Secure Software Programming and Vulnerability Analysis

Christopher Kruegel chris@auto.tuwien.ac.at
http://www.auto.tuwien.ac.at/~chris

Operations and Denial of Service

Overview

- Security issues at various stages of application life-cycle
  - mistakes, vulnerabilities, and exploits
  - avoidance, detection, and defense
- Architecture
  - security considerations when designing the application
- Implementation
  - security considerations when writing the application
- Operation
  - security considerations when the application is in production

Overview

- Separation of development and operations staff
  - people are unaware of problems and risks in the other domain
  - for example, a developer considers the OS and network secure
- Running secure applications on insecure OS, or vice versa
- Attackers choose path of least resistance
  - go for the underlying infrastructure if easier
- Ensure that application can be deployed in a safe environment

➢ Security is everybody’s problem

Operations

- Besides direct access to applications, attacker can try alternative paths
- Administrative access can be a problem
  - standard remote access (e.g., ssh, telnet)
  - usually reachable from within the whole enterprise
  - convenient
  - often not as well protected
  - attacker can obtain access at the OS level and circumvent application defense
  - user-level access at OS level is a problem too

Operations

- Besides direct access to applications, attacker can try alternative paths
- Administrative access can be a problem
  - standard remote access (e.g., ssh, telnet)
  - usually reachable from within the whole enterprise
  - convenient
  - often not as well protected
  - attacker can obtain access at the OS level and circumvent application defense
  - user-level access at OS level is a problem too

Operations

- Good practice takes a holistic approach
  - all aspects are equally important

1. Secure the network
2. Secure the operating system
3. Deploy application with diligence
4. Follow good operational practice

Secure the Network

- Allow essential network services only
  - good firewall configuration
  - be careful when multiple interfaces are in use

- Use secure protocols
  - obviously, no clear text protocols
  - administrative access should be at least as secure as application

- Separate production data from management data
  - use two separate networks
  - also good in case of denial of service attacks

Secure the Network

- Monitor for unauthorized activities
  - deploy intrusion detection systems
  - at least, on network level (e.g., Snort)
  - if you monitor bad behavior, don't flame the source immediately
    could be spoofed source, or misconfigurations

- Defense in depth
  - use multiple layers of defense
  - firewall, tightened switches, IDS, personal firewalls

- Log events
  - detection, but also accountability and forensics
  - log on dedicated (hardened, stealth) machine

Secure the Operating System

- Secure baseline
  - after initial installation, harden the OS
  - turn off unwanted network services
    - remove daemons from startup scripts
    - local firewall
  - tighten file access control
    - use principle of least privilege
  - remove unwanted binaries
    - no compiler on a Web server
  - install latest patches
  - make installation process repeatable

Deploy Application with Diligence

- Set up correct file permissions
  - especially for configuration files
- Enable event logging
  - make sure that someone reads these logs
  - send regularly an email summary to administrator
- Use compartmentalization
  - `chroot()` is common
  - privilege separation with different users
- Also applies to third-party code

Good Operational Practice

- Manage privileges
  - use different roles, users, and groups
  - developers, users, and operational staff can get different privileges
- Manage user accounts
  - centralized account management
  - also check for application / database accounts
- Treat temporary or contract personal appropriately
  - shared accounts for all temporaries results in loss of accountability

Good Operational Practice

- Configuration and patch management
  - use standardized configuration tools and procedures
  - not only consider reliability and stability an issue
  - patch also production machines

- Test your configuration
  - changes to configurations and patches might break applications
  - previously test these changes
  - separate test network is convenient
  - if too expensive, use virtual machine software (`VMware`, `bochs`)

Good Operational Practice

- Conduct backups securely
  - doing backups is vital for every data center
  - storing the backups off-site is even better
  - but, the data needs to be transported and stored securely

- Threat and risk analysis
  - who could attack, how could the attack happen, what are the assets

- Incident handling plans
  - what happens in case of an attack
  - backup systems, shut down operations

Good Operational Practice

- Stay current
  - invest time to familiarize yourself with security issues
- Perform audits
  - code reviews
  - penetration tests
  - request external opinions
- Avoid mission creep

  “every firewall become useless after some time as more and more rules are added”
- Don’t pass the buck or do shortcuts because it is easier

Operations
and
Denial of Service

Denial of Service

- Definition
  - explicit attempt by attackers to prevent legitimate users of a service from using that service
  - not all service outages (even those that result from malicious activity) are necessarily denial of service attacks
- Examples
  - attempts to "flood" a network, thereby preventing legitimate network traffic
  - attempts to disrupt connections between two machines, thereby preventing access to a service
  - attempts to disrupt service to a specific system or person

Denial of Service

- Impact
  - disable computer or network
  - depending on organization, disabling complete organization
- Asymmetric denial of service (DoS)
  - attacker uses only limited resources against a large victim
- Modes of Attack

  1. consumption of scare, limited, or non-renewable resources
  2. destruction or alteration of configuration information
  3. physical destruction or alteration of (network) components

Denial of Service

1. Consumption of scare, limited, or non-renewable resources
   - computers and networks require certain things to operate properly: CPU time, bandwidth, memory, access to other computers, and environmental resources (e.g., power)

   1. network connectivity
      - consume entries in the receive queue (SYN attack)
   2. consume bandwidth
      - send a lot of packets
   3. use victim resources against itself
      - connect chargen and echo services
      - smurf attack

Denial of Service

4. fill file system with data or files (to use up inodes)
   - anonymous ftp servers
   - systems without quota
5. generate excessive amount of log entries
6. generate excessive amount of mail messages
7. generate excessive amounts of processes
   - fork bombs
8. exploit lock-out scheme
   - account disabling after a few attempts
9. sending input that crashes OS or applications
   - WinNuke

Denial of Service

2. Destruction or alteration of configuration information
   - change router information
   - change Windows Registry information

3. Physical destruction or alteration of (network) components
   - cut wires
   - blow up NOC (network operation center)

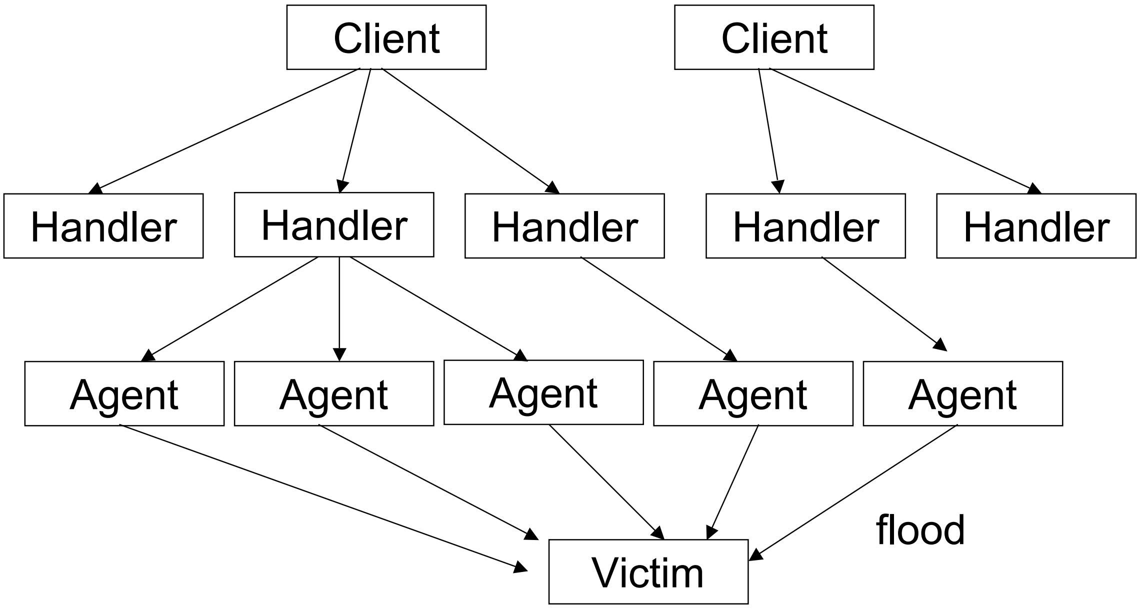
Denial of Service

- Many tools for DoS available
- Especially for distributed denial of service (DDoS)
- Distributed denial of service
  - many coordinated attackers overflow one victim
  - Trinoo, Stacheldraht, Tribal Flood Network (TFN)

- Stacheldraht
  - involved hosts:
    - client hosts: are used to control handlers (1:n relationship)
    - handler hosts: are used to control agents (1:n relationship), n < 1000
    - agent hosts: send the ICMP echo request to the victim
  - all communication is encrypted (TCP + ICMP)

`http://staff.washington.edu/dittrich/misc/stacheldraht.analysis`

Denial of Service



Client

Client

Handler

Handler

Handler

Handler

Handler

Agent

Agent

Agent

Agent

Agent

Victim

flood

Denial of Service

- Defense mechanisms
  - difficult to do locally
  - especially with spoofed source addresses and changing content
  - traffic shaping
    - rate limit incoming traffic
    - use well-configured firewalls
  - infrastructural techniques
    - cooperating routers
    - push back
    - path identification
  - client puzzles
    - client has to solve a resource intensive task to continue communication

Denial of Service

- Syn cookies
  - technique to prevent syn floods
  - particular choice of initial 32 bit TCP sequence number
  - top 5 bits
    - t mod 32, where t is a 32-bit time counter that increases every 64 seconds
  - next 3 bits
    - an encoding of an MSS selected by the server in response to the client's MSS
  - bottom 24 bits
    - a server-selected secret function of the client IP address and port number, the server IP address and port number, and t.
  - no “receive queue” needed anymore
  - when second packet from client is received (finishing 3-way handshake), just check for validity of `ack` value

Summary

- Operations

  1. Secure the network
  2. Secure the operating system
  3. Deploy application with diligence
  4. Follow good operational practice

- Denial of service
  - explicit attempt by attackers to prevent legitimate users of a service from using that service

  1. consumption of scare, limited, or non-renewable resources
  2. destruction or alteration of configuration information
  3. physical destruction or alteration of (network) components