From IDS to Security Intelligence

10%

Bo Luo
Associate Professor, EECS
Director, Information Assurance Lab, ITTC
The University of Kansas, Lawrence, KS, USA
bluo@ku.edu; http://www.ittc.ku.edu/~bluo

The Road to Better Situational Awareness

Intrusion Detection Systems (1990)

Network flows, Host Intrusion Detection logs, etc.

Security Information and Event Management (SIEM) (mid-2000)

Alarm Correlation

Big Data Security/Analytics (now)

Variety of Data, Security Intelligence

Data Analytics for Intrusion Detection

- By *Cloud Security Alliance*:
- 1st generation: Intrusion detection systems – Security architects realized the need for layered security (e.g reactive security and breach response) because a system with 100% protective security is impossible.

Data Analytics for Intrusion Detection

- 2nd generation: Security information and event management (SIEM) – Managing alerts from different intrusion detection sensors and rules was a big challenge in enterprise settings. SIEM systems aggregate and filter alarms from many sources and present actionable information to security analysts.

Data Analytics for Intrusion Detection

不考

- 3rd generation: Big Data analytics in security (2nd generation SIEM) – Big Data tools have the potential to provide a significant advance in actionable security intelligence by reducing the time for correlating, consolidating, and contextualizing diverse security event information, and also for correlating long-term historical data for forensic purposes.

Intrusion

- What is Intrusion?
  - remote root compromise
  - web server defacement
  - guessing / cracking passwords
  - copying databases containing credit card numbers
  - viewing sensitive data without authorization
  - running a packet sniffer
  - distributing pirated software
  - using an unsecured modem to access internal network
  - impersonating an executive to get information
  - using an unattended workstation

Intrusion

- Intrusion
  - “an Intrusion is unauthorized access to and/or activity in an information system.”
  - if an authorized action … exploits a vulnerability … causes a compromise …it becomes a successful attack/intrusion
    - outsider gained access to a protected resource
    - a buffer overflow has been exploited and then execute attack code inside a legitimate program
    - a program or file has been modified
    - system is not behaving “as it should”

Intruder Behavior

- Intrusion steps:
  - Target acquisition and information gathering
    - **select the target using IP lookup tools such as NSLookup, Dig, and others**
    - **map network for accessible services using tools such as NMAP**
    - **identify potentially vulnerable services, such as pcAnywhere**
  - Initial access
    - **brute force (guess) pcAnywhere password**
  - Privilege escalation
    - **install remote administration tool such as DameWare**
    - **wait for administrator to log on and capture his password**

Intruder Behavior

- Typical intrusion steps:
  - Information gathering or system exploit
    - **use that password to access remainder of network**
    - **Transfer large numbers of documents to external repository**
  - Maintaining access
    - **Install rootkit with backdoor for later access**
    - **Modify or disable anti-virus and IDS**
  - Covering tracks
    - **Use rootkit to hide files installed**
    - **Edit log files to remove entries generated during intrusion**

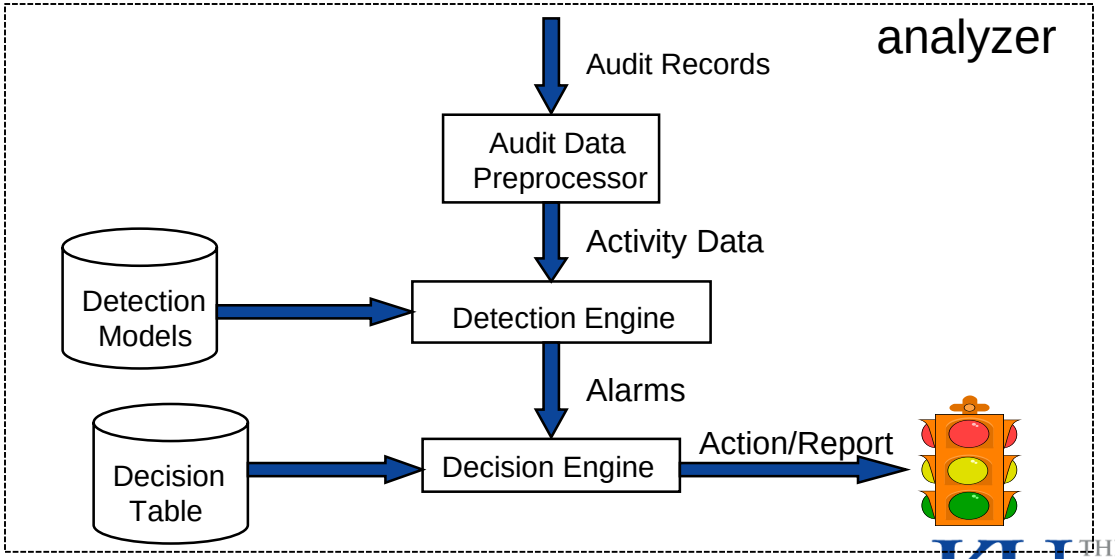
Intrusion Detection

- **Intrusion Detection** [RFC 2828]
  - a security service that monitors and analyzes system events for the purpose of finding, and providing real-time or near real-time warning of, attempts to access system resources in an unauthorized manner.

- **Intrusion prevention**
  - an extension of ID with exercises of access control to protect computers from exploitation

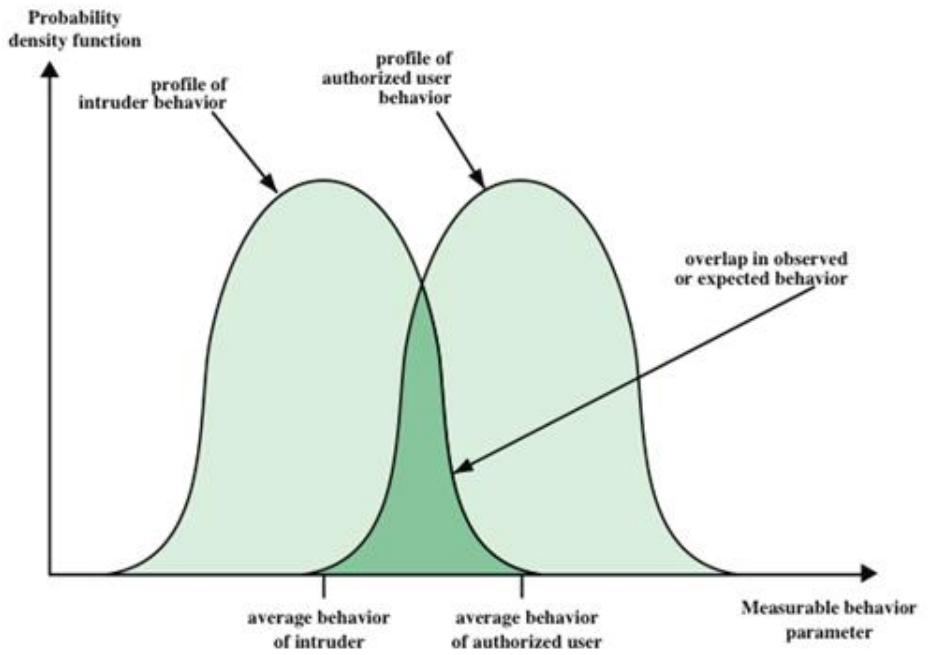
Intrusion Detection Systems

- IDS comprises three logical components:
  - sensors
    - collect data: packets, logs, system call traces, etc.
  - analyzers
    - determine if intrusion has occurred
  - user interface
    - view output or control system behavior

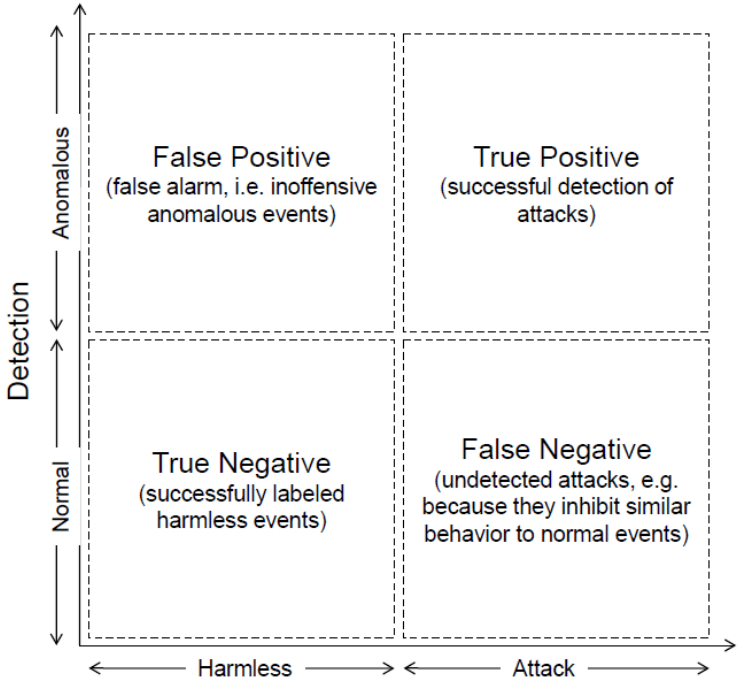


IDS Principles

- IDS is based on the assumption that intruder behavior differs from legitimate users
- Overlap in behaviors causes problems
  - false positives or false alarms detection intrusion is wrong
  - false negatives



Detection Quality



false： 误判
negative：恶意
True：好的

- FN is more severe in signature-based misuse detection
- FP is more severe in statistical anomaly detection

- An IDS with too many errors becomes useless.

Detection Quality

- Implications to IDS
  - deploy IDS to appropriate point/layer with sufficiently high base rate
  - design algorithms to reduce false alarm rate
- Ideally, IDS is desired to have a high detection rate and low false alarm rate
  - very difficult to meet this standard
  - because of the base-rate fallacy
  - in general, if # of intrusion is low compared to # of legitimate uses, the false alarm rate is high

Bayesian Detection Rate

skip

- Formal model:
  - Two random variables: given an event
    - *A* denotes an Alarm is generated
    - *I* denotes the event is indeed an Intrusion
  - **detection rate** (true positive): $P(A|I)$
  - So, **false negative rate** $P(!A|I)$ false alarm
  - **false positive rate**: $P(A|!I)$
  - So, **true negative rate** $P(!A|!I)$
  - *Bayesian detection rate*: $P(I|A)$
    - given an alarm, how likely it is a real intrusion?

Base-rate Bayesian Fallacy

- According to Bayes Rule:

$$\Pr(I|A) = \frac{\Pr(A|I) \cdot \Pr(I)}{\Pr(A)}$$

- If we know
  - $\Pr(I)$: the attack probability
    - assume 1 attack every 10,000 uses, $\Pr(I) = 0.0001$
  - $\Pr(A)$: probability of an alarm (unknown!)
    - Can derive $\Pr(A) = \Pr(A|I)\Pr(I) + \Pr(A|!I)\Pr(!I)$

Base-rate Bayesian Fallacy

- According to Bayes Rule:

$$\Pr(I|A) = \frac{\Pr(A|I) \cdot \Pr(I)}{\Pr(A)}$$

- If we know
  - $\Pr(I)$: the attack probability
    - assume 1 attack every 10,000 uses, $\Pr(I) = 0.0001$
  - $\Pr(A)$: probability of an alarm (unknown!)
    - Can derive $\Pr(A) = \Pr(A|I)\Pr(I) + \Pr(A|!I)\Pr(!I)$
    - First, assume the IDS is 99% accurate
      - $\Pr(A|I) = 0.99$
      - $\Pr(!A|I) = 1 - \Pr(A|I) = 0.01$
      - $\Pr(!A|!I) = 0.99$
      - $\Pr(A|!I) = 0.01$
    - $\Pr(A) = 0.99 \times 0.0001 + 0.01 \times 0.9999 = 0.010098$

Base-rate Bayesian Fallacy

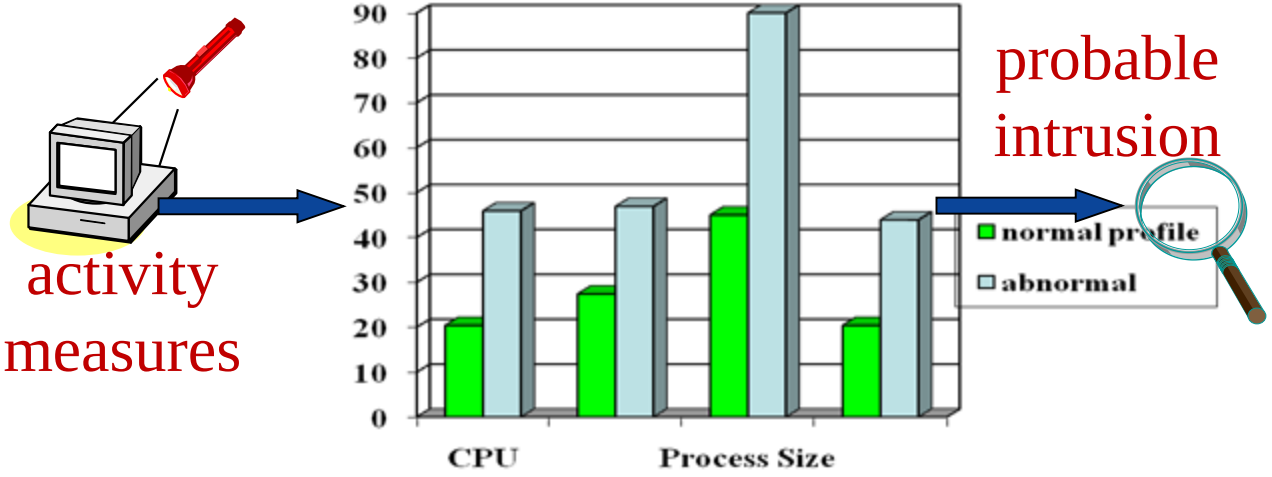
- According to Bayes Rule:

$$\Pr(I|A) = \frac{\Pr(A|I) \cdot \Pr(I)}{\Pr(A)}$$

- Now:
  - $\Pr(I|A) = \frac{0.99 \times 0.0001}{0.010098} = 0.0098 = 0.98\%$
- Implications:
  - a 99% accurate detector only leads to 1% accurate detection
    - 99 false alarms per true alarm
  - This is a core problem with IDS!
  - Need to suppression of false alarms
    - however difficult!

Intrusion Detection Approaches

- Anomaly detection
  - detects activity that deviates from the normal behavior
  - defines a profile describing “normal” behavior
    - involves the collection of data relating to the behavior of legitimate users over a period of time
  - detects potential attacks
    - analyzes the observed behavior to decide if it is of a legitimate user or of an intruder

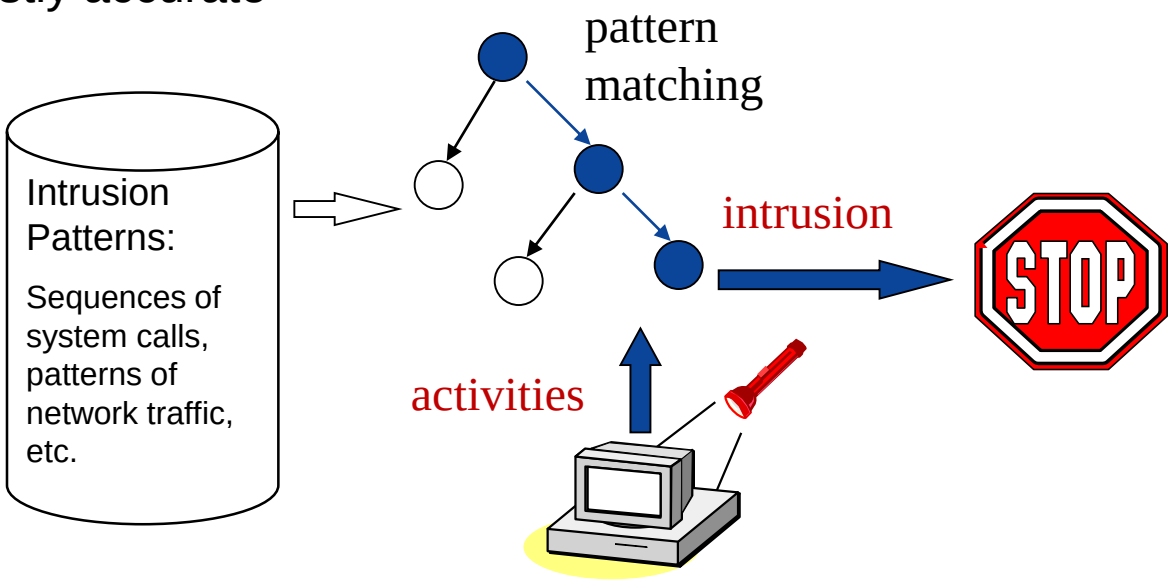


Intrusion Detection Approaches

- Model the legitimate user behavior in a training phase
  - A variety of classification approaches
  - Statistical
    - analysis of the observed behavior using univariate, multivariate, or time-series models of observed metrics
  - Knowledge based
    - uses an expert system that classifies observed behavior according to a set of rules to model legitimate behavior
  - Machine-learning
    - automatically determine a suitable classification model from the training data using data mining techniques
  - Need to consider efficiency and cost of the detection process

Intrusion Detection Approaches

- Signature/Heuristic detection (misuse detection) 已知的
  - uses a set of known malicious data patterns (signatures) or attack rules (heuristics)
    - only identifies known attacks for which it has patterns or rules
  - compares with current behavior
    - mostly accurate



Intrusion Detection Approaches

- Signature approaches
  - match a large collection of known patterns of malicious data
  - widely used in anti-virus products, network traffic scanning proxies, and NIDS
  - signatures are large enough to minimize false alarm rate
- Rule-based heuristic identification
  - define identified suspicious behavior in rules
    - even when the behavior is within the bounds of established patterns of usage
    - rules are system-specific, attack-specific
  - use rules to identify known penetrations or penetrations that would exploit known weaknesses
  - E.g., snort

Intrusion Detection Deployment

- Host-based IDS
  - use OS auditing and monitoring/analysis mechanisms to find malware
  - monitors the characteristics of a single host for suspicious activity
- Network-based IDS
  - deploying sensors at strategic locations
  - monitors network traffic, analyzes network, transport, and application protocols to identify suspicious activity
- Distributed or hybrid IDS
  - combines information from a number of sensors in a central analyzer

Host-Based IDS

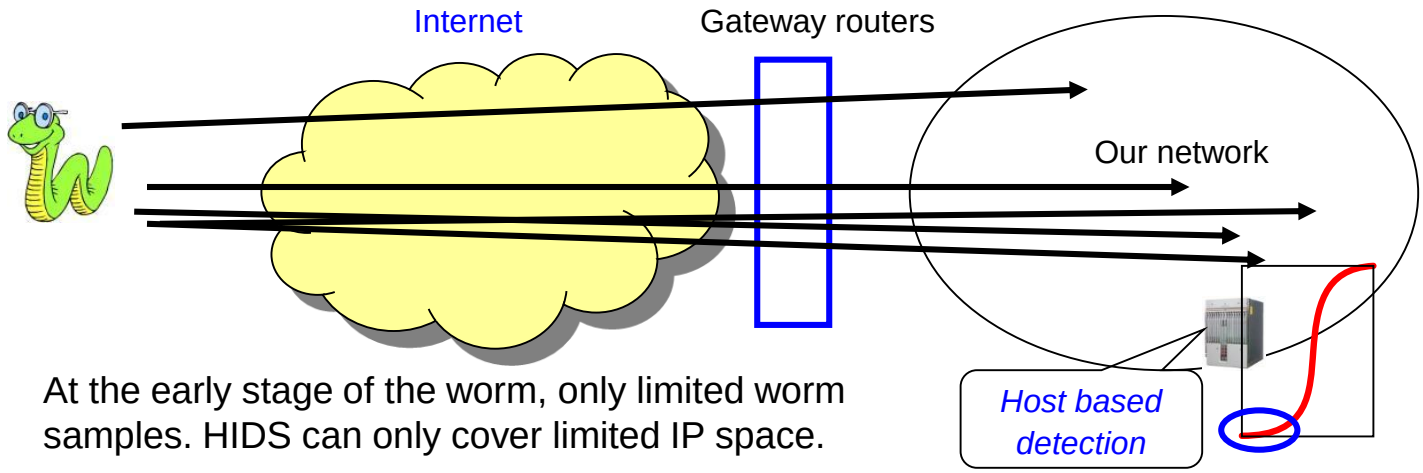
- Host-based
  - adds a specialized layer of security software to vulnerable or sensitive systems
  - execute full static and dynamic analysis of a program
- Monitors activity on the system in a variety of ways
  - System call traces: UNIX, Linus, Windows
  - Registry access: Windows
  - Audit (log file) records
  - File integrity checksums: tripwire
- Can detect both external and internal intrusions
  - detect intrusions, log suspicious events, and send alerts

Host-Based IDS

- Can use either anomaly or signature and heuristic approaches
  - Anomaly HIDS
    - mostly based on system call traces
  - Signature HIDS
    - widely used in anti-virus, anti-malware products

Host-Based IDS

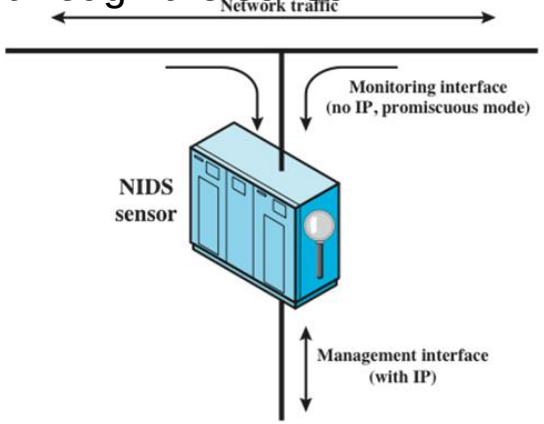
- Problems:
  - user dependent:
    - need to install/update IDS on all user machines!
  - can be tampered
    - if attacker takes over machine, can tamper with IDS binaries and modify audit logs
  - only local view of the attack



At the early stage of the worm, only limited worm samples. HIDS can only cover limited IP space.

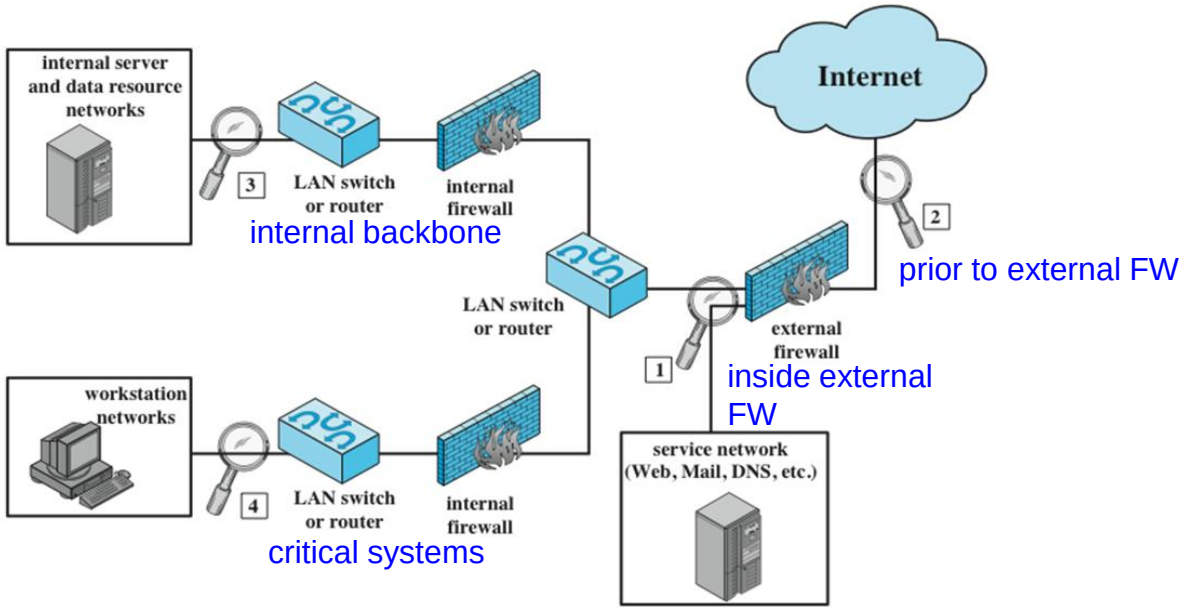
NIDS

- Network-based IDS
  - monitors traffic at selected points on a network
  - inspects network traffic
    - examines traffic packet by packet in real or close to real time at network, transport, application levels
  - deploys sensors at strategic locations
    - Inline sensor
      - inserted into a network segment
      - traffic that it is monitoring must pass through the sensor
    - Passive sensors
      - monitors a copy of network traffic
      - traffic does not pass through



NIDS

- Network-based IDS consists of
  - a number of sensors
  - one or more servers for NIDS management functions
  - one or more management consoles for the human interface
  - placement of sensors: an example



NIDS Detection Techniques

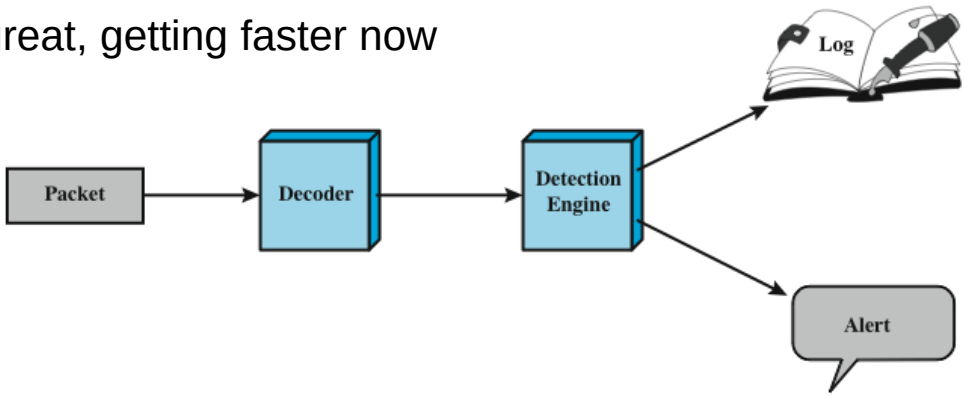
- Signature detection
  - reconnaissance and attack
    - Application layer: DHCP, DNS, NFS, FTP, Telnet, HTTP, SIP, IRC, rlogin/rsh, RPC, SMTP. IMAP, POP, etc.
    - Transport layer: TCP, UDP
    - Network layer: IPv4, IPv6, ICMP
  - unexpected application services, policy violations
- Anomaly detection
  - used for denial of service attacks, scanning, worms
  - Stateful Protocol Analysis (SPA)
    - a subset of anomaly detection
    - compares observed network traffic against predetermined universal vendor supplied profiles of benign protocol traffic

NIDS Alert

- Alert is generated when a violation is detected
  - used by analysis module: refine parameters, algorithms
  - used by security administration: design prevention techniques
  - information to be logged:
    - Timestamp
    - Connection or session ID
    - Event or alert type
    - Rating
    - Network, transport, and application layer protocols
    - Source and destination IP addresses
    - Source and destination ports, or ICMP types and codes
    - Number of bytes transmitted over the connection
    - Decoded payload data (application requests and responses)
    - State-related information

NIDS Example: SNORT

- Snort is a lightweight NIDS
  - signature based (more precisely, rule based)
    - first released in 1997 but still updated/maintained today
    - easily configured
  - easily deployed on nodes
    - can be run as a sniffer (IDS) or inline (IPS)
  - real-time packet capture, uses small amount of memory and processor time
    - 1 CPU w/ 1000 signatures can process 500MBps
      - not great, getting faster now



IDS

- Requirements
  - run continually
  - be fault tolerant
  - resist subversion
  - impose a minimal overhead on system
  - configured according to system security policies
  - adapt to changes in systems and users
  - scale to monitor large numbers of systems
  - provide graceful degradation of service
  - allow dynamic reconfiguration

IDS vs. Firewall/IPS

考到这里，后面不考

- Network IDS
  - Passive monitoring
  - Fail-open
- Firewall/IPS 入侵防御系统
  - Active filtering
  - Fail-close: not let data in!

Security Intelligence

- Data-driven information security
  - bank fraud detection: credit companies have done this for decades.
  - anomaly-based intrusion detection systems.
- Custom-built infrastructure to mine Big Data for fraud detection was not economical to adapt for other fraud detection uses.
- Off-the-shelf Big Data tools and techniques are now bringing attention to analytics for fraud detection in healthcare, insurance, and other fields.

Security Intelligence

Traditional Systems

- More rigid, predefined schemas
- Data gets deleted
- Complex analyst queries take long to complete

Big Data Promise

- Structured and unstructured data treated seamlessly
- Keep data for historical correlation (e.g., 10 years)
- Faster query response times

Security Intelligence

- IBM's Security Intelligence: predictive analytics, prioritized threat data, proactive response
  - Multi-vendor event correlation
  - Global monitoring
  - Threat prioritization
  - Sophisticated intelligence reporting
  - Real-time analysis
  - Automated Intelligence

Data Analytics for Intrusion Detection

- Traditional techniques vs. BD
  - Storing and retaining a large quantity of data was not economically feasible. Most event logs and other recorded computer activity were deleted after a fixed retention period (e.g., 60 days).
  - Performing analytics and complex queries on large, structured data sets was inefficient.
  - Traditional tools (rigid, defined schemas) were not designed to analyze and manage unstructured data.
  - Big Data tools (e.g., Piglatin scripts and regular expressions) can query data in flexible formats.
  - Big Data systems use cluster computing infrastructures reliable and available.

Data Analytics for Intrusion Detection

- Security intelligence with big data
  - collecting data at a massive scale from many internal enterprise sources and external sources such as vulnerability databases;
  - performing deeper analytics on the data;
  - providing a consolidated view of security-related information;
  - achieving real-time analysis of streaming data.
  - Big Data tools still require system architects and analysts to have a deep knowledge of their system in order to properly configure the Big Data analysis tools.

Examples

- Zions Bancorporation: using Hadoop clusters and business intelligence tools to parse more data more quickly than with traditional SIEM tools.
  - the quantity of data and the frequency analysis of events are too much for traditional SIEMs to handle alone.
  - traditional system: searching among a month's load of data could take between 20 minutes and an hour.
  - new Hadoop system running queries with Hive: get the same results in about one minute.
  - incorporation of unstructured data and multiple disparate data sets into a single analytical framework

Examples

- APT
  - An Advanced Persistent Threat (APT): targeted attack against a high-value asset or a physical system.
  - “low-and-slow” mode: maintains a low profile in the networks and allows for long execution time.
    - Unlike mass-spreading malware: worms, viruses, and Trojans,
  - Stolen user credentials or zero-day exploits to avoid triggering alerts.
  - can take place over an extended period of time while the victim organization remains oblivious to the intrusion.
  - 2010 Verizon data breach investigation report: in 86% of the cases, evidence about the data breach was recorded in the organization logs, but the detection mechanisms failed to raise security alarms.

Examples

- APT
  - Among the most serious information security threats that organizations face today.
  - operated by highly-skilled, well-funded and motivated attackers targeting sensitive information from specific organizations and operating over periods of months or years.
  - have become very sophisticated and diverse in the methods and technologies used, particularly in the ability to use organizations' own employees to penetrate the IT systems by using social engineering methods.
  - spear-phishing messages that are customized for each victim (e.g., emails, SMS, and PUSH messages); specially crafted malware that may contain zero-day exploits

Examples

- APT
  - detection relies heavily on the expertise of human analysts
  - custom signatures and perform manual investigation.
  - labor-intensive, difficult to generalize, and not scalable
  - Existing anomaly detection proposals commonly focus on obvious outliers (e.g., volume-based), but are ill-suited for stealthy APT attacks and suffer from high false positive rates.

Examples

- Beehive: Behavior Profiling for APT Detection
  - RSA Labs: however subtle the attack might be, the attacker's behavior should cause the compromised user's actions to deviate from their usual pattern.
  - APT attacks consist of multiple stages (e.g., exploitation, command-and-control, lateral movement, and objectives): each action by the attacker provides an opportunity to detect behavioral deviations from the norm.
  - Correlating these seemingly independent events can reveal evidence of the intrusion, exposing stealthy attacks that could not be identified with previous methods.

Examples

- Beehive: Behavior Profiling for APT Detection
  - detectors of behavioral deviations: “anomaly sensors,”
  - each sensor examining one aspect of the host’s or user’s activities
    - a sensor may keep track of the external sites a host contacts in order to identify unusual connections (potential command-and-control channels),
    - profile the set of machines each user logs into to find anomalous access patterns (potential “pivoting” behavior in the lateral movement stage),
    - study users’ regular working hours to flag suspicious activities in the middle of the night,
    - track the flow of data between internal hosts to find unusual “sinks” where large amounts of data are gathered (potential staging servers before data exfiltration).

Examples

- Beehive: Behavior Profiling for APT Detection
  - Triggering one sensor: presence of a singular unusual activity,
  - Triggering of multiple sensors: more suspicious behavior
  - Human analyst: the flexibility of combining multiple sensors according to known attack patterns (e.g., command-and-control communications followed by lateral movement) to look for abnormal events
  - Further investigation or to generate behavioral reports of a given user's activities across time.
  - Preliminary results: *Beehive* is able to process a day's worth of data (around a billion log messages) in an hour and identified policy violations and malware infections that would otherwise have gone unnoticed.

Examples

- Large-scale distributed computing for APT detection
  - APT detection: use large-scale methods to cover all possible attack paths.
  - model the APT as an attack pyramid: possible attack goal at the top, lateral planes representing the environments where the events can be recorded (e.g., user plane, network plane, application plane, or physical plane).
  - group all of the events recorded in an organization that could potentially be relevant for security using flexible correlation rules that can be redefined as the attack evolves.
  - detection rules (e.g., signature based, anomaly based, or policy based) to detect possible malicious activities within each context and across contexts

Examples

- Large-scale distributed computing for APT detection
- MapReduce paradigm
  - More efficiently handle highly unstructured data with arbitrary formats that are captured by many types of sensors (e.g., Syslog, IDS, Firewall, NetFlow, and DNS) over long periods of time.
  - Massive parallel processing mechanism: use much more sophisticated detection algorithms than the traditional SQL-DBMS (transactional workloads with highly structured data)
  - Users have the power and flexibility to incorporate any detection algorithms into the Map and Reduce functions.
  - Potential to help to analyze more data at once, to cover more attack paths and possible targets, and to reveal unknown threats in a context closer to the target.