Midterm Review - Part 1

Security Concepts

Vulnerabilities, Threats, Attacks, and Controls

Know the basic terms and definitions:

1. Vulnerability
2. Threat (within threat, we also consider Method, Opportunity, and Motive)
3. Attack
4. Control

Types of Attacks

Know the four types of attacks and be able to give examples:

1. Interception
2. Interruption
3. Modification
4. Fabrication

Security Goals — CIA

Know the three primary security goals

1. Confidentiality
2. Integrity
3. Availability

Some add *Non-repudiation* and *Assurance*

Apply the CIA concepts to specific examples.

Case Study: Heartbleed

Describe the vulnerability, threat, attack, and control.

What security goals were violated?

Case Study: The Debian Fiasco

Describe the vulnerability, threat, attack, and control.

What security goals were violated?

Vulnerabilities

Types of Vulnerabilities

Relate classification of attacks (interception, interruption, modification, fabrication) to system assets (hardware, software, and data)

Attacks on Software

Buffer Overflow Attack (basics)

What is it? Give an example.

How do you prevent it?

Incomplete Mediation

What is it? Give an example.

How do you prevent it?

Time-of-check to Time-of-use Errors (or Race Errors)

What is it? Give an example.

How do you prevent it?

Side Channel Attacks

What is a side channel attack?

Give an example of a “side channel” and how it might be exploited.

How might you prevent exploitation of a side channel vulnerability?

Standards

Interoperability, assurance of market share. Important for security *because it is hard!*

The Internet Organization

Internet Architecture Board

Internet Engineering Steering Group

Internet Engineering Task Force (IETF)

Internet Drafts are proposed standards; Requests for Comment (RFCs) are published standards.

RFCs — not just security, but many Internet standards.

National Institutes of Standards and Technology (NIST)

Publish Federal Information Processing Standards (FIPS) and Special Publications (SP)

International Telecommunications Union (ITU)

"is responsible for studying technical, operating, and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis."

Stack-Buffer Overflow

Attacking the in/out package

Why is the in/out package "interesting" from a security perspective?

Owner and permissions
User Input

Basic fuzzing with Python

Find the vulnerability

Which function in the in/out package causes the buffer overflow?
Which C library function should the programmer *not* have used and why? What is an alternative that would have been better?

What does the stack look like (roughly) in the vulnerable function?

Exploitation Challenges

Knowledge of stack frame location

Where *exactly* is the function's return address?
Where *exactly* can we locate malicious code?

String processing

Malicious code must survive string processing

Exploit Components

There are three components of a basic stack-buffer overflow attack. All three components are part of a single string that will be passed to the vulnerable program as user input.

Shellcode (see sample)

What is the purpose of shellcode?

What are the two major constraints facing a shellcode writer?

Return Address

Why might an attacker include multiple copies of the return address?

NOP Sled

What is a NOP?

What is the purpose of the NOP sled (block of NOPs before the shellcode)?

Stack-Buffer Overflow Protection

Buffer overflow recap

Attacker must:

1. Overwrite values on stack
2. Execute the code on the stack.
3. Predict location of code in memory.

Modern systems protect against these things.

Stack Protection

Goal is to prevent attacker from overwriting anything important (esp. return address)

Terminator Canaries

What is a terminator canary? How is it constructed?

What type of overflow does it protect against? How?

Some weaknesses with terminator canaries

May not protect against non-string based overflow attacks

May be defeated if multiple overwrites are possible — one overwrite to modify control information on the stack, a second to fix-up the canary

Random Canaries

What is a random canary? How is it constructed?

What type of overflow does it protect against? How?

Some weaknesses with random canaries

Local variables *may* not be protected

Function arguments *may* not be protected, at least within the vulnerable function

Attacker may be able to retrieve or guess random canary value (low entropy). Some bits of canary may be derived from “guessable” sources

What overflow protection does Linux provide? How is it enabled or disabled?

Stack Execution Protection

What is it? Where is it supported — CPU or OS?

How does Linux implement stack execution protection? How is it enabled or disabled?

Some weaknesses with stack execution protection

CPU may not support it (older CPUs) or it may be disabled in BIOS
May not be enabled in virtualized environment

Attackers have developed workarounds

Return-to-libc / return-to-system — describe and indicate how it bypasses stack execution protection

Address Space Layout Randomization

What is the purpose of ASLR? What different types of ASLR are there?

What types of ASLR are supported in Ubuntu Linux? Which types of ASLR did you implement in Project 1?

Who can disable ASLR on Linux?

Classification of Malicious Software - Propagation

Viruses

Replicate by modifying other files or programs

Require *user assistance* to replicate

Give an example

Trojan Horse

Masquerades as useful or desirable software, enticing users to install

Includes malicious functionality

Give an example

Worms

Spread *without* injecting code into other applications

Typically spread *without user assistance*

Give an example

Trapdoors (or Backdoors)

Method to obtain access that bypasses usual authentication measures

A type of *Insider Attack*; may be malicious or benign

Example: Ken Thompson's *Reflections on Trusting Trust*. (linked from the Lecture 6 web page)

Logic Bombs

Code created to take destructive action given a specific *trigger*

Another form of *Insider Attack*

Give an example

Viruses in Depth

Virus Types

File viruses — what are they? Give an example.

Macro viruses — what are they? Give an example.

Boot sector viruses — what are they? Give an example.

Virus Signatures

What is a virus signature?

How are signatures used to defend against viruses?

Given a ClamAV signature, describe how it is used

Encrypted, Polymorphic, and Metamorphic Viruses

Why would a virus writer encrypt the virus code?

What limitations are there on encryption of the code?

What is a *Polymorphic Virus*?

A *Metamorphic Virus* attempts to defeat signature recognition by re-writing its own code. Know the following methods for re-writing code:

1. Garbage code insertion
2. Register Use Exchange
3. Code Block Permutation / Jump Insertion
4. Code Integration

Worms

Worms can spread without user action.

Example: The Morris Worm, 1988 (http://en.wikipedia.org/wiki/Morris_worm)

What vulnerabilities did the worm exploit in order to spread?

What design flaw turned this “experiment” into a denial-of-service?

Example: Stuxnet, 2010 (http://en.wikipedia.org/wiki/Stuxnet)

A *zero-day vulnerability* is a vulnerability that was previously unknown to the general public. What was special about Stuxnet with regard to zero-days?

Stuxnet targeted a Siemens Programmable Logic Control (PLC). Why is this a concern? Recall the TED video.

How was Stuxnet able to install device drivers that were trusted by Windows?

Was Stuxnet harmful?