OWASP Switzerland Chapter
December 2015

OWASP
The Open Web Application Security Project

Reliable log data transfer

About (r)syslog, logstash, and log data signing

A field report

pascal.buchbinder@adnovum.ch

Agenda

- Why we need log data transfer
- Syslog
  - UDP vs TCP
  - Necessary tools (for Apache httpd)
  - Reliability
- Logstash
- Log data signing (Apache httpd, Logstash)

Transferring log data

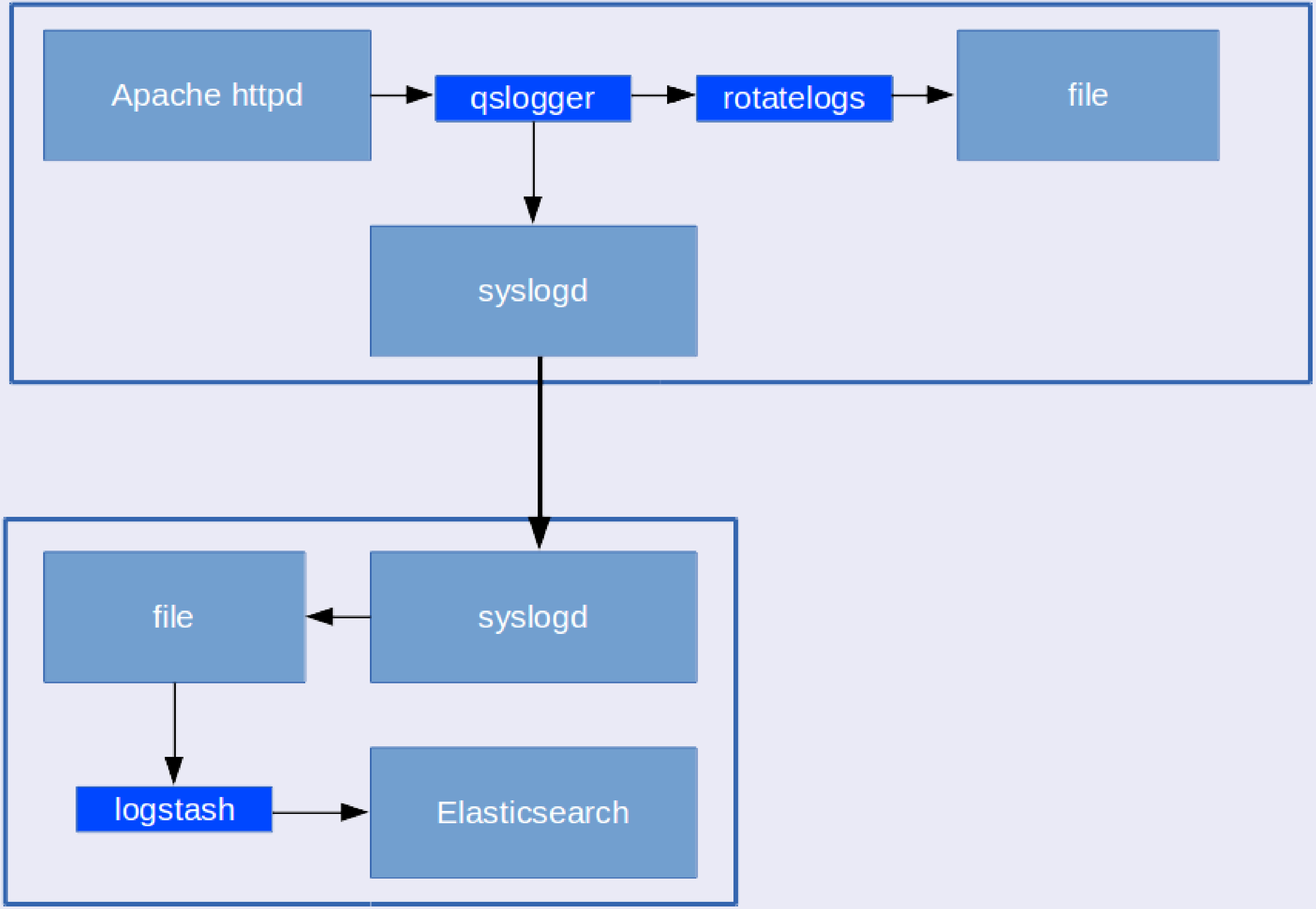
- Many distributed systems
- Need to collect log data centralized
  - Prevention from data loss / manipulation
  - Archiving (transaction audit, PCIDSS)
  - Alerting / monitoring
  - Viewing / troubleshooting
  - Statistics (planning, trends, anomaly)
  - Reporting / DWH
  - Accounting
- Reliability: Are we losing any messages?

syslog vs proprietary - characteristics

- Syslog
  - – Embedded into your software: direct data transfer (via local syslogd or direct connection) to loghost.
    - Software usually available on every (Unix) host.
  - – Standardized protocol, format, levels, facilities, etc.
- Proprietary
  - – External daemon: usually appending to files and forwarding the data to the loghost.
    - Software needs to be installed on every host.
  - – Vendor specific software, configuration, and protocol (server and client side).

Setup syslog

OWASP
The Open Web Application Security Project



Apache httpd and syslog

- Piped logging
- qslogger:
  - Writes data to syslogd and stdout (local file)
  - Filtering by severity (don't forward debug messages to loghost)
  - Severity detection: set message's level at syslog protocol

UDP vs TCP/TLS

- UDP has less overhead (faster)
- Non-blocking
- No flow control
- Plaintext:
  - No confidentiality
  - No key management

- TCP has more overhead (slower)
- Blocking
- Flow control and error handling
- Encrypted:
  - Confidentiality
  - Key management

Syslog: reliable transfer I

- UDP: Potential data loss.
  - If your server (receiver) becomes too busy (thousands of log messes per second from many clients).
  - When your server (receiver) is down.
- TCP: No data loss.
  - As long as your server (receiver) is available.
  - You may configure buffers (memory or file) to store messages temporary if receiver is not available.
  - Syslog is still non-blocking, even we loose messages due to full buffers or an unavailable receiver.

Syslog: reliable transfer II

- Priority: rsyslogd may drop low priority (level) messages preferring high priority ones.

```
$SystemLogRateLimitInterval 2
$SystemLogRateLimitBurst 5000
$SystemLogRateLimitSeverity 6
```

  - **Know your infrastructure's limitation!**
  - Decide whether messages with low severity shall have lower priority.
  - Mainly for UDP setup.

Syslog: reliable transfer III

OWASP
The Open Web Application Security Project

- Cluster: setup a primary and secondary log host.

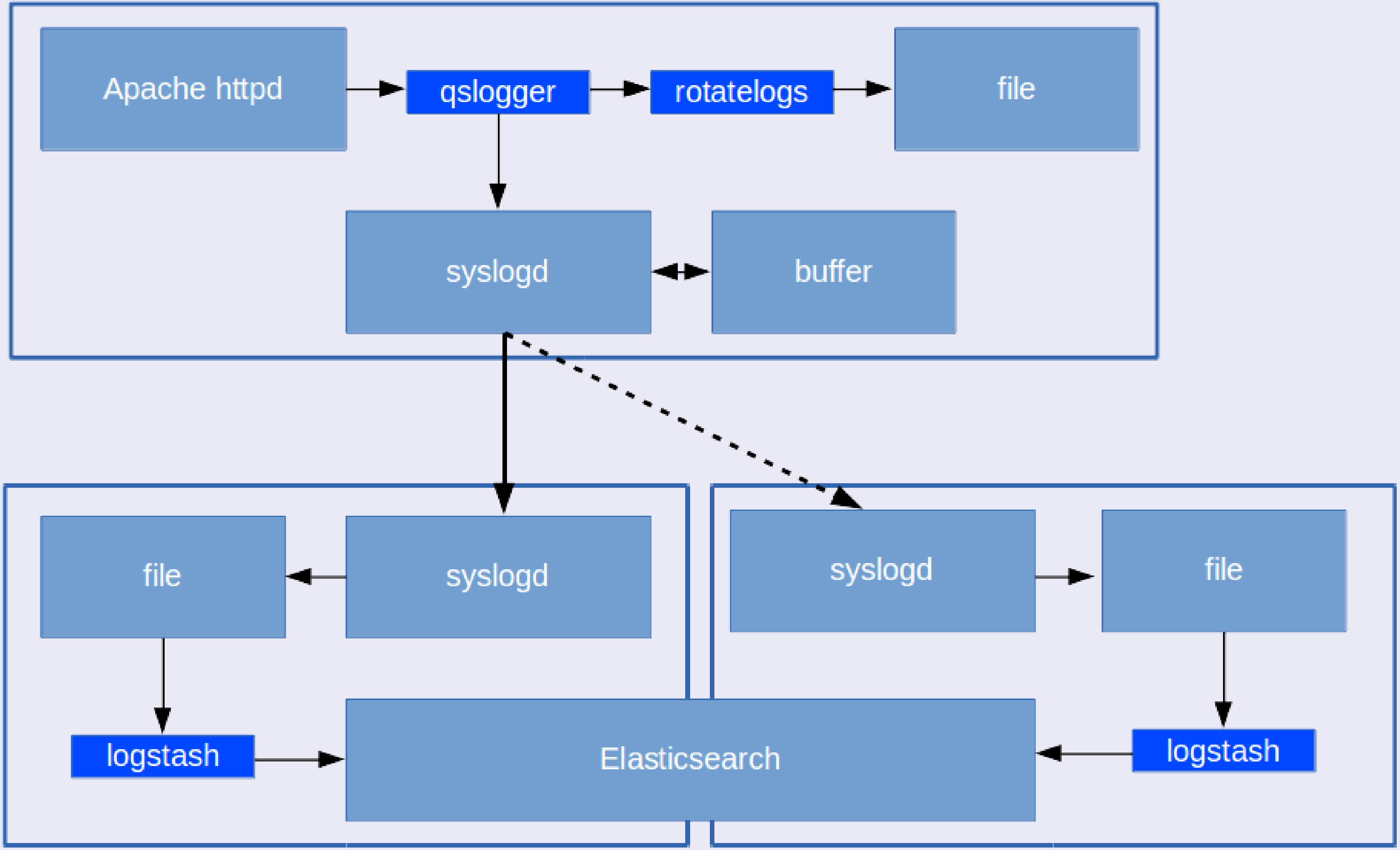
```
local3.* @@logmaster.adnovum.ch
$ActionExecOnlyWhenPreviousIsSuspended on
&@@logslave.adnovum.ch
& /var/log/syslogbuffer
$ActionExecOnlyWhenPreviousIsSuspended off
```

  - Allows you to maintain your log host (receiver), e.g. reboot the server.
  - Makes only sense within a setup using TCP.

Setup syslog - cluster

OWASP

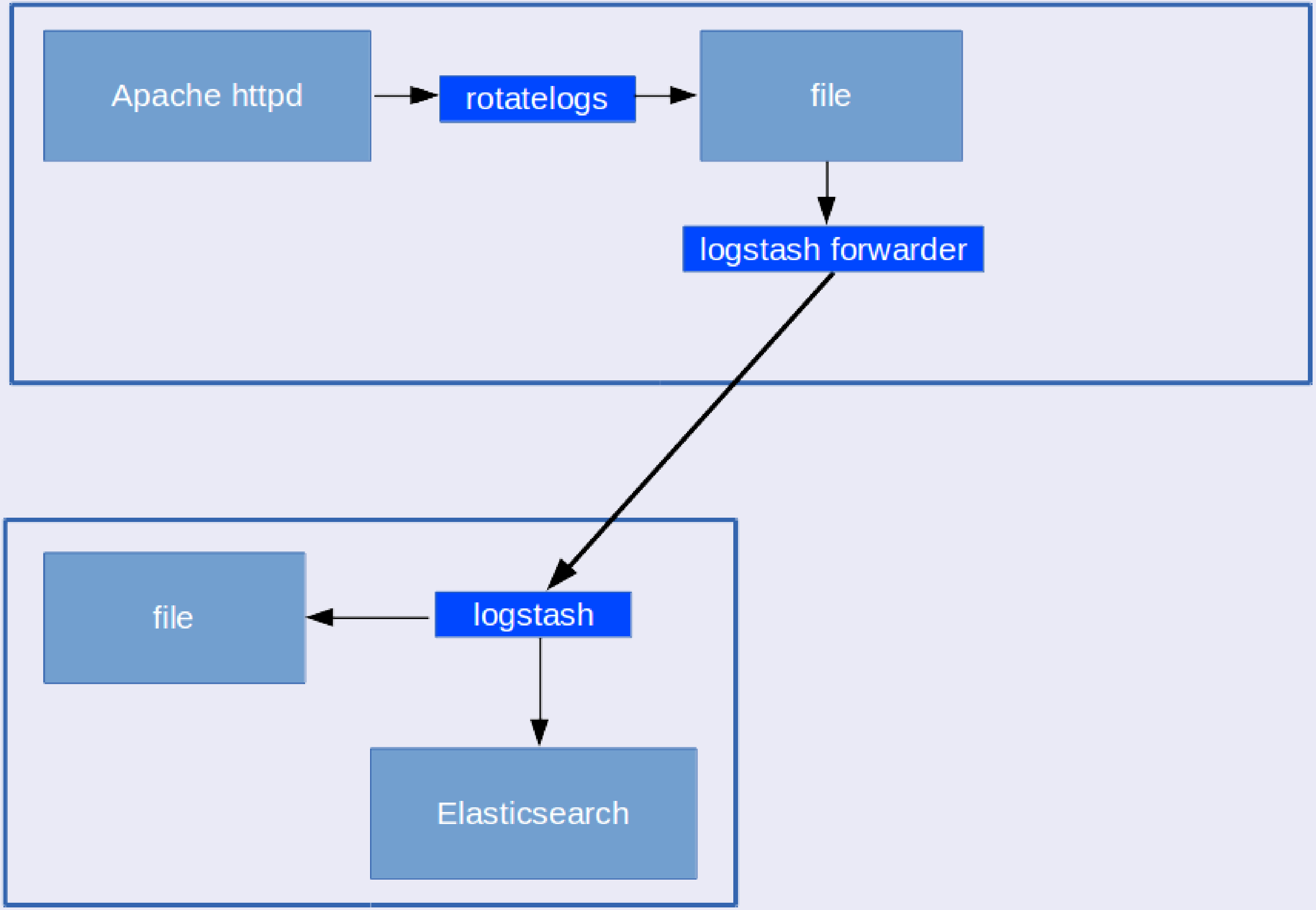
The Open Web Application Security Project



Logstash forwarder

- Open Source (Apache license)
- Locally installed daemon
- Appends to files
  - Free buffers: buffering works event when files are rotated while reading (no message loss until a rotated file gets deleted before the data has sent)
- Forward data using the lumberjack protocol
- Supports 2-way SSL (mutual authentication)

Logstash forwarder setup



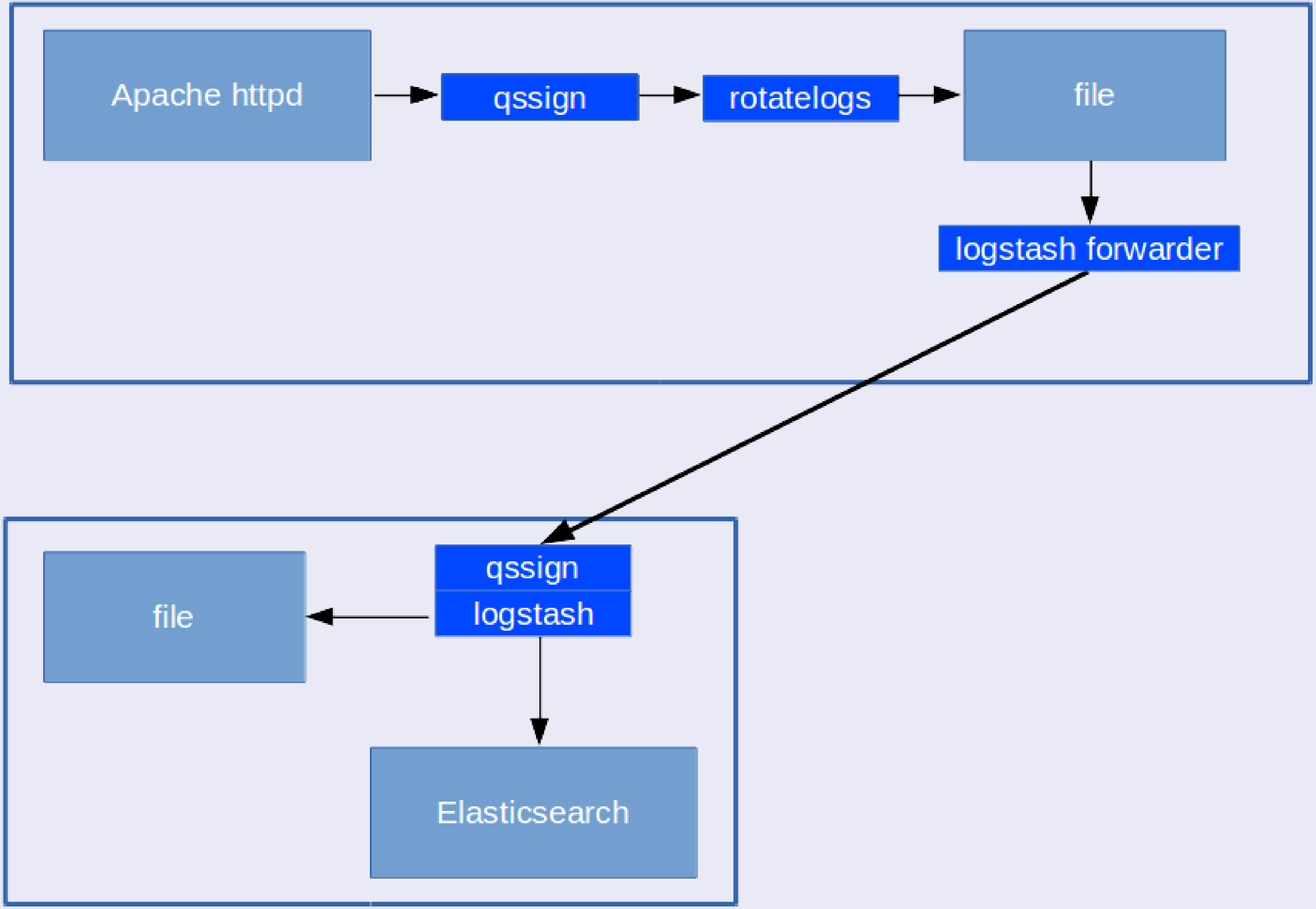
Signatures

- How do you know not loosing log data?
  - – Easy in lab while testing (by counting messages).
  - – Difficult in real production environment.
- Signing log data solves this problem:
  - – Each message is signed (corrupt messages).
  - – Sequence number shows data loss (or injected messages).

Signatures - setup

OWASP

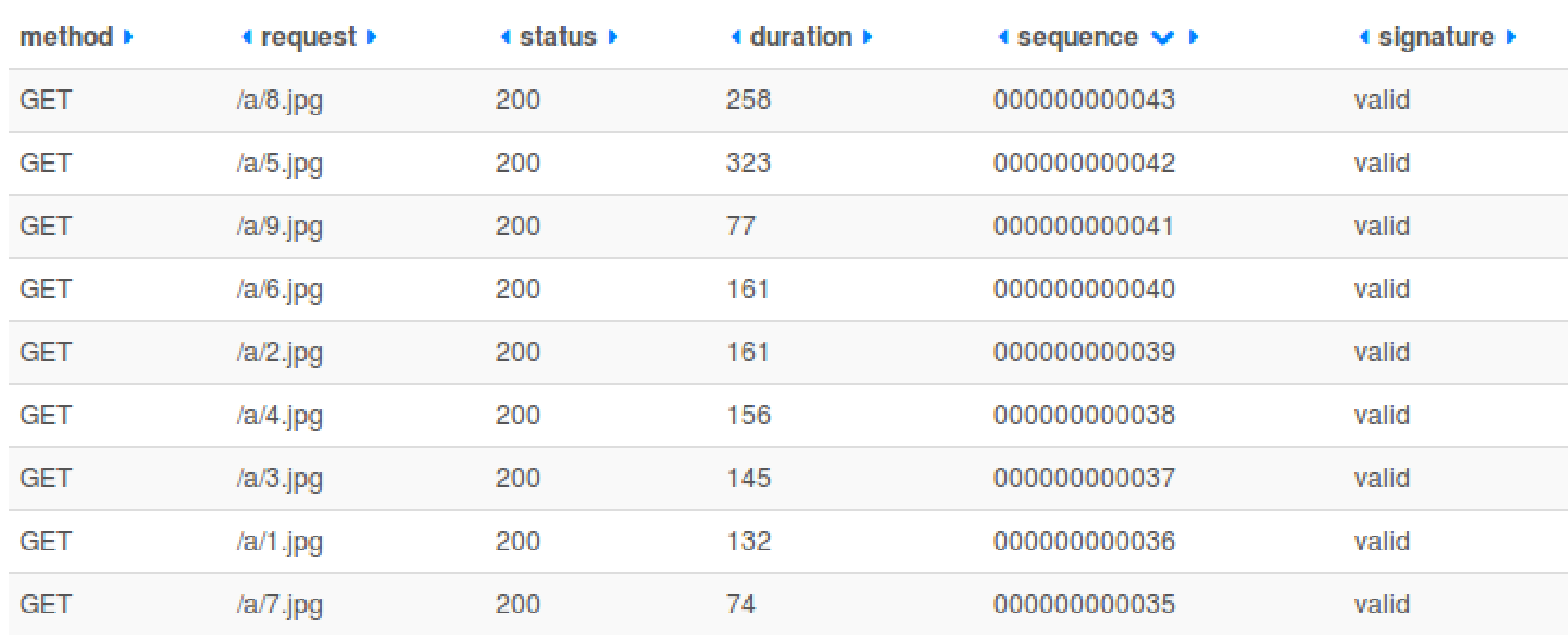
The Open Web Application Security Project



Signatures - tools

- Sign each message using a dedicated tool (qssign, piped logging, PSK)
  - Adds sequence number and signature.
- Verify signature by using a logstash filter plug-in
  - Verifies each message at server (receiver): signature and sequence number.
  - Potential problem: checking sequence within a cluster setup (switching multiple times).

Signatures - ELK



method	request	status	duration	sequence	signature
GET	/a/8.jpg	200	258	000000000043	valid
GET	/a/5.jpg	200	323	000000000042	valid
GET	/a/9.jpg	200	77	000000000041	valid
GET	/a/6.jpg	200	161	000000000040	valid
GET	/a/2.jpg	200	161	000000000039	valid
GET	/a/4.jpg	200	156	000000000038	valid
GET	/a/3.jpg	200	145	000000000037	valid
GET	/a/1.jpg	200	132	000000000036	valid
GET	/a/7.jpg	200	74	000000000035	valid

grok&qssign filter configuration

```
filter {
  grok {
    match => [ "message", "%{GREEDYDATA:data} %{INT:sequence}#%{NOTSPACE:hmac}" ]
    tag_on_failure => [ ]
  }
  if [data] {
    qssign {
      message => "data"
      source => "path"
      sequence => "sequence"
      hmac => "hmac"
      secret => "/var/opt/keys/keypass.sh"
    }
    mutate {
      replace => [ "message", "%{data}" ]
      remove_field => [ "data" ]
    }
  } else {
    mutate {
      add_field => [ "signature", "missing" ]
    }}}
```

Summary

- Syslog
  - Usually works “out of the box”
  - UDP works well under “normal” conditions
  - No data loss when using a TCP and a cluster/buffers
- Logstash forwarder
  - Additional software, configuration, certificates
  - No data loss even your log host is temporary down
- Signatures
  - Know when loosing data
  - No message injection or manipulation

Software

- http://opensource.adnovum.ch/mod_qos/qslogger.1.html
- http://opensource.adnovum.ch/mod_qos/qssign.1.html
- http://mod-qos.cvs.sourceforge.net/viewvc/mod-qos/src/tools/logstash-filter-qssign/
- http://www.rsyslog.com/
- https://www.elastic.co/products/logstash