INTRODUCING OSSEC

host-based IDS

Saturday 21st November, 2015

Theresa Meiksner

BSidesVienna 0x7DF (2015)

Overview

1. What is OSSEC?
2. Architectural overview
3. Why do we need log analysis?
4. How to detect a rootkit with OSSEC?
5. ELK Stack Integration
6. Live-DEMO

whoami

- SysAdmin@s-itsolutions
- tm@aremai.net
- `http://www.aremai.net`
- `http://github.com/aremai`
- hellslide@jabber.ccc.de

What is OSSEC?

What is OSSEC?

OSSEC is a open-source host-based intrusion detection system.

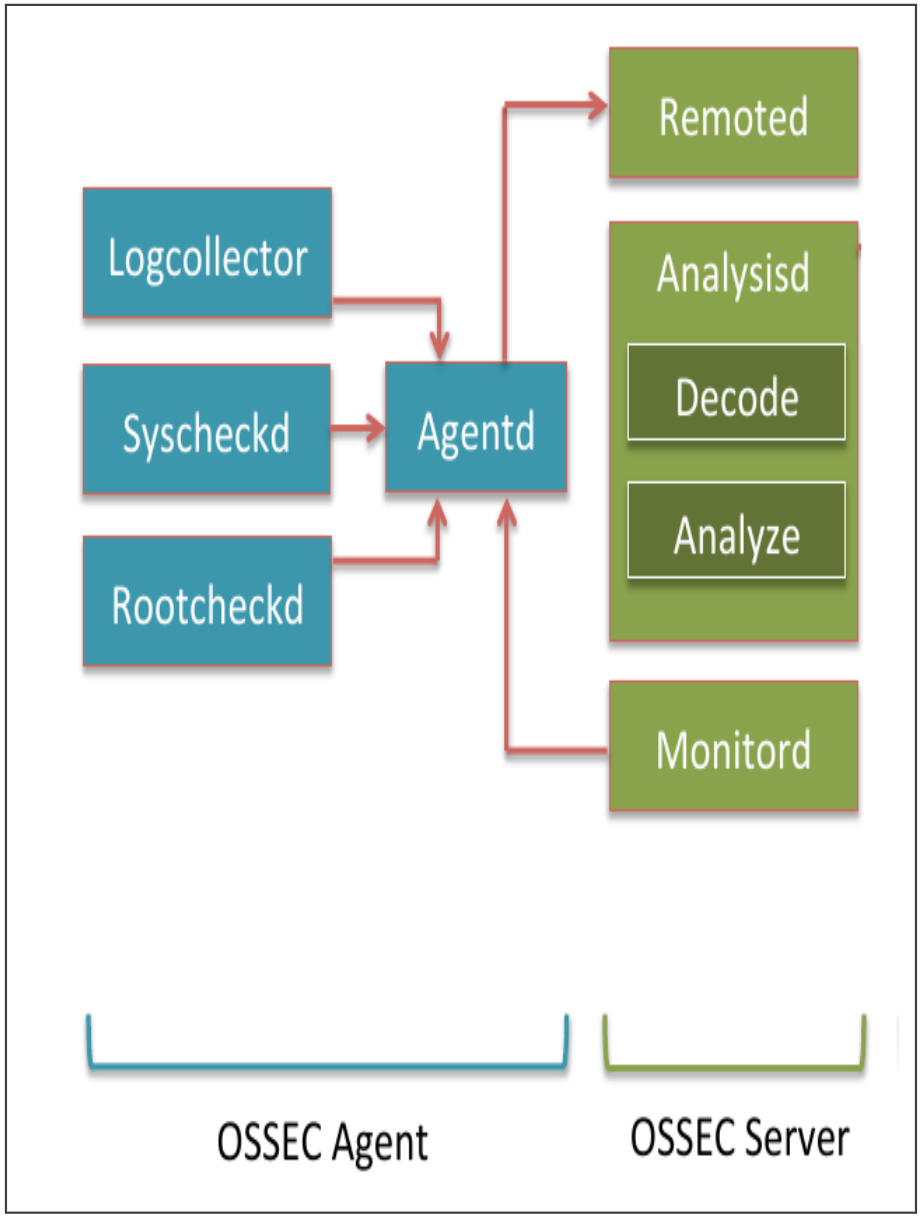
Main tasks

- Log analysis
- File Integrity Monitoring (UNIX & Windows)
- Host-based anomaly detection (rootkit detection)
- Real time alerting & Active Response

- `http://www.ossec.net`
- `http://www.github.com/ossec/ossec-hids`

Architectural overview

OSSEC Processes

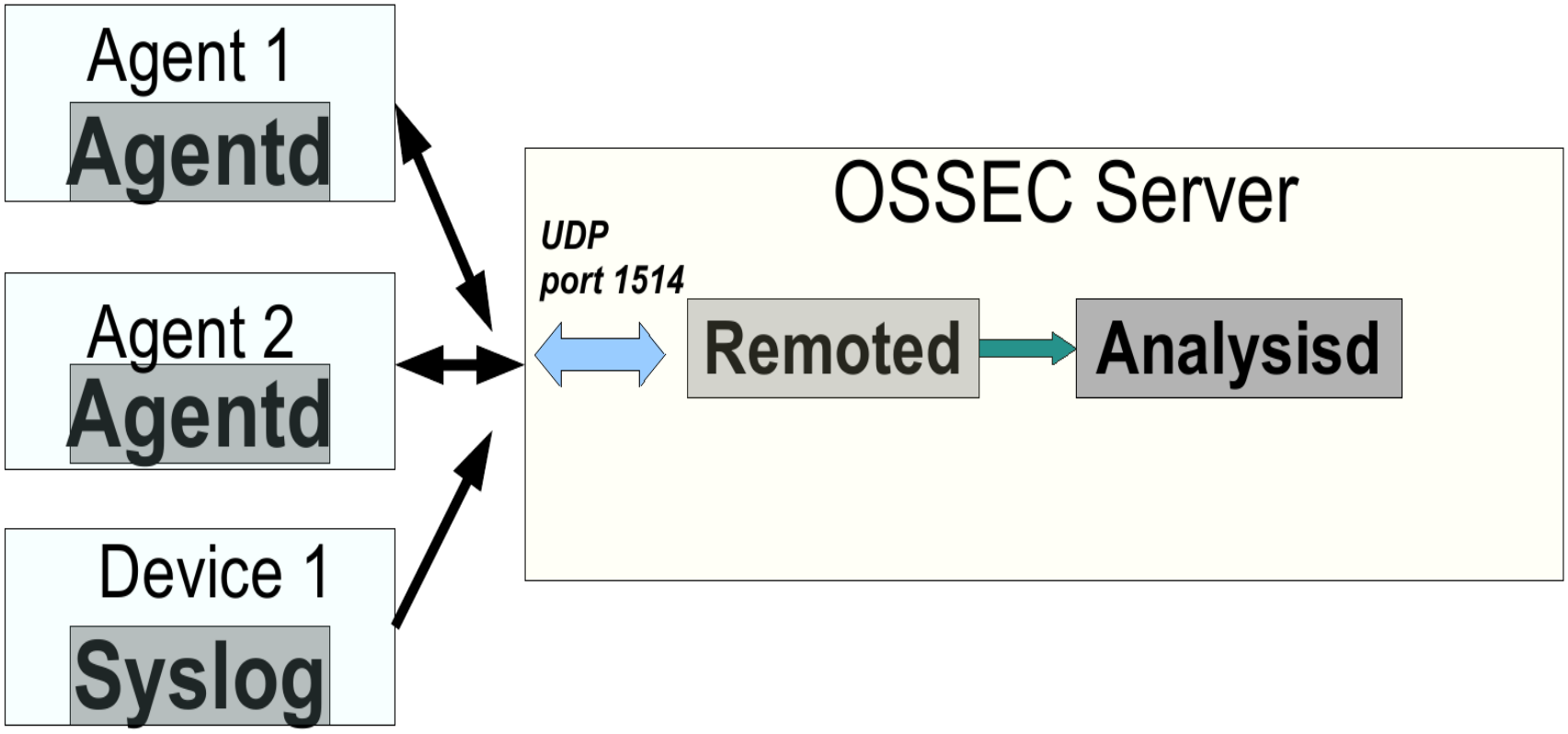


OSSEC Processes

- Each process is executed with limited privileges and tasks
  - all processes (except for logcollector) run in a chroot environment
  - all processes (except for logcollector) are executed with separate (unprivileged) users

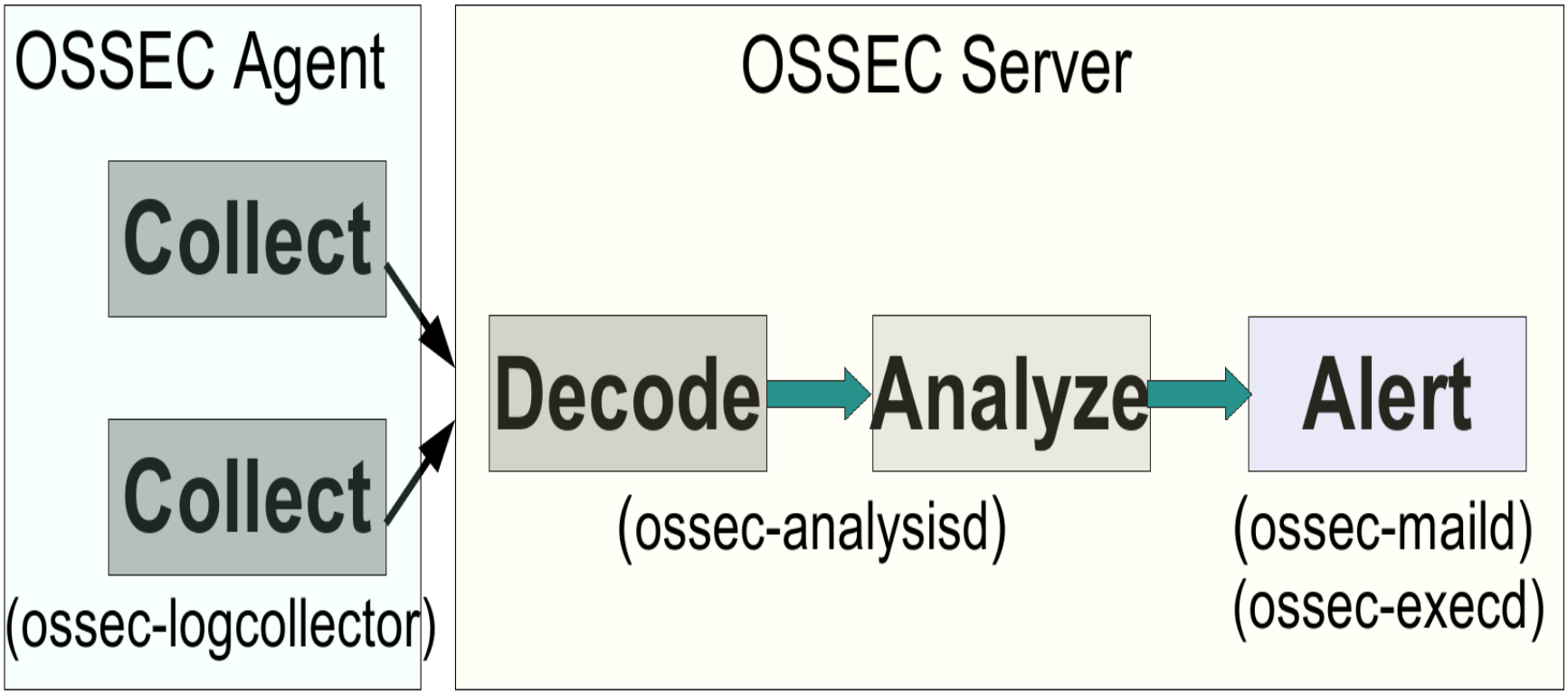
- /var/ossec/bin/ossec-control start script that executes the OSSEC processes in the right order.

Network Communication



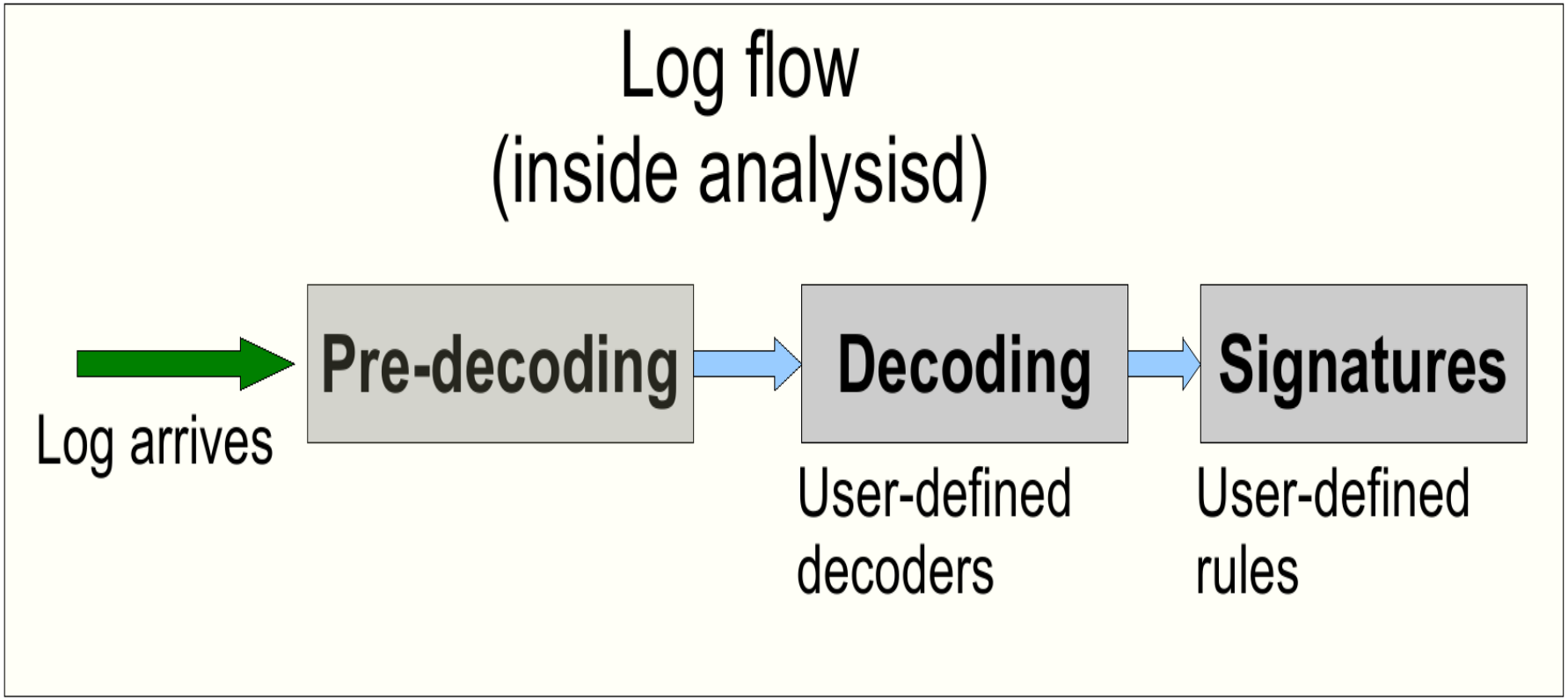
- compresses the log messages with zlib
- encrypted channel with pre-shared keys (blowfish)
- syslog protocol UDP port 1514 (FW clearance!)

Log Flow (agent/server)



- ossec-logcollector on the agent collects all the logs
- ossec-analysisd on the manager analysis the log entries
- ossec-maild sends out alerts
- ossec-execd used for Active Response (Real-Time Alerting)

Internal Log Flow



- 3 parts:
  - Pre-decoding (extracts known fields from the Syslog header)
  - Decoding (identifies key information: SRC IP, Username)
  - Signatures (user-defined rules)

Why do we need log analysis?

Why analyze logs?

- logs are essential for troubleshooting a problem
- not just intrusions or potential security risks
- but also identifying everyday problems
- without logs you have no idea what's happening on your system.

How to detect a rootkit with OSSEC?

How can we detect them?

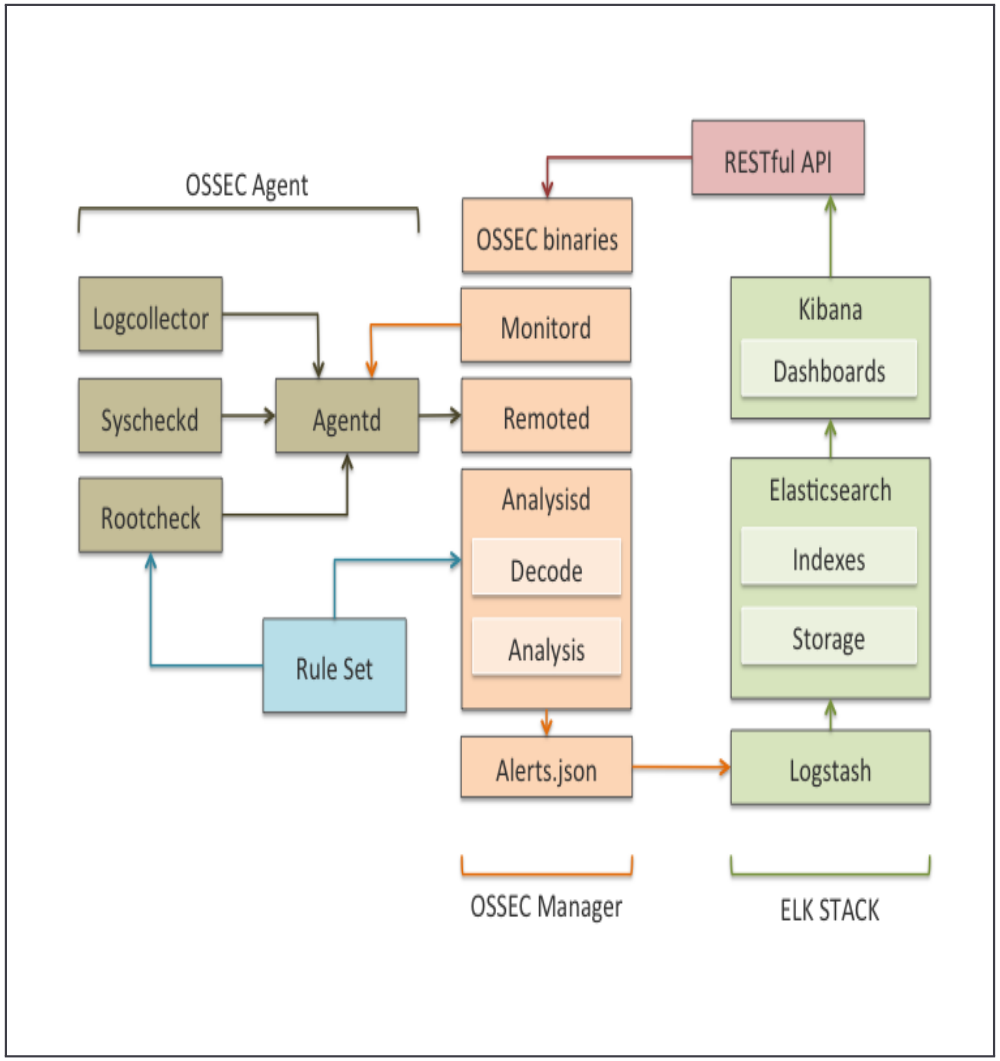
- OSSEC monitors changes of files, directories and commands by performing file integrity checks on these files. –> syscheck module.
- file integrity monitoring: comparing _current_ checksums (hashes) of files with known “good” hashes.
- directories that are hashed by default include: /bin, /usr/bin, /sbin, /usr/sbin and /etc
- Interval of each syscheck: 79200 seconds (22 hours) easily configurable in /var/ossec/etc/ossec.conf
- two files for rootkit detection in OSSEC:
  - rootkit_files.txt contains a list of file names known to be user mode rootkits.
  - rootkit_trojans.txt contains signatures that known rootkits have embedded in the binary file. by default the binaries in /bin, /sbin, /usr/bin and /usr/sbin are searched.

Signature detection

- Rootcheck module extracts strings from binaries and uses a RegEx to identify a match. Referred to as “signature detection” -> many rootkits contain unique strings in trojaned versions of Linux utilities, e.g ps or netstat.
- additional signatures can be added to the rootkit_trojans.txt
- Rootcheck module generates an alert if there’s a discrepancy in information about a file, process port or network interface.
- relevant linux utitilies for Rootkits are:
  - ps
  - stat
  - netstat

ELK Stack Integration

enhanced OSSEC with ELK Stack Integration



Links

- http://www.ossec.net
- http://github.com/ossec/ossec-hids
- http://github.com/wazuh
- http://www.wazuh.com

Live-DEMO