Lecture 13 - 11/01/2019

Denial of Service **2**
- Defences 2
- Response 2

Intrusion Detection **3**
- IDS 3
  - Requirements 3
- Models 4
  - Anomaly Model 4
  - Misuse Model 4
  - Specification Model 4
- Architecture 4
  - Agent 4
  - Director 4
  - Notifier 4

Denial of Service

Defences

- Limit/Remove ability to send packets with spoofed source addresses
    - This should be done at the ISP level as an ISP knows where requests should be coming from.
- Limit the rate at which certain packets can be sent
- Use SYN cookies
    - Keep the state on the requesters machine, not yours.
- Block the use of IP-directed broadcasts
    - This would prevent amplification attacks
- Use captchas or application level prevention
- Keeping your systems up to date and patch
    - This is a good practice anyway.
- Mirror and replicate your servers.
    - Load balancers/delivery networks/secondary standby servers

Response

- Must have a good incident response plan
- Who will you contact?
    - There will be a technical person that will be able to get your network back online
- How are you going to contact them?
    - If your system is down you might not be able to contact them using conventional methods
- Must have an automated network monitoring and intrusion detection system.
- Capture packets & analyze
- Consider tracing flow of packets back to the source
- Might need to switch to alternate servers
- Update response plan when attack is over

Intrusion Detection

- Software trespass
    - Virus, worm, trojan, durex
- User trespass
    - Masquerader: Outside person using a legitimate users credentials
    - Misfeasor: Insider that is doing illegitimate activities
    - Clandestine user: Could be either of the two, but they're trying to hide their actions.
- When an internal employee is fired, make an image of their hard drive

IDS

- Software that monitors and analyzes system events
- Attempts to detect unauthorized access.
- Users and software can be boiled down to a predictable pattern

Requirements

It must be/have:

- Fully automated
    - Run without human interaction or configuration
- Able to recover from crashes or failures
- Resists subversion
    - It has to monitor itself for tampering
- Minimal overhead
- Configurable
    - Configurable to conform with your system’s security policies
- Adaptable
    - Users will change suddenly (promotion/fired/e.t.c) and the system must be able to adapt to these changes
- Scalable
- Degrade gracefully
    - If some parts stop working, the rest should still work as expected
- Dynamically reconfigurable
    - If security policy or user behaviour has changed, you should be able to change configuration on the fly.
- Detect a wide variety of attacks
- Detect in near real time
- Alert people in a clear manner
- Accurate

Models

Anomaly Model

- Has an idea of what a normal system looks like
- Anything that deviates from what the normal system should be is flagged as an attack
- “Here is my good system, anything that isnt my good system is bad”
- System has to be free of attack while building a model of what a good system should look like.

Misuse Model

- Any action that is associated with bad behaviour
- An example is a having a virus signature and scanning for that signature
- You have to know what the bad behaviours are
- Open to zero-day attacks

Specification Model

- Not associated with users, but software
- A program will have specifications for exactly what it is supposed to do
- If a program acts outside these specifications, flag it.

Architecture

Normally there are 3 components

Agent

- Gathers information from various service
- Host network,

Director

- How much information should come from the agent
- Makes decisions

Notifier