# **Splunk Brute‑Force Hunting Lab**

**Short description**

This repository documents and provides configuration examples for a small hunting lab that ingests logs into **Splunk Enterprise** running on Ubuntu, with **Splunk Universal Forwarders** on Windows 10 and Kali Linux. The goal: detect and investigate brute‑force authentication attempts (password spraying, credential stuffing, repeated failed logins) using searches, dashboards, and alert examples.

## Contents

* **docs/** — setup notes and commands used to install Splunk Enterprise (Ubuntu) and Universal Forwarder (Windows 10, Kali).
* **forwarder-configs/** — example inputs.conf, outputs.conf, and app stanzas for forwarders.
* **searches/** — saved searches and detection queries for brute‑force detection.
* **dashboards/** — sample dashboard panels (XML/JSON) for visualizing failed logins.
* **LICENSE —** I used a free trail of the Splunk.

## Architecture overview

* **Splunk Enterprise (Ubuntu)** — single-node indexer + search head for the lab.
* **Universal Forwarder (Windows 10)** — forwards Windows Security logs, Application logs, Setup logs (Event etc.).
* **(Kali Linux)** — this is an hunting machine to do attacks and generate logs for the win10.
* Forwarders send data to Splunk indexer on the forwarding port (default 9997).

## Goals / Use cases

* Detect high volumes of failed authentication attempts from a single source IP.
* Detect many usernames tried from one IP (password spraying / credential stuffing).
* Detect many failed attempts against a single account from multiple sources (account‑targeting).
* Detect success after a burst of failures (likely compromise).

## Prerequisites

* Ubuntu server (recommended LTS) for Splunk Enterprise. Splunk 8.x / 9.x compatible — use the correct package for your OS.
* Windows 10 machine(s) with Splunk Universal Forwarder installed and configured to forward Security logs.
* Kali Linux for hunting.
* Network connectivity: forwarders must reach the Splunk indexer on the configured port (default 9997).
* Splunk admin or power user account to create saved searches, alerts and dashboards.

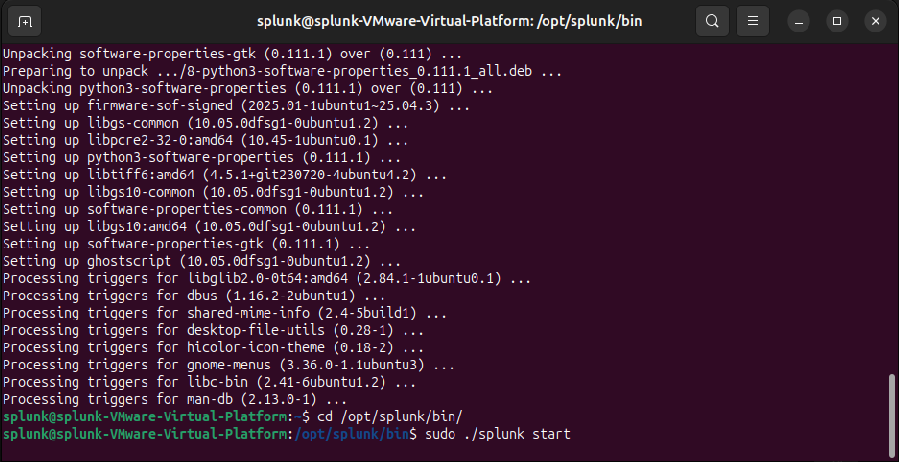
## **Quick install & configuration (lab)**

### Splunk Enterprise (Ubuntu) — quick steps

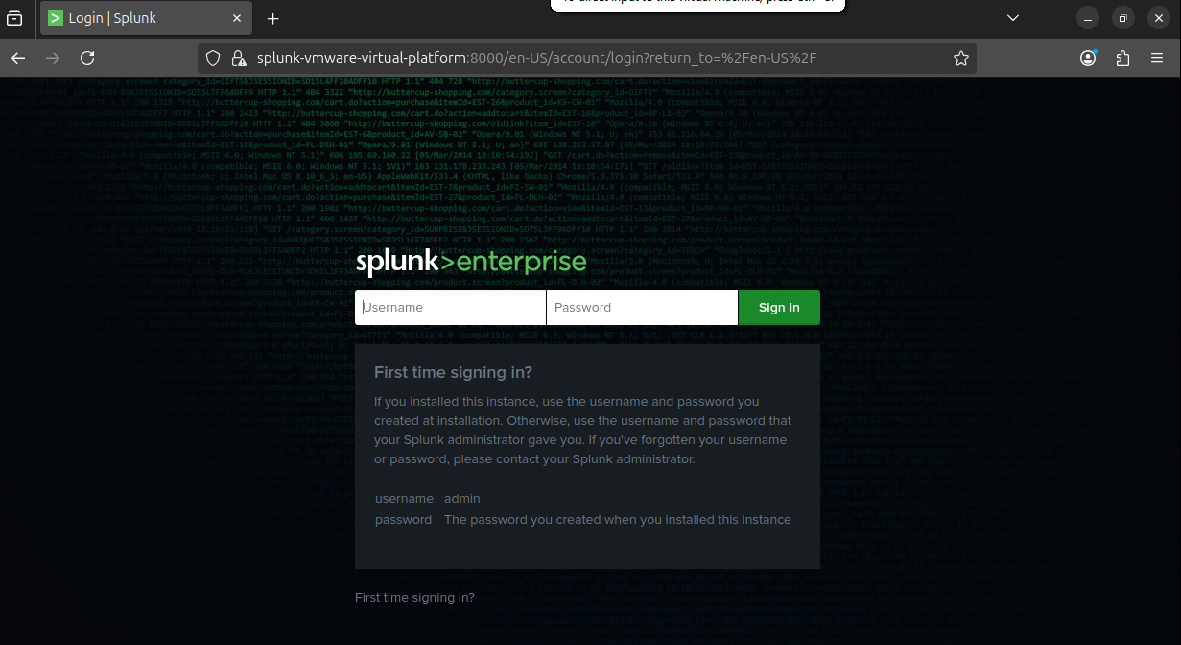
1. Download Splunk Enterprise package for Linux and place the package on the Ubuntu host.
2. Install (example for Debian/Ubuntu .deb):

sudo dpkg -i /tmp/splunk\_package\_name.deb  
sudo /opt/splunk/bin/splunk start --accept-license  
sudo /opt/splunk/bin/splunk enable boot-start

1. Open web UI: https://<splunk-host>:8000 and log in with admin credentials.
2. Enable receiving on indexer: *Settings → Forwarding and receiving → Receive data → New TCP input* (port 9997).
3. Started the Splunk with the **sudo ./splunk start command**

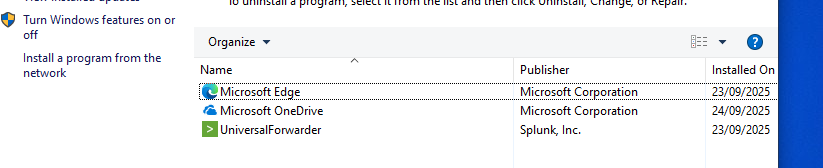


Splunk Login page



### Windows 10 Universal Forwarder (UF)

1. Download Splunk Universal Forwarder for Windows and install as admin.



1. Configure outputs.conf to point to the indexer (example location: C:\Program Files\SplunkUniversalForwarder\etc\system\local\):

[tcpout]  
defaultGroup = indexers  
  
[tcpout:indexers]  
server = splunk-indexer.example.local:9997

1. Enable Windows Security event monitoring (example inputs.conf):

[WinEventLog://Security]  
disabled = false

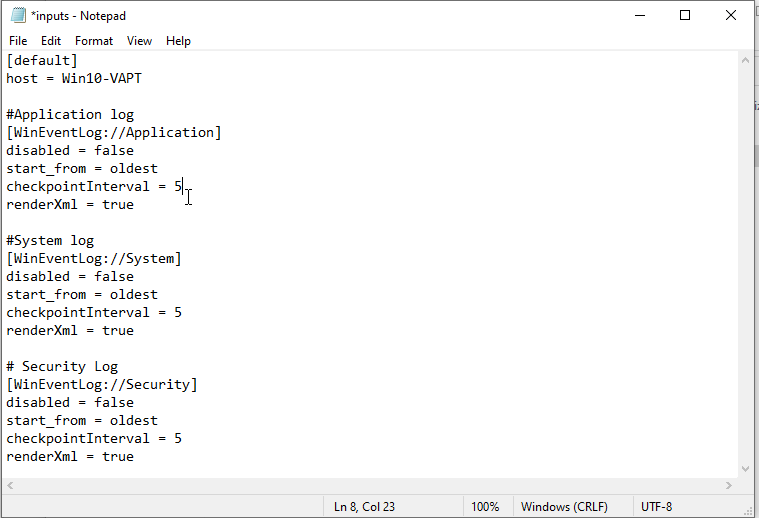
Start\_from= oldest  
renderXml = true  
checkpointInterval = 5

[WinEventLog://Setup]  
disabled = false

Start\_from= oldest  
renderXml = true  
checkpointInterval = 5

[WinEventLog://Application]  
disabled = false

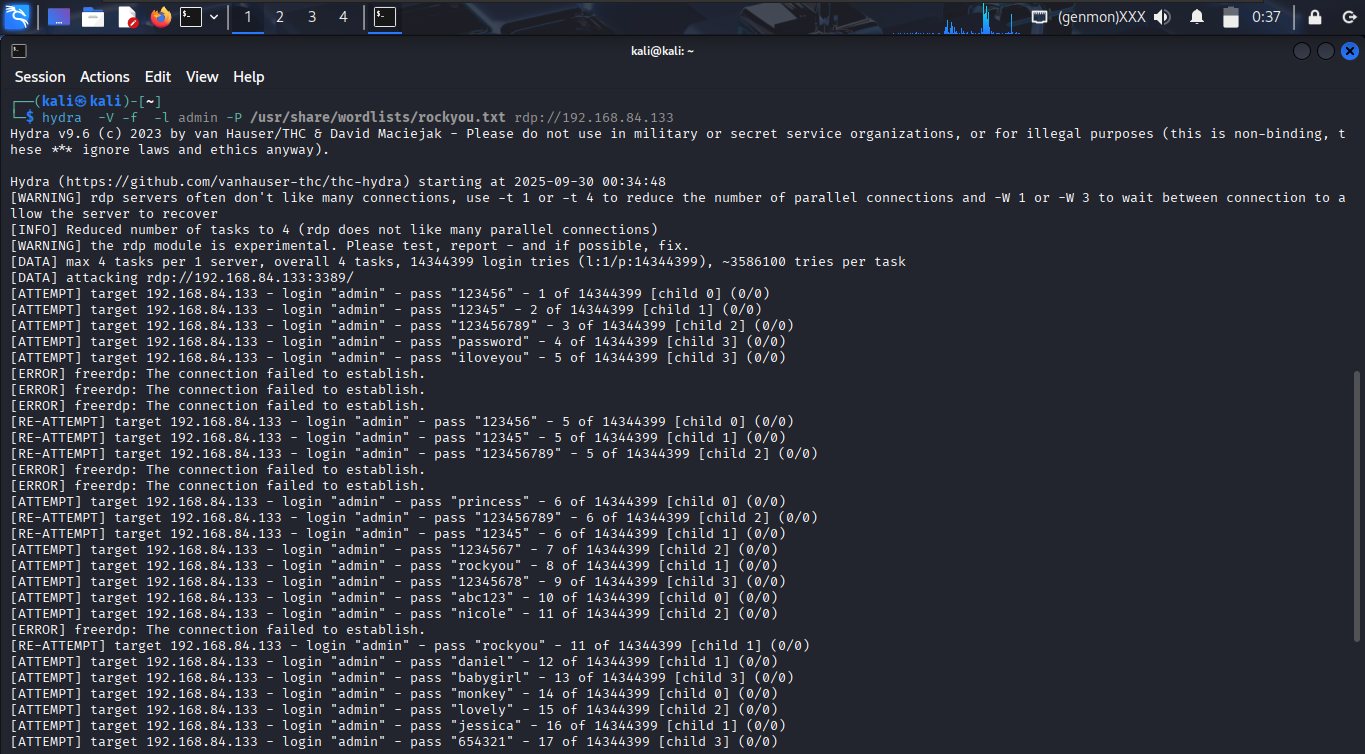
Start\_from= oldest  
renderXml = true  
checkpointInterval = 5



1. Restart the UF service.

### Kali Linux

1. Download the .ovm file from the <https://ww.kali.org>
2. Import that in the VMware and turn it on
3. Update the kali linux with the below command cmd = **sudo apt update && sudo apt upgrade –y**
4. Then use the hydra tool for the burt force attack
5. Command to use hydra cmd = **hydra -V –f -l admin –P /usr/share/wordlists/rockyou.txt rdp://<ip>**



1. Hit enter
2. After 2-minute press **clt+c** to exit stop the attack

## Recommended Splunk searches (detections)

Replace index=”\*” and sourcetype=... to match your environment.

1. **Many failed logins from same source IP**

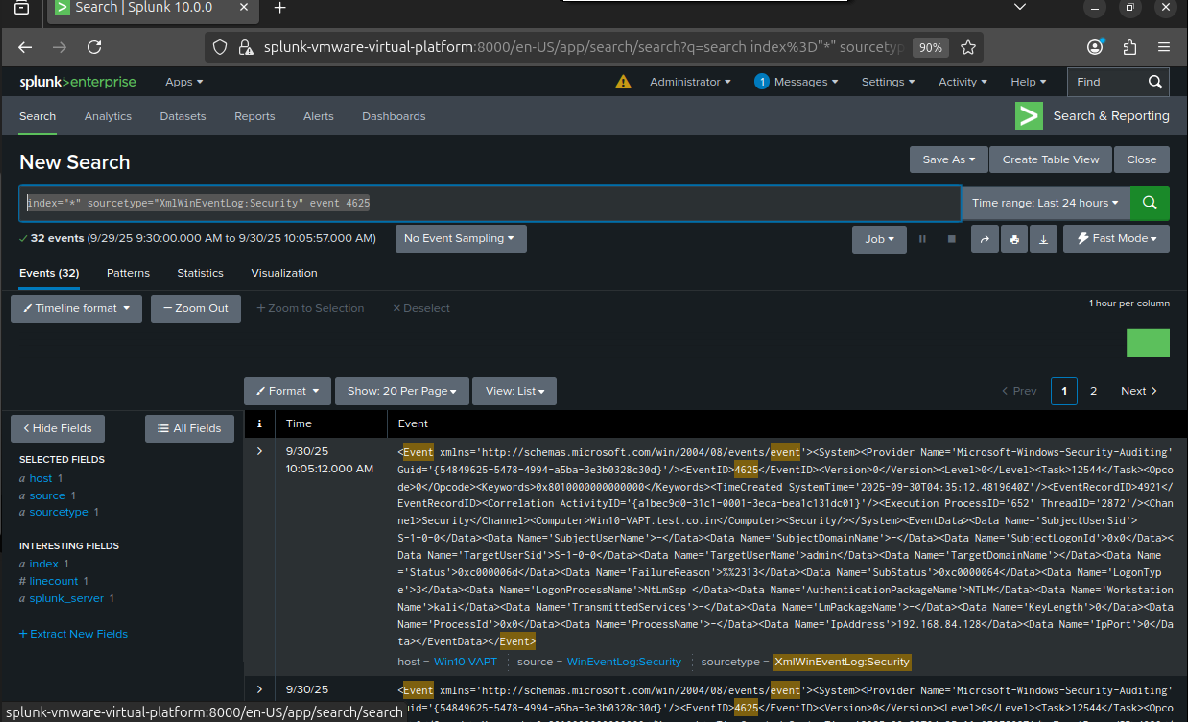
index=os (sourcetype=linux\_auth OR sourcetype="WinEventLog:Security")  
("failed password" OR "failure" OR "Authentication failed" OR EventCode=4625)  
| eval src=coalesce(src\_ip, src, IpAddress)  
| stats count AS failed\_count dc(user) AS distinct\_usernames latest(\_time) AS lastSeen by src  
| where failed\_count >= 10  
| sort -count

1. **Password spraying / many usernames from one source**

index=os (sourcetype=linux\_auth OR sourcetype="WinEventLog:Security")  
("failed password" OR "invalid user" OR EventCode=4625)  
| eval src=coalesce(src\_ip, src, IpAddress)  
| stats count AS attempts dc(user) AS unique\_users by src  
| where unique\_users >= 5 AND attempts >= 20

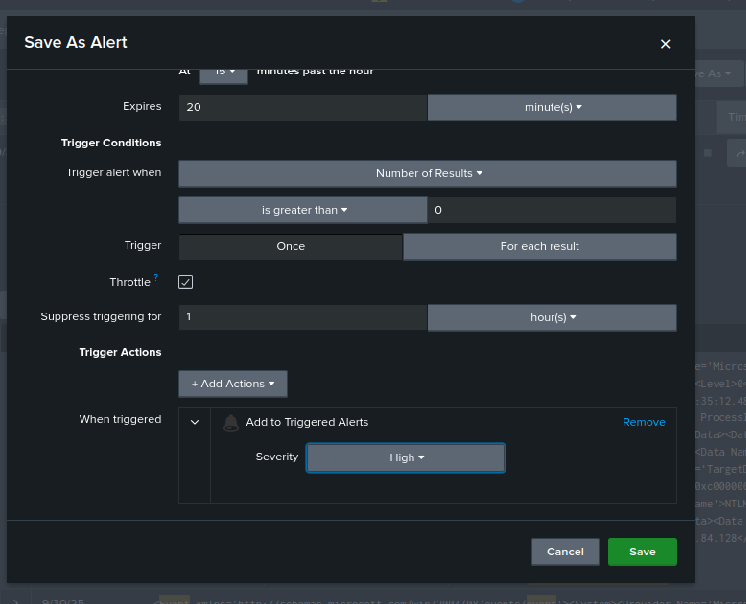
1. **Brute‑force followed by a success (possible compromise)**

index=os (sourcetype=linux\_auth OR sourcetype="WinEventLog:Security")  
| eval status=case(match(\_raw,"Failed|failure|failed password|authentication failure"),"FAIL",  
 match(\_raw,"Accepted|success|Logon Type"),"SUCCESS",1=1,"OTHER")  
| streamstats window=50 current=t count AS seq by user  
| eventstats min(if(status="FAIL",\_time,null())) AS firstFail by user  
| where status="SUCCESS" AND \_time - firstFail <= 600  
| table user src \_time firstFail \_raw

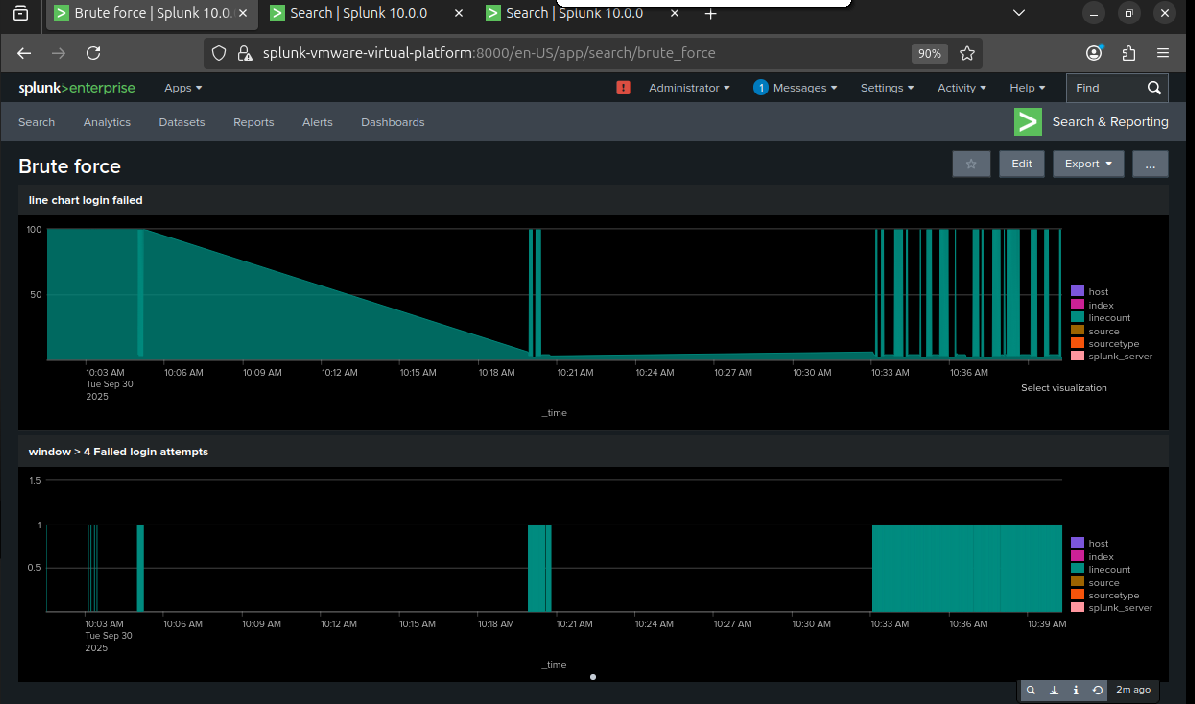


## Alerts & dashboard suggestions

* **High severity alert**: failed\_count >= 50 in 10 minutes from one src → trigger alert /email alert/ incident.

****

* **Medium severity**: distinct\_usernames >= 10 AND attempts >= 30 → create investigation ticket.
* Dashboard panels:
  + Top source IPs by failed attempts (last 24h).
  + Timeline: failed vs successful authentication attempts.
  + Top targeted accounts.
  + Geo map of sources (GeoIP).

****

## Investigative playbook (brief)

1. **Triage** — note source IP, targeted accounts, timestamps and counts.
2. **Enrich** — GeoIP, reverse DNS, threat intel lookups.
3. **Check pivot** — any successful logins? which hosts and times?
4. **Contain** — block IP(s) at perimeter or WAF if confirmed malicious.
5. **Remediate** — reset compromised credentials, enable/require MFA, rotate keys.
6. **Document & learn** — update detection thresholds, add suppressions to reduce false positives.



**This is only for the education purpose don’t use this for any illegal activity**