CIS 505: Software Systems
Lecture Note : Security

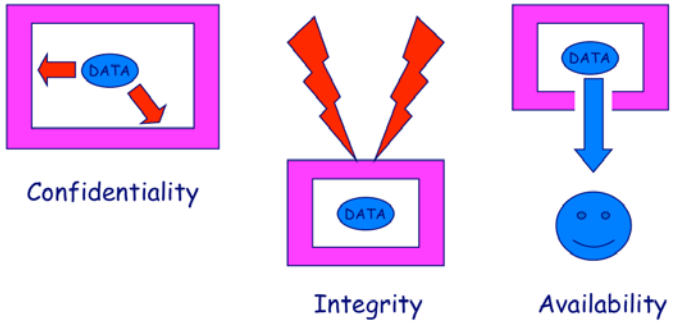
Insup Lee
Department of Computer and Information Science
University of Pennsylvania

CIS 505, Spring 2007

What is Security?

- Dictionary Definition: protection or defense against attack, interference, espionage, etc.
- Computer Security Classification:
  - Confidentiality (or Secrecy)
    - Protecting against unauthorized data disclosure and ensuring the authenticity of the data's source
  - Integrity
    - Preventing unauthorized data modification
  - Availability (or Necessity)
    - Preventing data delays or denials (removal)

Goals of Security



Source: GUNTER

Terminology

- Vulnerability (weakness/defects that can be exploited)
  - Ill-chosen passwords
  - Software bugs
  - Communication without encryption
  - Incorrect set-ups
- Attack (ways of exploiting vulnerability)
  - Password crackers
  - Viruses and worms
  - Denial of service
- Intruders (adversaries that try to attack)
  - Terrorists
  - Espionage
  - Hackers

Security Goals

- Data Confidentiality
  - Keep data and communication secret
  - Privacy of personal financial/health records, etc.
  - Military and commercial relevance
- Data Integrity
  - Protect reliability of data against tampering
  - Can we be sure of the source and content of information?
- System Availability
  - Data/resources should be accessible when needed
  - Protection against denial of service attacks

Security threats

- Interception
- Interruption
- Modification
- Fabrication

Security Policy

- Security policy is a written statement describing what assets are to be protected and why, who is responsible, which behaviors are acceptable or not.
- The policy addresses
  - Physical security
  - Network security
  - Access authorizations
  - Virus protection
  - Disaster recovery

Specific Elements of a Security Policy

- Authentication
  - Who is trying to access the site?
- Access Control
  - Who is allowed to logon and access the site?
- Secrecy
  - Who is permitted to view selected information
- Data integrity
  - Who is allowed to change data?
- Audit
  - What and who causes selected events to occur, and when?

Security mechanisms

- Encryption
- Authentication
- Authorization
- auditing

The News

BBC used to entice cyber victims

People are being warned about spam e-mails containing BBC News stories designed to trick them into visiting malicious websites.

Cyber criminals are using the messages to exploit a recently discovered flaw in Microsoft's Internet Explorer.

If users click on the link, they are taken to a fake website that installs

The e-mails direct users to a fake BBC News website

Exploit for Vulnerability in Microsoft Internet Explorer

added March 22, 2006 | updated March 27, 2006

US-CERT is aware of an active exploitation of a vulnerability in the way Microsoft Internet Exp

February 27, 2006

Cyberthieves Silently Copy Your Passwords as You Type

By TOM ZELLER Jr.

Most people who use e-mail no

Public Exploit Code for a Vulnerability in Apple Safari Browser

added February 21, 2006

US-CERT is aware of publicly available exploit code for a vulnerability in Apple Safari Browser. The Apple Safari browser will automatically open "safe" file types, such as pictures, movies, and archive files. A system

Some Statistics …

- It is estimated that PC Viruses cost businesses approximately $55 Billion in damages in 2003
- Sapphire/Slammer SQL worm required roughly 10 minutes to spread worldwide making it by far the fastest worm to date.
- 3780 Software vulnerabilities reported in 2004

Stats from www.cert.org and www.securitystats.com

Software Vulnerabilities

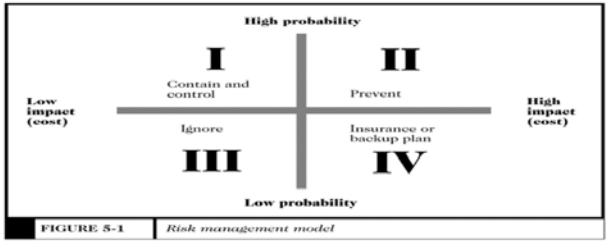
- Everyday we read about new software vulnerabilities in the news
- On 2nd April, 2005 :
  - Microsoft Windows HTML Help ActiveX Control Cross-Domain Vulnerability
  - Multiple Denial of Service Vulnerabilities in Cisco IOS
  - Multiple Vulnerabilities in Microsoft Windows Components
  - Visit **www.cert.org** for the latest

More Analysis...

Stats from http://nvd.nist.gov/

Risk analysis

- Countermeasures are procedures, either physical or logical, that recognize, reduce, or eliminate a threat



What do we mean by security?

- What is a secure program?
- What is computer security?

When is a program secure?

- When it does exactly what it should?
  - Not more.
  - Not less
- But how do we know what a program is supposed to do?
  - Somebody tells us? (But do we trust them?)
  - We write the code ourselves? (But what fraction of the software you use have you written?)

When is a program secure?

- 2nd try: A program is secure when it doesn't do something it shouldn't.
- Easier to specify a list of "bad" things:
  - Delete or corrupt important files
  - Crash my system
  - Send my password over the Internet
  - Send threatening e-mail to the present posing as me
  - How do you specify all "bad" things?
- And… what if most of the time the program doesn't do bad things, but occasionally it does? Is it secure?

When is a program secure?

- Claim: Perfect security does not exist.
  - Security vulnerabilities are the result of violating an assumption about the software (or, more generally the entire system).
  - Corollary: As long as you make assumptions, you're vulnerable.
  - And: You *always* need to make assumptions!
- Example: Buffer overflows
  - Assumption (by programmer) is that the data will fit in the buffer.
  - This leads to a vulnerability: Supply data that is too big for the buffer (thereby violating the assumptions)
  - Vulnerabilities can be exploited by an attack.

When is a program secure enough?

- Security is all about tradeoffs
  - Performance
  - Cost
  - Usability
  - Functionality
- The right question is: how do you know when something is secure enough?
  - Still a hard question
  - Requires understanding of the tradeoffs involved
- Is Internet Explorer 6 secure enough?
  - Depends on context

Attacks

- OS Security
  - Trapdoors, Trojan Horse, Buffer Overflow and its solutions
- Network Security
  - Worms, Morris internet worm
  - Viruses
    - Kinds of viruses (Replication and Payload views)
    - Bootstrap Viruses, Melissa macro virus, Antivirus Techniques

How to think about tradeoffs?

- What is it that you are trying to protect?
  - Music collection vs. nuclear missile design data
- How valuable is it?
- In what way is it valuable?
  - Information may be important only to one person (e.g. private e-mail or passwords)
  - Information may be important because it is accurate and reliable (e.g. bank's accounting information)
  - A computer system may be important because of a service it provides (e.g. Google's web servers)

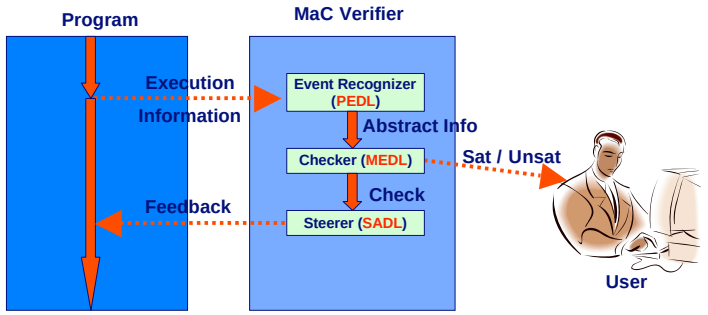
Security Techniques

- Cryptography
  - Can ensure confidentiality and integrity
  - Typically used for authentication
- Firewalls, passwords, access control
  - Authorization mechanisms
- Java bytecode verifier
  - Memory safety against malicious/defective code

Downloaded software

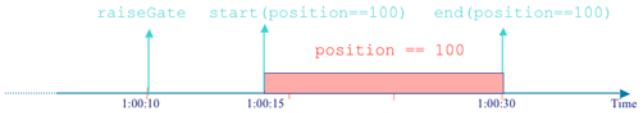
- Sandboxing: encapsulate programs in a box but be liberal on what to accept
  - Java sandbox confines Java applet actions to a security model-defined set of rules
  - Rules apply to all untrusted applets, applets that have not been proven secure
- Verification: analyze code before executing but then minimize runtime checks
  - proof-carrying code
- Certification: trust someone else to analyze code and execute with no checking
  - Signed Java applets contain embedded digital signatures which serve as a proof of identity

Java-MaC (Monitoring and Checking)



[Kim, Viswanathan, Kannan, Lee, Sokolsky, FMSD 2004]

Design of the MaC Languages



- Must be able to reason about both time instants and information that holds for a duration of time in a program execution.
  - Events and conditions are a natural division, which is also found in other formalisms such as SCR.
  - Conditions, which are true or false for a finite duration of time (e.g., is variable x >5?)
  - Events, which are either present or absent at some instant of time (e.g., is the control right now at the end of method f?).
- Need temporal operators combining events and conditions in order to reason about traces.

Logical Foundation

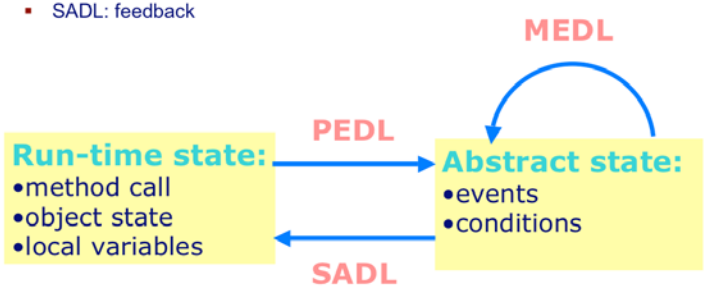
$$C ::= c \mid \text{defined}(C) \mid [E_1, E_2) \mid \neg C \mid C_1 \vee C_2 \mid C_1 \wedge C_2$$
$$E ::= e \mid \text{start}(C) \mid \text{end}(C) \mid E_1 \vee E_2 \mid E_1 \wedge E_2 \mid E \text{ when } C$$

- Conditions interpreted over 3 values: true, false and undefined.
- [., .) pairs a couple of events to define an interval.
- `start` and `end` define the events corresponding to the instant when conditions change their value.

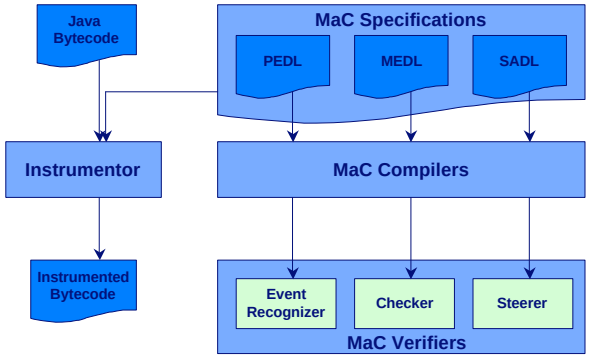
[Lee et al, ICPDP 99]

The MaC languages

- PEDL: abstraction
- MEDL: abstract transformation
- SADL: feedback



Java-MaC Framework



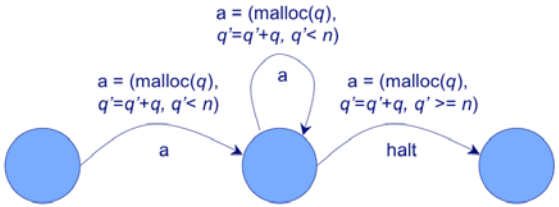
Similar techniques, different purposes

- Check security policy
  - Security automata, edit automata
  - Model-Carrying Code (MCC)
- Intrusion detection
  - Extract from the target program to ensure that the program has not been tampered
  - Signature-based approach

Security Policy in Security/Edit Automata

Example (modified from [BLW02]): Limit the amount of memory that an application can allocate for itself

Property: application must not allocate memory more than *n*



Must not allocate more than n

PEDL

```
export event mallocCall;
monmeth int malloc(int);
event mallocCall = startM(malloc(int));
```

MEDL

```
import event mallocCall;
import action halt;
var int memory;
alarm violateMemoryPolicy = end(memory < 1000);
mallocCall -> {  // value(mallocCall,0) returns arg of malloc()
   memory' = memory + value(mallocCall,0);
}
violateMemoryPolicy -> { invoke(halt); }
```

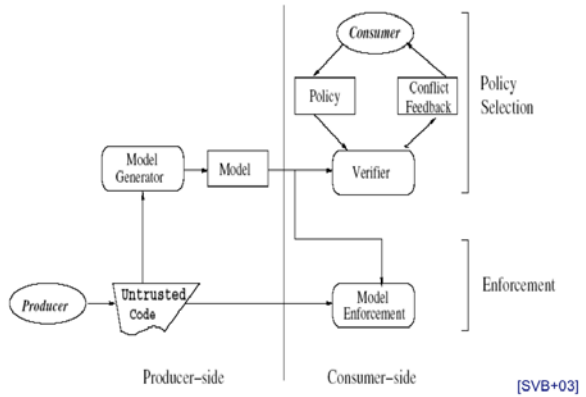
SADL

```
steering action halt =  // exit before next malloc() call
 { call System.exit(); } before call malloc(int);
```


Model-Carrying Code (MCC)

- How can we run untrusted code on our machine?
  - Untrusted code comes with a model of its security-relevant behavior
  - Users have their own security policies
  - Employ two types of checking
- Static checking: to ensure that untrusted program's model respects user's security policy
  - Use model checking to check that Beh(Model) are in Beh(Policy)
- Run-time checking: to ensure that program behaves as specified by model
  - Use runtime checking with
    - Model is a specification (Automata)
    - Events are system calls

MCC Framework



[SVB+03]

Security Techniques

- Protection from Covert channels
  - Shielding the computer to prevent interception and subsequent interpretation of electromagnetic radiation
- Social aspects of security
  - Controlling who is allowed to make changes to a computer system (both its hardware and software)
- Physical aspects of security
  - Laptop theft at UC Berkeley

Some Possible Scenarios

- Alice buys a book from Bob's book store.
- Inter-corporate trading for Charlie's Plastic Company.
- Daisy electronic market.
- …

Alice Buys a Book

- Alice shops for a book on the internet using WWW.
- She finds the desired book from Bob's book store and makes the order using a web form provided by Bob's.
- Bob confirms that the order really comes from Alice's.
- She sends her credit card number, suitably encrypted.
- The book is delivered through UPS.

Inter-Corporate Trading

- Charlie's Plastic Makers is a medium-sized company in Canada with long-established requirements for high-quality plastic which it buys from Plasticorp.
- Plasticorp aims to reduce costs of customer transactions by using secure messaging with its regular customers.
- Origin and confidentiality of all correspondence must be ensured.

Daisy's Electronic Market

- Daisy is an entrepreneurial small businessperson who works from her home basement.
- She buys items from suppliers willing to do business wholly electronically, repackages them, and sells them through a WWW storefront.
- Effective marketing of the web page and very low overhead provide Daisy's competitive edge.

What are the issues?

- **Accountability** -- Security relevant activities on a system can be traced to individuals who may be held responsible for their actions
- **Availability** -- System resources are safeguarded from tampering and are available for authorized users at the time and in the format needed
- **Access Control** -- Access to the system resources is limited to authorized individuals, entities, or processes
- **Confidentiality** -- Information is not accessed by or disclosed to unauthorized individuals, entities, or processes
- **Identification and Authentication** -- Verification that the originator of a transaction is the originator
- **Integrity** -- Information is not undetectably altered or destroyed by an unauthorized person or process
- **Non-repudiation** -- Undeniable proof of participation by the sender and/or receiver in a transaction
- **Privacy** – individual rights to nondisclosure