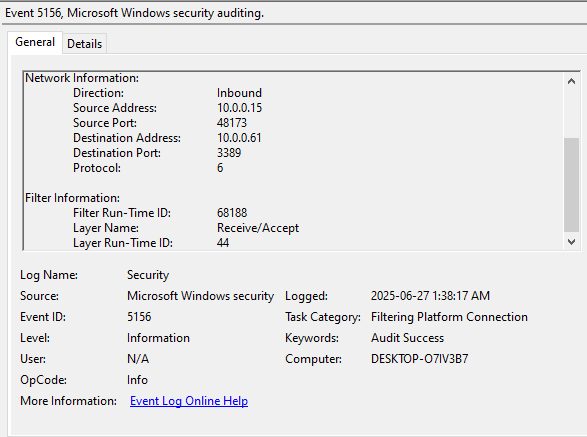
## Verify in Event Viewer (Security Log on Windows VM):

**Filter for Event IDs:** 5152 (for dropped packets) and potentially 5157 (for blocked connections). 5156 - **Nmap scanned an *open port* on Windows VM** for which the Windows Firewall has an **explicit "Allow" rule.**

**Content:** For each event, check the "General" and "Details" tabs.

* **Source Address:** Your Xubuntu VM's IP. 10.0.0.15
* **Destination Address:** Your Windows VM's IP. 10.0.0.61
* **Destination Port:** The port being scanned. 3389
* **Protocol:** TCP. Represented as 6
* **Process Information:** Often, you'll see "N/A" for the process name if the packet was dropped at the firewall without hitting an application. However, if an *allow* rule existed and the packet reached a process, that process might be listed. Process Information means Application information





# Scan 2: TCP Connect Scan (-sT -Pn)

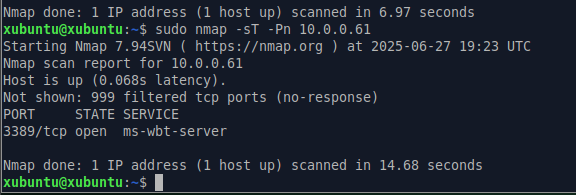
This performs a full TCP 3-way handshake (SYN, SYN-ACK, ACK). It's less stealthy.

## Why attackers use it:

While less stealthy than a SYN scan (as it completes the full TCP handshake), it's a reliable and straightforward scan. It's often used when an attacker doesn't have raw packet privileges (e.g., if they are not root/administrator on their attacking machine) or when SYN scans are being aggressively dropped and they want to ensure a connection. It's also effective against some older, less sophisticated firewalls.

From Xubuntu:

Sudo nmap -sT -Pn 10.0.0.61

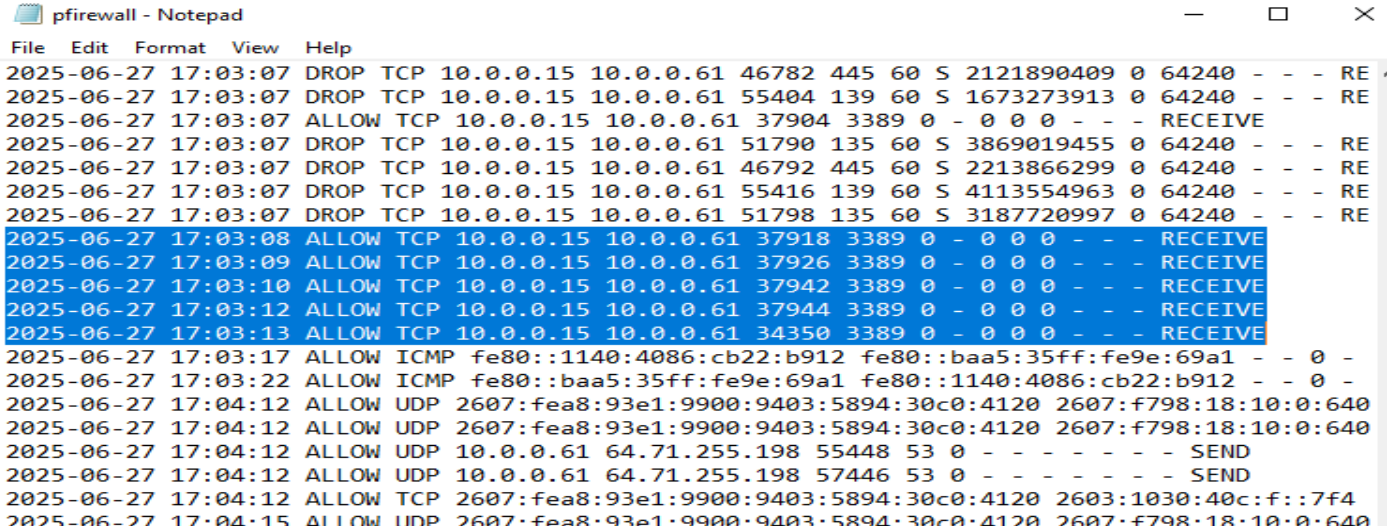


## Verify pfirewall.log (Windows VM):

**For Closed/Filtered Ports:** Still many DROP TCP entries with S (SYN flag), similar to the -sS scan.

**For *Open* Ports (if any, e.g., RDP 3389, SMB 445 if allowed):**

* You should see an **ALLOW TCP** entry for the initial SYN packet.



* You might also see an ALLOW TCP entry for the outgoing SYN-ACK packet (Windows responding).

Why the "missing" outgoing SYN-ACK log entry is acceptable (for your purpose):

* As we discussed, the pfirewall.log can sometimes be less granular about outgoing *responses* to inbound-initiated successful connections. The Event Viewer's Security Log (5156) also focuses on the inbound allowed connection. While ideal logging would show all three parts explicitly with clear directionality, the combination of Nmap's "open" report and the inbound ALLOW entries in both logs provides more than enough confirmation that the firewall successfully allowed the full connection to Port 3389.

## Verify in Event Viewer (Security Log on Windows VM):

* + **Filter for Event IDs:** 5152 (drops), 5157 (blocked connections), **5158 (allowed packets), 5159 (allowed connections)**.
  + **Content:**
    - For drops, similar to -sS.
    - For allowed connections (Event ID 5158/5159), you might see more detailed information, potentially including the **process** that accepted the connection (e.g., svchost.exe for RDP).

The absence of 5158/5159, while seeing 5156, simply means:

* The RDP service (or whatever service is on 3389) successfully bound to its port a while ago (likely at system startup or service start).
* Your firewall is actively logging the *actual network connections* (5156) as they occur.

What we do know and have confirmed:

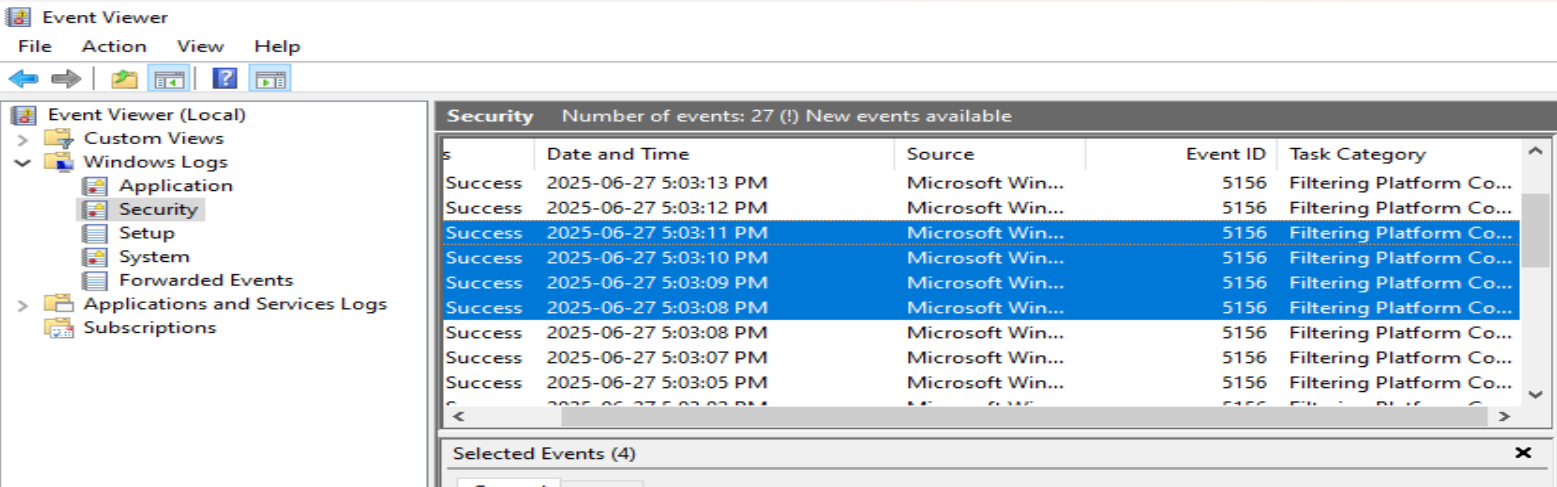
**Nmap reports 3389/tcp open ms-wbt-server:** This is the most crucial piece of evidence. Nmap only reports a TCP port as "open" after successfully completing the full TCP 3-way handshake (SYN, SYN-ACK, ACK). This **proves that the outgoing SYN-ACK packet was sent by your Windows VM and received by Nmap.** The connection literally could not have been established otherwise.

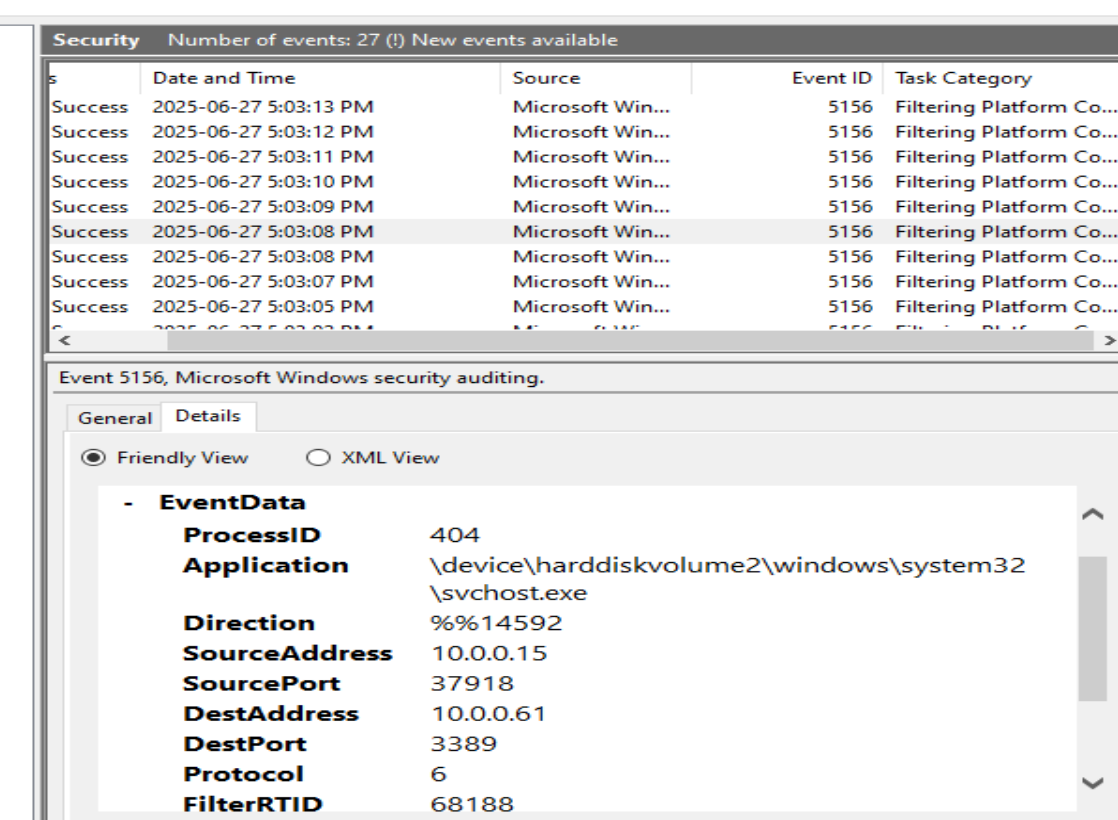
**Two ALLOW TCP entries in pfirewall.log (from Xubuntu to Windows VM):**

* These two entries correspond to the **incoming SYN packet** from Nmap and the **incoming ACK packet** from Nmap.
* They confirm that your Windows Firewall explicitly allowed these essential inbound components of the handshake to reach the RDP service.

**Event ID 5156 in the Security Log:**

* This event explicitly states "The Windows Filtering Platform has allowed a connection."
* This is a higher-level confirmation from Windows that the entire connection attempt was permitted by its firewall system.





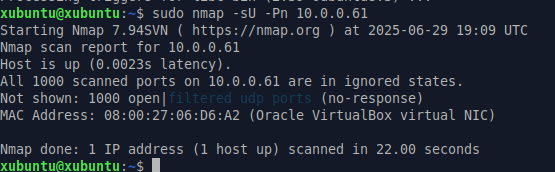
# Scan 3: UDP Scan (-sU -Pn)

## Why attackers use it :

Many critical services run over UDP (e.g., DNS (53), SNMP (161/162), DHCP (67/68), NTP (123)). Attackers often probe UDP ports because they are frequently overlooked or less rigorously secured compared to TCP ports. UDP scans can be slower and trickier to interpret but are essential for a full picture of a target's services.

When no UDP services are active on windows VM the scan will return the following result:

If the firewall silently drops the UDP packet (no response at all, no ICMP unreachable), Nmap marks it as **"open|filtered"** or **"filtered."**

****

Let’s have some UDP services actively running on windows VM prior to the scan.

**SNMP Agent:** You can enable the SNMP agent in Windows (Control Panel -> Programs and Features -> Turn Windows features on or off -> SNMP Simple Network Management Protocol). It typically listens on **UDP port 161**.

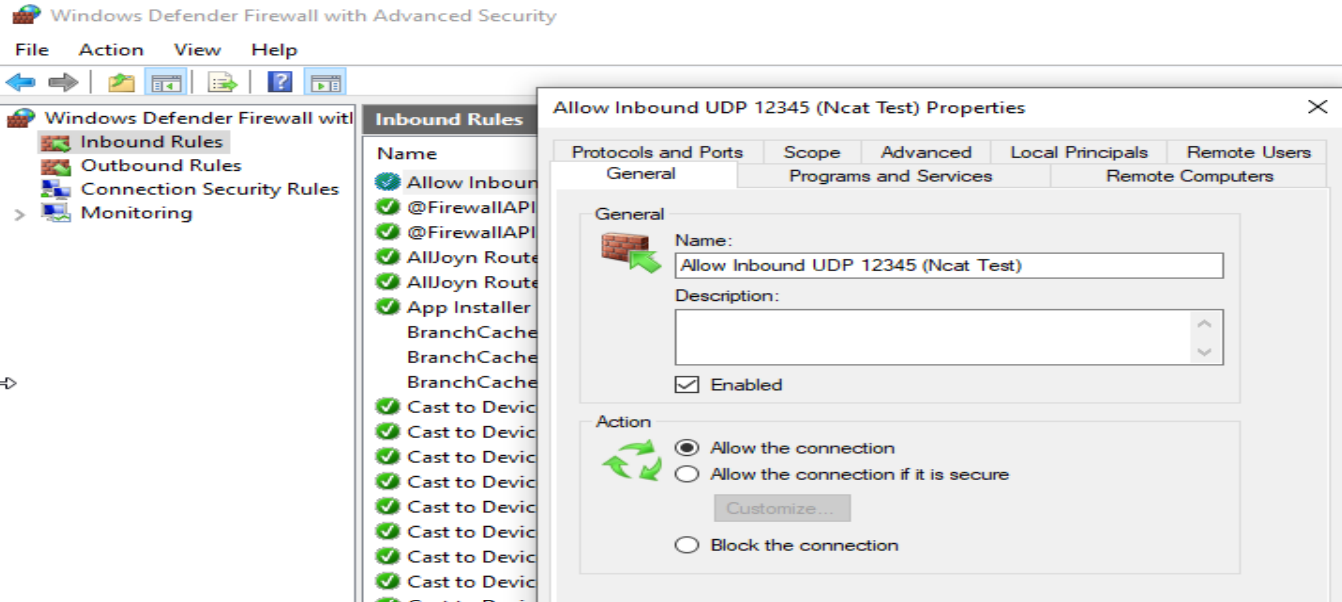
**NetCat (nc):** For a simple, temporary listener on any arbitrary UDP port (e.g., 12345), you can use NetCat.

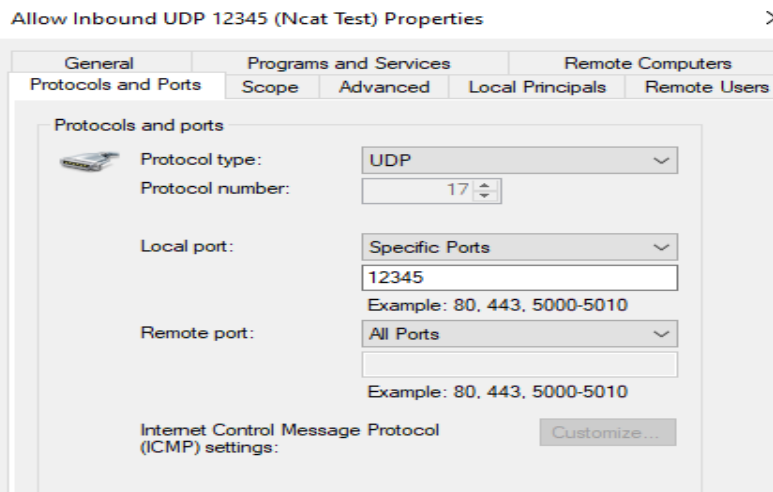
* Download a Windows version of NetCat.
* Open Command Prompt (as Administrator) in the directory where nc.exe is.
* Run: nc.exe -ulp 12345 (This will listen on UDP port 12345). Keep this window open during the scan.

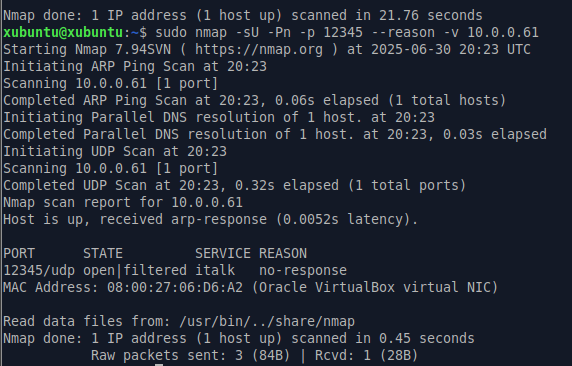


**Windows Firewall Rules for UDP:**

* **For the "open" UDP service:** You will need an **inbound "Allow" rule** in Windows Firewall for the specific UDP port you want Nmap to find as "open" (e.g., UDP 53 for DNS, UDP 161 for SNMP, or UDP 12345 for NetCat).
* **For other UDP ports:** The default Windows Firewall behavior is usually to block unallowed inbound connections. This is good, as it will allow you to see DROP UDP entries in your pfirewall.log for ports where nothing is listening or no allow rule exists.

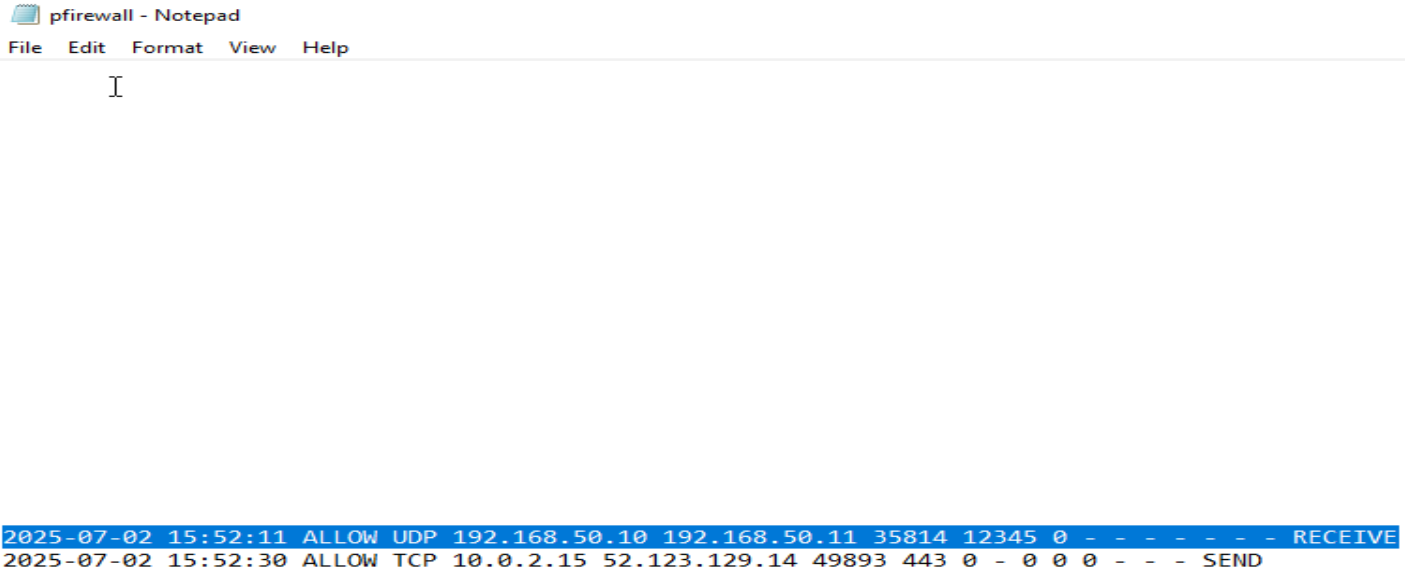






## What to Verify in pfirewall.log (Windows VM):

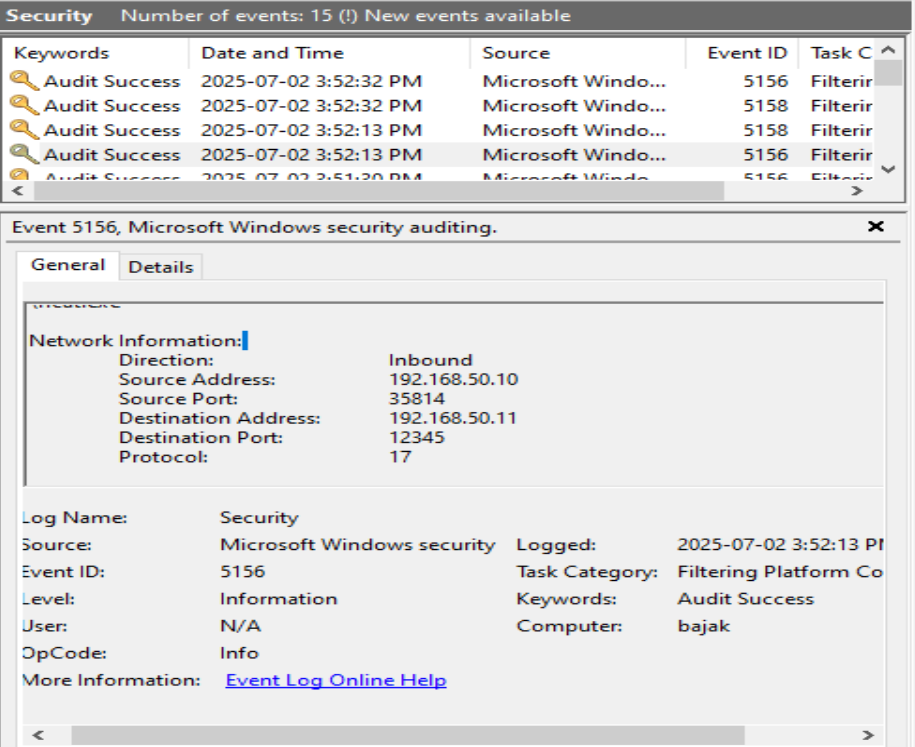
* **Expected Entries:** Allow UDP entries.
* **action:** Look for Allow.
* **protocol:** Look for UDP.
* **dst-port:** The UDP ports being scanned netcat: 12345



## What to Verify in Event Viewer (Security Log on Windows VM):

* **Filter for Event IDs:** 5156

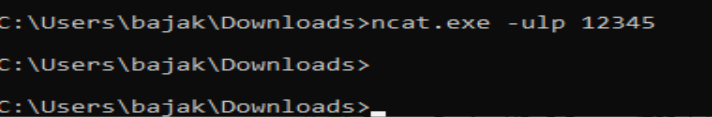
**The firewall received the UDP packet which means it permitted the UDP scan where the firewall is configured to allow the traffic.**



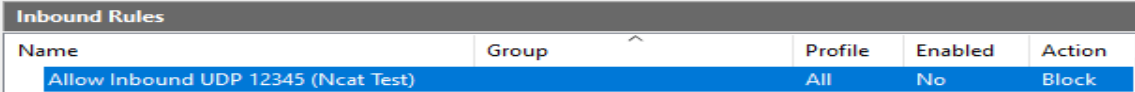
**NEGATIVE SCENARIO:**

Now let us make sure the following are in place on Windows VM:

* 1. NO ncat service running.

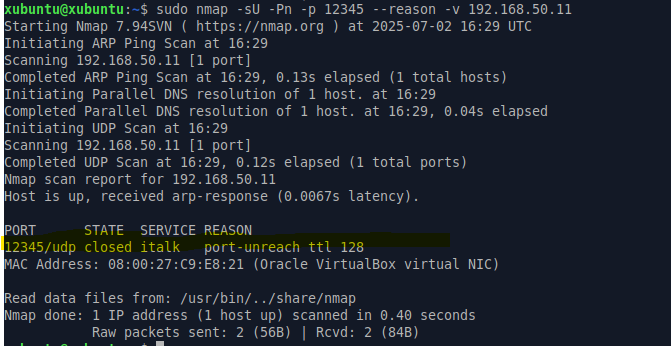


* 1. We should not have allow rule for UDP 12345 in inbound rules on windows firewall.



**Expected Result**: Nmap report 12345/udp closed.

This test has to be performed with windows firewall log set to OFF, otherwise we cannot see 12345/udp state as Closed in Nmap response.

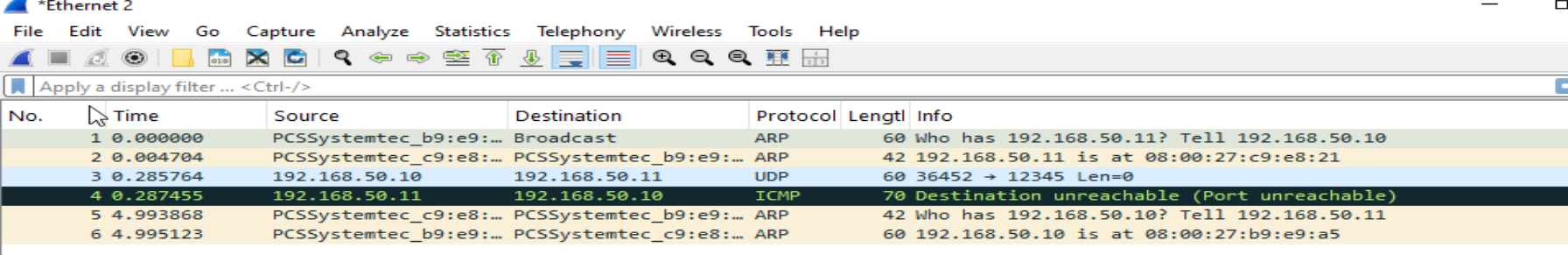


Though the outbound rule is set to capture the windows response with an ICMP type 3 code 3 destination unreachable the firewall was behaving quirky.

So, when I disabled the firewall on Windows VM, the OS was finally free to generate and transmit the ICMP type 3 and code 3 packet. Nmap saw that clear rejection and confidently marked the port as 12345/udp closed.

Firewall log and security log didn’t generate anything because this test (Nmap UDP Scan against a Windows VM machine with turning off the windows firewall).

The Wireshark was able to capture the windows response with an ICMP type 3 code 3 destination unreachable / port unreachable as seen in the screen shot below.



(Nmap UDP Scan against a Windows VM machine with turning ON the windows firewall).

Then pfirewall log should have DROP UDP entry and Security log should have Event IDs 5152.

# Scan 4: Enable Sysmon Event ID 3 Logging

**Goal: Detect full TCP connections (like connect scans) on the host**

⚠️ Sysmon will not log dropped or partial (stealth SYN) connections — only **established** connections.

**Sysmon -** System Monitor (Sysmon) is a Windows system service and device driver that, once installed on a system, remains resident across system reboots to monitor and log system activity to the Windows event log.  It provides detailed information about process creations, network connections, and changes to file creation time. By collecting the events it generates using [Windows Event Collection](https://msdn.microsoft.com/library/windows/desktop/bb427443(v=vs.85).aspx) or [SIEM](https://en.wikipedia.org/wiki/security_information_and_event_management) agents and subsequently analyzing them, you can identify malicious or anomalous activity and understand how intruders and malware operate on your network. The service runs as a [protected process](https://learn.microsoft.com/en-us/windows/win32/services/protecting-anti-malware-services-#system-protected-process), thus disallowing a wide range of user mode interactions.

**Sysmon download instructions**:

1. Download Sysinternals Suite from the link below.

<https://learn.microsoft.com/en-us/sysinternals/downloads/sysinternals-suite>

1. After download run the following commands as an administrator on command line.

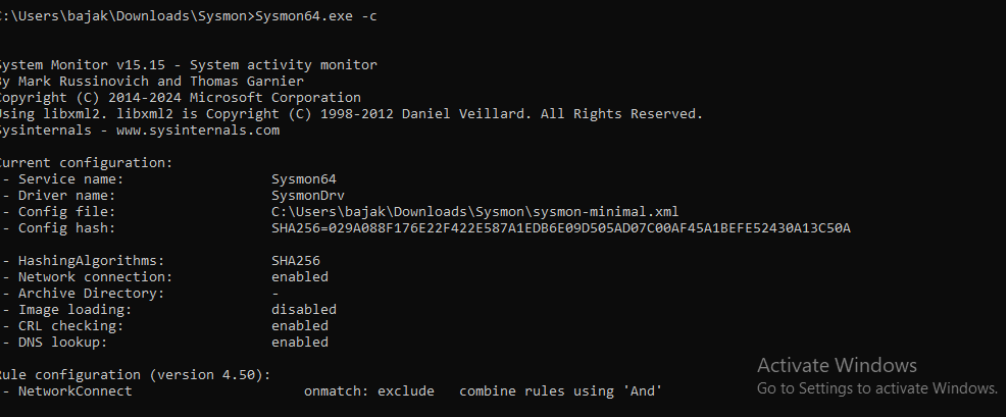
**Note**: you have to navigate to path where the sysinternals suite is downloaded and run the below command from that path.

sysmon64.exe -accepteula -i sysmonconfig-export.xml

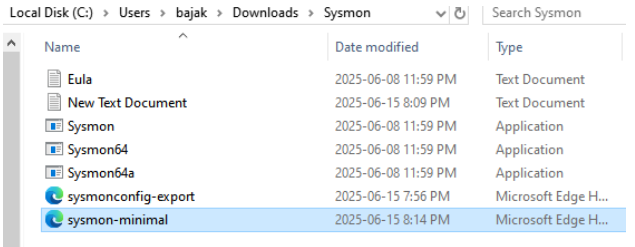
**The primary goal of sysmonconfig-export.xml** is to enable Sysmon to capture **high-quality, security-relevant events** from Windows systems, while simultaneously filtering out the immense volume of "noise" that would otherwise overwhelm event logs and SIEMs. It's designed to focus on activities that are most indicative of malicious behavior or key steps in an attack chain.

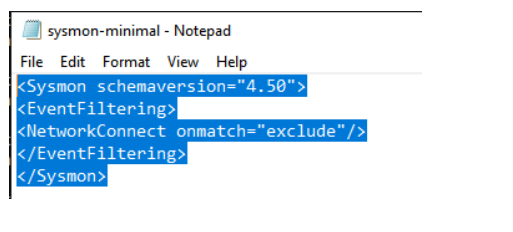
To verify the default configuration of Sysmon64 use the following command as seen in the screen shot below:

Sysmon64.exe -c



**Note**: the reason why you are seeing rule configuration for network connect is because, the Sysmon has been updated with below configuration file in order to capture network connection Event ID 3.





Command to update the active Sysmon configuration

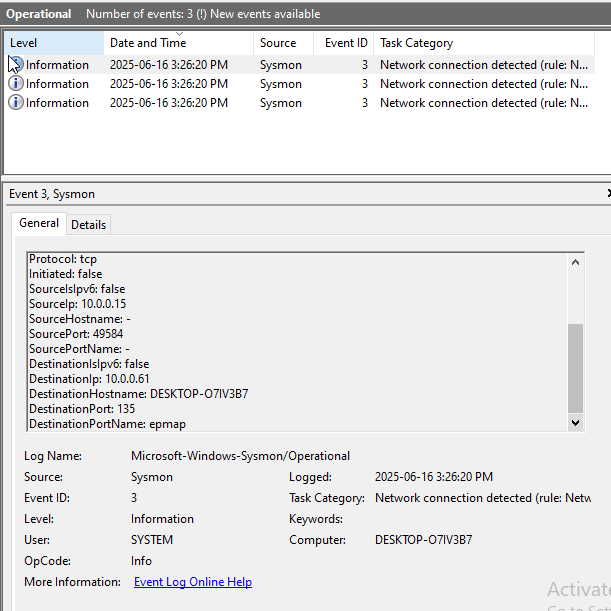
**sysmon.exe -c C:\Path\To\Your\minimal\_sysmon\_config.xml**

or

**C:\Users\bajak\Downloads\Sysmon>Sysmon.exe -c sysmon-minimal.xml**

When you scan a Windows VM (with Sysmon installed) from another machine (e.g., Xubuntu), the following should occur:

* Nmap sends SYN or full TCP packets
* Windows responds to those packets
* Sysmon (with correct config) logs Event ID 3 for inbound connections
* **As seen in the screen shot below, when we perform nmap scan of windows vm from xubuntu vm, as expected we see Sourceip: 10.0.0.15, Destinationip: 10.0.0.61**

****

**Below is the manual query to filter TCP protocol only events from the logs**

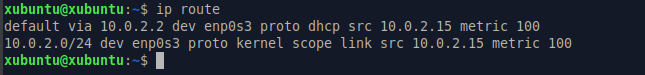
**<QueryList>  
  <Query Id="0" Path="Microsoft-Windows-Sysmon/Operational">  
    <Select Path="Microsoft-Windows-Sysmon/Operational">  
      \*[System[(EventID=3)]] and \*[EventData[Data[@Name='Protocol']='tcp']]  
    </Select>  
  </Query>  
</QueryList>**

# Dual Adapter setup on Virtual Machines:

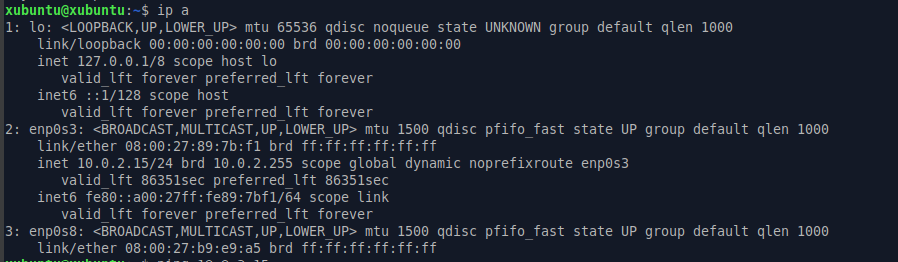
Xubuntu

Adapter 1: set to NAT (for internet access).

Adapter 2: set to Internal Network (VM-to-VM communication).



Default gateway is 10.0.2.2 VirtualBox NAT router. Traffic heading to internet leaves through enp0s3 which is NAT adapter.

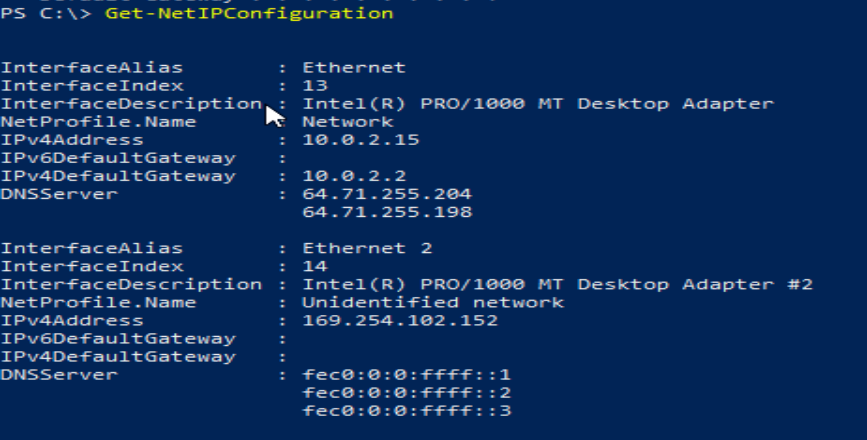


If enp0s8 doesn’t show any IP address means it’s not handling internet traffic. It’s likely just waiting there for VM-to-VM communication on the internal network.

Windows

Adapter 1: set to NAT (for internet access).

Adapter 2: set to Internal Network (VM-to-VM communication).



If you see in the above screen shot the windows assigned Automatic Private IP addressing (APIPA) Address.

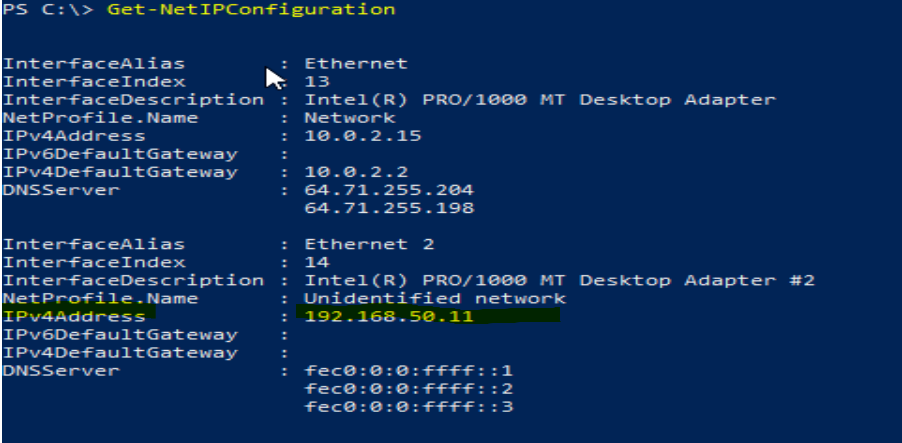
Windows assign this when the adapter is set to DHCP, but no DHCP server available on the network.

Since adapter 2 is connected to an internal network, and virtual box doesn’t provide DHCP for internal networks, windows fall back to this 169.254.x.x address. It’s a sign that the adapter is up, but didn’t get a usable IP.

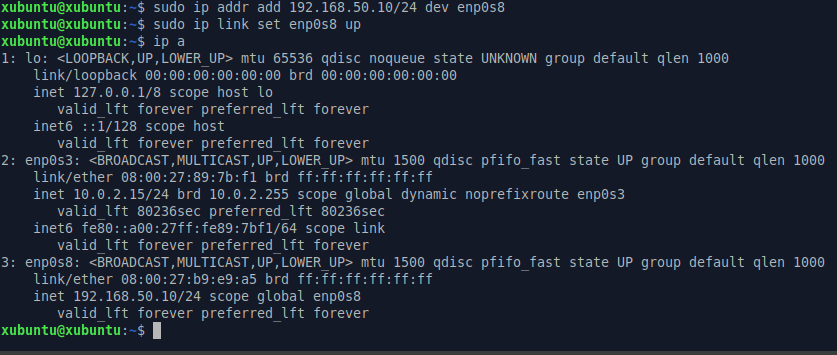
If we want Adapter 2 to be usable for VM-to-VM communication we have to Assign a static IP manually:

Go to network connections, right click adapter 2 properties set IP: 192.168.50.11, Subnet: 255.255.255.0 and gateway blank.

As seen in screen shot below able to see the static IP which was set manually.



Similarly let’s assign a static IP manually for adapter 2 on Xubuntu VM:

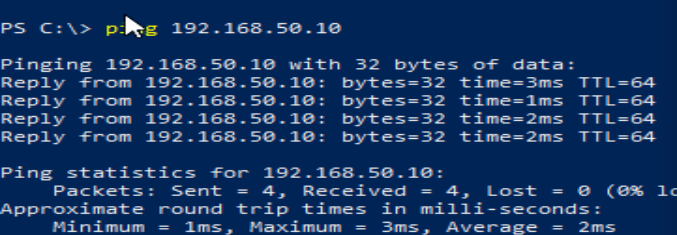
as seen in the screen shot above adapter 2 shows the static ip address which was set manually.

So, for both windows and Xubuntu VM in order to communicate one on one via internal network they have to use the following IP addresses:

Windows VM: 192.168.50.11

Xubuntu VM: 192.168.50.10

As seen below windows VM is able to communicate to Xubuntu VM on internal network.



As seen below Xubuntu VM is able to communicate to Windows VM on internal network.

