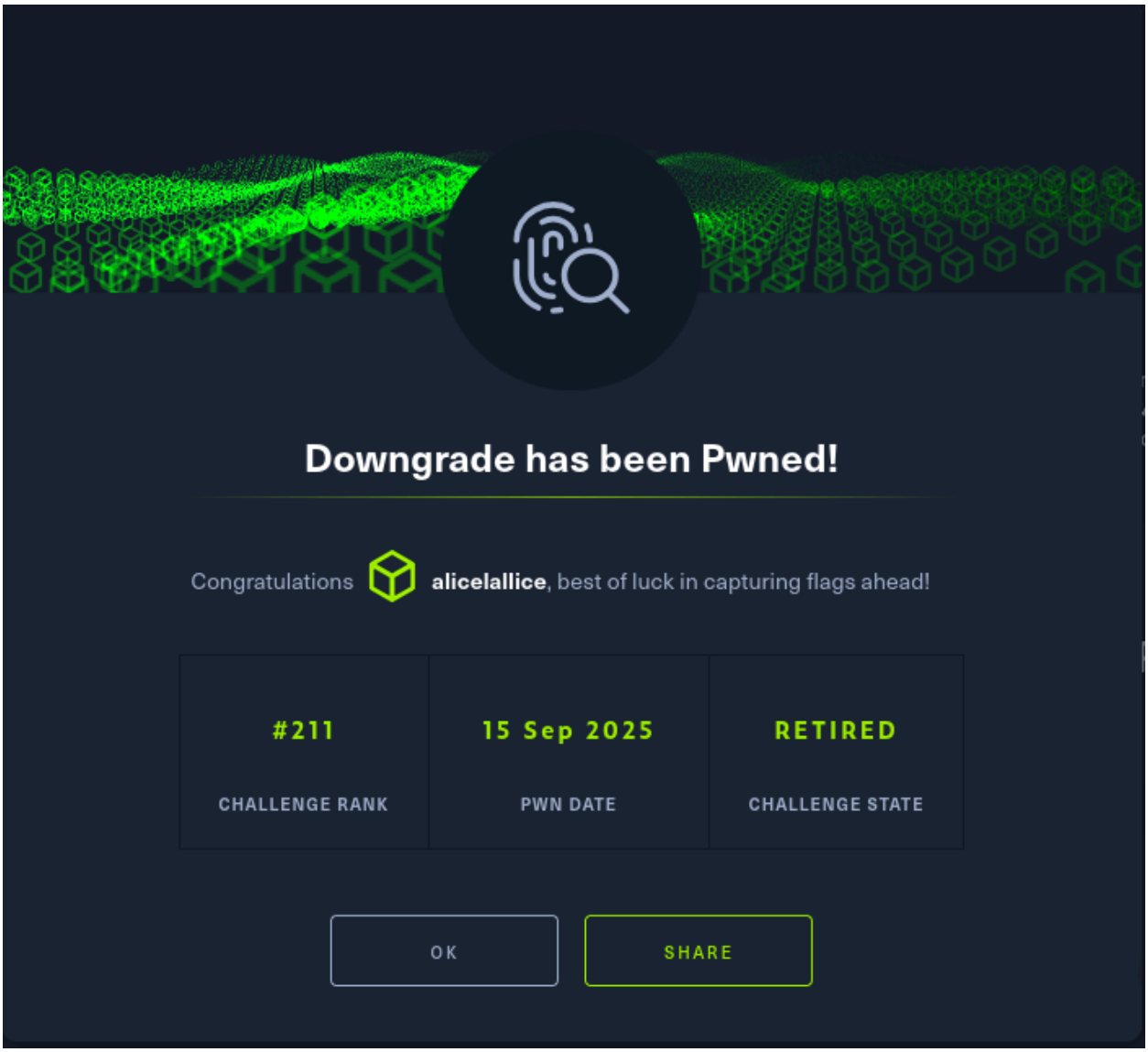
Downgrade

Types	forensic
CTF	HTB



Connect to the docker

```
┌──(kali㉿kali)-[~]
└─$ nc 94.237.57.115 47902

+-----------+-----------------------------------------------------------+
|   Title   |                        Description                        |
+-----------+-----------------------------------------------------------+
| Downgrade |          During recent auditing, we noticed that          |
|           |      network authentication is not forced upon remote     |
|           |        connections to our Windows 2012 server. That       |
|           |            led us to investigate our system for           |
|           |  suspicious logins further. Provided the server's event   |
|           |        logs, can you find any suspicious successful       |
|           |                          login?                           |
+-----------+-----------------------------------------------------------+
```

check what files are given

```
┌──(kali㉿kali)-[~/Desktop/htb/Logs]
└─$ ls
Application.evtx                                                    Microsoft-Windows-PrintService%4Admin.evtx
HardwareEvents.evtx                                                 Microsoft-Windows-RemoteDesktopServices-RdpCoreTS%4Admin.evtx
'Internet Explorer.evtx'                                            Microsoft-Windows-RemoteDesktopServices-RdpCoreTS%4Operational.evtx
'Key Management Service.evtx'                                       Microsoft-Windows-Resource-Exhaustion-Detector%4Operational.evtx
Microsoft-Windows-ApplicationResourceManagementSystem%4Operational.evtx  Microsoft-Windows-RestartManager%4Operational.evtx
Microsoft-Windows-AppModel-Runtime%4Admin.evtx                      Microsoft-Windows-Security-SPP-UX-Notifications%4ActionCenter.evtx
Microsoft-Windows-AppReadiness%4Admin.evtx                          Microsoft-Windows-ServerManager-DeploymentProvider%4Operational.evtx
Microsoft-Windows-AppReadiness%4Operational.evtx                    Microsoft-Windows-ServerManager-MgmtProvider%4Operational.evtx
Microsoft-Windows-AppXDeployment%4Operational.evtx                  Microsoft-Windows-ServerManager-MultiMachine%4Admin.evtx
Microsoft-Windows-AppXDeploymentServer%4Operational.evtx            Microsoft-Windows-ServerManager-MultiMachine%4Operational.evtx
Microsoft-Windows-AppXDeploymentServer%4Restricted.evtx             Microsoft-Windows-Shell-ConnectedAccountState%4ActionCenter.evtx
Microsoft-Windows-Bits-Client%4Operational.evtx                     Microsoft-Windows-Shell-Core%4ActionCenter.evtx
Microsoft-Windows-CodeIntegrity%4Operational.evtx                   Microsoft-Windows-Shell-Core%4Operational.evtx
Microsoft-Windows-Compat-Appraiser%4Operational.evtx                Microsoft-Windows-SmartCard-DeviceEnum%4Operational.evtx
Microsoft-Windows-CoreApplication%4Operational.evtx                 Microsoft-Windows-SmbClient%4Connectivity.evtx
Microsoft-Windows-Crypto-DPAPI%4BackUpKeySvc.evtx                   Microsoft-Windows-SMBClient%4Operational.evtx
Microsoft-Windows-Crypto-DPAPI%4Operational.evtx                    Microsoft-Windows-SmbClient%4Security.evtx
Microsoft-Windows-DataIntegrityScan%4Admin.evtx                     Microsoft-Windows-SMBServer%4Audit.evtx
Microsoft-Windows-DataIntegrityScan%4CrashRecovery.evtx             Microsoft-Windows-SMBServer%4Connectivity.evtx
Microsoft-Windows-DeviceSetupManager%4Admin.evtx                    Microsoft-Windows-SMBServer%4Operational.evtx
Microsoft-Windows-DeviceSetupManager%4Operational.evtx              Microsoft-Windows-SMBServer%4Security.evtx
Microsoft-Windows-Dhcp-Client%4Admin.evtx                           Microsoft-Windows-StorageSpaces-Driver%4Diagnostic.evtx
Microsoft-Windows-Dhcpv6-Client%4Admin.evtx                         Microsoft-Windows-StorageSpaces-Driver%4Operational.evtx
Microsoft-Windows-Diagnosis-DPS%4Operational.evtx                   Microsoft-Windows-StorageSpaces-ManagementAgent%4WHC.evtx
Microsoft-Windows-DSC%4Admin.evtx                                   Microsoft-Windows-Storage-Tiering%4Admin.evtx
Microsoft-Windows-DSC%4Operational.evtx                             Microsoft-Windows-TaskScheduler%4Maintenance.evtx
Microsoft-Windows-Forwarding%4Operational.evtx                      Microsoft-Windows-TaskScheduler%4Operational.evtx
Microsoft-Windows-GroupPolicy%4Operational.evtx                     Microsoft-Windows-TerminalServices-LocalSessionManager%4Admin.evtx
'Microsoft-Windows-HomeGroup Control Panel%4Operational.evtx'       Microsoft-Windows-TerminalServices-LocalSessionManager%4Operational.evtx
Microsoft-Windows-International%4Operational.evtx                   Microsoft-Windows-TerminalServices-Printers%4Admin.evtx
Microsoft-Windows-Iphlpsvc%4Operational.evtx                        Microsoft-Windows-TerminalServices-Printers%4Operational.evtx
Microsoft-Windows-Kernel-ApphelpCache%4Operational.evtx             Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Admin.evtx
Microsoft-Windows-Kernel-Boot%4Operational.evtx                     Microsoft-Windows-TerminalServices-RemoteConnectionManager%4Operational.evtx
Microsoft-Windows-Kernel-EventTracing%4Admin.evtx                   Microsoft-Windows-TWinUI%4Operational.evtx
Microsoft-Windows-Kernel-PnP%4Configuration.evtx                    Microsoft-Windows-TZSync%4Operational.evtx
Microsoft-Windows-Kernel-PnPConfig%4Configuration.evtx              Microsoft-Windows-UAC%4Operational.evtx
Microsoft-Windows-Kernel-Power%4Thermal-Operational.evtx            Microsoft-Windows-User-Loader%4Operational.evtx
Microsoft-Windows-Kernel-ShimEngine%4Operational.evtx               Microsoft-Windows-UserPnp%4ActionCenter.evtx
Microsoft-Windows-Kernel-StoreMgr%4Operational.evtx                 Microsoft-Windows-UserPnp%4DeviceInstall.evtx
```

> *Which event log contains information about logon and logoff events? (for example: Setup)*

```
                                                    Microsoft-Windows-WMI-Activity%4Operational.evtx
                                                    OpenSSH%4Admin.evtx
                                                    OpenSSH%4Operational.evtx
                                                    Security.evtx
                                                    Setup.evtx
                                                    System.evtx
                                                    'Windows PowerShell.evtx'
ion-FileDownloadManager%4Operational.evtx           WitnessClientAdmin.evtx
```

We focus on `Security.evtx` because Windows records logon/logoff/auth events in the **Security** event log
Answer : **Security**

> What is the event id for logs for a successful logon to a local computer? (for example: 1337)

Why Security.evtx?

Windows segregates event logs by type. The **Security** log stores authentication-related events (logon, logoff, account changes). Event IDs you should know:

- **4624** = successful logon
- **4625** = failed logon
- **4634** = logoff
- **4648**, **4672**, **4688** = related security events

So to find suspicious successful logons, search **4624** entries inside `Security.evtx`.

Answer : **4624**

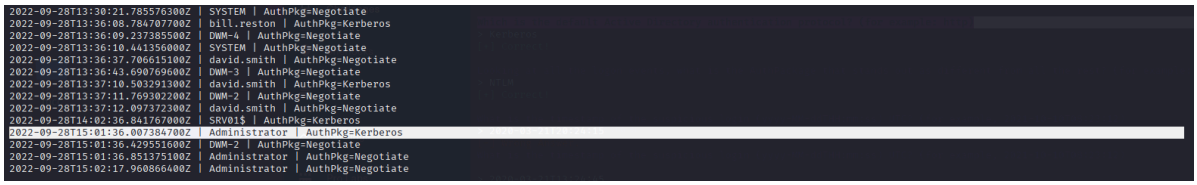
> *Which is the default Active Directory authentication protocol? (for example: http)*

```
┌──(kali㉿kali)-[~/Desktop/htb/Logs]
└─$ sed -i '1d' Security.xml

┌──(kali㉿kali)-[~/Desktop/htb/Logs]
└─$ sed -i '1i <Events xmlns="http://schemas.microsoft.com/win/2004/08/events/event">' Security.xml
echo '</Events>' >> Security.xml

┌──(kali㉿kali)-[~/Desktop/htb/Logs]
└─$ xmlstarlet sel -N ev="http://schemas.microsoft.com/win/2004/08/events/event" \
-t \
-m '//ev:Event[ev:System/ev:EventID="4624"]' \
-v 'concat(ev:System/ev:TimeCreated/@SystemTime, " | ", ev:EventData/ev:Data[@Name="TargetUserName"], " | AuthPkg=", ev:EventData/ev:Data[@Name="AuthenticationPackageName"])' -n \
Security.xml
```

Explanation:

- `N ev=...` registers the XML namespace used in EVTX-exported XML (required for XPath).
- `m` matches `Event` nodes whose `EventID` is `4624`.

- `v 'concat(...)'` prints a formatted line combining desired fields



So if you search your `Security.evtx` for **4624 events** and then check the value of `AuthenticationPackageName`, you'll often see **Kerberos** unless the system fell back to NTLM

Why Kerberos is the answer

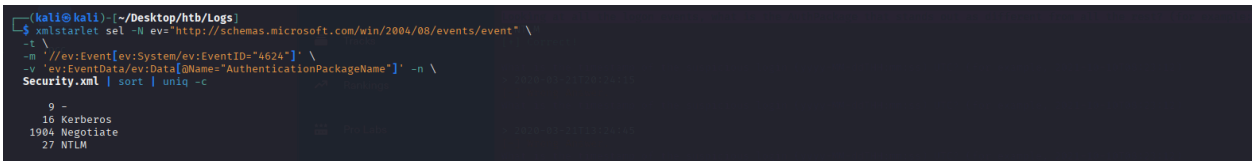
- Kerberos is faster and more secure (ticket-based, symmetric encryption).
- It supports mutual authentication (client ↔ server).
- It's required for features like single sign-on (SSO).
- That's why Microsoft made it the default.

> *Looking at all the logon events, what is the AuthPackage that stands out as different from all the rest? (for example: http)*

Look for the odd AuthenticationPackage

Most AD domain logons should show `Kerberos`. The challenge hint was a downgrade: network authentication not forced, so find where the logon used **NTLM** instead of Kerberos.

```
xmlstarlet sel -t \
  -m '//Event[System/EventID=4624]' \
  -v 'EventData/Data[@Name="AuthenticationPackageName"]' -n \
  Security.xml | sort | uniq -c
```



Identify the suspicious one

- The **majority** will be `Kerberos` (the default for Active Directory).
- If see **NTLM** (or anything else, e.g. `MSV1_0`), that's the **odd one out** and the answer.

Answer : **NTLM**

> *What is the timestamp of the suspicious login (yyyy-MM-ddTHH:mm:ss) UTC? (for example, 2021-10-10T08:23:12)*

Extract all successful logons (4624) with timestamp + user + IP + AuthPkg

```
xmlstarlet sel -t \
  -m '//Event[System/EventID=4624]' \
  -v 'concat(System/TimeCreated/@SystemTime, " | User=", EventData/Data
[@Name="TargetUserName"], " | Ip=", EventData/Data[@Name="IpAddress"],
" | AuthPkg=", EventData/Data[@Name="AuthenticationPackageName"])' -n \
  Security.xml
```

Most NTLM events are for `vagrant` or `ANONYMOUS LOGON`. The one that stands out is the **Administrator** NTLM logon at:

Answer: 2022-09-28T13:10:57 UTC

Why this one:

- It's the only NTLM 4624 entry for **Administrator** (others are `vagrant`/anonymous).
- A domain **Administrator** authenticating with **NTLM** (instead of Kerberos) is unusual and therefore suspicious.

Why the Administrator NTLM entry is suspicious

- `Administrator` is a high-privilege account — any deviation is notable.
- AD default is **Kerberos**; seeing a domain **Administrator** authenticate with **NTLM** suggests a downgrade or fallback to an older protocol — possibly due to misconfiguration or an attack that forced NTLM.
- The challenge premise explicitly mentioned "network authentication is not forced," hinting NTLM usage.

So the event to pick is the NTLM 4624 for `Administrator`

Final answer and flag

- **Event log:** `Security`

- **Event ID for successful logon:** `4624`
- **Default AD auth protocol:** `Kerberos`
- **AuthPackage that stands out:** `NTLM`
- **Suspicious timestamp (UTC):** `2022-09-28T13:10:57`
- **Flag:** `HTB{34sy_t0_d0_4nd_34asy_t0_d3t3ct}`

Teaching tips — what to emphasize to learners

1. **Know the log types:** Security is the primary log for authentication.
2. **Learn key Event IDs:** 4624/4625/4634 are fundamentals.
3. **Understand fields:** `LogonType`, `IpAddress`, `AuthenticationPackageName` often reveal remote vs local and Kerberos vs NTLM.
4. **Use namespaces in XML queries:** EVTX-exported XML uses a namespace; forgetting to include it in `xmlstarlet` will return no results.
5. **Triangulate suspiciousness:** Look at authentication method, account name, logon type, and IP. The combination indicates abnormal activity.
6. **Practice on real EVTXs:** many CTFs provide log sets; run these exact commands to build muscle memory.

Remediation & Notes (real-world)

- **Enforce network-level authentication (NLA)** for RDP to prefer Kerberos and prevent certain downgrade attacks.
- **Disable NTLM where possible** or audit and control which systems/services require it.
- **Monitor and alert on NTLM usage** for privileged accounts.
- **Harden administrator accounts** (PSM, privileged access workstations) to reduce remote exposure.