Intrusion Detection Systems

Elevated to the Next Level

Alien8 - Matthias Petermann

22nd Chaos Communication Congress

Agenda

- Attacks and Intrusion Methods
- Why Intrusion Detection?
- IDS Technologies
- Basic Problems
- A hybrid IDS framework
- Remaining problems
- Basic correlation
- An advanced correlation approach

Attacks and Intrusion Methods (1)

- Automatic attacks
  - Worms / Viruses
  - Trojan horses
  - Makes lots of noise
- Manual attacks
  - Difficult to find
  - Cover, Concealment, Camouflage

SONY

Attacks and Intrusion Methods (2)

- Methods
  - Local attacks
    - Privilege Escalation
    - Buffer Overflows
    - Format String attacks
    - Race conditions
    - ...
  - Remote attacks
    - Buffer Overflows
    - Remote Discovery
    - Denial of Service
    - Trojans of all kinds (Bots)
    - ...

Real Life in someones network

- Some have to live with:
  - Crappy software
  - 0day exploits
  - Black boxes
  - Lazy admins
  - Non patch-able systems
  - Trade offs

short *real live environments*

What's Intrusion Detection good for?

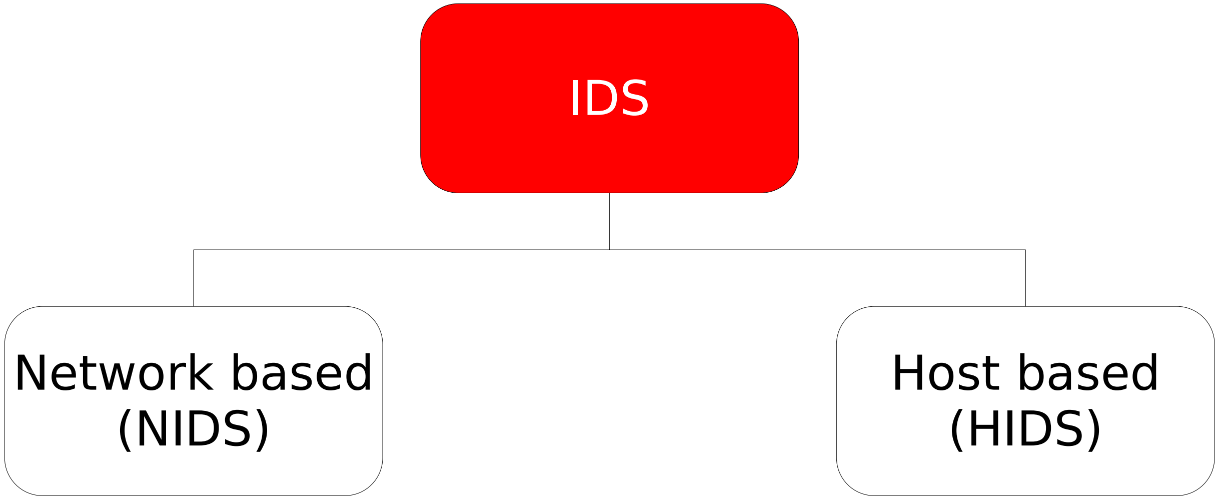
Discover what is going on!

- Intrusion Detection Systems help to:
  - Recognise damage and affected systems
  - Evaluating incidents
  - Trace back intrusions
  - Forensic analysis
- It doesn't compensate for bad security!

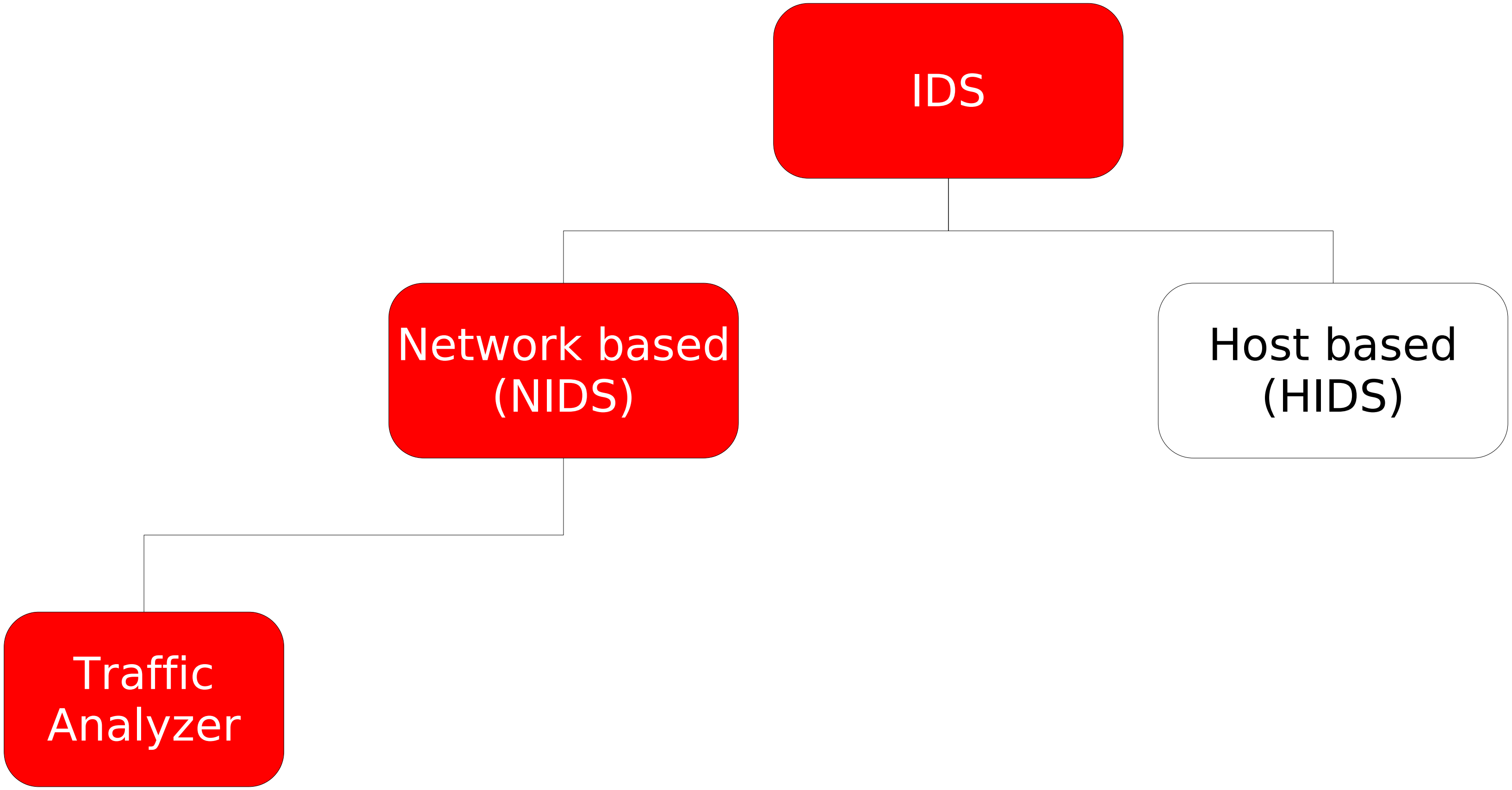
IDS Technologies

IDS

IDS Technologies



IDS Technologies



Network based Technologies (1)

- Traffic analyser (e.g. Snort)
  - Pre-processors for:
    - Detecting portscans
    - Reassembling TCP-streams
    - Decoding RPC, HTTP, ...
    - Detecting viruses (ClamAV plugin)
  - Signature based pattern matching engine:
    - Detecting traffic pattern
    - Detecting protocol violations (x-mas scan)

Snort Signature Rule Examples

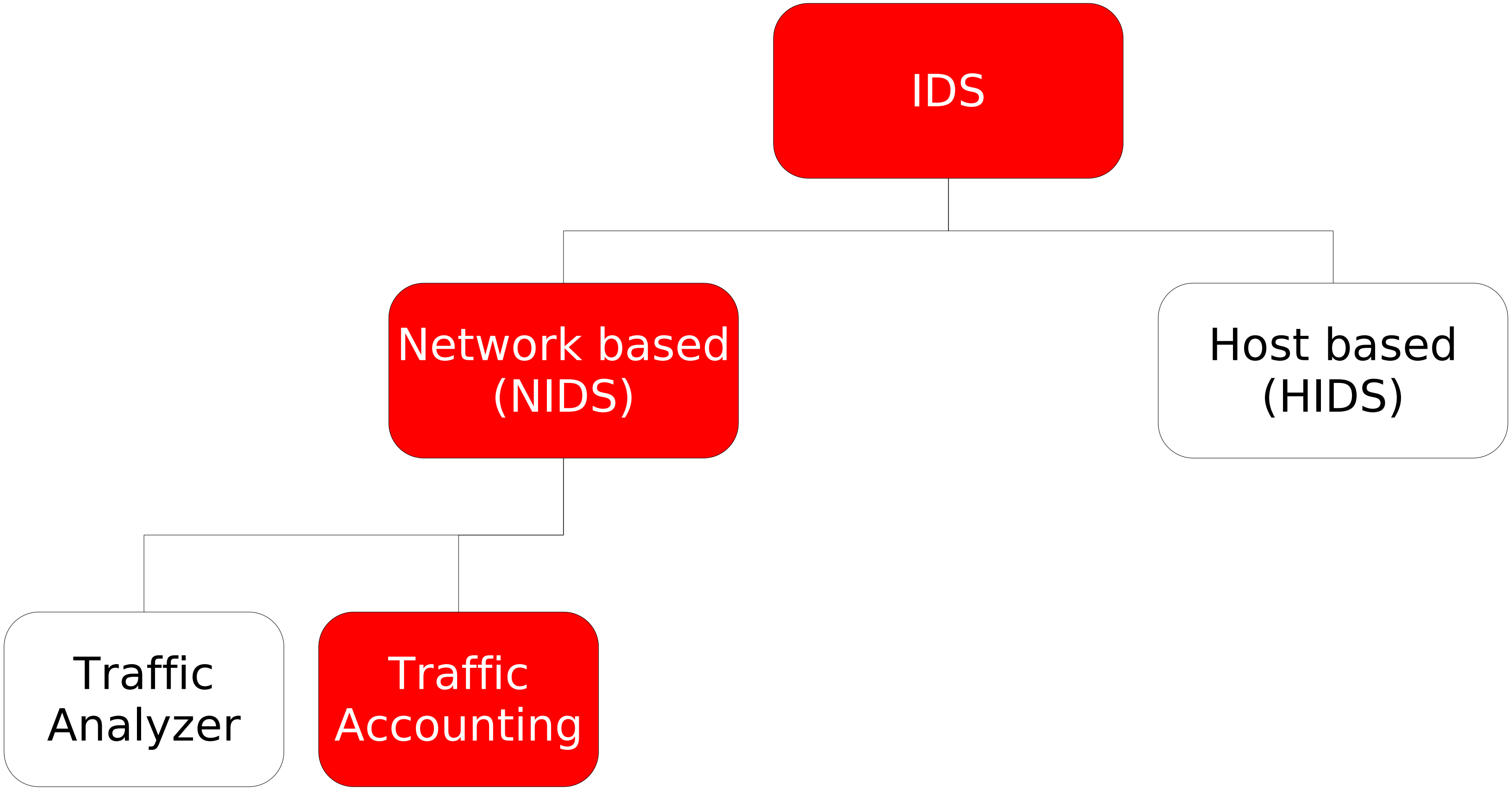
Basic rule to match e. g. telnet connections:

```
alert tcp $EXTERNAL_NET any <> $HOME_NET 23
(msg:"Port23-TRAFFIC tcp port 23
traffic";flow:stateless; classtype:misc-activity;
sid:523; rev:1;)
```

Basic rule to match NetBus backdoor activity:

```
alert tcp $HOME_NET 12345:12346 -> $EXTERNAL_NET any
(msg:"BACKDOOR netbus active"; flow:from_server,
established; content:"NetBus"; reference:arachnids,
401; classtype:misc-activity; sid:109; rev:5;)
```

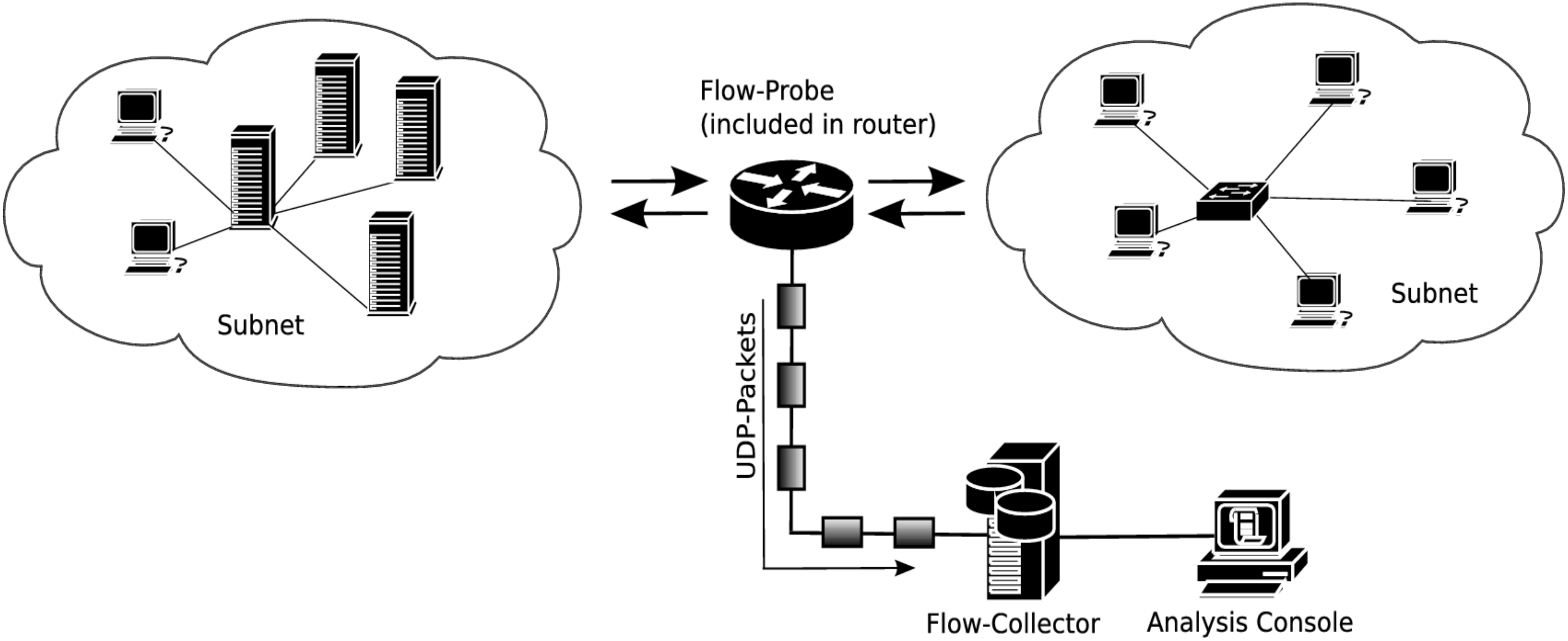
IDS Technologies



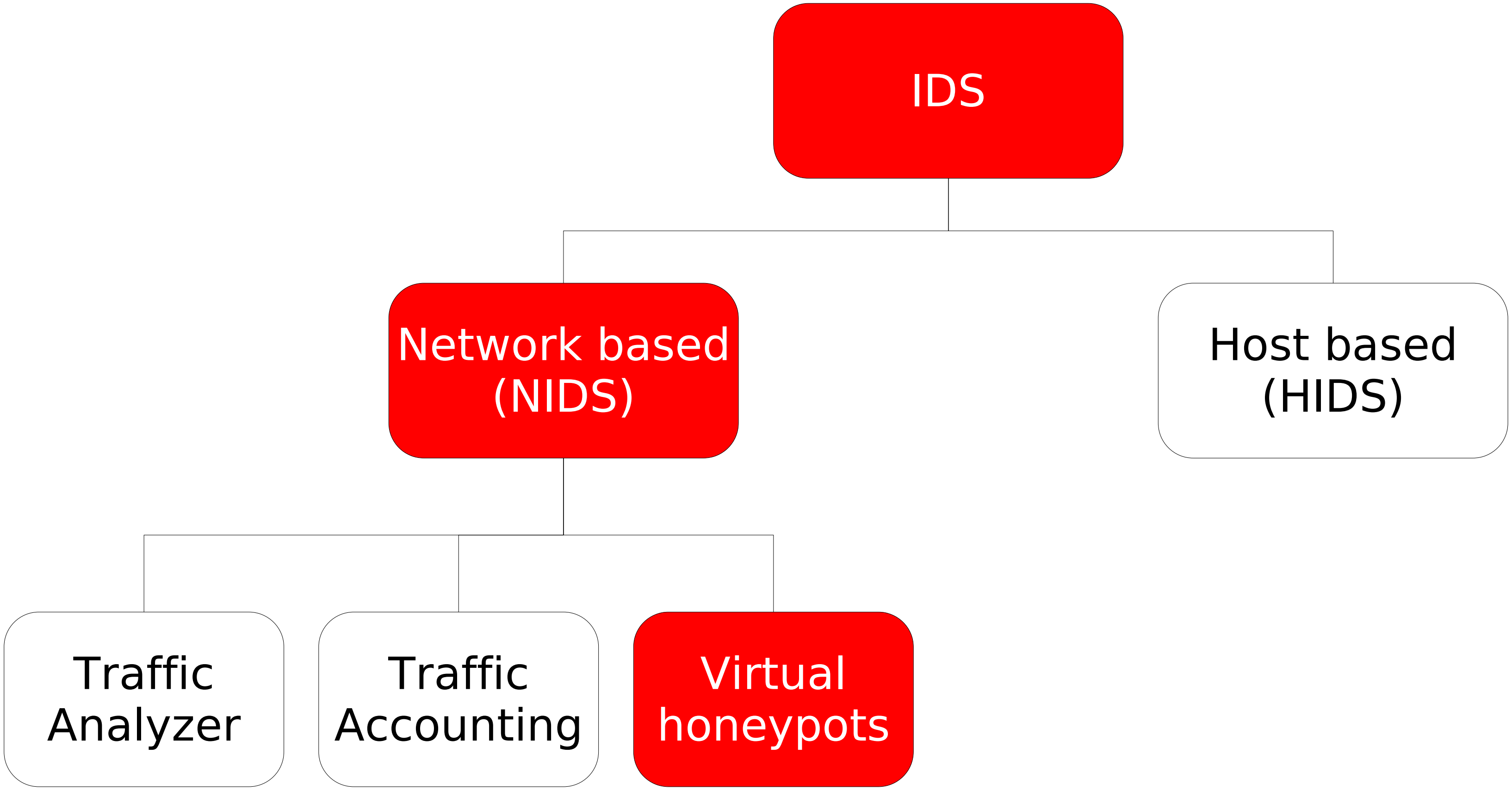
Network based Technologies (2)

- Traffic Accounting (e. g. NetFlow)
  - NetFlow is a standardised protocol
  - Invented for accounting purposes
  - Implementation:
    - Flow-probes and flow-collectors
    - Implemented in routers and switches
    - Implementation: fprobe, flow-tools
  - Value for IDS:
    - Detection of anomalies in network utilisation
  - *Please don't tell Mr. Schäuble about it*

NetFlow Components



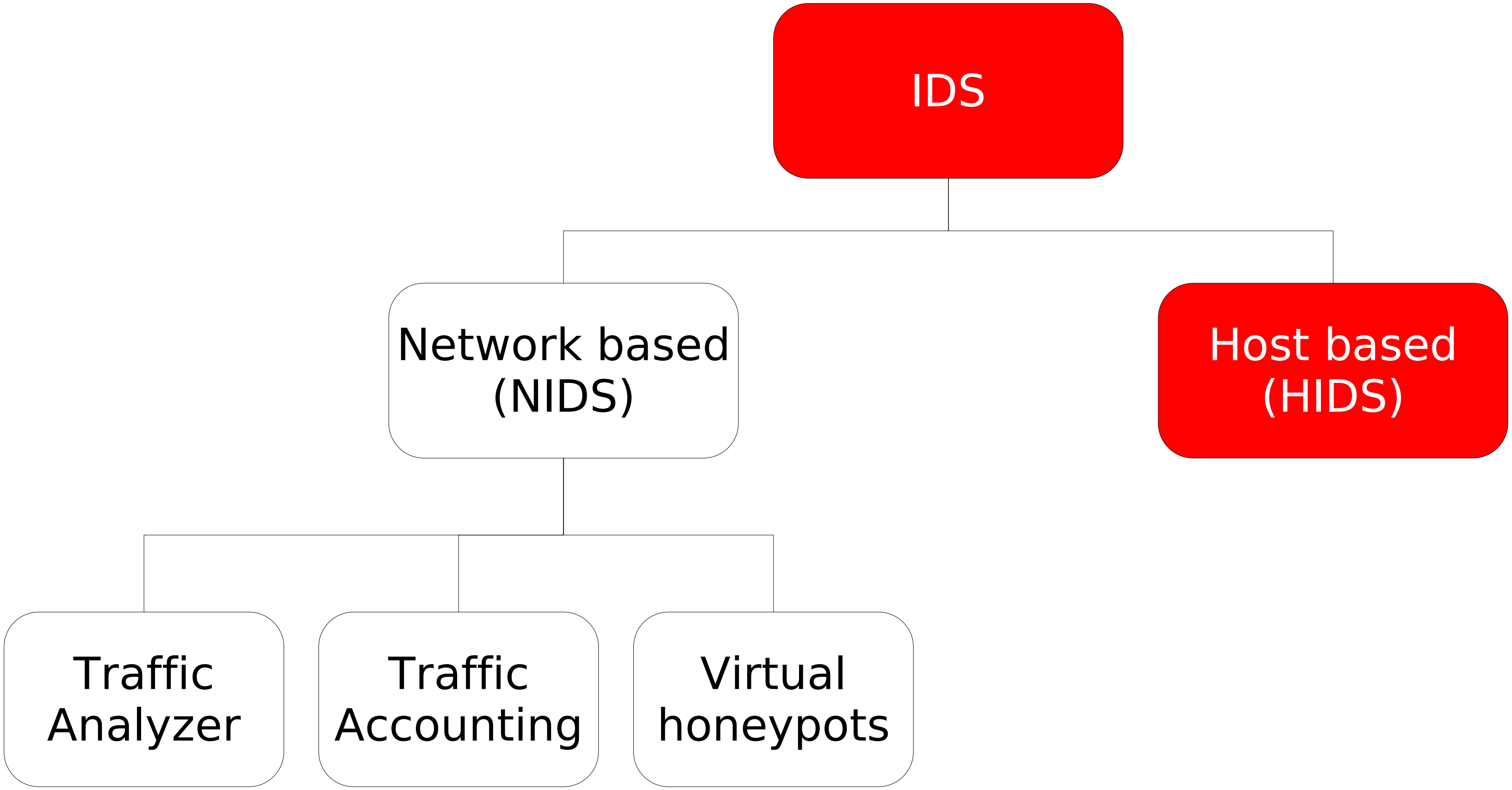
IDS Technologies



Virtual honeypots/-nets

- Honeypot = dedicated system with traps
- No production purpose: access to a honeypot is always suspect!
- “real” honeypots costly to deploy
- -> virtual honeypots (e.g. Honeyd)
  - Emulates whole network topology (routers, switches)
  - Emulates hosts with identity of choice (nmap based)
  - Scriptable “fake”-services
  - Supports forwarding to real services
- Supplement to qualify IDS events

IDS Technologies


IDS

Network based (NIDS)

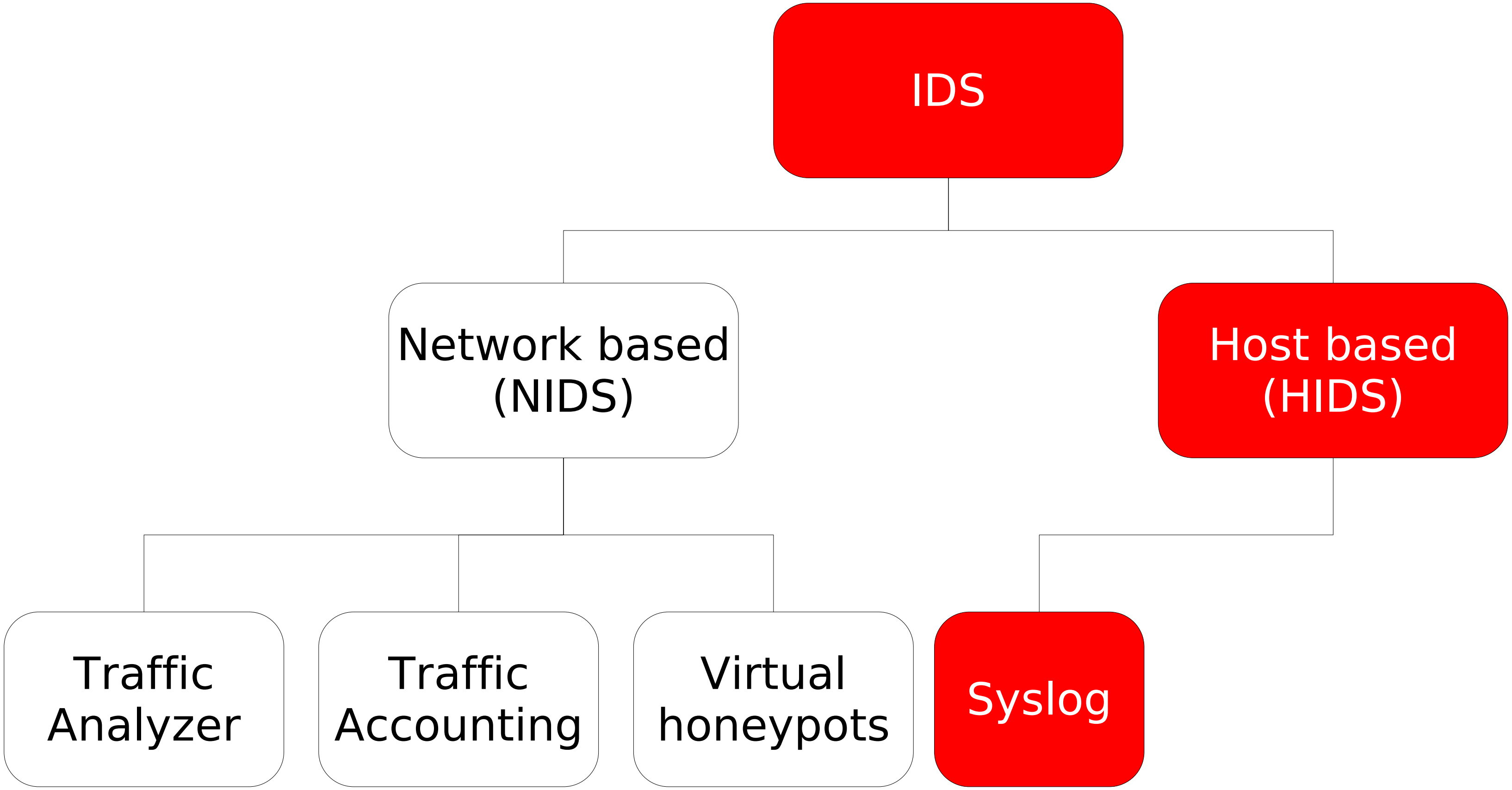
Host based (HIDS)

Traffic Analyzer

Traffic Accounting

Virtual honeypots

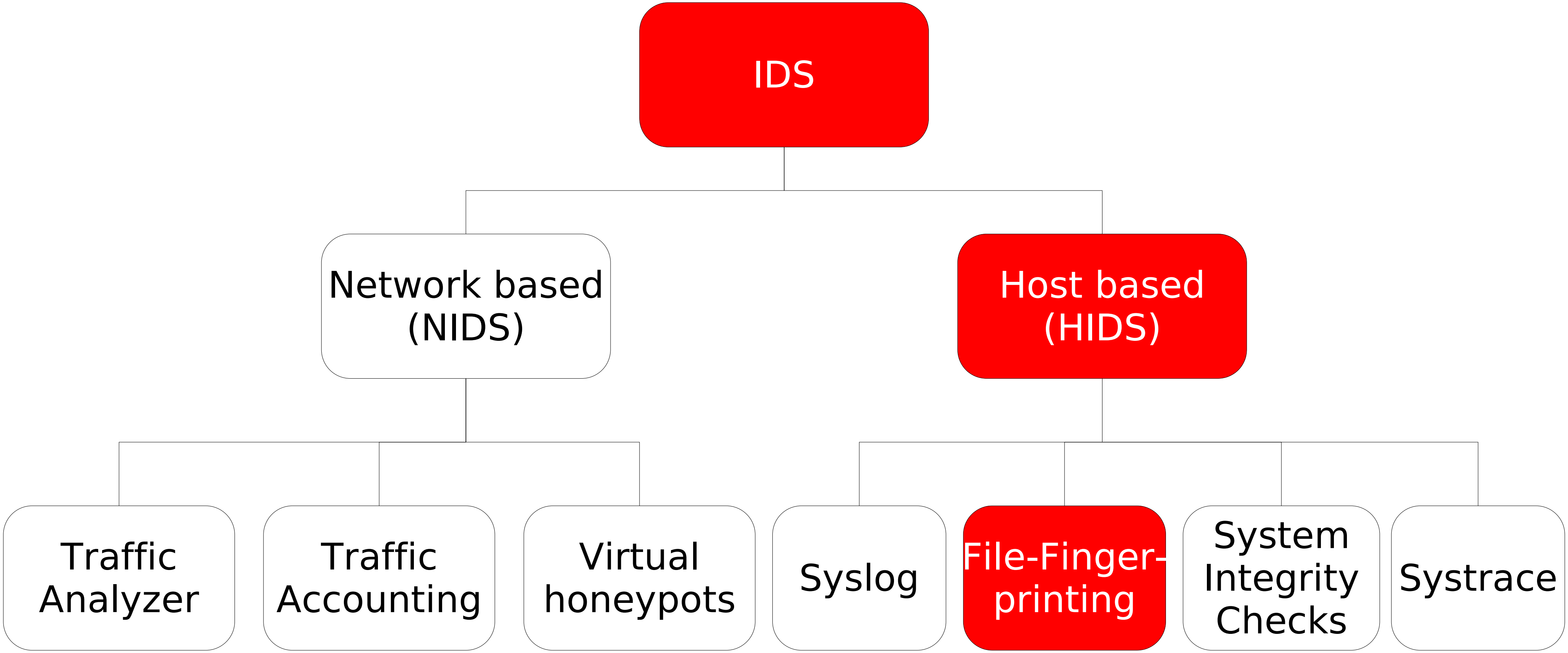
IDS Technologies



Host based Technologies (1)

- Syslog
  - Centralised logging facility for almost everything
  - Analyzing log files tells you about:
    - Failed / successful logins
    - Access to services such as web- or mail servers
    - Firewall (accepted / blocked packets)
    - Creation of new users
    - Hardware events
    - Mounts
    - ...
  - Hard to wipe out logs if logged to external system
  - Tools for analysis: logcheck

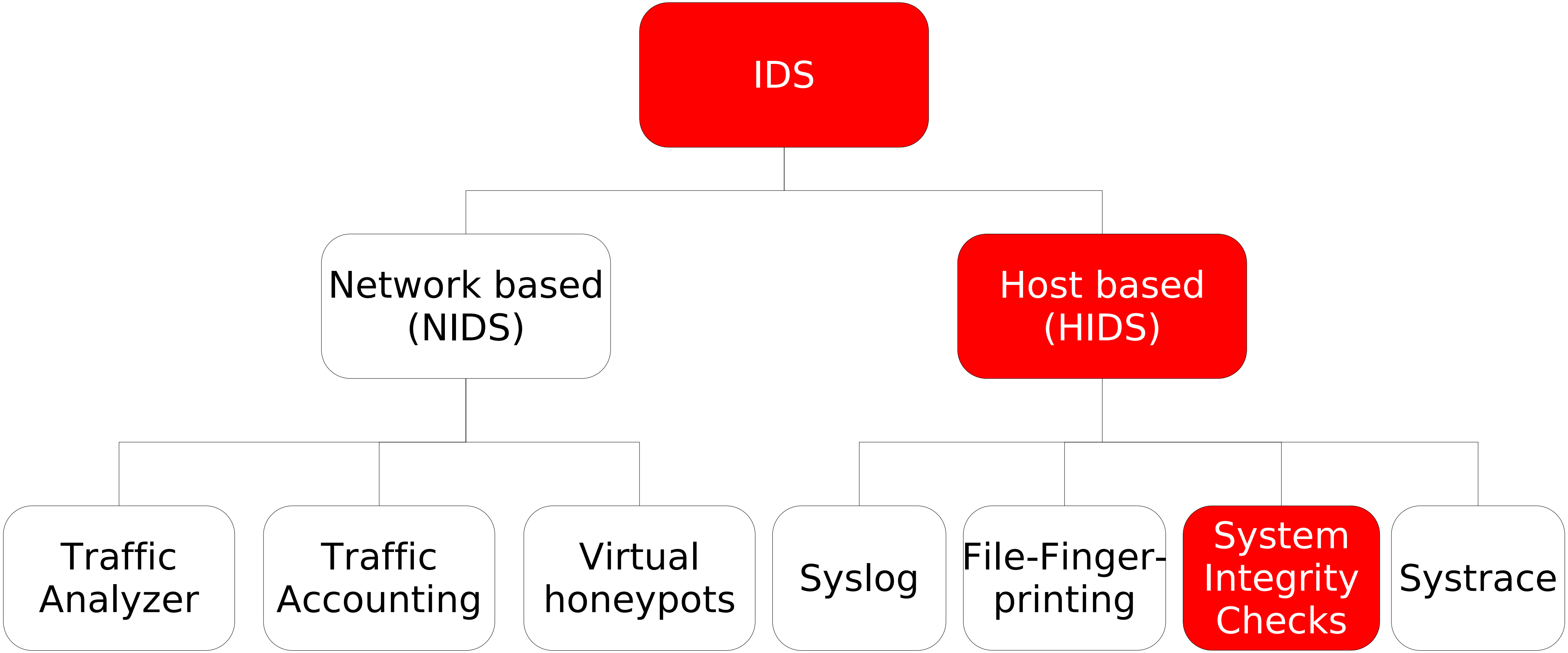
IDS Technologies



Host based Technologies (2)

- File-Fingerprinting
  - Calculates and checks cryptographic hashes of files
  - Detect changed files
  - Additional features (e.g. by Samhain):
    - Detect changed file access rights and time
    - Creation of new files
    - owner/group changes
    - Deletion of files / log files
    - Detect kernel rootkits on Linux and FreeBSD
  - Value for IDS: Detect manipulation of files, Remember: *Everything is a file*

IDS Technologies



IDS

Network based (NIDS)

Host based (HIDS)

Traffic Analyzer

Traffic Accounting

Virtual honeypots

Syslog

File-Finger-printing

System Integrity Checks

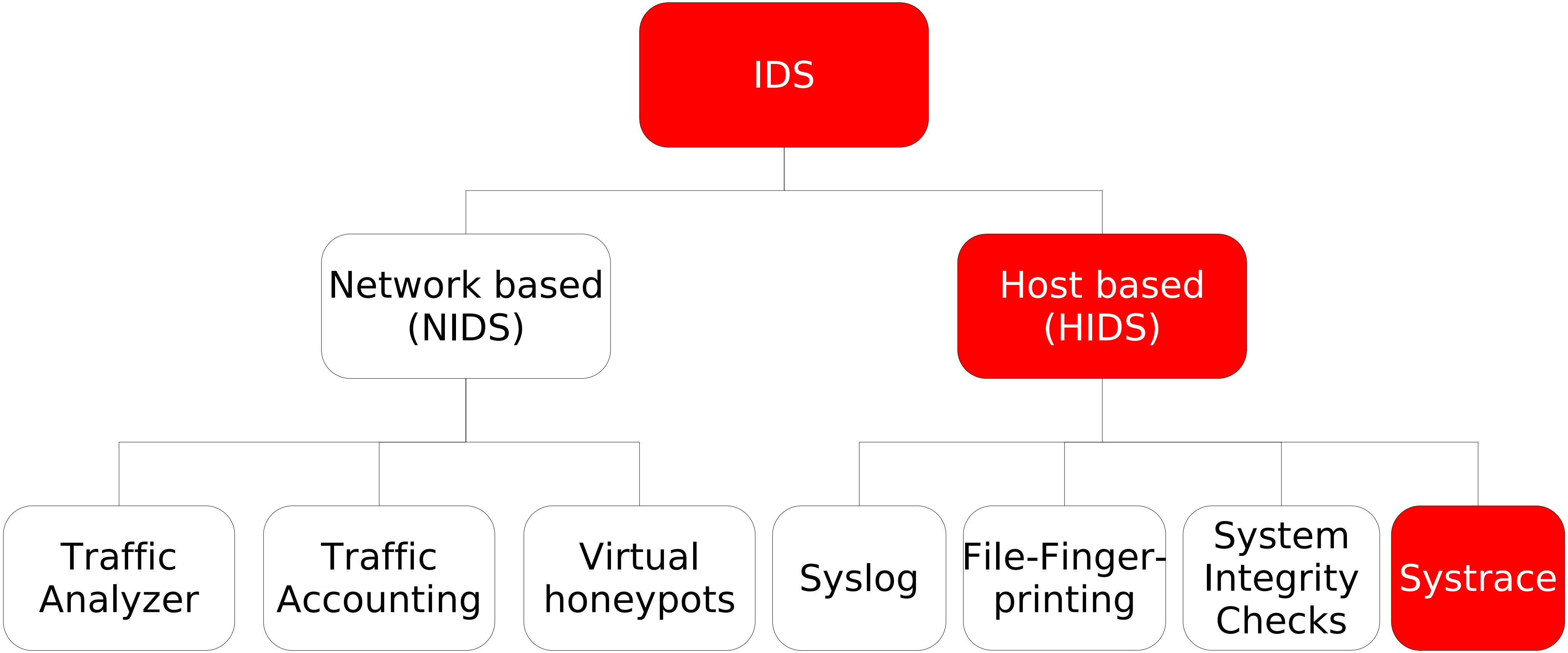
Systrace

Host based Technologies (3)

- System integrity checks
  - Chkrootkit
    - Looks for traces of known root kits
  - Tiger
    - Listening processes
    - Package database checks
      - Unknown files
  - Vulnerability checks
  - Historical performance data
    - Look for anomalies

diversity of tools

IDS Technologies

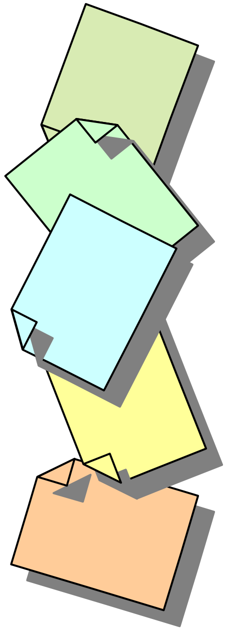


Host based Technologies (4)

- Systrace
  - Security layer for syscalls
  - Can be enabled for selected processes
  - Requested syscall has to match policy
  - Policy manager processes syscall requests
  - Denied syscalls will be logged
  - Implementations
    - Natively included in OpenBSD and NetBSD
    - Kernel patches for Linux and FreeBSD
- RBAC (Role based access control)
  - grsec, rsbac

Current Problems

- IDS implementations not designed to co-operate
- Different storage formats for IDS events
  - Snort: MySQL, flat-files, binary files...
  - NetFlow: sending UDP packets to collector
  - Syslog: flat files or syslog server
  - Samhain: MySQL, Yule, Flat-File
  - Honeyd: flat file
- Distributed data storage
- No common / comprehensive analysis tools (one to do it all)



Requirements for the Ideal System $^{(\mathrm{TM})}$

- Standardised storage format
- Centralised data storage
- Common analysis tool

The Intrusion Detection Message Exchange Format (IDMEF)

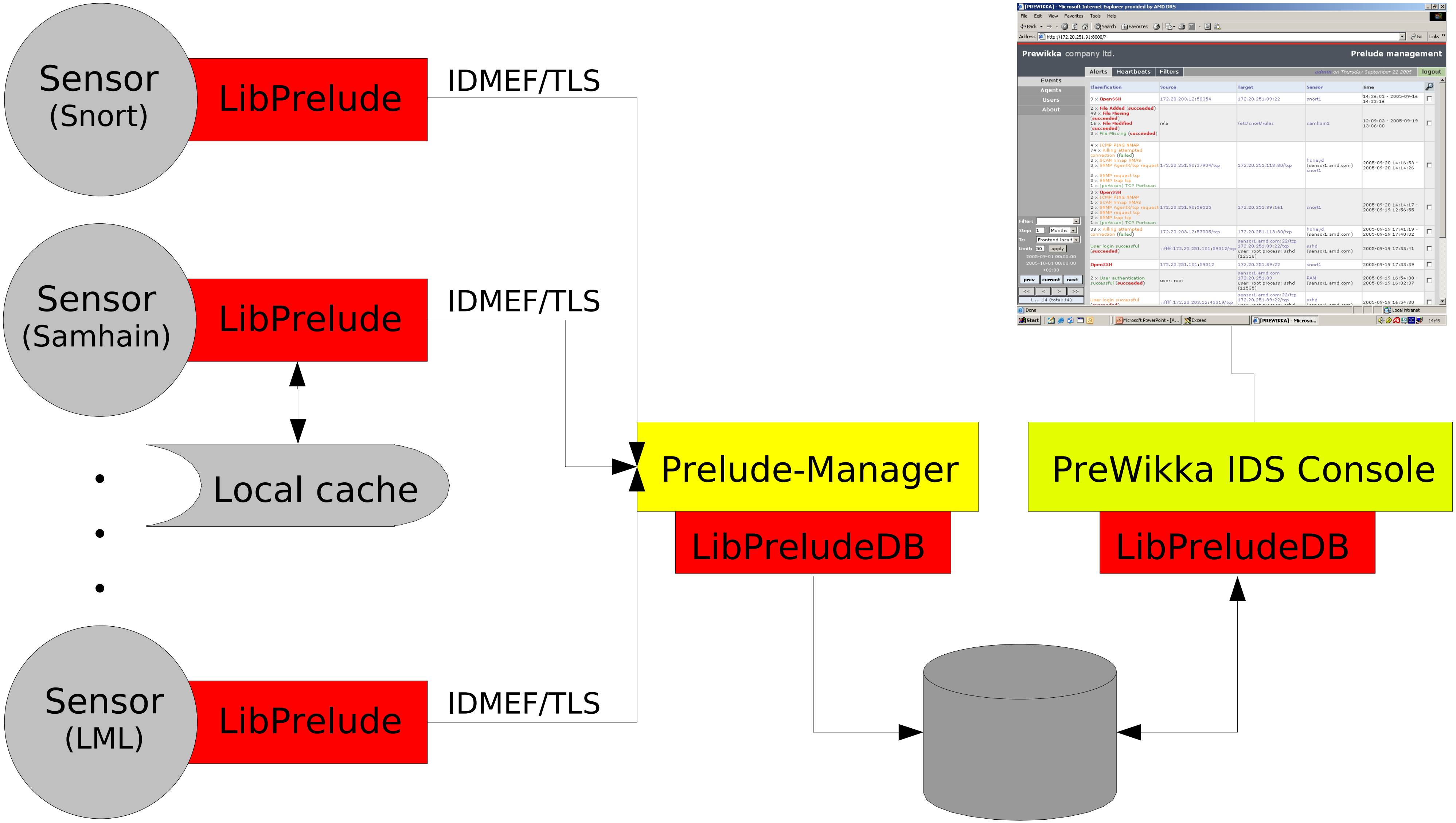
- Problem: Sensors provide different data
  - NIDS: IP-addresses, TCP-flags, payload
  - HIDS: file-names, access-rights
- How to store this in a general format?
  - IDMEF is an object oriented format
  - Reference implementation in XML
- Yet another file format?
  - No! IDMEF is an IETF Internet Draft
  - Undergoes evaluation to become RFC

→ one format to store 'em all!

IDMEF Example

```
<IDMEF-Message>
  <Alert messageid="5086374041697">
    ...
    <CreateTime ntpstamp="0xc739ad2d.0xa4069000">
     2005-12-01T18:11:09.640725+01:00</CreateTime>
    ...
    <Source spoofed="unknown">
      <Node category="unknown">
        <Address category="unknown">
          <address>172.20.203.12</address>
        </Address>
      </Node>
      ...
    </Source>
    ...
  </Alert>
</IDMEF-Message>
```

The Prelude-IDS Framework (1)



Classification	Source	Target	Sensor	Time	
9 x OpenSSH	172.20.203.12:58354	172.20.251.89:22	snort1	14:26:01 - 2005-09-16 14:22:16	□
2 x File Added (succeeded) 48 x File Missing (succeeded) 16 x File Modified (succeeded) 3 x File Missing (succeeded)	n/a	/etc/snort/rules	samhain1	12:09:03 - 2005-09-19 13:06:00	□
4 x ICMP PING NMAP 74 x Killing attempted connection (failed) 3 x SCAN nmap XMAS 3 x SNMP AgentX/tcp request 3 x SNMP request tcp 3 x SNMP trap tcp 1 x (portscan) TCP Portscan	172.20.251.90:37904/tcp	172.20.251.118:80/tcp	honeyd (sensor1.amd.com) snort1	2005-09-20 14:16:53 - 2005-09-20 14:14:26	□
3 x OpenSSH 2 x ICMP PING NMAP 1 x SCAN nmap XMAS 2 x SNMP AgentX/tcp request 2 x SNMP request tcp 2 x SNMP trap tcp 1 x (portscan) TCP Portscan	172.20.251.90:56525	172.20.251.89:161	snort1	2005-09-20 14:14:17 - 2005-09-19 12:56:55	□
38 x Killing attempted connection (failed)	172.20.203.12:53005/tcp	172.20.251.118:80/tcp	honeyd (sensor1.amd.com)	2005-09-19 17:41:19 - 2005-09-19 17:40:02	□
User login successful (succeeded)	::ffff:172.20.251.101:59312/tcp	sensor1.amd.com:22/tcp 172.20.251.89:22/tcp user: root process: sshd (12318)	sshd (sensor1.amd.com)	2005-09-19 17:33:41	□
OpenSSH	172.20.251.101:59312	172.20.251.89:22	snort1	2005-09-19 17:33:39	□
2 x User authentication successful (succeeded)	user: root	sensor1.amd.com 172.20.251.89 user: root process: sshd (11535)	PAM (sensor1.amd.com)	2005-09-19 16:54:30 - 2005-09-19 16:32:37	□
User login successful	::ffff:172.20.203.12:45319/tcp	sensor1.amd.com:22/tcp 172.20.251.89:22/tcp	sshd	2005-09-19 16:54:30	□

The Prelude-IDS Framework (2)

- Already Prelude-enabled sensors:
  - Snort
  - Samhain
- Others:
  - Use Prelude-LML!
  - log file analyser (PCRE, map to IDMEF)
- Special cases:
  - Client-API in C, Python and Perl

Remaining problems...

- Distributed IDS sensors will report many events
  - Multiple sensors distributed all over
  - Different sensor technologies

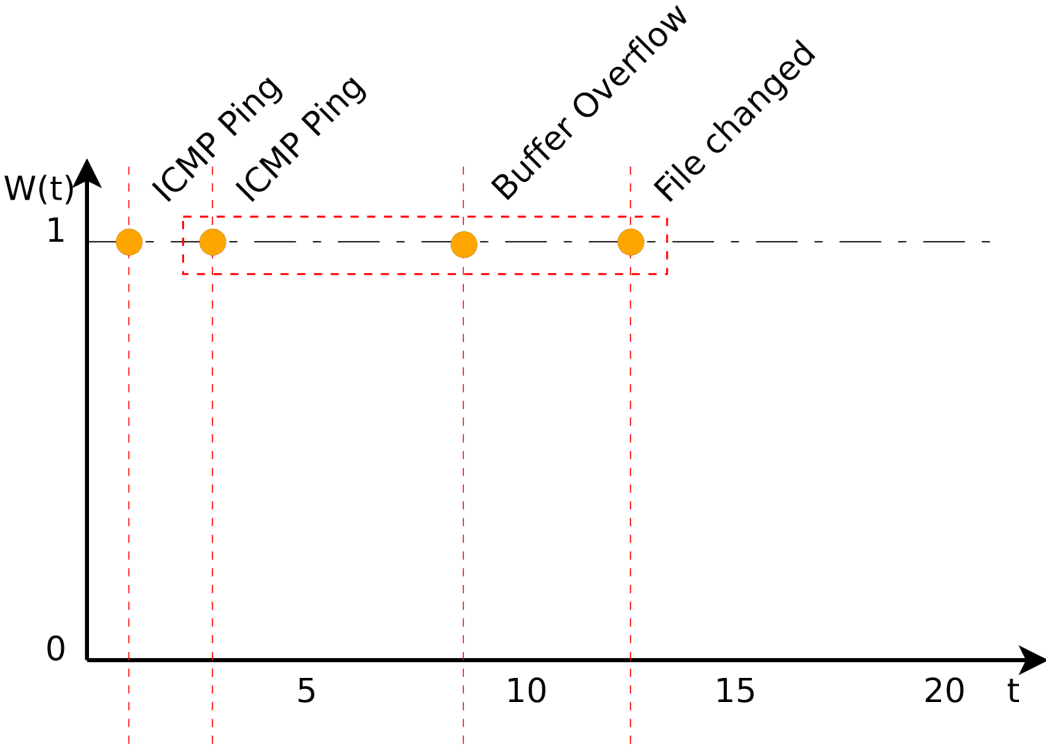
- Human admin unable to investigate every single event

- Single events don't give a reliable shape of an incident

To many events

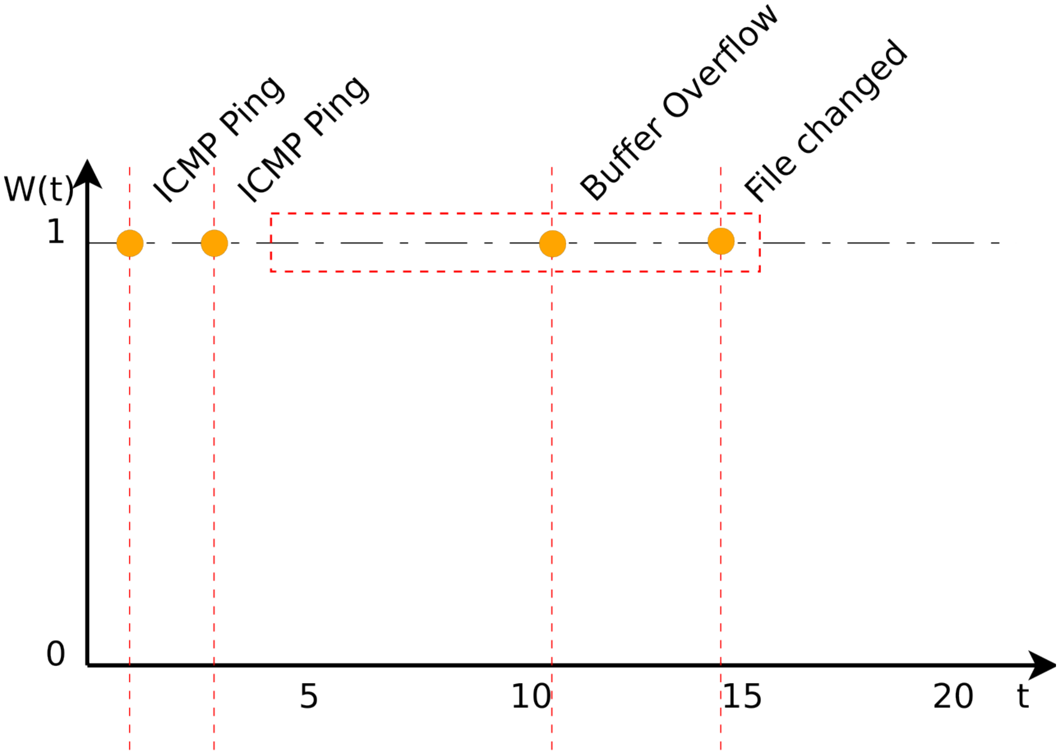
Basic correlation principle

- Events in a defined time window
- Define rule that matches timely appearance of events that could belong together
- Conjunction of events by AND



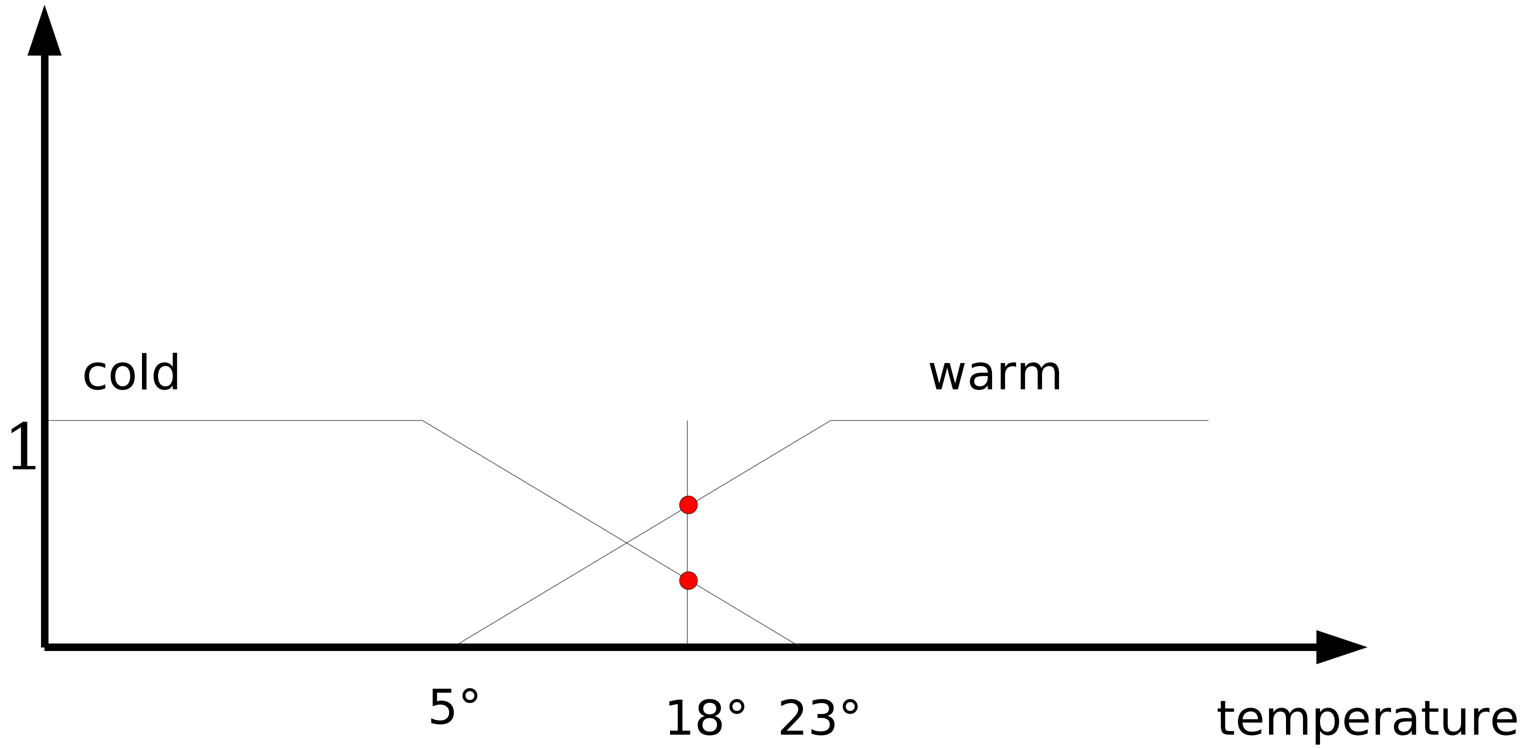
Problem: sharp rules

- Sharp rules too exact for dynamic behaviour
- One failure in rule -> wrong conclusion
- “Binary” conclusions are insufficient
- *Not the way one will investigate what has happened*

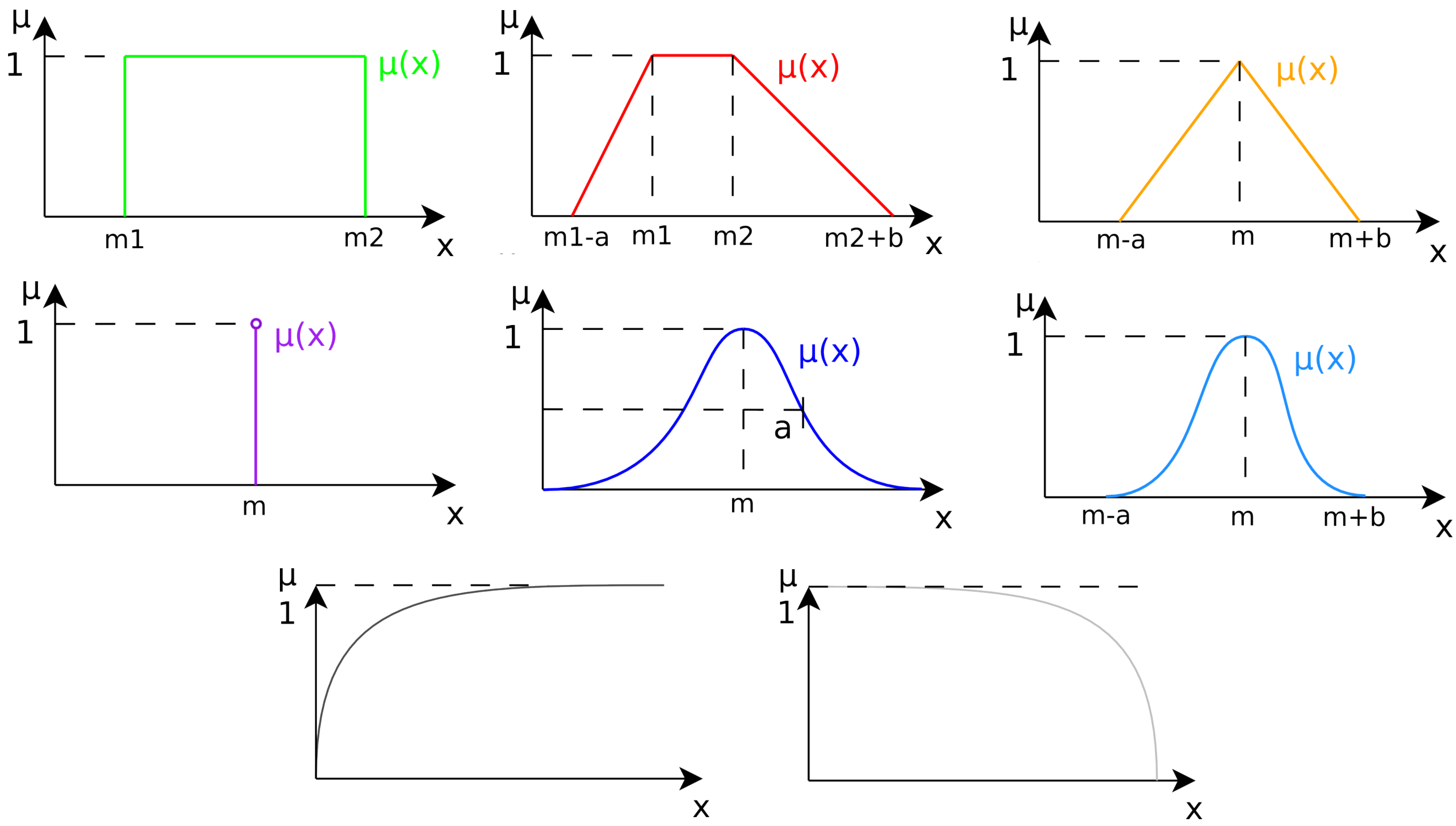


Short Fuzzy Set Intro

- Extension to classic sets
- Fuzzy [set|logic|control]
- Membership function

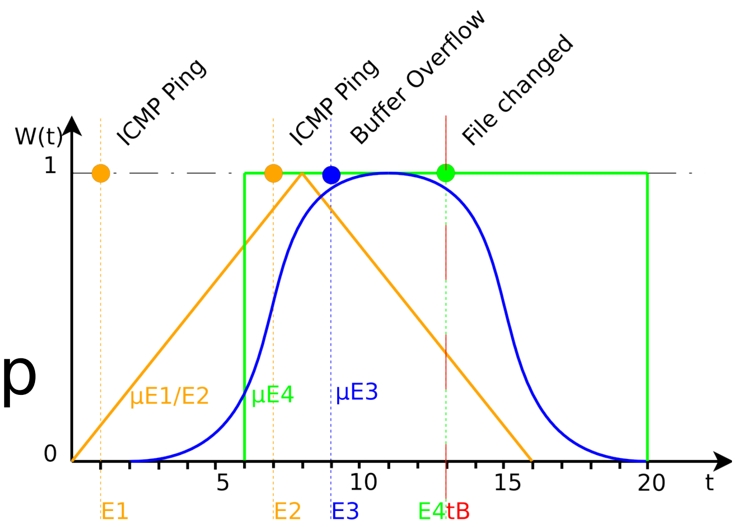


More Membership Functions



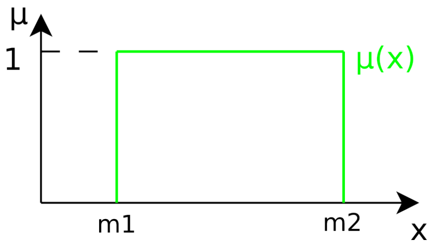
Applying Fuzzy Sets to IDSs

- Formulate a “Fuzzy-rule”, containing:
  - Events
  - Membership function w/ parameters
  - Limits, repetition function
- Evaluate the “Fuzzy-rule”
  - Search for matching events
  - Calculate grade of membership
- Correlation:
  - Membership grade -> probability values
  - Result: application of combination theory -> multiplication of membership grades



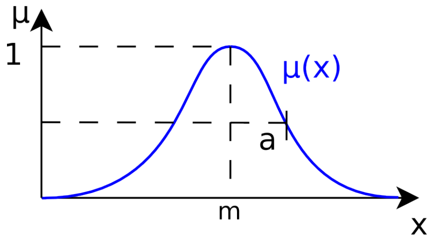
Simple Example: A basic Worm Attack

- File Changed



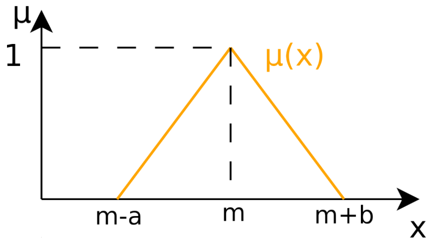
μE4(t4)=1.0

- Buffer Overflow

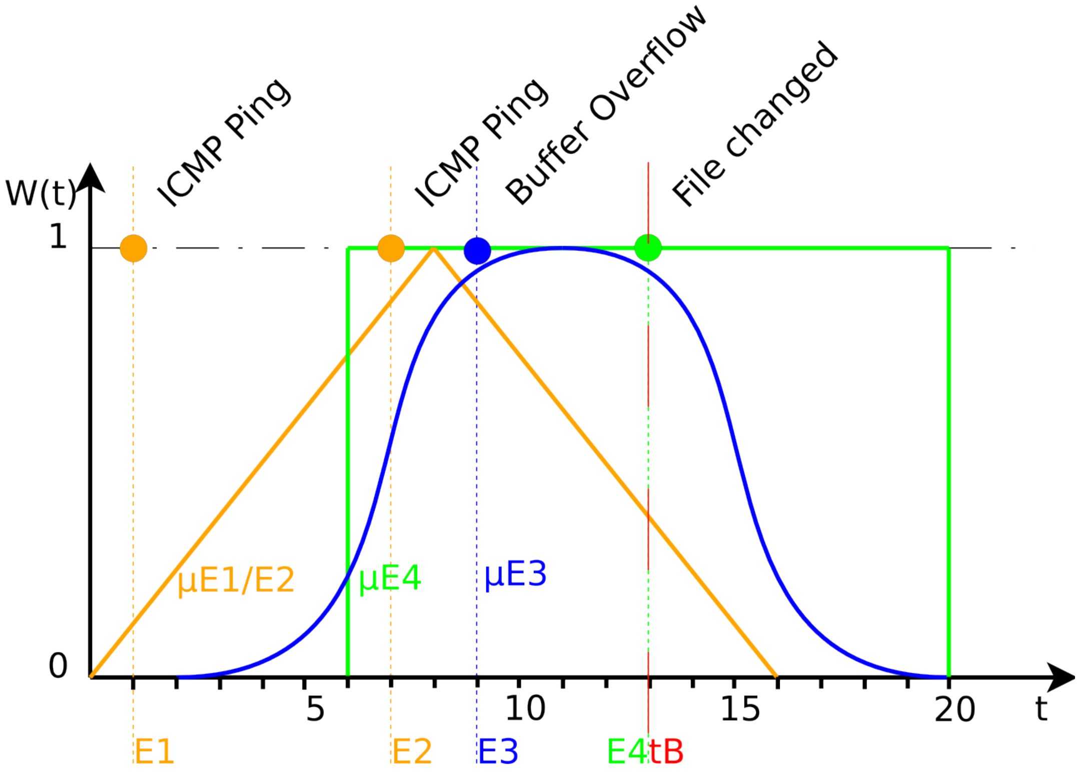


μE3(t3)=0.9

- ICMP Ping



μE2(t2)=0.8



Likelihood of the incident:

μ = μE4(t4) * μE3(t3) * μE2(t2)
μ = 1.0 * 0.9 * 0.8
μ = 0.72

Fuzzy IDS Evaluation

- Fuzzy rules help to improve correlation results
  - wider rule definitions -> wider range of results
  - sharper rule definitions -> more precise results
- Adjustable parameters
  - Stretch or compress membership functions
  - Rate quantity of events
- Implementation
  - Rule-based evaluation/correlation module for Prelude-IDS
  - Statistic analysis of intrusion attempts / report generation
  - Instant Messaging, level of escalation

Conclusion

- Use all the data sources you can get
- Use clever methods to *summarise*, *correlate* and *evaluate* the data
- Look at the reports