Central Logging

Introduction

- Central Syslog, a Challenge
- Example central syslog implementations
- Typical Problems
- Share Experiences / Knowledge Base

Central Syslog, a Challenge

One of the first things an Attacker does is to stop logging/clean these traces

- Modifications of logging (and some other activities) trigger alarms.
- Make sure your logging data can be trusted / lock it down.
- Make sure you have a single point where to analyse your data.
- This is independent of the number of hosts you care about.
- Having logs modified by an attacker are painful.

Central Syslog, a Challenge

Overview

- Central logs are a powerful tool for Intrusion Detection and Incident Response
- Problem: Variety of products
- Even more ways to implement it
- How to do it right?

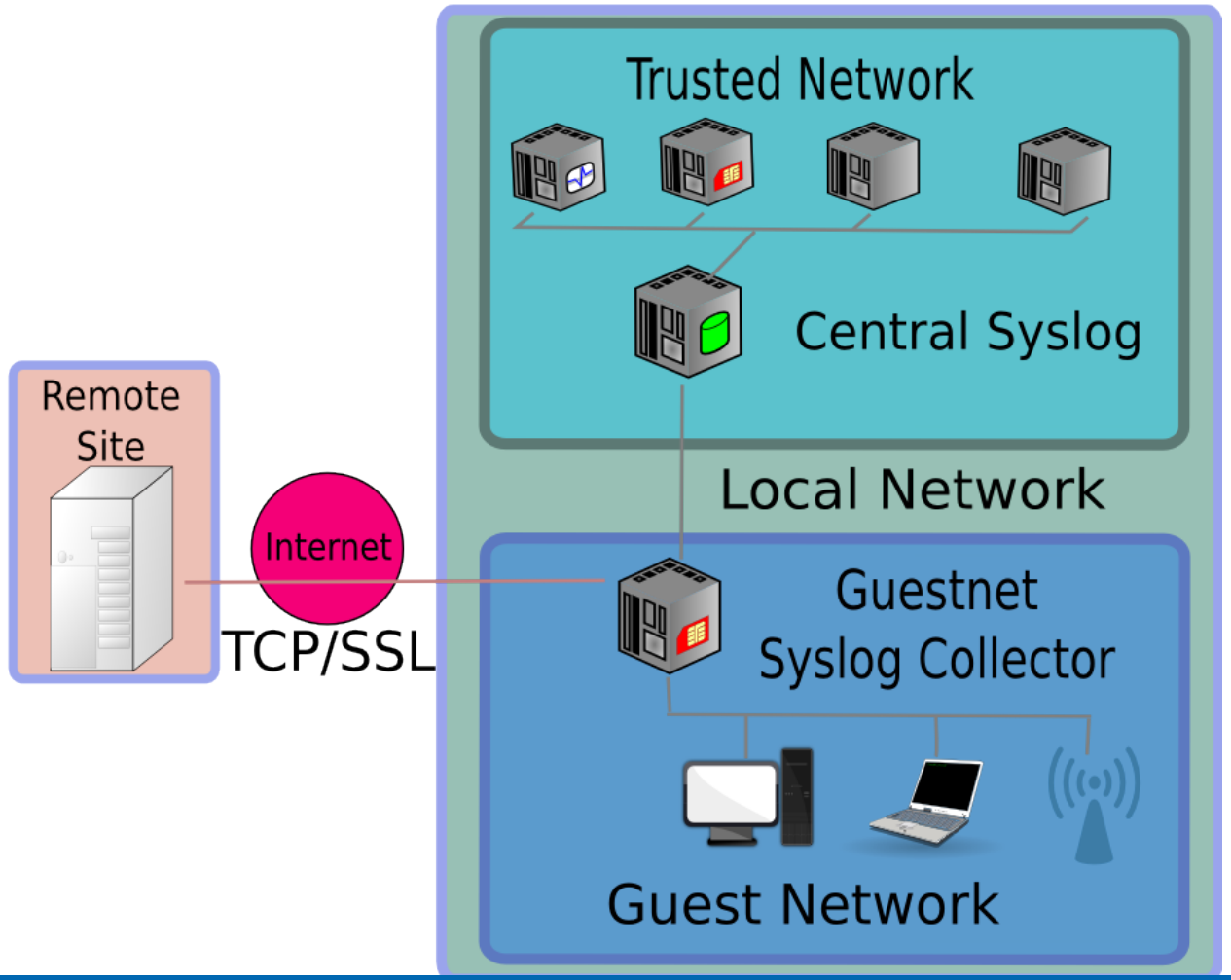
Example central syslog implementations

How is your site organized?

- What are your network segments?
- Are there log sources external from your trusted network?
- Place the central logger in the innermost segment, secure/limit access to it
- If you want to include log sources from not trusted networks, have a tiered approach
- Make sure logging information is transported securely.

Example central syslog implementations

Example Network



Example central syslog implementations

Implementation Details

- Hosts from the trusted network send syslog via UDP (loss of an ”acceptable” amount of information?)
- External hosts: TCP/TLS
- Internal hosts; complete syslog info important: TCP, local cache

CERN central logs

- Logs collected:
  - Standard syslog
  - *Netlog*: all high level network activity
  - *snoopy*: all execution on hosts
- What for?
  - Live analysis:
    - Known threats detection
    - SSH logins (per account)
  - Forensics/analysis

Old Infrastructure

- One central rsyslog server:
  - Storing received data
  - Backups on tape
  - Forwarding to analysis
- Analysis:
  - Rsyslog server with special configuration

Issues

- Loosing logs?
- Changing server IP and/or name (cached)
- Scalability:
  - Number of hosts
  - Size ($\sim$ 400GB/day, up to 700GB/day)
- TCP: blocking queues

Issues and mitigations

- Loosing logs?
  $\rightarrow$ Monitoring via crafted *events*
- Changing server IP and/or name (cached)
  $\rightarrow$ On puppet hosts: rsyslog restarted
- Scalability:
  - Number of hosts
  - Size ($\sim$ 400GB/day, up to 700GB/day)

  $\rightarrow$ 3 central servers + 2 analyzers
  $\rightarrow$ Number of files open set to max ($\sim$ 1000)
- TCP: blocking queues
  $\rightarrow$ Using UDP

New Infrastructure

Flume as transport, pushing to:

- Spark: Live analysis
- HDFS: used via Shark for forensics
- Elasticsearch/Kibana: for administrators & debugging

Central Logging/Problems

Collecting/Storing Syslog data

- Neither just send everything via udp nor just use tcp will work
- Non Syslog apps?
- Common log format (logstash)
- Plain files ? Databases
- Databases
- logrotate!

Central Logging/Problems

Making sense of Syslog data

- Common data structure facilitates re-usable analysis tools, like:
- IDS, log in from ”unusual” regions/IPs (different site-specific!) approaches available
- IR, Connections to suspicious IPs
- Activities connected to problematic DN (logs from CE/WMS/SE etc)

Central Logging/Problems

Sharing expertise / What would be nice ...

- Common basic configuration to collect/store/rotate logfiles
- Common log file format, generic filters
- Re-Usable IDS tools (blacklist IPs, ssh intrusions)
- Re-Usable IR tools (activity of a DN, Connections to/from a IP)
- Ready to go Secure logging infrastructure for clouds (the credit card owners will need it)
- syslog functionality Monitoring plugins (nagios etc)
- How to leverage collaboration on that? Format?