www.crysys.hu

Logging and network forensics

Tamás Holczer
CrySyS Lab
Budapest University of Technology and Economics
holczer@crysys.hu

Outline

- Logging (partly based on the presentation of Péter Höltzl)
- Network forensics (partly based on the presentations of Levente Buttyán and András Gazdag)
- ELK stack partly based on the presenattion of Gergő Ládi

What is a log

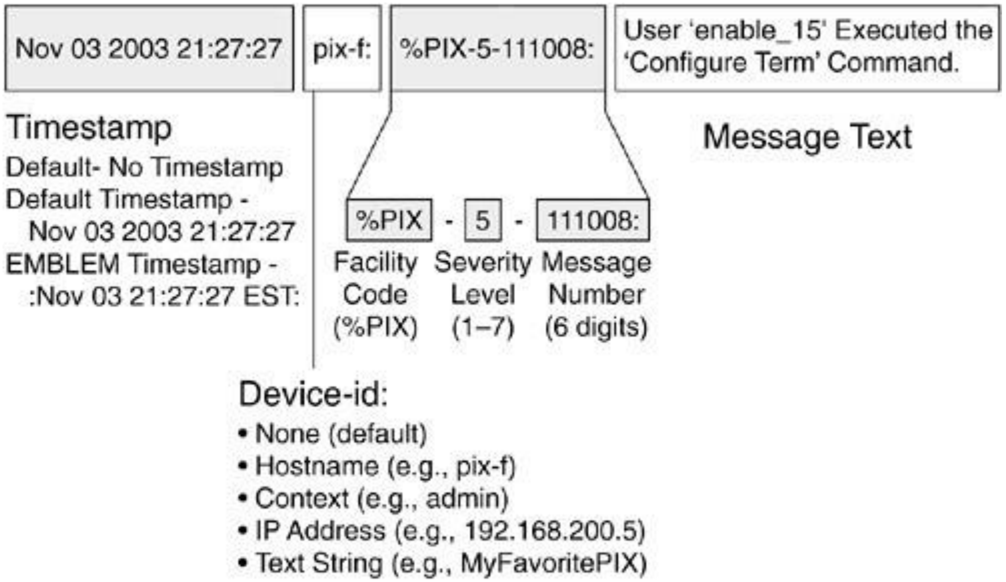
- LOG = record of events (entries)
- Goal of logging:
  - Debugging
  - Performance optimization
  - Authorized and unauthorized activity recording
  - Record of compliance
  - Policy

Who creates logs

- Applications
  - Web server
  - Email server
  - VPN
  - DHCP Server
  - AV
  - …
- Network devices
  - Firewall
  - Switch
  - …
- OS
- IDS, IPS
- …

BSD syslog (RFC3164)

- Developed in 80s
- Orig: part of sendmail
- RFC3164 2001
  - Documents the status
- RFC5424 2009
  - Standardizes syslog
  - Obsolotes 3164



- Device ID: usually hostname (no FQDN)
- Facility: kern, user mail, daemon, auth…
- Severity: 0-Emergency, 7-Debug
- MSG: Latin-1 free text

Problems with Syslog

- UDP 514
- No unique identifier for events
- No acknowledgement
- No security (integrity protection or encryption)
- Timestamp: no year or timezone in many cases
- No multiline messages
- No L7 acknowledgement

- Best effort service no reliability

Syslog API and syslogd

- Applications normally uses the Syslog API
- Syslog events goes to /dev/log
- syslogd collects records from /dev/log and stores them (default: /var/log/syslog) according to a configuration

```
#include <syslog.h>
syslog (LOG_MAKEPRI(LOG_LOCAL1, LOG_ERROR), "Unable to make network connection to %s.  Error=%m", host);
```

```
import syslog
syslog.syslog(syslog.LOG_ERR, ”Some error happened”)
```

Reliable Delivery for syslog (RFC 3195 2001)

- Based on original RFC3164
- Uses TCP (acknowledgement)
- Cryptographic protectin
  - Encryption: TLS_RSA_WITH_3DES_EDE_CBC_SHA
  - Authentication: based on MD5
- Raw profile: single line
- Cooked profile: multiline, xml

- Error codes from HTTP:
  - 200 Success
  - 500 General syntax error
  - …

New standard for syslog

- RFC5424 2009
- Obsolotes old RFC3164
- RFC5425 - RFC5424 over TLS
- RFC5426 - RFC5424 over UDP
- RFC5427 - PRI definitions
- RFC5848 – digitally signed RFC5424 (SDATA field)

- Well defined timestamp format
- Multiline
- TCP and TLS
- UTF-8

```
<165>1 2003-08-24T05:14:15.000003-07:00 192.0.2.1 myproc 8710
 - - %% It's time to make the do-nuts.
```

Problems with 5424

- No L7 acknowledgement
- No authentication (only implicit with optional TLS)
- Only optional integrity protection

- Not widely implemented (but: syslog-ng)

Other logging solutions

- Microsoft eventlog
  - EVT API -> file (%SystemRoot%\System32\winevt\Logs*.evtx)
  - Event Viewer
  - Local log facility
  - Remote log: RPC
- SQL (INSERT…)
- Text files (e.g Python import logging)
- CLF (Common Log Format) standard text log format for web server
  - 127.0.0.1 user-identifier frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326
- SNMP (Simple Network Management Protocol)
  - GET/SET Request
  - Trap
  - SNMP v1-2-2c-3 (<3: cleartext community strings, 3: confidentiality, integrity, auth)
- SDEE (Security Device Event Exchange)
  - Mainly for security events
  - Standard of International Computer Security Association
  - Mainly used by Cisco

Structured logging

- JSON (JavaScript Object Notation):
  - { "sender" : "michael" "recipient": { "name" : "michael", "name" : "andrea", "name" : "itay" } subject:"I <3 logs" }
- WELF (WebTrends Enhanced Log file Format):
  - pri=123 date=2015-08-17T10:10:10.000+01:00 host=test program=pf pid=123 IN=eth0 OUT= MAC=00:4a:54:c2:f7:e5:00:08:e5:ff:fd:90:08:00 SRC=1.2.3.4 DST=5.6.7.8 LEN=40 TOS=0x00 PREC=0x00 TTL=49 ID=0 DF PROTO=TCP SPT=51777 DPT=80 WINDOW=0 RES=0x00 RST URGP=0
- XML

Common problems

- Different formats
- Not normalized (e.g. timestamp)
- String instead of structured text
- Volume problems
  - High EPS (event per second)
  - Lof of concurrent connections
  - 1 event creates lot of messages

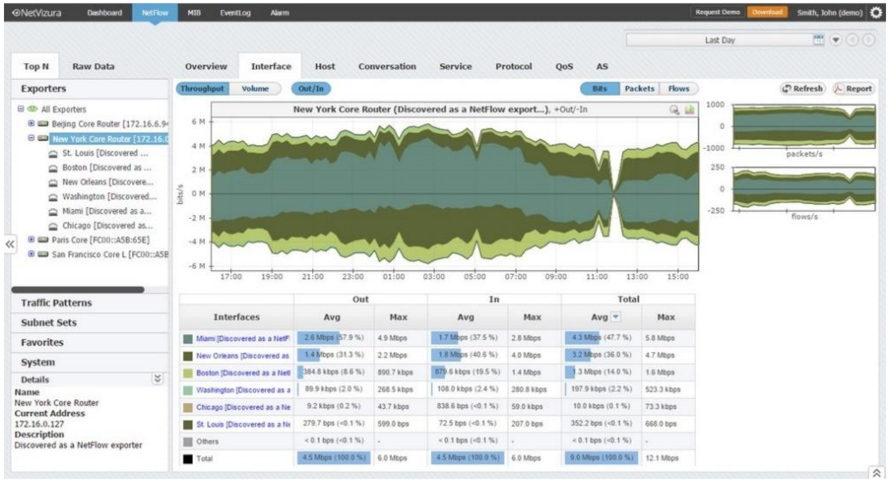
Storage questions

- Local storage
  - No traffic
  - Hard to use
- Central storage
  - Network usage
  - „Safe” place (attacker cannot erase after compromise)
- Mixed storage
  - Locally interesting
  - Locally interesting but without storage (router, switch)
  - Globally interesting

- Encrypted storage?
- Digitally signed storage?

Packet capture

- Full packet capture (mainly for forensics, later in this lecture)
- Flow collection
  - Who communicates with whom at when
  - No payload collected
  - NetFlow / IPFIX (NetFlow v10)
  - Source IP, Destination IP, Source port, Destination port, Time, Header fileds…
  - Sender: router, switch, firewall, server…
  - Destination: flow collector
  - Analyser: dashboard, report, alert

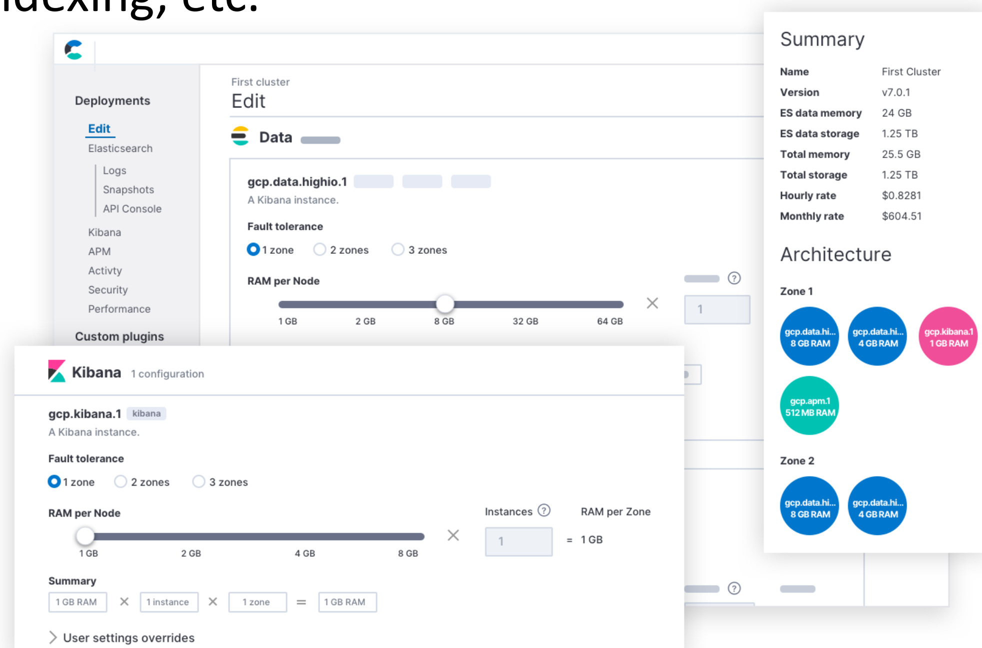


The ELK Stack

- ELK = elasticsearch, logstash, kibana
- One of the most popular log management and analytics solutions
- All open source software

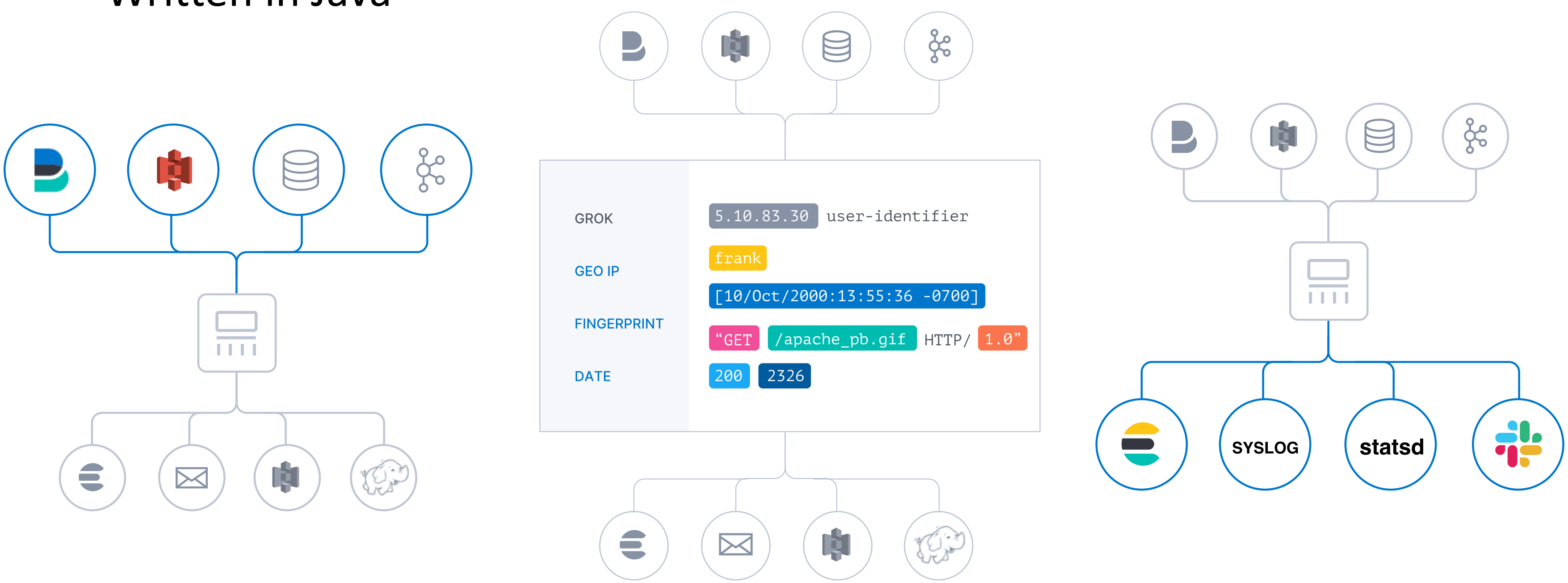
The ELK Stack

- Elasticsearch
  - A search and analytics engine, based on Apache Lucene
  - A NoSQL database
  - Has a REST API
  - Sharding and replica support
  - Plugins for analysis, alerting, indexing, etc.
  - Written in Java



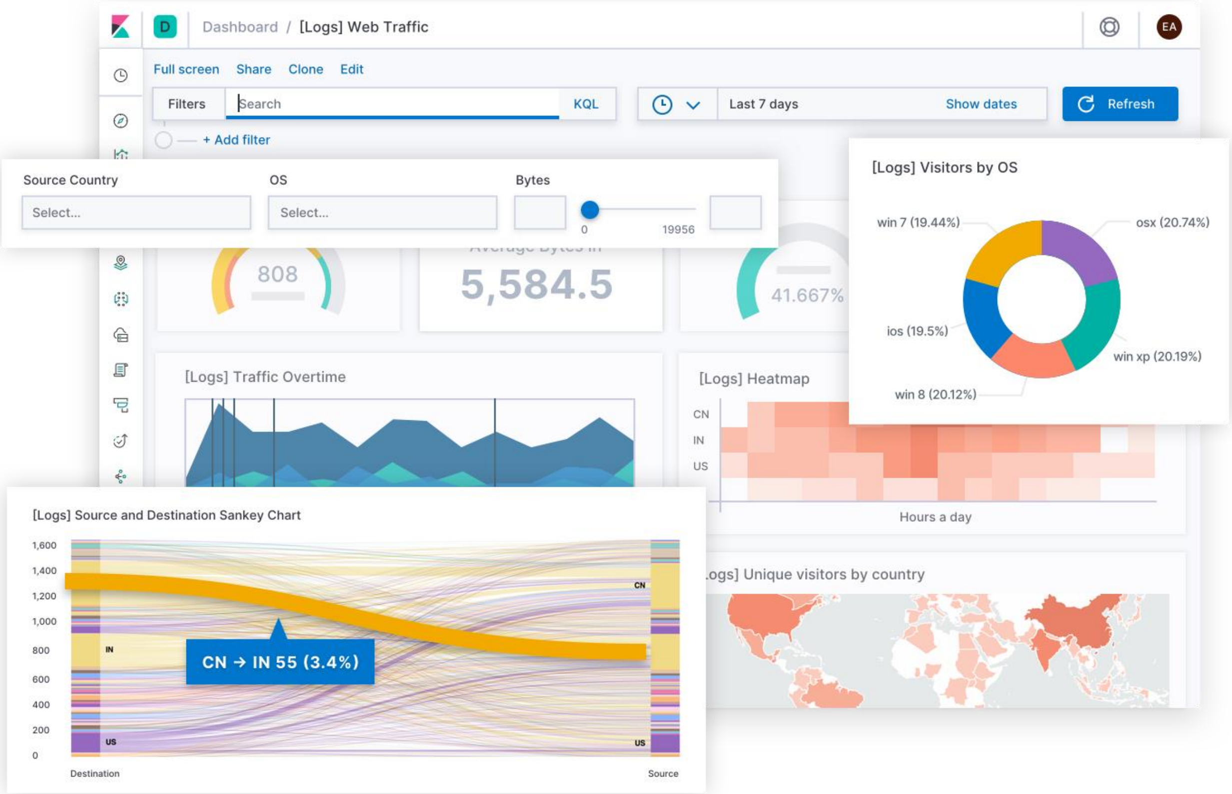
The ELK Stack

- Logstash (Alternative: Fluentd/td-agent)
  - Collects and processes logs from different sources
  - Supports more than 50 different source formats
  - Supports several output formats
  - In our case, it feeds data to Elasticsearch
  - Written in Java



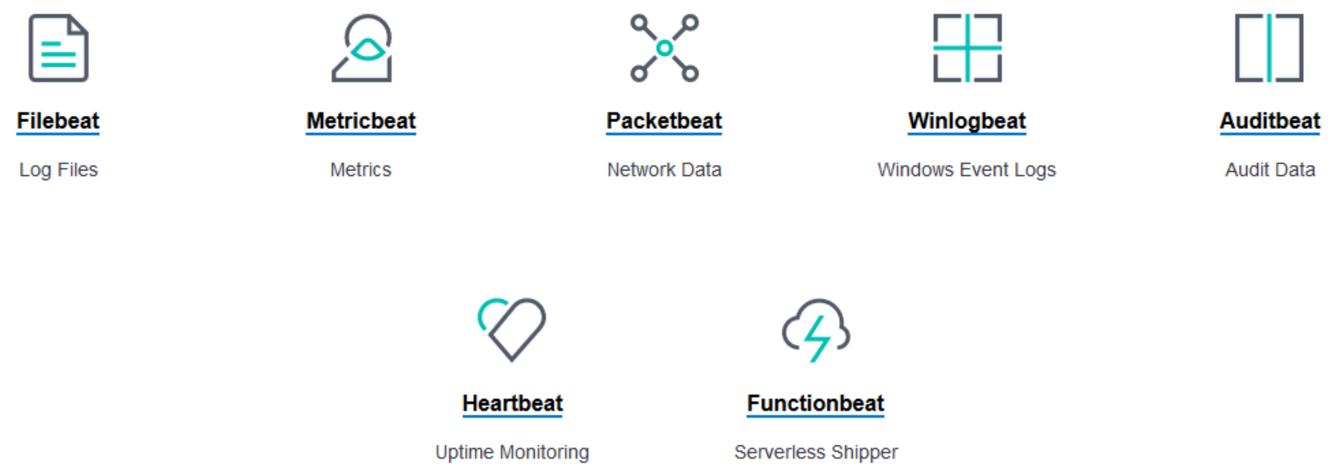
The ELK Stack

- Kibana
  - A web interface to query data in Elasticsearch
  - Data visualization, dashboards
  - Written in node.js

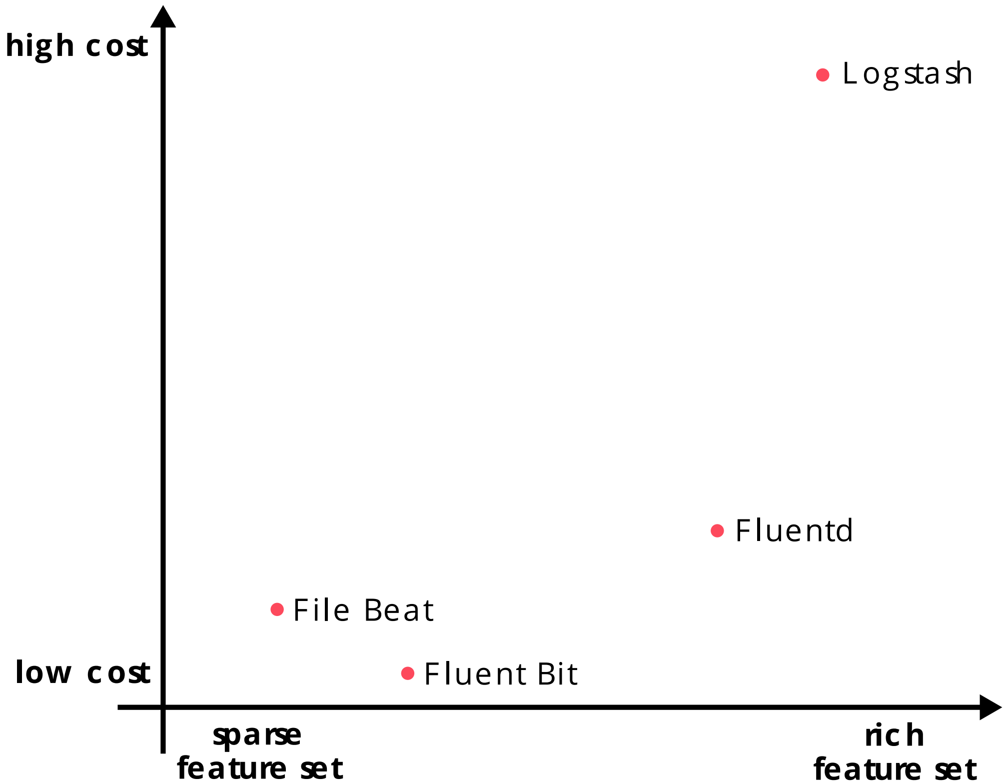


The ELK Stack

- Beats (Alternative: Fluentbit)
  - Originally not part of the ELK stack
  - Collects and feeds extra information (not necessarily logs)
  - Written in Go



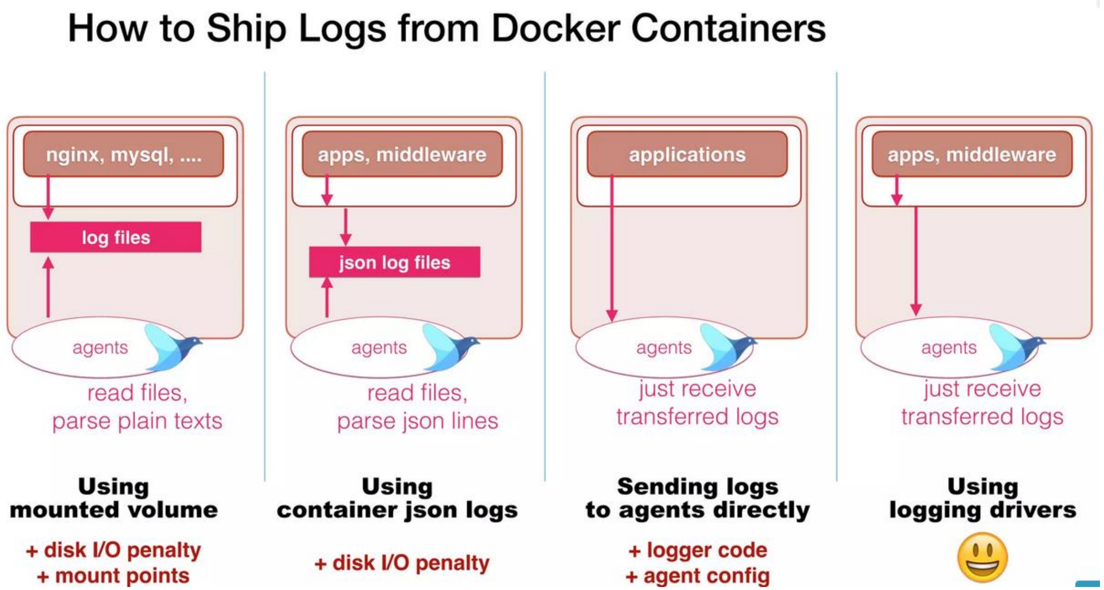
Logstash vs Beats vs Fluentd vs Fluentbit



Source: velebit.ai

Distributed logging

- Microservices (containers)
- Everchanging infra (no fixed storage/network/roles)
  - Transfer logs asap
  - Push logs (instead of pull)
  - Inject names/tags into log records (filter logs based on tags later)

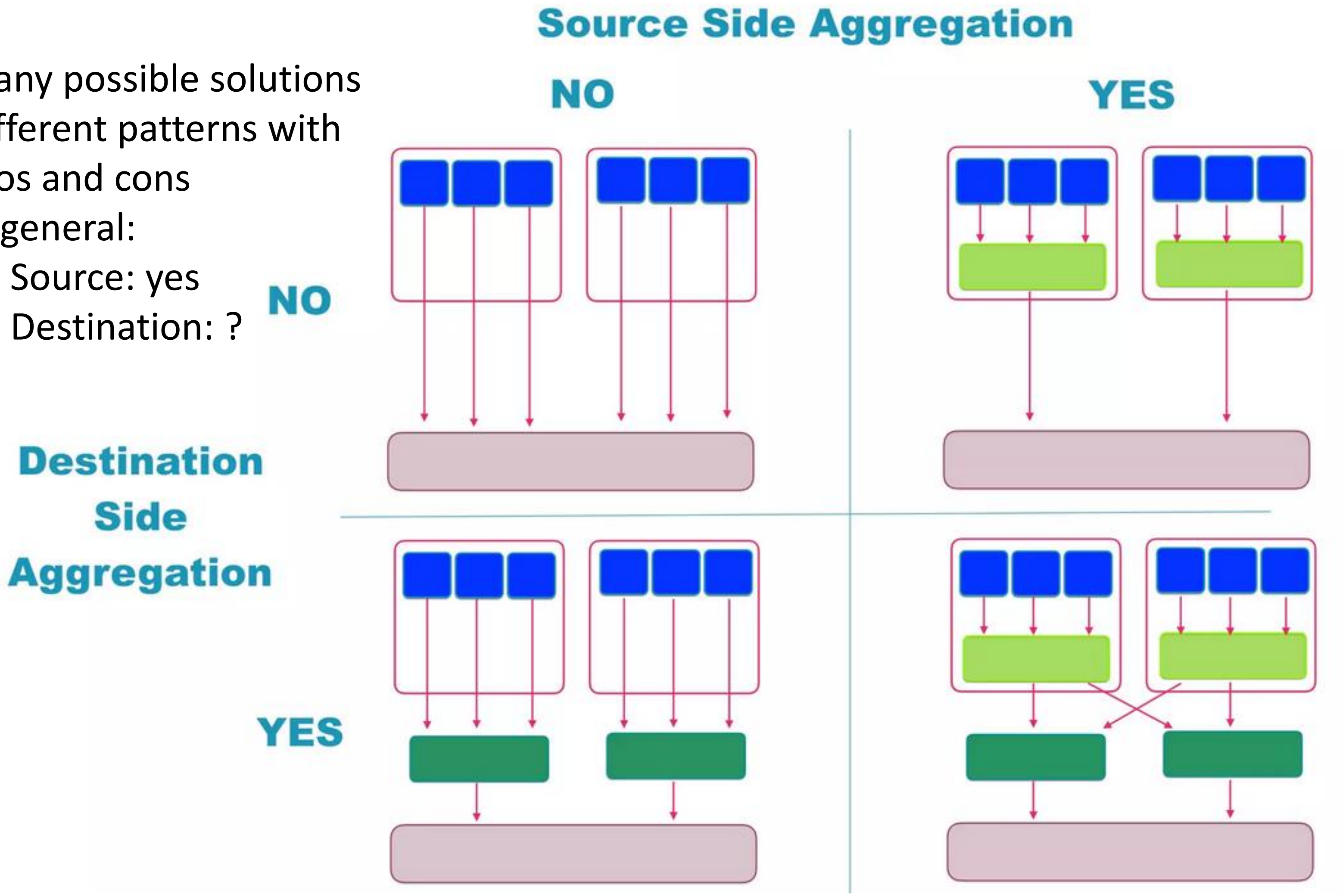


Source:
The Patterns of Distributed Logging and Containers
(by Satoshi Tagomori)

Where to aggregate?

Many possible solutions
Different patterns with pros and cons
In general:

- Source: yes
- Destination: ?



Source: The Patterns of Distributed Logging and Containers (by Satoshi Tagomori)

Network forensics: Introduction

- network-based evidence can be used to
  - confirm or dispel suspicions surrounding an alleged computer security incident
  - accumulate additional evidence and information
  - verify the scope of a compromise
  - identify additional parties involved
  - determine a timeline of events occurring on the network
  - ensure compliance with a desired activity
- collecting network-based evidence includes
  - setting up a computer system to perform network monitoring
  - deploying the network monitor
  - evaluating the effectiveness of the network monitor
- types of network monitoring:
  - event monitoring
    - events are alerts that something has occurred on the network
  - trap-and-trace monitoring
    - records session data summarizing the network activity; includes header fields and flags
  - full content capture
    - acquiring raw packets collected from the ”wire”

Setting up a network monitoring system

- determine your goals for performing the network surveillance
- ensure that you have the proper legal standing to perform the monitoring activity
- acquire and implement the proper hardware and software
- ensure the security of the monitoring platform, both electronically and physically
- ensure the appropriate placement of the monitor on the network
- evaluate your network monitor performance

Determining goals

- goals influence the selection of hardware, software, and filters to be used for collecting network-based evidence, as well as their placement in the network topology
- examples for potential goals:
  - watch traffic to and from a specific host
  - monitor traffic to and from a specific network
  - monitor a specific person's actions
  - verify intrusion attempts
  - look for specific attack signatures
  - focus on the use of a specific protocol
- make sure that the policies in place support these goals
  - e.g., are you allowed to monitor the activity of your employees?

Choosing appropriate hardware

- commercially avaliable network diagnostic and troubleshooting hardware
  - can capture data reliably and at the full rate, but ...
  - lack remote management capabilities and proper storage space
  - usually cost a lot of money
- intrusion detection systems
  - easy to deploy
  - have remote management capabilities and storage space
  - but they cannot perform packet capture reliably
- homegrown solutions
  - easy to customize to fit your needs
  - choose CPU power, amount of RAM, disk space carefully
  - the key issue is to ensure your system has the horsepower required to perform its monitoring function

Choosing appropriate software

- many free tools capture network traffic as well as, or better than, their commercial counterparts
  - e.g., tcpdump, WireShark, Arkime (Moloch)

- selection of the appropriate tool may depend on:
  - which host operating system will you use?
  - do you want to permit remote access to your monitor or access your monitor only at the console?
  - do you want to implement a “silent” network sniffer?
  - do you need portability of the capture files?
  - what are the technical skills of those responsible for the monitor?
  - how much data traverses the network?

Operating system

- choose an operating system with the following features:
  - robust TCP/IP networking stack
  - secure remote access (e.g., via SSH)
  - simple mechanisms for disabling unnecessary services and implementing a local firewall
  - ability to run on many types of hardware, with minimal memory and processor requirements
  - low cost (ideally free)

- Unix/Linux variants are typically good choices
  - e.g., FreeBSD satisfies the above requirements and it is optimized for performance

Stealthy operation

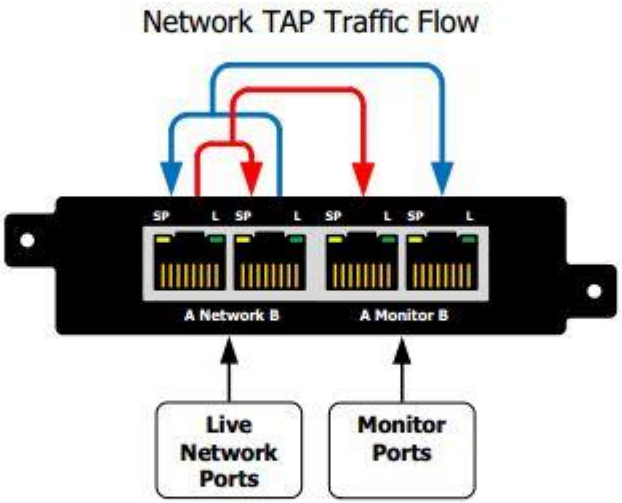
- a ***silent sniffer*** is a system that will not respond to any packets it receives (directed IP datagrams, broadcast, or multicast)

- steps for achieving stealthiness:
  - configure the monitoring interface to speak only TCP/IP
    - » some other protocols, such as NetBIOS, create a lot of traffic that would compromise the location of your monitor
    - » on Windows systems, you need to make sure that you unbind all protocols except for TCP/IP
    - » Unix/Linux systems are generally configured out of the box to communicate with TCP/IP only
  - disable the monitoring interface from responding to ARP packets
    - » most Unix/Linux systems support ifconfig command-line options to turn off ARP on the listening interface
    - » if the monitoring software requires an IP address on the listening interface, assign the system a null IP address (0.0.0.0)
  - alternatively, one can use a one-way Ethernet cable for silent monitoring
    - » disconnect the transmit wires on the network cable
    - » inexpensive, yet very effective

- before deployment of the monitoring system, it is a good idea to run a port scanner (e.g., Nmap) against it, as well as a sniffer detection tool (e.g., Microsoft Promqry or L0pht’s AntiSniff)

Deployment of the monitoring platform

- the placement of the network monitor is possibly the most important factor in setting up a surveillance system
  - modern networks are often switched
  - a switch segments the network
    - each frame is forwarded only via the port that connects the segment where the intended destination resides
    - if the monitor is on another segment, it will not be able to sniff the frame
  - connect the monitor to the SPAN port of the switch
    - special port via which all traffic is forwarded by the switch
  - if the SPAN port is already used, then use an Ethernet tap
    - a tap has (at least) three ports: an A port, a B port, and a monitor port
    - it passes all traffic between ports A and B unimpeded, but it also copies that same traffic to its monitor port



Evaluating the monitoring performance

- things to verify after deployment
  - the traffic monitoring program is executing appropriately
  - the monitoring platform can handle the load
  - the disk isn’t filling up rapidly

- e.g., on Unix/Linux, you can use the `df -h` and `top` commands

What it can and cannot do

- It cannot be used to map out a network
- It does not generate network data-Passive tool
- Only shows detail information about protocols it understand
- It can only capture data as well as the OS\Interface\Interface driver supports.
- An example of this is capturing data over wireless networks.

Wireshark advanced

- Analyze / Expert Information

Wireshark · Expert Information · Wi-Fi

Severity	Summary	Group	Protocol	Count
Note	A new tcp session is started with the same ports as an earli...	Sequence	TCP	
Note	This frame is a (suspected) retransmission	Sequence	TCP	
Chat	Connection establish request (SYN): server port 443	Sequence	TCP	

- Statistics

Wireshark · Conversations · Wi-Fi

Ethernet · 19 | IPv4 · 10 | IPv6 | TCP · 6 | UDP · 6

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
10.105.1.145	32096	192.168.50.152	443	3	198	3	198	0	0	0.035064	3.0130	525	0
10.105.1.145	32097	192.168.50.152	443	3	198	3	198	0	0	0.297399	3.0166	525	0
10.105.1.145	26914	10.105.1.254	443	4	1533	3	1163	1	370	0.434067	6.2598	1486	472
10.105.1.145	1077	152.66.249.134	443	4	1027	2	224	2	803	1.295190	0.9526	1881	6743
10.105.1.254	443	10.105.1.145	9521	4	1253	2	1116	2	137	1.872529	3.4918	2556	313
152.66.249.152	443	10.105.1.145	1064	3	220	2	138	1	82	0.794816	0.0022	—	—

- Custom protocol dissector (c/c++ or Lua)
- Analyse TLS based on saved keys (env: SSLKEYLOGFILE)

Summary

- Logging is essential for security and operations
- Many log standards exist
- Logs must be collected and stored in a searchable form
- Full packet capture is a special form of logging

Control questions

- What is the traditional BSD syslog format?
- What are the drawbacks of standard syslog format?
- What extensions are proposed for syslog?
- What is NetFlow/IPFIX used for?
- How to design a full packet solutions?
- What are the parts of the ELK stack? What is their task?

- Distributed logging: https://www.slideshare.net/tagomoris/the-patterns-of-distributed-logging-and-containers