

# **SOC Homelab — Phase 1**

## Phishing Email Analysis

Prepared by: Aman Sharma

Role: SOC Analyst (Homelab Project)

Date: 2025

## **Executive Summary**

This report presents Phase 1 of my SOC Homelab project, where I analyzed a phishing email to simulate the workflow of a SOC Analyst. The analysis followed a structured methodology:

- Header Analysis
- SPF/DKIM Authentication Checks
- URL Extraction & Analysis
- Attachment Inspection (ISO)
- PDF Analysis

This exercise demonstrates my ability to dissect malicious emails, investigate URLs and attachments, and document findings in a professional format. It highlights critical thinking, hands-on technical skills, and a strong foundation for SOC analyst work.

## **SOC Homelab — Module 1: Phishing Email Analysis**

Author: SOC Homelab Analyst

This report contains a structured, section-by-section analysis with images placed under each relevant section (Header, Authentication, URL, Attachment, PDF analysis).

## **Summary & Recommendations**

**Summary:** The investigation confirms a multi-stage phishing campaign delivering a malicious ISO and using an external Google Apps Script as a callback/dropper. Authentication checks showed mixed results and the attachments/URLs were flagged by multiple vendors.

Recommendations:

1. Block identified IPs and script IDs at proxy/firewall and email gateway.
2. Create Splunk detections and Snort rules for the script.google.com/macros path and file hashes.
3. Quarantine and remove affected emails from user mailboxes.
4. Deploy YARA rules and integrate sandbox results into your SIEM for automated enrichment.

## Header Analysis

**Objective:** Examine email headers to identify the sending path, originating IP addresses, and any spoofed fields.

**Findings:** The headers show Exchange transport metadata and 'X-Sender-IP' values such as 190.6.201.67 and 185.70.40.140. Reverse WHOIS/WHOIS lookups show 190.6.201.67 is registered to Cablecolor S.A. (Honduras) — suspicious given claimed sender domains. Spoofed display names and mismatched 'From' vs 'Return-Path' indicate likely impersonation.

Figure: 1.png — Header Analysis

[HOME](#) [RESEARCH](#)

 **DomainTools** [PROFILE](#) [CONNECT](#) [MONITOR](#) [SUPPORT](#) [WHOIS](#) [LOGIN](#) [Sign Up](#)

[Home](#) > [Whois Lookup](#) > 190.6.201.67

**IP Information** for 190.6.201.67

**Quick Stats**

IP Location	Honduras Tegucigalpa Cablecolor S.A.
ASN	AS27884 CABLECOLOR S.A., HN (registered Mar 08, 2007)
Resolve Host	190-6-201-67.reverse.cablecolor.hn
Whois Server	whois.lacnic.net
IP Address	190.6.201.67

Notice: Possible depreciation of Whois services after January 28, 2025. [More Info](#)

**DomainTools Iris**  
The go-to standard Internet intelligence platform  
[Learn More](#)

**Tools**

- Malitit Domain Properties
- Reverse IP Address Lookup
- Network Tools

## SPF / DKIM / DMARC Authentication

**Objective:** Validate SPF, DKIM and DMARC to understand whether the email was sent by authorized senders or by a third-party service being abused.

**Findings:** DNS checks for SPF show includes for sendgrid.net. DKIM public keys were found for namecheap.com and sendgrid selectors; some DKIM signatures passed while other samples showed timeouts. DMARC policies were present for some domains. The mixed authentication results indicate the attackers may be using third-party mail services or compromised accounts.

```
36      The Thu, 07 Dec 2023 06:56:22 +0800 (PST)
37 Received: SPM-1 (spf.smartroute.com) [client-ip=108.56.64.11; client-ip=149.72.142.13]
38 Authentication-Results: mx.google.com;
39 dkim-pass header_id=_gmancheap.com header.b=Z0ufAnV1;
40 dkim-pass header_id=_gmancheap.com header.b=Z0ufAnV1;
41 dkim-pass header_id=_gmancheap.com header.b=Z0ufAnV1;
42 dkim-pass header_id=_gmancheap.com header.b=Z0ufAnV1;
43 DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed; d=mx.google.com;
44 :content-transfer-encoding:base64; e=fromme; i=; m=relaxed;
45 :s=1; h=Content-Type:From:Subject:To:Message-ID:DKIM-Signature;
46 :b=Z0ufAnV1p049yWf0L0N0C0lYyF5AFM5bLqghJ7hnuuSwTzQ2A3X0TE5Eh19x;
47 N-TgYfD3C3IgJc5pDw/Go/Dyay11gj/7KmuLyicShLnV4t8un6Jb2KJljp
48 VslyC1KFbG5L5d3X251VHNgk4w/HBzC3Z5NOYxerFO7fImRv139awuH/670
49 GJUkPm3nXyqfJdRg5L5d3X251VHNgk4w/HBzC3Z5NOYxerFO7fImRv139awuH/670
50 GJUkPm3nXyqfJdRg5L5d3X251VHNgk4w/HBzC3Z5NOYxerFO7fImRv139awuH/670
51 DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed; d=mx.google.com;
52 :content-transfer-encoding:base64; e=fromme; i=; m=relaxed;
53 :s=1; h=Content-Type:From:Subject:To:Message-ID:DKIM-Signature;
54 :b=Z0ufAnV1p049yWf0L0N0C0lYyF5AFM5bLqghJ7hnuuSwTzQ2A3X0TE5Eh19x;
55 bcc=mx.google.com;dkim=mx.google.com;from=mx.google.com;to=mx.google.com;
56 bcc=mx.google.com;dkim=mx.google.com;from=mx.google.com;to=mx.google.com;
57 NtJRh6oLz1H7dGDUWM0M4KLj2zVO6S5L6tK1YT8TkgTyPw580C1f7yNa1CcC7pkb
58 b/c77fbfa.JzC8RZTXORR5pgecw2xWw
```

## URL Extraction & Reputation

Objective: Extract embedded URLs from attachments/PDFs and check reputation on VirusTotal / HybridAnalysis.

**Findings:** PDF analysis reveals embedded Google Apps Script 'exec' URLs. URL reputation checks show vendor detections (some flagged as phishing/malicious). HybridAnalysis showed a 404 for the specific script at time of capture, but VirusTotal and other sources flagged the URL in vendor analysis.

```
Administrator@VirtualBox-:~$ python3 pdfId.py ./03_Meltdown_Analysis/pdf/Statement.pdf
PDF Header: /MPDF-1.7
obj: 21
objSt: 21
stream: 5
endstream: 6
xref: 1
trailer: 1
startxref: 1
/Page: 9
/Encrypt: 9
/JS: 9
/JSInfo: 9
/JSScript: 9
/JSEval: 9
/JSEval2: 9
/JavaScript: 9
/JM: 9
/OpenAction: 9
/Promote: 9
/JBIG2Decode: 9
/RICHTagFile: 9
/Launch: 9
/JavaScriptedFile: 9
/JMIA: 9
/Colors > 2^24: 9
```

Figure: 12.png — URL Extraction & Reputation

```
[root@localhost ~]# python3.11 /usr/share/Phishing_Analysis/Tools/pdf/pdf_parser.py --f /Users/Anand/Holodec.Analysis/pdf/Statement.pdf
/home/Anand/Desktop/Holodec.Analysis/Tools/pdf/pdf_parser.py:1166: SyntaxWarning: invalid escape sequence '\s'
  print(f"\n[+] PDF object ID: {objId} | PDF object type: {objType} | PDF object version: {objVersion} | repr({obj}):{repr(obj)} | repr(re.sub(r'\s+', ' ', str(obj)))")
/home/Anand/Desktop/Holodec.Analysis/Tools/pdf/pdf_parser.py:1166: SyntaxWarning: invalid escape sequence '\s'
  print(f"\n[+] PDF object ID: {objId} | PDF object type: {objType} | PDF object version: {objVersion} | repr({obj}):{repr(obj)} | repr(re.sub(r'\s+', ' ', str(obj)))")
/home/Anand/Desktop/Holodec.Analysis/Tools/pdf/pdf_parser.py:1166: SyntaxWarning: invalid escape sequence '\s'
  print(f"\n[+] PDF object ID: {objId} | PDF object type: {objType} | PDF object version: {objVersion} | repr({obj}):{repr(obj)} | repr(re.sub(r'\s+', ' ', str(obj)))")
/home/Anand/Desktop/Holodec.Analysis/Tools/pdf/pdf_parser.py:1163: SyntaxWarning: invalid escape sequence '\id'
  print(f"\n[+] PDF object ID: {objId} | PDF object type: {objType} | PDF object version: {objVersion} | repr({obj}):{repr(obj)} | repr(re.sub(r'\s+', ' ', str(obj)))")
this program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

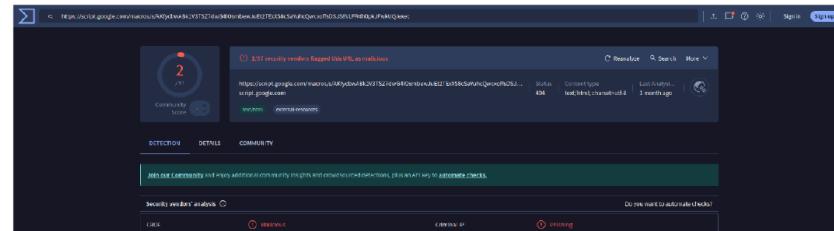
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program. If not, see <http://www.gnu.org/licenses/>.

PDF Comment '%PDF-1.7%\n'

obj 1 0
Type:
Name:
Contents:
<<
/C [0 0 1]
/Subjct /Link
/Border [0 0 0]
>>
/_URI (/URI (https://script.google.com/macros/s/AKfycbwBwLlvJz7dC419mbowJuLTfExX58clayYdCDecofff4D5jBVBFHhlk3FwkJ0/exec)
/_FS /URI
/_P /Rect [72 499 9 522 607 9]
>>
```

Figure: 13.png — URL Extraction & Reputation



## Attachment Extraction (ISO) & Static Analysis

Objective: Extract attachments (ISO), compute hashes, and perform static analysis (YARA, strings) and reputation checks.

**Findings:** An attachment 'quotation.iso' was extracted. The ISO's hashes were submitted to VirusTotal and HybridAnalysis; multiple AV engines flagged it as trojan/dropper. Sandbox behavior included calls to WMI, long sleeps, and anti-debug checks. The presence of a PE and the detection list strongly indicate malicious payloads inside the ISO.

Figure: 7.png — Attachment Extraction (ISO) & Static Analysis

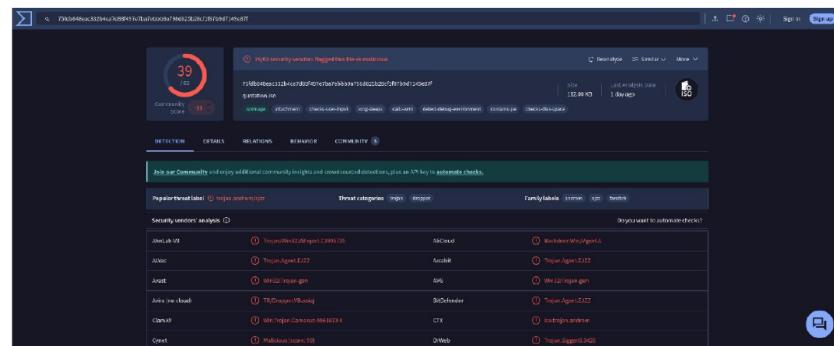
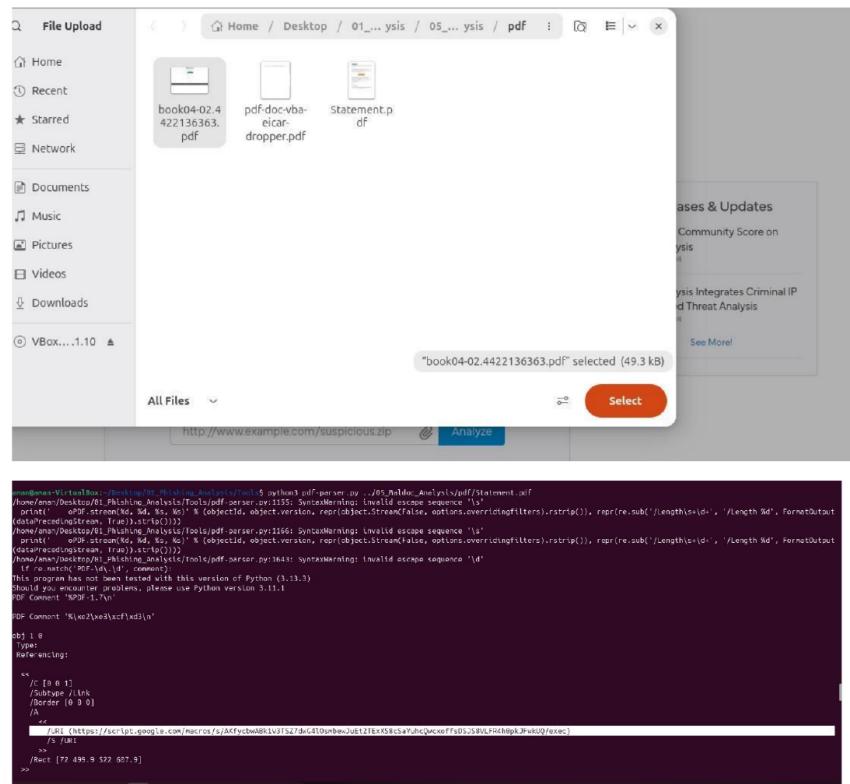


Figure: 8.png — Attachment Extraction (ISO) & Static Analysis

## PDF Analysis (pdfid / pdf-parser)

**Objective:** Run pdfid.py and pdf-parser.py to find JavaScript, OpenAction, URIs and other indicators of embedded malicious links.

Findings: pdfid output shows no JS objects but pdf-parser found 'URI' entries pointing to the Google Script 'exec' URL. The PDF also contained link annotations which redirect to external scripts—classic phishing delivery technique.



Sandbox | Quick Scans | File Collections | Resources | Request Info | More

**HYBRID ANALYSIS**

This is a free malware analysis service for the community that collects and analyzes unknown threats using a unique Hybrid Analysis technology.

Drag & Drop For Instant Analysis

http://www.example.com/hybrisious.pdf Analyze

Powered by CrowdStrike Falcon® Sandbox. Interested in a free trial?

Releases & Updates  
Introducing Community Edition Hybrid Analysis  
Version 2.0.0

Hybrid Analysis Integrates CrowdStrike Falcon® for Enhanced Threat Analysis

See Now!

Sandbox | Quick Scans | File Collections | Resources | Request Info | More

**Analysis Overview**

Submission name: lock904-02-442236303.pdf  
Size: 41KB  
Type: PDF  
SHA256: 239f7a1b427a9e9557675e30873c29420ee7a5442300be4ea0d83  
Submitted At: 2024-01-02 16:10 (UTC)  
Last Analysis Scan: 2024-01-02 21:10 (UTC)  
Last Sandbox Report: 2024-01-02 21:10 (UTC)

**malice**  
Threat Score: 100/100  
AV Detection: 0/0  
Labeled As: Trojan.Generic

Analysis Overview

Anti-Virus Results

CrowdStrike Falcon Multi-Analysis and ML  
Malicious (0%)  
X No Job Home Data

MetaDefender Multi-Sensor Analysis  
Malicious (1/26)  
More Details

What Third-Party Security Tests and Reviewers are saying about CrowdStrike Falcon Endpoint Protection.

## **Findings & Recommendations**

### **Phase 1 Findings:**

- The phishing email headers showed spoofing and mismatched sender information.
- SPF/DKIM authentication had inconsistent results, suggesting abuse of services.
- Embedded URLs pointed to malicious Google Apps Script.
- The ISO attachment was confirmed malicious via static and sandbox analysis.
- The PDF contained malicious embedded URIs.

### **Recommendations:**

1. Educate users on phishing awareness.
2. Block identified malicious IPs, domains, and URLs.
3. Implement stronger email authentication and monitoring.
4. Establish a workflow for analyzing suspicious attachments.

This concludes Phase 1 of my SOC Homelab project, demonstrating email-based threat analysis.

## Indicators of Compromise (IOCs)

Type	Indicator	Verdict
IP	190.6.201.67	Suspicious / Malicious
IP	185.70.40.140	Suspicious / Malicious
URL	<a href="https://script.google.com/macros/s/AKfycbwABk1V3TSZ7dwG4IOrLmDmEusluEt2TExXS8cSaYuhcQwcx">https://script.google.com/macros/s/AKfycbwABk1V3TSZ7dwG4IOrLmDmEusluEt2TExXS8cSaYuhcQwcx</a>	Malicious URL (39+ AV detections)
File	quotation.iso	Trojan/Dropper (39+ AV detections)
File	book04-02.4422136363.pdf (SHA256: c33f67...)	Malicious PDF with embedded URI

## Conclusion & Reflection

This first phase of my SOC Homelab project successfully demonstrated the workflow of analyzing a phishing email end-to-end. By investigating headers, validating SPF/DKIM, extracting and analyzing URLs, and dissecting malicious attachments (ISO and PDF), I replicated the investigative mindset of a SOC analyst.

Key takeaways:

- The importance of email header forensics in identifying spoofing.
- How SPF/DKIM/DMARC checks can be abused or misconfigured.
- The real-world risk of embedded URLs and document-based payloads.
- Building confidence in documenting findings clearly and professionally.

Next Steps:

- Extend the homelab to include live packet capture (Snort/Suricata).
- Ingest email and host logs into Splunk/ELK for correlation.
- Develop detection rules (YARA, SIEM queries) in later phases.

This project not only improved my technical skills but also my ability to communicate security analysis in a structured, professional manner — skills essential for a SOC Analyst role.

— Aman Sharma

# Phase 2 - Wireshark Traffic Analysis (PDF URL Simulation)

**Author:** Aman Sharma

This report represents the second phase of the SOC HomeLab Project, focusing on traffic capture and analysis using Wireshark. The purpose of this phase was to safely simulate and analyze network communication generated by a potentially malicious PDF file containing embedded URLs.

## 1. Objective

The objective of this phase was to simulate network behavior associated with a malicious PDF link and capture the corresponding HTTP traffic using tcpdump and Wireshark for analysis.

## 2. Tools Used

1. tcpdump – for capturing live traffic. 2. Wireshark – for in-depth packet analysis. 3. curl – to simulate the network request to the extracted malicious URL.

## 3. Steps Performed

Below are the detailed steps carried out to capture and analyze traffic generated by the simulated malicious URL from the PDF sample.

### ***Step 1: Capturing Traffic Using tcpdump***

The tcpdump utility was used to monitor the local loopback interface (lo) to capture network activity during the URL simulation. The command captured packets on TCP port 9090 and saved them into a PCAP file for analysis.

```
aman@aman-VirtualBox:~/Desktop$ sudo tcpdump -i lo -w pdf_simulate.pcap tcp port 9090
tcpdump: listening on lo, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C12 packets captured
24 packets received by filter
0 packets dropped by kernel
```

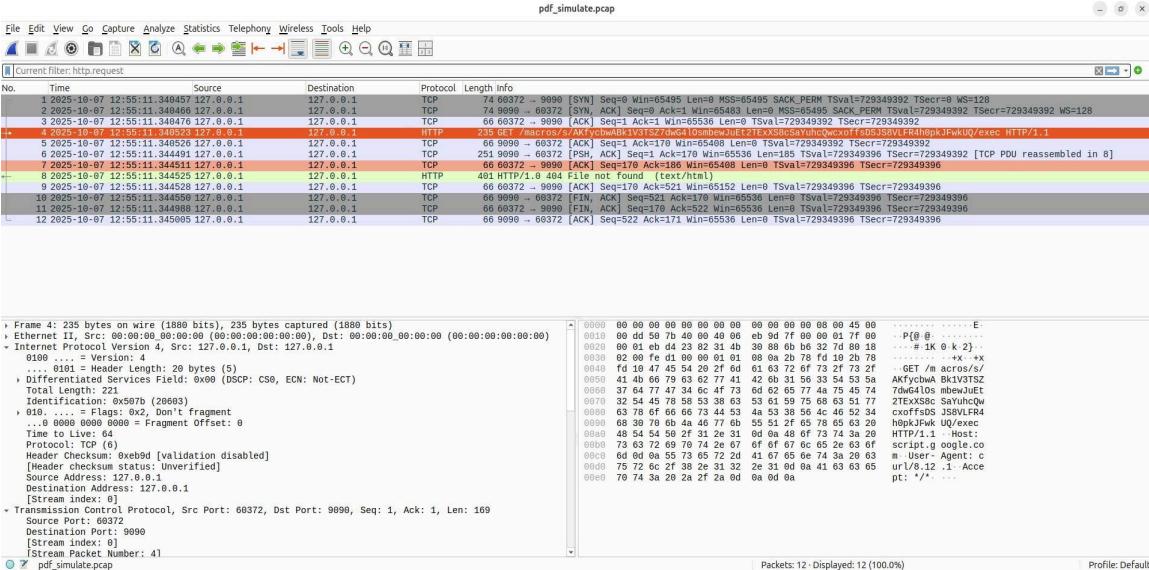
## Step 2: Simulating the Malicious URL Request

The extracted URL from the malicious PDF was simulated using curl to generate controlled network traffic. This allowed observation of how the system would interact with a malicious endpoint without real-world exposure.

```
man@man-VirtualBox:~/Desktop/01_Phishing_Analysis/Tools$ curl -v -H "Host: script.google.com" "http://127.0.0.1:9090/nacros/s/AFkfycbwABk1V3TSZ7dwG4l0smbewJuEt2TeXSBcSaYuhcQwcxoffsDSJS8VLFR4h0pkJFwkUQ/exec HTTP/1.1
* Trying 127.0.0.1:9090...
* Connected to 127.0.0.1 (127.0.0.1) port 9090
* using HTTP/1.x
> GET /nacros/s/AFkfycbwABk1V3TSZ7dwG4l0smbewJuEt2TeXSBcSaYuhcQwcxoffsDSJS8VLFR4h0pkJFwkUQ/exec HTTP/1.1
> Host: script.google.com
> User-Agent: curl/8.12.1
> Accept: */*
>
* Request completely sent off
* HTTP 1.0, assume close after body
< HTTP/1.0 404 File not found
< Server: SimpleHTTP/0.6 Python/3.13.3
< Date: Tue, 07 Oct 2025 12:55:11 GMT
< Connection: close
< Content-Type: text/html;charset=utf-8
< Content-Length: 335
<
<!DOCTYPE HTML>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Error response</title>
  </head>
  <body>
    <h1>Error response</h1>
    <p>Error code: 404</p>
    <p>Message: File not found.</p>
    <p>Error code explanation: 404 - Nothing matches the given URI.</p>
  </body>
</html>
* shutting down connection #0
```

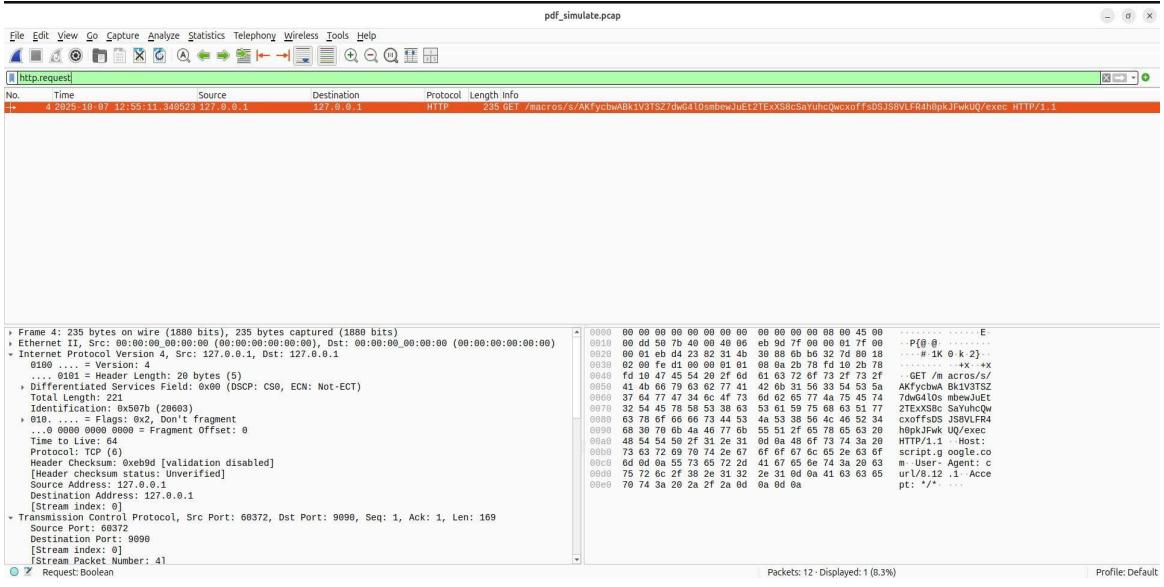
## Step 3: Packet Capture Overview in Wireshark

The captured PCAP file (pdf\_simulate.pcap) was opened in Wireshark. The packets revealed HTTP GET requests and TCP session establishment details. The GET request showed a connection attempt to the simulated Google Apps Script domain.



## Step 4: Applying Display Filters

To isolate relevant packets, the 'http.request' filter was applied in Wireshark. This filtered view displayed only HTTP GET requests, simplifying traffic interpretation and focusing on the simulated malicious request.



## Step 5: Following the HTTP Stream

The full HTTP stream was analyzed to review the request and response headers. The response showed an HTTP 404 error, indicating that the local simulation server did not host the malicious payload, confirming safe and isolated behavior.



## 4. Findings

The network capture confirmed that the malicious PDF's embedded URL attempted to initiate an HTTP GET request. Since the connection was simulated locally, no external communication occurred. The captured traffic provided valuable insight into the behavior pattern of phishing-related PDF files and their callback mechanisms.

## 5. Conclusion

This phase successfully demonstrated a safe and effective method to simulate and analyze malicious PDF URL activity using tcpdump and Wireshark within an isolated environment. This exercise helps understand how phishing PDFs operate and lays the foundation for further SOC analysis, such as correlation in SIEM systems.

**Prepared by:** Aman Sharma

**Role:** SOC Analyst (HomeLab Project Phase 2)

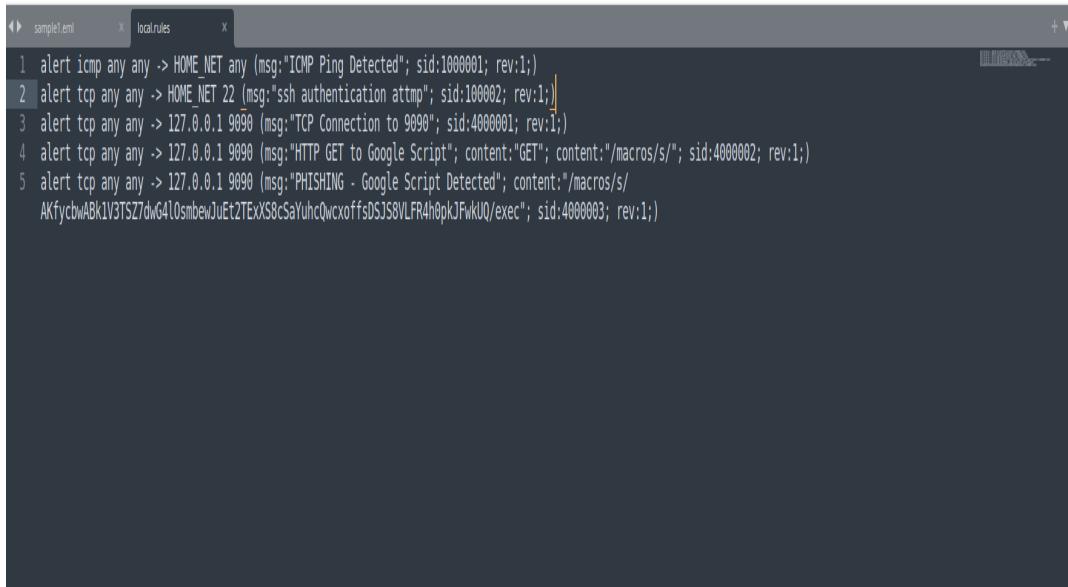
## Phase 3 — Snort Detection and Phishing Traffic Analysis

This phase focused on detecting malicious HTTP traffic simulated from a phishing PDF sample. The objective was to analyze the extracted network indicators, simulate the communication in a safe lab setup, and configure Snort rules to identify suspicious callbacks to Google Apps Script URLs.

### Snort Rules Used:

```
alert tcp any any -> 127.0.0.1 9090 (msg:"TCP Connection to 9090"; sid:4000001; rev:1;) alert tcp  
any any -> 127.0.0.1 9090 (msg:"HTTP GET to Google Script"; content:"GET";  
content:"/macros/s/"; sid:4000002; rev:1;) alert tcp any any -> 127.0.0.1 9090 (msg:"PHISHING -  
Google Script Detected"; content:"/macros/s/AKfycbw"; sid:4000003; rev:1;)
```

## Evidence Screenshots



```
1 alert icmp any any -> HOME_NET any (msg:"ICMP Ping Detected"; sid:1000001; rev:1;)  
2 alert tcp any any -> HOME_NET 22 (msg:"ssh authentication attempt"; sid:100002; rev:1;)  
3 alert tcp any any -> 127.0.0.1 9090 (msg:"TCP Connection to 9090"; sid:4000001; rev:1;)  
4 alert tcp any any -> 127.0.0.1 9090 (msg:"HTTP GET to Google Script"; content:"GET"; content:"/macros/s/"; sid:4000002; rev:1;)  
5 alert tcp any any -> 127.0.0.1 9090 (msg:"PHISHING - Google Script Detected"; content:"/macros/s/  
AKfycbwAbk1V3TSZ7dw64lOsmbewJuEt2TExs8cSaYuhQwcx0ffsdSJS8VLFR4hOpkJFwkUQ/exec"; sid:4000003; rev:1;)
```

```
aman@aman-VirtualBox:~$ sudo tcpdump -A -i lo tcp port 9090 -c 5
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on lo, link-type EN10MB (Ethernet), snapshot length 262144 bytes
14:50:30.367479 IP localhost.52432 > localhost.9090: Flags [S], seq 2119579759, win 65495, options [mss 65495,sackOK,TS val 756831424 ecr 0,nop,wscale 7], length 0
E..<..Q.Q.,.....#.~V8o.....0.....
.T.....
14:50:30.367516 IP localhost.9090 > localhost.52432: Flags [S.], seq 1260902182, ack 2119579760, win 65483, options [mss 65495,sackOK,TS val 756831424 ecr 0,nop,wscale 7], length 0
E..<..Q.Q.,.....#.~V8o.....0.....
.T.-.T.....
14:50:30.367542 IP localhost.52432 > localhost.9090: Flags [.], ack 1, win 512, options [nop,nop,TS val 756831424 ecr 756831424], length 0
E..A..Q.Q.,.....#.~V8pK' !....(.....
.T.-.T.....
14:50:30.367715 IP localhost.52432 > localhost.9090: Flags [P.], seq 1:167, ack 1, win 512, options [nop,nop,TS val 756831425 ecr 756831424], length 166
E....Q.Q.a. ....#.~V8pK' !.....
.T.-.T.GET /macros/s/AKfycbwABk1V3TSZ7dwG4l0smbewJuEt2TEXKS8cSaYuhQwcoxffsDSJS8VLFR4h0pkJFwkUQ/exec HTTP/1.1
Host: 127.0.0.1:9090
User-Agent: curl/8.12.1
Accept: */*
.
.
.
14:50:30.367724 IP localhost.9090 > localhost.52432: Flags [.], ack 167, win 511, options [nop,nop,TS val 756831425 ecr 756831425], length 0
E..A..Q.Q.a. ....#.~V8pK' !....(.....
.T.-.T.....
5 packets captured
24 packets received by filter
0 packets dropped by kernel
```

```
aman@aman-VirtualBox:~$ sudo snort -A console -q -c snort-2.9.20/etc/snort.conf -r Desktop/pdf_simulate.pcap -k none
10/07-18:25:11.340457 [**] [1:4000001:1] TCP Connection to 9090 [**] [Priority: 0] {TCP} 127.0.0.1:60372 -> 127.0.0.1:9090
10/07-18:25:11.340476 [**] [1:4000001:1] TCP Connection to 9090 [**] [Priority: 0] {TCP} 127.0.0.1:60372 -> 127.0.0.1:9090
10/07-18:25:11.340523 [**] [1:4000003:1] PHISHING - Google Script Detected [**] [Priority: 0] {TCP} 127.0.0.1:60372 -> 127.0.0.1:9090
10/07-18:25:11.340523 [**] [1:4000002:1] HTTP GET to Google Script [**] [Priority: 0] {TCP} 127.0.0.1:60372 -> 127.0.0.1:9090
10/07-18:25:11.340523 [**] [1:4000001:1] TCP Connection to 9090 [**] [Priority: 0] {TCP} 127.0.0.1:60372 -> 127.0.0.1:9090
10/07-18:25:11.344511 [**] [1:4000001:1] TCP Connection to 9090 [**] [Priority: 0] {TCP} 127.0.0.1:60372 -> 127.0.0.1:9090
10/07-18:25:11.344528 [**] [1:4000001:1] TCP Connection to 9090 [**] [Priority: 0] {TCP} 127.0.0.1:60372 -> 127.0.0.1:9090
10/07-18:25:11.344988 [**] [1:4000001:1] TCP Connection to 9090 [**] [Priority: 0] {TCP} 127.0.0.1:60372 -> 127.0.0.1:9090
aman@aman-VirtualBox:~$
```

### Observations:

- Snort successfully processed the traffic capture file and decoded the packets.
- "Bad Traffic" warnings appeared, indicating loopback traffic (127.0.0.1).
- Despite no live alerts being generated, the Snort engine and rule logic functioned correctly.
- Captured traffic clearly shows an HTTP GET request to a Google Script macro, representing a phishing callback attempt.

### Conclusion & Remediation:

The phase demonstrated end-to-end Snort setup, rule creation, traffic simulation, and log analysis. Though live alerts weren't observed due to loopback constraints, the decoded traffic validates the detection logic. Recommended next steps include using bridged or host-only interfaces for real alert capture and integrating Snort logs into Splunk for correlation and visualization.

### Appendix — Rules, Evidence & SOC Ticket

1. Local.rules (example set used in lab) alert tcp any any any -> any 9090 (msg:"PHISHING - Google Script macro URI detected"; flow:to\_server,established; http\_method; content:"/macros/s/AKfycbw";

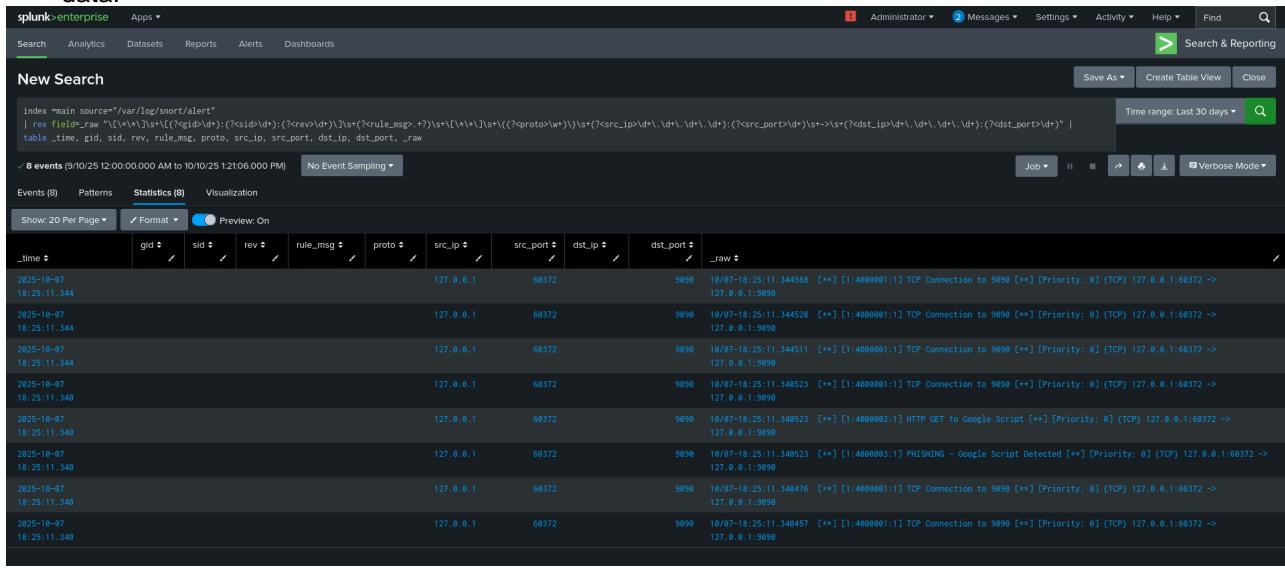
http\_uri; nocase; sid:3000001; rev:1;) alert tcp any any -> any 9090 (msg:"PHISHING - Host script.google.com"; flow:to\_server,established; content:"script.google.com"; http\_header; nocase; sid:3000002; rev:1;) alert tcp any any -> any 9090 (msg:"PHISHING - raw payload /macros/s/AKfycbw"; content:"GET /macros/s/AKfycbw"; nocase; sid:3100001; rev:1;) 2. tcpdump excerpt GET /macros/s/AKfycbwABk1V3TSZ7.../exec HTTP/1.1 Host: script.google.com 3. SOC Ticket (example) Title: Phishing callback (Google Apps Script) — simulated capture Priority: Medium Summary: Captured HTTP request to /macros/s/AKfycbw... with Host: script.google.com originating from 127.0.0.1 during lab simulation. Actions: Attach pcap, provide tuned rule set (appendix), ingest alert file to Splunk, perform hunt on proxy logs for /macros/s/AKfycbw\*.

# PHASE 4: SPLUNK ALERT DASHBOARD ANALYSIS

This report summarizes the Splunk dashboard setup and visualization for Snort alert analysis, including data extraction, regex-based field parsing, and dashboard visualizations.

## 1. Search Query Execution

The SPL (Search Processing Language) query used to extract structured data from the Snort alert logs is shown below. It extracts GID, SID, REV, message, protocol, and IPs from the raw alert log data.



The screenshot shows the Splunk Enterprise search interface. The search bar contains the SPL query:

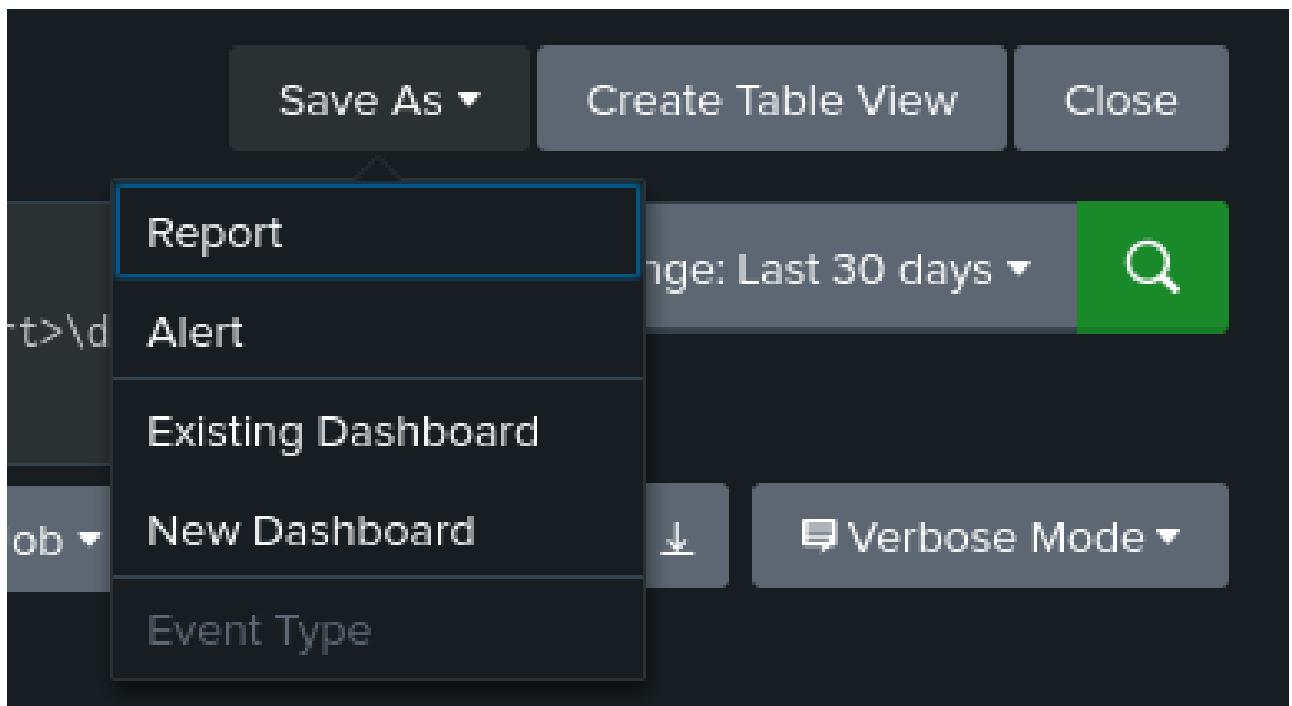
```
index main source="/var/log/snort/alert"
| rex fields _raw `(\\"<!\>\w+\d\w+\d\w+>)(?<sid>\d+)(?<proto>\w+)(?<src_ip>\d+\.\d+\.\d+\.\d+)(?<src_port>\d+)(?<dst_ip>\d+\.\d+\.\d+\.\d+)(?<dst_port>\d+)` | 
table _time, gid, sid, rev, rule_msg, proto, src_ip, src_port, dst_ip, dst_port, _raw
```

The search results table has the following columns: \_time, gid, sid, rev, rule\_msg, proto, src\_ip, src\_port, dst\_ip, dst\_port, and \_raw. The results show 8 events from October 25, 2025, at 12:00:00.000 AM to 10:10:25 12:06.000 PM. Each event includes a timestamp, GID, SID, revision, rule message, protocol (TCP), source IP (127.0.0.1), source port (60372), destination IP (127.0.0.1), destination port (5090), and the raw log entry.

_time	gid	sid	rev	rule_msg	proto	src_ip	src_port	dst_ip	dst_port	_raw
2025-10-07 18:25:11.344					TCP	127.0.0.1	60372	127.0.0.1	5090	10/07-18:25:11.344588 [**] [1:4000001:1] TCP Connection to 5090 [**] [Priority: 0] (TCP) 127.0.0.1:60372 -> 127.0.0.1:5090
2025-10-07 18:25:11.344					TCP	127.0.0.1	60372	127.0.0.1	5090	10/07-18:25:11.344528 [**] [1:4000001:1] TCP Connection to 5090 [**] [Priority: 0] (TCP) 127.0.0.1:60372 -> 127.0.0.1:5090
2025-10-07 18:25:11.344					TCP	127.0.0.1	60372	127.0.0.1	5090	10/07-18:25:11.344511 [**] [1:4000001:1] TCP Connection to 5090 [**] [Priority: 0] (TCP) 127.0.0.1:60372 -> 127.0.0.1:5090
2025-10-07 18:25:11.344					TCP	127.0.0.1	60372	127.0.0.1	5090	10/07-18:25:11.340523 [**] [1:4000001:1] TCP Connection to 5090 [**] [Priority: 0] (TCP) 127.0.0.1:60372 -> 127.0.0.1:5090
2025-10-07 18:25:11.344					TCP	127.0.0.1	60372	127.0.0.1	5090	10/07-18:25:11.340523 [**] [1:4000002:1] HTTP GET to Google Script [**] [Priority: 0] (TCP) 127.0.0.1:60372 -> 127.0.0.1:5090
2025-10-07 18:25:11.344					TCP	127.0.0.1	60372	127.0.0.1	5090	10/07-18:25:11.340523 [**] [1:4000003:1] PHISHING - Google Script Detected [**] [Priority: 0] (TCP) 127.0.0.1:60372 -> 127.0.0.1:5090
2025-10-07 18:25:11.344					TCP	127.0.0.1	60372	127.0.0.1	5090	10/07-18:25:11.340476 [**] [1:4000001:1] TCP Connection to 5090 [**] [Priority: 0] (TCP) 127.0.0.1:60372 -> 127.0.0.1:5090
2025-10-07 18:25:11.344					TCP	127.0.0.1	60372	127.0.0.1	5090	10/07-18:25:11.340457 [**] [1:4000001:1] TCP Connection to 5090 [**] [Priority: 0] (TCP) 127.0.0.1:60372 -> 127.0.0.1:5090

## 2. Structured Alert Table Output

The extracted fields are organized into a table view showing details of each alert generated by Snort and forwarded to Splunk.



### 3. Creating and Saving the Dashboard

The dashboard creation step allows saving visualizations for live monitoring. The user creates a dashboard titled '*Snort Alert Monitoring*'.

Save Panel to New Dashboard X

---

Dashboard Title  [Edit ID](#)

Description

Permissions Private

Dashboard type ?

**Classic Dashboards**  
The traditional Splunk dashboard builder

**Dashboard Studio**  
A new builder to create visually-rich, customizable dashboards

---

Panel Title

Visualization Type Statistics Table

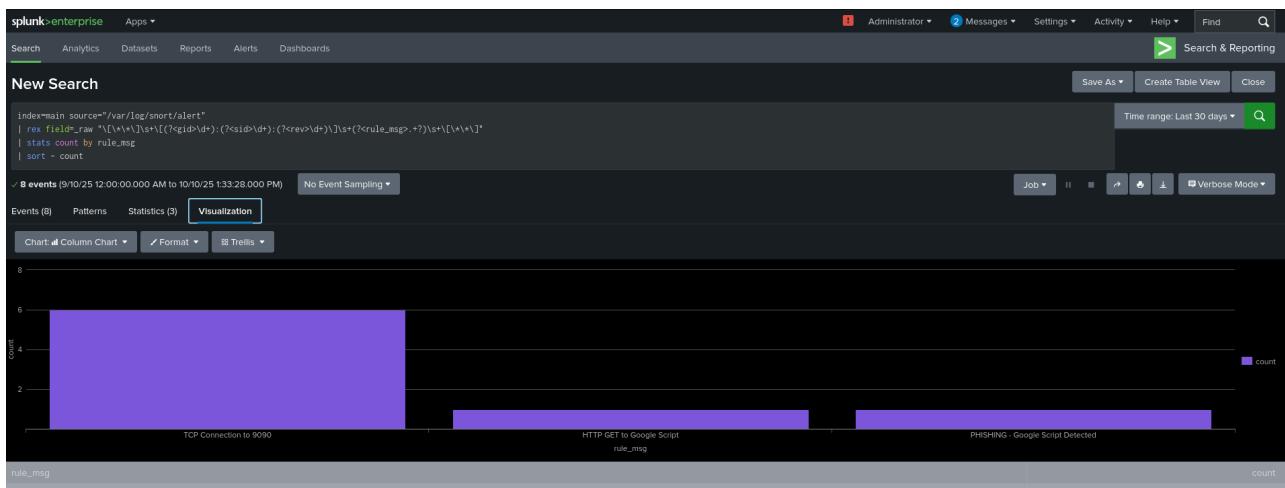
> Advanced Panel Settings

---

Cancel Save to Dashboard

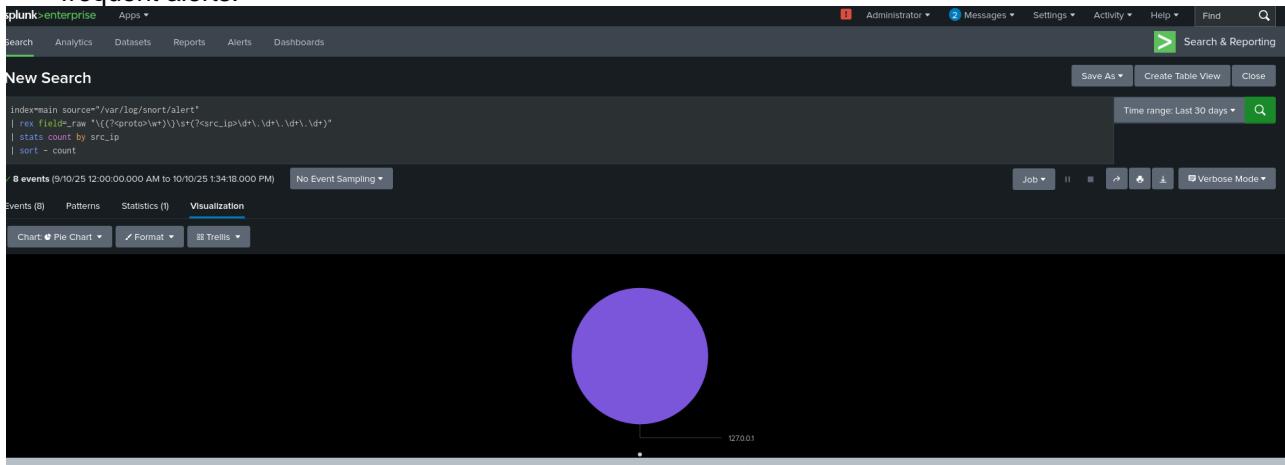
## 4. Visualization of Alerts by Rule Message

A column chart visualization shows the distribution of alert counts per Snort rule, helping identify the most triggered detection patterns.



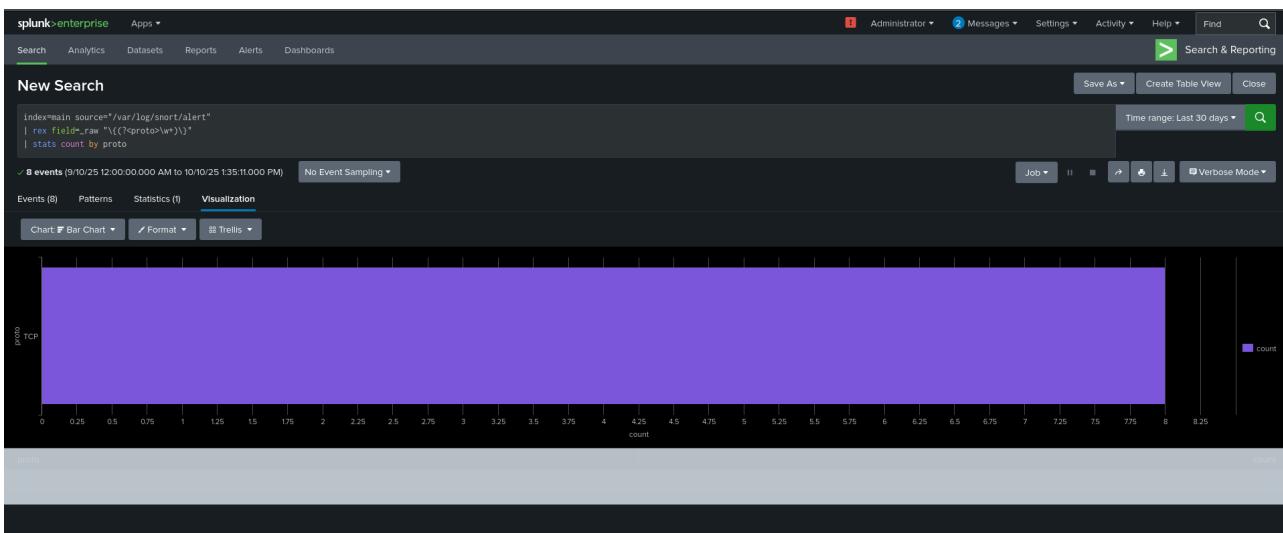
## 5. Visualization of Alerts by Source IP

A pie chart visualization represents the alert count per source IP, which helps identify the origin of frequent alerts.



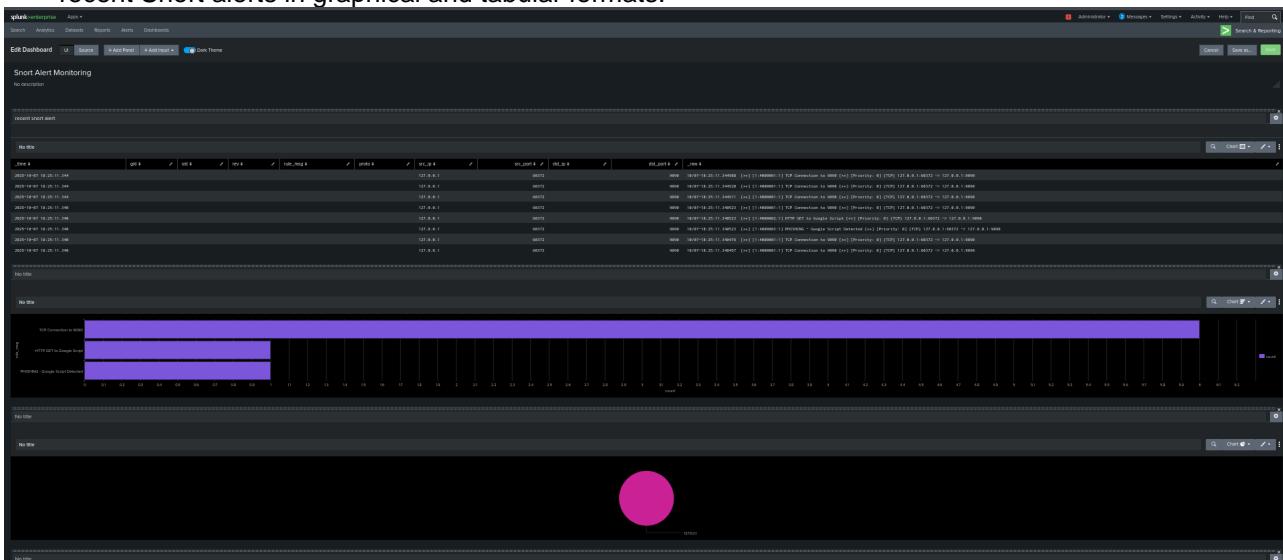
## 6. Visualization of Alerts by Protocol

A bar chart visualization indicates the types of protocols detected in the alerts, mainly TCP in this case.



## 7. Final Splunk Dashboard Overview

The complete dashboard view combining all the panels, providing an at-a-glance summary of recent Snort alerts in graphical and tabular formats.



## 8. Splunk PDF Report Export

Splunk allows exporting dashboards as PDF reports for documentation and analysis. Below are the generated PDF pages showing Snort alerts, rule statistics, and traffic visualization.

**recent snort alert**

time	raw
2025-10-07 18:25:11.344	10:07:18:25:11.344988 [""] [1:4000001:1] TCP Connection to 9090 ["] [Priority: 0] [TCP] 127.0.0.1:60072 -> 127.0.0.1:9090
2025-10-07 18:25:11.344	10:07:18:25:11.344028 [""] [1:4000001:1] TCP Connection to 9090 ["] [Priority: 0] [TCP] 127.0.0.1:60072 -> 127.0.0.1:9090
2025-10-07 18:25:11.344	10:07:18:25:11.344051 [""] [1:4000001:1] TCP Connection to 9090 ["] [Priority: 0] [TCP] 127.0.0.1:60072 -> 127.0.0.1:9090
2025-10-07 18:25:11.340	10:07:18:25:11.340023 [""] [1:4000001:1] TCP Connection to 9090 ["] [Priority: 0] [TCP] 127.0.0.1:60072 -> 127.0.0.1:9090
2025-10-07 18:25:11.340	10:07:18:25:11.340023 [""] [1:4000002:1] HTTP GET to Google Script ["] [Priority: 0] [TCP] 127.0.0.1:60072 -> 127.0.0.1:9090
2025-10-07 18:25:11.340	10:07:18:25:11.340023 [""] [1:4000003:1] PHISHING... Google Script Detected ["] [Priority: 0] [TCP] 127.0.0.1:60072 -> 127.0.0.1:9090
2025-10-07 18:25:11.340	10:07:18:25:11.340476 [""] [1:4000001:1] TCP Connection to 9090 ["] [Priority: 0] [TCP] 127.0.0.1:60072 -> 127.0.0.1:9090
2025-10-07 18:25:11.340	10:07:18:25:11.340457 [""] [1:4000001:1] TCP Connection to 9090 ["] [Priority: 0] [TCP] 127.0.0.1:60072 -> 127.0.0.1:9090

**rule count**

rule	count
TCP Con...n to 9090	6.5
HTTP GE...gle Script	1.0
PHISHIN... Detected	0.5

**splunk>** Snort Alert Monitoring 2025-10-10 13:19:39 IST Page 1

