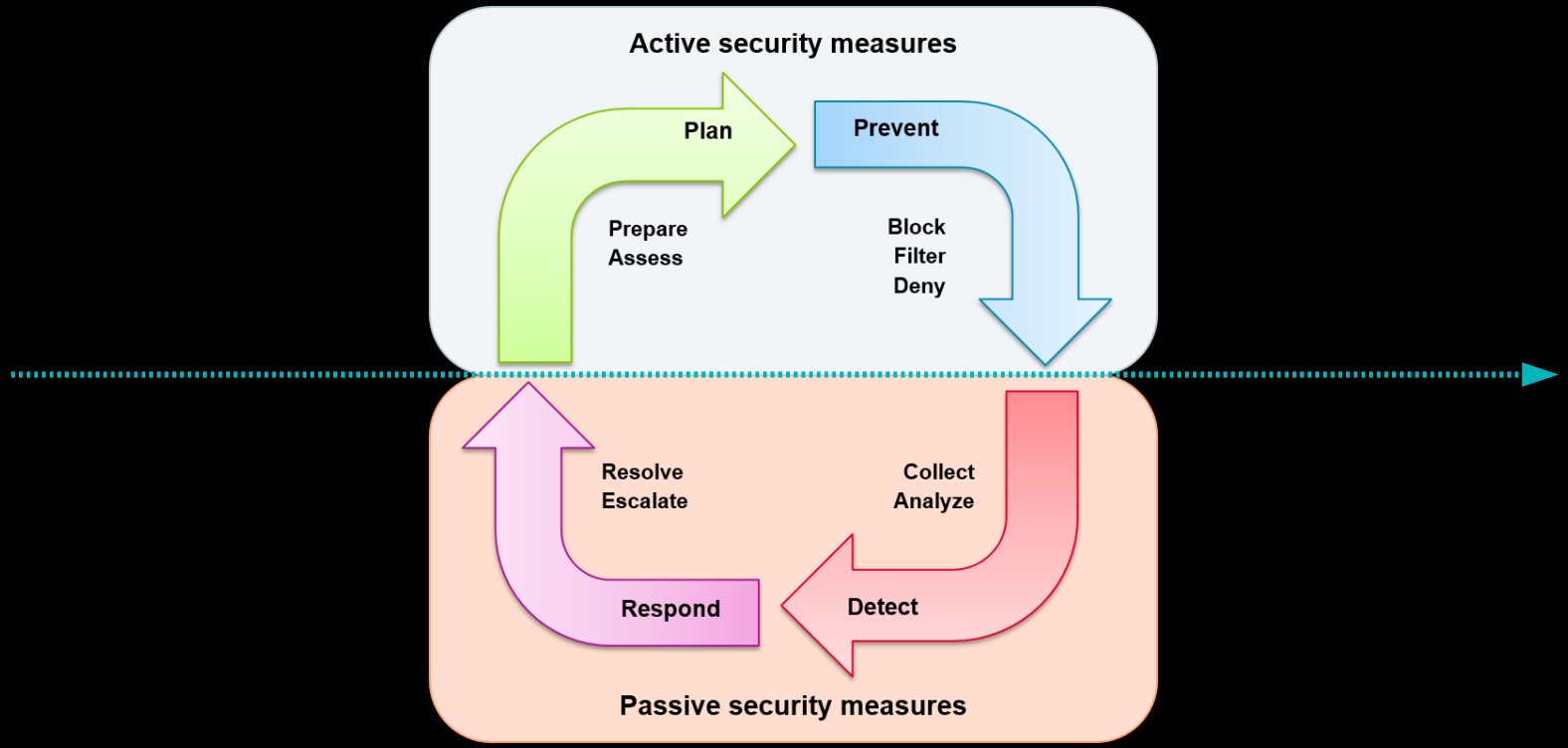
Secure logging with `syslog-ng`

Forward integrity and confidentiality of system logs

Stephan Marwedel
FOSDEM 2020
Security Devroom

AIRBUS

The security cycle

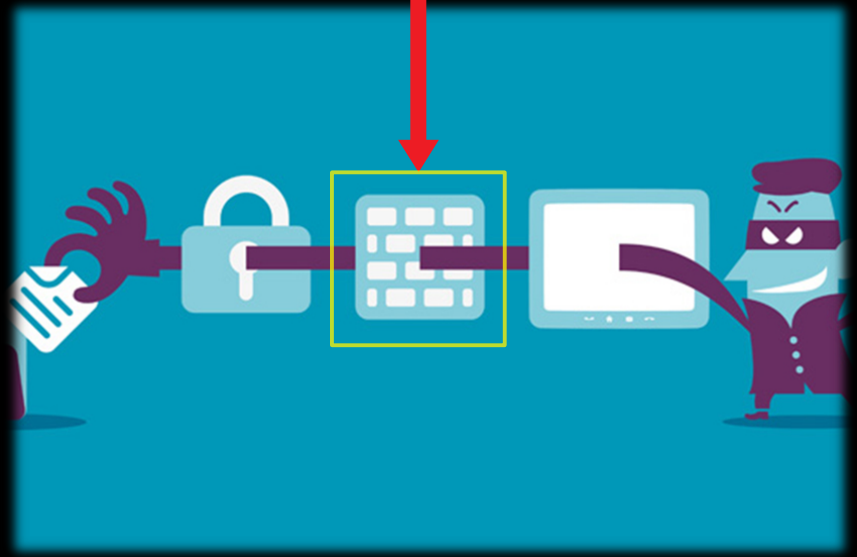


Security monitoring objective

Make the attacker visible

Instrument the system

Perform continuous log analysis

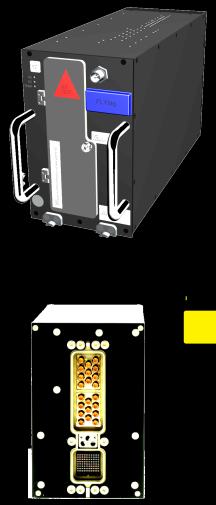


Secure logging threat model

- Successful compromise of log host
- Full control over log device
- Hide traces
  - Add log entries
  - Remove log entries
  - Edit log entries

System log integrity principle

System log host



System log file

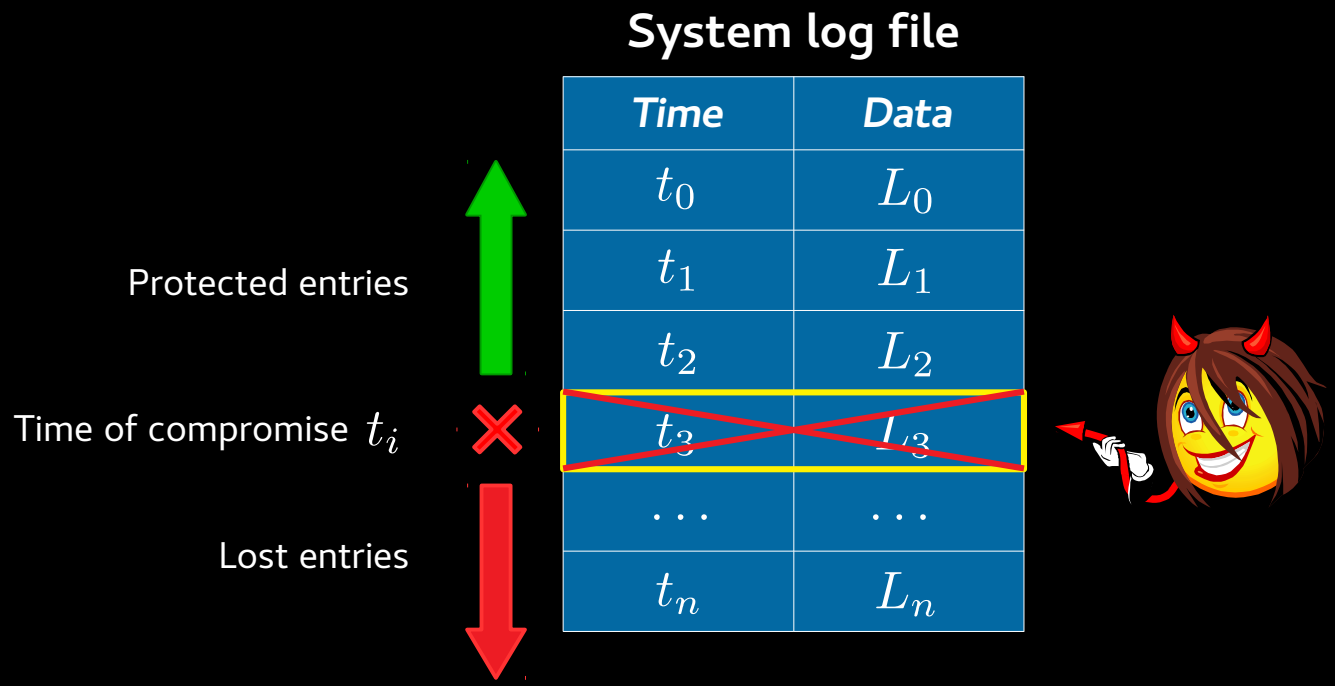
Time	*Data*
t_0	L_0
t_1	L_1
t_2	L_2
~~t_3~~	~~L_3~~
$\dots$	$\dots$
t_n	L_n

System log file

Time	*Data*
t_0	L_0
t_1	L_1
t_2	L_2
t_3	L'_3
$\dots$	$\dots$
t_n	L_n

A verifier will detect that L'_3 has been tampered with

Forward integrity principle



- Compromise at t_i means no integrity guarantee for $t_n \geq t_i$ with $n \geq i$
- Log entries are still integrity protected for $t_n < t_i$ with $n < i$

Forward integrity algorithm

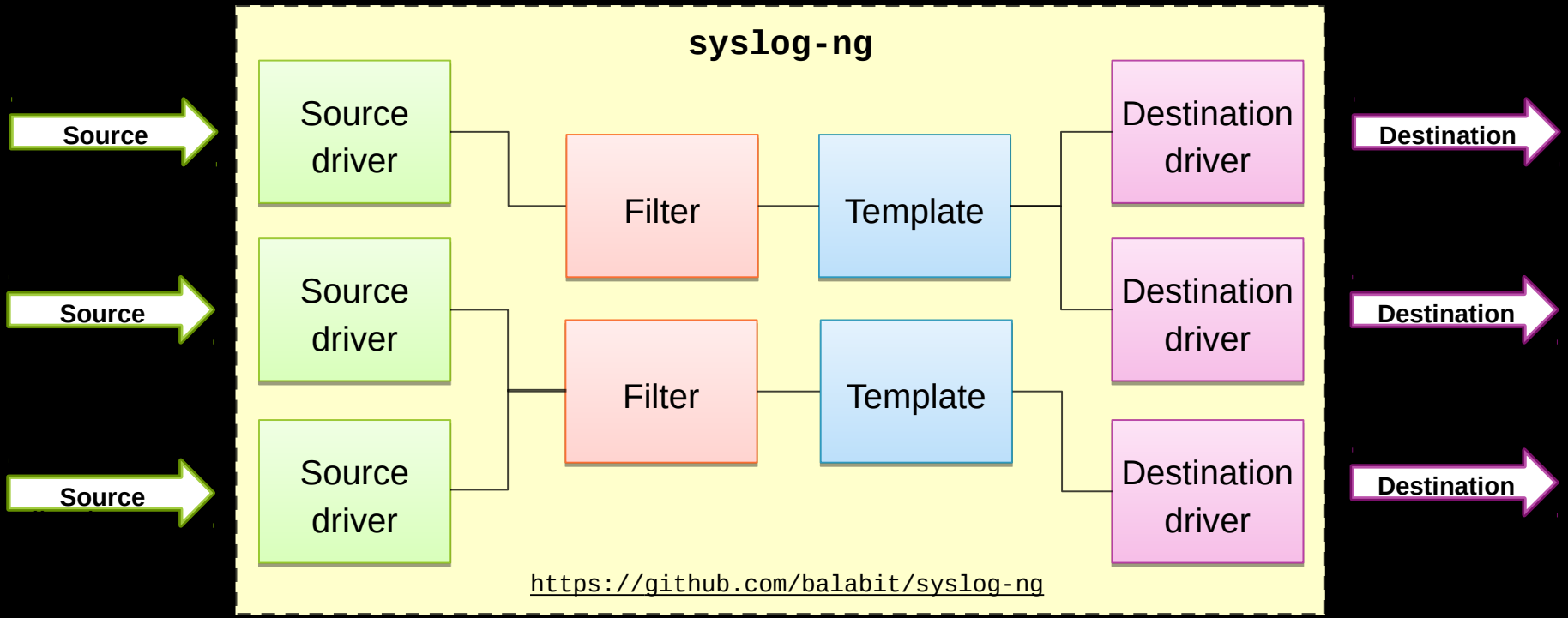
- Share key K_0 and compute $K_i := evolve(K_{i-1})$
- Compute individual integrity tags per log entry
- Compute aggregated integrity tag for the whole log file:
  $AggMAC_i := HMAC_{K_i}(AggMAC_{i-1}, HMAC_{K_i}(L_i))$
- Delete previous K_{i-1} and $AggMAC_{i-1}$
- At time of compromise t_i the attacker has access to K_i but not to K_{i-1}
- The integrity tag $AggMAC$ protects the whole log file

Integrity protected system log file

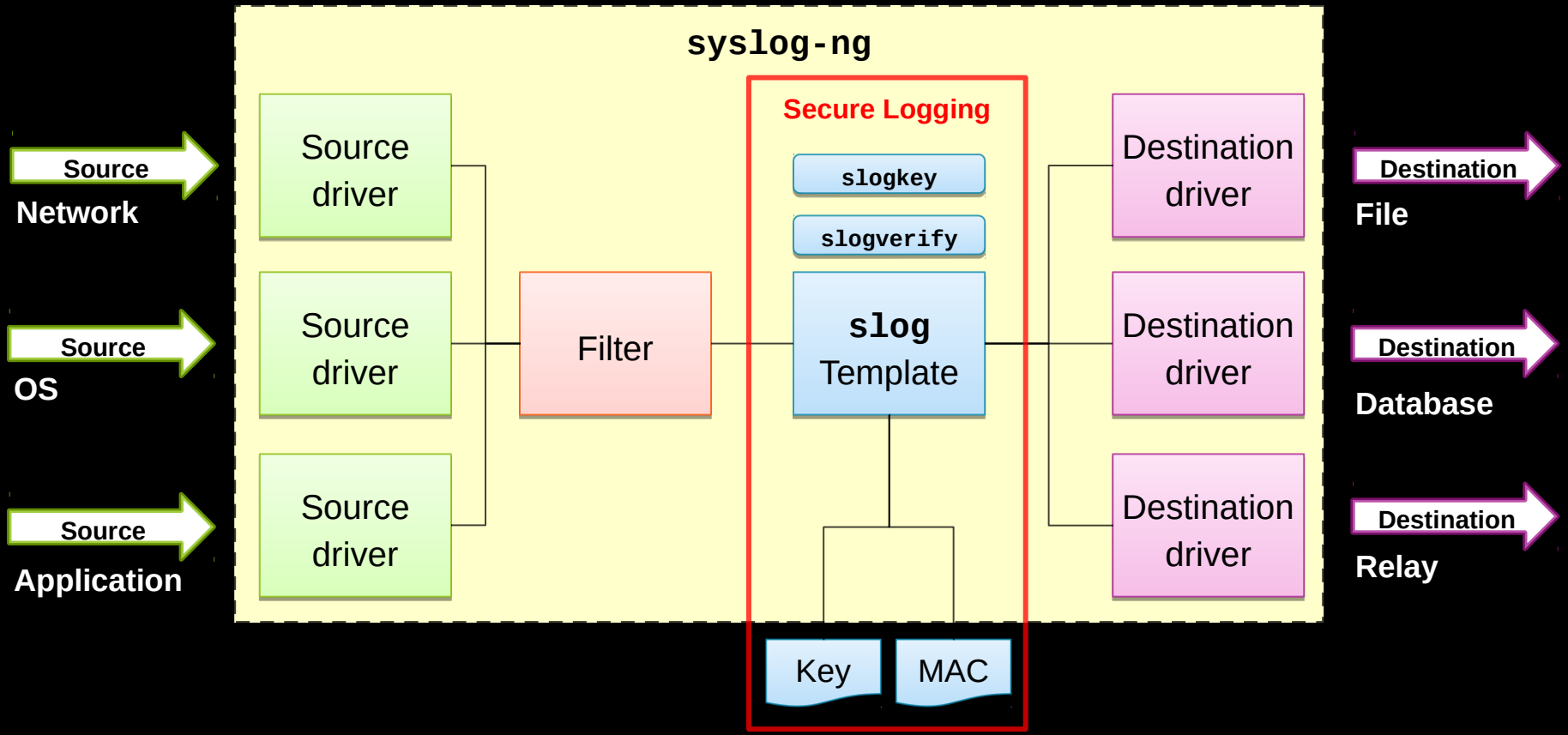
Time	*Data*	*Integrity tag*
t_0	L_0	$HMAC_{K_0}(L_0)$
t_1	L_1	$HMAC_{K_1}(L_1)$
t_2	L_2	$HMAC_{K_2}(L_2)$
t_3	L_3	$HMAC_{K_3}(L_3)$
$\dots$	$\dots$	$\dots$
t_n	L_n	$HMAC_{K_n}(L_n)$

$AggMAC_i := HMAC_{K_i}(AggMAC_{i-1}, HMAC_{K_i}(L_i))$

syslog-ng overview



Secure logging implementation



AIRBUS

Secure logging example

Original input at source

```
Dies ist eine Log Nachricht
Und dies auch
Hier kommt mal eine laengere Nachricht
```

Log messages

```
OFMBAAAAAAA=:LouI2vSfIJAuq17CjQdBeqh1YdgvwqFY9RyxTcQk2u0yc+Tqfm14OmOdU+LpC+alJMnPn3aT/A==
OVMBAAAAAAA=:UWEhUdN2d+iADsPtBFKVGBNB+nGRnm/D03m23/OMJ/jpdpXd6SQ5cb4=
OlMBAAAAAAA=:4r5Hw8kyXytlkF5z/nIWwdm8J4XOylKxBY572tlqOINg0vjAVDbOoo1mjsh4LHswEqW/xCJSbiu96QFFXqFyqaxc
```

Output of successful log verification

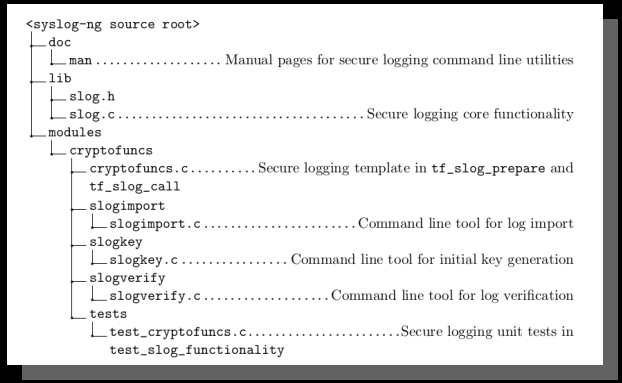
```
0000000000000000: Dies ist eine Log Nachricht
0000000000000001: Und dies auch
0000000000000002: Hier kommt mal eine laengere Nachricht
```

Example `syslog-ng.conf`

```
source s_network {
  network(
    transport ("udp")
    port(514)

    # NOTE : Secure logging requires this flag to be set
    flags(store-raw-message)
  );
);


# Secure logging template with key and MAC file locations
template t_slog {
  template("$(slog -k /var/slog/host.key -m /var/slog/mac.dat $RAWMSG)\n");
};

# Destination that uses the secure logging template
destination d_local {
  file("/var/log/messages.slog" template(t_slog));
};

log {
  source(s_network);
  destination(d_local);
};
```

Implementation and performance

- 6 new source files to `syslog-ng`
- No new dependencies were introduced
- All cryptographic operations rely on OpenSSL
- Excellent performance when using AES-NI
  - Intel Core i7 6th Gen @ 2.2GHz 9000 log entries/s
  - Typical log host with $2 \cdot 10^5$ entries in 24 hours
  - $7.3 \cdot 10^7$ log entries during 1 year of operation
  - Key derivation in < 1s



AIRBUS

Challenges

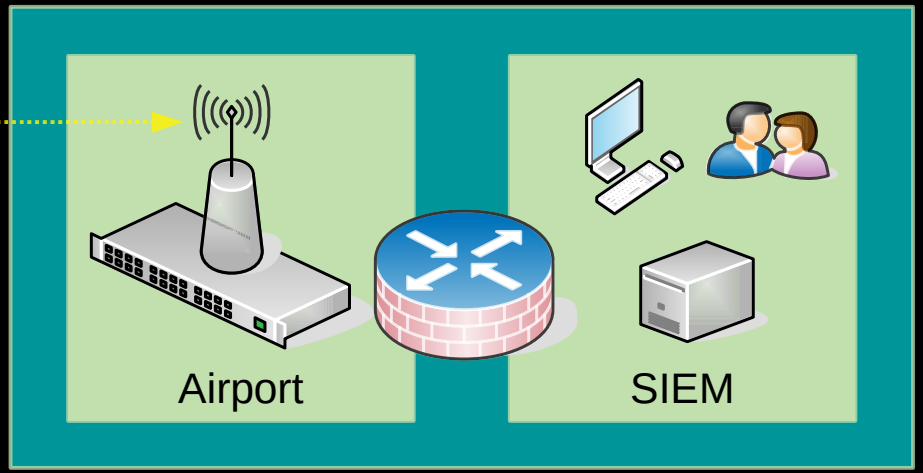
- Log system behavior under load
- `syslog-ng` internal API poorly documented
- No `syslog-ng` developers guide available
- Complex build system
- Packaging for target platform must be performed manually
- No log rotation

Example scenario

Airborne segment

- Key derivation
- Log record creation

Ground segment



- Log record relay
- Log record analysis

Summary

Achievements

- Tamper evident secure log system with easy integration into existing `syslog-ng` installations
- Performance on log host superior to `systemd` forward secure sealing
- Efficient offline log file verification
- Log verification can be integrated into existing SIEM solution
- Industrial readiness

Future work

- Crash recovery: Restore log entries that might have been lost during a system crash

Fragen?
Questions?
Perguntas?
Frågor?
שאלות?

Stephan Marwedel
Product Security Engineer

Airbus Engineering – Aircraft Security
Kreetslag 10, 21129 Hamburg – Germany
E-Mail: stephan.marwedel@airbus.com
Phone: +4940-743-85635