# Host-Based Intrusion Detection System

# Host-Based Intrusion Detection System

- Used to analyze data that originates on the computers, such as application and operating system event logs and file attributes.
- That data is then scanned for pre-defined patterns of misuse
- Becoming increasingly important because of the insider threat.

# Host-Based Intrusion Detection System

- Host event logs contain information about file accesses and program executions associated with inside users.
- This proximity also provides excellent trending and damage control assessments.
- If protected correctly, event logs may be entered into court to support prosecution of computer criminals.

# Host-Based Intrusion Detection

- Host-based systems guard against:
  - Abuse of privilege : When a user has root, administrative or some other privilege and uses it in an unauthorized manner.
  - Contractors with elevated privileges : When an administrator gives a contractor elevated privileges to install an application, but forgets to remove those privileges.

- Ex-employees utilizing their old accounts
  - Policies to delete accounts when individuals leave might take time, leaving a window for a user to log back in.
- Modifying website data
  - Against government agencies in particular, that result in uncomplimentary remarks posted on web sites.

# Other threats to Hosts

- Critical data access and modification
- Changes in security configuration

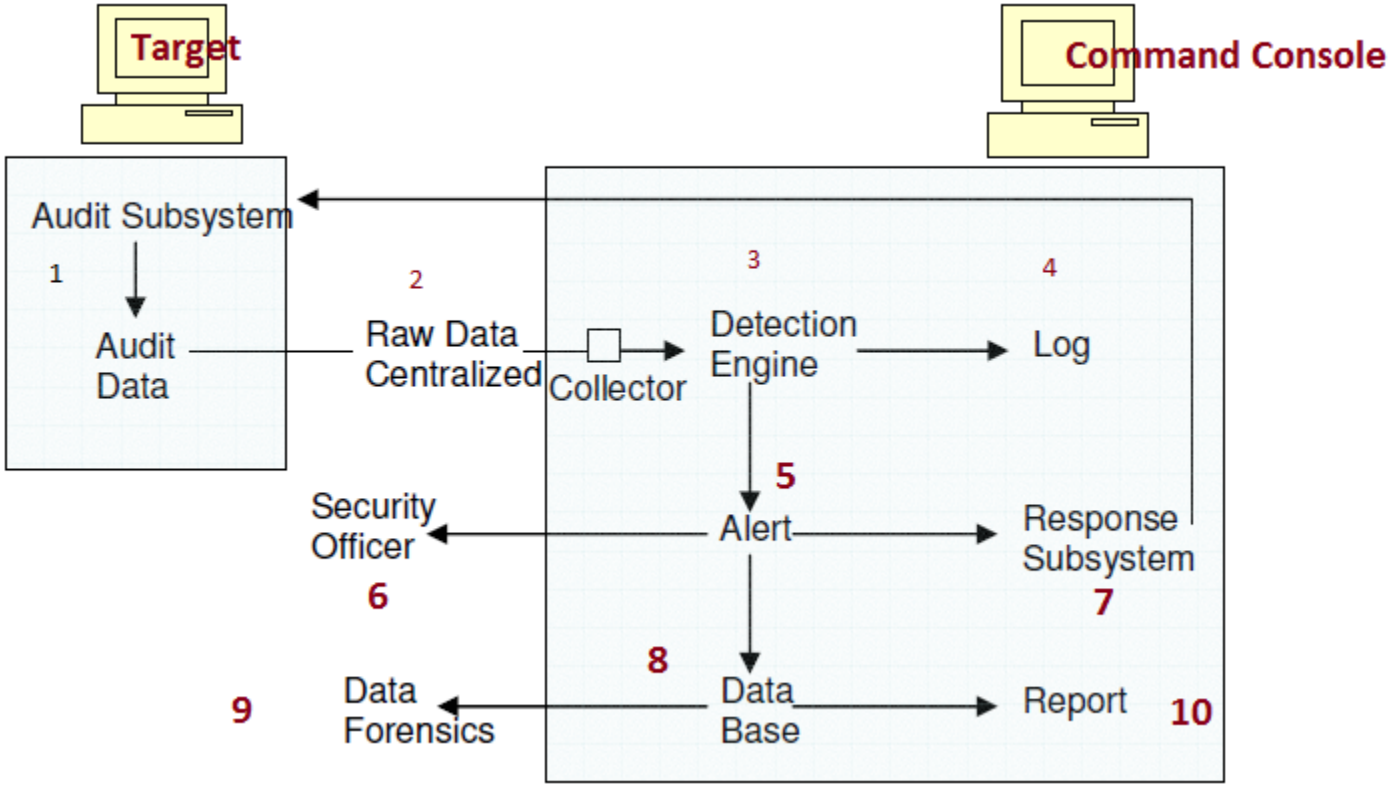
# Host-Based Intrusion Detection Architecture

- HIDs are usually agent based
- Agents are small executables that run on the target system
- Agents communicate with a central control computer (command console)

# Host-Based Intrusion Detection Architecture

- In the centralized architecture, data is forwarded to an analysis engine running independently from the target.
- In the distributed architecture, raw data is analyzed in real time on the target first and then only alerts are sent to the command console.

# Centralized Host-Based Intrusion Detection Architecture



1. An event record is created.
2. The target agent transmits the file to the command console. This happens at predetermined time intervals over a secure connection.
3. The detection engine, configured to match patterns of misuse, processes the file.

4. A log is created that becomes the data archive for all the raw data

5. An alert is generated.

6. The security officer is notified.

7. A response is generated. The response subsystem matches alerts to predefined responses or can take response commands from the security officer.

- Responses include reconfiguring the system, shutting down a target, logging off a user, or disabling an account.

8. The alert is stored.

9. The raw data is transferred to a raw data archive. This archive is cleared periodically to reduce the amount of disk space used.

By using both the stored data in the database and the raw event log archive, data forensics is used to locate long-term trends and behavior

10. Reports are generated. Reports can be a summary of the alert activity.

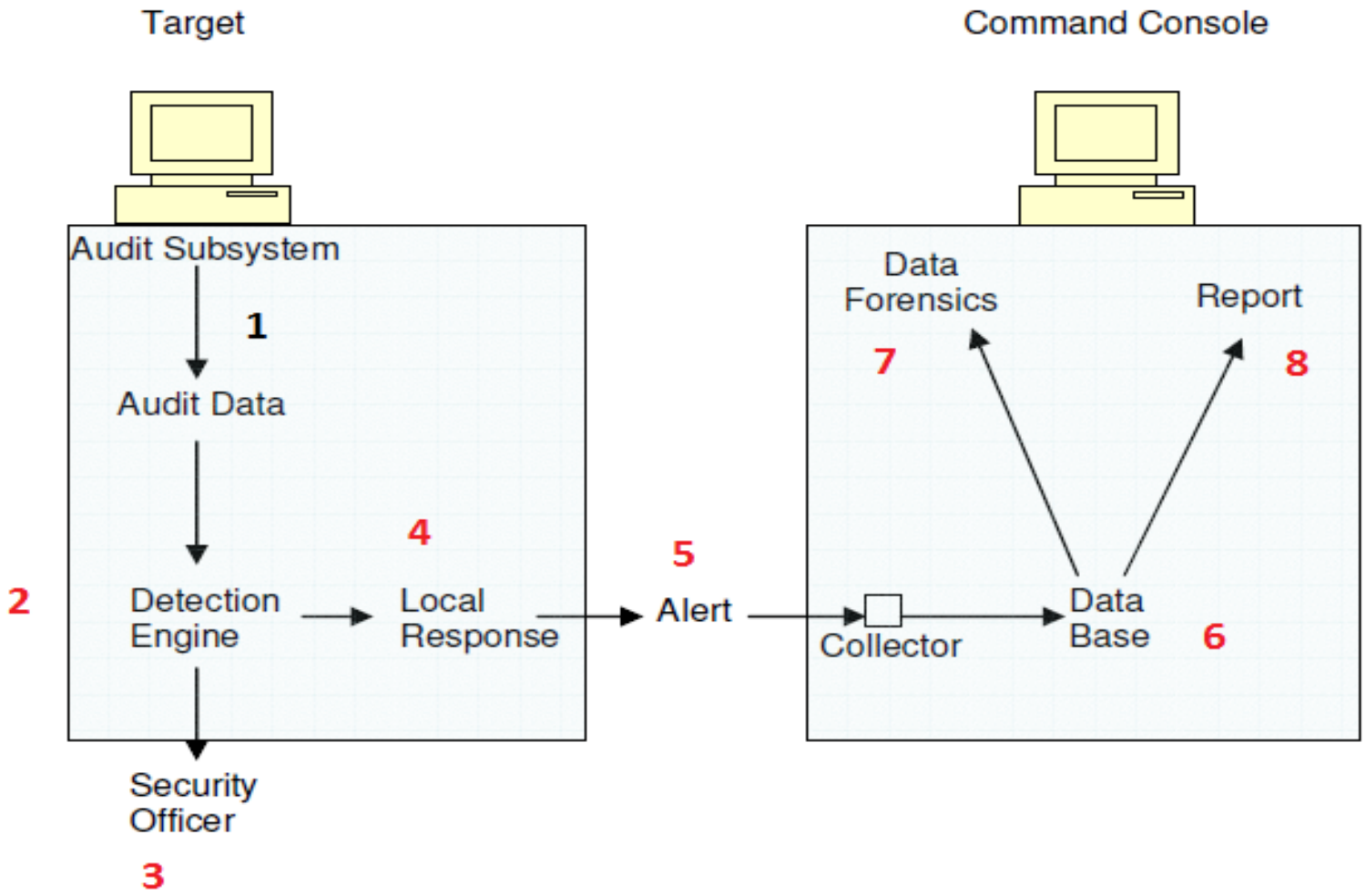
# Advantages and Disadvantages of a Centralized Detection Architecture

| Advantages | Disadvantages |
| --- | --- |
| No performance degradation on target<br>Statistical behavioral information<br>Multi-host signatures<br>Raw data archive for prosecution support | No real-time detection<br>No real-time response |

# Distributed Real-Time Host-Based Intrusion Detection Architecture

- The lifecycle of an event record through a distributed real-time architecture is similar to centralized approach, except that the record is discarded after the target detection engine analyzes it.

# Distributed Real-Time Host-Based Intrusion Detection Architecture



1. An event record is created.

2. The file is read in real-time and processed by a target resident detection engine.

3. The security officer is notified. Some systems notify directly from the target, while others notify from a central console.

4. A response is generated. The response may be generated from the target or console.

5. An alert is generated then sent to a central console.

6. The alert is stored. Statistical behavioral data outside alert data are not usually available in this architecture.

7. Data forensics is used to search for long-term trends. However, because there is no raw data archive and no statistical data, this capacity is limited.

8. Reports are generated.

# Advantages and Disadvantages of a Distributed Real-Time Architecture

| Advantages | Disadvantages |
| --- | --- |
| Real-time alerting<br>Real-time response | Performance degradation on target<br>No statistical behavioral information<br>No multi-host signatures<br>No raw data archive for prosecution support<br>Reduced data forensics capabilities<br>Gaps in data analysis when system is offline |

# Deterrence

- host-based intrusion detection detects insider misuse
- Host-based systems are designed more to deter insiders, but can’t effectively deter outsiders.

# Real-Time Response

- Host-based systems provide poor real-time response but cannot effectively protect against one-time catastrophic events.
- They are, however excellent at detecting and responding to long-term attacks

# Determining the Extent of a Compromise

- Host-based systems stand out when it comes to determining the extent of a compromise after loss.
- They usually maintain large databases that indicate historical information that could serve as evidence in the prosecution of a misuser.

# Trending

- Host-based systems maintain large databases of behavioral information that can be mined for trends indicative of misuse.
- An example : identifying a user who is scanning for sensitive data.

# Policies

- Policies drive the operation of an intrusion detection system.
- Effective policy can significantly reduce performance degradation and resource costs associated with operating a host based detection system
- Audit and detection policies are two primary policies that need to be managed effectively.

# Audit Policy

- Audit policy defines which end user actions will result in an event record being written to an event log.
- Reducing the number of event log records collected can reduce the performance related issues with host-based systems

# Audit Policy

- Audit policies are complex affairs involving multiple flags in numerous locations.
- The large majority of events are caused by system object accesses.
- To control the audit subsystem to a fine detail, audited events are restricted to access of mission critical files, folders, and objects only.

# Audit Policy

- Capturing legitimate behaviors introduces the concept of positive and negative auditing.

# Negative Audit Policy

- Traditional exception auditing
- This includes:
  - Failed login attempts
  - Failed file reads
  - Failed file modifications

# Positive Audit Policy

- Positive auditing is the process of logging both successful and failed events.
- This represents a reduction in the number of false positives.
- A specific example is keeping track of all users who access a certain directory.

# Audit Policy Trade-Off

- Enabling event failures would indicate potential problem areas, while enabling full auditing would provide too much information for view and cause significant performance penalties.
- The exception auditing became the standard audit policy.

# Effective Audit Policy

- The key to a successful intrusion detection system is an effective audit policy.
- An effective audit policy is one that provides a suitable number of event records.
- That is, not so much that they can't effectively be analyzed, and not so little that interesting behaviors are lost.

# Detection Policy

- It defines the patterns that are detected in the event log records.
- Signature recognition is a common mechanism in host based systems.
- The key to good detection policy is properly configured signatures
- Audit policies and detection policies are dependent on each other.

# Information Sources for HIDs

- Host-based IDSs (HIDS) detect attacks against a specific host by analyzing audit data produced by the host operating systems.
- Audit sources include:
  - OS event log
    - syslog
    - UNIX binary kernel log
  - MS Windows event log
    - System log
    - Security log
    - Application log

# Syslog

- Syslog is an audit facility provided by many UNIX-like operating systems.
- This service receives a text string from the application, prefixes it with a time stamp and the name of the system on which the application runs, and then archives it, either locally or remotely.
- Because of their simplicity, syslog events are used extensively by applications(*login, sendmail, nfs, http*, including security-related tools such as *sudo, klaxon,* or *TCP wrappers*)

# System Log

- Contains events related to Windows services and drivers
- It tracks events during system startup, hardware and controller failures

# Security log

- Tracks security related events, also recourse usage.
  - Logons, logoffs
  - Change to access rights
  - System startup and shutdown

# Application log

- It is used for events generated by applications
- Might record file access errors, problem with application configuration

# Types of events detected by HIDs

- A host based IDS involves the installation of agents on the monitored host.
- These agents then monitor the host for security events occurring within that host.

# Types of events detected by HIDs

- The types of events commonly detected by host based IDS:
  - Code Analysis
  - Network Traffic Analysis
  - File system Monitoring
  - Log Analysis
  - Network Configuration Monitoring
  - Rootkit Detection

# Code Analysis

- Code analysis methods are
  - Code behaviour analysis
  - Buffer overflow detection
  - System call monitoring

# Network Traffic Analysis

- Host based IDS agents often include a host based firewall that can restrict incoming and outgoing traffic for each application on the system.

# File system Monitoring

- Can be performed by using
  - file integrity checking
  - file attribute checking
  - file access attempts

# Log Analysis

- Some agents can monitor and analyse operating system and application logs to identify malicious activity
- These logs may contain information on system events such as shutting down the system and starting a service; audit records.

# Network Configuration Monitoring

- Some agents can monitor a host's current network configuration and detect changes to it.
- By monitoring registry, host-based intrusion detection systems can detect alterations in system configurations, user groups, and installed software
- Policy violations and misuses can be detected using the information acquired from Windows Registry.

# Rootkit Detection

- Host-based intrusion detection systems can detect rootkits conducting signature based scans or finding anomalies in the results of different system calls

# Configuration and Maintenance of HIDS

Following steps are used to tune a system

1. Determine optimum placement of sensors
2. Determine normal baselines and implement a basic configuration
3. Analyse the logs and alarms whilst normal operations are conducted
4. Filter out false positives through traffic analysis and system use and implement additional filtering as necessary
5. Determine responses to alert types
6. Apply new tuned configuration and return to step 2 as required

# TOOLS FOR HIDS

*OSSEC*

- It is scalable, multi-platform, open source Host-based Intrusion Detection System
- It has a powerful correlation and analysis engine, integrating log analysis; file integrity checking; Windows registry monitoring; centralized policy enforcement; rootkit detection;
- It has real-time alerting and active response.

- *TRIPWIRE*
  - It Detects Improper change, including additions to, deletions from and modifications of file systems and identifies the source.
  - Performs rootkit detection and file integrity checks
  - has a command line based human computer interface

# Case Study: OSSEC Host Based Intrusion Detection System

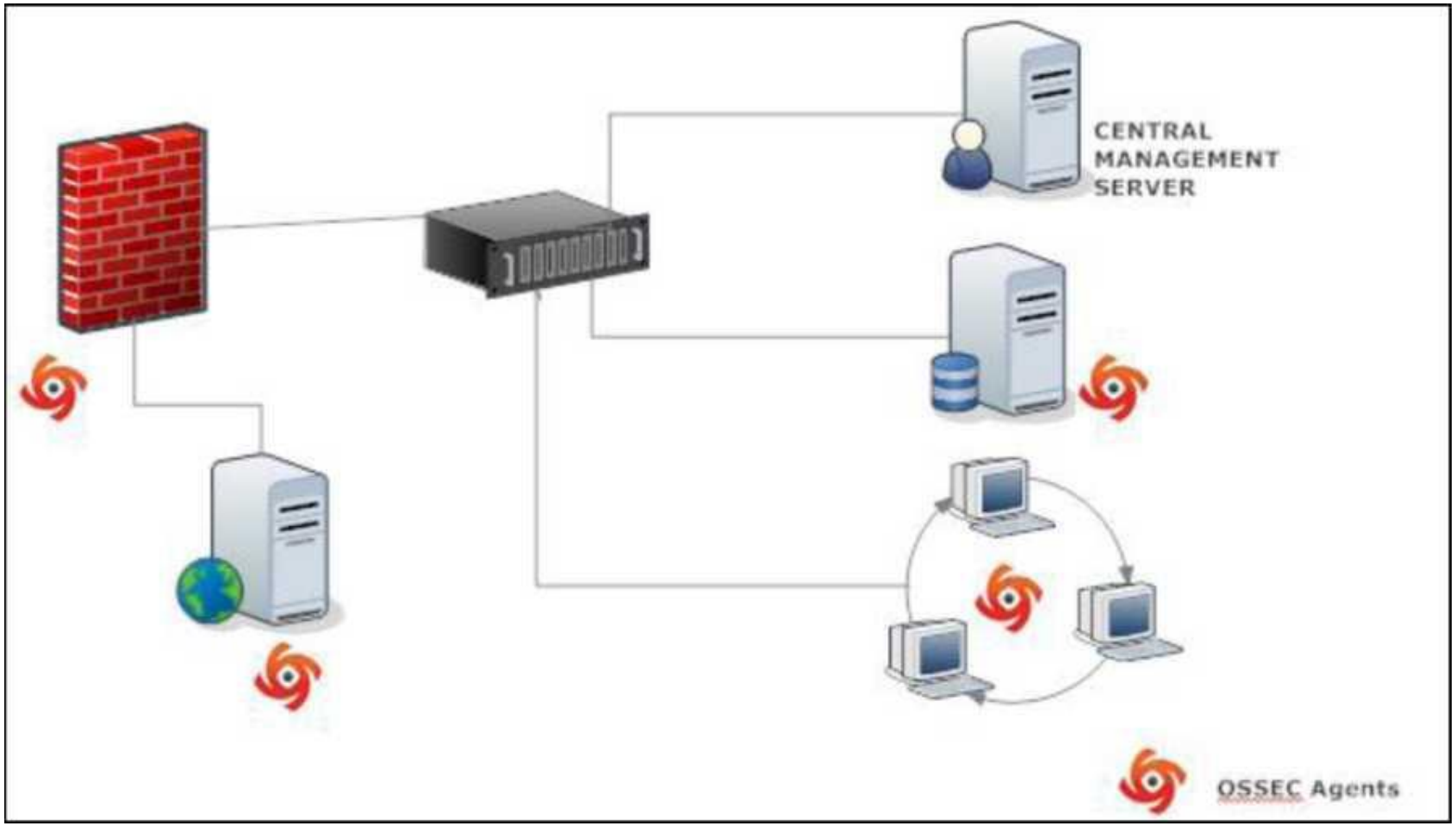
# About OSSEC

- Open source host based intrusion detection system (HIDS)
- It provides
  - Filesystem integrity checking
  - Registry monitoring on Windows
  - Active response
    - -- Can be scripted for almost any behaviour
  - Rootkit detection

# How OSSEC Works

- Three modes
  - Local, client, server
- Client server model
  - Clients receive configuration from server
  - Clients send logs to server over an encrypted channel

# OSSEC Deployment



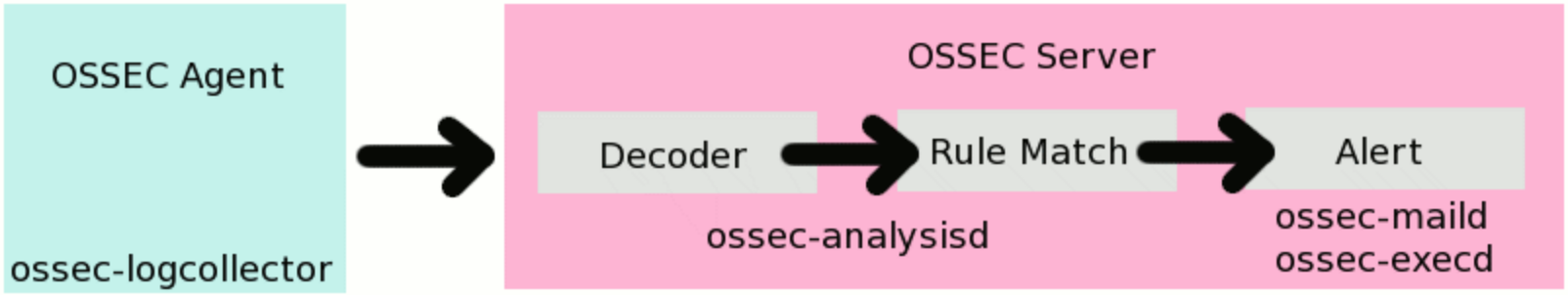
- The agent programs are installed on host machines, firewall and application server.
- All events detected by agents are send to central management server.
- Central management server determines the alert type and preventive action.

# OSSEC Network Communication

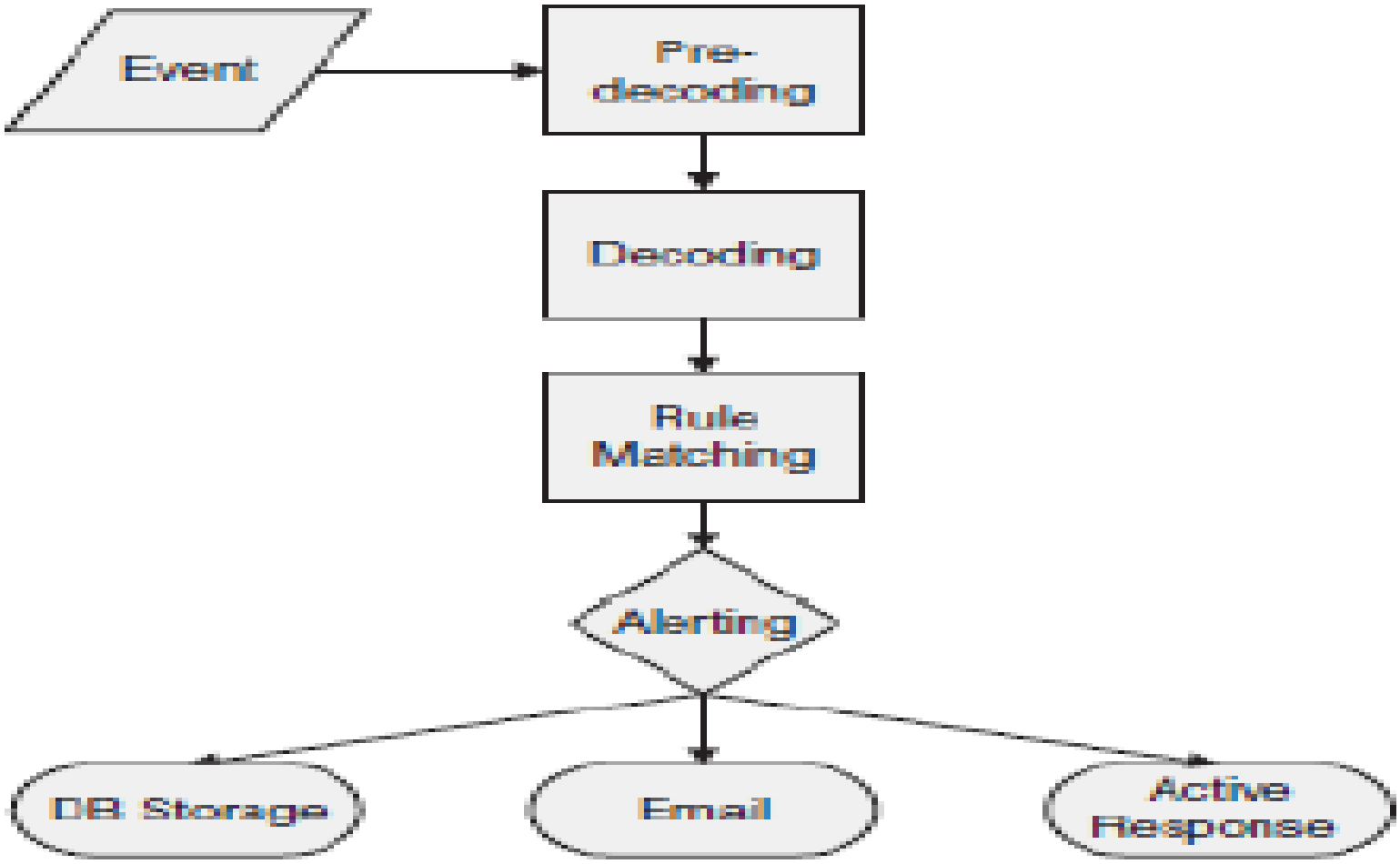
- Compressed messages (using zlib)
- Blowfish based encryption using pre-shared keys
- Logs specified on client are collected and sent to manager for analysis

# OSSEC Client Server Architecture

# OSSEC Data Flow



# OSSEC Process Flow



# OSSEC Decoders

- OSSEC uses decoders to parse log files
- Decoders are written as XML
- Extracts useful data fields from log entries to use for rule and alert matching including:
  - Source IP and/or port
  - Destination IP and/or port
  - Program name or user name

# OSSEC Rules

- OSSEC rules are stored as XML files
- Rules are hierarchical
- By default OSSEC includes rules for:

  apache, arpwatch, asterisk, cisco-ios, courier, firewalls, IDS systems, IMAP, McAfee antivirus, MS auth, MS DHCP, Exchange, Microsoft FTPD, MySQL, Bind, Netscreen, PAM, postfix, Postgres, ProFTP, Roundcube, sendmail, samba, Squit, SSH, Symantec AV, Syslog, Telnet

# Rule Format

- Rules are assigned priority levels, usually from 1 (lowest) to 15
- Rules trigger based on:
  - Pattern matching in strings
  - Timing between matches
  - Dependence on other rules
  - Time of day
  - Hostnames
  - Applications

# OSSEC Alerts

- Creating custom alerts is relatively easy
- Default settings include alerting on:
  - Web attacks
  - SSH brute force
  - Buffer overflows and program crashes
  - Firewall events

# Alert Behavior

- When a rule triggers an alert several actions can be configured:
  - Logging
  - Sending an e-mail alert
    - Sending a SMS alert
  - Executing an active response script

# Response mechanisms

- Response mechanisms can be passive and active
- For passive response, OSSEC can log events, and send email or sms to the responsible personnel.
- As active response, OSSEC central management server can send command to its agents to take countermeasure to the intrusion.

# Active Response

- To initiate an active response, OSSEC requires executable scripts on agent machines.
- The active response can be triggered on a particular agent or a group of agents.
- The server sends command to the target agent to run executable to take preventive action.

# Log File Monitoring

- OSSEC can be configured to monitor any log it can gain access to
- OSSEC monitors specific logs by default, including:
  - Syslog
  - Apache http logs
  - Mail logs

# Monitoring Scripts

- OSSEC can be used to monitor the output of custom scripts
- For instance, OSSEC can generate alerts based on changes to NMAP scan results of specific hosts
- Can also log scripted alerts to common log (syslog)

# Generating Reports

- OSSEC includes ossec-reportd
  - Can be used to generate summary reports
  - Ex. show all brute force attempts and usernames used and number of times attempted
- OSSEC can also log to a database so that SQL can be used for reporting
- Custom scripts can be used to parse alert logs

# OSSEC Community

- Extremely active user community
  - Developer mailing list
- OSSEC mailing list (and Google group)
  - http://groups.google.com/group/ossec-list
- OSSEC wiki
  - http://www.ossec.net/wiki
- Commercial support from Trend Micro
  - ossec.purchase@trendmicro.com

# Overall Impact

- Extend service offerings to client groups
- Centralized log reporting
- Difficulty in upgrades between versions
- Coordinating pre-shared keys can be problematic
- Sometimes agents become unresponsive
- Volume of alerts
- In testing OSSEC is great for early warning but not so good in a post compromise situation

# Reference


- The Practical Intrusion Detection Handbook, Paul E. Proctor, Prentice Hall
- Trend Micro, (2013), Open Source Security, OSSEC Home Page, Available : http://www.ossec.net/