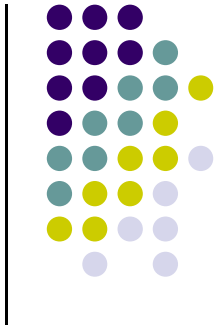
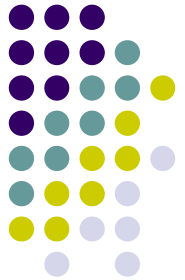


Attack Evaluation and Mitigation Framework

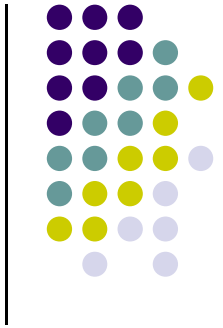
Laura Gheorghe, Răzvan Rughiniş, Nicolae Ţăpuş
Politehnica University of Bucharest, Romania
laura.gheorghe@cs.pub.ro, razvan.rughinis@cs.pub.ro, ntapus@cs.pub.ro

Context



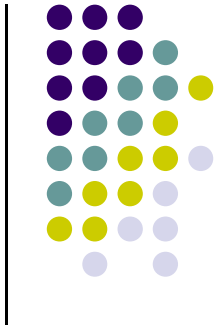
- Increase in frequency and severity of network-based attacks
- Intrusions and targeted attacks may result in the loss of:
  - critical data
  - business availability
  - time
  - reputation
  - money
- Firewalls are not always efficient against intrusion attempts
- The solution: intrusion detection and prevention systems

Intrusion detection and prevention systems



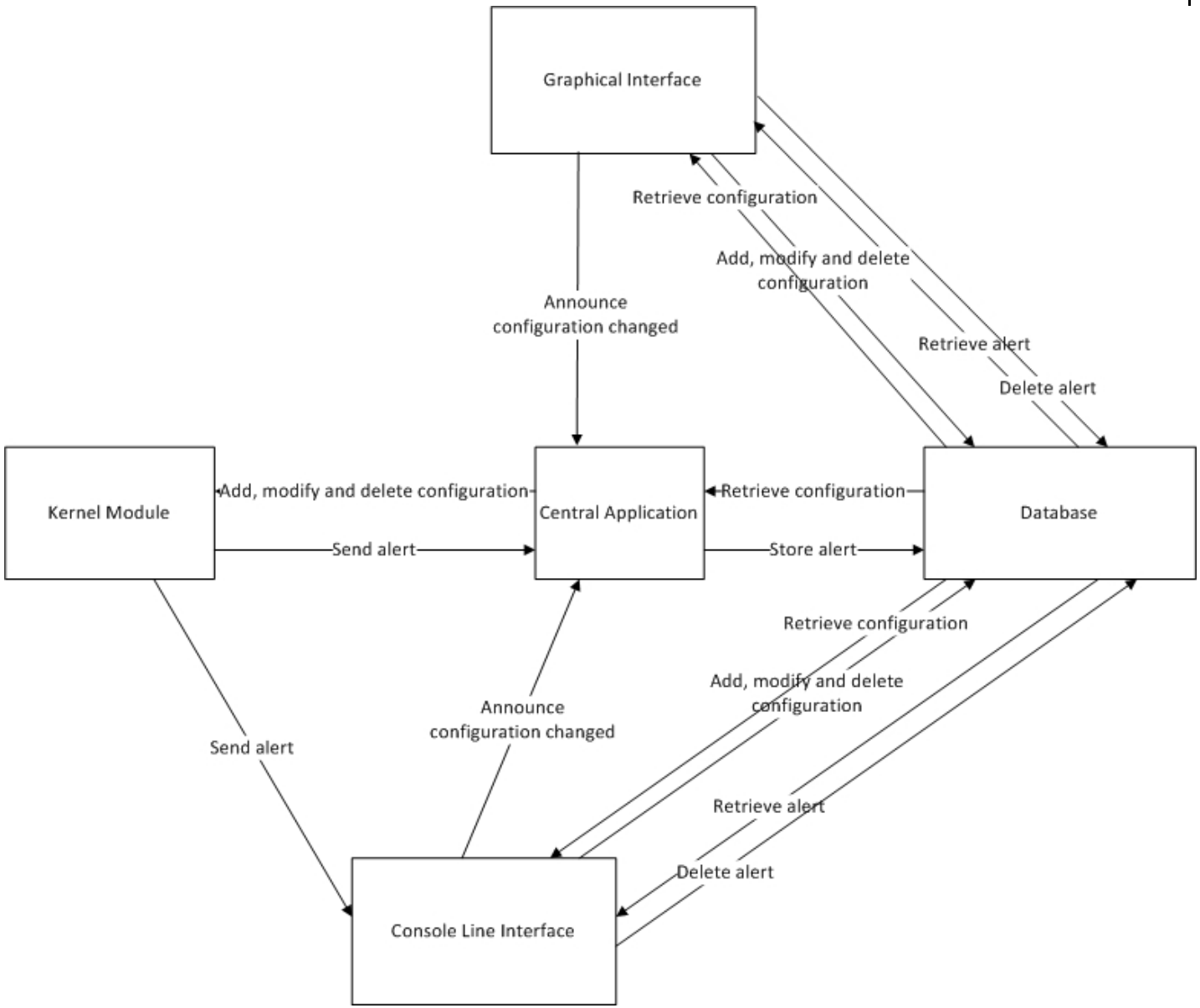
- Intrusion: the attempt to compromise the confidentiality, integrity and availability of a resource
- Detection: the process of monitoring system or network events in order to detect intrusions
- Prevention: the attempt to stop the intrusion from happening
- Two categories of IDPS depending on the location:
  - Host-based IDPS
  - Network-based IDPS
- Two types of intrusion detection:
  - Anomaly-based intrusion detection
  - Signature-based intrusion detection

Types of signatures and attacks

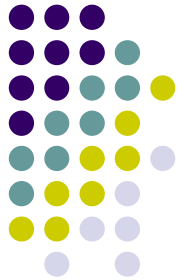


- Signature - description of network traffic generated by attackers
- Atomic signatures
  - Conditions for a single packet
  - Minimal resources
  - When the context is not important
- Flood signatures
  - Denial of Service attacks
  - Host-based: Traffic directed at one specific host
  - Network-based: Traffic directed at an entire network
- Sweep signatures
  - Reconnaissance attacks
  - Multiple connections to:
    - multiple hosts
    - multiple ports on a single host

System architecture

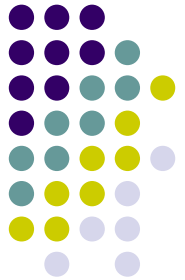


Central application



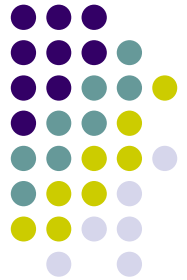
- Role: keep the sensor configuration up to date
- Loads configuration:
  - Retrieves sensor configuration from the database
  - Sends configuration to the kernel module
- Updates configuration:
  - Receives messages from the interfaces if configuration has been changed
  - Sends changes to the kernel module
- Stores alerts:
  - Receives alert messages from the kernel module
  - Stores alerts in the database

Kernel Module



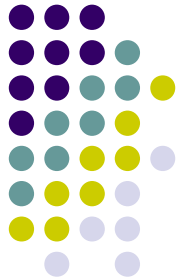
- A chain of five software components called engines
  - two firewall-based engines
  - three signature-based engines
- Packet analyzed by each engine in a sequential order
- Analysis result:
  - Packet blocking
  - Send packet to the next engine
- If the packet is not blocked the engines, it will be forwarded

Firewall-based engines



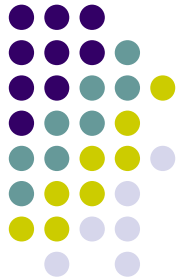
- Stateless firewall engine
  - Two lists of rules: static and dynamic, checked sequentially
  - A rule contains:
    - the conditions to be matched
    - action
      - Block packet
      - Send to next engine
- Stateful inspection engine
  - Valid connections
    - initiated from a host in the internal network
    - the destination is defined as a public server
  - monitors the TCP and UDP connections by maintaing their state

Signature-based engines



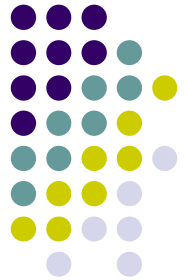
- Every engine stores a list of signatures
- Atomic engine
  - Match between the packet and a signature
- Flood engine
  - Analyses and stores packets that match a signature
  - Packets stored in a list with count, peak and gap values
  - Packets per second counter
- Sweep engine
  - Analyses and stores packets that match a signature
  - Packets stored in a list with associated count value
  - Maximum number of packets per time period

Graphical Interface



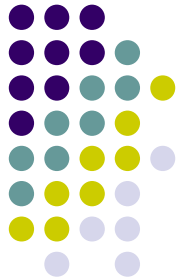
- Enable/disable engine
- Alerts can be visualized
- Signature-based engines
  - Signature tables can be displayed
  - Add and modify signatures
  - Delete signatures
- Stateless firewall engine
  - Display, add, modify, delete firewall rules
- Stateful firewall engine
  - Display, add, modify, delete internal networks and servers
- Announce changes to the central application

Command line interface



- Organized on levels, for example:
  - Engine level
  - Signature level
  - Header level
- Specific configurations for each level
- “list” command
  - Display current level configuration
- “delete command
  - Delete current level configuration
- “exit” command
  - Return to the previous level
- Announce changes to the central application

Conclusion



- Analyze traffic in kernelspace
  - Low latency
  - Minimal overhead
  - Control over packets
    - They can be blocked without other applications like iptables
    - Ability to prevent attacks
- Availability during configuration
  - Updates made by the central application
  - Restarting the application is not necessary
- Signature definition granularity
- Multiple types of alerts: email, pop-up, terminal alert
- High degree of modularity, relying on distinct engines