wazuh.

SIEM

Detecting and Blocking RDP Brute Force Attacks with Wazuh

Created By: Amir Raza

Follow Me: www.linkedin.com/in/amirsoc

Wazuh Active Response: Detecting and Blocking RDP Brute Force Attacks

What is RDP (Remote Desktop Protocol)?

Remote Desktop Protocol (RDP) is a Microsoft technology that allows you to connect to and control a Windows computer over a network or the internet. When using RDP, you can view the remote desktop, run applications, transfer files, and manage the system just as if you were sitting in front of it.
RDP is extremely useful for IT administrators, helpdesk teams, and remote workers. However, because it provides direct access to a system, it can also be a target for attackers if it is not properly secured.

What is an RDP Brute Force Attack?

An RDP brute force attack occurs when an attacker repeatedly tries different username and password combinations in an attempt to log into a Windows machine's remote desktop. This is typically done with automated tools that can attempt thousands of logins in a short period.
If successful, the attacker gains full remote control over the system, allowing them to steal data, deploy malware, or use the system to attack others.
Since RDP is widely deployed, it has become a common entry point for cybercriminals.

Wazuh Active Response for RDP Brute Force Protection

Wazuh is an open-source security platform that can automatically detect and respond to RDP brute force attacks. It continuously monitors Windows Security logs for repeated failed RDP login attempts from the same IP within a short time. When such suspicious activity is detected, Wazuh triggers its Active Response mechanism.

Active Response executes a predefined action — typically using the Windows netsh advfirewall command — to block the attacker's IP address through the Windows Firewall. This block can be temporary (e.g., 24 hours) to avoid locking out legitimate users. The process is fully automated, occurs in real time, and all actions are logged for auditing and compliance purposes, effectively minimizing the attacker's opportunity to compromise the system.

Wazuh Implementation Flow for RDP Brute Force Detection & Blocking

1. **Remote Desktop Protocol (RDP) Login Attempts Generate Logs**
   When any user or attacker attempts to log in to a Windows machine via RDP, Windows Security Event Logs record these login activities. Each login attempt, whether successful or failed, generates an event. Specifically, failed RDP login attempts create events with Event ID 4625 that include details such as the username tried, source IP address, timestamp, and logon type (logon type 10 usually indicates an RDP remote session). These logs are crucial data points that form the basis for detecting brute force attacks.

2. **Wazuh Agent Monitors Windows Security Event Logs Locally**
   The Wazuh Agent installed on the Windows endpoint continuously monitors these Security Event Logs in real time, parsing the entries related to authentication, including failed logon attempts. The agent uses decoders to interpret Windows Event Channel log format and extract structured information such as event ID, logon type, source IP, and timestamps. This enables the agent to detect suspicious events locally as they occur.

3. **Agent Sends Collected Log Data Securely to Wazuh Manager**
   Once the agent parses and structures the log data, it transmits these event details securely over the network to the central Wazuh Manager. The manager aggregates data from multiple agents for centralized correlation, alerting, and action. This communication typically uses secured protocols and authentication to maintain data integrity and confidentiality.

4. **Wazuh Manager Applies Detection Rules to Incoming Log Data**
   The manager analyzes the incoming event data according to predefined and custom detection rules. These rules filter for specific events, such as failed RDP login attempts with event ID 4625 and logon type 10 (remote interactive logon). The base detection rule flags each individual failed attempt by generating an alert entry including the attacker's IP, user targeted, and time of failure.

5. **Correlation of Multiple Failed Attempts Triggers Brute Force Alert**
   To avoid false positives from isolated failures, Wazuh employs correlation rules that aggregate repeated alerts. For example, if three or more failed RDP login attempts occur from the same IP address within a 2-minute window, this

correlation triggers a high-level brute force alert. This alert indicates a strong probability that a brute force attack is in progress rather than benign login errors.

6. **Alert Generation and Display on Wazuh Dashboard**
   After triggering the brute force detection rule, the Wazuh Manager generates an alert that is visible on the Wazuh Dashboard. Security analysts and administrators can view this alert with detailed information—such as the source IP, number of failed attempts, time window, affected user account, and severity level. The dashboard provides filtering and search capabilities for real-time monitoring.

7. **Active Response Mechanism Automatically Blocks the Attacker's IP**
   Once the brute force alert is detected (triggering the specific correlation rule ID), Wazuh executes the Active Response mechanism automatically. This system runs a predefined command locally on the Windows Agent, for instance, Windows Firewall command-line tool `netsh advfirewall` to add a firewall rule blocking inbound connections from the attacker's IP address. This block prevents further login attempts from that IP at the network level, stopping the attack effectively and quickly.

8. **Blocking Duration and Automatic Unblocking**
   The IP block imposed by the firewall can be configured to last for a specific duration, commonly 24 hours. After this timeout, the Active Response mechanism automatically removes the block to avoid permanent lockouts and reduce administrative overhead. This behavior is configured in Wazuh's active response settings.

9. **Logging of Active Response Actions for Auditing**
   All automatic blocking and unblocking actions are logged transparently in the agent's `active-responses.log` file and also reported to the manager. This provides an audit trail to verify which IPs were blocked, at what time, and under what conditions, supporting compliance and forensic investigations.

10. **Continuous Monitoring and Fine-Tuning**
    Organizations should continuously monitor the Wazuh Dashboard and logs to review alerts and blocked IPs. Administrators may tune detection thresholds, frequency values, and time windows to optimize detection accuracy and minimize false positives. Regular updates to Wazuh rules and agents ensure protection against evolving attack patterns

Implementation Steps for Wazuh RDP Brute Force Detection and Active Response

My Wazuh Manager is currently running on Kali Linux and is in a healthy, active status.
sudo systemctl status wazuh-manager

```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl status wazuh-manager
[sudo] password for kali:
● wazuh-manager.service - Wazuh manager
     Loaded: loaded (/usr/lib/systemd/system/wazuh-manager.service; enabled; pr>
     Active: active (running) since Thu 2025-08-14 21:48:16 PKT; 9min ago
 Invocation: 222a37b9f82d451196643d83c54534ac
    Process: 1169 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (co>
      Tasks: 178 (limit: 5614)
     Memory: 873.7M (peak: 1.3G)
        CPU: 1min 312ms
     CGroup: /system.slice/wazuh-manager.service
             ├─1566 /var/ossec/framework/python/bin/python3 /var/ossec/api/scri>
             ├─1567 /var/ossec/framework/python/bin/python3 /var/ossec/api/scri>
             ├─1568 /var/ossec/framework/python/bin/python3 /var/ossec/api/scri>
             ├─1571 /var/ossec/framework/python/bin/python3 /var/ossec/api/scri>
             ├─1574 /var/ossec/framework/python/bin/python3 /var/ossec/api/scri>
             ├─1600 /var/ossec/bin/wazuh-integratord
             ├─1624 /var/ossec/bin/wazuh-authd
             ├─1636 /var/ossec/bin/wazuh-db
             ├─1710 /var/ossec/bin/wazuh-execd
             ├─1747 /var/ossec/bin/wazuh-maild
             ├─1758 /var/ossec/bin/wazuh-analysisd
             ├─1786 /var/ossec/bin/wazuh-syscheckd
             ├─1857 /var/ossec/bin/wazuh-remoted
             ├─1926 /var/ossec/bin/wazuh-logcollector
```

Step 1: Prepare Detection Rules

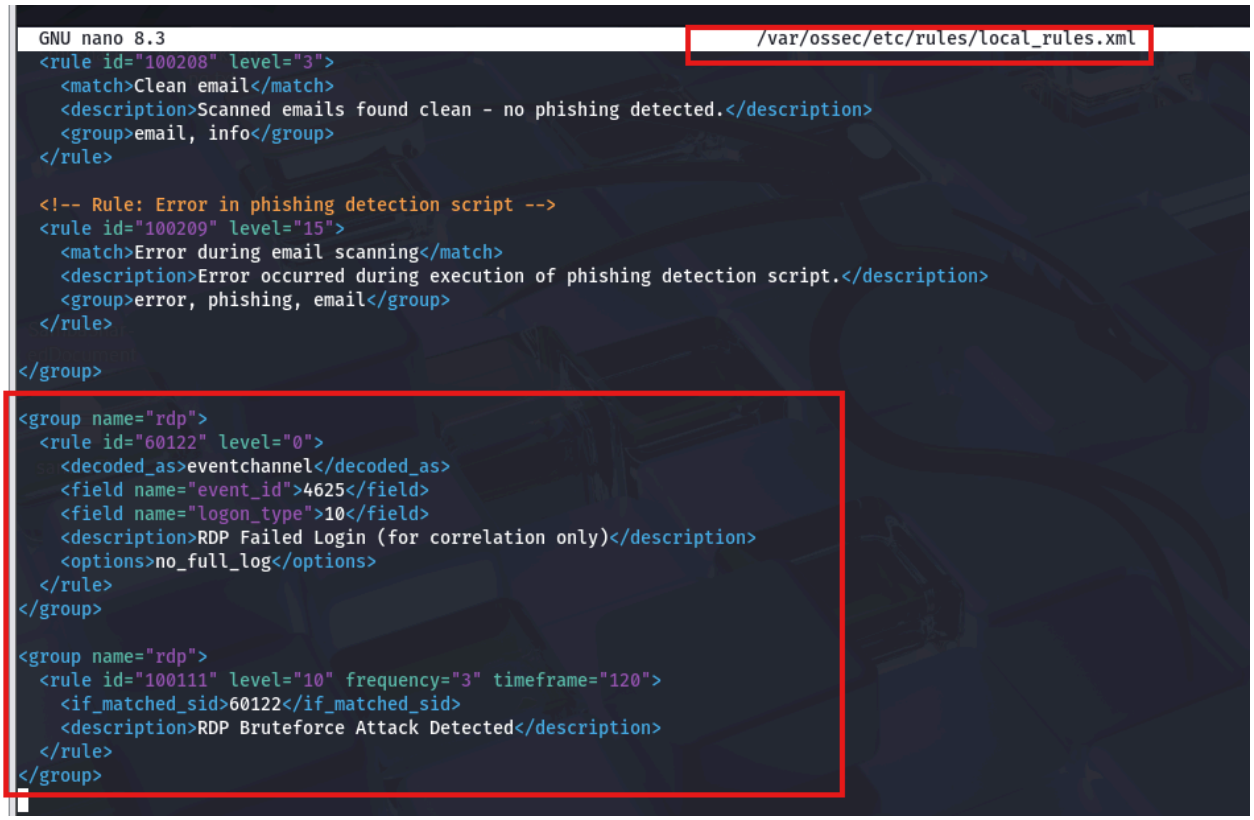
Now we will edit the `local_rules.xml` file to add custom rules for detecting and blocking RDP brute force attempts.The `local_rules.xml` file lets us create custom detection rules for our needs. Here, we add rules to spot RDP brute force attacks — one to log each failed RDP login, and another to alert only if the same IP fails three times within two minutes. This helps catch real attacks while avoiding false alarms from single mistakes.
sudo nano /var/ossec/etc/rules/local_rules.xml

Add the following snippets to the local_rules.xml file on your Wazuh manager or agent:

```
<group name="rdp">
  <rule id="60122" level="0">
```

```
    <decoded_as>eventchannel</decoded_as>
    <field name="event_id">4625</field>
    <field name="logon_type">10</field>
    <description>RDP Failed Login (for correlation only)</description>
    <options>no_full_log</options>
  </rule>
</group>

<group name="rdp">
  <rule id="100111" level="10" frequency="3" timeframe="120">
    <if_matched_sid>60122</if_matched_sid>
    <description>RDP Bruteforce Attack Detected</description>
  </rule>
</group>
```



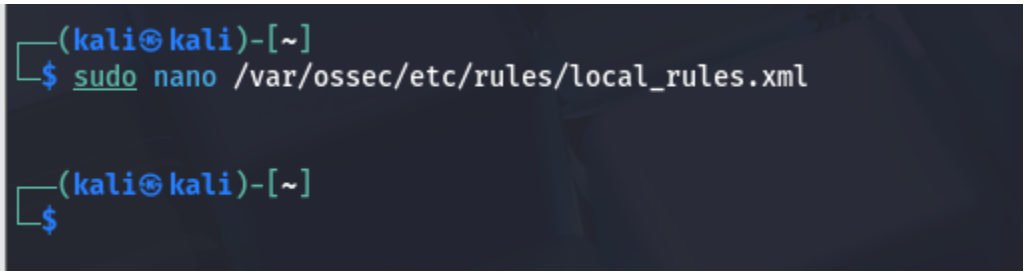
Explanation:

Rule ID 60122 is designed to detect every failed RDP login attempt by matching Windows Security event ID 4625 with logon type 10, which specifically represents remote interactive logons over RDP.

This rule runs silently in the background and does not display alerts in the Wazuh dashboard. The purpose of this is to avoid alerting on single failed logins that may simply be user mistakes, such as typing the wrong password once.
Instead of alerting directly, rule 60122 stores these failed login events so they can be analyzed later by another rule.
Rule ID 100111 acts as a correlation rule. It continuously monitors the events captured by rule 60122 and looks for suspicious patterns.
If the same IP address triggers three failed login attempts within a two-minute window, rule 100111 recognizes this as a possible brute force attack and immediately generates a high-level alert.
This two-stage detection approach ensures that occasional, harmless login errors do not trigger alerts, but repeated and rapid failures are flagged as malicious.
By separating detection (rule 60122) from correlation and alerting (rule 100111), the configuration keeps the Wazuh dashboard clean, reduces false positives, and gives security teams only the alerts that matter most.
This workflow improves efficiency by focusing attention on genuine threats while maintaining a record of all failed login attempts for auditing or investigation purposes.

To add the rules, open the **`local_rules.xml`** file with:

sudo nano /var/ossec/etc/rules/local_rules.xml



Paste your new rules into the file, save the changes (**CTRL + O**, **ENTER**), and exit (**CTRL + X**).

Once the rules are added, restart the Wazuh Manager to check if they are valid:

sudo systemctl restart wazuh-manager

If the restart completes without errors, your rules are valid and applied successfully. If the service fails to restart and returns an error, it means there's a syntax or formatting issue in the XML file that must be corrected before Wazuh can load the rules

```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl restart wazuh-manager


┌──(kali㉿kali)-[~]
└─$
```

So there is no error in my rules.

Step 2: Configure the Active Response Command on Wazuh Manager

sudo nano /var/ossec/etc/ossec.conf

```
┌──(kali㉿kali)-[~]
└─$ sudo nano /var/ossec/etc/ossec.conf

[sudo] password for kali:

┌──(kali㉿kali)-[~]
└─$
```

Define the blocking command using Windows native netsh.exe tool to manage firewall rules:

```
<command>
  <name>netsh</name>
  <executable>netsh.exe</executable>
  <timeout_allowed>yes</timeout_allowed>
</command>
```

This block defines the **`netsh` command** that Wazuh will use to block IP addresses on Windows agents. By specifying `<executable>netsh.exe</executable>`, Wazuh knows which program to run. `<timeout_allowed>yes</timeout_allowed>` ensures that the command can execute with a set duration (like temporary 24-hour blocks). This command itself does not block anything yet; it just registers the tool with Wazuh so it can be used later.

Step 3: Configure Active Response Trigger

In the ossec.conf file on the Wazuh manager (usually /var/ossec/etc/ossec.conf), add:

```
<active-response>
  <disabled>no</disabled>
  <command>netsh</command>
  <location>local</location>          <!-- Runs on the affected Windows agent -->
  <rules_id>100111</rules_id>          <!-- Only trigger on correlated brute force alerts -->
  <timeout>86400</timeout>             <!-- Block for 24 hours (86400 seconds) -->
</active-response>
```

Explanation:

This block tells Wazuh **when and how** to execute the `netsh` command.

`<disabled>no</disabled>` activates the response.
`<command>netsh</command>` links this trigger to the command defined above.
`<location>local</location>` ensures the command runs on the affected Windows agent, where the firewall rules actually exist.
`<rules_id>100111</rules_id>` ensures that the command only runs when the RDP brute force correlation rule is triggered, preventing unnecessary firewall blocks.
`<timeout>86400</timeout>` blocks the attacker's IP for 24 hours (86400 seconds) and then automatically removes the block, ensuring legitimate users aren't permanently locked out.

Workflow:

Wazuh detects repeated failed RDP login attempts (Rule 100111).
The active response is triggered.
Wazuh executes `netsh.exe` on the Windows agent to block the attacking IP.
The IP remains blocked for 24 hours and is automatically removed afterward.
All actions are logged in Wazuh for auditing and monitoring.

Save the configuration and restart the manager.

sudo systemctl restart wazuh-manager

```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl restart wazuh-manager

┌──(kali㉿kali)-[~]
└─$
```

Step 4: Configure the Wazuh Agent on the Windows Endpoint

Edit the agent's ossec.conf file (commonly at C:\Program Files (x86)\ossec-agent\ossec.conf).
Enable active response and specify the rule:
Run the following command on command prompt and open the command prompt by admin rights.
cd "C:\Program Files (x86)\ossec-agent"

```
C:\Users\administrator>cd C:\Program Files (x86)\ossec-agent

C:\Program Files (x86)\ossec-agent>notepad ossec.conf

C:\Program Files (x86)\ossec-agent>_
```

notepad ossec.conf

```
<active-response>
  <disabled>no</disabled>
  <command>netsh</command>
  <rules_id>100111</rules_id>
  <timeout>86400</timeout>
</active-response>
<localfile>
  <location>Security</location>
  <log_format>eventchannel</log_format>
</localfile>
```

```
*ossec - Notepad
File  Edit  Format  View  Help
    <disabled>yes</disabled>
    <run_daemon>yes</run_daemon>
    <bin_path>C:\Program Files\osquery\osqueryd</bin_path>
    <log_path>C:\Program Files\osquery\log\osqueryd.results.log</log_path>
    <config_path>C:\Program Files\osquery\osquery.conf</config_path>
    <add_labels>yes</add_labels>
  </wodle>
  <!-- Active Response for brute force attack -->
  <active-response>
    <disabled>no</disabled>
    <command>netsh</command>
    <rules_id>100111</rules_id>
    <timeout>86400</timeout>
  </active-response>
  <localfile>
    <location>Security</location>
    <log_format>eventchannel</log_format>
  </localfile>
```

Make sure the agent monitors the Security Event Log:

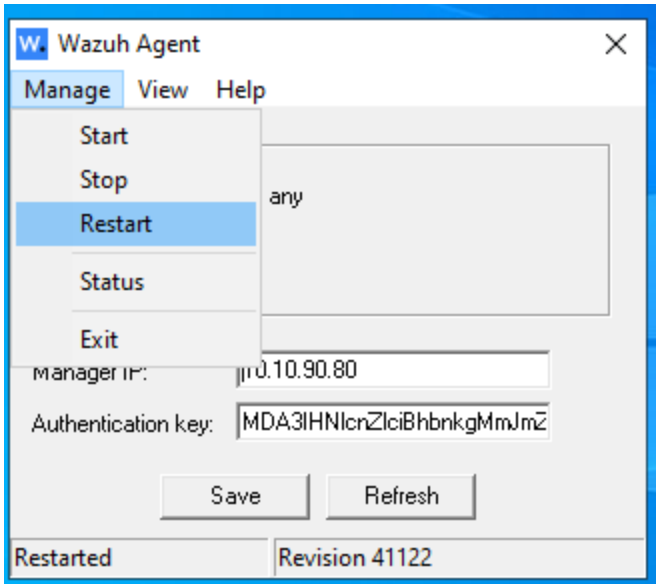
After making changes in the `ossec.conf` file of the **Wazuh Agent**, you need to restart the agent service so the new configuration takes effect.

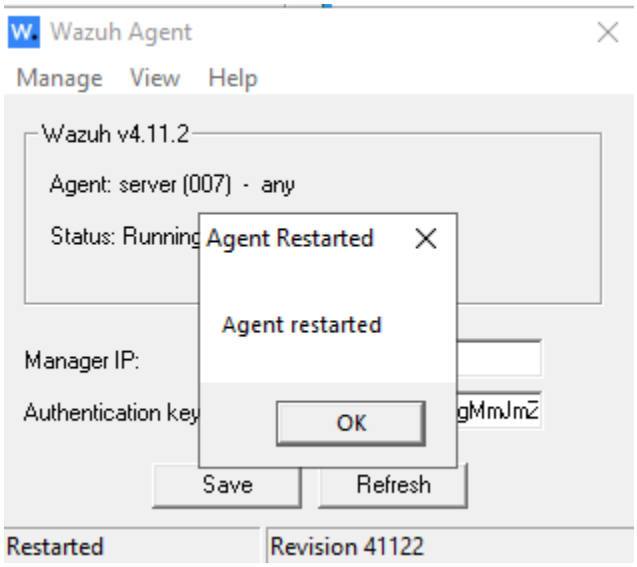
Open PowerShell as Administrator and run the command:
Restart-Service -Name wazuh

```
PS C:\Users\administrator> Restart-Service -Name wazuh
PS C:\Users\administrator>
```

Using Graphical User Interface (GUI)

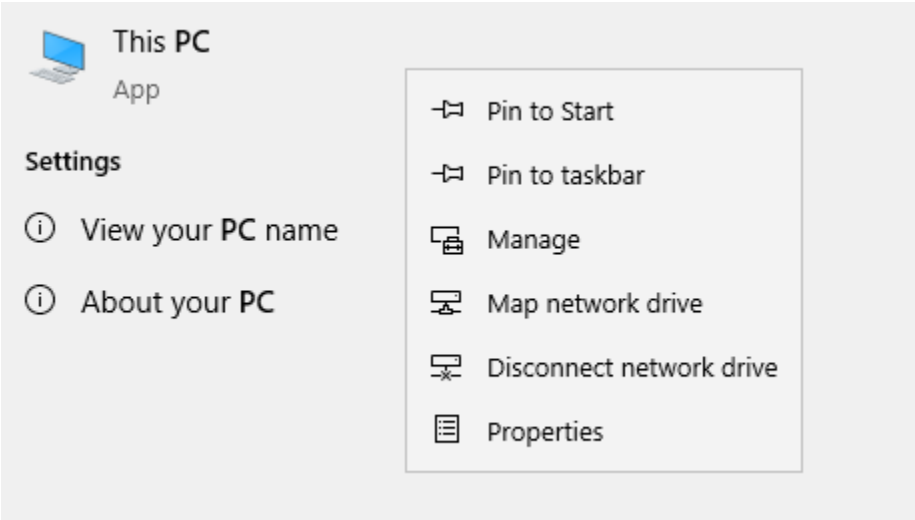
Click on the Start Menu and go to the Search bar.
Type “Wazuh Agent” in the search box.
From the results, right-click on Wazuh Agent Manager.
Select Restart (or first Stop and then Start if Restart option is not available).
This will restart the agent without using command line.



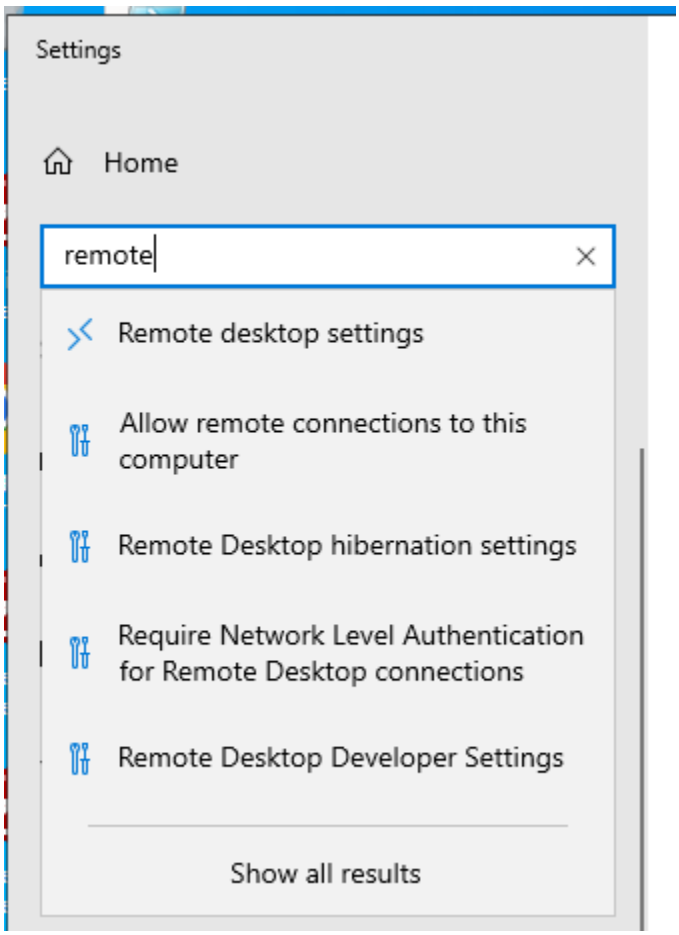


Step 5: Enable Remote Desktop (RDP) on Windows Machine

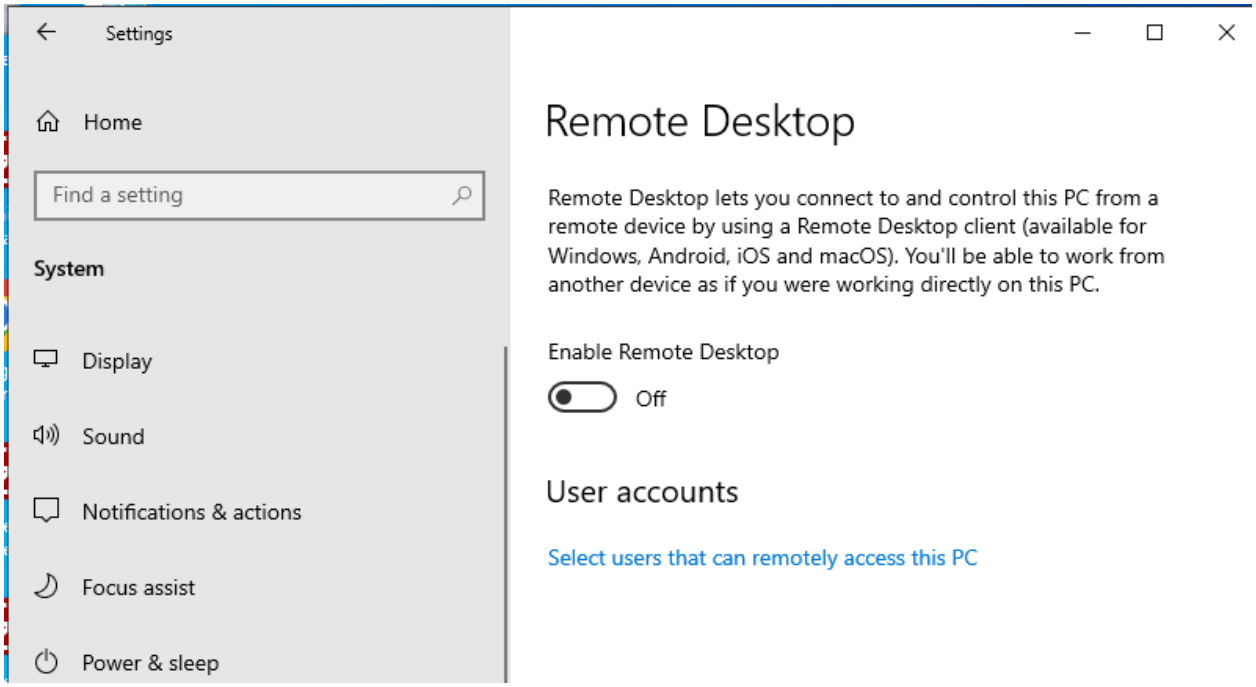
On your Windows machine, right-click This PC and select Properties.

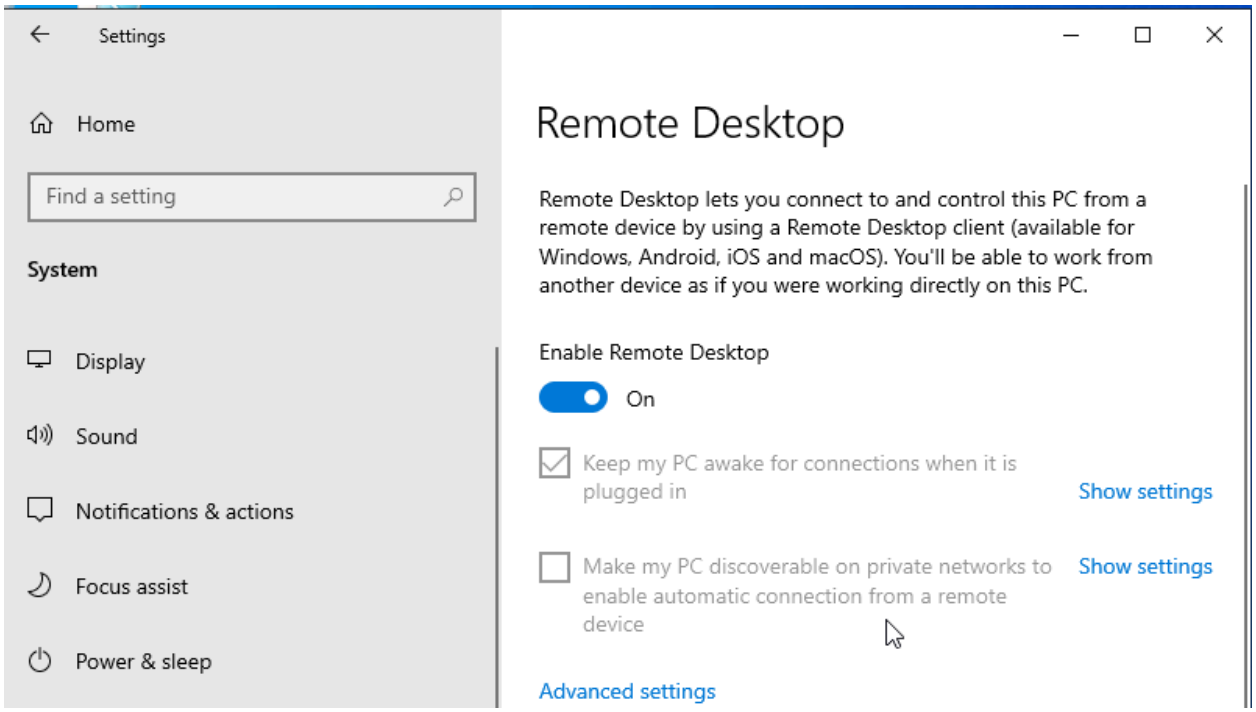


In the left-hand menu, open Remote Desktop settings.



Under the Remote Desktop section, turn ON the option to enable Remote Desktop.





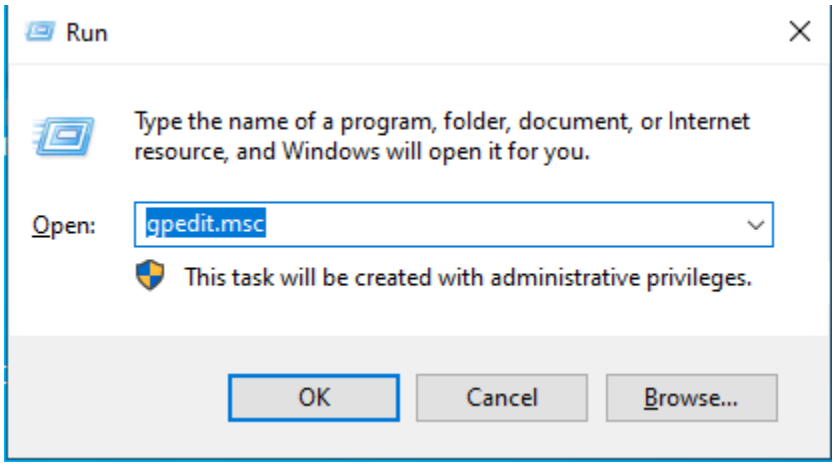
(Optional for testing) Disable the checkbox “Require devices to use Network Level Authentication (NLA)”.

Disable Network Level Authentication (NLA) using GUI:

1. Open Control Panel → System and Security → System.
2. Click Remote settings on the left.
3. In the Remote Desktop section, uncheck:
   - "Allow connections only from computers running Remote Desktop with Network Level Authentication (recommended)".
4. Click Apply and OK.

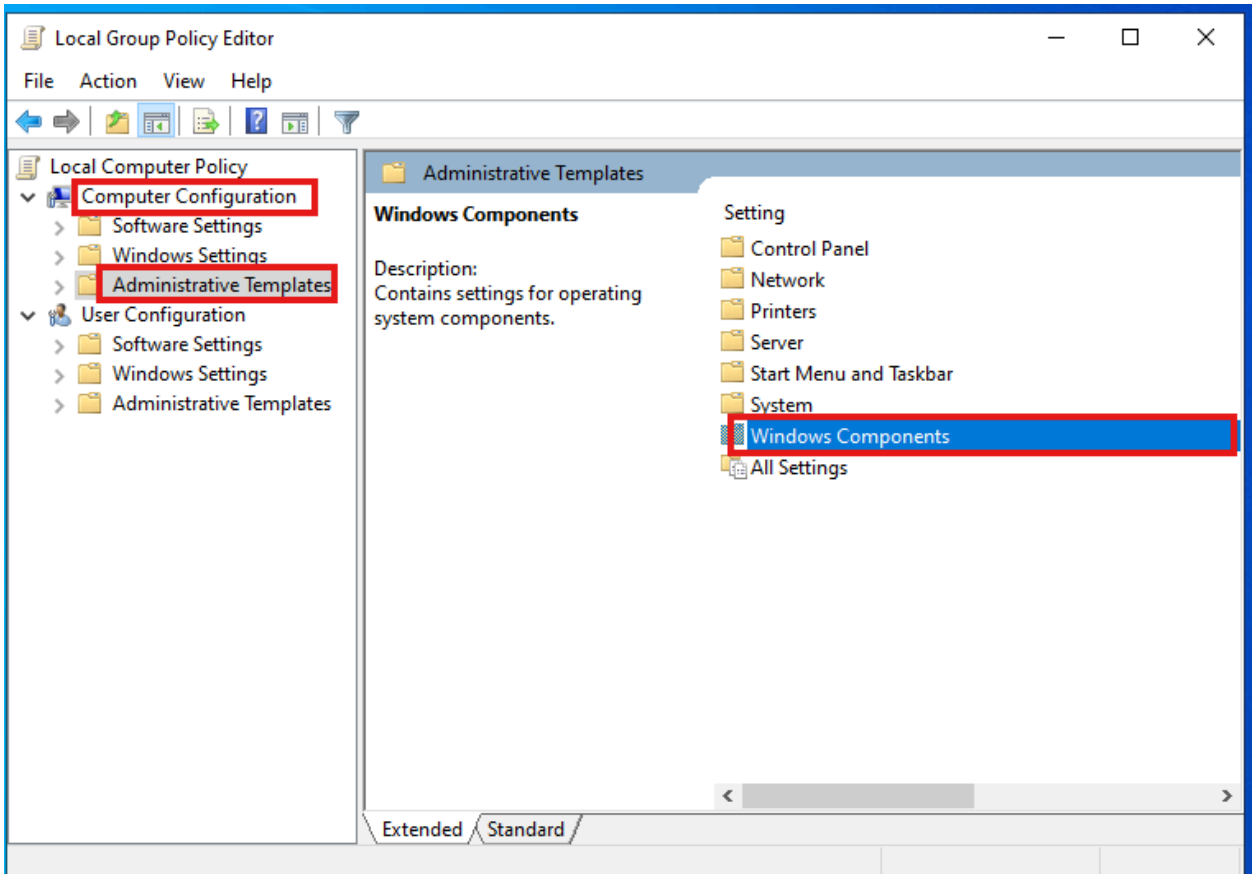
Or disable NLA via Group Policy Editor if the option is greyed out:

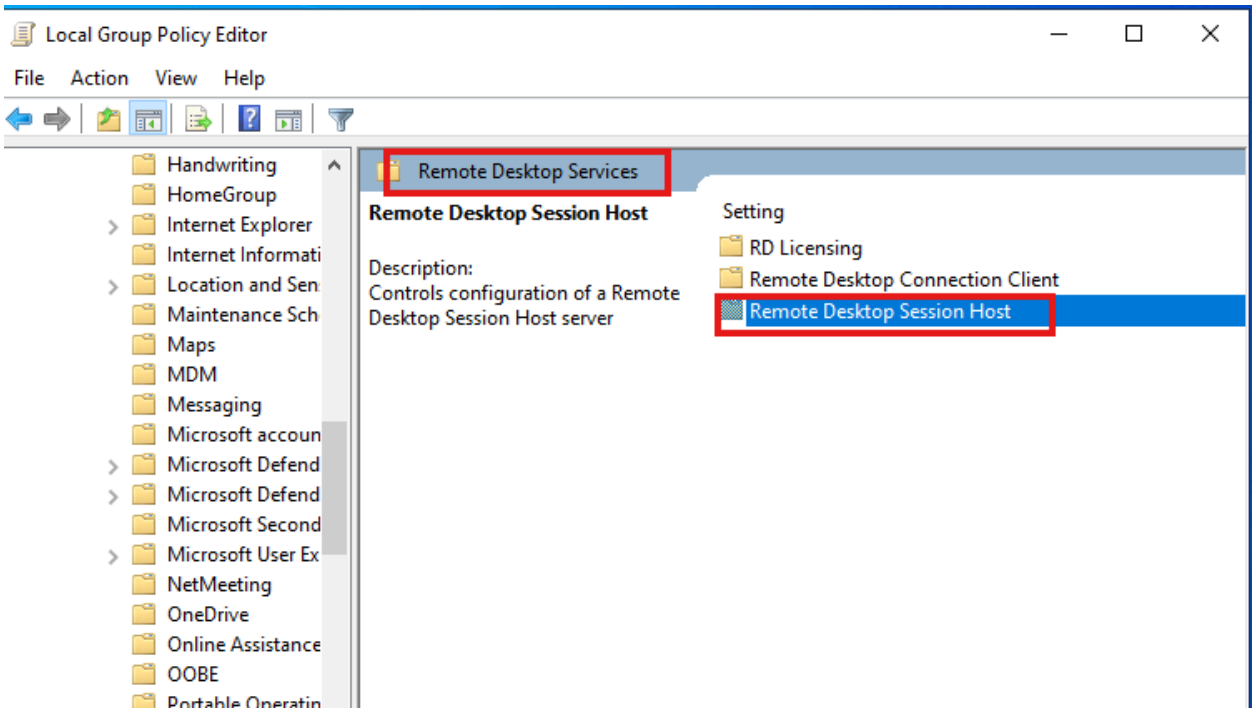
1. Run `gpedit.msc`.

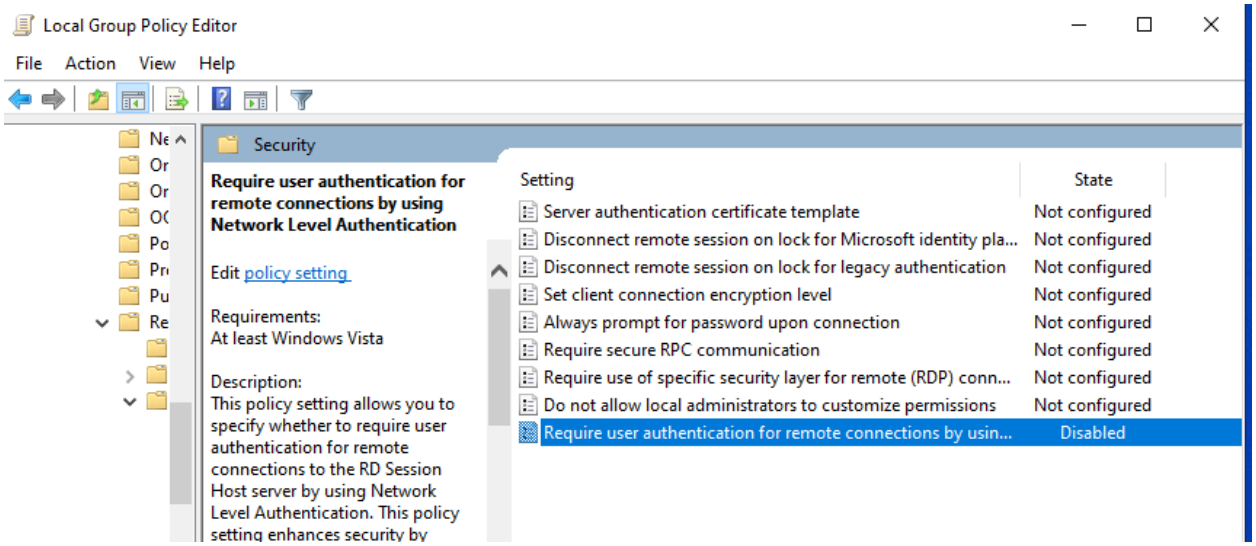


2. Navigate:
   - `Computer Configuration` → `Administrative Templates` → `Windows Components` → `Remote Desktop Services` → `Remote Desktop Session Host` → `Security`.

3. Find and Disable:
   - "Require user authentication for remote connections by using Network Level Authentication".
4. Run `gpupdate /force` to apply.


Local Group Policy Editor
File
Action
View
Help
Local Computer Policy
Computer Configuration
Software Settings
Windows Settings
Administrative Templates
User Configuration
Software Settings
Windows Settings
Administrative Templates
Administrative Templates
Windows Components
Description:
Contains settings for operating system components.
Setting
Control Panel
Network
Printers
Server
Start Menu and Taskbar
System
Windows Components
All Settings
Extended
Standard


Local Group Policy Editor
File
Action
View
Help
Handwriting
HomeGroup
Internet Explorer
Internet Informati
Location and Sen
Maintenance Sch
Maps
MDM
Messaging
Microsoft accoun
Microsoft Defend
Microsoft Defend
Microsoft Second
Microsoft User Ex
NetMeeting
OneDrive
Online Assistance
OOBE
Remote Desktop Services
Remote Desktop Session Host
Description:
Controls configuration of a Remote Desktop Session Host server
Setting
RD Licensing
Remote Desktop Connection Client
Remote Desktop Session Host



This allows easier login attempts during initial testing but is less secure. It should be re-enabled later.
Click Apply or Save to confirm the changes, then close the settings window.

Step 6: Testing the Setup

From a Linux machine (e.g., Kali Linux), install Hydra tool:
sudo apt-get update
sudo apt-get install hydra

```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get update
[sudo] password for kali:
Get:1 https://packages.wazuh.com/4.x/apt stable InRelease [17.3 kB]
Get:2 https://packages.wazuh.com/4.x/apt stable/main amd64 Packages [46.8 kB]
Get:3 https://packages.wazuh.com/4.x/apt stable/main amd64 Contents (deb) [1,957 kB]
Get:5 https://pkgs.tailscale.com/stable/debian bullseye InRelease
Get:6 https://pkgs.tailscale.com/stable/debian bullseye/main amd64 Packages [13.0 kB]
Get:4 http://kali.download/kali kali-rolling InRelease [41.5 kB]
Get:7 http://kali.download/kali kali-rolling/main amd64 Packages [21.1 MB]
Get:8 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [51.3 MB]
Get:9 http://kali.download/kali kali-rolling/contrib amd64 Packages [118 kB]
Get:10 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [325 kB]
Get:11 http://kali.download/kali kali-rolling/non-free amd64 Packages [200 kB]
Get:12 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [911 kB]
Fetched 76.0 MB in 21s (3,549 kB/s)
Reading package lists... Done

┌──(kali㉿kali)-[~]
└─$ sudo apt-get install hydra

Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
hydra is already the newest version (9.5-3).
hydra set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 2104 not upgraded.

┌──(kali㉿kali)-[~]
```

Prepare username and password lists (user.txt, pass.txt).
Execute a brute force test attack targeting your Windows machine:

echo "administrator" > user.txt
echo "123456" > pass.txt
echo "password" >> pass.txt

```
┌──(kali㉿kali)-[~]
└─$ echo "administrator" > user.txt
echo "123456" > pass.txt
echo "password" >> pass.txt


┌──(kali㉿kali)-[~]
└─$ sudo nano user.txt

┌──(kali㉿kali)-[~]
└─$ sudo nano pass.txt

┌──(kali㉿kali)-[~]
```

sudo nano user.txt
sudo nano pass.txt

```
  GNU nano 8.3                                                    user.txt
administrator
admin
Akmal
Altaf
Aslam
```

```
  GNU nano 8.3                                                    pass.txt
123456
password
9876654
3457654
98763
```

sudo hydra -L user.txt -P pass.txt -t 1 rdp://<WINDOWS_IP_ADDRESS>

```
┌──(kali㉿kali)-[~]
└─$ sudo hydra -L user.txt -P pass.txt -t 1 rdp://[redacted]

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret
service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and et
hics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-08-21 15:49:41
[WARNING] the rdp module is experimental. Please test, report - and if possible, fix.
[DATA] max 1 task per 1 server, overall 1 task, 49 login tries (l:7/p:7), ~49 tries per task
[DATA] attacking rdp://[redacted]:3389/
[STATUS] 25.57 tries/min, 26 tries in 00:01h, 23 to do in 00:01h, 1 active
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-08-21 15:51:34


┌──(kali㉿kali)-[~]
```

Check Windows Firewall Rules on the Agent

Run the following command in an elevated Command Prompt or PowerShell:

```
netsh advfirewall firewall show rule name=all
```

Purpose:

Confirms which firewall rules are currently active.

Identifies if any IP addresses or connections have been blocked due to suspicious activity.I show only one.

```
PS C:\Users\administrator> netsh advfirewall firewall show rule name=all

Rule Name:                            Tailscale-In
----------------------------------------------------------------------
Enabled:                              Yes
Direction:                            In
Profiles:                             Domain,Private
Grouping:
LocalIP:                              fd7a:115c:a1e0::8f36:262b/128
RemoteIP:                             Any
Protocol:                             Any
Edge traversal:                       No
Action:                               Allow
```

```
Rule Name:                            WAZUH ACTIVE RESPONSE BLOCKED IP
----------------------------------------------------------------------
Enabled:                              Yes
Direction:                            In
Profiles:                             Domain,Private,Public
Grouping:
LocalIP:                              Any
RemoteIP:
Protocol:                             Any
Edge traversal:                       No
Action:                               Block
```

The above demonstrates that the manager executed an active response, successfully blocking the IP from which the RDP attempts originated.

After successful tests, re-enable Windows firewall profiles to maintain security:

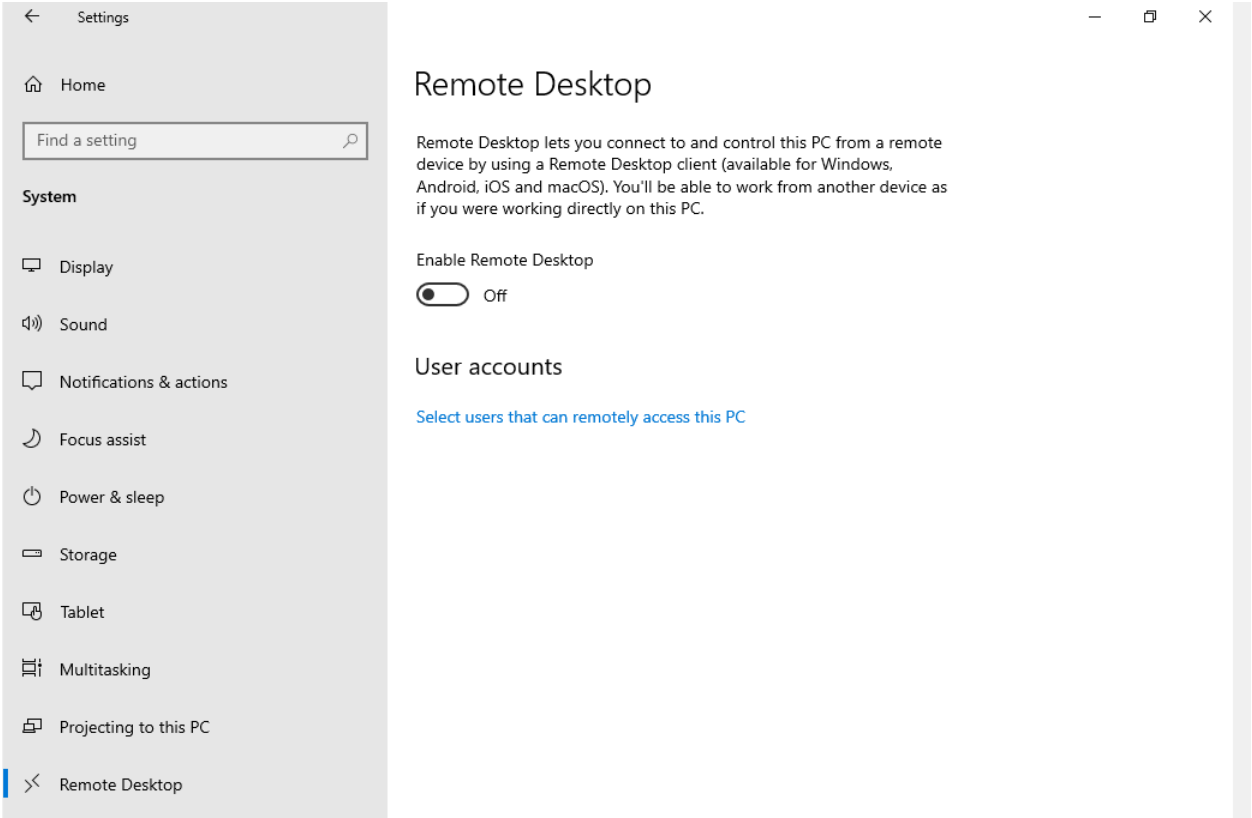
Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled True

```
PS C:\Users\administrator>
>> Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled True
PS C:\Users\administrator>
```

After Testing Blocking Remote Desktop (RDP) Access on Windows

Step 1: Disable Remote Desktop via System Settings

Under the Remote Desktop section, turn off the option to disable Remote Desktop.



Step 2: Block RDP via Windows Firewall

Open PowerShell as Administrator.
Disable all existing RDP firewall rules:

```
Get-NetFirewallRule -DisplayGroup "Remote Desktop" |
Disable-NetFirewallRule
```

Step 3:(Optional) Create an explicit block rule for additional security:

```
New-NetFirewallRule -DisplayName "Block RDP" -Direction Inbound
-Protocol TCP -LocalPort 3389 -Action Block
```

This ensures that all inbound RDP connections, including those over Tailscale or local networks, are blocked.

```
PS C:\Users\administrator> Get-NetFirewallRule -DisplayGroup "Remote Desktop" | Disable-NetFirewallRule
PS C:\Users\administrator>
>> New-NetFirewallRule -DisplayName "Block RDP" -Direction Inbound -Protocol TCP -LocalPort 3389 -Action Block


Name                          : {8fee7289-2d79-4219-915a-0fd83a42e6a9}
DisplayName                   : Block RDP
Description                   :
DisplayGroup                  :
Group                         :
Enabled                       : True
Profile                       : Any
Platform                      : {}
Direction                     : Inbound
Action                        : Block
EdgeTraversalPolicy           : Block
LooseSourceMapping            : False
LocalOnlyMapping              : False
Owner                         :
PrimaryStatus                 : OK
Status                        : The rule was parsed successfully from the store. (65536)
EnforcementStatus             : NotApplicable
PolicyStoreSource             : PersistentStore
PolicyStoreSourceType         : Local
RemoteDynamicKeywordAddresses : {}
PolicyAppId                   :
```

Monitoring RDP Activity via Wazuh Dashboard

The Wazuh Dashboard provides centralized monitoring of security events and alerts across endpoints.
Navigate to Threat Hunting or Security Events to view real-time alerts.
Rule ID 100111 detects suspicious RDP activity, such as unauthorized logins or brute-force attempts.
Dashboard displays alert details: source IP, affected host, timestamp, and severity.
Purpose: Quickly identify and respond to potential RDP threats.

	Aug 21, 2025 @ 15:50:36.5...	server	Active response: active-response/bin/netsh.exe - add	3	657
	Aug 21, 2025 @ 15:50:33.1...	server	RDP Bruteforce Attack Detected	10	100111
	Aug 21, 2025 @ 15:50:31.8...	server	Logon Failure - Unknown user or bad password	5	60122
	Aug 21, 2025 @ 15:50:30.5...	server	Active response: active-response/bin/netsh.exe - add	3	657
	Aug 21, 2025 @ 15:50:29.3...	server	Logon Failure - Unknown user or bad password	5	60122
	Aug 21, 2025 @ 15:50:29.2...	server	RDP Bruteforce Attack Detected	10	100111
	Aug 21, 2025 @ 15:50:27.6...	server	Active response: active-response/bin/netsh.exe - add	3	657
	Aug 21, 2025 @ 15:50:27.1...	server	Logon Failure - Unknown user or bad password	5	60122
	Aug 21, 2025 @ 15:50:27.1...	server	Logon Failure - Unknown user or bad password	5	60122
	Aug 21, 2025 @ 15:50:25.1...	server	RDP Bruteforce Attack Detected	10	100111
	Aug 21, 2025 @ 15:50:25.0...	server	Logon Failure - Unknown user or bad password	5	60122
	Aug 21, 2025 @ 15:50:23.9...	server	Logon Failure - Unknown user or bad password	5	60122
	Aug 21, 2025 @ 15:50:23.5...	server	Active response: active-response/bin/netsh.exe - add	3	657

For detail Information click on the RDP Bruteforce Attack Detected

Table JSON

@timestamp	Aug 21, 2025 @ 15:50:33.141
_index	wazuh-alerts-4.x-2025.08.21
agent.id	007
agent.ip	10.0.2.15
agent.name	server
data.win.eventdata.authenticationPackageName	NTLM
data.win.eventdata.failureReason	%%2313
data.win.eventdata.ipAddress	[redacted]
data.win.eventdata.ipPort	0
data.win.eventdata.keyLength	0
data.win.eventdata.logonProcessName	NtLmSsp
data.win.eventdata.logonType	3
data.win.eventdata.processId	0x0
data.win.eventdata.status	0xc000006d
data.win.eventdata.subStatus	0xc0000064

data.win.eventdata.subjectUserSid	S-1-0-0
data.win.eventdata.targetUserName	Akmal
data.win.eventdata.targetUserSid	S-1-0-0
data.win.eventdata.workstationName	kali
data.win.system.channel	Security
data.win.system.computer	server.server.local
data.win.system.eventID	4625
data.win.system.eventRecordID	71898
data.win.system.keywords	0x8010000000000000
data.win.system.level	0
data.win.system.message	"An account failed to log on. Subject: Security ID: S-1-0-0 Account Name: - Account Domain: - Logon ID: 0x0
data.win.system.opcode	0
data.win.system.processID	772

input.type	log
location	EventChannel
manager.name	kali
previous_output	{"win":{"system":{"providerName":"Microsoft-Windows-Security-Auditing","providerGuid":"{54849625-5478-4994-a5ba-3e3b0328c30d}","eventID":"4625","version":"0","level":"0","task":"12544","opcode":"0","keywords":"0x8010000000000000","systemTime":"2025-08-21T10:53:46.3325048Z","eventRecordID":"71897","processID":"772","threadID":"5320","channel":"Security","computer":"server.server.local","severityValue":"AUDIT_FAILURE","message":"\"An account failed to log on.\r\n\r\nSubject:\r\n\tSecurity ID:\t\tS-1-0-0\r\n\tAccount Name:\t\t-\r\n\tAccount Domain:\t\t-\r\n\tLogon ID:\t\t0x0\r\n\r\nLogon Type:\t\t\t3\r\n\r\nAccount For Which Logon Failed:\r\n\tSecurity ID:\t\tS-1-0-0\r\n\tAccount Name:\t\tAkmal\r\n\tAccount Domain:\t\t-\r\n\r\nFailure Information:\r\n\tFailure Reason:\t\tUnknown user name or bad password.\r\n\tStatus:\t\t\t0xC000006D\r\n\tSub Status:\t\t0xC000006A\r\n\r\nProcess Information:\r\n\tCaller Process ID:\t0x0\r\n\tCaller Process Name:\t-\r
rule.description	RDP Bruteforce Attack Detected
rule.firedtimes	22
rule.frequency	3
rule.groups	rdp
rule.id	100111
rule.level	10
rule.mail	true
timestamp	Aug 21, 2025 @ 15:50:33.141

For active response section more detail click on " Active response: active-response/bin/netsh.exe - add"

Table JSON

@timestamp	Aug 21, 2025 @ 15:50:27.611
_index	wazuh-alerts-4.x-2025.08.21
agent.id	007
agent.ip	10.0.2.15
agent.name	server
data.command	add
data.origin.module	wazuh-execd
data.origin.name	node01
data.parameters.alert.agent.id	007
data.parameters.alert.agent.ip	10.0.2.15
data.parameters.alert.agent.name	server
data.parameters.alert.data.win.eventdata.authenticationPackageName	NTLM
data.parameters.alert.data.win.eventdata.failureReason	%%2313
data.parameters.alert.data.win.eventdata.ipAddress	[redacted]
data.parameters.alert.data.win.eventdata.ipPort	0
data.parameters.alert.data.win.system.eventRecordID	71892
data.parameters.alert.data.win.system.keywords	0x8010000000000000
data.parameters.alert.data.win.system.level	0
data.parameters.alert.data.win.system.message	"An account failed to log on. Subject: Security ID: S-1-0-0 Account Name: - Account Domain: - Logon ID: 0x0
data.parameters.alert.data.win.system.opcode	0
data.parameters.alert.data.win.system.processID	772
data.parameters.alert.data.win.system.providerGuid	{54849625-5478-4994-a5ba-3e3b0328c30d}
data.parameters.alert.data.win.system.providerName	Microsoft-Windows-Security-Auditing
data.parameters.alert.data.win.system.severityValue	AUDIT_FAILURE
data.parameters.alert.data.win.system.systemTime	2025-08-21T10:53:41.1145984Z
data.parameters.alert.data.win.system.task	12544
data.parameters.alert.data.win.system.threadID	5320
data.parameters.alert.data.win.system.version	0

data.parameters.program	active-response/bin/netsh.exe
data.srcip	[redacted]
data.version	1
decoder.name	ar_log_json
decoder.parent	ar_log_json
full_log	2025/08/21 15:53:41 active-response/bin/netsh.exe: {"version":1,"origin":{"name":"node01","module":"wazuh-execd"},"command":"add","parameters":{"extra_args":[],"alert":{"timestamp":"2025-08-21T06:50:25.118-0400","rule":{"level":10,"description":"RDP Bruteforce Attack Detected","id":"100111","frequency":3,"firedtimes":20,"mail":true,"groups":["rdp"]},"agent":{"id":"007","name":"server","ip":"10.0.2.15"},"manager":{"name":"kali"},"id":"1755773425.10310919","previous_output":"{\"win\":{\"system\":{\"providerName\":\"Microsoft-Windows-Security-Auditing\",\"providerGuid\":\"{54849625-5478-4994-a5ba-3e3b0328c30d}\",\"eventID\":\"4625\",\"version\":\"0\",\"level\":\"0\",\"task\":\"12544\",\"opcode\":\"0\",\"keywords\":\"0x8010000000000000\",\"systemTime\":\"2025-08-21T10:53:40.9774113Z\",\"eventRecor
id	1755773427.10334552
input.type	log
location	active-response\active-responses.log
manager.name	kali
rule.description	Active response: active-response/bin/netsh.exe - add
rule.firedtimes	20
rule.gdpr	IV_35.7.d

Summary:

This project focuses on monitoring and securing Remote Desktop Protocol (RDP) access using Wazuh for threat detection and automated response. Initially, RDP was enabled on the Windows machine to perform thorough connectivity and functionality testing, both locally and over Tailscale. The system provides real-time alerts on potential brute-force attacks, ensuring proactive protection against unauthorized login attempts. Windows Firewall rules were validated, and active response logs on the agent were reviewed to confirm that automated IP blocking was functioning correctly. Following testing, Remote Desktop access was securely disabled through system settings, and existing firewall rules were disabled while a new explicit block rule was applied to TCP port 3389 to prevent unauthorized access. All activities, including connection attempts and automated responses, are logged for monitoring and auditing purposes, ensuring traceability and accountability. This structured approach enhances system security, enforces effective threat mitigation, and provides a scalable framework for monitoring and responding to RDP-related attacks.