LogData Analysis System
→ Idea and Realization

Prof. Dr. (TU NN)
Norbert Pohlmann

Institute for Internet Security - if(is)
University of Applied Sciences Gelsenkirchen
https://www.internet-sicherheit.de

Content

Content

LogData Analysis System (LAS)
→ Aims and outcomes of this lecture

Aims

- To introduce an Internet Early Warning System with a log-data approach

- To explore the structure of the LogData Analysis System (LAS)

- To analyze the results of the LogData Analysis System (LAS)

- To assess the value the LogData Analysis System (LAS)

At the end of this lecture you will be able to:

- Understand what is meant by the LogData Analysis System.

- Know something of the structure of the LogData Analysis System.

- Know what the results of the LogData Analysis System could be.

- Understand the capabilities and limitations of the LogData Analysis System.

Content

© Prof. Norbert Pohlmann, Institute for Internet Security - if(is), University of Applied Sciences Gelsenkirchen, Germany

- **Analysis of the raw data on the network layer,** captured "off the wire"

  - **Analysis of the direct communication**

  - Detection of the steps of an attack, shortly before they are performed
    - Analysis by the interpretation of anonymized communication parameters
      - E.g. **Internet Analysis System** (The entire communication is being monitored.)

  - Analysis of the sensitive communication data (content)
    - **Intrusion Detection Systems** like **Snort**

- **Analysis of LogData on the host/application level**

  - High quality information

  - LogData describe complex incidents on a higher level than the communication parameters captured from the line

  - **Analysis of the result of the communication**

    - Analysis by the **rating of facts / real incidents!**

  - Detection of a step of an attack, after it has been performed!

  - *Enables*

    - *Conclusions* about the *reaction of the system*, that was target of the attack

    - *Reconstruction* of the *attack chain*

  - E.g. **LogData Analysis System** of the **if(is)**

- **Log data / log / logbook**

  „… is the type of record originating form nautical shipping to record daily events and procedures similar to a diary."[1]

- **Log file**

  „... contains the automatically produced log of all or selected actions of processes on a computer system.."[1]

1) Wikipedia

- **Syslog**

  „... is the de-facto-standard for the transfer of log data in an IP network.
  The term "syslog" is commonly used for the actual syslog network protocol as well as for the application or library, which sends and receives syslog messages."[1]
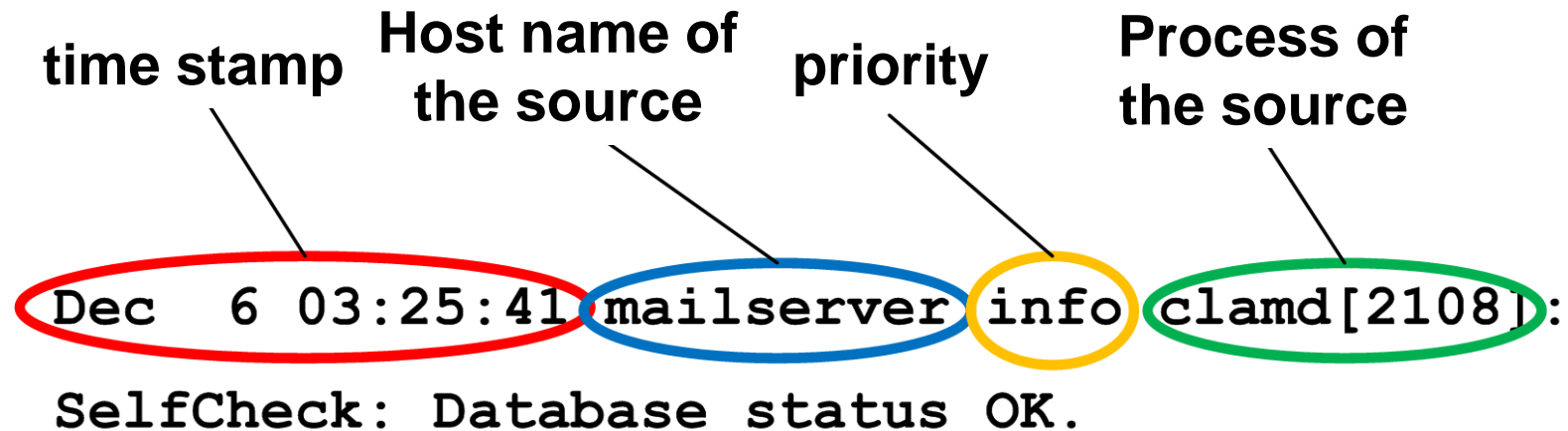

- **syslog-ng**

  „... is the common syslog server on a linux- and unix platform.
  This program implements the syslog protocol and offers a number of extensions, which are supposed to correct known vulnerabilities of the protocol.
  Syslog-ng is an open source software solution."[1]

1) Wikipedia

LogData
→ Composition of a syslog message

- **Syslog** messages can be composed of different fields, depending on the used system configuration.

- *Example for a typical syslog message:*

<div align="center">

time stamp   **Host name of the source**   **priority**   **Process of the source**

`Dec  6 03:25:41` `mailserver` `info` `clamd[2108]:`

`SelfCheck: Database status OK.`

</div>

- The lower part is the actual message

LogData
→ General information

- Contain important information for the system administrator about
    - shortages in resources
    - hard- und software problems
    - **security problems** and **attacks**

- Are often neglected and underestimated (Method "LogData")
    - Large, unclear amounts of data
    - Large initial effort
    - Bad coding, no common format for logs

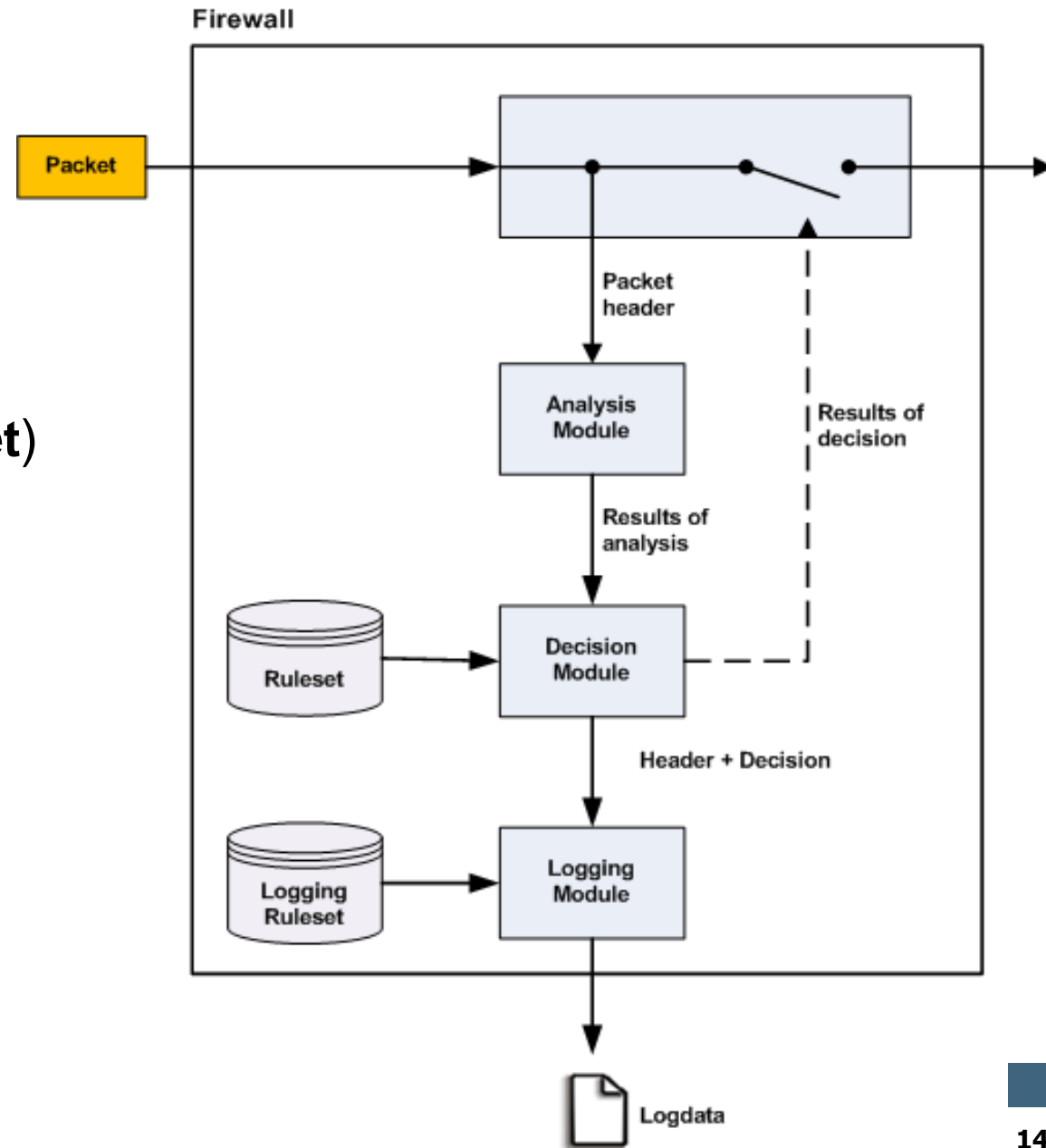- Can be used as basis for **forensics** and **legal prosecution** of attackers

Content

- **Right amount** of log data is an important input parameter of the system

- The more log data, the more information and the better the conclusions
    - if to little is logged, the system won't work
    - if everything is logged, the system won't work as well
        - important information will be suspended by noise
        - Performance

- Whether a log entity consists security relevant information, can most of the time only be determined in correlation to other log entities
    - One miscarried connection is most likely a lapse
    - If another 999 miscarried connections are recorded, it was most likely one puzzle piece of an attack to gain access

- Security relevant information hidden in log data **< 5%**, most of the time **implicit**

- **Example firewall**

  - Packet Headers are logged

  - But not every header

    - avoid extreme overhead

    - definition of rules for logging (**Logging Ruleset**) within the firewall

    - correct behavior of the system depends on this logging rule set

    - changes of firewall restrictions (firewall rule set) **do not** influence the functioning


Firewall

Packet

Packet header

Analysis Module

Results of decision

Results of analysis

Ruleset

Decision Module

Header + Decision

Logging Ruleset

Logging Module

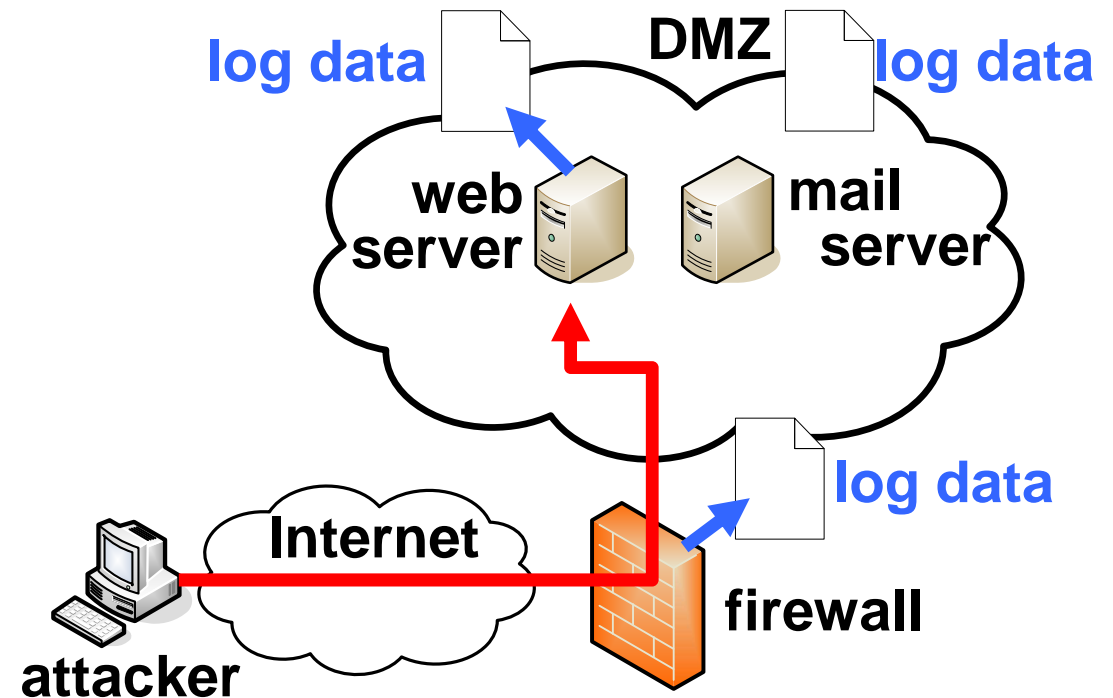Logdata

- **Firewall logs**

  - ***always the first packet*** of a connection request, therefore one log entity per TCP connection

  - all packets, which were ***rejected or dropped*** by the firewall

- Reduction of packet header to a smaller subset, which is logged

- Most attacks, which can be detected by firewall log files, can be detected with the help of this reduced subset as well

SYN

SYN/ACK

ACK

data transmission

Log packet

FIN

FIN/ACK

ACK
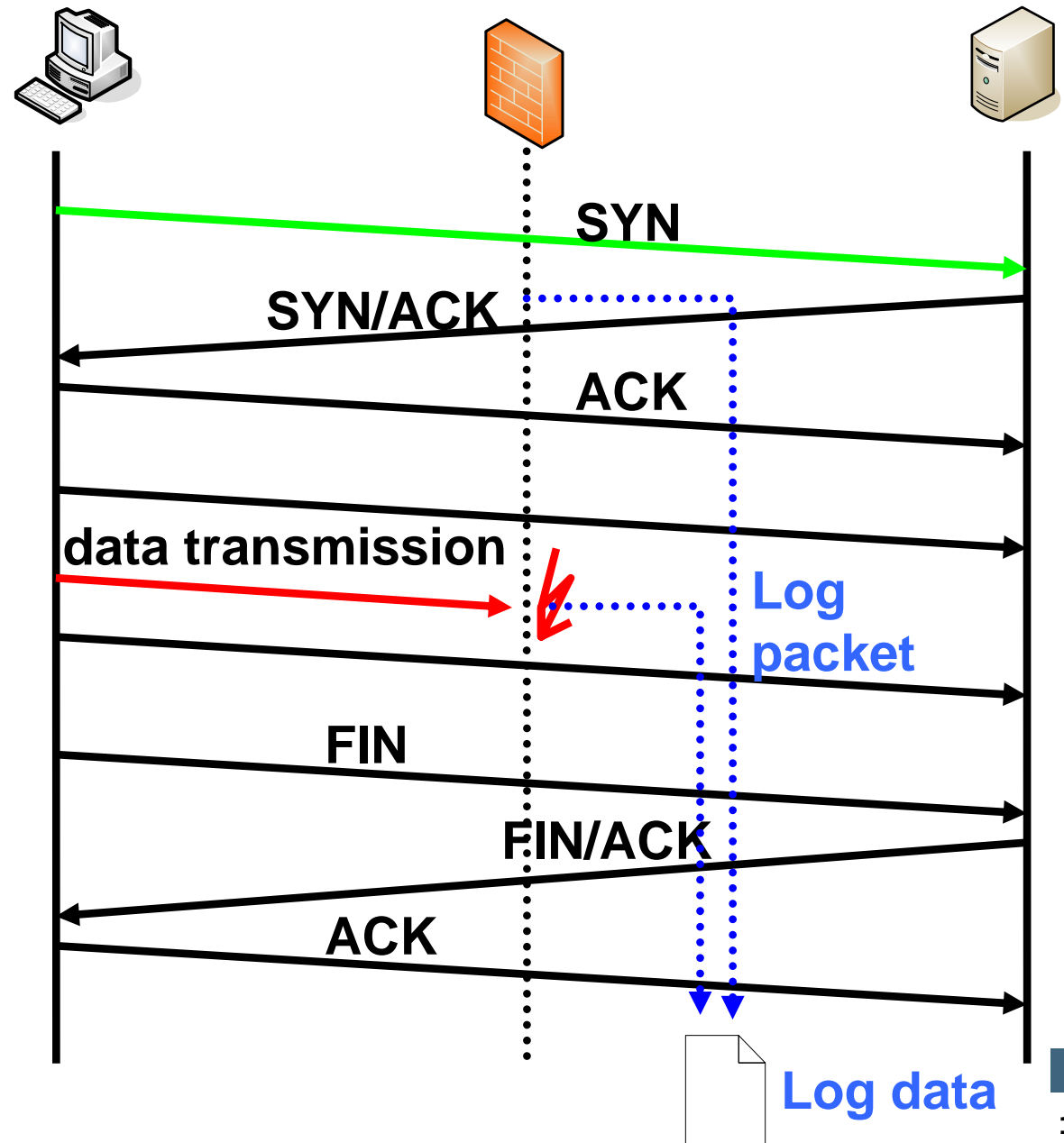
Log data

LogData Analysis System
→ Idea (1/3)

- In log files the entire communication of a service is recorded

- Communication with potential attackers is therefore also recorded

- Behavior of the attacker considerably differs from the behavior of a normal user

- Attacks therefore leave significant patterns in the log files
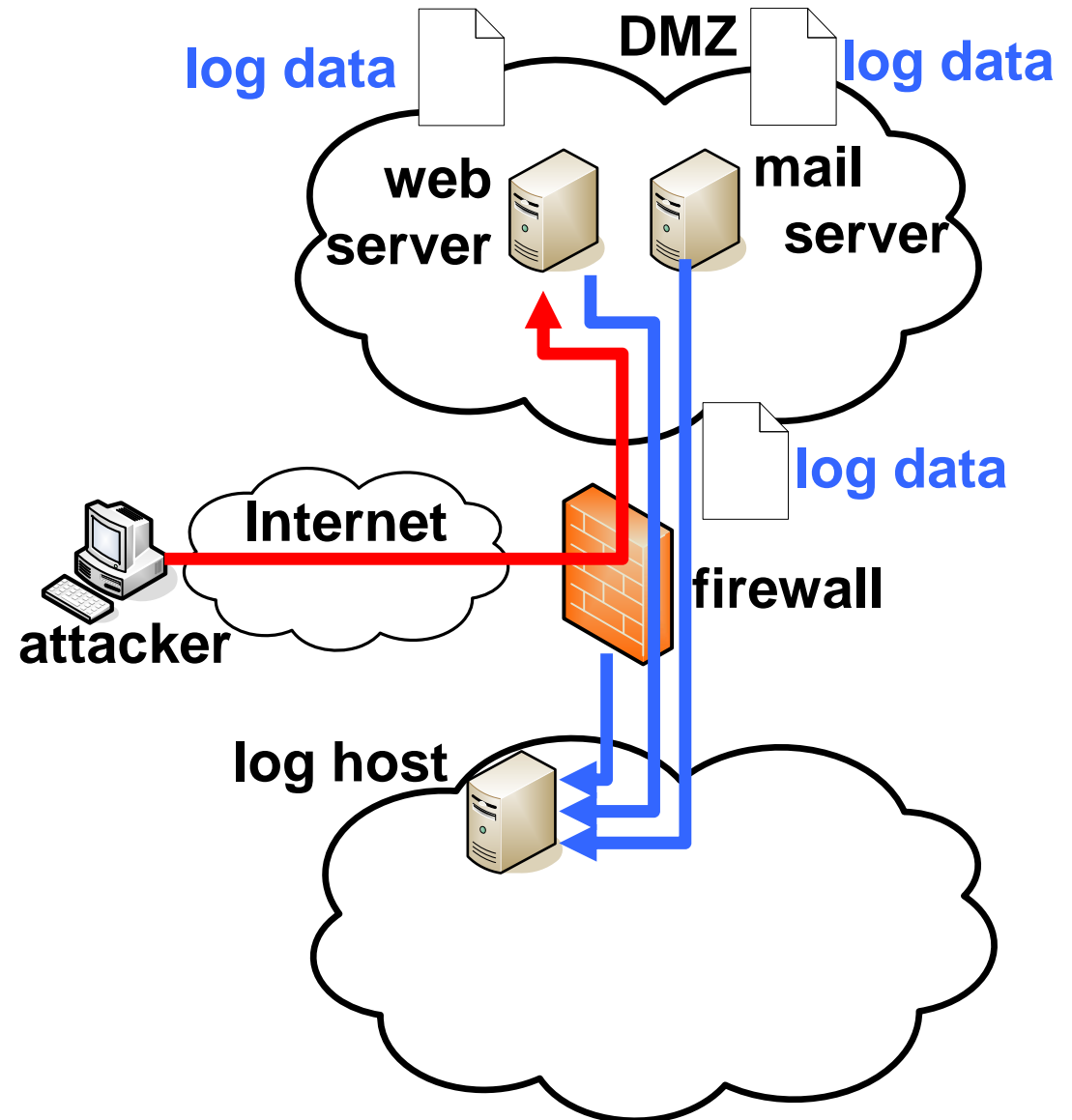
LogData Analysis System
→ Possible sources of log data

- Components connected to the Internet are especially qualified for the detection of security relevant incidents, like e.g.

  - Firewalls (e.g. Iptables)

  - Mail server (e.g. Sendmail, Postfix)

  - Web server (e.g. Apache)

  - VPN server (e.g. OpenVPN)

  - DNS server (e.g. BIND)

  - VoIP server (e.g. Asterisk)

  - NIDS (Network Intrusion Detection Systems, e.g. Snort)

  - Remote Shell (e.g. sshd)

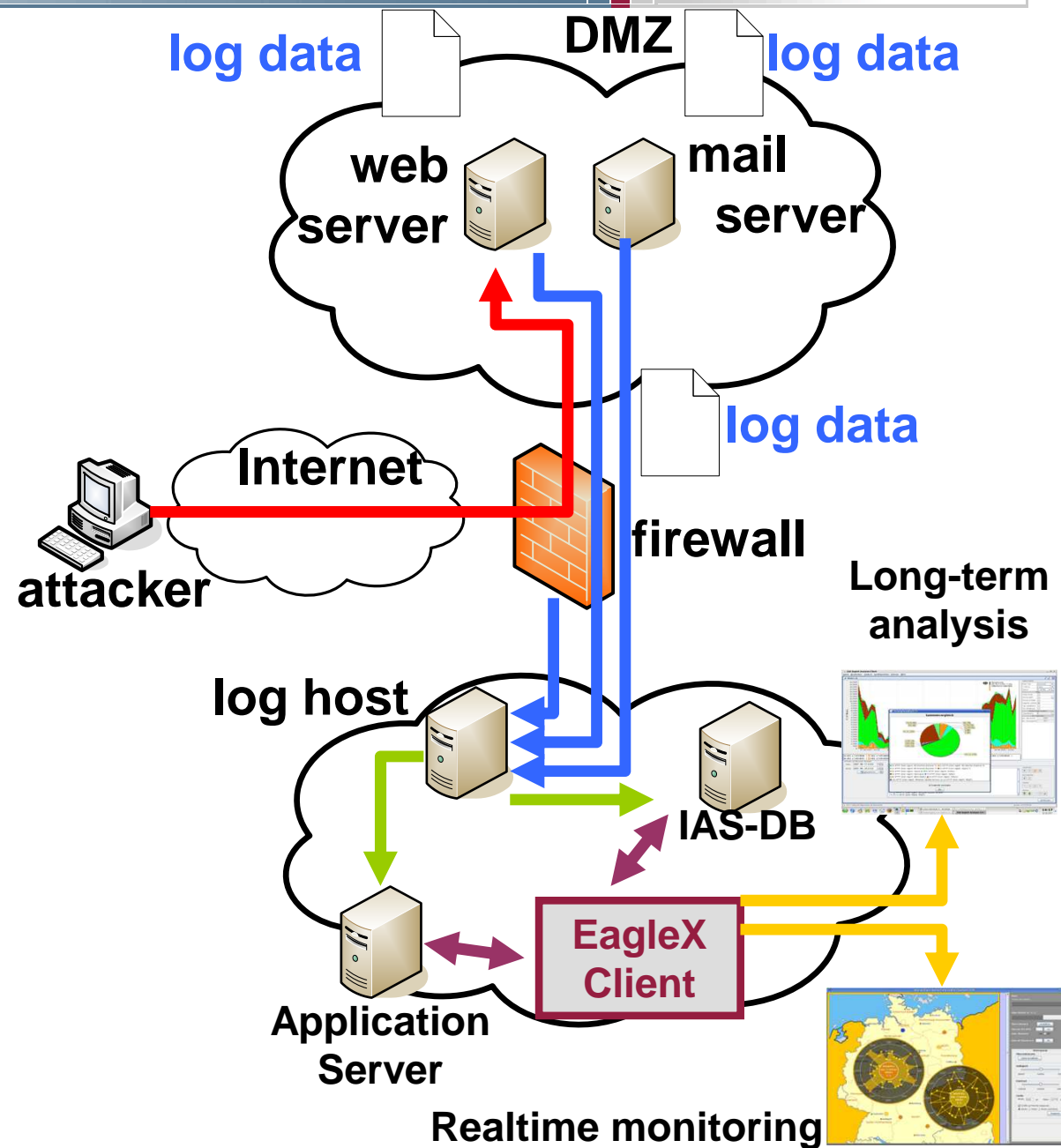 - etc.

LogData Analysis System
→ Idea (2/3)

- Log files are combined as a real time data flow at a centralized log host (**Centralized Logging**)

- Improves the clarity

- Correlation of the data

- Easier processing

- Realtime log data flow is being analyzed to detect attacks (**Intrusion Detection**)

  - **Anonymized long-term analysis**

  - **Realtime monitoring and alerting**

log data **DMZ** **log data**

web server **mail server**

log data

Internet

attacker **firewall**

Long-term analysis

log host

IAS-DB

Application Server

EagleX Client

Realtime monitoring

- **Proceeding**

  - Data flow is analyzed based on rules almost in realtime

  - Detection of attack patterns in a **local view**

  - Use of proper algorithms and analyzing methods

    - **Detection of patterns**

      - Precise detection of known patterns

        - e.g. SYN/FIN-Scan

    - **Anomaly Detection**

      - Detection of anomalies in the communication by the use of threshold analysis on layer 4

      - Resolution by the use of a heuristic

      - Dynamic adaptation of the threshold based on the threshold values over the prior 30 minutes
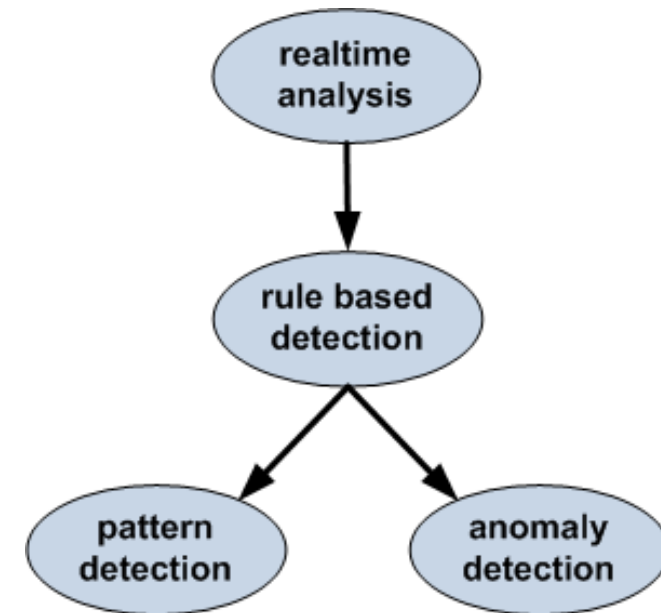
- **Goals**

  - Alerting in the case of an detected attack

  - Timely reaction on attacks

  - Take counteractive measures against a concrete threat

    - Stop services, close ports, deactivate user accounts, shut down systems, ...

  - **Minimization of damage**

  - Can be used as basis for **forensics** and **legal prosecution** of attackers

- Alerts are connected to the original log data and provide these in the scenario of damage

  - Contain sensitive, possibly privacy relevant information

    - IP addresses

    - E-mail addresses

    - Usernames (and sometimes passwords)

  - Information about the alerts are only provided to the operator of the LogData Analysis System.

- **Proceeding**

  - Applying of the principle of using **parameters (descriptors)** (tally sheets) to the logged incidents

    - Definition of events in the log files as parameters for the tally sheet (descriptors)

    - Counting of the occurrence of these events (log data)

    - Visualization of the occurrence corresponding to the time line.

    - …

  - **Anonymization** of the log data

  - Recycling of all IAS applications (tools) possible

    - Report system

    - Neural networks

  - At the moment the occurrence for more than 650.000 different parameters (descriptors) for incidents in firewall log data is monitored

- **Goals**

  - Complement the data stock (Knowledge Base) the **IAS**

  - Realization of a **reference system** to correlate the results of the IAS

  - Combination of the statistical (anonymized) log data of different local networks to one **global view**

  - Statistical analyzing of the log data

    - Description of patterns, profiles and technology trends

    - Overview on the current state of the Internet

    - Detection of attacks and anomalies

    - Forecast of patterns and attacks

Synergy between
→ Realtime and long-term analysis
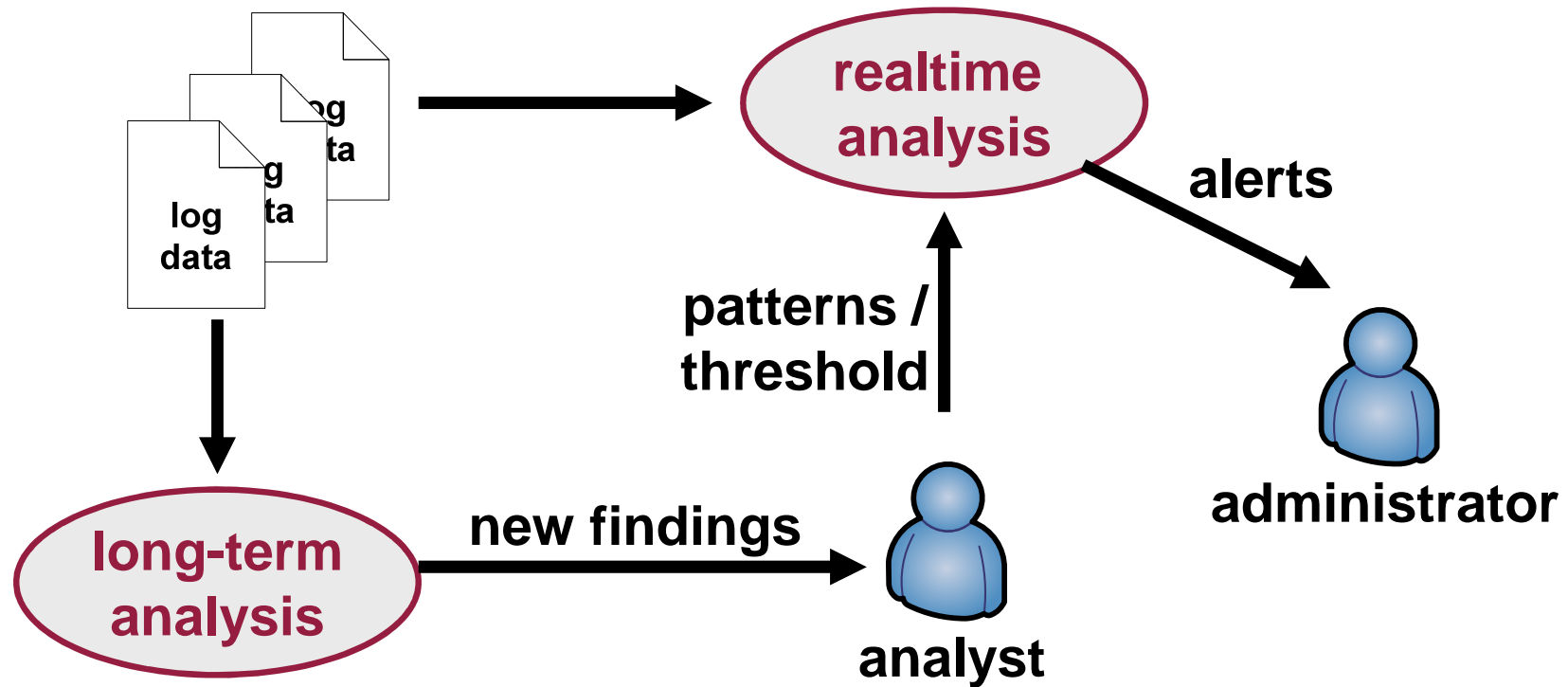
- Realtime analysis needs input from long-term analysis
  - Evaluation of the communication behavior
  - Derivation of thresholds and patterns
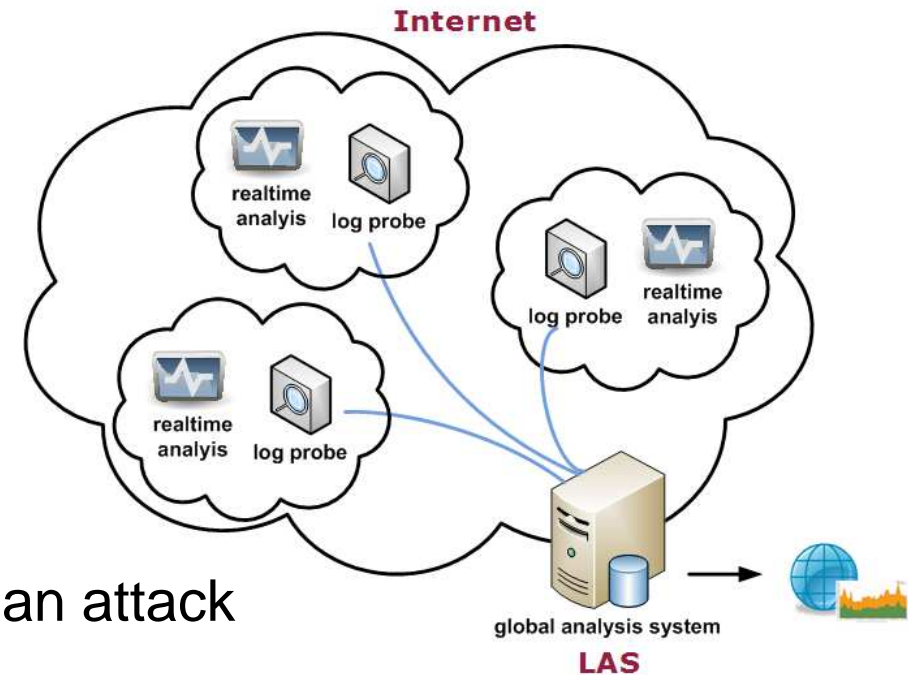
LogData Analysis System
→ Privacy

- **Long-term analysis**

  - Principle of the counter values for parameters (tally sheet)

  - **Anonymization by design**

  - No violation of privacy laws

- **Realtime analysis**

  - Visualization of the log data in case of an attack (damage has occurred)

    - Just log data, which can identify attacks, is displayed

    - All other log data is dropped (after a period of time)

  - Log data is not automatically stored

  - Log data of incidents is removed after 24 hours (or 7 days) from the GUI of the LogData Analysis System

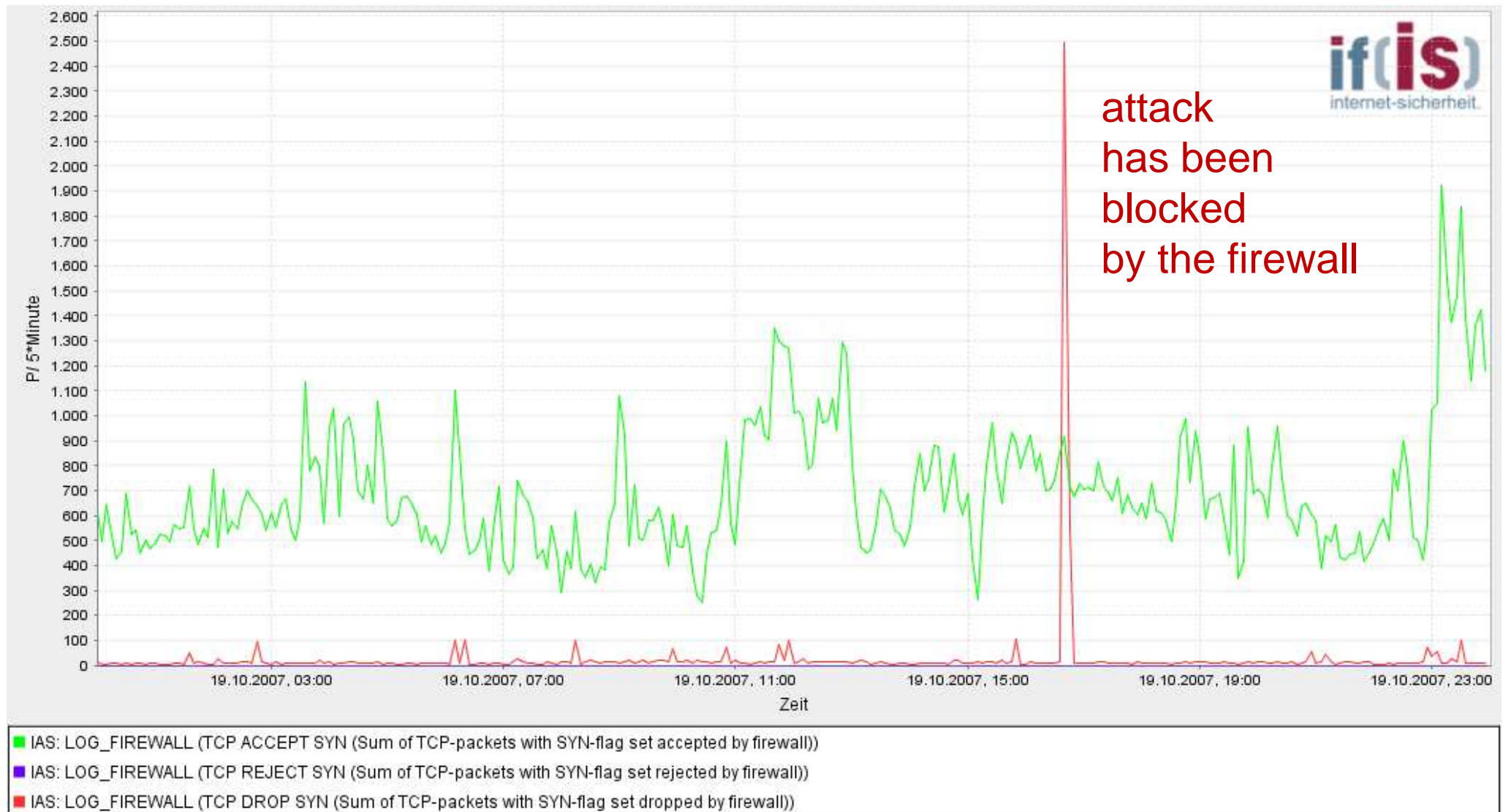 - Important log data can manually be stored (legal actions)

Internet

realtime analyis   log probe

log probe   realtime analyis

realtime analyis   log probe

global analysis system

LAS

Content

- **Aim and outcomes of this lecture**

- **Classification**

- **LogData Analysis System**

- # Examples

- **Data flow and data management**

- **Summary**

■ TCP **ACCEPT**, **REJECT**, **DENY (DROP)**

attack
has been
blocked
by the firewall

■ IAS: LOG_FIREWALL (TCP ACCEPT SYN (Sum of TCP-packets with SYN-flag set accepted by firewall))
■ IAS: LOG_FIREWALL (TCP REJECT SYN (Sum of TCP-packets with SYN-flag set rejected by firewall))
■ IAS: LOG_FIREWALL (TCP DROP SYN (Sum of TCP-packets with SYN-flag set dropped by firewall))
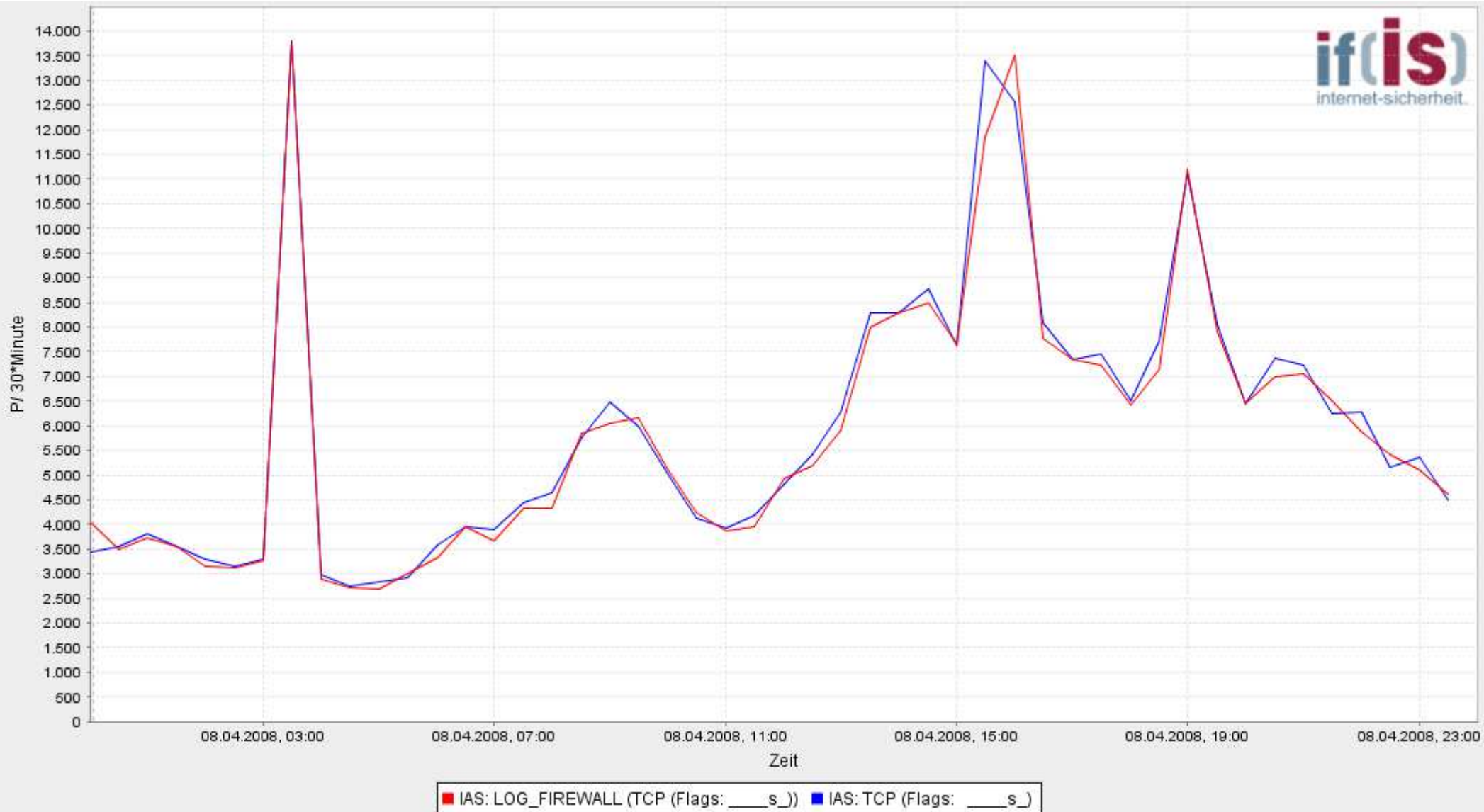
LogData Analysis System
→ Evaluation

- Both curves are recorded by the LAS

    - Green shows that the firewall accepts TCP packets

    - Red shows the count of packets denied by the firewall, in this case an attack has been blocked

LogData Analysis System
→ Example − reference system

- What information can we extract from the figure?

- Both curves are almost identical

- Difference: different time periods

- We can extract, that the LAS can be used as a reference system in some environments for the IAS, since similar findings are achieved.

LogData Analysis System
→ Example – detection of anomaly

- **Buffer overflow attack** against a "ssh" daemon

- Log entity is **not common** during **normal operation**

```
Oct 11 14:27:26 host sshd[6169]: fatal: Local: Corrupted check bytes
                                  on input.
Oct 11 14:27:28 host sshd[6253]: fatal: Local: crc32 compensation
                                  attack: network attack detected
```

- **Dictionary attack against "ssh" daemon**

  - **Normal behavior**

    - User logs in with his username on the remote system

    - Most of the time he will not need more than **3 false attempts**

    - Communication on port 22

  - **Attack situation**

    - Attacker tries different usernames over a huge period of time to gain access to the system

    - Time slots between attempts are so short, that these attempts can be identified as machine initiated

LogData Analysis System
→ Example – threshold analysis (2/4)

- **Dictionary attack against "ssh" daemon:** sshd logs
  - → **Invalid User**

```
Oct 11 22:53:05 listserver sshd[17200]: Invalid user diablo from ::ffff:213.195.77.228
Oct 11 22:53:05 listserver sshd[17202]: Invalid user b1ablo from ::ffff:213.195.77.228
Oct 11 22:53:06 listserver sshd[17204]: Invalid user paradise from ::ffff:213.195.77.228
Oct 11 22:53:07 listserver sshd[17206]: Invalid user paradisse from ::ffff:213.195.77.228
Oct 11 22:53:07 listserver sshd[17208]: Invalid user baggio from ::ffff:213.195.77.228
Oct 11 22:53:08 listserver sshd[17210]: Invalid user roberto from ::ffff:213.195.77.228
Oct 11 22:53:08 listserver sshd[17212]: Invalid user kim from ::ffff:213.195.77.228
Oct 11 22:53:09 listserver sshd[17214]: Invalid user space from ::ffff:213.195.77.228
Oct 11 22:53:10 listserver sshd[17216]: Invalid user globe from ::ffff:213.195.77.228
Oct 11 22:53:10 listserver sshd[17218]: Invalid user oscar from ::ffff:213.195.77.228
Oct 11 22:53:11 listserver sshd[17220]: Invalid user simbol from ::ffff:213.195.77.228
Oct 11 22:53:11 listserver sshd[17222]: Invalid user addicted from ::ffff:213.195.77.228
Oct 11 22:53:12 listserver sshd[17224]: Invalid user red from ::ffff:213.195.77.228
Oct 11 22:53:12 listserver sshd[17226]: Invalid user pink from ::ffff:213.195.77.228
Oct 11 22:53:13 listserver sshd[17228]: Invalid user blue from ::ffff:213.195.77.228
Oct 11 22:53:14 listserver sshd[17232]: Invalid user postgres from ::ffff:213.195.77.228
Oct 11 22:53:15 listserver sshd[17234]: Invalid user accept from ::ffff:213.195.77.228
Oct 11 22:53:15 listserver sshd[17236]: Invalid user leo from ::ffff:213.195.77.228
Oct 11 22:53:16 listserver sshd[17238]: Invalid user zeppelin from ::ffff:213.195.77.228
Oct 11 22:53:16 listserver sshd[17240]: Invalid user hacker from ::ffff:213.195.77.228
Oct 11 22:53:17 listserver sshd[17242]: Invalid user olga from ::ffff:213.195.77.228
Oct 11 22:53:18 listserver sshd[17244]: Invalid user boris from ::ffff:213.195.77.228
Oct 11 22:53:18 listserver sshd[17246]: Invalid user mathew from ::ffff:213.195.77.228
Oct 11 22:53:19 listserver sshd[17248]: Invalid user testing from ::ffff:213.195.77.228
Oct 11 22:53:19 listserver sshd[17250]: Invalid user galaxy from ::ffff:213.195.77.228
Oct 11 22:53:21 listserver sshd[17254]: Invalid user venice from ::ffff:213.195.77.228
Oct 11 22:53:21 listserver sshd[17256]: Invalid user user3 from ::ffff:213.195.77.228
Oct 11 22:53:22 listserver sshd[17258]: Invalid user sa from ::ffff:213.195.77.228
```

LogData Analysis System
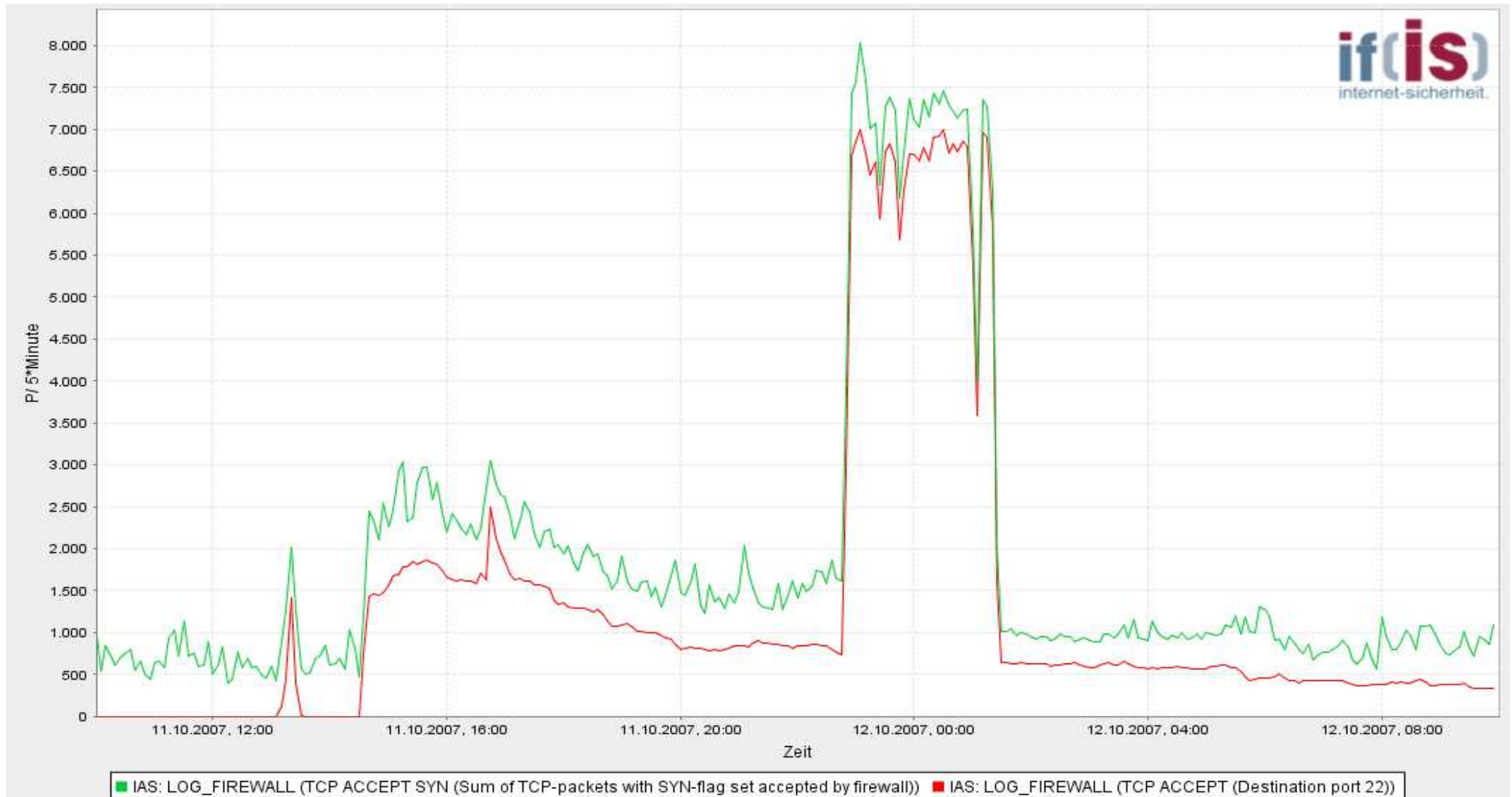## → Example – threshold analysis (3/4)

- **Dictionary attack against "ssh" daemon:** iptables logs

```
Oct 11 22:53:06 fb5gwint kernel: forward Rule 13 - ACCEPT IN=eth0 OUT=eth2
SRC=213.195.77.228 DST=194.94.127.15 LEN=60 TOS=0x00 PREC=0x00 TTL=51 ID=16514 DF PROTO=TCP
    SPT=42178 DPT=22 WINDOW=5840 RES=0x00 SYN URGP=0
Oct 11 22:53:06 fb5gwint kernel: forward Rule 13 - ACCEPT IN=eth0 OUT=eth2
SRC=213.195.77.228 DST=194.94.127.14 LEN=60 TOS=0x00 PREC=0x00 TTL=51 ID=58854 DF PROTO=TCP
SPT=56611 DPT=22 WINDOW=5840 RES=0x00 SYN URGP=0
Oct 11 22:53:06 fb5gwint kernel: forward Rule 19 - ACCEPT IN=eth0 OUT=eth2
SRC=213.195.77.228 DST=194.94.127.67 LEN=60 TOS=0x00 PREC=0x00 TTL=51 ID=45431 DF PROTO=TCP
SPT=40058 DPT=22 WINDOW=5840 RES=0x00 SYN URGP=0
Oct 11 22:53:06 fb5gwint kernel: forward Rule 130 - ACCEPT IN=eth0 OUT=eth4
SRC=213.195.77.228 DST=194.94.127.91 LEN=60 TOS=0x00 PREC=0x00 TTL=51 ID=41406 DF PROTO=TCP
SPT=51667 DPT=22 WINDOW=5840 RES=0x00 SYN URGP=0
Oct 11 22:53:06 fb5gwint kernel: forward Rule 13 - ACCEPT IN=eth0 OUT=eth2
SRC=213.195.77.228 DST=194.94.127.18 LEN=60 TOS=0x00 PREC=0x00 TTL=51 ID=59151 DF PROTO=TCP
SPT=53684 DPT=22 WINDOW=5840 RES=0x00 SYN URGP=0
Oct 11 22:53:06 fb5gwint kernel: forward Rule 13 - ACCEPT IN=eth0 OUT=eth2
SRC=213.195.77.228 DST=194.94.127.15 LEN=60 TOS=0x00 PREC=0x00 TTL=51 ID=4704 DF PROTO=TCP
SPT=42361 DPT=22 WINDOW=5840 RES=0x00 SYN URGP=0
Oct 11 22:53:06 fb5gwint kernel: forward Rule 13 - ACCEPT IN=eth0 OUT=eth2
SRC=213.195.77.228 DST=194.94.127.14 LEN=60 TOS=0x00 PREC=0x00 TTL=51 ID=39887 DF PROTO=TCP
SPT=56798 DPT=22 WINDOW=5840 RES=0x00 SYN URGP=0
Oct 11 22:53:06 fb5gwint kernel: forward Rule 13 - ACCEPT IN=eth0 OUT=eth2
SRC=213.195.77.228 DST=194.94.127.30 LEN=60 TOS=0x00 PREC=0x00 TTL=51 ID=9012 DF PROTO=TCP
SPT=40580 DPT=22 WINDOW=5840 RES=0x00 SYN URGP=0
Oct 11 22:53:06 fb5gwint kernel: forward Rule 13 - ACCEPT IN=eth0 OUT=eth2
SRC=213.195.77.228 DST=194.94.127.15 LEN=60 TOS=0x00 PREC=0x00 TTL=51 ID=63803 DF PROTO=TCP
SPT=42547 DPT=22 WINDOW=5840 RES=0x00 SYN URGP=0
Oct 11 22:53:07 fb5gwint kernel: forward Rule 136 – ACCEPT IN=eth0 OUT=eth4
SRC=213.195.77.228 DST=194.94.127.91 LEN=60 TOS=0x00 PREC=0x00 TTL=51 ID=42206 DF PROTO=TCP
SPT=60308 DPT=22 WINDOW=5840 RES=0x00 SYN URGP=0
```

LogData Analysis System
→ Example – threshold analysis (4/4)

- **Dictionary attack against "ssh" daemon**: parameters

- **MySQL port scan**

  - **Normal behavior**

    - User performs directed SQL requests on port 3306 to a MySQL server with a known IP address

  - **Attack situation**

    - Attacker sends lots of requests to lots of different IP addresses in a very short time interval

    - Destination IP addresses are incrementally increased

    - A complete network section is therefore scanned for MySQL servers

    - Does the attacker receive a response, he has therefore identified a MySQL server in the network

    - This information can be used for a further attack

LogData Analysis System
→ Example – threshold analysis

- **MySQL port scan:** iptables logs

```
Oct 19 06:12:55 fb5gwint kernel: forward Rule 13 - ACCEPT IN=eth0 OUT=eth2
SRC=63.97.52.4 DST=194.94.127.1 LEN=48 TOS=0x00 PREC=0x00 TTL=112 ID=32274 DF   PROTO=TCP
SPT=1703 DPT=3306 WINDOW=65535 RES=0x00 SYN URGP=0
Oct 19 06:12:55 fb5gwint kernel: forward Rule 13 - ACCEPT IN=eth0 OUT=eth2
SRC=63.97.52.4 DST=194.94.127.2 LEN=48 TOS=0x00 PREC=0x00 TTL=112 ID=32275 DF PROTO=TCP
SPT=1704 DPT=3306 WINDOW=65535 RES=0x00 SYN URGP=0
Oct 19 06:12:55 fb5gwint kernel: Internet Rule 12 - DENY IN=eth0 OUT=eth2
SRC=63.97.52.4 DST=194.94.127.3 LEN=48 TOS=0x00 PREC=0x00 TTL=113 ID=32276 DF PROTO=TCP
SPT=1705 DPT=3306 WINDOW=65535 RES=0x00 SYN URGP=0
Oct 19 06:12:55 fb5gwint kernel: forward Rule 13 - ACCEPT IN=eth0 OUT=eth2
SRC=63.97.52.4 DST=194.94.127.4 LEN=48 TOS=0x00 PREC=0x00 TTL=112 ID=32277 DF PROTO=TCP
SPT=1706 DPT=3306 WINDOW=65535 RES=0x00 SYN URGP=0
Oct 19 06:12:55 fb5gwint kernel: forward Rule 13 - ACCEPT IN=eth0 OUT=eth2
SRC=63.97.52.4 DST=194.94.127.6 LEN=48 TOS=0x00 PREC=0x00 TTL=112 ID=32279 DF PROTO=TCP
SPT=1708 DPT=3306 WINDOW=65535 RES=0x00 SYN URGP=0
Oct 19 06:12:55 fb5gwint kernel: forward Rule 13 - ACCEPT IN=eth0 OUT=eth2
SRC=63.97.52.4 DST=194.94.127.5 LEN=48 TOS=0x00 PREC=0x00 TTL=112 ID=32278 DF PROTO=TCP
SPT=1707 DPT=3306 WINDOW=65535 RES=0x00 SYN URGP=0
Oct 19 06:12:55 fb5gwint kernel: forward Rule 13 - ACCEPT IN=eth0 OUT=eth2
SRC=63.97.52.4 DST=194.94.127.7 LEN=48 TOS=0x00 PREC=0x00 TTL=112 ID=32280 DF PROTO=TCP
SPT=1709 DPT=3306 WINDOW=65535 RES=0x00 SYN URGP=0
Oct 19 06:12:55 fb5gwint kernel: forward Rule 13 - ACCEPT IN=eth0 OUT=eth2
SRC=63.97.52.4 DST=194.94.127.8 LEN=48 TOS=0x00 PREC=0x00 TTL=112 ID=32281 DF PROTO=TCP
SPT=1710 DPT=3306 WINDOW=65535 RES=0x00 SYN URGP=0
Oct 19 06:12:55 fb5gwint kernel: forward Rule 13 - ACCEPT IN=eth0 OUT=eth2
SRC=63.97.52.4 DST=194.94.127.9 LEN=48 TOS=0x00 PREC=0x00 TTL=112 ID=32288 DF PROTO=TCP
SPT=1711 DPT=3306 WINDOW=65535 RES=0x00 SYN URGP=0
Oct 19 06:12:55 fb5gwint kernel: forward Rule 13 - ACCEPT IN=eth0 OUT=eth2
SRC=63.97.52.4 DST=194.94.127.11 LEN=48 TOS=0x00 PREC=0x00 TTL=112 ID=32293 DF PROTO=TCP
SPT=1716 DPT=3306 WINDOW=65535 RES=0x00 SYN URGP=0
```

LogData Analysis System
→ Example – threshold analysis

- **MySQL port scan:** parameters (descriptors)
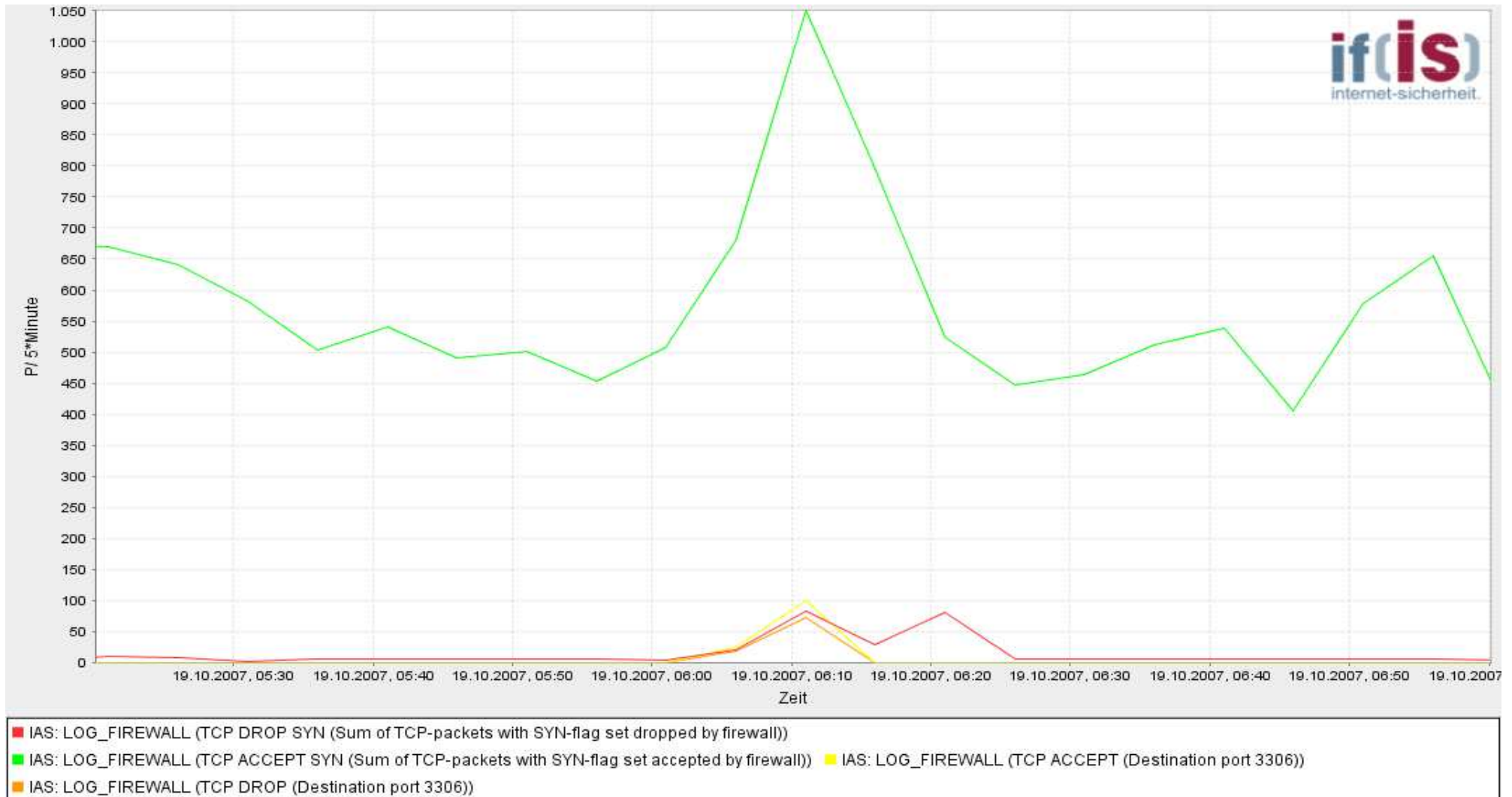
- **E-mail spam attack**

  - **Normal behavior**

    - if(is) runs a spam trap for research proposes

    - Normally no smtp communication should take place with the spam trap, therefore no communication should be recorded in the log files

  - **Attack situation**

    - smtp communication explodes in the network

    - Always the same attacker sends lots of requests on port 25 in a very short time interval, therefore he tries to deliver a lot of e-mails

    - 2 - 3 mails per second

LogData Analysis System
→ Example – threshold analysis

- **E-mail spam attack:** iptables logs

```
Oct 19 03:36:11 fb5gwint kernel: forward Rule 11 - ACCEPT IN=eth0 OUT=eth2
SRC=81.173.240.68 DST=194.94.127.38 LEN=60 TOS=0x00 PREC=0x00 TTL=55 ID=53356 DF PROTO=TCP
SPT=44742 DPT=25 WINDOW=5840 RES=0x00 SYN URGP=0
Oct 19 03:36:11 fb5gwint kernel: forward Rule 11 - ACCEPT IN=eth0 OUT=eth2
SRC=81.173.240.68 DST=194.94.127.38 LEN=60 TOS=0x00 PREC=0x00 TTL=55 ID=52314 DF PROTO=TCP
SPT=44745 DPT=25 WINDOW=5840 RES=0x00 SYN URGP=0
Oct 19 03:36:12 fb5gwint kernel: forward Rule 11 - ACCEPT IN=eth0 OUT=eth2
SRC=81.173.240.68 DST=194.94.127.38 LEN=60 TOS=0x00 PREC=0x00 TTL=55 ID=57563 DF PROTO=TCP
SPT=44747 DPT=25 WINDOW=5840 RES=0x00 SYN URGP=0
Oct 19 03:36:12 fb5gwint kernel: forward Rule 11 - ACCEPT IN=eth0 OUT=eth2
SRC=81.173.240.68 DST=194.94.127.38 LEN=60 TOS=0x00 PREC=0x00 TTL=55 ID=22408 DF PROTO=TCP
SPT=44749 DPT=25 WINDOW=5840 RES=0x00 SYN URGP=0
Oct 19 03:36:12 fb5gwint kernel: forward Rule 11 - ACCEPT IN=eth0 OUT=eth2
SRC=81.173.240.68 DST=194.94.127.38 LEN=60 TOS=0x00 PREC=0x00 TTL=55 ID=9402 DF PROTO=TCP
SPT=44752 DPT=25 WINDOW=5840 RES=0x00 SYN URGP=0
Oct 19 03:36:13 fb5gwint kernel: forward Rule 11 - ACCEPT IN=eth0 OUT=eth2
SRC=81.173.240.68 DST=194.94.127.38 LEN=60 TOS=0x00 PREC=0x00 TTL=55 ID=17306 DF PROTO=TCP
SPT=44753 DPT=25 WINDOW=5840 RES=0x00 SYN URGP=0
Oct 19 03:36:13 fb5gwint kernel: forward Rule 11 - ACCEPT IN=eth0 OUT=eth2
SRC=81.173.240.68 DST=194.94.127.38 LEN=60 TOS=0x00 PREC=0x00 TTL=55 ID=54107 DF PROTO=TCP
SPT=44756 DPT=25 WINDOW=5840 RES=0x00 SYN URGP=0
Oct 19 03:36:14 fb5gwint kernel: forward Rule 11 - ACCEPT IN=eth0 OUT=eth2
SRC=81.173.240.68 DST=194.94.127.38 LEN=60 TOS=0x00 PREC=0x00 TTL=55 ID=5660 DF PROTO=TCP
SPT=44758 DPT=25 WINDOW=5840 RES=0x00 SYN URGP=0
Oct 19 03:36:14 fb5gwint kernel: forward Rule 11 - ACCEPT IN=eth0 OUT=eth2
SRC=81.173.240.68 DST=194.94.127.38 LEN=60 TOS=0x00 PREC=0x00 TTL=55 ID=2279 DF PROTO=TCP
SPT=44760 DPT=25 WINDOW=5840 RES=0x00 SYN URGP=0
Oct 19 03:36:14 fb5gwint kernel: forward Rule 11 - ACCEPT IN=eth0 OUT=eth2
SRC=81.173.240.68 DST=194.94.127.38 LEN=60 TOS=0x00 PREC=0x00 TTL=55 ID=22672 DF PROTO=TCP
SPT=44762 DPT=25 WINDOW=5840 RES=0x00 SYN URGP=0
```

LogData Analysis System
→ Example – threshold analysis

- **E-Mail spam attack: parameters (descriptors)**
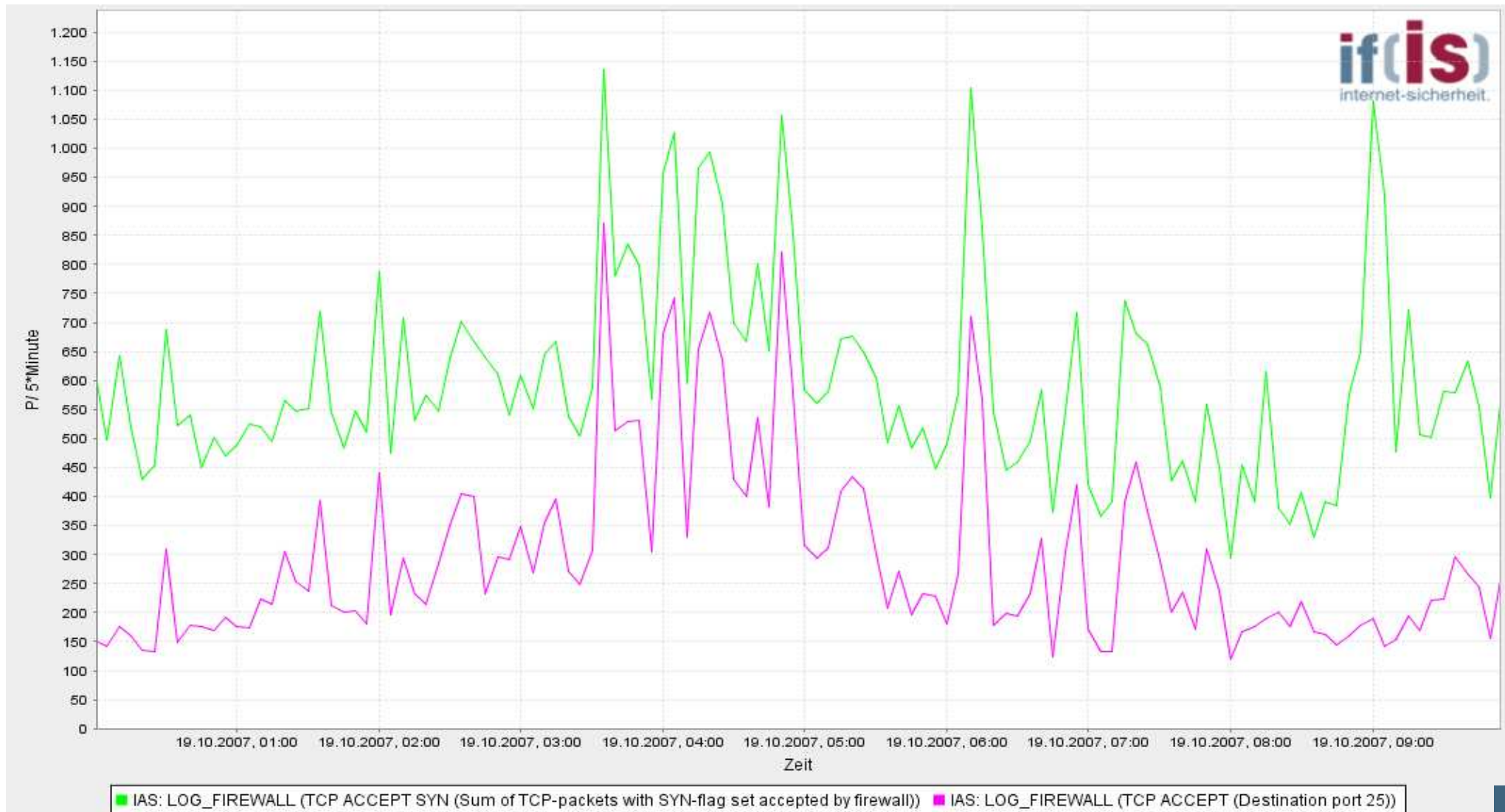
IAS: LOG_FIREWALL (TCP ACCEPT SYN (Sum of TCP-packets with SYN-flag set accepted by firewall))   IAS: LOG_FIREWALL (TCP ACCEPT (Destination port 25))

LogData Analysis System
→ Example – threshold analysis

- **Spam attack using Windows Messenger Service**

  - **Normal behavior**

    - A user can send short messages to another user by means of the Windows Messenger Service on UDP ports 1026 and 1027

    - Most of the time the user will not send more than a couple of messages per minute

  - **Attack situation**

    - Attacker sends spam randomly on ports 1026 & 1027 to many IP addresses of entire network sections

    - Packet delivery with extremely high frequency

    - Requests are often blocked by firewalls to prevent spam

LogData Analysis System
→ Example – threshold analysis

■ **Spam attack using Windows Messenger Service:** iptables logs

```
Oct 19 13:40:24 fb5gwint kernel: forward Rule 157 - DENY IN=eth0 OUT=eth4
SRC=221.209.110.13 DST=194.94.127.84 LEN=485 TOS=0x00 PREC=0x00 TTL=45 ID=0 DF PROTO=UDP
SPT=47632 DPT=1027 LEN=465
Oct 19 13:40:24 fb5gwint kernel: forward Rule 157 - DENY IN=eth0 OUT=eth4
SRC=221.209.110.13 DST=194.94.127.88 LEN=485 TOS=0x00 PREC=0x00 TTL=45 ID=0 DF PROTO=UDP
SPT=47632 DPT=1027 LEN=465
Oct 19 13:40:24 fb5gwint kernel: forward Rule 197 - DENY IN=eth0 OUT=eth0
SRC=221.209.110.13 DST=194.94.127.59 LEN=485 TOS=0x00 PREC=0x00 TTL=45 ID=0 DF PROTO=UDP
SPT=47632 DPT=1026 LEN=465
Oct 19 13:40:24 fb5gwint kernel: forward Rule 197 - DENY IN=eth0 OUT=eth0
SRC=221.209.110.13 DST=194.94.127.57 LEN=485 TOS=0x00 PREC=0x00 TTL=45 ID=0 DF PROTO=UDP
SPT=47632 DPT=1026 LEN=465
Oct 19 13:40:24 fb5gwint kernel: forward Rule 197 - DENY IN=eth0 OUT=eth0
SRC=221.209.110.13 DST=194.94.127.56 LEN=485 TOS=0x00 PREC=0x00 TTL=45 ID=0 DF PROTO=UDP
SPT=47632 DPT=1026 LEN=465
Oct 19 13:40:24 fb5gwint kernel: forward Rule 157 - DENY IN=eth0 OUT=eth4
SRC=221.209.110.13 DST=194.94.127.86 LEN=485 TOS=0x00 PREC=0x00 TTL=45 ID=0 DF PROTO=UDP
SPT=47632 DPT=1027 LEN=465
Oct 19 13:40:24 fb5gwint kernel: forward Rule 157 - DENY IN=eth0 OUT=eth4
SRC=221.209.110.13 DST=194.94.127.87 LEN=485 TOS=0x00 PREC=0x00 TTL=45 ID=0 DF PROTO=UDP
SPT=47632 DPT=1026 LEN=465
Oct 19 13:40:24 fb5gwint kernel: forward Rule 157 - DENY IN=eth0 OUT=eth4
SRC=221.209.110.13 DST=194.94.127.86 LEN=485 TOS=0x00 PREC=0x00 TTL=45 ID=0 DF PROTO=UDP
SPT=47632 DPT=1026 LEN=465
Oct 19 13:40:24 fb5gwint kernel: forward Rule 197 - DENY IN=eth0 OUT=eth0
SRC=221.209.110.13 DST=194.94.127.123 LEN=485 TOS=0x00 PREC=0x00 TTL=45 ID=0 DF PROTO=UDP
SPT=47632 DPT=1026 LEN=465
Oct 19 13:40:24 fb5gwint kernel: forward Rule 197 - DENY IN=eth0 OUT=eth0
SRC=221.209.110.13 DST=194.94.127.120 LEN=485 TOS=0x00 PREC=0x00 TTL=45 ID=0 DF PROTO=UDP
SPT=47632 DPT=1026 LEN=465
```

LogData Analysis System
→ Example – threshold analysis

- **Spam attack using Windows Messenger Service:** parameters (descriptors)
  - wms over UDP ports 1026/1027
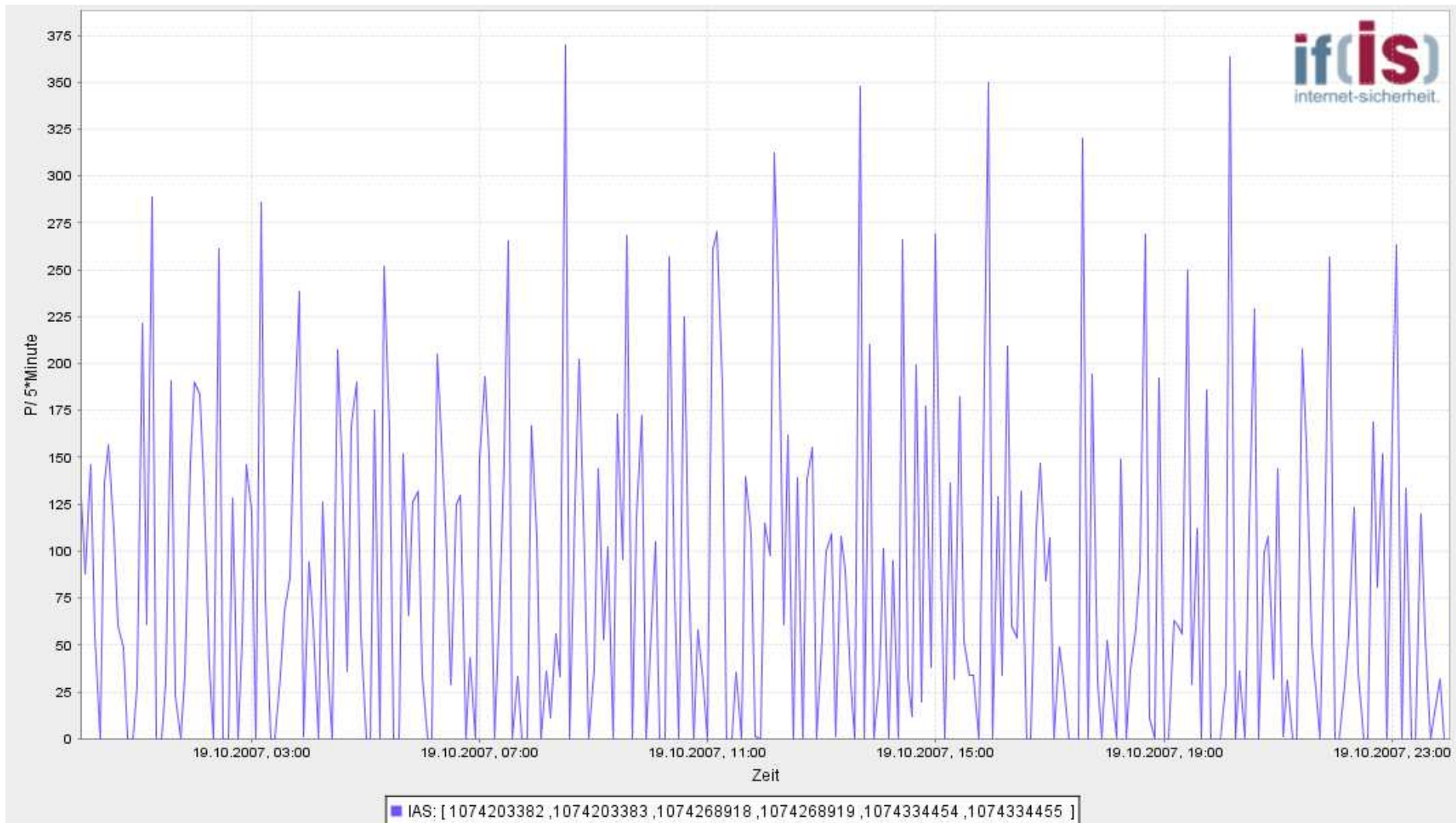
LogData Analysis System
→ Example − detection of patterns

- **SYN/FIN port scan** towards the SMTP port of the computer department's mail server

  - SYN/FIN flag combination not defined by the RFC 793

  - Older implementations of firewalls do not filter these packets

    - Is the port of the destination system behind the firewall closed, it will answer with RST/ACK

    - Is the port open, it will answer with SYN/ACK

  - Attacker can evade the firewall to detect open ports on destination systems

```
Oct 10 04:20:21 fb5gwint info kern kernel: forward Rule 157 - DENY
IN=eth0 OUT=eth4 SRC=12.158.171.206 DST=194.94.127.84
LEN=1500 TOS=0x00 PREC=0x00 TTL=114 ID=5859 DF PROTO=TCP
SPT=3826 DPT=25 WINDOW=64011 RES=0x00 ECE URG RST SYN FIN URGP=0
```

LogData Analysis System
→ Example – detection of patterns

- **DNS Port Scan**

    - DNS uses UDP for transportation

    - The header of the UDP packet has a length of 8 byte

    - If a UDP packet with a length of 8 byte is recorded the packet has no additional payload.

    - An "empty" packet is a clear indication for a port scan, in this case for DNS (port 53)

```
Mai 23 20:53:03 fb5gwint info kern kernel: forward Rule 13 - ACCEPT IN=eth0
OUT=eth2 SRC=65.36.167.120 DST=194.94.127.23 LEN=28 TOS=0x00 PREC=0x00
TTL=115 ID=9066 PROTO=UDP SPT=2531 DPT=53 LEN=8
```

- **Win Nuke**

  - Is a logical DoS attack

  - Exploits a vulnerability of older windows systems (<W2k)

  - A tcp packet on port 139 (NetBIOS) with a set urgent pointer flag results in a blue screen and therefore with a system crash when received

```
Mar 19 16:26:27 fb5gwint info kern kernel: forward Rule 11 - ACCEPT IN=eth0
OUT=eth2 SRC=122.146.50.29 DST=194.94.127.97 LEN=48 TOS=0x00 PREC=0x00
TTL=115 ID=9066 PROTO=TCP SPT=2531 DPT=139 WINDOW=65535 RES=0x00 SYN URG
URGP=3168
```

- **Land Attack**

  - Exploits vulnerabilities within the tcp/ip stack of different operating systems

  - Attacker sends a tcp packet with a set SYN flag, with identical source and destination addresses

  - An unpatched system generates a packet with the set SYN/ACK flag combination, addressing it to itself

  - Due to a faulty implementation the SYN/ACK flag combination is interpreted as a SYN flag, resulting in an infinite loop sending requests and responses to itself

  - This attack seemed to be irrelevant recently until the same vulnerability was reintroduced with Windows XP SP2 and Windows 2003

```
Dec 17 13:47:37 fb5gwint info kern kernel: forward Rule 11 - ACCEPT IN=eth0
OUT=eth2 SRC=194.94.127.84 DST=194.94.127.84 LEN=48 TOS=0x00 PREC=0x00
TTL=115 ID=9066 PROTO=TCP SPT=3412 DPT=6712 WINDOW=65535 RES=0x00 SYN
```

LogData Analysis System
→ Example – correlation

- Extraction from `/var/log/deamon.log`

  ```
  Oct 09 17:47:03 host in.ftpd[16273]: connect from 202.10.30.49
  ```

- Extraction from `/var/log/auth.log`

  ```
  Oct 09 17:47:08 host PAM_unix[16273]: check pass; user unknown
  Oct 09 17:47:08 host PAM_unix[16273]: authentication failure; (uid=0)

    -> **unknown** for ftp
                                              service
  Oct 09 17:47:13 host PAM_unix[16273]: check pass; user unknown
  Oct 09 17:47:13 host PAM_unix[16273]: authentication failure; (uid=0)

    -> **unknown** for ftp
                                              service
  ```
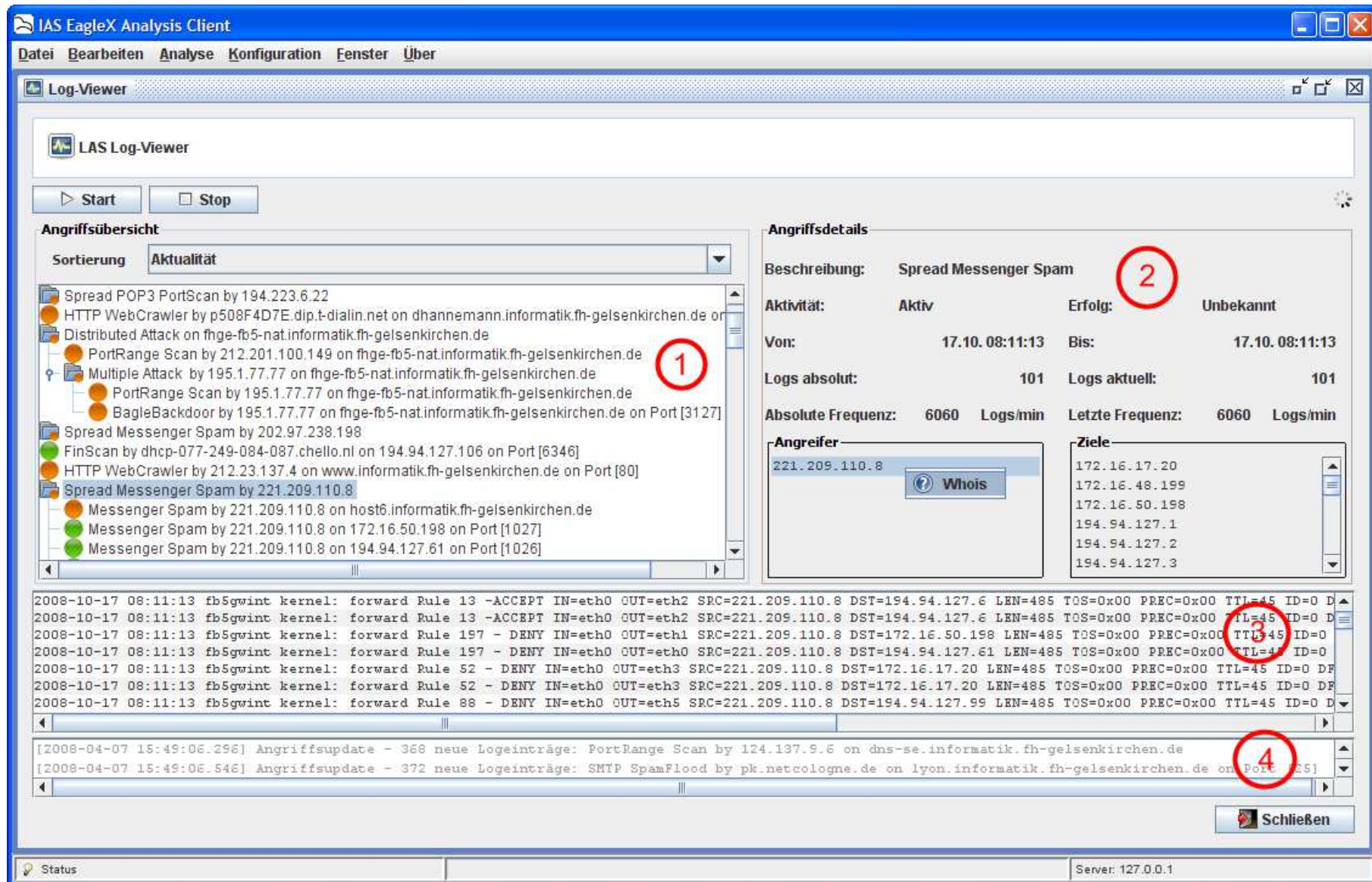
- By the use of the process ID, which is identical in both logs, and by the chronological reference (timestamp) a correlation between those log entities can be established.

- By doing this, the attacker's IP address can be determined

LogData Analysis System
→ User Interface (1/3)

© Prof. Norbert Pohlmann, Institute for Internet Security - if(is), University of Applied Sciences Gelsenkirchen, Germany

LogData Analysis System
→ User Interface (2/3)

(1) **attack overview**

- Displays the current attack situation using a tree structure
- Correlation of the events of different modules used for analysis
- Color coding depending on the outcome of the attack

(2) **Detailed view**

- Displays the details of an attack, which has been marked in the attack overview
  - IP addresses of all attackers and the destinations of the attacks
  - Occurrence of log data entries as well as the frequency

(3) **Logs**

- Original log data, which have resulted in the detection of the attack

(4) **Message Box**

- Displays status information on attack updates

LogData Analysis System
→ User Interface (3/3)

- Using the right mouse button within the detailed view a whois query can be performed for each IP address

- Allows fast overview on the source of the attack

- The example show the result of a query on a attack, which was performed by a machine positioned in China

© Prof. Norbert Pohlmann, Institute for Internet Security - if(is), University of Applied Sciences Gelsenkirchen, Germany

Logdaten-Analyse-System
→ Results (1/2)

- **March 2008**

LAS detected attacks
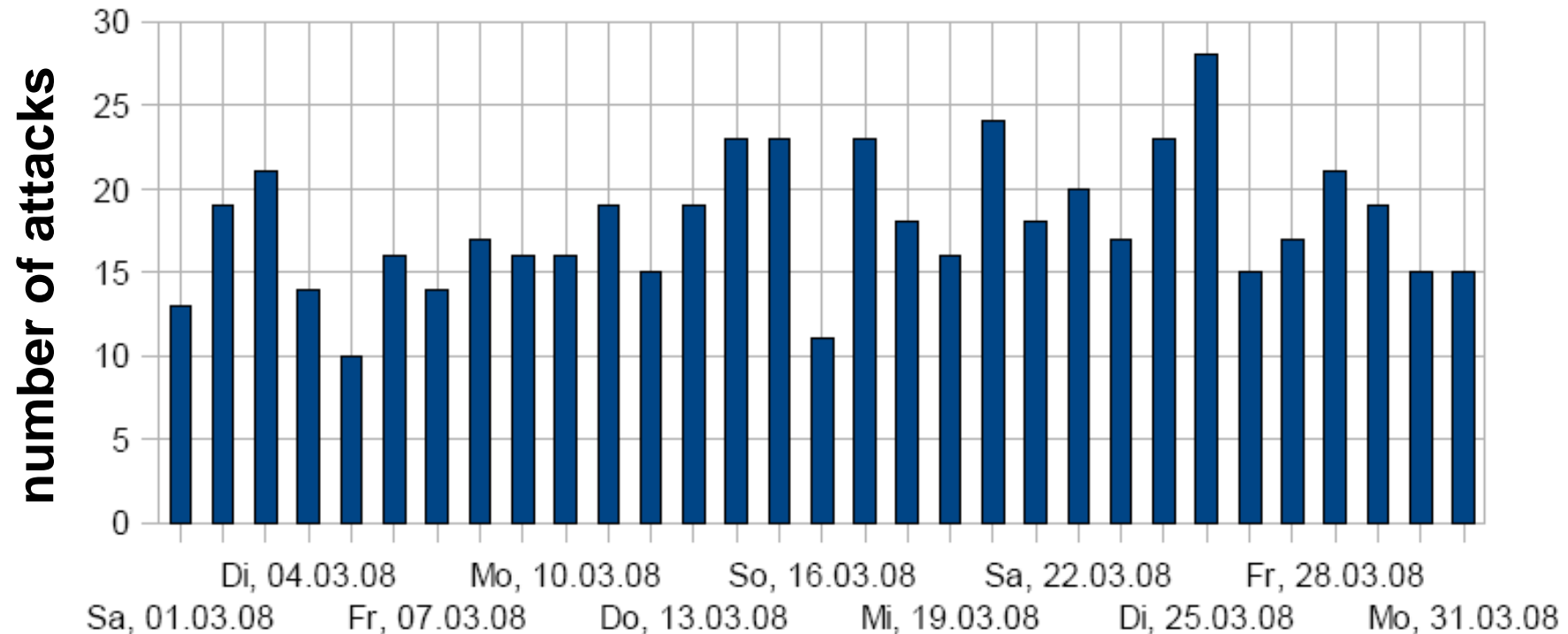
- **555 attacks were detected in the observation period**

- **In average 18 per Day**

- **8 different attack types a day**

- **56% of all attacks were scans**

Logdaten-Analyse-System
→ Results (2/2)

- **March 2008**

  - No DoS attack

  - SSH attacks generated the biggest amount of log data

  - Very low rate of false positive < 5%

| Angriff	Absolut	Relativ
DNS Scan	249	44,8%
HTTP/HTTPS Angriff	65	11,7%
SSH Brute Force	59	10,6%
Messenger Spam	29	5,2%
Mail Scan (SMTP, POP3, IMAP)	26	4,7%
SMTP-Spam flood	23	4,1%
DB Scan (MsSQL, MySQL, Oracle,...)	12	2,2%
Port Range Scan	9	1,6%
Sonstige	83	15,0%
Summe	555	100,0%

Content

- **Aim and outcomes of this lecture**

- **Classification**

- **LogData Analysis System**

- **Examples**

- **Data flow and data management**

- **Summary**

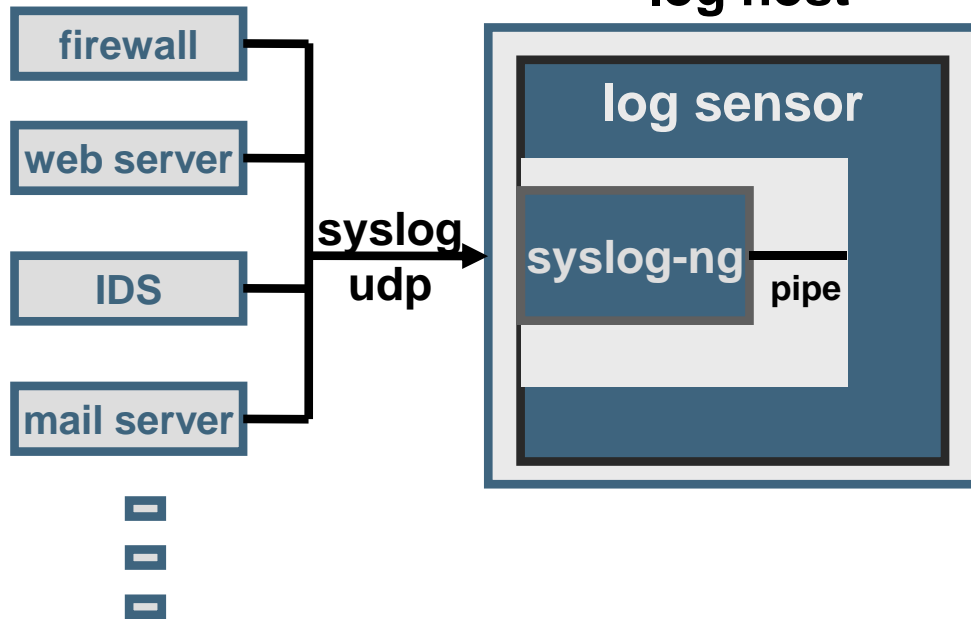# LogData Analysis System
→ Data flow & data management (1/7)

log sources

firewall

web server

IDS

mail server

syslog udp
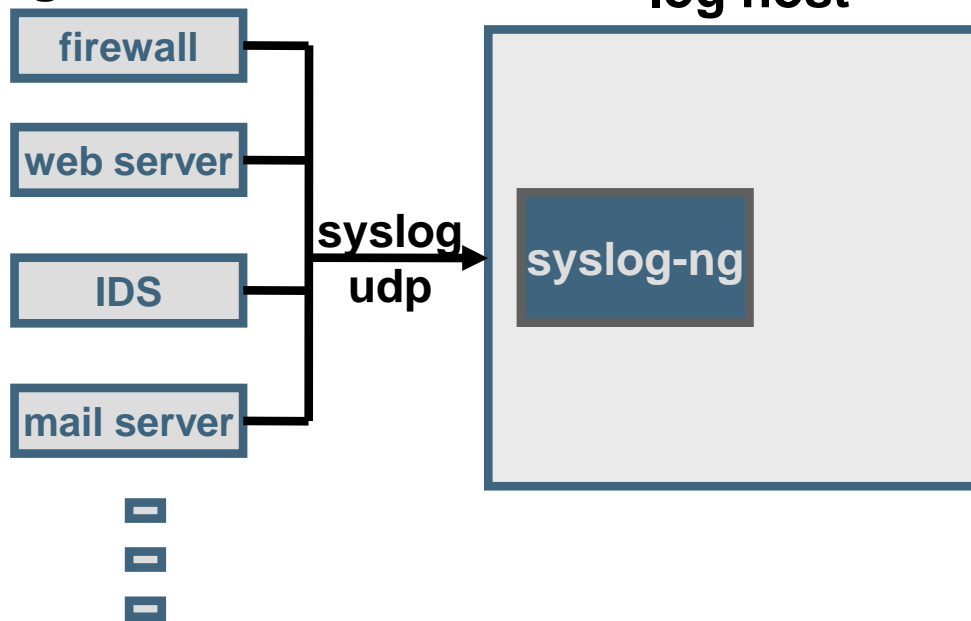
log host

syslog-ng

- Log data is transferred to the log host using syslog/udp

- syslog-ng daemon receives the log data flow

- If one of the log sources has syslog-ng as well, the transmission can also be performed encrypted using a TCP connection.

LogData Analysis System
→ Data flow & data management (2/7)

log sources

log host

```
firewall
web server          log sensor
                    ┌──────────────┐
IDS    syslog  ──→  │ syslog-ng    │
       udp          │         pipe │
mail server         └──────────────┘
```
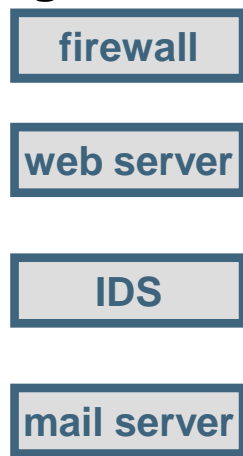
- syslog-ng writes the received log data into a pipe.

- The log sensor reads out the pipe and normalizes the data.

- If necessary, syslog can write the data in files or DBs using batch programs. But it is **not necessary, to make the data persistent**.
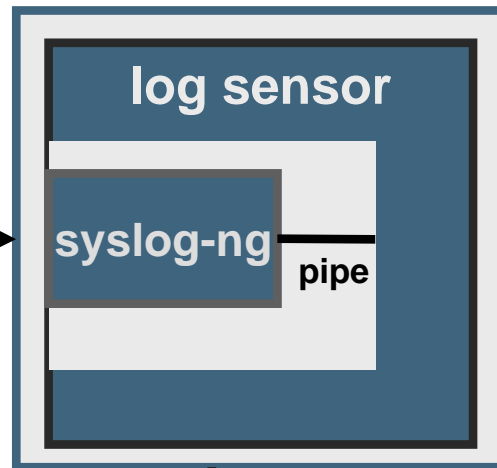
log sources

log host

```
firewall
web server
IDS
mail server
```

syslog udp

log sensor

syslog-ng **pipe**

save parameters (descriptors)

transfer system → **IAS-DB**
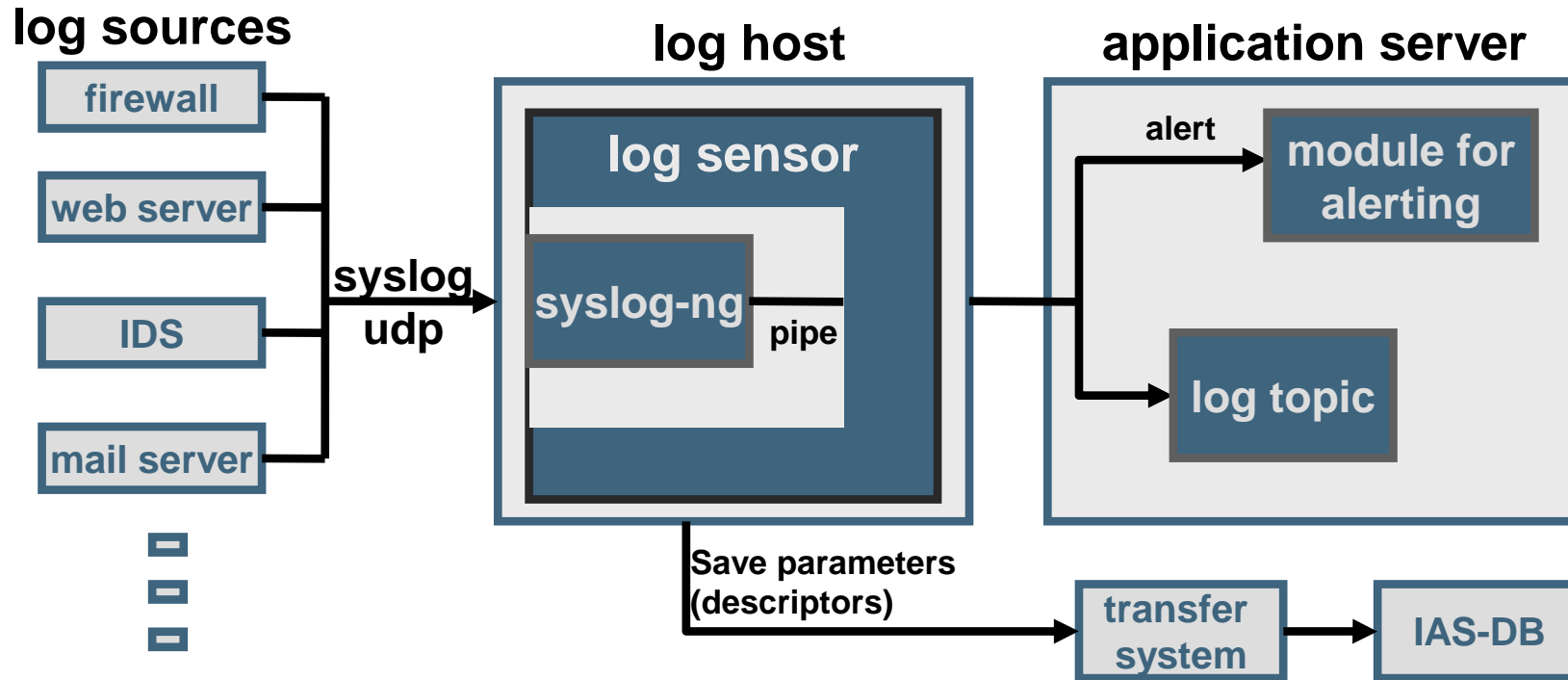
- The parameters (descriptors) are counted within the log sensor.

- The anonymized statistical data (descriptors) are send to the transfer system and stored tin the IAS-DB.

- The normalized log data is analyzed in the realtime analysis module of the log sensor to detect attacks.

- The detected attacks are sent to the centralized log topic module of the application server.

- For detected attacks, which have also been successful, an alert is generated within the alerting module of the IAS.
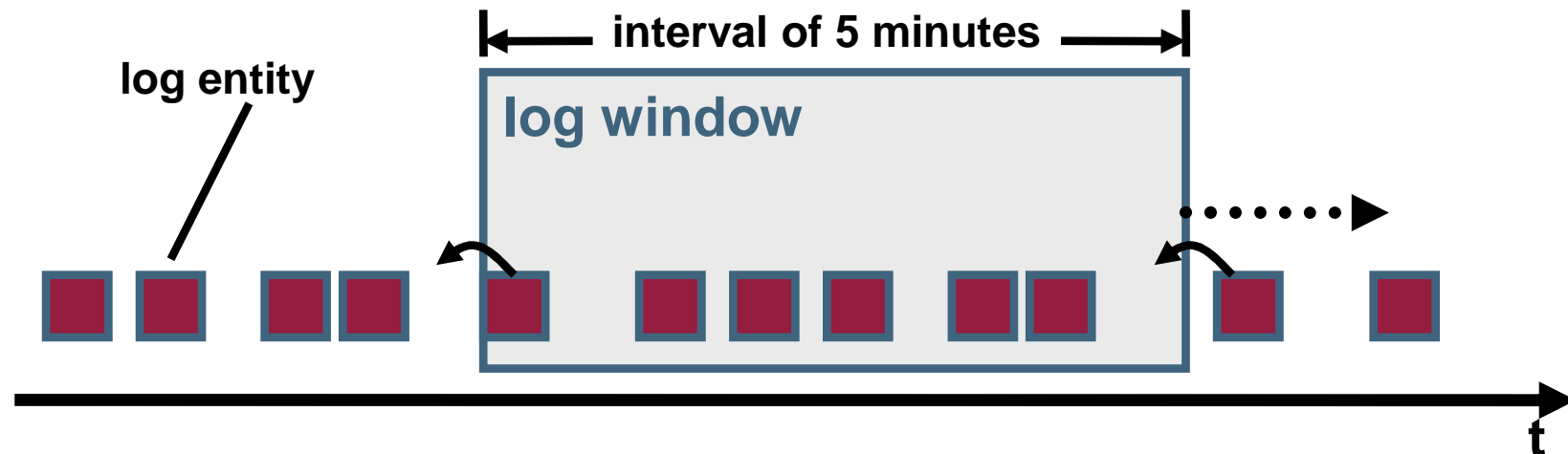
LogData Analysis System
→ Data flow & data management (5/7)

interval of 5 minutes

log entity

log window

t
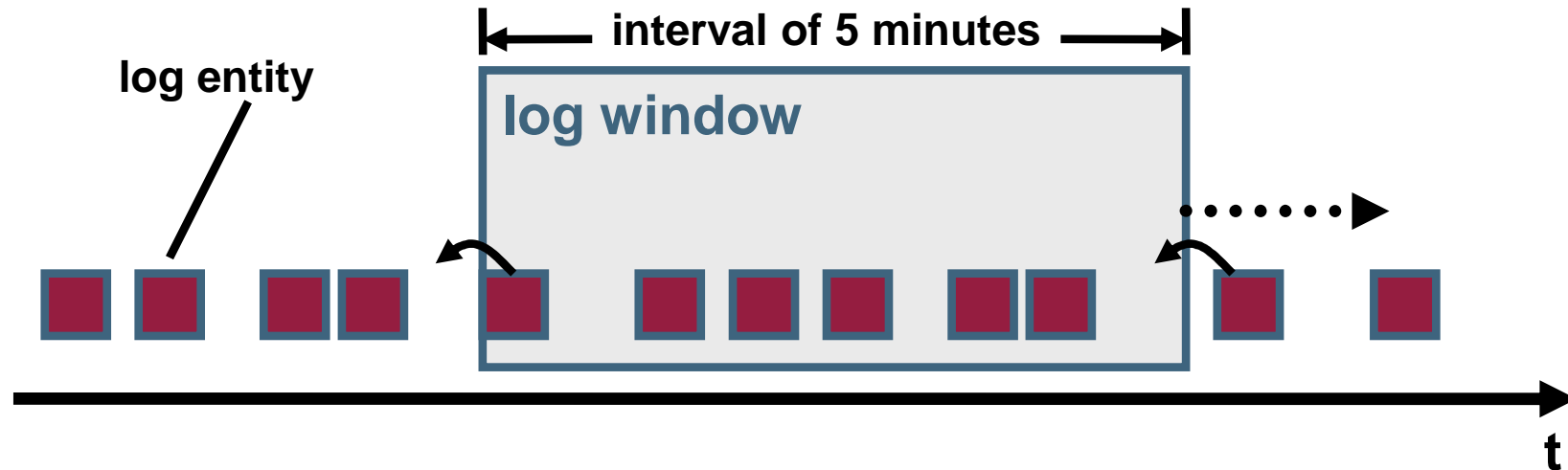
- Within the realtime analysis module all log data of one type is added to one log window.

- Log data, which is older than 5 minutes, drops out of the log window and is deleted from the memory.

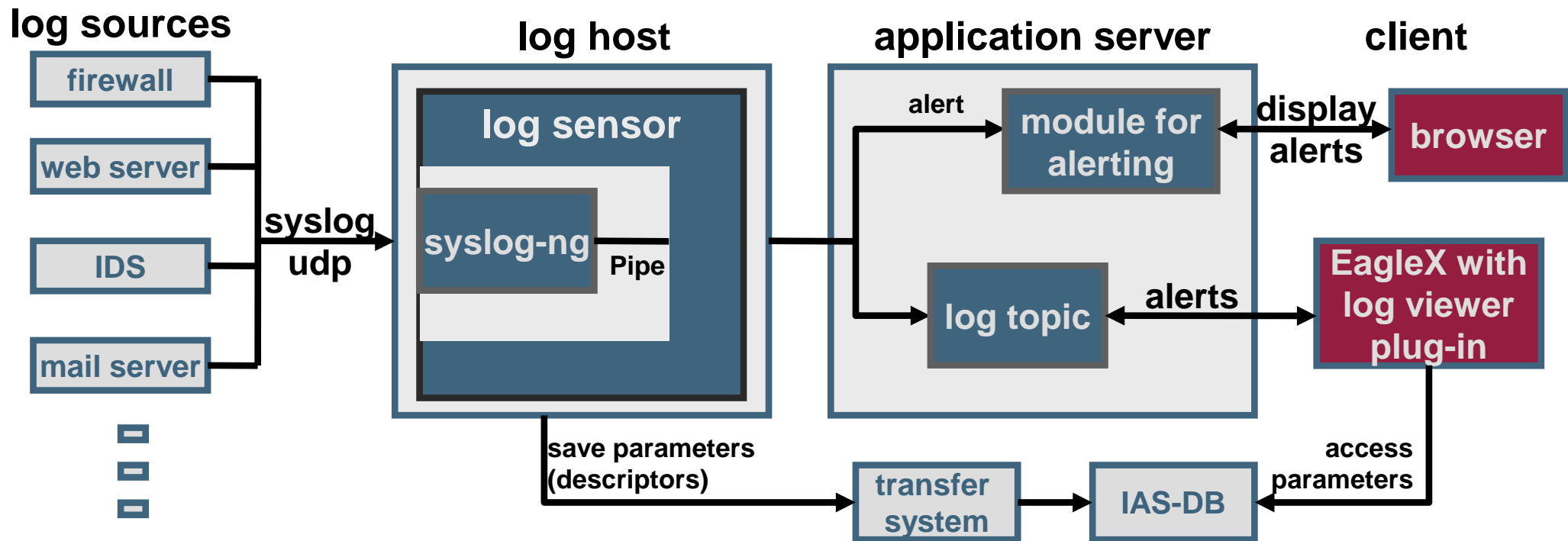- If the count of log entities exceeds the threshold, a more specific analysis of the log data in this log window is performed **(threshold analysis).**

- If an attack is detected, an alert is generated.

- All available log data, which matches the attack pattern, is visualized and made accessible.

- Alerts and the related log data, coming from the log topic module, can be accessed and visualized by the EagleX Client with a log viewer plug-in.

- If necessary the data belonging to an attack can be made persistent manually from within the plug-in.

- It is also a possible scenario, to create a pseudonymized version of the sensitive information in the log data depending on individual access rights and then to display this version.

Content

- **Aim and outcomes of this lecture**

- **Classification**

- **LogData Analysis System**

- **Examples**

- **Data flow and data management**

- ## Summary

LogData Analysis System (LAS)
→ Cons

- Log data can be extremely comprehensive

  - Very unclear (take a look at the examples)

  - Beneficial information is hidden / implicit

  - Very small amount of security relevant information < 5%

- There is no standardized format for logs

  - Logs of different applications can be different, even if they describe the same incidents

  - Makes the automated processing and interpretation more difficult

- Formats for log data are often badly or almost not documented

- The analysis of log data requires an exact understanding of the applications and technical processes, from which the data is gathered (**expert knowledge**).

- NIDS generate an alert as soon as an attack has been detected by the use of the data collected right from the wire

    - No information about the attack chain or the results, since the attack has not been performed by the time of the detection.

- Log data can describe,

    - if an attack has been successful

    - what kind of actions the attacker has performed on the targeted system

        - How a system has reacted on the sent data and how the data was interpreted

        - Allows the attack chain to be reconstructed

        - forensics and legal prosecution of attackers

- Correlation of the log data allows the detection of spread attacks, which are performed in parallel to multiple systems (**Spread Attacks**).

LogData Analysis System (LAS)
→ Capabilities

- Depending on a policy, which defines the incidents that should be logged, the LogData Analysis System can be optimized for the specific environment it is operating in.

LogData Analysis System (LAS)
→ Summary

- A log data early warning systems will help to detect attacks.

- Log data helps to reconstruct the attack chain and can be used for forensics.

LogData Analysis System (LAS)
→ References

- [1] Abe Singer and Tina Bird, Building a Logging Infrastructure, 2004

- [2] Kevin J. Schmidt, Threat analysis using log data, 2007

- [3] Dario Valentino Forte, The "Art" of Log Correlation

- [4] Risto Vaarandi, Tools and Techniques for Event Log Analysis, 2005

Links:

Institute for Internet Security:
http://www.internet-sicherheit.de/forschung/aktuelle-projekte/internet-frhwarnsysteme/

Home of syslog-ng

http://www.balabit.com/network-security/syslog-ng/