IC3 - Network Security

M.Sc. in Information Security

Royal Holloway, University of London

IC3 - Network Security

Lecture 11

Intrusion Detection and Prevention

Objectives of Lecture

- Background and Motivation for IDS/IPS
- Describe the Classes of ID Approaches
- Discuss Approaches for Anomaly Detection
- Discuss Approaches for Signature Detection
- Review Example Implementations
- Intrusion Prevention Systems
- Honeypots and Honeynets

Contents

Motivation for IDS/IPS

- Systems and networks will be compromised, almost regardless of what we do for perimeter security
  - Defenders have to get it right all the time, an attacker just has to get lucky once
- Even in the absence of external attackers, insiders can cause considerable damage or e.g. conduct industrial espionage
  - These users must have access to the target systems and networks by definition

Defining IDS

- An Intrusion Detection System (IDS) is a host or network based security component monitoring activities and identifying patterns of behavior or traffic indicating possible violations of security policy
- The difference between IDS and IPS often tends to be in the eye of marketing, and frequently amounts to s/d/p/g

Classifying Countermeasures (1)

- Preemptive countermeasures
  - Active elimination of threats is generally not possible since it may be hard to identify the adversary and it will usually be illegal to act in such a way
- External prevention
  - Stopping attacks outside one's own enclave: Typically firewalls
- External deterrence
  - Threat of prosecution, notifying potential attackers of ongoing monitoring and surveillance
- Internal prevention
  - Compartmentalization, internal firewalls and hardening of hosts

Classifying Countermeasures (2)

- Internal deterrence
  - Sanctioning of security policy breaches
- Detection of attacks
  - This is the defensive positioning of the IDS
- Deception
  - Providing decoys and attractive targets: Honeypots and honeynets
- Active defense
  - Active and/or autonomous countermeasures (e.g. locking down user accounts and isolating compromised hosts)

The Ad Hoc Approach

- Popularized by Cliff Stoll's book "The Cuckoo's Egg" (1990)
  - Sysadmin discovers $0.75 discrepancy in billing for computer time and chases down hackers in Germany working for KGB in an effort lasting many months
- Manual examination of system logs for anomalies or even correlations among events is next to impossible
- Manually analyzing network traffic for signs of irregularities is impossible

Classifying Penetration Types

- External Penetration
  - Attackers do not have access to identification and authorization credentials or is able to circumvent auditing and access control mechanisms
- Internal Penetration
  - Attackers use the I&A credentials of another user or system entity
  - Also known as masquerade
- Misuse
  - Behavior counter to security policy by authenticated users

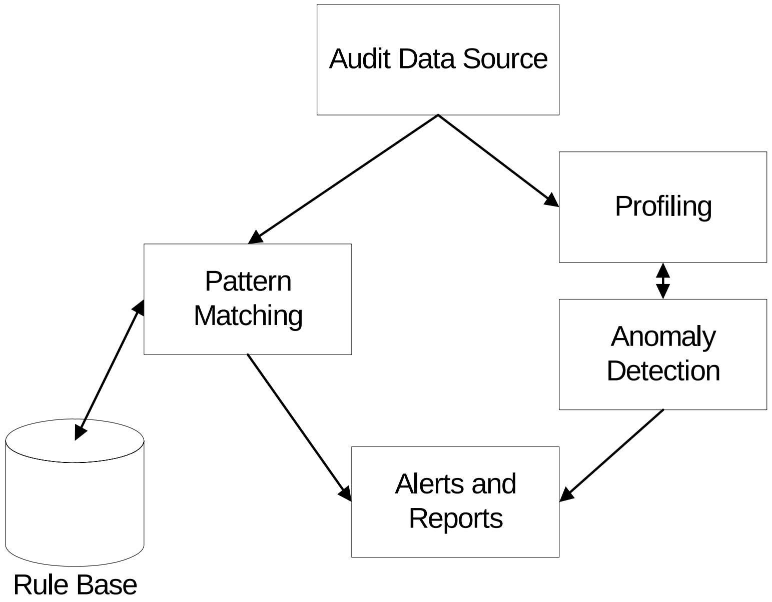
Historic Origins of IDS

- The masquerade problem was first recognized in military systems, even non-networked configurations
  - First described in a classic report by Anderson (1980): "Computer Security Threat Monitoring and Surveillance"
- Idea: Masquerade can be detected using audit data including
  - System use outside regular working hours
  - Abnormal frequency of use
  - Abnormal number of accesses to data and files
  - Abnormal access patterns (for both files and application programs)

Automated Intrusion Detection

- Detecting anomalous patterns or patterns matching known problematic activities
  - Volume of traffic is one significant impediment
  - Relevant patterns may be spread out over a long time and several hosts
  - Patterns may be too complex to see for a human
- Original proposal by Anderson: Automated techniques for reducing the volume of audit data
  - Extraction of relevant features and anomalies
  - Problems:
    - Quality of audit data is key for automated processing
    - Determining the precise criteria for matching

An Abstract IDS Model



Intrusion Detection Techniques

- Anomaly Detection
  - Does not require prior knowledge of adversary behavior patterns
  - IDS must learn to discriminate between normal system behavior and anomalous behavior
    - It can therefore also identify new patterns
  - A clear separation between normal and abnormal behavior is rarely possible
    - This requires a careful trade-off analysis between sensitivity and specificity

Anomaly Detection (2)

- User behavior changes over time
  - New tasks are assigned, users learn to handle the system in better ways
  - The definition of “normal” must therefore change as well to accommodate this conceptual drift
  - Anomaly detection IDS must track changes to avoid false positives and maintain appropriate definitions of normal and abnormal behavior
    - This also implies that attackers can induce the IDS to ignore undesirable behavior by slowly adapting behavioral patterns towards the desired action

Anomaly Detection (3)

- Some challenges for Anomaly Detection IDS
  - Large numbers of sensors (including OS audit data sources, network sensors, packet sniffers)
  - High temporal resolution
    - This results in high-dimensional feature vector spaces
    - Analysis and feature space reduction require not just precision but also speed
  - IDS data set may be skewed by the presence of attacks and compromised systems in its learning/ training phase
    - IDS will use compromised systems and networks as a baseline for comparison

Signature-based IDS (1)

- Audit data, network traffic and reports, etc. are compared against pre-defined patterns
  - Creation of these patterns (signatures) is typically done manually and error-prone
    - Signature authors must know the target and attacks precisely and be able to extract the critical distinguishing features of the attacks
  - Compared to anomaly IDS the specificity is significantly better (assuming good signatures)
  - Effective only against known attacks or minor variations on those attacks
    - Even minor changes to an attack can lead to a failure to match if the signature is over-specified

Specification-Based IDS

- Inverting the signature-based approach:
  - Specify legitimate behavior
  - Raise alert if deviations from this specified behavior are detected
- Usability for non-trivial application programs and system processes is dubious
  - Even if specifications and sources of application programs are available, their complexity will be too large to describe behavior with adequate precision
  - Alternatively, rough specification granularity reduces the sensitivity of the detection mechanism

Additional Taxonomic Identifiers (1)

- Source of audit and raw data
  - Host-based sources (and IDS): Sensors are emplaced locally on nodes to be monitored
    - Allows direct use of operating system audit data
    - Additional instrumentation (e.g. fine-grained kernel monitoring) is possible at this level
    - Application and user-specific data is available
  - Network-based sources: Raw data (e.g. packet sniffers) or aggregate data (e.g. SNMP traps) is collected
    - Requires instrumentation at all relevant locations, difficult to achieve in switched networks
    - Network data stream often lacks important contextual data

Additional Taxonomic Identifiers (2)

- Reaction to detected attacks
  - Passive reaction: Usually simply notification of the network/security administrator
  - Active reaction
    - Attempt to limit damage without requiring human intervention
      - Measures may affect only local systems
        - » E.g. increase in audit granularity
      - Measures affecting the attacking node
        - » E.g. attempts to isolate attacking nodes
    - Can be turned against the defender and be problematic in case of false positives as well

Additional Taxonomic Identifiers (3)

- Delay until attacks are recognized
  - Real-time IDS: Fixed upper bound on time elapsing between attack and detection
    - Most IDS cannot provide such bounds
  - Post factum analysis is a mode of operation supported by almost all systems
- Granularity of processing
  - Processing of sensor data as soon as they arise vs. batch processing of observations
- Location of processing: Local, centralized, hybrid
- Location of collection: Individual sensors, distributed systems,…

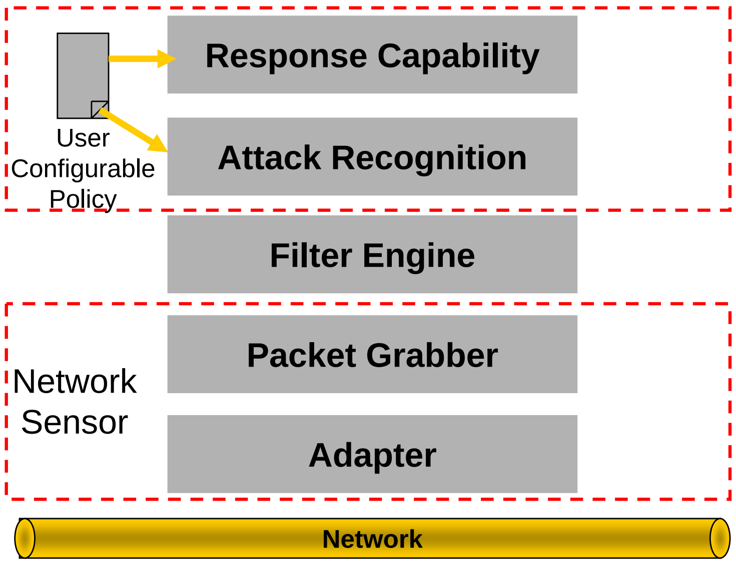
IDS Architecture Elements

- Modern IDS typically consist of a distributed set of sensors – either located on hosts or on network
- Centralized management system (console) for monitoring and administering the sensor network, to analyze data, report and react.
- Ideally:
  - Protected communications between sensors and console
  - Protected, tamper-resistant storage for signature database/logs
  - Secure console configuration
  - Secured signature updates from vendor
  - Secured state information for anomaly detection
  - IDS must be capable of identifying manipulation attempts (self-defense capability)

Example: Simple Network IDS

- Uses network packets (from a sniffer or host) as the data source
  - This can simply be a network adapter running in promiscuous mode
  - Objective is to monitor and analyze all traffic on a given network segment in (near) real-time
- Attack recognition can use several techniques to recognize patterns signifying potential attacks, e.g.
  - Pattern, expression or bytecode matching
  - Frequency or threshold crossing (e.g. detection of port scanning activity)
  - Correlation of lesser events (not much of this in commercial systems because of problems with specificity)

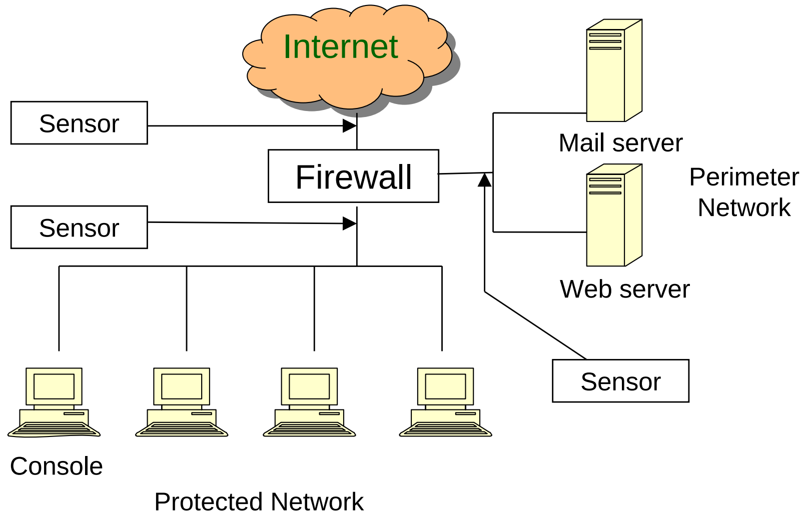
Simple Network-Based IDS



Placement of Network IDS (1)

- Deployment options:
  - Outside firewall
  - Just inside firewall
    - Combination of both will detect attacks getting through firewall and may help to refine firewall rule set.
  - Behind remote access server
  - Between business units
  - Between corporate network and partner networks
- Sensors may need to be placed in all switched network segments

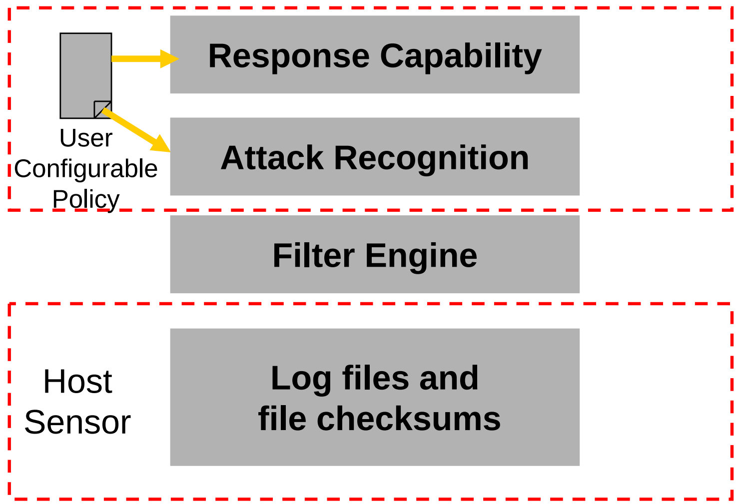
Placement of Network IDS (2)



Host-Based IDS

- Typically monitors system, event, and security logs on Windows and syslog in Unix environments
  - May use custom sensors (e.g. implemented as kernel modules)
- Checks key system files and executables via checksums at regular intervals for unexpected changes
  - Popularized by the Tripwire utility, now part of Windows Vista
- Some products can use regular-expressions to refine attack signatures
- Some products listen to port activity and alert when specific ports are accessed – resulting in a limited/partial NIDS capability

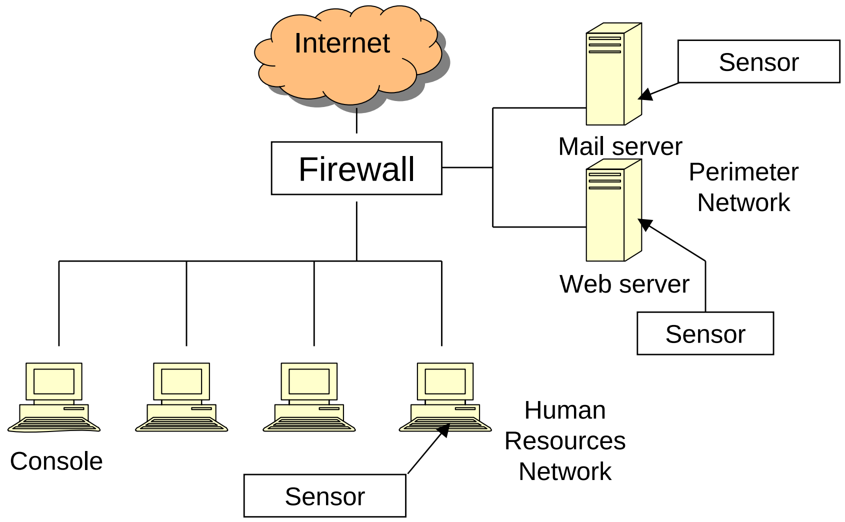
Host-Based IDS



Placement of Host-Based IDS

- Deployment options:
  - Key servers that contain mission-critical and sensitive information
  - Web servers
  - FTP and DNS servers
  - E-commerce database servers, etc.
  - Other high value assets
    - May also emplace these randomly to obtain probabilistic measure of hosts becoming compromised

Placement of Host-Based IDS



Analytical Techniques for IDS

- Basic concepts for anomaly detection: Anderson (1980)
- Continued development towards a first formalized model by Peter Neumann and Dorothy Denning (1987ff)
  - These formed the basis for the IDES system at SRI and almost all subsequent IDS
- Metrics developed included
  - Event counters
  - Time intervals
  - Resource measures

Statistical Models for Anomaly Detection

- Random variable x
- Sequence of observations $x_1,\ldots,x_n$
- Based on one or more observations x_{n+1} the IDS must decide if an anomaly is present
- Simplest approach: Operative models
  - Comparison of observations against fixed thresholds
  - Alerts are raised on exceeding thresholds
  - Thresholds must be determined (manually, using heuristics) from prior observations and may require revisions later

1^{st} and 2^{nd} Order Moments (1)

- Use 1^{st} and 2^{nd} order moments
  - Averaging previous observations:

$$\mu_x = \frac{1}{n}\sum_{i=1}^{n} x_i$$

  - Standard deviations from prior observations:

$$\sigma_x = \sqrt{\frac{1}{n}\left\{\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2\right\}}$$

1^{st} and 2^{nd} Order Moments (2)

- New observations x_{n+1} are defined as abnormal if they are found outside the confidence interval defined by *d* standard deviations
  - Chebyshev's inequality: Probability that the new observation is outside the interval is $1/d^2$
- This model is applicable for all metrics and observations
- There is no requirement to model the bounds explicitly
- Conceptual drift can be accomodated by e.g. weighing newer observations more than older observations

Multivariate Approaches

- Extensions of simple statistical models: Correlations between two or more metrics
  - Identifies relations between multiple variables
- Example: Factor analysis
  - Identifies covariances between sets of variables through a finite set of hidden (latent) variables
  - Assumption: Variable dependencies are linear, there is no uncorrelated noise, variations are separate
  - Permits estimates of linear relations and the amount of variations

Multidimensional Scaling

- Permits the detection of global similarities between observations by reducing the dimensionality of the observation space
- For each 2 objects *i,j* define a proximity metric p_{ij} (to be smaller if *i,j* have higher similarity)
- Configuration *X* represents a set of *n* points in a *m*-dimensional space, *n* x *n* matrix of n coordinates of the points on *m* axes of a Cartesian coordinate system. Distance in X is given as

$$d_{ij} = \sqrt{\sum_{a=1}^{m}\left(x_{ia} - x_{ja}\right)^2}$$

MDS Approaches

- Different MDS can now be constructed through the choice of a mapping function $f(p_{ij})$:
  - Absolute MDS:
    Distance between points i,j: $f(p_{ij}) = d_{ij}$
  - Relational MDS: Uses multiplicative constant b such that $f(p_{ij}) = bp_{ij}$ for all defined p_{ij}
  - Interval MDS: Uses a linear function *f()*
  - Nonmetric MDS: Operation does not occur directly on proximity metric; instead, *f()* can be an arbitrary (order-preserving) transformation of proximity values

Markov Processes for Anomaly Detection

- Random process in which transition probabilities from one state to the next depend solely on the preceding state:

$$p(S) = p(s_1 \cdots s_n) = p(s_1)\prod_{i=2}^{n} p(s_i \mid s_{i-1})$$

- Only event counters are suitable as metrics, but each individual observation can be a random variable
- First order Markov process: Only a single preceding observation is considered
  - Can be viewed as a 2D matrix
  - Anomaly is detected if a value in the matrix exceeds a threshold

Time Series for Anomaly Detection

- Sequence and time distance between observations $x_1 \ldots x_n$ are recorded
- Observations are considered abnormal if the probability of an observation occurring at the measured point in time is low
- This allows the identification of trends over longer periods of time compared to simple statistical techniques based on 1st and 2nd order moments
  - However, the computational complexity compared to these approaches is also significantly larger

Exotica: Genetic Algorithms (1)

- Can be considered an iterative optimization technique
- Attempt at modeling natural selection and genetics
  - Variables are considered as genes and are mapped onto chromosomes
  - Candidates for solutions of the optimization problem are an initial population, with improvements of the population through
    - Natural selection: Favorable traits are passed on
    - Mutation and recombination: Random changes and intermixing of “parent” chromosomes
  - Disadvantage: May not work in Kansas

Exotica: Genetic Algorithms (2)

- Basic pattern for genetic algorithms:
  - Determine initial polulation (e.g. randomly, through modification of existing genomes): This population must be sufficiently diverse
  - Evaluation of fitness for individual chromosomes through a numerical fitness function (e.g. Bohachevsky function)
  - Selection of chromosomes with highest fitness values, pseudorandom combination of other high fitness chromosomes, and elimination of low fitness chromosomes
  - Recombination and mutation: Random pairing of chromosomes results in two new child chr., mutations change only individual genes

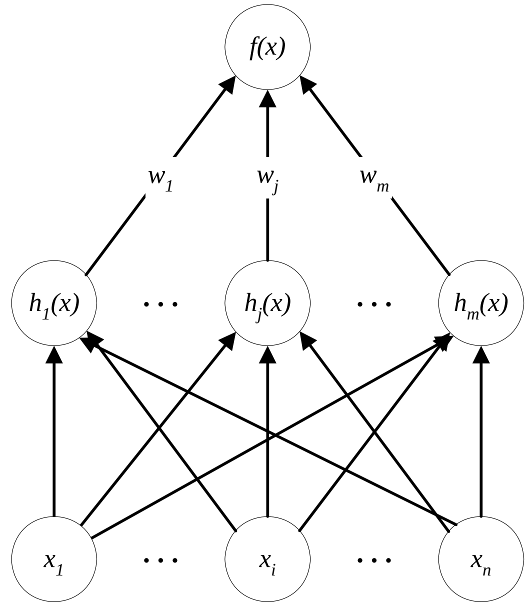
Exotica: Neural Networks (1)

- Nonlinear regression / discriminant / data reduction algorithm
- Biophysical analogy: Neurons “fire” if certain thresholds are reached on input axons: Threshold function is usually sigmoid function
- Neurons can be arranged in layers, processing is then directional
- Optimization objective is the reduction of the sum of classification errors for a training data set
  - This can be achieved by assigning suitable weights to the input values

Exotica: Neural Networks (2)

- Appropriate choice of the threshold function permits the representation as a differential equation
  - This allows the reverse computation of weighting influences across several layers
  - Changes to weighting realizes a gradient function
- Other variants
  - Radial basis functions
    - Simple structure with a single association layer
  - Self-Organizing (Kohonen) Maps
    - Self-configuring topological mapping functions

Simple Radial Basis Function



Exotica: Immunological Analogies

- The immune system of vertebrates must distinguish unknown proteins from self proteins and classify some as dangerous
  - Random detection patterns are generated in T cells
  - T cells showing self-recognition (autoimmune reaction) must be eliminated
- Analogy: Use strings carrying observation vectors as equivalent to proteins and develop detectors for such strings
  - Detectors representing harmless behavior must be culled from the detector set

Production Systems for Signature Detection

- Sets of rules with a premise part and at least one consequence part, potentially also including a conditional branch
  - Characteristics of attacks or attack components are decomposed into such rules by domain experts
  - Systems have historically been interpreted (and hence slow)
  - Depends heavily on the quality of the knowledge provided by the experts
    - Rule set must be kept minimal to ensure adequate performance and must also be comprehensive
    - With too many rules, a combinatorial “explosion” can occur

State Transition Models

- Modeling of attacks/incidents as a sequence of discrete events
  - Assignment of events to actors (entities)
  - Temporal sequence of events
- Modeling can e.g. occur in a state transition graph: Events are transitions/edges, states are represented as vertices (e.g. achieving root privileges)
  - Efficient representation e.g. through finite automata
  - Parallel computation of multiple automata is possible
  - Permits intuitive modeling of expert knowledge
  - Independent of timing constraints

Example Systems

- Seminal research on IDS at SRI in Stanford
- Many foundational results and theoretical model came from a family of projects at SRI (Denning, Neumann, Lunt):
  - IDES primarily investigated anomaly detection
    - Assumption: Legitimate user behavior is predictable
  - Later additions included signature detection using an expert/production system approach
  - NIDES incorporated the use of multiple hosts as sources
    - Data was converted into canonical form prior to processing at centralized site
  - EMERALD is a distributed framework for sensors and mechanisms

Example Systems: Hyperview

- Considers audit data to be a multivariate time series
  - Users are „dynamic processes“
- Two components: Expert system and neural network
  - Time series are mapped onto the NN
  - Partial feedback of NN output
    - Permits integration of memory into NN
  - Expert system also served as input system for neural network and provided additional information for decision processes

Example Systems: IDIOT

- „Intrusion Detection in Our Time“
- IDIOT models attacks as colored Petri nets
  - This permits the parallel consideration of several alternatives for a possible attack
  - There can be an arbitrary number of paths between two vertices in the Petri net using different transition sequences
  - The model lends itself to intuitive and elegant visualization
  - Efficient and suitable for real-time IDS
    - Very large CPN are commonplace e.g. in industrial control simulation environments, good tool support

Example Systems: Snort

- Snort is a fast, flexible, small-footprint, open-source NIDS developed by the security community and a “benevolent dictator”
- Lead coder: Marty Roesch, now founder of Sourcefire (http://www.sourcefire.com)
- Initially developed in late 1998 as a sniffer with consistent output, unlike protocol-dependent output of tcpdump
- Licensed under GPL, rule set has a different license

Rules in Snort

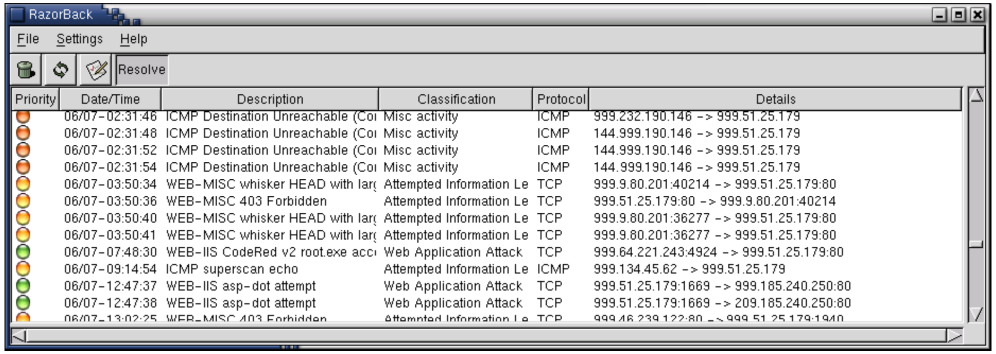
- Snort rules are extremely flexible and are easy to modify, unlike many commercial NIDS
- Sample rule to detect SubSeven trojan:

```
alert tcp $EXTERNAL_NET 27374 -> $HOME_NET any
   (msg:"BACKDOOR subseven 22"; flags: A+; content:
   "|0d0a5b52504c5d3030320d0a|"; reference:arachnids,485;
   reference:url,www.hackfix.org/subseven/; sid:103;
   classtype:misc-activity; rev:4;)
```

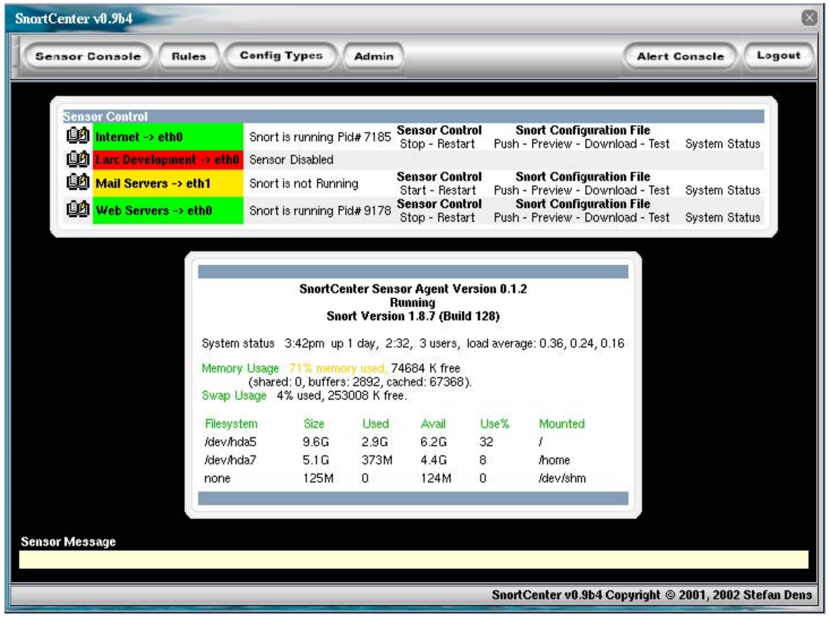
- Elements before parentheses comprise rule header
- Elements in parentheses are rule options

Post-Processing Snort Output

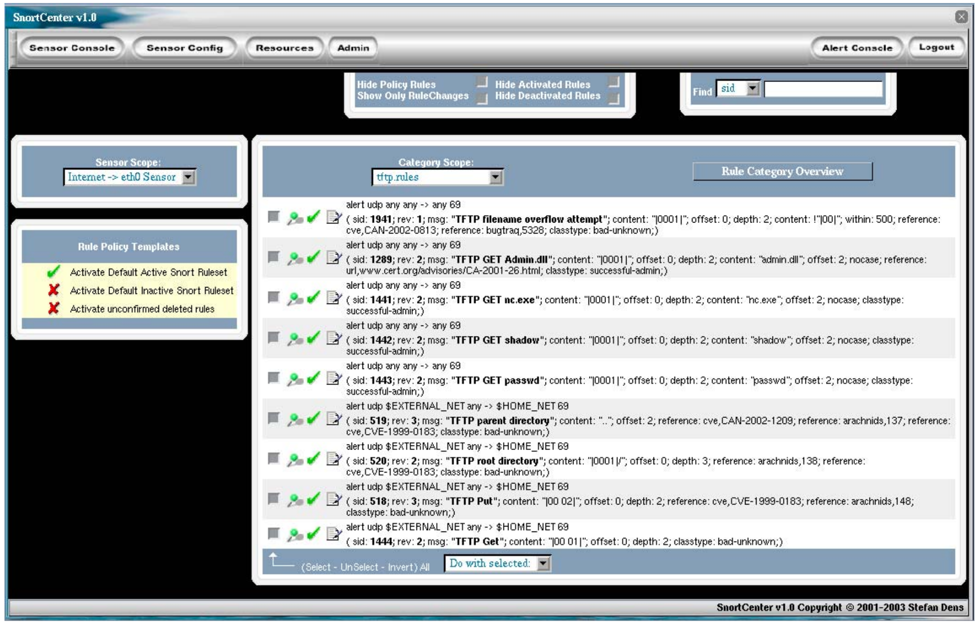
- Filtering and postprocessing is almost inevitable even with a simple pattern matching approach like Snort
  - Large number of add-on and layered tools (also Meta-IDS) including RazorBack, SnortCenter,…



SnortCenter (1)



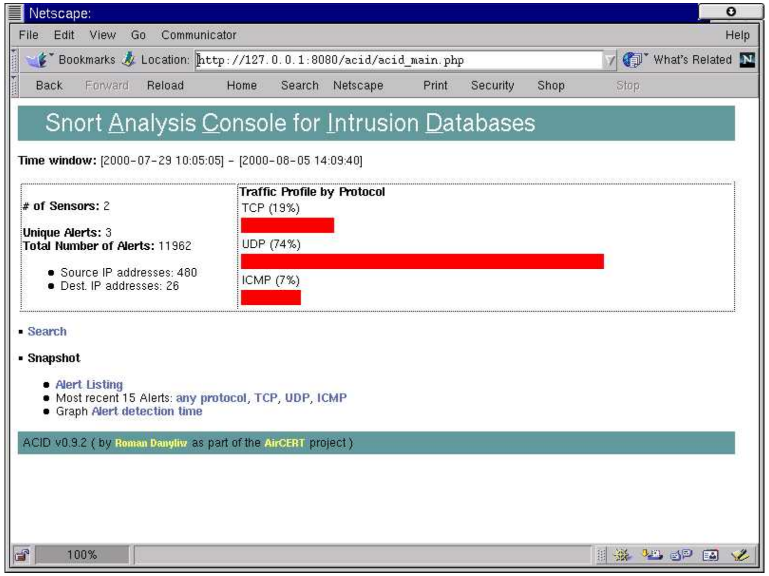
SnortCenter (2)



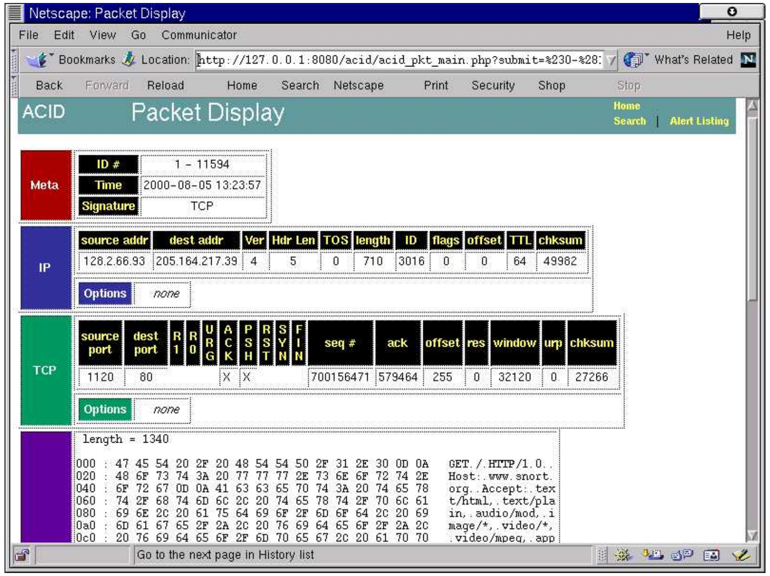
ACID (1)

- Analysis Console for Intrusion Databases (ACID)
  - http://acidlab.sourceforge.net/
  - PHP-based analysis engine to search and process a database of security events generated by various IDSes, firewalls, and network monitoring tools
  - Query-builder and search interface, packet viewer (decoder), alert management, chart and statistics generation

ACID (2)



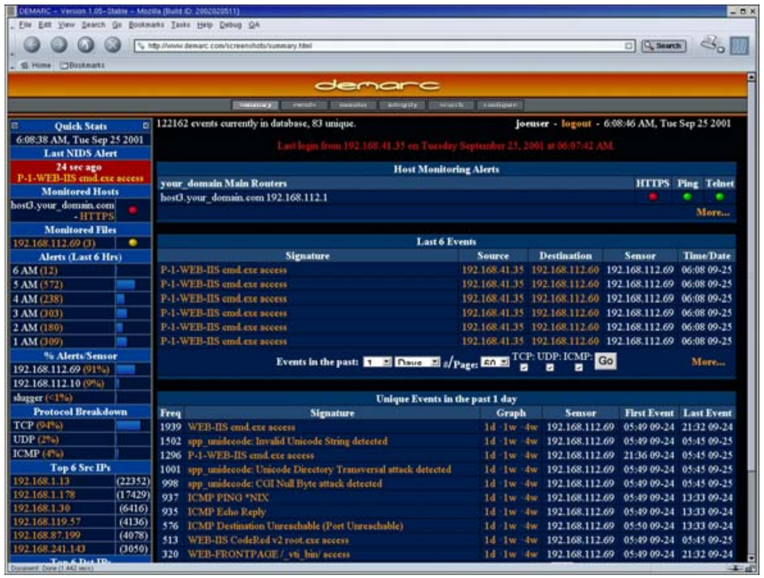
ACID (3)



Demarc (1)

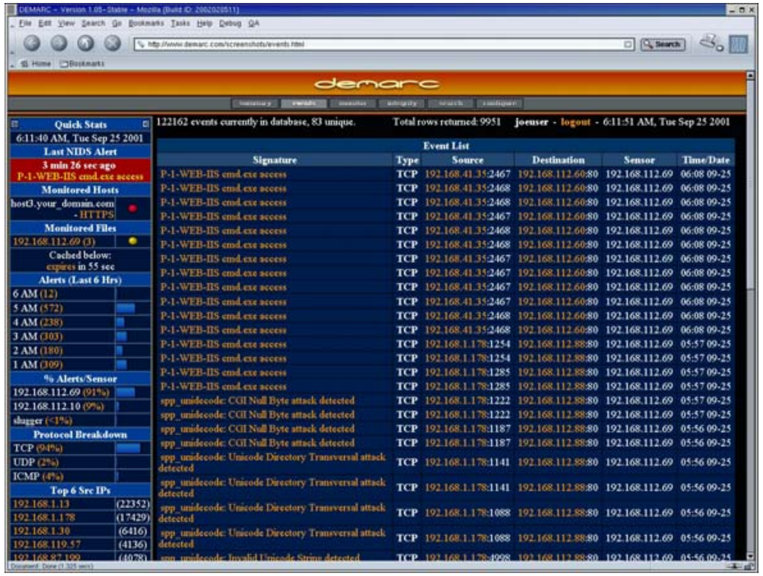
- NIDS management console, integrating Snort with the convenience and power of a centralized interface for all network sensors
  - www.demarc.com
  - Monitor all servers / hosts to make sure network services such as a mail or web servers remain accessible at all times
  - Monitor system logs for anomalous log entries that may indicate intruders or system malfunctions

Demarc (2)



Demarc (3)

Royal Holloway University of London



Intrusion Prevention Systems

Royal Holloway University of London

- Relatively new (marketing) term
- Essentially a combination of access control (firewall/router) and intrusion detection systems
  - Often shared technologies between stateful inspection and signature recognition (“looking deep into the packet”)
  - *Inline* network IDS allows for instant access control policy modification
- Modest success so far mainly with protection against flooding-type (DoS) attacks

Defining IPS

Royal Holloway University of London

- Can be defined as an in-line product that focuses on identifying and blocking malicious network activity in real time
- Two general categories:
  - rate-based products
  - content-based (also referred to as signature- and anomaly-based)
- Often look like firewalls and often have some basic firewall functionality
- But firewalls block all traffic except that which they have a reason to pass
- IPSs pass all traffic except that which they have a reason to block

Rate-Based IPS

Royal Holloway University of London

- Block traffic based on load:
  - too many packets
  - too many connects
  - too many errors
- In the presence of too much of anything, the rate-based IPS kicks in and blocks, throttles or otherwise mediates the traffic
- Most useful rate-based IPS include a combination of powerful configuration options with range of response technologies
  - For example, limit queries to your DNS server to 1,000 per second
  - Other simple rules covering bandwidth and connection limiting

Limitations of Rate-Based IPS

- Biggest problem with deploying rate-based IPS products is deciding what constitutes an overload
- For any rate-based IPS to work properly, need to know not only what "normal" traffic levels are (on a host-by-host and port-by-port basis) but also other network details such as how many connections your Web servers can handle
- Most products do not provide any help but require a “trained” system engineer
- Because rate-based IPSs require frequent tuning and adjustment, they will be most useful in very high-volume Web, application and mail server environments

Content-Based IPS

- Block traffic based on attack signatures and protocol anomalies
- Worms, e.g. Blaster and MyDoom, that match a signature can be blocked
- Packets that do not comply to TCP/IP RFCs can be dropped
- Suspicious behaviour such as port scanning triggers the IPS to block future traffic from a single host
- The best content-based IPSs offer a range of techniques for identifying malicious content and many options for how to handle the attacks
  - simply dropping bad packets
  - dropping future packets from the same attacker
  - reporting and alerting strategies
- IDS-like technology for identifying threats and blocking them, content-based IPSs can be used deep inside the network to complement firewalls and provide security policy enforcement

Honeypots and Honeynets

- Technology used to track, learn and gather evidence of hacker activities
- Definition
  - “… a resource whose value is being attacked or compromised”

    Laurence Spitzner, “The value of honeypots”, SecurityFocus, October 2001
- Strategically placed systems designed to mimic production systems, but not reveal “real” data
- Modes of operation
  - Baiting
  - Waiting
  - Collating
  - Disseminating

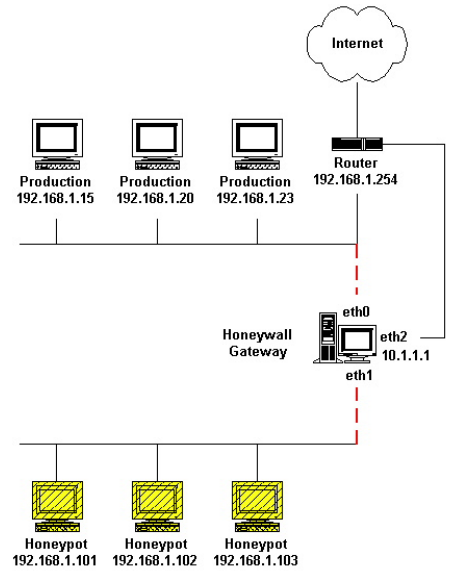
Honeypot Flavors

- Level of Involvement
  - Low Involvement: Port Listeners
  - Mid Involvement: Fake Daemons
  - High Involvement: Real Services
- Risk increases with level of involvement
  - Real services can cause real damage…

Honeynets

- Network of honeypots
- Supplemented by firewalls and intrusion detection systems - Honeywall
- Advantages:
  - "More realistic" environment
  - Improved possibilities to collect data
- World-wide net of research activities
  - http://www.honeynet.org

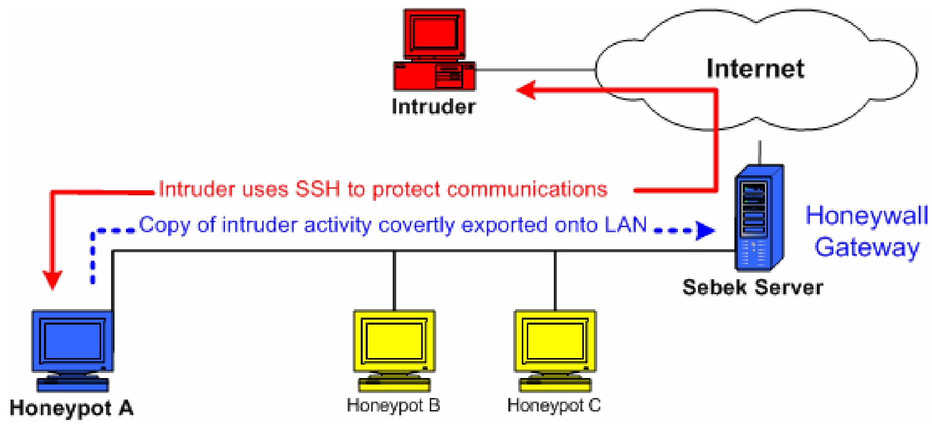
Sample Honeynet Topology



Sebek

- Sebek is a data capture tool designed to capture all of the attackers activities on a honeypot, without the attacker knowing it
- Consists of two components:
  - Client that runs on the honeypots, its purpose is to capture all of the attackers activities (keystrokes, file uploads, passwords) then covertly send the data to the server
  - Server which collects the data from the honeypots. The server normally runs on the Honeywall gateway
- Since the Sebek client runs as a kernel module on the honeypots, it can capture all activity, including encrypted, such as SSH, IPSec
- If this looks suspiciously like a rootkit… it almost is

Honeynet Topology with Honeywall



Happy Holidays!