SIEMENS

Corporate Technology

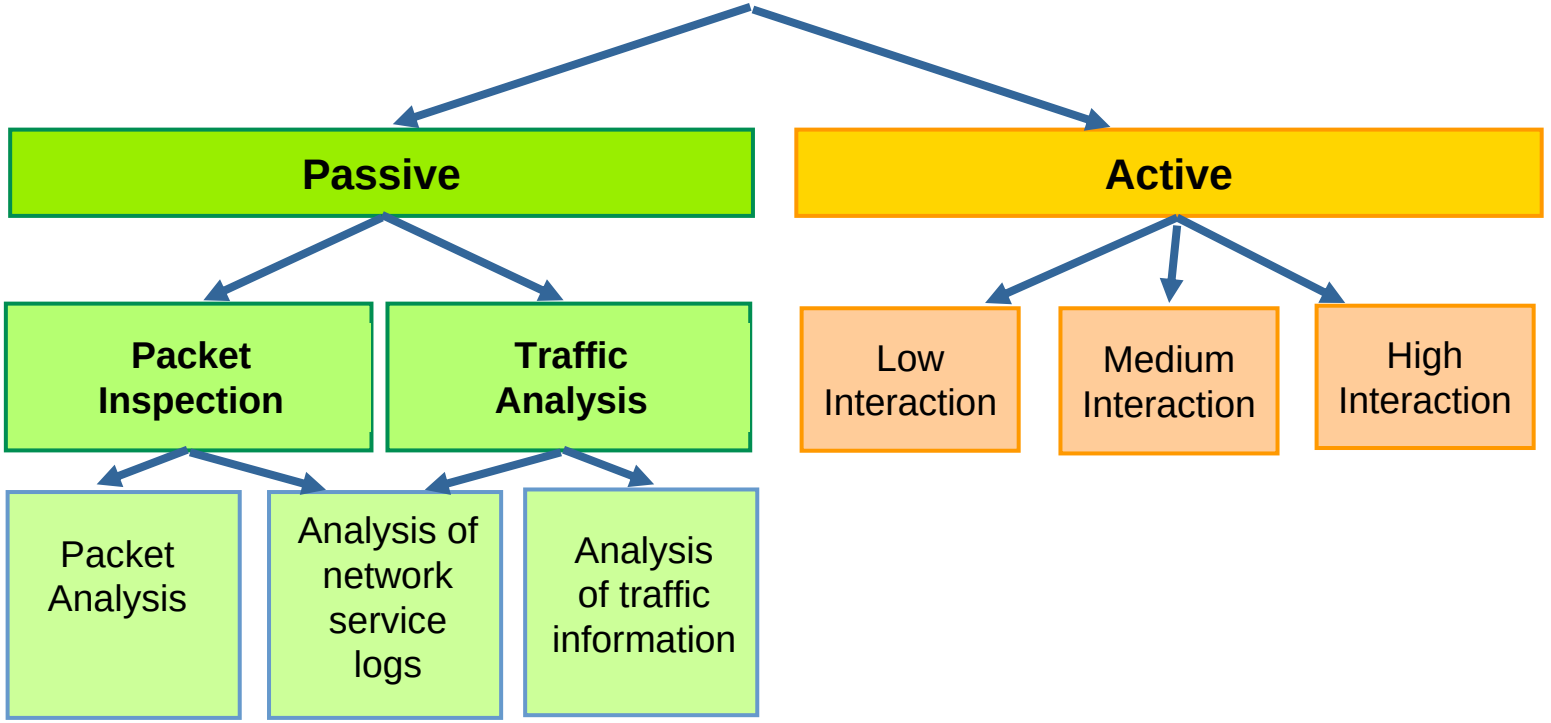
Experiences with Building, Deploying and Running a remote-controlled easily installable Network Sensor

Bernd Grobauer, Siemens CERT

Imagine the following situation?

- You have no control over the network,
  - managed largely by infrastructure provider
  - no possibility to collect data
- but want to know whether there is malware activity on the network
- All you do have is helpful IS contacts that may be able to tweak their local infrastructure a bit -- if you ask them nicely

Methods for Network-based Malware-Detection

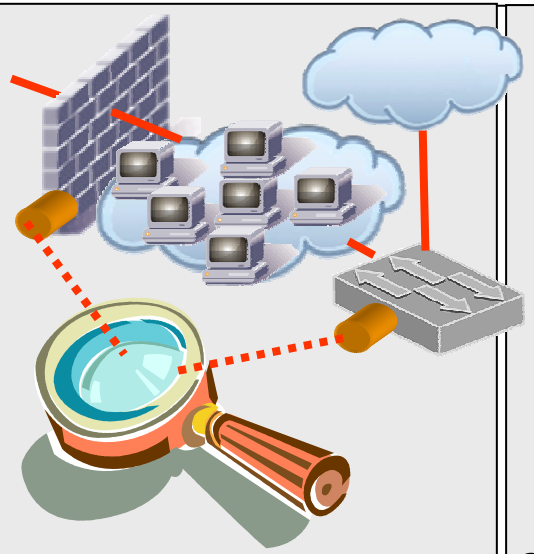


- **DNS**
- **Proxy**
  - **HTTP**
  - **…**
- **…**

- **Statistics**
- **Blocked**
  - **Router**
  - **Firewall**
- **Flows**
- **…**

Data Sources for Network-based Malware Detection:

Collector



Packet inspection and analysis of log files is traditionally used on traffic from/to the hosts to be protected
⇒ illegitimate traffic must be found within lots of legitimate traffic
⇒ attacks on / compromises of actual assets can be observed

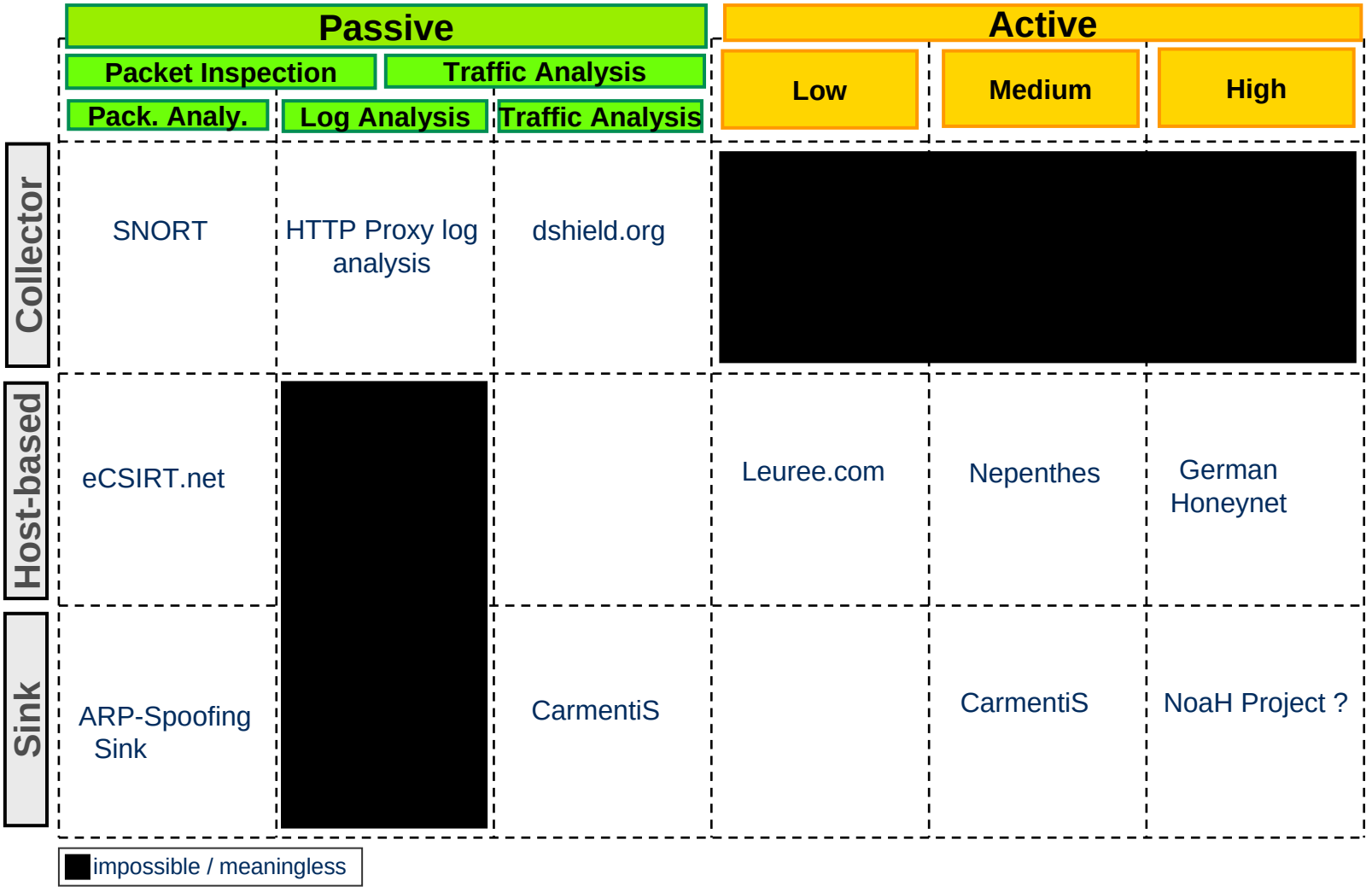
Host-based

Collecting data from a single host mostly useful for protecting that host, less so for learning about threats to the rest of the network
⇒ if host used as honey-pot, then all observed traffic is suspicious
⇒ only attacks on / compromises of single host can be observed

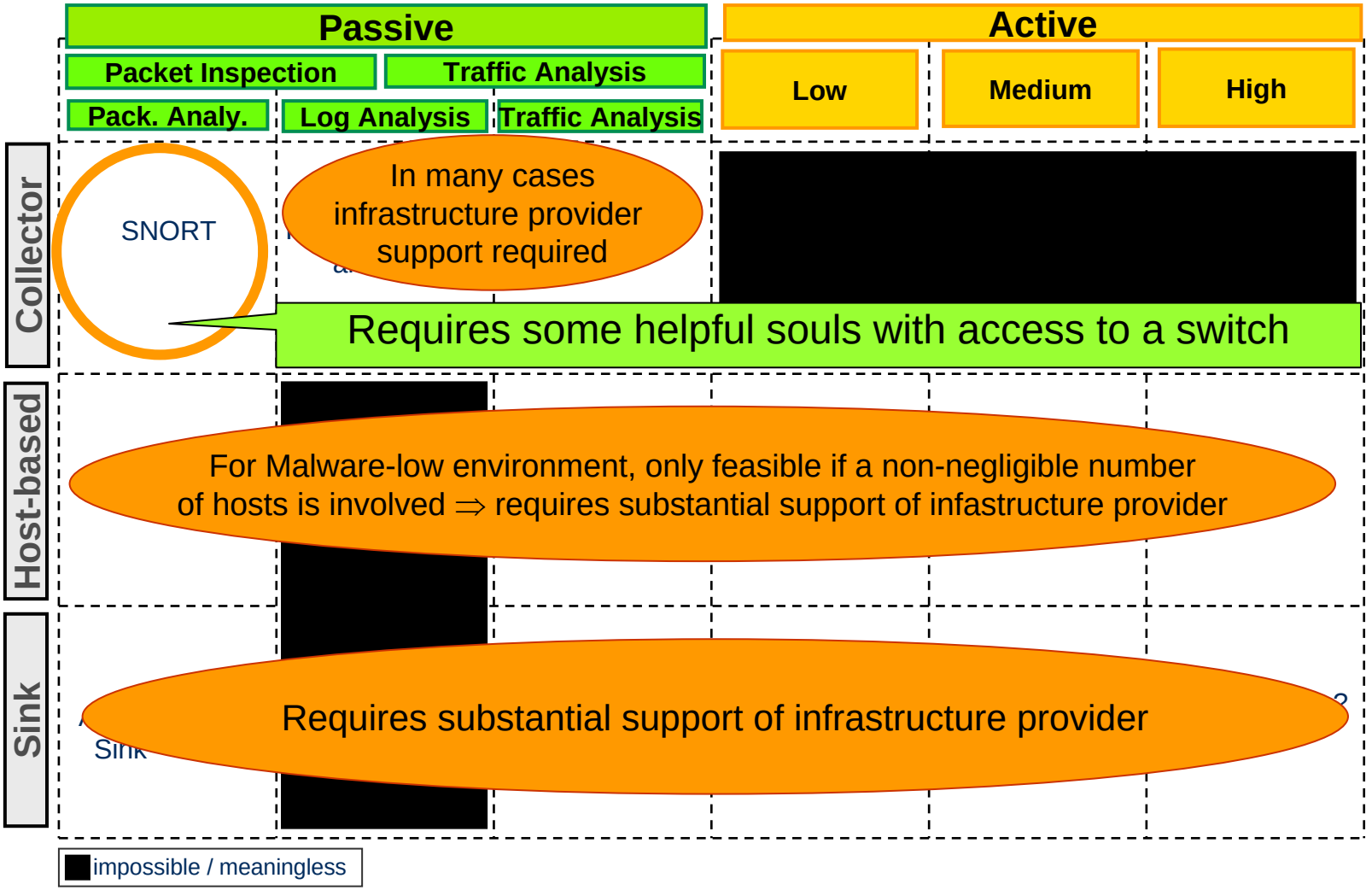
Network Sink

Network sink = routing configuration that directs traffic to unused/bogus IPs to a central location for monitoring purposes monitor traffic
⇒ all observed traffic is suspicious compromised
⇒ only already assets trying to contact IPs in network sink can be observed

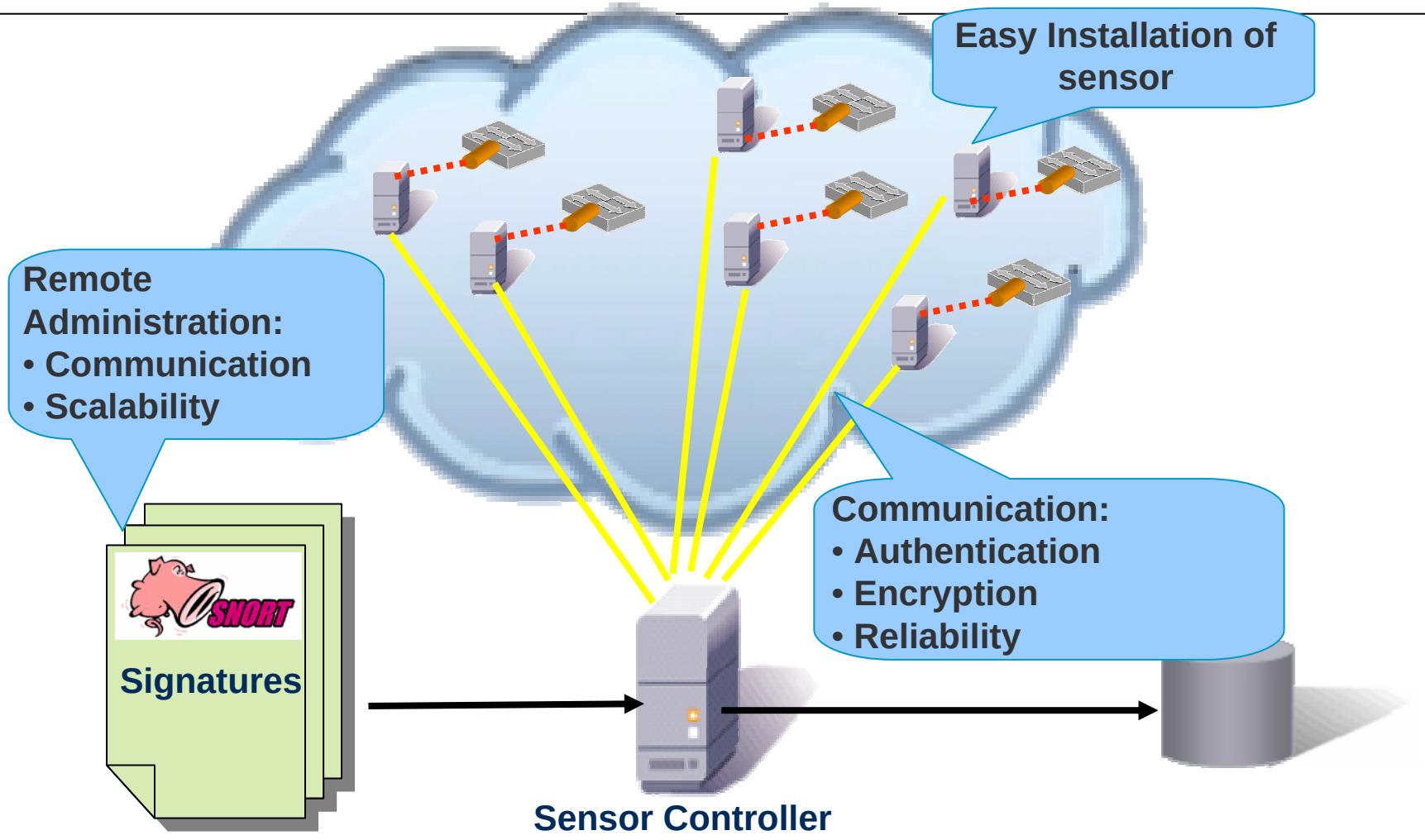
Combining Detection Methods and Data Sources



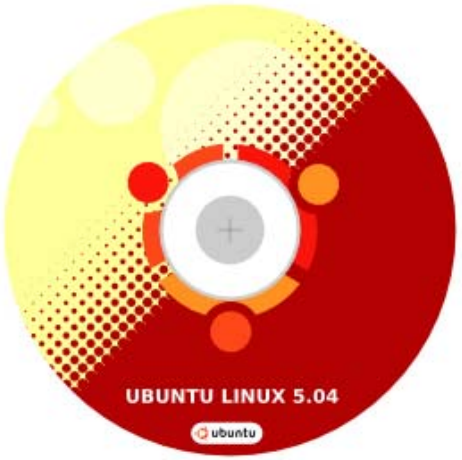
Combining Detection Methods and Data Sources



Remote-sensor Architecture and Requirements



Distributed Worm Sensors: Easy installation of sensor



Sensor based on Linux distribution "Ubuntu"
- Linux free & requires little resources
- Ubuntu offers easy mechanism for creating "Life CDs"

⇒Sensor can be created by setting up a single sensor and creating a life CD

Customization of sensor disk for each user via USB-stick, containing
- token for authentication
- configuration details (network settings, etc.)

Easy usage:
- download CD-image
- ask CERT for authentication token
- save token & config. on memorystick
- use any old PC as sensor

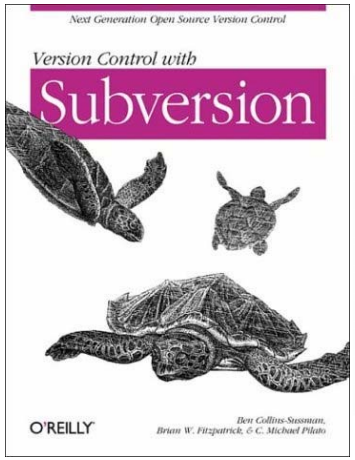
Distributed Worm Sensors: Communication between sensor and controller

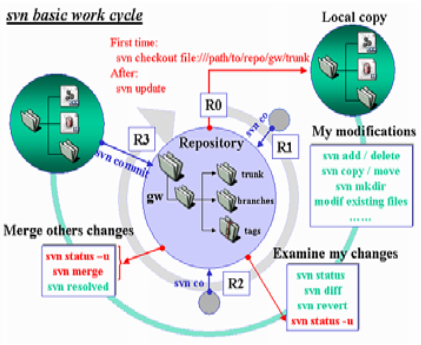
Prelude Framework

- Framework for communication between IDS sensors, IDS concentrators and IDS controller
- Client-server authentication using X.509 certificates
- Spooling functionality: during breakdown of connectivity, data is buffered
- Communication based on IDMEF standard for incident data

- Setup for Sensor Disk:
- Central controller runs Prelude manager
- Sensor Disk runs Prelude manager as concentrator for
  - local SNORT sensor
  - local host IDS monitoring log files
- Standard X.509 certificates can be used
  - as authentication tokens for sensors
  - to authenticate central server
- disruption in connectivity (network problems, maintenance of central server) no problem

Distributed Worm Sensors: Remote Administration (I)





Subversion version control system offers

- client-server model for centralized repository of text
- client-server authentication using X.509 certificates
- support for merging changes between related development branches

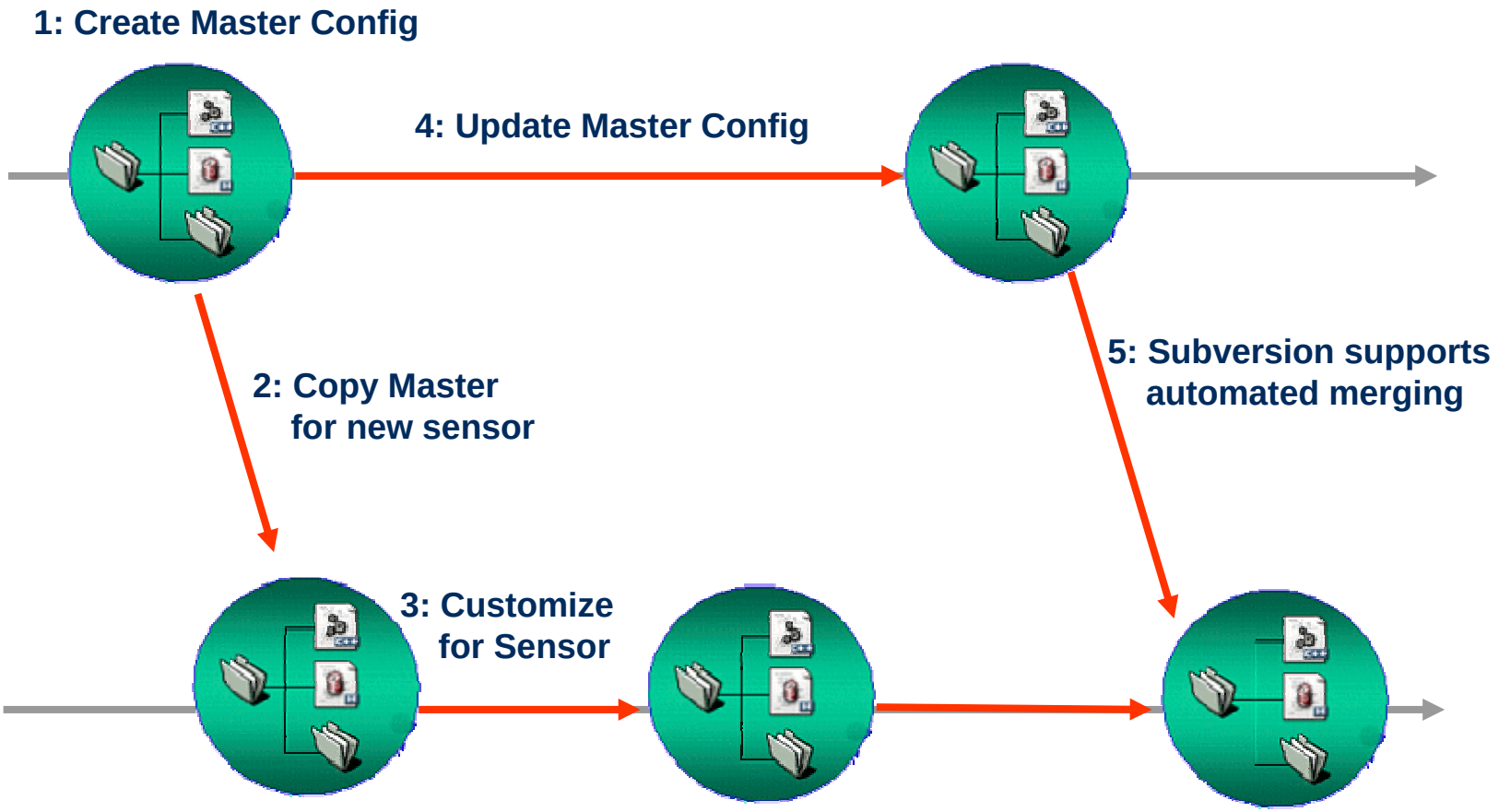
Setup for sensor disk:
Configuration for all sensors maintained within subversion repository
sensor connects to repository (using X.509 authentication token) and downloads configuration

- directly after startup
- regularly during operation

configuration maintenance scalable to many sensors:

- sensor configurations based on template
- changes in template can be merged into individual sensor configurations

Distributed Worm Sensors: Remote Administration (II): Scalability



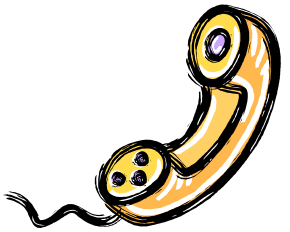
Sensor Installation

- Remote sensor administrator fills in configuration file (template distributed together with sensor certificate)
- Certificate is password protected; password must be entered during boot
- Upon request, boot dialogue helps with identifying right network interfaces (administrator plugs cable and dialogue provides feedback)
- If there is network connectivity, the sensor contacts the central controller, downloads the current IDS configuration and starts sniffing.

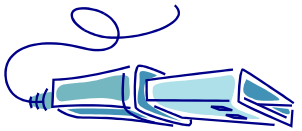
```
[controller_link]
iface = <interface>
mode = static
address = <sensor-IP>
netmask = <netmask>
gateway = <gateway IP>
nameserver = <DNS-server IP>

[monitored_link]
iface = <interface>

[debug]
ssh_access=<IP of central controller>
```

Lessons learned: Installation

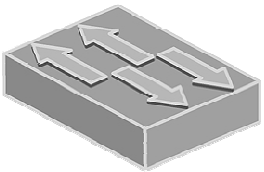
Installation per Life-CD works pretty well



- Boot menu should provide clear problem descriptions and allow retries without forcing a reboot
  ⇒ easier diagnosis/support per telephone mail



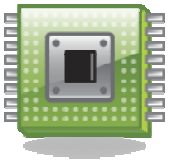
- Support for finding „right“ network device very helpful



- Main installation problem: switch configuration
  - sensor sees nothing
  - sensor sees too little / wrong network

  ⇒ useful enhancement: reporting of IPs that are seen by sensor for debugging purposes

Lessons learned: Stability



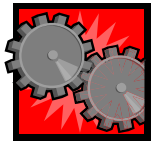
- Biggest stability problem: power outages (esp. in non-European countries)
- At the beginning, needed to tinker with parameters for log rotation / log deletion to avoid running out of memory
- Helpful: script reporting daily on
  - vanished sensors (usually due to power outage)
  - changes in amount of traffic that is monitored (switch reconfigured? , cable unplugged?)
  - (at the beginning): information about free memory

  Information about monitored traffic and memory are sent to sensor controller with sensor heartbeat

Lessons learned: IDS operation

- Also old Pentium III boxes can monitor large amounts of traffic if restricted to most relevant patterns for detecting network worms

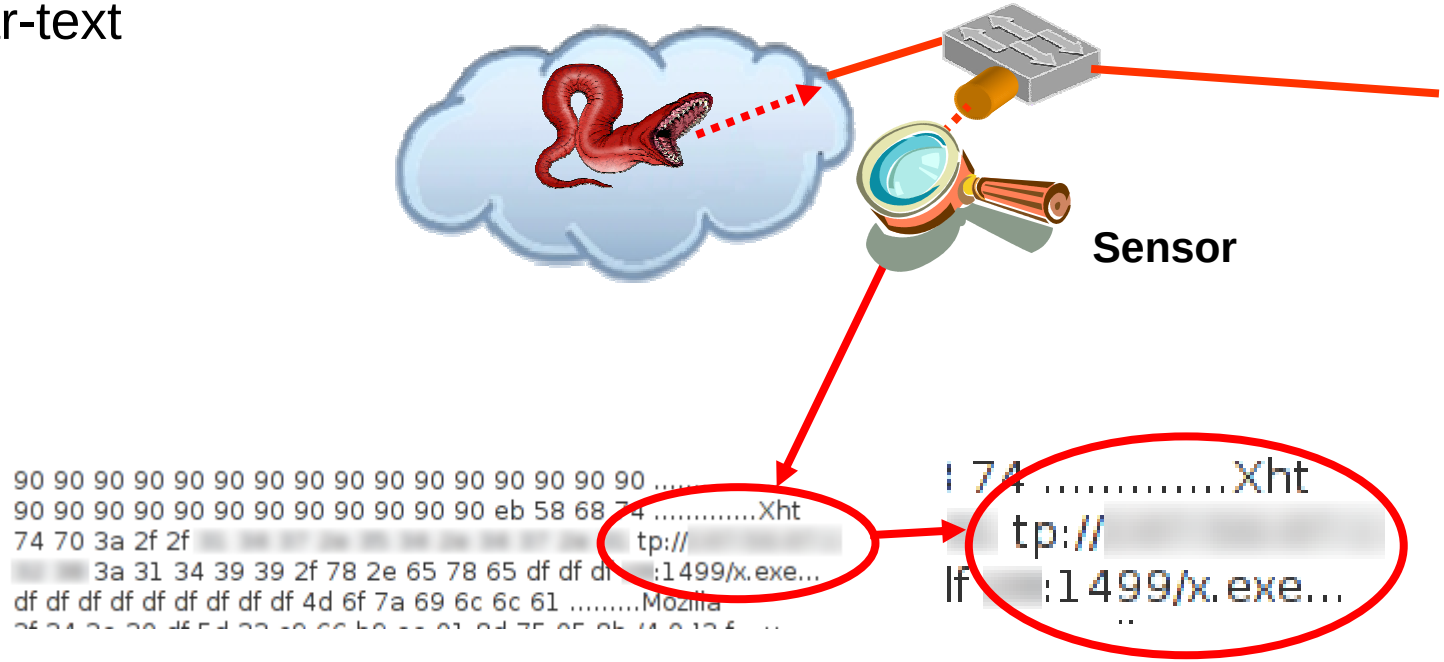
- Dedicated sensors that are able to monitor traffic between web proxy and internal hosts can be used to watch for malicious drive-by-downloads, spyware activity, etc.



- Useful: script on sensor controler checks incident reports every 10 minutes; if new attacking IPs have been reported, NetBios-information is requested from these IPs so as to aide incident handling

PS: A fun thing to do: Catching malware specimen

- Some shellcode shows the download URL for the malware binary in clear-text



- Write a script on sensor controller to analyse contents of detected packets (contained in Prelude-message) and download binaries with mget ⇒ keep your malware analysis folks happy ☺