Incident Response Playbook

RDP Brute-Force Attack
Detection, Containment, Eradication, and Recovery

Prepared by: Youssef Sherif
Date: October 03, 2025

1. Preparation

Objective:

Set up a controlled lab environment to simulate and detect brute-force attacks on Windows Server 2022 RDP (port 3389) using Hydra, with monitoring and logging via Suricata, Zeek, Wireshark, and Splunk.

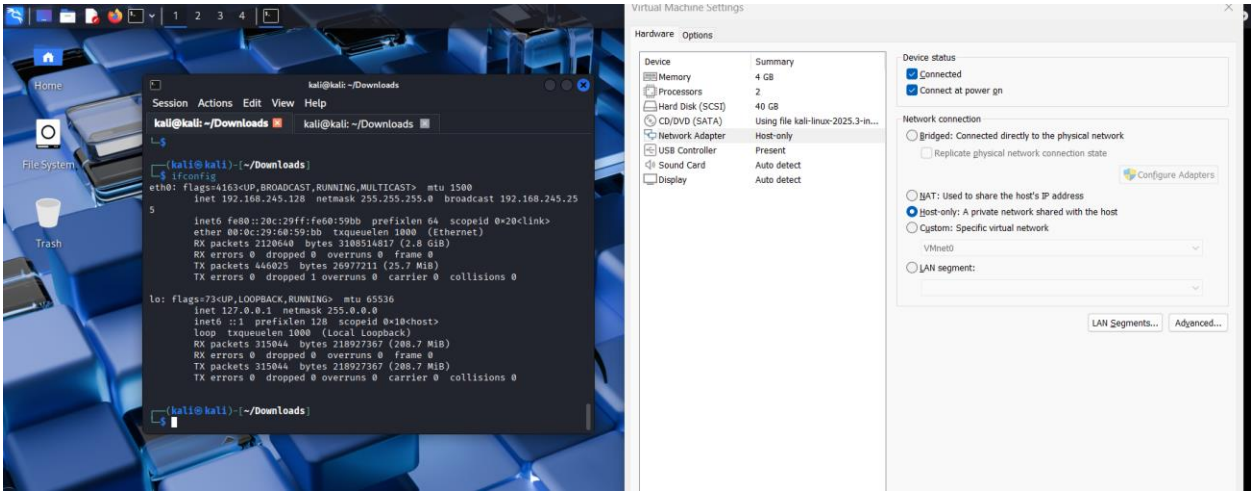
Key Activities:

- **Environment Setup:**
  - **VMware Workstation 17** running (host-only network to ensure isolation from production/external networks):

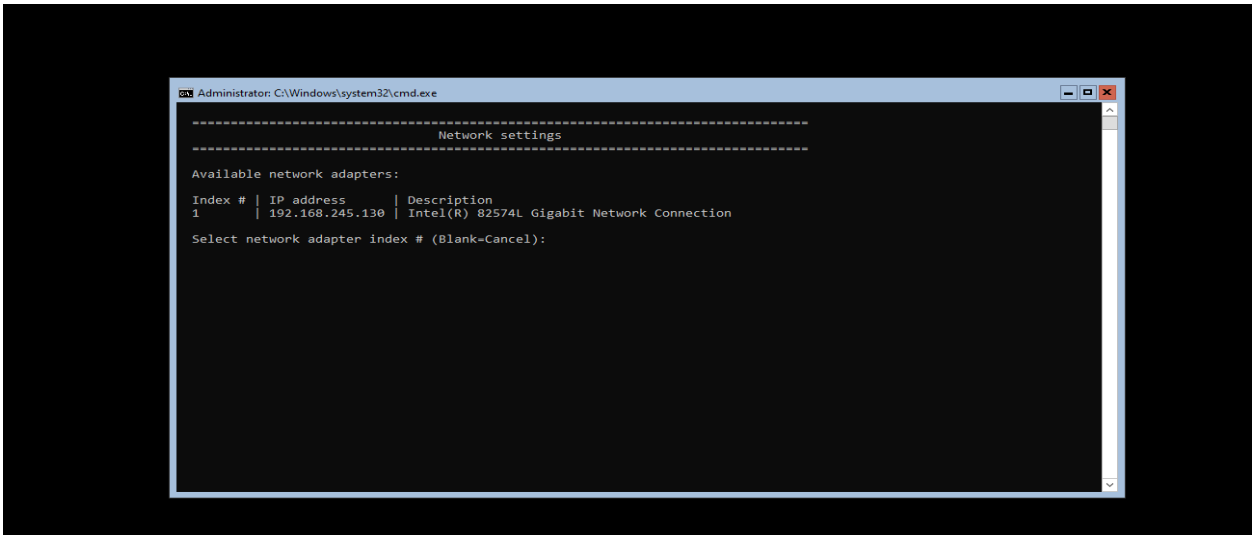
```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sS -sV -O -p- -T4 192.168.245.130 -oA nmap_scan_192.168.245.129

Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-03 14:43 EDT
Nmap scan report for 192.168.245.130
Host is up (0.00065s latency).
Not shown: 65533 filtered tcp ports (no-response)
PORT     STATE SERVICE       VERSION
3389/tcp open  ms-wbt-server Microsoft Terminal Services
5985/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
MAC Address: 00:0C:29:66:B7:7F (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 2016|10|11 (94%)
OS CPE: cpe:/o:microsoft:windows_server_2016 cpe:/o:microsoft:windows_10 cpe:/o:microsoft:windows_11
Aggressive OS guesses: Microsoft Windows Server 2016 (94%), Microsoft Windows 10 - 11 (87%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 113.27 seconds
```

```
┌──(kali㉿kali)-[~]
└─$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:60:59:bb brd ff:ff:ff:ff:ff:ff
    inet 192.168.245.128/24 brd 192.168.245.255 scope global dynamic noprefixroute eth0
       valid_lft 1245sec preferred_lft 1245sec
    inet6 fe80::20c:29ff:fe60:59bb/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

- **Kali Linux 2025.3** (Attacker + Splunk server, IP: 192.168.245.128)



- **Windows Server 2022** (Victim, IP: 192.168.245.130, RDP enabled on port 3389).
  - Target account under attack: **Administrator** account on Windows Server.



- **Security Tool Deployment:**
  - **Hydra v9.6** → used on Kali for brute-force attempts against RDP.
  - **Suricata 7.0.11** → deployed on Kali to monitor RDP traffic, with a custom detection rule in local.rules
  - With editing the suricata configuration using suricata.yaml file

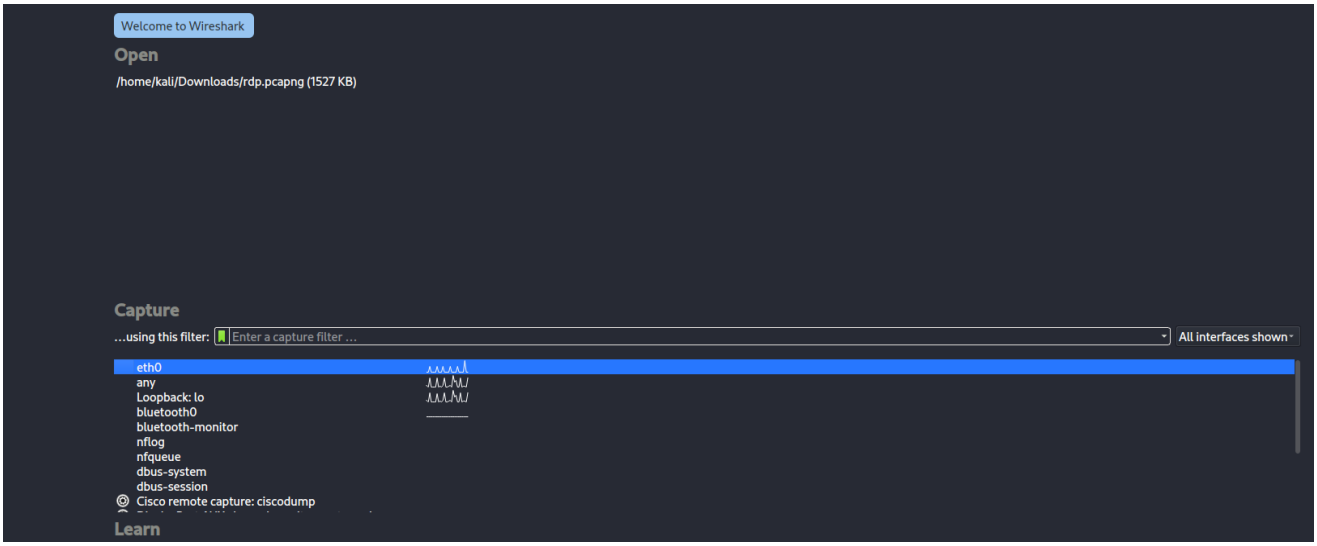
```
File  Actions  Edit  View  Help
  GNU nano 8.2                                                    suricata.yaml
%YAML 1.1
---

# Suricata configuration file. In addition to the comments describing all
# options in this file, full documentation can be found at:
# https://docs.suricata.io/en/latest/configuration/suricata-yaml.html

# This configuration file generated by Suricata 7.0.7.
suricata-version: "7.0"

##
## Step 1: Inform Suricata about your network
##

vars:
  # more specific is better for alert accuracy and performance
  address-groups:
    HOME_NET: "[192.168.245.0/24]"
    #HOME_NET: "[192.168.0.0/16]"
    #HOME_NET: "[10.0.0.0/8]"
    #HOME_NET: "[172.16.0.0/12]"
    #HOME_NET: "any"

    EXTERNAL_NET: "!$HOME_NET"
    #EXTERNAL_NET: "any"

    HTTP_SERVERS: "$HOME_NET"
    SMTP_SERVERS: "$HOME_NET"
    SQL_SERVERS: "$HOME_NET"
    DNS_SERVERS: "$HOME_NET"
    TELNET_SERVERS: "$HOME_NET"
    AIM_SERVERS: "$EXTERNAL_NET"
    DC_SERVERS: "$HOME_NET"
    DNP3_SERVER: "$HOME_NET"
    DNP3_CLIENT: "$HOME_NET"
    MODBUS_CLIENT: "$HOME_NET"
    MODBUS_SERVER: "$HOME_NET"
    ENIP_CLIENT: "$HOME_NET"
    ENIP_SERVER: "$HOME_NET"

  port-groups:
    HTTP_PORTS: "80"
    SHELLCODE_PORTS: "!80"
    ORACLE_PORTS: 1521
    SSH_PORTS: 22
    DNP3_PORTS: 20000
    MODBUS_PORTS: 502
    FILE_DATA_PORTS: "[$HTTP_PORTS,110,143]"
                                                          [ Read 2193 lines ]
```

```
    #
    # This parameter has no effect if auto-config is disabled.
    #
    ports: [0-1,2-3]

    # When auto-config is enabled the hashmode specifies the algorithm for
    # determining to which stream a given packet is to be delivered.
    # This can be any valid Napatech NTPL hashmode command.
    #
    # The most common hashmode commands are:  hash2tuple, hash2tuplesorted,
    # hash5tuple, hash5tuplesorted and roundrobin.
    #
    # See Napatech NTPL documentation other hashmodes and details on their use.
    #
    # This parameter has no effect if auto-config is disabled.
    #
    hashmode: hash5tuplesorted

##
## Configure Suricata to load Suricata-Update managed rules.
##

default-rule-path: /var/lib/suricata/rules

rule-files:
  - suricata.rules
  - /etc/suricata/rules/local.rules
##
## Auxiliary configuration files.
##

classification-file: /etc/suricata/classification.config
reference-config-file: /etc/suricata/reference.config
# threshold-file: /etc/suricata/threshold.config

##
## Include other configs
##

# Includes:  Files included here will be handled as if they were in-lined
# in this configuration file. Files with relative pathnames will be
# searched for in the same directory as this configuration file. You may
# use absolute pathnames too.
#include:
#  - include1.yaml
#  - include2.yaml
```

```
alert tcp any any <> any 3389 (msg:"RDP CONNECTION ATTEMPTS"; sid:100001; rev:1;)
```

Logs stored in /var/log/suricata/ → eve.json (JSON alerts for Splunk), fast.log, stats.log.

- **Zeek 8.0.1** → deployed on Kali with RDP protocol analyzer enabled, generating logs in /opt/zeek/logs/current/rdp.log.
- **Wireshark 4.4.9** → configured to capture **all traffic** during the brute-force simulation. A filtered subset (rdp.pcap) containing only RDP packets (tcp.port == 3389 or rdp) will be extracted and analyzed using Zeek to generate protocol logs (rdp.log using zeek tool).



- **Splunk Enterprise 10.0.1 (build c486717c322b)** → installed on Kali and acting as the central SIEM receiver for forwarded logs.

- **Evidence Storage Policy:**
  - Central log and capture directory on Kali: /opt/logs/rdp-bruteforce-lab/.
  - Data formats: JSON (eve.json), plain alert logs (fast.log), Zeek TSV (rdp.log), PCAPs (rdp.pcap).
  - Retention: minimum **7 days** for analysis.

- **Policies & Documentation:**
  - Lab is isolated using VMware host-only networking to prevent any traffic to/from production or the Internet.
  - Only the Kali VM is authorized to perform brute-force attempts.
  - Normal baseline traffic captured prior to attack for comparison.

- **Backups & Inventory:**
  - VM snapshots of Kali and Windows taken before running Hydra.
  - Inventory maintained with IPs, tool versions, and log locations:
    - Hydra v9.6
    - Suricata 7.0.11
    - Zeek 8.0.1
    - Wireshark 4.4.9
    - Splunk Enterprise 10.0.1

2. Identification Phase

Objective

The purpose of the Identification Phase is to detect and confirm malicious activity within the network. Multiple security tools were utilized to monitor traffic and identify indicators of a brute-force attack targeting the Windows Server 2022 RDP service.

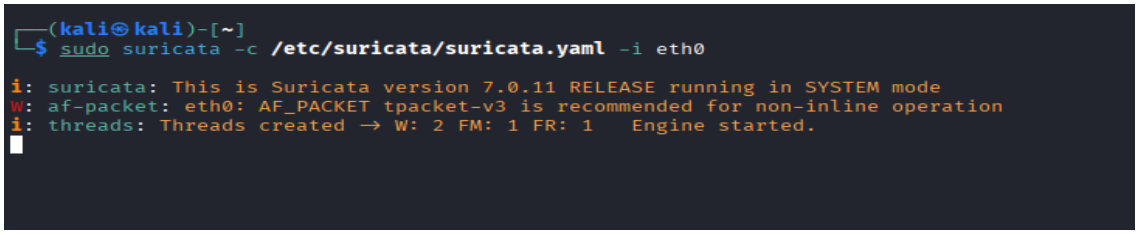
Tools Used

- **Suricata (IDS)** – For real-time detection of brute-force attempts.
- **Wireshark** – For packet capture and analysis of RDP traffic.
- **Splunk** – For centralized log analysis and visualization.
- **Zeek** – For network monitoring and extracting behavioral insights.

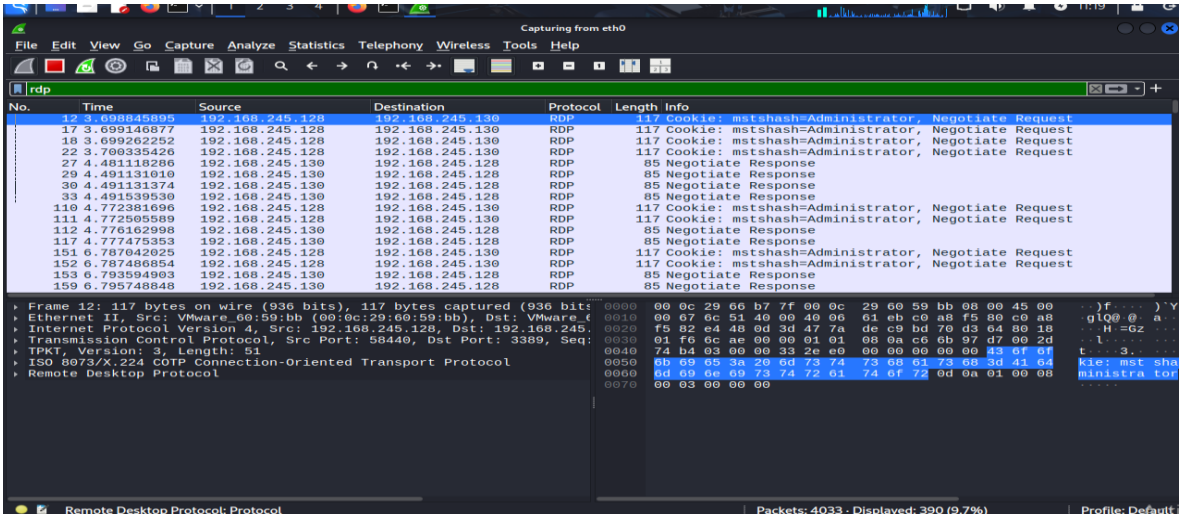
Evidence of Brute Force Attack

The detection process involved analyzing alerts and logs generated by the IDS and traffic analysis tools.

- **Suricata**: Detected multiple alerts associated with RDP brute-force attempts.

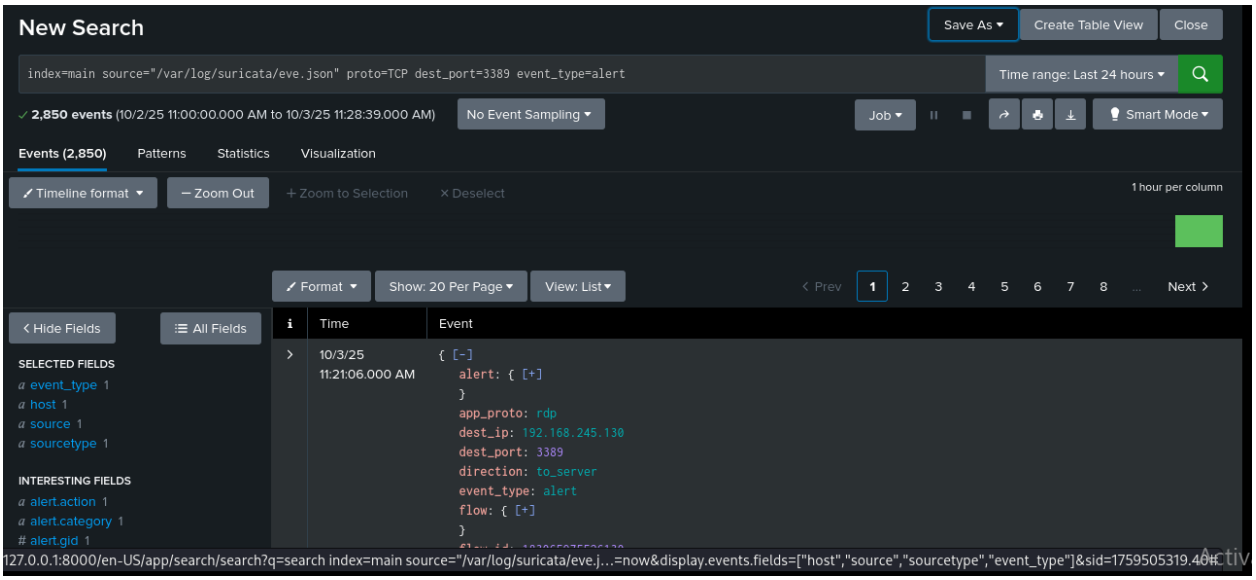


- **Wireshark**: Captured repeated TCP connection attempts to port 3389 (RDP), confirming excessive login attempts.



- **Zeek**: Generated connection and authentication logs highlighting repeated RDP session attempts.

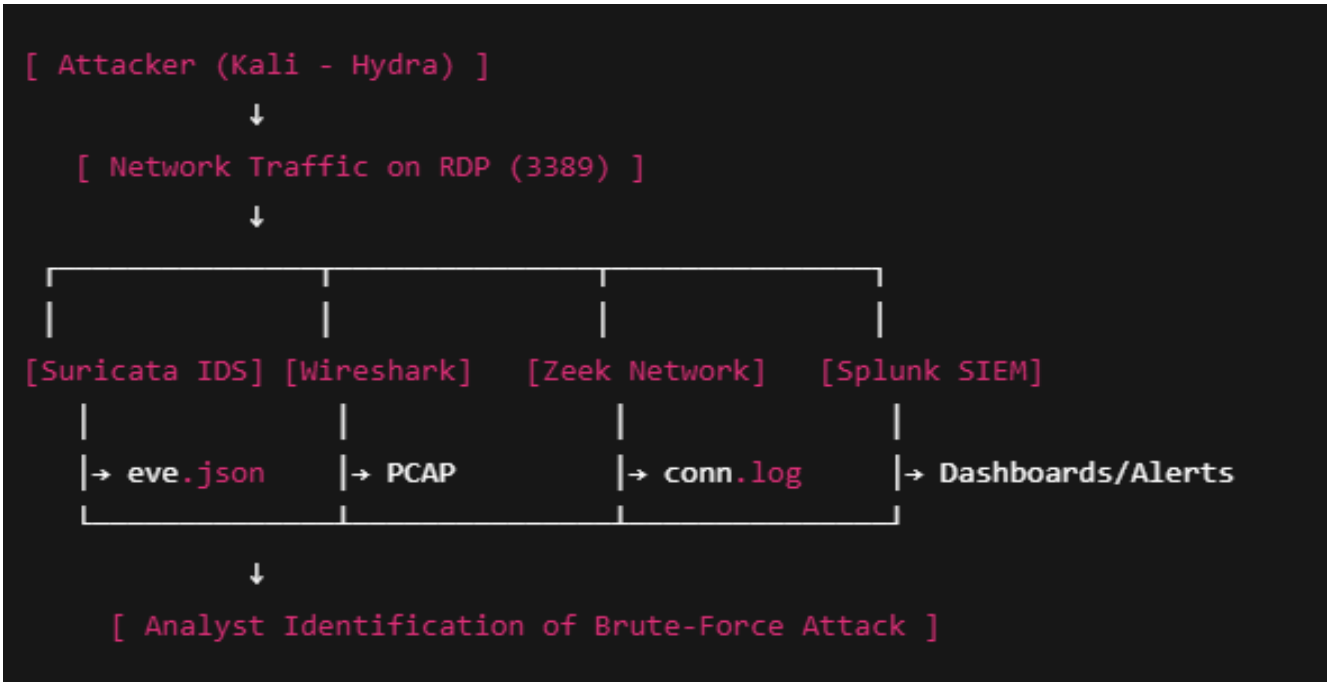
- **Splunk**: Visualized Suricata/Zeek logs, showing high-frequency RDP login failures from the attacker IP.



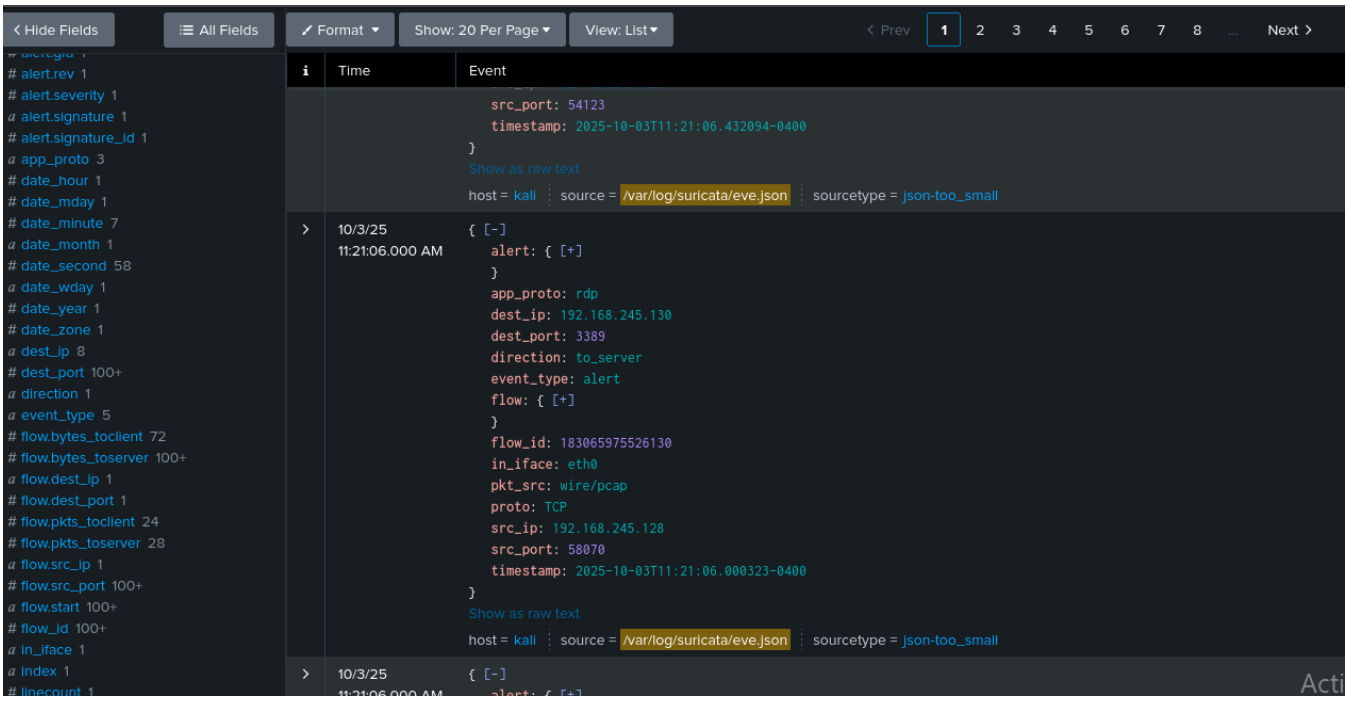
Findings

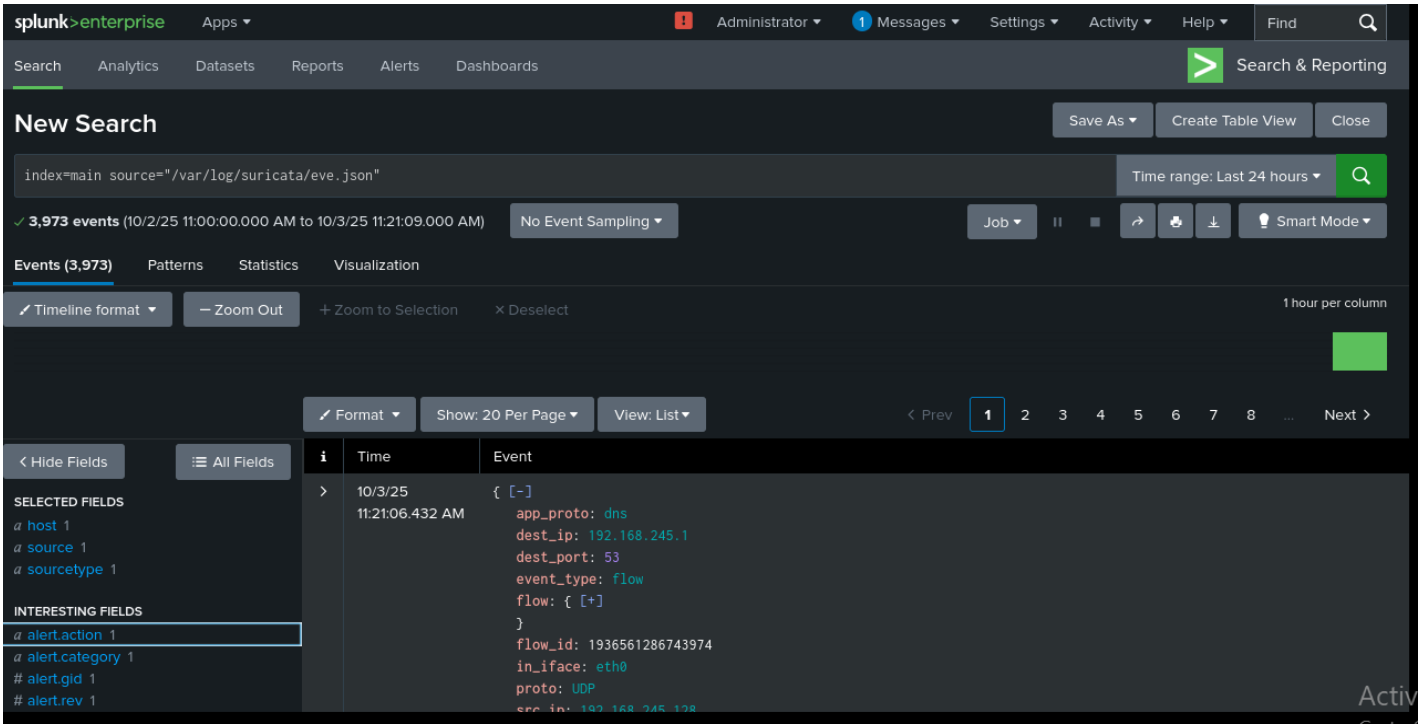
From the collected evidence:

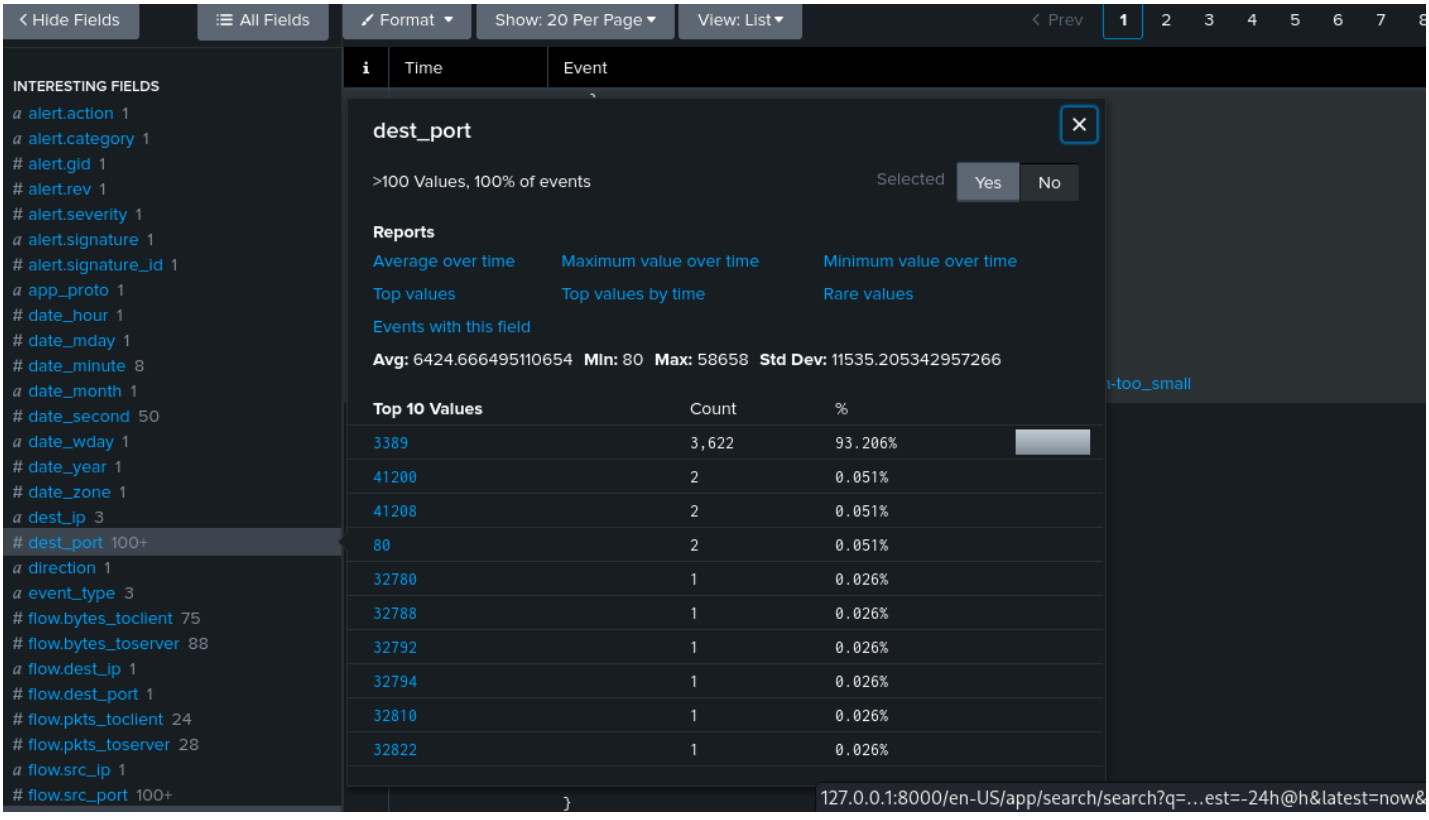
- Suricata raised **multiple RDP brute-force alerts**.
- Wireshark traffic analysis confirmed **continuous login attempts** from the attacker.
- Zeek logs supported this by showing **unusual session behavior**.
- Splunk correlated the data, confirming the attack's **frequency and severity**.

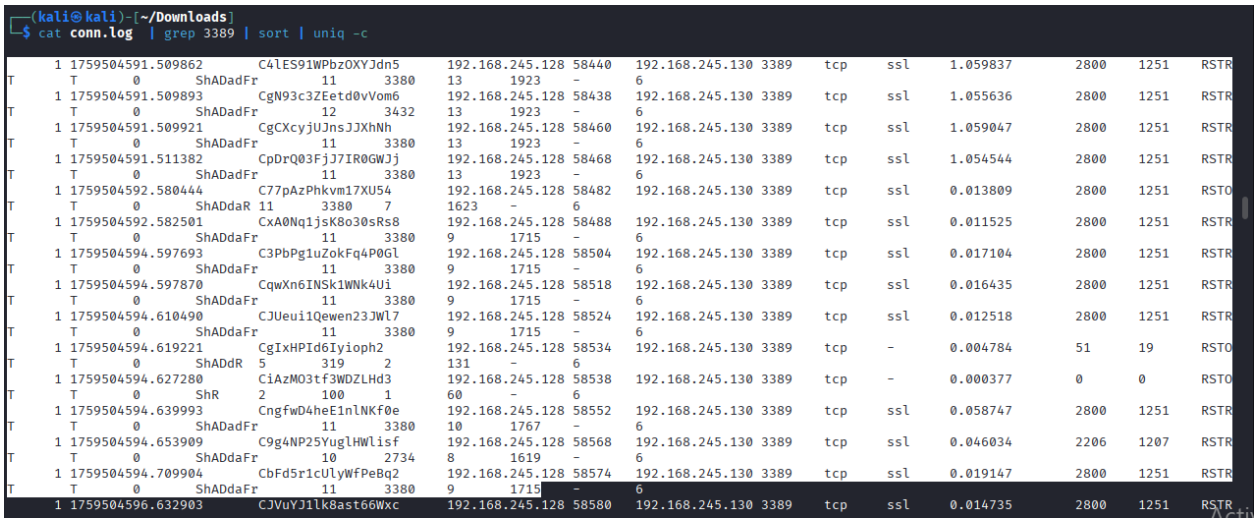



```
[3389][rdp] host: 192.168.245.130   login: Administrator   password: Cyber@2025
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-10-03 11:17:06
```




splunk>enterprise
Apps
Administrator
1 Messages
Settings
Activity
Help
Find
Search
Analytics
Datasets
Reports
Alerts
Dashboards
Search & Reporting
New Search
Save As
Create Table View
Close
index=main source="/var/log/suricata/eve.json"
Time range: Last 24 hours
3,973 events (10/2/25 11:00:00.000 AM to 10/3/25 11:21:09.000 AM)
No Event Sampling
Job
Smart Mode
Events (3,973)
Patterns
Statistics
Visualization
Timeline format
Zoom Out
Zoom to Selection
Deselect
1 hour per column
Format
Show: 20 Per Page
View: List
Prev
1 2 3 4 5 6 7 8 ... Next
Hide Fields
All Fields
SELECTED FIELDS
a host 1
a source 1
a sourcetype 1
INTERESTING FIELDS
a alert.action 1
a alert.category 1
alert.gid 1
alert.rev 1
Time
Event
10/3/25
11:21:06.432 AM
{ [-]
app_proto: dns
dest_ip: 192.168.245.1
dest_port: 53
event_type: flow
flow: { [+]
}
flow_id: 1936561286743974
in_iface: eth0
proto: UDP


Hide Fields
All Fields
Format
Show: 20 Per Page
View: List
Prev
1 2 3 4 5 6 7 8
INTERESTING FIELDS
a alert.action 1
a alert.category 1
alert.gid 1
alert.rev 1
alert.severity 1
a alert.signature 1
alert.signature_id 1
a app_proto 1
date_hour 1
date_mday 1
date_minute 8
a date_month 1
date_second 50
a date_wday 1
date_year 1
date_zone 1
a dest_ip 3
dest_port 100+
a direction 1
a event_type 3
flow.bytes_toclient 75
flow.bytes_toserver 88
a flow.dest_ip 1
flow.dest_port 1
flow.pkts_toclient 24
flow.pkts_toserver 28
a flow.src_ip 1
flow.src_port 100+
Time
Event
dest_port
>100 Values, 100% of events
Selected
Yes
No
Reports
Average over time
Maximum value over time
Minimum value over time
Top values
Top values by time
Rare values
Events with this field
Avg: 6424.666495110654 Min: 80 Max: 58658 Std Dev: 11535.205342957266
Top 10 Values
Count
%
3389 3,622 93.206%
41200 2 0.051%
41208 2 0.051%
80 2 0.051%
32780 1 0.026%
32788 1 0.026%
32792 1 0.026%
32794 1 0.026%
32810 1 0.026%
32822 1 0.026%
-too_small
127.0.0.1:8000/en-US/app/search/search?q=...est=-24h@h&latest=now&



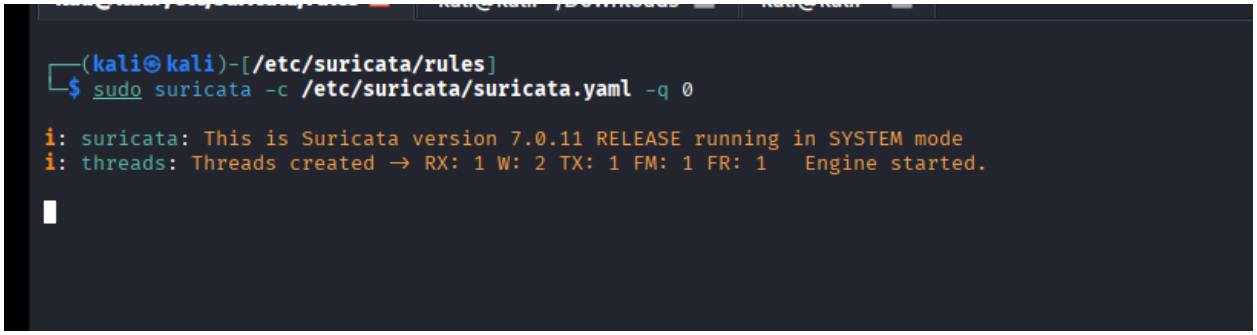
3. Containment Phase

Objective

The goal of the containment phase was to stop the active RDP brute-force attack while maintaining evidence for further forensic analysis. Since detection alone was insufficient, Suricata was transitioned from an Intrusion Detection System (IDS) to an Intrusion Prevention System (IPS) to actively block malicious traffic in real time.

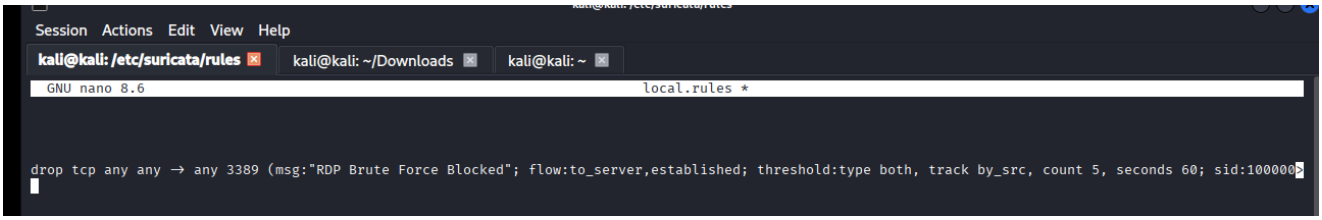
Containment Actions

- **Suricata in IPS Mode**
  - Reconfigured Suricata to run in IPS mode using NFQUEUE.
  - This enabled packet interception and blocking, preventing malicious traffic from reaching the victim.

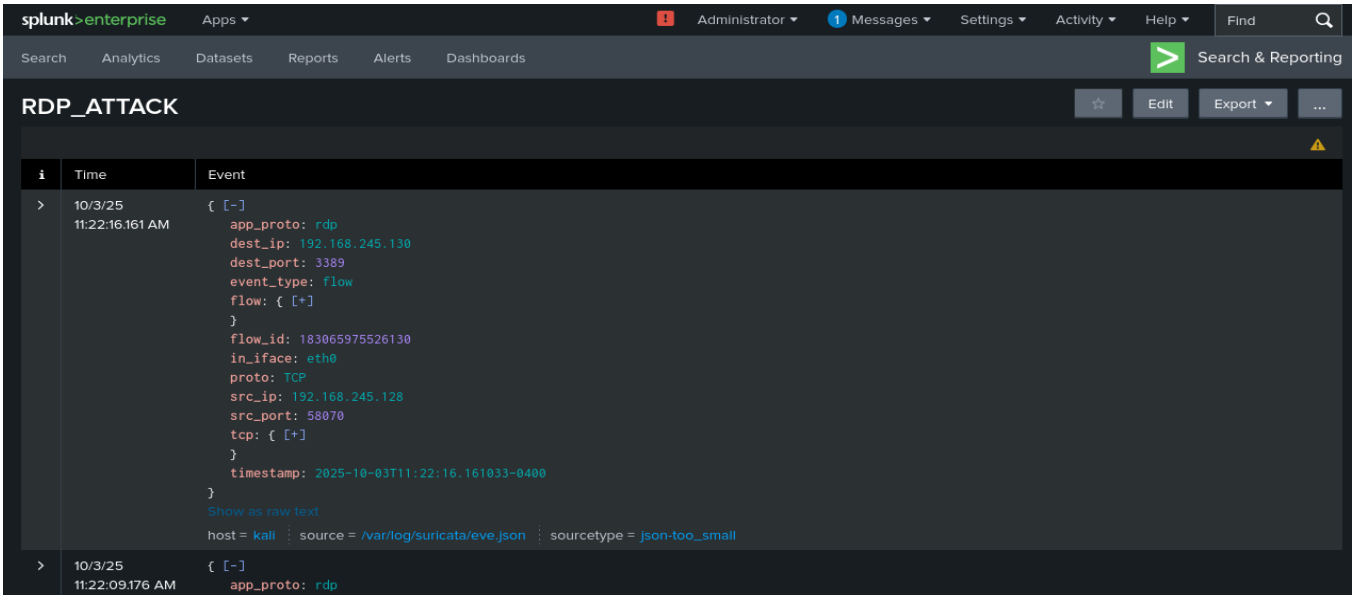


- **Custom Suricata Rule for RDP Brute Force**
  - A drop rule was added to detect repeated failed login attempts on TCP port 3389 (RDP).
  - The rule enforced a threshold of **5 attempts per 60 seconds** from the same source, automatically blocking the attacker.

Rule Example:



- **Log Monitoring and Validation**
  - Verified Suricata logs (fast.log and eve.json) to confirm that packets were being dropped.
  - Used Splunk to visualize and monitor drop events in real-time.



- **Independent Verification**
  - Ran Wireshark on the victim side to confirm that blocked RDP attempts did not reach the target host.
  - Used Zeek for session analysis to ensure no valid RDP connections were established after containment.

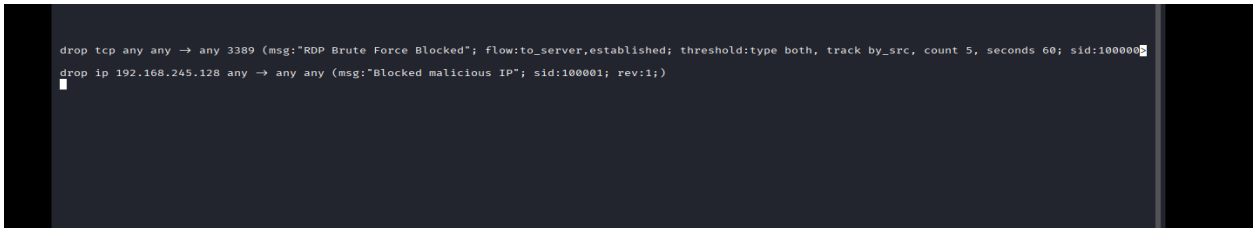
4. Eradication Phase

Objective

The main objective of the eradication phase is to completely remove the attacker's access from the environment and ensure that no persistence mechanisms or vulnerabilities remain that could allow re-entry.

Actions Taken

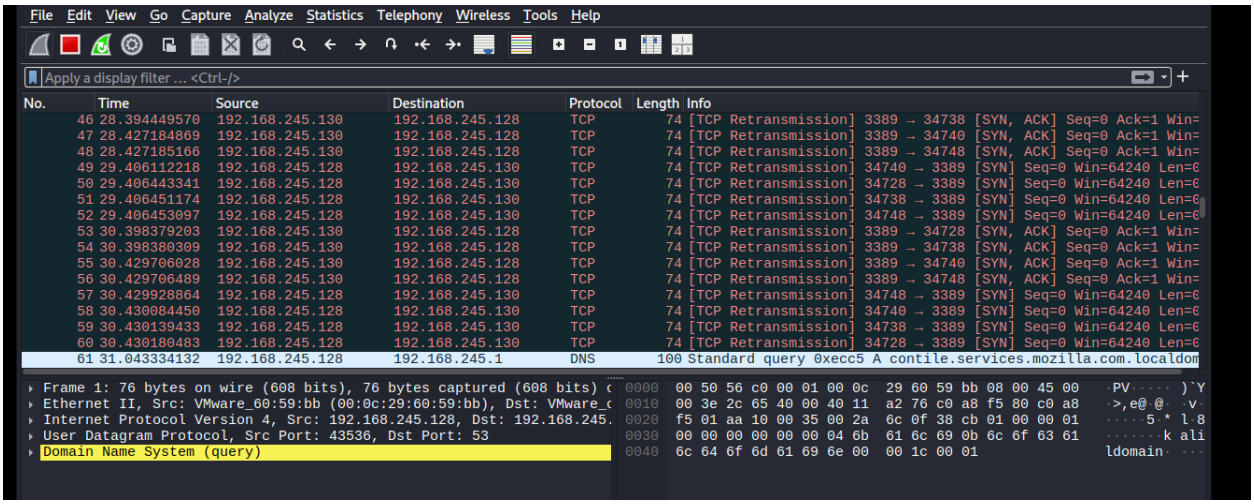
- **Blocked malicious IP addresses** identified during the brute-force attempts by creating custom Suricata IPS rules.
- **Disabled weak or unused services** to reduce the attack surface.
- **Applied security patches** to address known vulnerabilities on the target system.
- **Reviewed accounts and credentials** to ensure no unauthorized access remains.



Verification

- Suricata logs (eve.json) were checked to confirm that traffic from attacker IPs was dropped.

```
┌──(kali㉿kali)-[~/Downloads]
└─$ hydra -l Administrator -P /opt/wordlists/list.txt rdp://192.168.245.130

Hydra v9.6 (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 ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-10-03 12:20:29
[WARNING] rdp servers often don't like many connections, use -t 1 or -t 4 to reduce the number of parallel connections and -W 1 or -W 3 to wait between conn
ection to allow the server to recover
[INFO] Reduced number of tasks to 4 (rdp does not like many parallel connections)
[WARNING] the rdp module is experimental. Please test, report - and if possible, fix.
[DATA] max 4 tasks per 1 server, overall 4 tasks, 58 login tries (l:1/p:58), ~15 tries per task
[DATA] attacking rdp://192.168.245.130:3389/
[ERROR] freerdp: The connection failed to establish.
[ERROR] freerdp: The connection failed to establish.
[ERROR] freerdp: The connection failed to establish.
[ERROR] freerdp: The connection failed to establish.
[ERROR] all children were disabled due too many connection errors
0 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-10-03 12:21:01
```



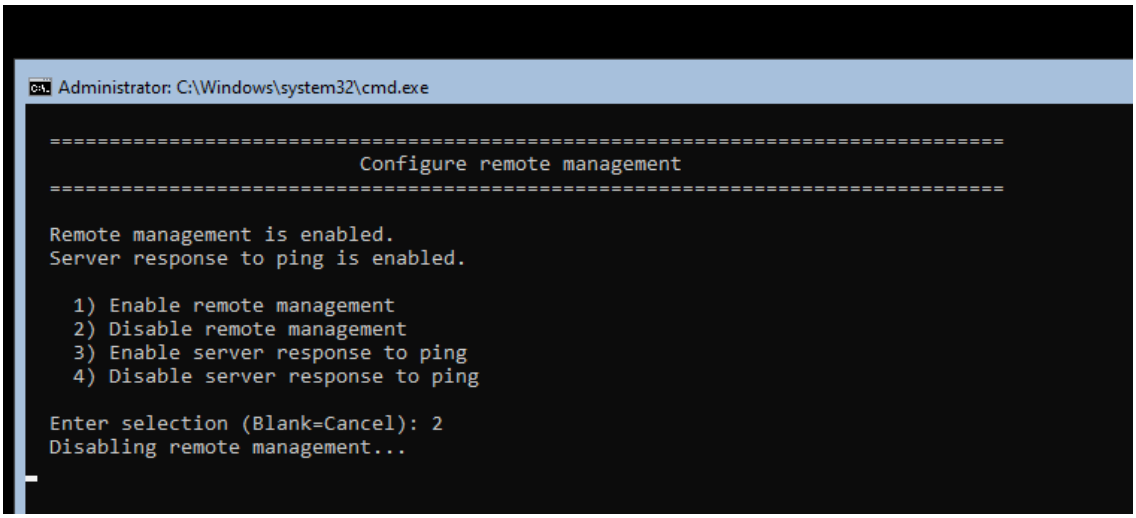
5.Recovery Phase

Objective

The recovery phase ensures that normal business operations are restored securely, while continuously monitoring for any signs of the attacker attempting to regain access.

Actions Taken

- **Restored affected services** after verifying no malicious activity persisted.
- **Re-enabled RDP service** on the Windows Server only after implementing stronger security controls.



- **Applied stronger authentication**:
  - Enforced complex password policies.
  - Limited RDP access to specific trusted IP addresses.
  - Implemented account lockout thresholds to stop brute-force attempts.
- **Continuous monitoring** was enabled through Suricata and Splunk to detect any anomalies during recovery.

Tools Used

- **Suricata (IPS mode)** → to drop any new brute-force attempts.
- **Splunk** → to monitor authentication logs and network activity for anomalies.
- **Wireshark** → for on-demand verification of traffic during testing.
- **Zeek** → to validate no suspicious network flows reappeared.

Verification

- Confirmed in **Splunk** that no failed RDP login attempts occurred after implementing security controls.
- Verified in **Suricata logs** that dropped traffic corresponded to attacker IPs, and no new malicious IPs appeared.
- Conducted controlled testing to ensure legitimate users could log in without disruption.

6. Lessons Learned Phase

Objective

The purpose of the lessons learned phase is to document the incident in detail, evaluate the effectiveness of the response, and recommend improvements to strengthen the organization's security posture for the future.

Incident Summary

- **Attack Type:** RDP brute-force attack.
- **Source:** Malicious IPs (e.g.192.168.245.128).
- **Detection:** Suricata (IDS) alerts captured in eve.json and forwarded to Splunk.
- **Containment:** Suricata switched to IPS mode, dropping attacker connections.
- **Eradication:** Custom Suricata rules blocked malicious IPs; unnecessary services disabled.
- **Recovery:** RDP hardened with stronger authentication and monitoring; confirmed no further attacks succeeded.

What Worked Well

- **Suricata IDS/IPS** effectively detected and blocked brute-force attempts.
- **Splunk** provided centralized visibility into alerts and authentication logs.
- **Zeek and Wireshark** supported detailed verification and analysis.
- Switching Suricata from IDS to IPS provided **real-time protection**.

What Could Be Improved

- Lack of a firewall meant all defense relied on Suricata; adding a firewall would provide layered security.
- No multi-factor authentication (MFA) was in place on RDP — this should be added to prevent brute-force risks.
- Alert forwarding to Splunk should be fine-tuned to ensure **all drops and alerts** appear without delay.

Recommendations

1. **Implement MFA** for all remote logins.
2. **Deploy a dedicated firewall** in addition to Suricata IPS for layered defense.
3. **Regular patching** of servers and services to reduce attack surface.
4. **Enforce account lockout policies** and strong password requirements.
5. **Conduct regular incident response drills** to validate procedures.
6. **Enable automated blocking** in Splunk (SOAR or correlation searches) for faster containment in the future.