CHAPTER 7

Network Security Monitoring

Chapter Overview

01

Security Onion

NSM platform — architecture and integrated toolset

02

Zeek (formerly Bro)

Protocol analysis and structured network logging

03

Snort, Sguil & Squert

Signature IDS and analyst workflow

04

Elastic Stack

Log aggregation, search, and visualization

05

Text-Based Log Analysis

Command-line tools for rapid log parsing

Security Onion

The NSM platform for defenders

Security Onion Architecture

Security Onion: free, open-source Linux distro for NSM and log management

Three node types: Manager, Search, Sensor

Sensors: tap or span port, capture raw traffic and generate logs

Manager: centralized configuration, alert management, user access

Search nodes: scalable Elasticsearch for large environments

Integrates: Zeek, Suricata, Elastic Stack, Kibana, Fleet, CyberChef

Suitable for SOC, hunt teams, and incident response teams

Security Onion Deployment Modes

Standalone: all components on one system — lab/small deployment

Distributed: dedicated sensors, search nodes, and manager — enterprise deployment

Import: offline analysis of PCAP files — no live traffic required

Eval mode: simplified setup for evaluation — not for production

Sensor deployment: span/mirror port from managed switch or inline tap

TAP (Test Access Point): passive optical or copper tap — no traffic impact

Sizing: 1 Gbps sustained traffic requires ~8 CPU cores and fast SSD storage

Tuning: whitelist internal scanners, backup tools to reduce noise

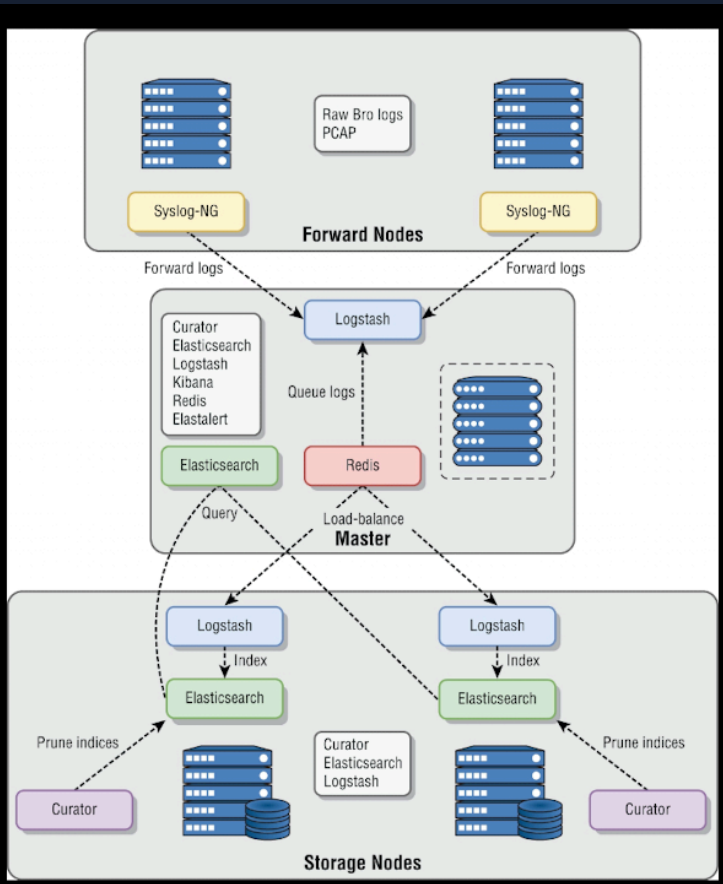


Figure 7.1: The recommended Security Onion architecture

Snort, Sguil & Squert

Signature detection and alert triage

Snort / Suricata + Alert Management

Snort / Suricata: signature-based IDS — match packets against rule sets

- **Rule sets**: Emerging Threats, Snort Community, custom organizational rules

Sguil: analyst console — correlates IDS alerts with full packet capture

- **Click-to-pivot**: Sguil alert → full packet reconstruction → Wireshark

Squert: web-based interface for Sguil database — summary views, search

Alert fatigue is real: tune rules to reduce false positives

Combine signature detection (Snort) with behavioral (Zeek) for best coverage

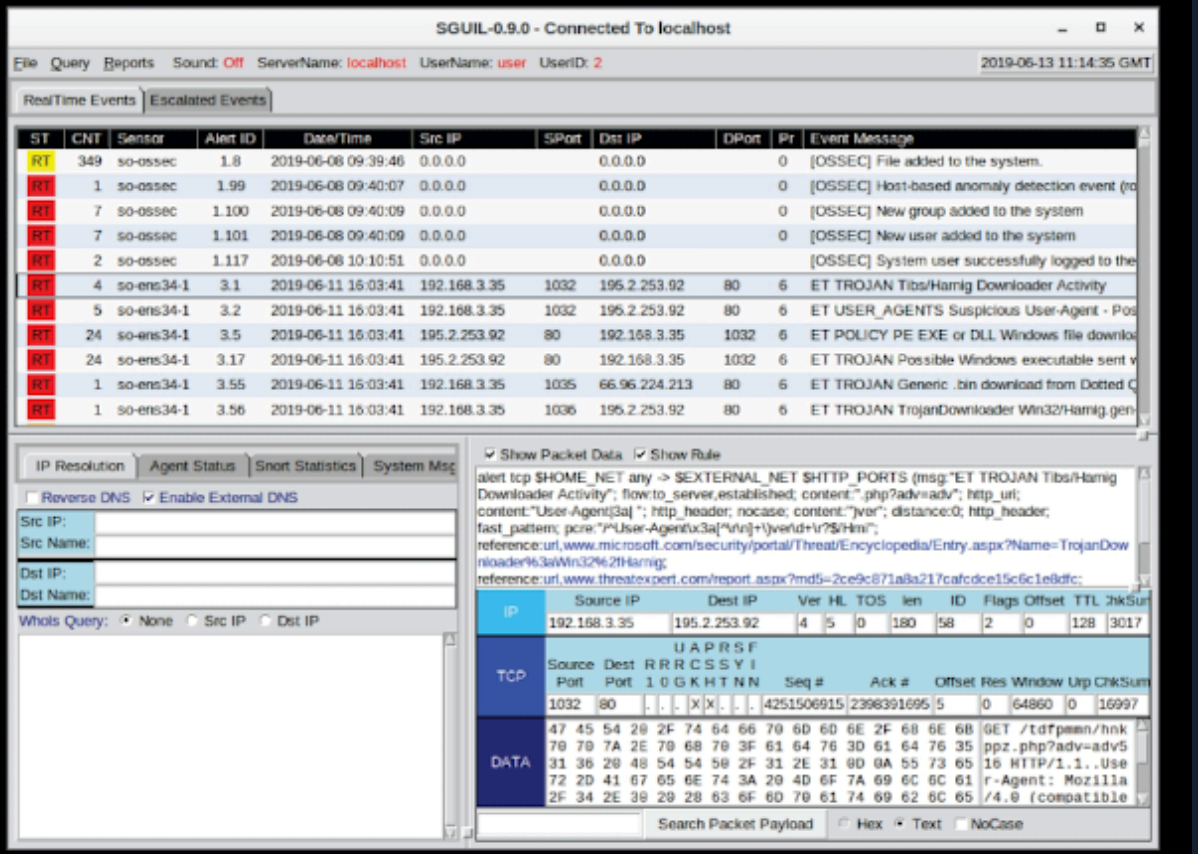
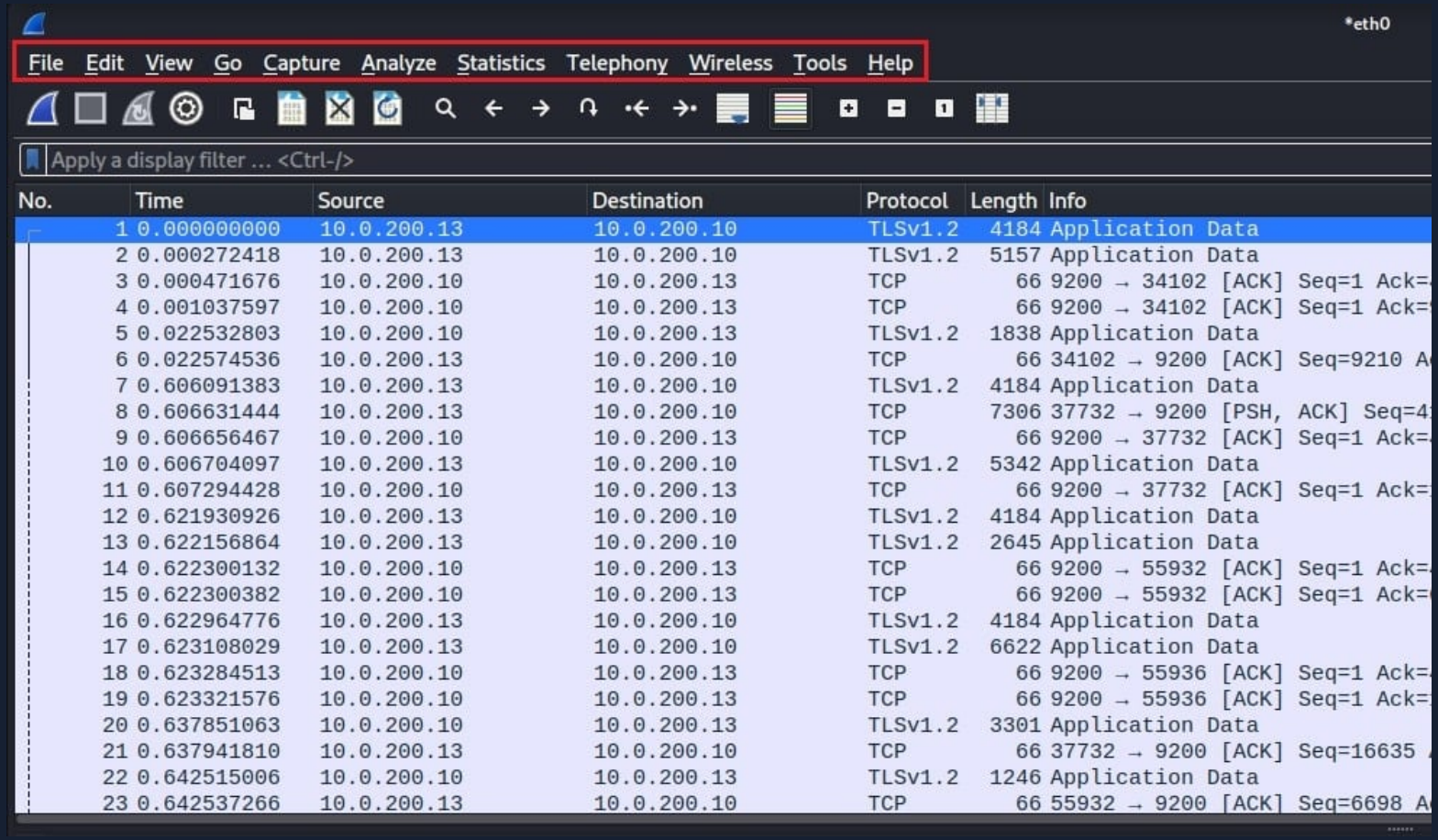


Figure 7.2: The RealTime Events queue of the Sguil interface



Wireshark network capture interface — inspecting raw packets during incident investigation

Snort / Suricata Rule Syntax

Rule format: action protocol src src_port -> dst dst_port (options)

Example: alert tcp any any -> $HOME_NET 4444 (msg:'Metasploit default port'; sid:9000001; rev:1;)

Action types: alert (log + alert), drop (IPS mode, block), pass (ignore)

Content matching: content:'cmd.exe'; nocase; — search for strings in payload

PCRE (Perl Compatible Regular Expression):

pcre:'/\/bin\/sh/i'; — regex for shell commands in payload

Flow: flow:established,to_server; — match only established connections

Threshold: threshold:type limit, track by_src, count 5, seconds 60; — rate limiting

Emerging Threats: free community rules updated daily at rules.emergingthreats.net

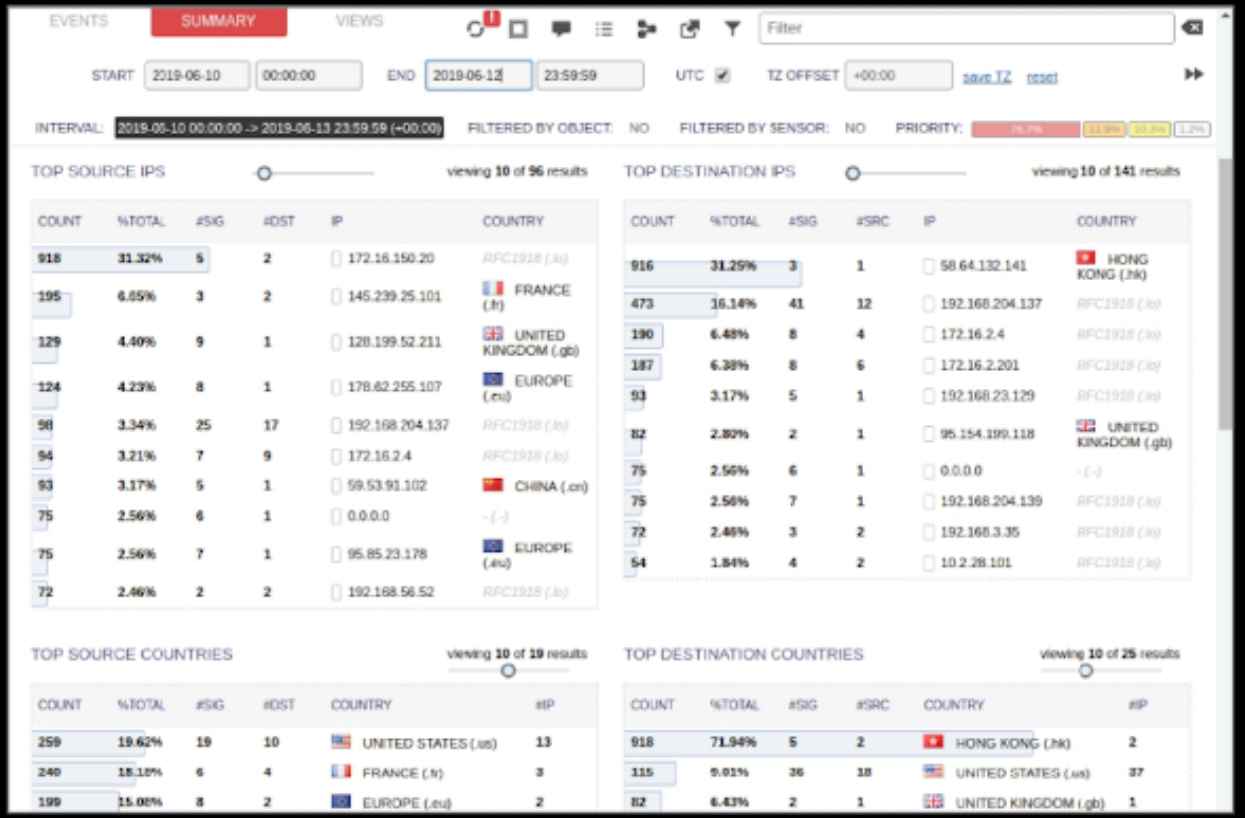


Figure 7.7: The Squert Summary tab dashboards

Zeek (formerly Bro)

Structured protocol-aware network logging

Zeek Log Types

Core Log Files

conn.log: every TCP/UDP/ICMP connection (src, dst, duration, bytes)

dns.log: all DNS queries and responses

http.log: HTTP requests — method, URI, user-agent, response code

ssl.log: TLS/SSL sessions — cert details, SNI, cipher suite

files.log: all file transfers with MIME type and hash

Threat Hunting with Zeek

Long-duration connections: beacons to C2

High-volume DNS: DNS tunneling or DGA

Unusual user agents: malware C2 callbacks

Self-signed certs on **non-standard ports**

zeek-cut: command-line tool to extract specific fields from logs

Zeek Log Field Reference — Key Fields

conn.log key fields: ts, id.orig_h, id.orig_p, id.resp_h, id.resp_p, proto, duration, orig_bytes, resp_bytes, conn_state

conn_state values: S1=established, SF=normal close, REJ=rejected, RSTO=aborted, S0=attempt no reply

dns.log: ts, query, qtype_name, rcode_name, answers — hunt for NXDomain storms (DGA activity)

http.log: method, host, uri, user_agent, status_code — spot malware beaconing by user-agent

ssl.log: server_name (SNI), subject, issuer, cipher, validation_status — find self-signed certs

files.log: fuid, mime_type, filename, md5, sha1 — hash every file transferred over HTTP/SMB

weird.log: protocol anomalies and parser errors — often reveals malformed C2 traffic

zeek-cut usage: zeek-cut id.orig_h id.resp_p conn_state < conn.log — extract specific fields

Threat Hunting with Zeek Logs

C2 beaconing: group conn.log by dest IP, compute interval standard deviation — low σ = beacon

DNS tunneling: cat dns.log | zeek-cut query | awk '{print length($1)}' | sort -n | tail -20

High-query domains: cat dns.log | zeek-cut query | sort | uniq -c | sort -rn | head -20

User-agent analysis: cat http.log | zeek-cut user_agent | sort | uniq -c | sort -rn

Large outbound: cat conn.log | zeek-cut id.resp_h orig_bytes | sort -k2 -rn | head -20

Self-signed TLS: cat ssl.log | zeek-cut server_name issuer | grep 'CN=.*CN=' | head -20

New external IPs: baseline known-good IPs, then filter: grep -vFf known_ips.txt

East-west SMB: connections on port 445 between internal workstations — never normal

Elastic Stack & Text Analysis

Search and visualize at scale

Elastic Stack & Command-Line Analysis

Elastic Stack (ELK)

A suite of tools to aggregate, index, search, and display large amounts of information

Elasticsearch: distributed search and storage

Logstash: ingest, parse, and enrich logs

Kibana: dashboards, search, and visualizations

Beats: lightweight shippers (Filebeat, Packetbeat)

Enables hunting across millions of events in seconds

Text-Based Log Analysis

grep -E 'pattern' zeek/conn.log: quick searches

awk -F'\t' '{print $5}' conn.log | sort | uniq -c | sort -rn

zeek-cut id.orig_h id.resp_h id.resp_p conn_state < conn.log

cat http.log | zeek-cut -d ts host uri | grep -v '200'

Command-line fu: fast, no SIEM required, works on any log

Kibana Hunting Queries

KQL syntax: agent.type:zeek AND network.transport:tcp AND destination.port:4444

High-volume DNS: filter on event.dataset:zeek.dns, aggregate by dns.question.name

Unusual user agents: http.request.user_agent.original: * AND NOT (*Mozilla* OR *Chrome*)

Long-lived connections: network.duration > 3600 AND destination.geo.country_iso_code != 'US'

Internal scanning: source.ip: 10.0.0.0/8 AND destination.port: 445 AND event.type: connection

Dashboards: pre-built Zeek, Suricata, and host dashboards in Security Onion

Timeline pivot: from alert to full packet capture with one click in Sguil/Security Onion

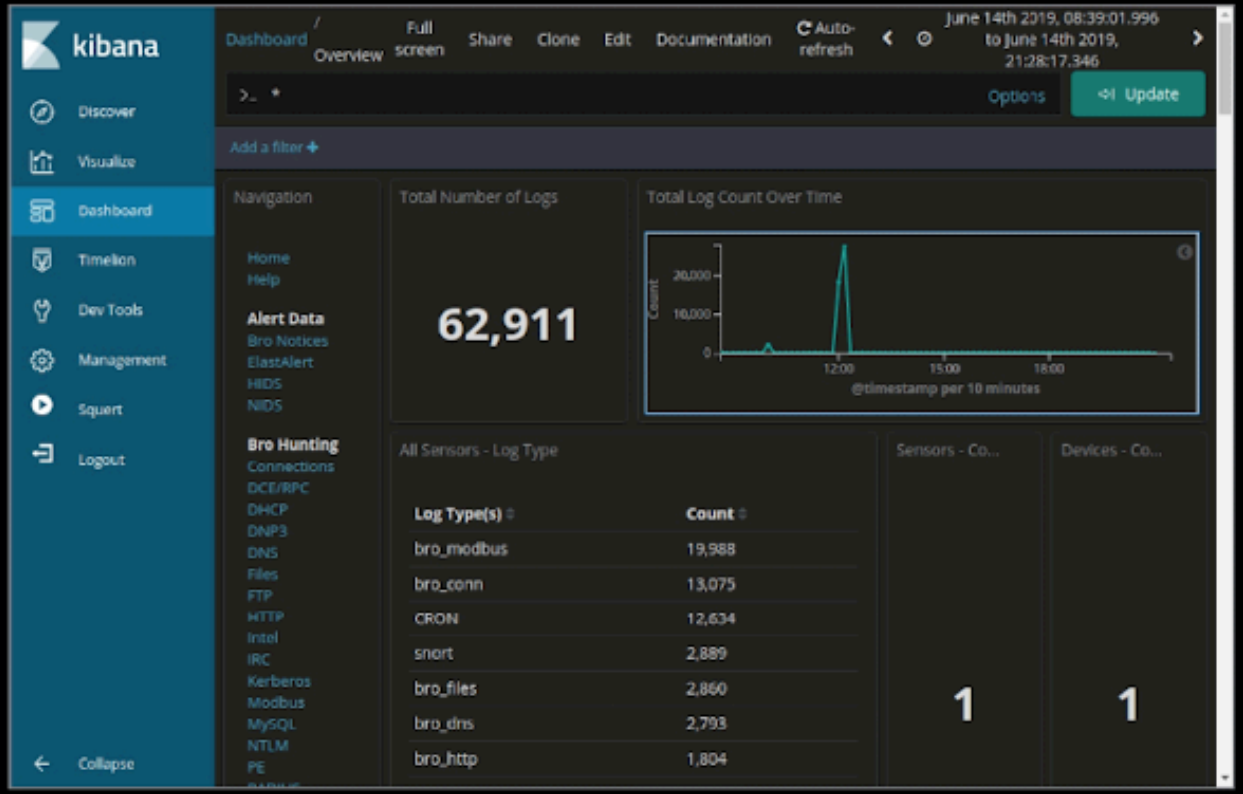


Figure 7.8: The Kibana web interface

Network Baselining & Full Packet Capture

Baseline: document normal traffic patterns before hunting for abnormal

Top talkers: which internal hosts generate most traffic? Any unusual spikes?

Time-of-day patterns: spikes at 3 AM when users are offline = automated activity

New connections: hosts or IPs not seen in the past 30 days — immediate hunt focus

pcap: raw binary capture of every packet — the ground truth of network forensics

Wireshark display filter: http.request.method=='POST' && ip.dst==1.2.3.4

Follow TCP Stream: reconstruct attacker's full session from individual packets

Export HTTP objects: File → Export Objects → HTTP — recover transferred files

Network Forensics Tools

NetworkMiner: parses pcap and automatically extracts files, images, credentials, and host data

Wireshark: full pcap analysis — display filters, stream reassembly, protocol dissectors

tcpdump: command-line capture and display — tcpdump -i eth0 -w capture.pcap

ngrep: network grep — ngrep 'password' -q -W byline port 80 — search packet payloads

Moloch (Arkime): full packet capture + indexed search — scales to enterprise traffic volumes

RITA (Real Intelligence Threat Analytics): automated analysis of Zeek logs for C2 beaconing

CapTipper: Python tool for HTTP traffic analysis from pcap — extracts and replays flows

The right tool depends on scale: Wireshark for single files; Arkime/Elastic for fleet-level pcap

Conclusion

Network visibility is essential — you cannot detect what you cannot see

Zeek provides rich, structured logs that complement signature-based IDS

Elastic Stack scales log search to enterprise traffic volumes

Combine IDS alerts with full packet capture for complete incident context

Text-based analysis is a fast, lightweight alternative when tooling isn't available

Baselines are the foundation — without normal, you cannot spot abnormal

Knowledge Check