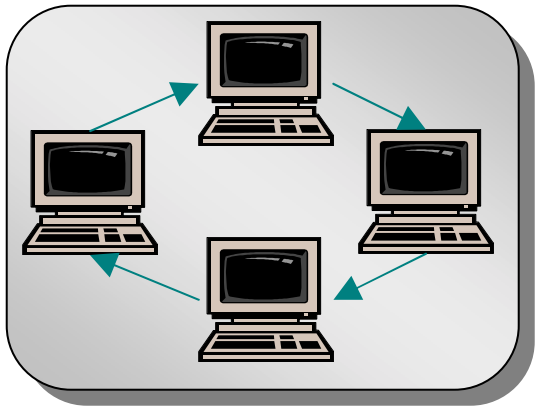


2003 IEEE International Workshop on Information Assurance (IWIA 2003)
March 24, 2003

Intrusion Detection Force: An Infrastructure For Internet-Scale Intrusion Detection

Lawrence Teo[1,2] Yuliang Zheng[1,2] Gail-Joon Ahn[1]

[1]Laboratory of Information Integration, Security, and Privacy (LIISP)
University of North Carolina at Charlotte,
Charlotte, NC, USA

[2]Calyptix Security Corporation
Charlotte, NC, USA

{lcteo,yzheng,gahn}@uncc.edu

The Problem

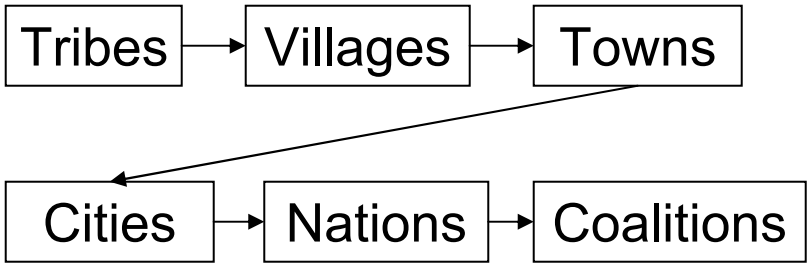
- Current Intrusion Detection Systems (IDSs)
  - **Deployed separately**
  - **Do not communicate** with one another
  - Do not communicate with other security products
- **Sophisticated** and **large-scale attacks** are becoming increasingly common
  - SQL Slammer
- **Potential** of current IDSs is becoming **limited**

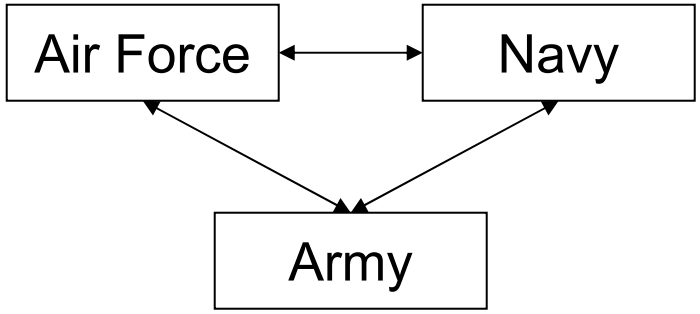
Information Sharing

- Organizations will be able to **talk to one another**
- Share information on **ongoing** and **early attacks**
- Take **proactive measures** before the attacks occur
- A **more intelligent way** to detect and respond to sophisticated attacks

Intrusion Detection Force (IDF)

- An **Internet-Scale** Intrusion Detection Force (IDF)
  - Information sharing among organizations
  - **Network data**, *not* proprietary data
- Detect and reduce the impact of
  - Distributed denial of service attacks
  - Internet worms, such as **SQL Slammer**
  - Evasive attacker activity
  - Abuse of systems as launchpads





Outline

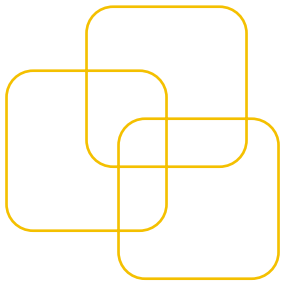
- Requirements of the IDF
- Architecture and Design
- Major Components
- Applications
- Current Tests
- Future Directions

Requirements of the IDF

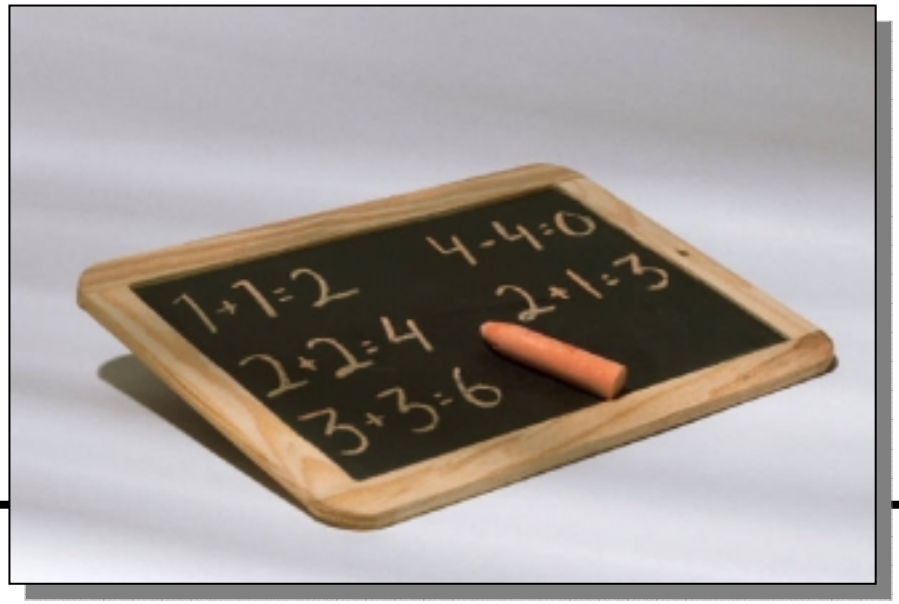
Information Sharing ▪ "Glue" of the IDF ▪ Inter-organizational info sharing ▪ Network data, not proprietary ▪ Privacy	**Scalability** ▪ Scale to millions of hosts ▪ Gradual rollout ▪ Planning to scale
Security ▪ Hostile, non-trusting environment ▪ Unreliable underlying network ▪ Confidentiality ▪ Integrity ▪ Availability	**Survivability** ▪ Capability of system to fulfill mission in a timely manner ▪ Unreliable underlying network ▪ Application-level fault tolerance ▪ Existing system-level fault tolerance mechanisms

Other Requirements

- Interoperability
  - Support for **heterogeneous** environments
- Extensibility
  - Design **new potential applications**
- Balance between usability and security
  - Usable for both **novice and advanced users**



Architecture & Design



Architecture and Design

- How do we fulfill the requirements?
- Seven design decisions
- Hierarchical model
- Entities (Basic building blocks)

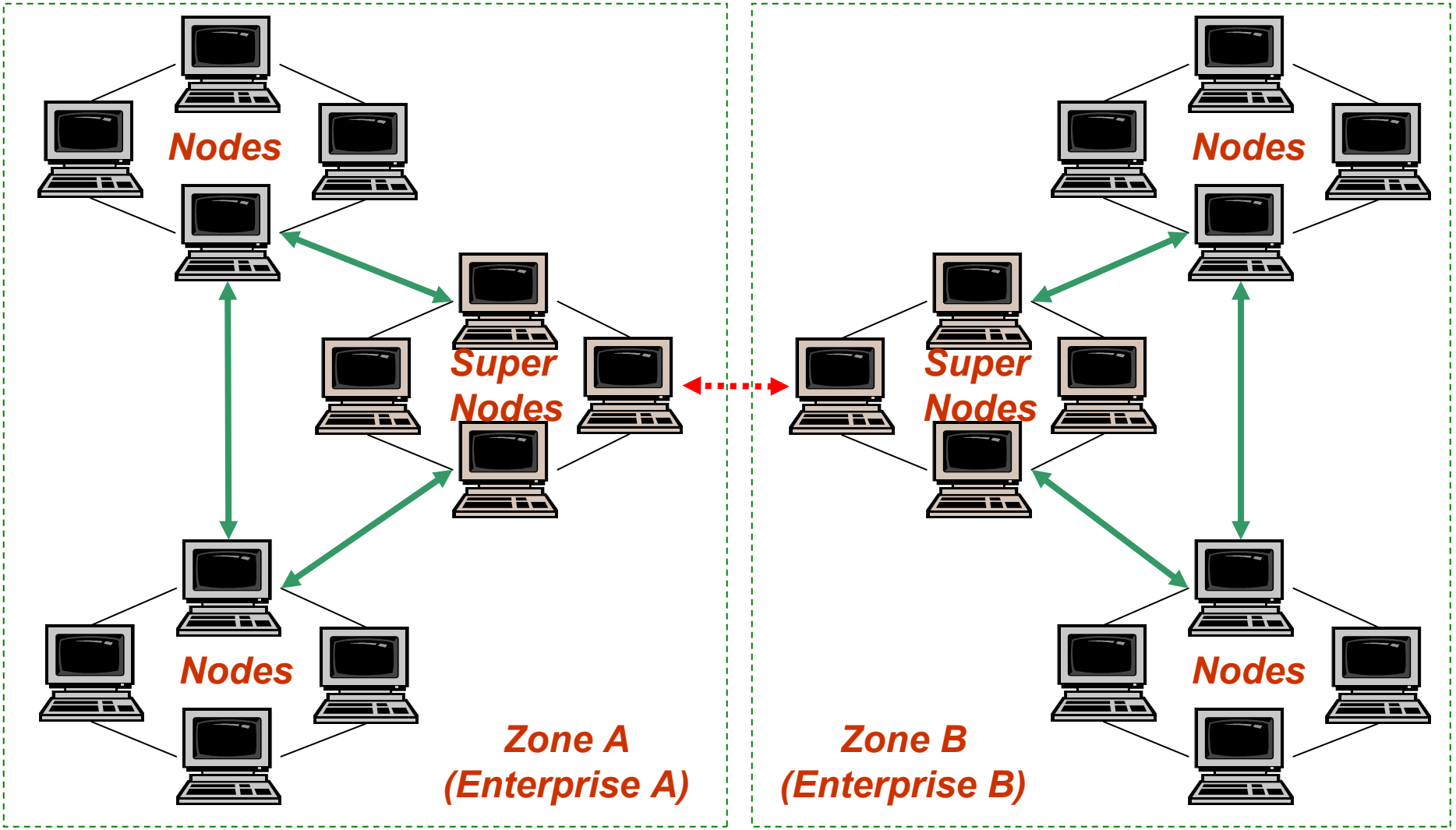
Seven Design Decisions

1. Real-time detection
2. Active/passive response
3. Audit sources from both host and network
4. High degree of interoperability
5. Distributed data collection
6. Distributed data analysis
7. High security on IDF itself

IDF Hierarchical Model

- Two levels
- Simple
  - Only two types of entities to implement
  - Keep it as simple as possible, because the IDF is going to be very large
- By allowing the entities to link to each other, we can achieve scalability

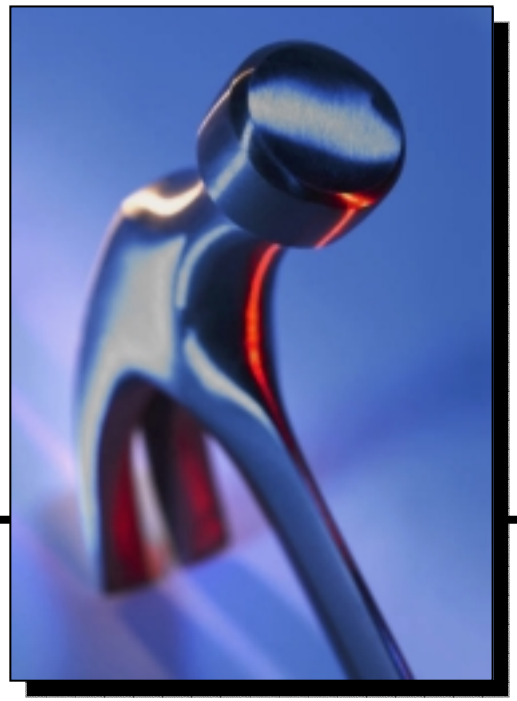
:::IDF Architecture



IDF Entities

- Node
  - Host running an IDF agent
- Collective
  - Collection of nodes. Nodes in a collective forward information to each other
- Supernode
  - Special node that provides higher-level services to collectives
- Super-collective
  - Collection of supernodes
- Zone
  - Area of the IDF under the authority of the supernode collective

Major Components



Eight Major Components

Foundation

1. IDF Adaptation Layer
2. Communication and Recovery Subsystem

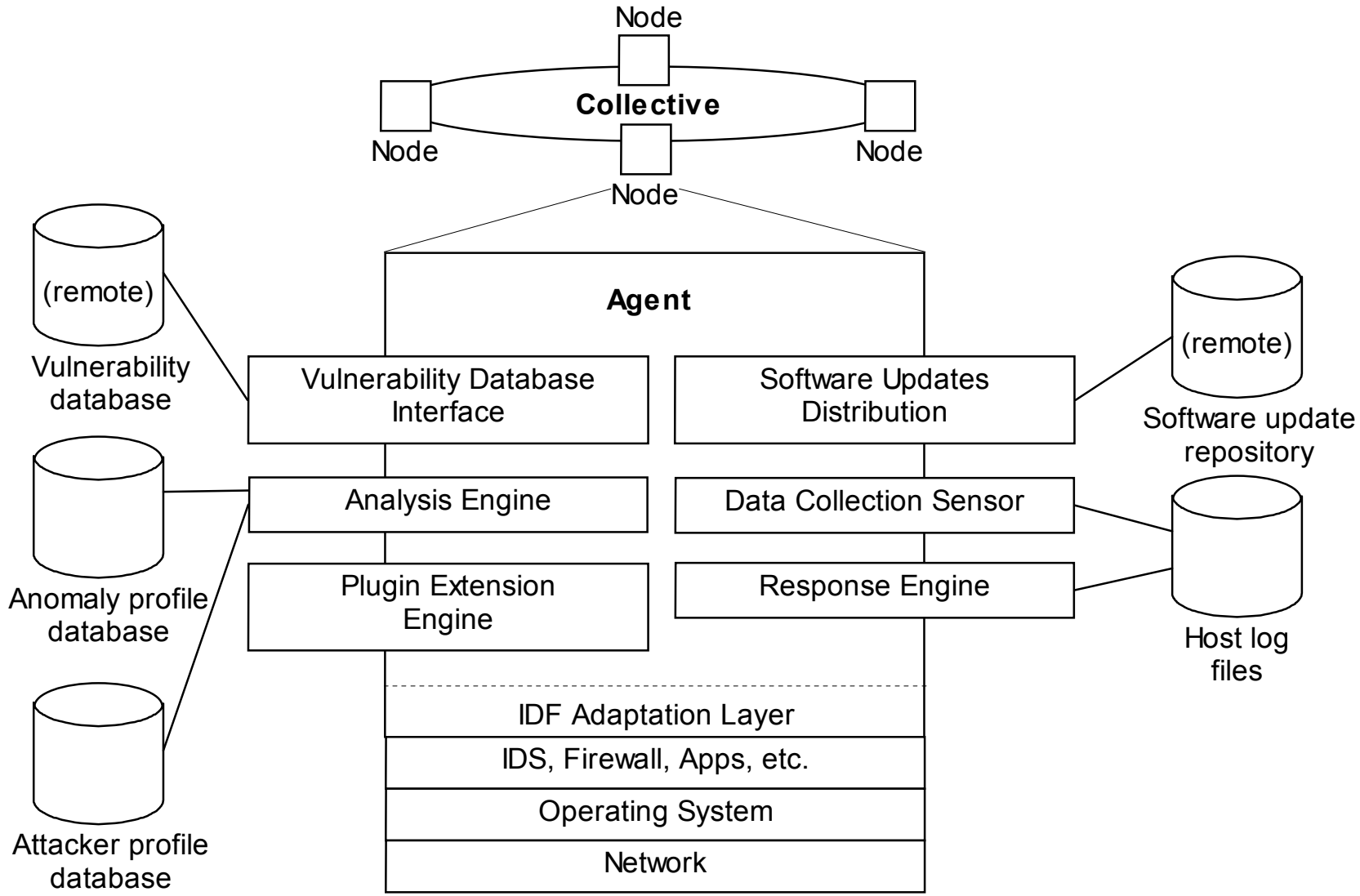
Core Components

3. Data Collection Sensor
4. Analysis Engine
5. Response Engine

Other Components

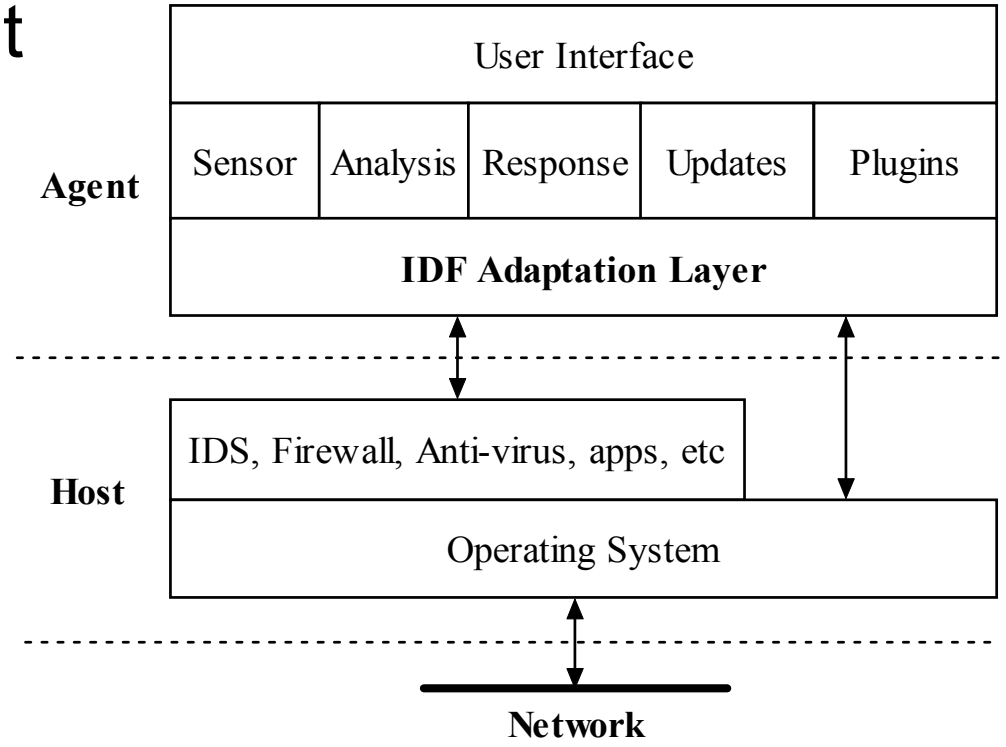
6. Vulnerability Database Interface
7. Software Updates Distribution Engine
8. Plugin Extension Engine

IDF Components



IDF Adaptation Layer (IDFAL)

- Integration component
- Interfaces with existing security technologies
  - Firewalls, IDSs, etc.
- Translates platform-specific details for higher-level IDF services



Communication and Recovery Subsystem

- Communication at the node level
  - Node-to-node, node-to-supernode, supernode-to-supernode, supernode-to-node
  - Node addressing and routing, congestion control
  - Registration of node with supernode
- Communication at the collective level
  - Node additions and removals, and node states
- Recovery
  - Data replication
  - Collective reconstruction

Core Components

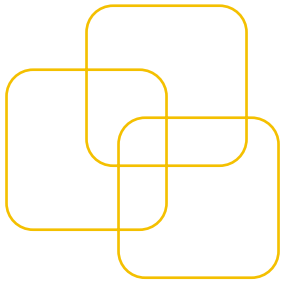
- Data Collection Sensor
  - Gather events, wrap around IDSs
- Analysis Engine
  - Multiple analysis techniques to identify suspicious trends across zones
  - Audit reduction
- Response Engine
  - Generic interface to actual response mechanism
  - Countermeasures and proactive measures

Other Components

- Vulnerability Database Interface (VDI)
  - Interface to IDF and public vulnerability databases
  - Up-to-date information for software updates and analysis engine
- Software Updates Distribution Engine
  - Preemptive fix to vulnerabilities before they are exploited
- Plugin Extension Engine
  - Allow third-party plugins
  - Plugins need to be authenticated

Applications

- Internet-scale intrusion detection
  - Large-scale distributed intrusion detection
- Proactive intrusion prevention
  - Preemptively set up defenses to prevent intrusions
- Policy enforcement
  - Nodes and zones facilitate policy enforcement
- Trust management
  - Trust management among nodes
- Incident handling
  - Assist law enforcement with results previously not attainable



Tests

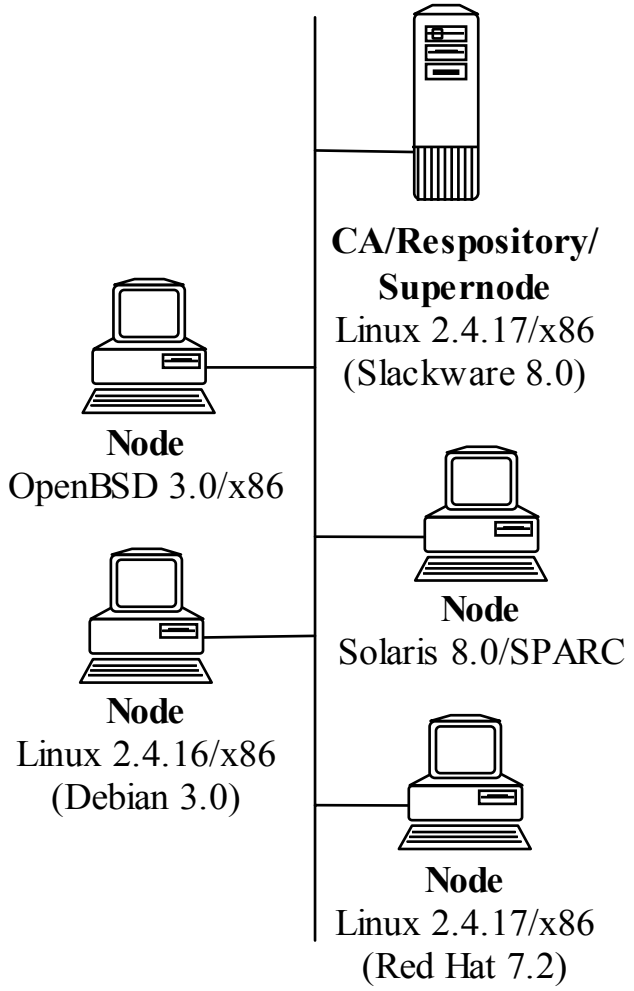
Current Prototype

Software updates distribution engine: Small, fast to prototype, self-contained

Test requirements	Software updates distribution engine
Interoperability	Developing software updates in platform independent manner is not difficult
Basic node-supernode communication	Easy to define roles of nodes and supernodes
Response times	Can measure delivery and update times

Performing the Test

- Register nodes with supernodes
  - Issue signed certificates beforehand
  - Register node configuration with supernodes in XML format via SSL
- Make updates available on repositories
- Once matching config is found, deliver the update to the node and apply it



Results

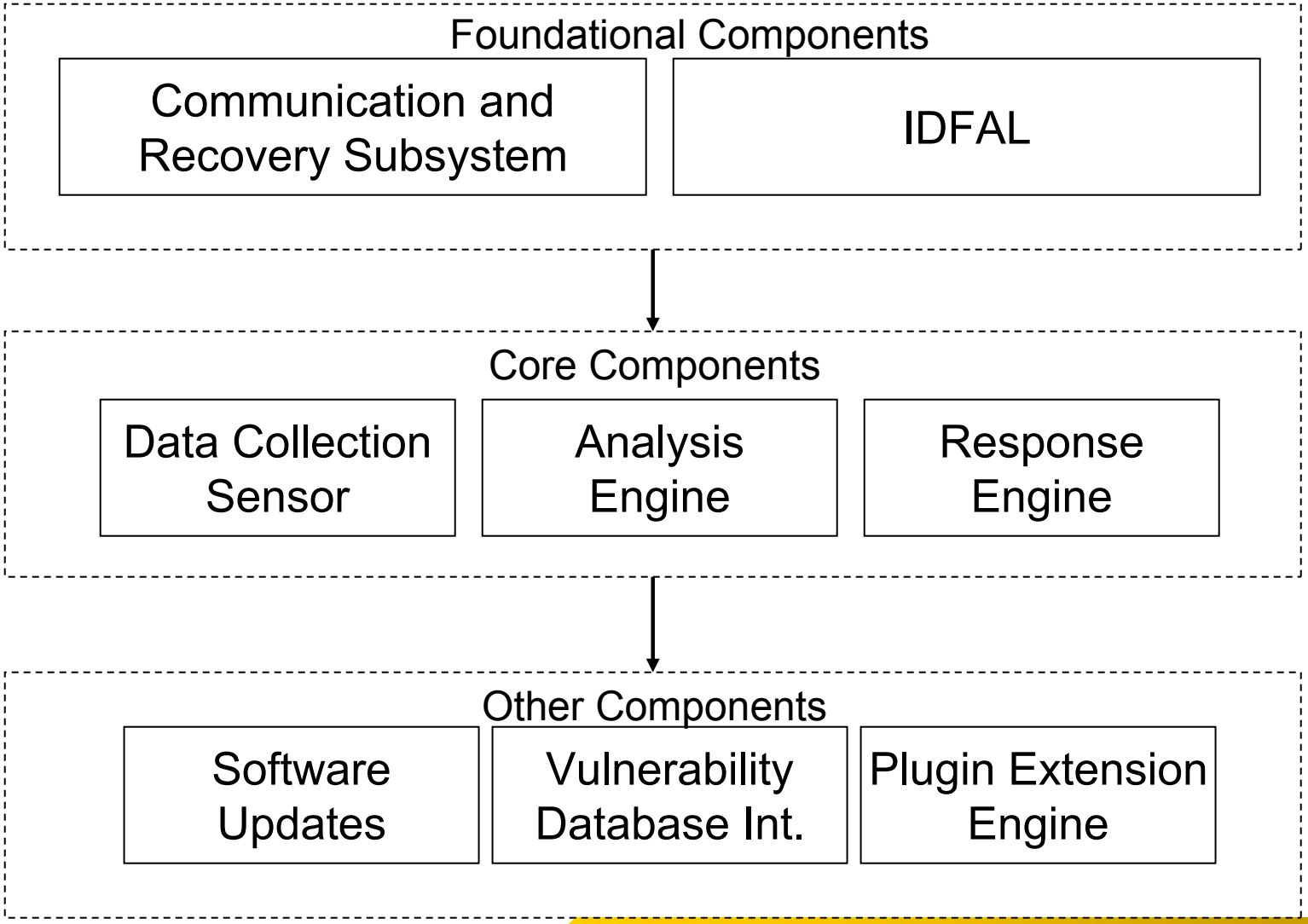
Update	Method	CPU	Size	S1	S2
sudo 1.6.5	Custom	400MHz	71.2KB	0.06s	0.06s
pine 4.44	rpm	166MHz	2632KB	1.55s	19.76s
wu-ftpd 2.6.1	dpkg	700MHz	250KB	0.24s	4.1s

- S1 = Download speed
- S2 = Installation speed
  - Including integrity check

Future Directions

- Design of communication and recovery protocols
- Privacy mechanisms
- Interoperability
  - Upgrading test systems
  - Adding new machine architectures, firewalls, IDSs, etc.
- Scalability
  - Simulation of collectives and super-collectives with virtual machines
- Extending into wireless area

Development Plan



Related Work

- Distributed Intrusion Detection Systems
  - AAFID, GrIDS, DIDS, CSM
- Internet-Scale Operating Systems
  - Chord, OceanStore, Tapestry
- Peer-to-Peer Systems
  - Gnutella, distributed.net, Freenet

Discussion

- Information sharing
- Privacy
- Legislation
- Achieving survivability is a difficult challenge
  - One method: use out-of-band non-Internet-based communication methods

Other IDF-related Publications

- L. Teo, Y. Zheng: **"Secure and Automated Software Updates Across Organizational Boundaries"**, *Proceedings of the 2002 IEEE Workshop on Information Assurance*, West Point, NY, June 17-19, 2002, pp. 212-219.

- L. Teo, G. Ahn, Y. Zheng: **"Dynamic and Risk-Aware Network Access Management"**, *Proceedings of the 8th ACM Symposium on Access Control Methods and Technologies (SACMAT 2003)*, June 1-4, 2003, Como, Italy. (To appear)

Conclusion

- Information sharing is **critical** for future security systems
- **Intrusion Detection Force (IDF)** is our ambitious project to **enable Internet-scale intrusion detection** and **response**

Questions?