Network Security - Week 8

João Soares

DCC/FCUP

2023

Proactive vs Reactive

Previously...

- We want to keep bad guys out
  - Authentication prevents intrusions
  - Firewalls are a form of intrusion prevention
  - Virus defenses aimed at avoiding intrusions
  - Locking the door on your car

Proactive vs Reactive

Previously...

- We want to keep bad guys out
  - Authentication prevents intrusions
  - Firewalls are a form of intrusion prevention
  - Virus defenses aimed at avoiding intrusions
  - Locking the door on your car

Intrusion Detection Systems

- What to do if they get in?
- Detect attacks in progress
- Look for *unusual* or *suspicious* activity
- IDS evolved from log file analysis

Classes of Intruders - Cyber Criminals

- Individuals or members of an organized crime group, with the goal of financial reward
- Activities include, but are not limited to
  - Identity theft
  - Theft of financial credentials
  - Corporate espionage
  - Data theft
  - Data ransoming

- Information exchanged in underground forums to trade tips/data and coordinate attacks
- Anonymous networks (Tor et. al.) are very good for this

Classes of Intruders - State-Sponsored Organizations

- Groups of hackers sponsored by governments to conduct espionage or sabotage activities

- Also known as Advanced Persistent Threats
- Covert nature
- Persistence over extended periods

- Widespread nature and scope by a wide range of countries (China, Russia, USA, UK, and intelligence allies)

Classes of Intruders - Activists

- Individuals motivated by social or political causes
  - Working as insiders
  - Members of a larger group
- Also known as hacktivists
- Skill level often not high
- Goal is to promote and publicize their cause, typically through:
  - Website defacement
  - Denial-of-service attacks
  - Theft and distribution of data, resulting in negative publicity or compromise of their targets

Classes of Intruders - Others

- Hackers with motivations other than previously listed

- Include classic hackers/crackers
- Motivated by technical challenge or peer-group esteem and reputation
- Many responsible for discover new vulnerabilities

- Given the wide availability of attack toolkits, there is a pool of “hobby hackers” exploring system and network security challenges

Insider attacks

- Among most difficult to detect and prevent
- Employees have access & systems knowledge

Motivation is key

- Revenge or entitlement
- Employment terminated
- Stealing customer data for competitor

IDS may help, but also...

- Least privilege configuration
- Monitor logs
- Strong authentication
- Termination to block access

Intruder Skill Levels

- **Apprentice**
- **Journeyman**
- **Master**

Intruder Skill Levels

- **Apprentice**
  - Hackers with minimal technical skill, who primarily use existing attack toolkits
  - They likely comprise the largest number of attackers, including many criminal/activist attackers
  - Given their use of existing known tools, these attackers are the easiest to defend against
  - Also known as “script-kiddies” from plug-and-play usage
- **Journeyman**
- **Master**

Intruder Skill Levels

- **Apprentice**
- **Journeyman**
  - Hackers with sufficient technical skills to modify and extend attack toolkits to use newly discovered, or purchased, vulnerabilities
  - They may be able to locate new vulnerabilities to exploit that are similar to some already known
  - Adapt tools for use by others
  - These hackers are found in all intruder classes
- **Master**

Intruder Skill Levels

- **Apprentice**
- **Journeyman**
- **Master**
  - Attackers with high-level technical skills capable of discovering brand new categories of vulnerabilities
  - Write new powerful attack toolkits
  - Some of the better known classical hackers are at this level
  - Some are employed by state-sponsored organizations
  - Defending against these attacks is of the highest difficulty

Intruder Behavior

1. Target acquisition and information gathering
2. Initial access
3. Privilege escalation
4. Information gathering or system exploit
5. Maintaining access
6. Covering tracks

Intruder Behavior

1. Target acquisition and information gathering
2. Initial access
3. Privilege escalation
4. Information gathering or system exploit
5. Maintaining access
6. Covering tracks

A time window for efficient countermeasures

- Only critical from step 3 onwards (most of the time)
- Monitoring system to react during this multi-stage process

Examples of 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 a pirated software
- Using an unsecured AP to access internal network
- Impersonating an executive to get information
- Using an unattended workstation

IDS Requirements

- Availability
  - Run continuously
  - Provide graceful degradation of service

IDS Requirements

- Availability
  - Run continuously
  - Provide graceful degradation of service
- Security
  - Be fault tolerant
  - Resist subversion

IDS Requirements

- Availability
  - Run continuously
  - Provide graceful degradation of service
- Security
  - Be fault tolerant
  - Resist subversion
- Performance
  - Impose a minimal overhead on a system
  - Scale to monitor large number of systems

IDS Requirements

- Availability
  - Run continuously
  - Provide graceful degradation of service
- Security
  - Be fault tolerant
  - Resist subversion
- Performance
  - Impose a minimal overhead on a system
  - Scale to monitor large number of systems
- Adaptability
  - Configured according to system security policies
  - Adapt to changes in systems/users/attack patterns
  - Allow dynamic reconfiguration

IDS Architecture

Three components:

- Sensors - collect data
- Analyser - assess intrusions
- User interface - view output for the control system behavior

IDS Architecture

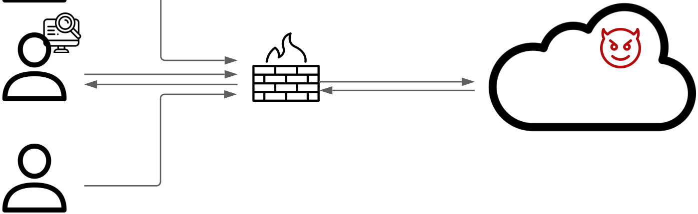
Three components:

- Sensors - collect data
- Analyser - assess intrusions
- User interface - view output for the control system behavior

Configurations

- Host-based IDS
  - Dedicated to a specific machine or service
  - Monitors the characteristics of a single host for suspicious activity
- Network-based IDS
  - Monitors network traffic
  - Analyses transport/application data to identify suspicious activity
- Distributed or Hybrid IDS
  - Combines information from multiple sensors
  - Host and network based, combined in a central analyser

Host-Based IDS



- Monitor activities on hosts for
  - Known attacks; Suspicious behavior
- Designed to detect attacks such as
  - Buffer overflow; Escalation of privilege
- Can detect both external and internal intrusions
- Little or no view of network activities

Examples of Host-Based IDS

- OSSec
  - Log analysis and file integrity checking
  - Policy monitoring and rootkit detection
  - Real-time alerting and active response

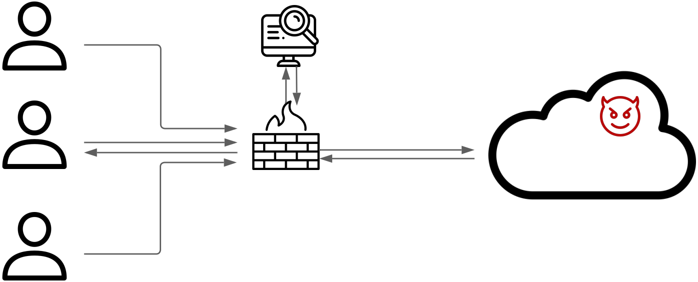
Examples of Host-Based IDS

- OSSec
  - Log analysis and file integrity checking
  - Policy monitoring and rootkit detection
  - Real-time alerting and active response
- Tripwire
  - Data integrity tool
  - Useful for monitoring and alerting specific file changes
  - Some commercial versions

Examples of Host-Based IDS

- OSSec
  - Log analysis and file integrity checking
  - Policy monitoring and rootkit detection
  - Real-time alerting and active response
- Tripwire
  - Data integrity tool
  - Useful for monitoring and alerting specific file changes
  - Some commercial versions
- AIDE
  - Advanced Intrusion Detection Environment
  - File and directory integrity checker

Network-Based IDS



- Monitor activity at selected points of the network for known attacks
- Examines network transport and application level protocols
- Designed to detect attacks such as:
  - Denial-of-service; network probes; malformed packets
- Some overlap with firewall
- Little to no view of host-based attacks

Examples of Network-Based IDS

- Snort
  - Network intrusion prevention and detection system
  - Combines signature, protocol and anomaly-based inspection

Examples of Network-Based IDS

- Snort
  - Network intrusion prevention and detection system
  - Combines signature, protocol and anomaly-based inspection
- Bro/Zeek
  - Focused on network security monitoring
  - A comprehensive platform for general network traffic analysis

Examples of Network-Based IDS

- Snort
  - Network intrusion prevention and detection system
  - Combines signature, protocol and anomaly-based inspection
- Bro/Zeek
  - Focused on network security monitoring
  - A comprehensive platform for general network traffic analysis
- Suricata-IDS
  - Real-time intrusion detection (IDS)
  - In-line intrusion prevention (IPS)
  - Network security monitoring (NSM)

IDS Methodologies

Signature Detection

- Set of known malicious data patterns or attack rules
- Also known as misuse detection
- Can only identify known attacks for which it has patterns or rules

IDS Methodologies

Signature Detection

- Set of known malicious data patterns or attack rules
- Also known as misuse detection
- Can only identify known attacks for which it has patterns or rules

Anomaly Detection

- Involves the collection of data relating to the behavior of legitimate users over a period of time
- Observed behavior is analysed to determine whether it matches a legitimate user or an intruder
- Pattern recognition and machine learning approaches

- Failed login attempts may suggest a password cracking attack
- IDS sets rule *N failed login attempts in M seconds* as an attack signature. Listens for messages and looks for signatures
- A pattern identified as a signature triggers a warning
- A lot of specificity involved:
  - Administrator knows what attack triggered the system
  - Allows for timely responses...
  - ... Or a verification for false alarms

Signature Detection
Minutia

- Suppose IDS warns whenever N or more failed logins occur in M seconds
  - Define N and M to reduce false alarms
  - Do this based on “normal” behavior
  - But normal behavior can be neither easy to define
  - Nor static on the system lifecycle

Signature Detection

Minutia

- Suppose IDS warns whenever N or more failed logins occur in M seconds
  - Define N and M to reduce false alarms
  - Do this based on “normal” behavior
  - But normal behavior can be neither easy to define
  - Nor static on the system lifecycle

Adversary - An arms race

- An oblivious Carlos can get caught
- But, knowing the signature, he can try $N - 1$ logins every M seconds
- Signature detection slows Carlos, but doesn’t stop it

Signature Detection

A more robust approach

- Goal is to detect “almost” signatures
- Look for $\sim N$ login attempts in $\sim M$ seconds

Signature Detection
A more robust approach

- Goal is to detect "almost" signatures
- Look for $\sim N$ login attempts in $\sim M$ seconds

- Considerations
  - Warn of possible password cracking attempts
  - What are reasonable values for $\sim$?
  - Establish a degree of confidence based on N and M
  - Can use statistical analysis, heuristics, etc.
  - Must not increase false alarm rate too much

Signature Detection

Advantages

- Simple
- Detects common, known threats
- Accurate identification of attacks upon detection
- Efficient (if we have a reasonable number of signatures)

Signature Detection

Advantages

- Simple
- Detects common, known threats
- Accurate identification of attacks upon detection
- Efficient (if we have a reasonable number of signatures)

Disadvantages

- Signature files must be kept up to date
- Number of signatures may become very large
- Can only detect known attacks
- Unexpected variations on known attacks may avoid detection

Anomaly Detection IDS

Goal: Detect new attacks automatically

Anomaly Detection IDS

Goal: Detect new attacks automatically

- Statistical anomaly detection uses statistical techniques to detect attacks. In simple terms:
  1. Establish baseline (normal) behavior
  2. Define the threshold from normal to abnormal behavior
  3. Gather new statistical data
  4. Measure deviation from baseline
  5. If the threshold is exceeded, issue an alarm
- A.k.a. behavior-based detection

Anomaly Detection IDS

Classification approaches

Statistical

- Analysis of the observed behavior using univariate, multivariate, or time-series models of observed metrics

Knowledge-based

- Approaches use an expert system that classifies observed behavior according to a set of rules that model legitimate behavior

Machine Learning

- Approaches automatically determine a suitable classification model from the training data using data mining techniques

Statistical

- The good
  - Simple
  - Lightweight
  - No big assumptions on known behavior
- The bad
  - Selecting variables for analysis
  - Tough to balance false positives v false negatives

Knowledge-based

Machine Learning

Statistical

Knowledge-based

- The good
  - Robust
  - Flexible
- The bad
  - Time consuming
  - Requires in-depth knowledge of application patterns

Machine Learning

Statistical

Knowledge-based

Machine Learning

- The good
  - Automatic classification
  - After training, fairly efficient
- The bad
  - Computational and time consuming
  - Overfitting is an issue

Detection of “Anomalies”

How can we measure the normal behavior of a system?

- Must measure during representative behavior
- Cannot be measured during an attack
- Normal is the statistical mean
- Must also allow for variance to know what is abnormal

Detection of “Anomalies”

How can we measure the normal behavior of a system?

- Must measure during representative behavior
- Cannot be measured during an attack
- Normal is the statistical mean
- Must also allow for variance to know what is abnormal

- On top of fancy modelling techniques:
  - Bayesian statistics
  - Linear discriminant analysis
  - Quadratic discriminant analysis

Detecting Anomalies - an Example

- Suppose we monitor use of three commands
  - `open`, `read`, `close`

Detecting Anomalies - an Example

- Suppose we monitor use of three commands
  - `open`, `read`, `close`
- Under normal use we observe Alice do:
  - `open`, `read`, `close`, `open`, `open`, `read`, `close`, ...

Detecting Anomalies - an Example

- Suppose we monitor use of three commands
  - `open`, `read`, `close`
- Under normal use we observe Alice do:
  - `open`, `read`, `close`, `open`, `open`, `read`, `close`, ...
- Four pairs are normal for Alice:
  - (`open`, `read`); (`read`, `close`); (`close`, `open`); (`open`, `open`)
- Can we use this to identify unusual activity?

Detecting Anomalies - an Example

Tactic: If the ratio of abnormal to normal pairs is established to be “too high”, warn of possible attack

Detecting Anomalies - an Example

Tactic: If the ratio of abnormal to normal pairs is established to be "too high", warn of possible attack

- Very inexpressive approach
- Can be improved by:
  - Use frequency of each pair
  - Use more than two consecutive commands
  - Include more commands/behavior in the model
  - More sophisticated statistical discrimination

Define the threshold to be .1

Over time, Alice has accessed file F_n at rate H_n:

H_0	H_1	H_2	H_3
.10	.40	.40	.10

Recently, Alice has accessed file F_n at rate A_n:

A_0	A_1	A_2	A_3
.10	.40	.30	.20

Detecting Anomalies - an Example

Define the threshold to be .1

Over time, Alice has accessed file F_n at rate H_n:

H_0	H_1	H_2	H_3
.10	.40	.40	.10

Recently, Alice has accessed file F_n at rate A_n:

A_0	A_1	A_2	A_3
.10	.40	.30	.20

- Is this normal usage for Alice?
- We can compute $S = \sum_{i=0}^{3}(H_i - A_i)^2 = .02$
  - $S < .1$ considered to be normal, so all ok

Detecting Anomalies - an Example

Define the threshold to be .1

Over time, Alice has accessed file F_n at rate H_n:

H_0	H_1	H_2	H_3
.10	.40	.40	.10

Recently, Alice has accessed file F_n at rate A_n:

A_0	A_1	A_2	A_3
.10	.40	.30	.20

- Is this normal usage for Alice?
- We can compute $S = \sum_{i=0}^{3}(H_i - A_i)^2 = .02$
  - $S < .1$ considered to be normal, so all ok
- **Question:** How do we account for variations during standard system usage?

Detecting Anomalies - an Example

- To allow for “normal” behavior to evolve organically, we update averages during usage: $H_n = 0.2A_n + 0.8H_n$

Detecting Anomalies - an Example

- To allow for “normal” behavior to evolve organically, we update averages during usage: $H_n = 0.2A_n + 0.8H_n$
- As such, we update as:
  - $H_2 = .2 * .3 + .8 * .4$
  - $H_3 = .2 * .2 + .8 * .1$

And thus:

H_0	H_1	H_2	H_3
.10	.40	.38	.12

Detecting Anomalies - rinse and repeat

Over time, Alice has accessed file F_n at rate H_n:

H_0	H_1	H_2	H_3
.10	.40	.38	.12

Recently, Alice has accessed file F_n at rate A_n:

A_0	A_1	A_2	A_3
.70	.10	.10	.10

- Is this normal usage for Alice?

Detecting Anomalies - rinse and repeat

Over time, Alice has accessed file F_n at rate H_n:

H_0	H_1	H_2	H_3
.10	.40	.38	.12

Recently, Alice has accessed file F_n at rate A_n:

A_0	A_1	A_2	A_3
.70	.10	.10	.10

- Is this normal usage for Alice?
- We can compute $S = \sum_{i=0}^{3}(H_i - A_i)^2 \approx .53$
  - $S > .1$ considered to be abnormal
  - Raise a red flag
  - ... or outright abort execution!

Anomaly Detection Issues

Constant evolution

- A static intrusion system places a huge burden on the admin
- But evolving IDS makes it possible to the attacker to manipulate the behavior and slowly convince IDS of an abnormal pattern
- Slow and steady can win the race

Anomaly Detection Issues

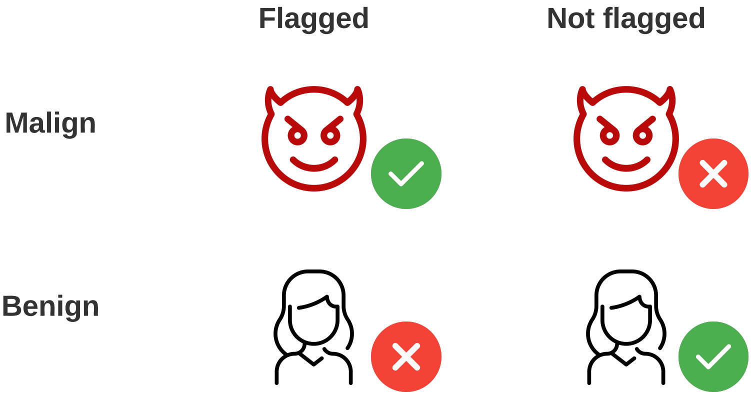
Constant evolution

- A static intrusion system places a huge burden on the admin
- But evolving IDS makes it possible to the attacker to manipulate the behavior and slowly convince IDS of an abnormal pattern
- Slow and steady can win the race

Types of IDS feedback

- Example: monitor failed login attempts
  - Burst of failures can occur - an attack?
  - ... or an admin that forgot his password?
- False positives (FP) - attack flagged when none is occurring
- False negatives (FN) - attack flagged as adequate behavior

Possible Outcomes



Base-Rate Fallacy

- Number of actual intrusions small when compared to the amount of data analysed
- *Base-rate fallacy* probability of some conditional event is assessed without considering the “base rate” of that event

Base-Rate Fallacy

- *Base-rate fallacy* probability of some conditional event is assessed without considering the “base rate” of that event

- Suppose an IDS is 99% accurate, 1% of FP/FN
- IDS generates 1,000,100 log entries
- Only 100 correspond to actual malicious events
- Because of the success rate, of the 100 malicious events, 99 will be detected as malicious **=** 1 **FN**
- Nevertheless, of the 1,000,000 benign events, 10,000 will be mistakenly identified as malicious = 10,000 FP
- Out of all 10,099 expected alarms, 10,000 are false alarms, **roughly** 99% **of all flagged attacks**

IDS Response Approaches

- A major part of the IDS has to do with how to detect intrusions
- How do we respond?

IDS Response Approaches

- A major part of the IDS has to do with how to detect intrusions
- How do we respond?

Approaches

- Preemptive blocking
- Infiltration
- Intrusion deterrence
- Intrusion deflection

IDS Response Approaches

Preemptive blocking

- Banishment vigilance
- Prevents intrusions before they occur
- Notes any sign of impending threats and blocks user of IP

IDS Response Approaches

Preemptive blocking

- Banishment vigilance
- Prevents intrusions before they occur
- Notes any sign of impending threats and blocks user of IP

Cons

- Risk of blocking legitimate users
- Attacker moves to a different machine

IDS Response Approaches

Infiltration

- Not a software methodology
- The process of infiltrating hacker/cracker online groups by security administrator
- Unusual, can backfire
- Admins depend on security bulletins
- More common with large-scale organizations (e.g. police)

IDS Response Approaches

Intrusion Deterrance

- Make the system a less palatable target
- Make the system seem less attractive
  - Hide the valuable assets
- Make the system seem more secure than it is
  - Have monitoring/security warnings
- Make any potential reward seem more difficult to attain than it actually is/might be

IDS Response Approaches

Intrusion Deflection

- Set up an attractive, but fake, system
- Subsystem with purpose to monitor/understand the attacker
- A.k.a. honeypots
- Lure the attacker into the system and monitor attacker activity and intrusion patterns

Honeypots

Decoy systems designed to

- Lure a potential attacker away from critical systems
- Gather data about the attacker's activity
- Encourage the attacker to stay on the system long enough for administrators to respond

- Systems are filled with fabricated information that a legitimate user of the system would not access
- Resources that have no production value
  - Incoming communication is, most likely, a probe, scan or attack
  - Initiated outbound communication suggests that the system has probably been compromised

Wrap up

Understanding system intrusion

- A wide range of threats
  - From expert cyber criminals
  - ... to script-kiddies

Wrap up

Understanding system intrusion

- A wide range of threats
  - From expert cyber criminals
  - ... to script-kiddies

Intrusion Detection Systems

- Intrusion countermeasure based on (potentially complex) patterns
- Mainly host-/network- based, depending on monitoring capabilities
- Different methodologies for detection
  - Signature Detection
  - Anomaly Detection
- Configuration requires nuanced understanding of system
- Responses to intrusion can also vary widely

Network Security - Week 8

João Soares

DCC/FCUP

2023