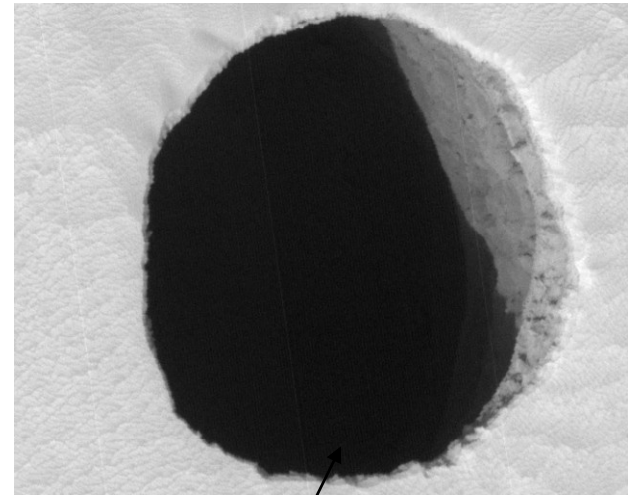


CSCD 396

Essential Computer Security

Fall 2009



Security Hole

Lecture 5 - Desktop Security

Vulnerabilities

Overview

- Learning Objectives
 - Understand OS Vulnerabilities
 - Windows vs. Linux vs. Mac
 - What are they
 - Why do they happen
 - Users - Passwords

Comparing Operating Systems

- Researchers have spent a lot of time studying vulnerabilities in operating systems
 - Which is better? Linux vs. Windows vs. Mac? Who has the fewest serious vulnerabilities?
- Other metric used -- how many successful attacks on a particular OS

Vulnerabilities

<http://blogs.zdnet.com/security/?p=758>

- Recent years, lots of comparisons
 - 2007 brought improved security with Windows Vista and Mac OS X Leopard (10.5)
 - Compiled security flaws in Mac OS X and Windows XP and Vista and placed them side by side
 - Vulnerability statistics from third party vendor Secunia and broke them down by Windows XP flaws, Vista flaws, and Mac OS X flaws

Table of Flaws Windows vs. Mac

Windows XP, Vista, and Mac OS X vulnerability stats for 2007

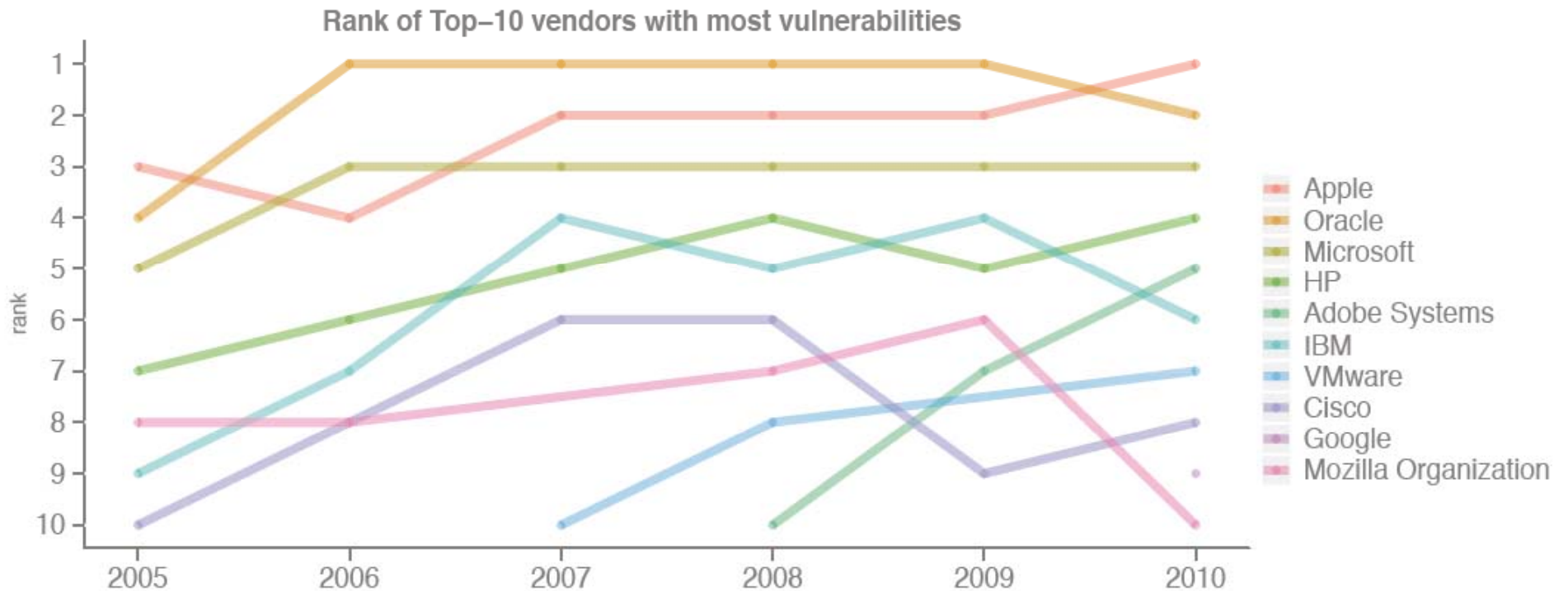
	XP	Vista	XP + Vista	Mac OS X
Total extremely critical	3	1	4	0
Total highly critical	19	12	23	234
Total moderately critical	2	1	3	2
Total less critical	3	1	4	7
Total flaws	34	20	44	243
Average flaws/month	2.83	1.67	3.67	20.25

Analysis of Data



- Apple had more than 5 times number of flaws per month than Windows XP and Vista in 2007
 - Most of these flaws were serious
 - This seems to go against conventional wisdom
- Noteworthy ...
 - Windows Vista showed fewer flaws than Windows XP, added Windows Defender and Sidebar added 4 highly critical flaws to Vista that weren't present in Windows XP

Things haven't gotten better for the mac...



Data from Competition to Support this View



- CanSecWest conference 2008
 - "PWN2OWN 2008" contest, where three laptop computers each equipped
 - Mac OSX 10.5.2, MS. Windows Vista Ultimate SP1, and Ubuntu 7.10 (linux)
 - Try to withstand some brilliant "hackers"
 - First person succeed take the laptop he/she breaks plus a cash prize

[http://blog.loaz.com/timwang/index.php/2008/03/30/
security_vulnerability_showdown_mac_os_v](http://blog.loaz.com/timwang/index.php/2008/03/30/security_vulnerability_showdown_mac_os_v)

Competition Data

- **Day 1**, under condition that only default OS could be targeted
 - No security breach for all three machines
- **Day 2**, applications like email clients, browsers were allowed targeting
 - Guess who was the first to get compromised?
 - First to go down, new MacBook Air with OSX 10.5.2, due to an undisclosed Safari vulnerability

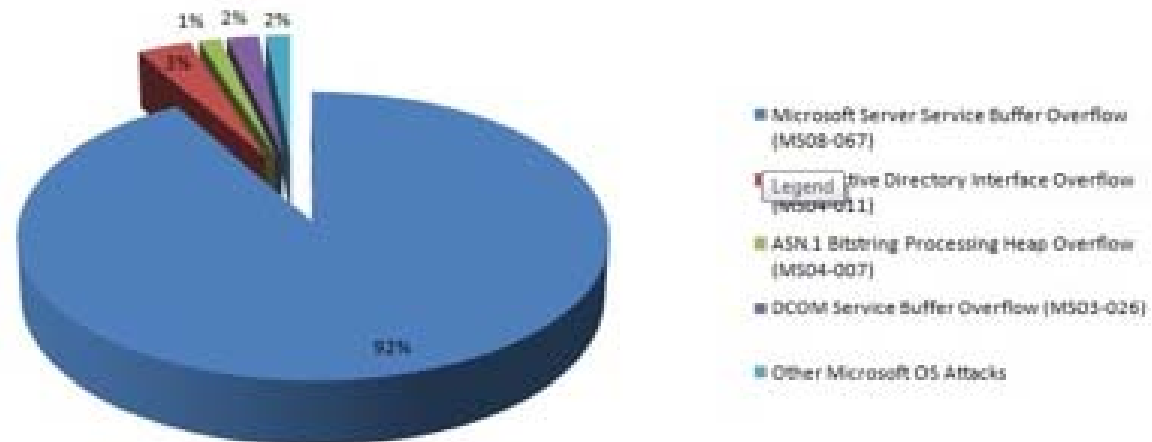
Competition Data

- **Day 3**, Windows Vista went down
 - Contest rule was even more relaxed where more "popular applications" installed on laptops
 - Previously unknown flaw in Adobe's Flash software, thus the Windows Vista fell...
- Sony Vaio laptop running Ubuntu remained "un-compromised" at end

Microsoft Vulnerabilities

- September 2009
- For past six months, over 90% of the attacks recorded for Microsoft targeted the buffer overflow vulnerability described in the Microsoft Security Bulletin MS08-067

Microsoft OS Attack % For Vulnerabilities



Buffer Overflow MS08-067

- Buffer overflow vulnerability in the Windows Server service
 - For systems running Windows 2000, XP, and Server 2003, a remote, unauthenticated attacker could exploit this vulnerability
 - Vista, attacker would need to be authenticated
 - Since the Server service runs with SYSTEM privileges, an attacker could take complete control of a vulnerable system
 - This IS the vulnerability that conficker exploited!

Why look at Comparisons?

- Good to question the myths surrounding security of OS's
 - You can make more informed buying decisions
 - You learn if you need to be more proactive if you are running a certain OS
 - Do you need to add antivirus, a firewall, spyware or other additional programs to protect your machine?

OS Vulnerabilities

- Look at details of OS vulnerabilities
 - 1. Buffer Overflow
 - Already touched on it
 - 2. Unvalidated input
 - 3. Race conditions
 - 4. Access-control problems
 - 5. Weaknesses in authentication, authorization

Buffer Overflow

- Every program that allows input
 - Needs to store the input in memory until it can use for its intended purpose
 - Examples: Web form, enter your name
Saving a file, enter the file name,
Search engine, enter the search string

Buffer overflow

- Although a program should check user input to make sure appropriate for purpose intended
 - For example, to make sure that a filename does not include illegal characters and does not exceed the legal length for filenames
 - Frequently the programmer does not bother
Programmer assumes that user will not do anything unreasonable

Buffer Overflows

- How are buffer overflows used to compromise your computer?
 - As part of long data input, attacker will include some of his own code
 - Then, he manipulates flow of the program in memory to execute his code
 - If the program he is overflowing is running with administrator privileges, his code has administrator privileges
 - Then, he can do anything to your computer

Unvalidated Input Attacks

- Any input received by your program from an untrusted source is a potential target for attack
- Hackers look at every source of input to the program and attempt to pass in malformed data of every type they can imagine
 - Called “Fuzzing Input”
- If the program crashes or otherwise misbehaves, the hacker then tries to find a way to exploit the problem

Race Condition



- A race condition exists when two events can occur out of sequence
- If correct sequence is required for the proper functioning of the program, this results in a bug
- If attacker can cause correct sequence not to happen and insert malicious code, change a filename, or otherwise interfere with the normal operation of the program, the race condition is a **security vulnerability**
- Attackers can sometimes take advantage of small time gaps in the processing of code
 - Interfere with the sequence of operations
 - Which they then exploit

Race Conditions

- There are two basic types of race condition that can be exploited
 - Time of check–time of use
 - Interprocess communication

Race Condition: Time of Check

- Application checks some condition before undertaking an action
- For example, it might check to see if a file exists before writing to it
 - An attacker, by continuously running a program that creates a new temporary file can create file in gap between when application checked to make sure temporary file didn't exist and when it opens it for writing
 - Application then opens attacker's file and writes to it ... system routine opens an existing file if there is one, and creates a new file only if there is no existing file

Race Condition: Interprocess Communication

- Separate processes—either within a single program or in two different programs—sometimes have to share information
 - For example, if two processes share same data, potential attacker to alter the data after one process sets it but before the other reads it
 - Solution to race conditions of this type is to use some locking mechanism to prevent one process from changing a variable until another is finished with it.

Access Control

- Many security vulnerabilities are created by the careless or improper use of access controls, or by the failure to use them at all
 - Many exploits involve an attacker somehow gaining more privileges than they ought to have
 - Privileges, also called permissions, are access rights granted by the operating system
 - Controls who is allowed to read and write files, see directories, execute a program

Access Control

- Of particular interest to attackers is the gaining of root privileges!!
 - Unrestricted permission to perform any operation on the system
 - An application running with root privileges can access everything and change anything
 - Many security vulnerabilities involve programming errors that allow an attacker to obtain root privileges
 - Some such exploits involve taking advantage of buffer overflows or race conditions ...

Authentication and Authorization

- Access control enforced by applications, can require a user to authenticate before granting authorization to perform an operation
- Authentication can involve requesting a user name and password, the use of a smart card, a biometric scan, or some other method
- Saw this with Vista – UAC method

Users as Vulnerabilities

- Often weakest link in the chain of security features protecting a user's data and software is the user himself, really?
 - As buffer overflows, race conditions, and other security vulnerabilities are eliminated from software
 - Attackers increasingly concentrate on fooling users into executing malicious code or handing over passwords, credit-card numbers, and other private information

Users as Vulnerabilities

- In February of 2005, a large firm that maintains credit information, Social Security numbers, and other personal information on virtually all U.S. Citizens
 - Revealed that they had divulged information on at least 150,000 people to scam artists who had posed as legitimate businessmen
- According to Gartner (www.gartner.com), phishing attacks cost U.S. banks and credit card companies about \$1.2 billion in 2003
- Estimate that between May 2004 and May 2005, approximately 1.2 million computer users in the United States suffered losses caused by phishing

Users and Passwords

- Fortunately or unfortunately ...
- Users must be entrusted with security of their own systems
 - Passwords still used extensively as way to authenticate people
 - Why are they still used?
 - Easy to use, know how to use them, people are familiar with them, cheap!!
 - Can be used both locally and remotely
 - On your home PC and over the Internet

Passwords

- While we may find them annoying, and even take them for granted,
- Important to remember why passwords are important
 - Passwords are often the first and possibly only defense against intrusion

Passwords

Passwords are a classic way to authenticate

- Advantages of passwords:
 - Seemingly they work everywhere
 - Easy to remember and use
 - Everyone knows how to use them

Password Weaknesses

- If password is sent in the clear, can be intercepted
- If password is encrypted, requires establishment of encryption key
- People choose bad passwords
- Passwords are easily observed
- Passwords can be sniffed by spyware

Disadvantages of Passwords



Note: Passwords are generally pretty weak

- University of Michigan: 5% of passwords were gobblue
- Passwords used in more than one place

Not just because bad ones selected: If you can remember it, then a computer can guess it

- Computers can often guess very quickly
- Easy to mount off-line attacks
- Easy countermeasures for on-line attacks

Disadvantages of Passwords

Attacker can access the hashed password

- Can guess and test passwords offline

Called “password cracking”

Lots of help

- John the Ripper
- Cain
- THC Hydra

How to Break Passwords

- Three main ways programs “crack” passwords
 1. **Dictionary attack** - tries thousands of words from dictionary files as possible passwords
 - Every word from dictionary is tested in a variety of modifications, cat – tac, cat1, cated
 - Encrypt words from list of English words, compare each encryption against stored encrypted version of users' passwords

How to Break Passwords

2. Brute Force Attack

- Finds passwords by checking all possible combinations of characters from the Symbol Set
 - You can make a big Brute-Force-Dictionary to implement Brute-Force attack

How to Break Passwords

3. Guessing Attack – Guess based on something “known”

- blank (none)
- words "password", "passcode", "admin" and their derivatives
- a row of letters from the qwerty keyboard -- qwerty itself, asdf, or qwertyuiop)
- user's name or login name
- name of their significant other, a friend, relative or pet
- birthplace or date of birth, or a friend's, or a relative's
- automobile license plate number, or a friend's, or a relative's
- office number, residence number or most commonly, their mobile number

Effectiveness of Password Guessing

Can AsK: How well do these work?

Guessing ... you decide

- September 2008, Yahoo e-mail account of Governor of Alaska and Vice President of the United States nominee Sarah Palin
- Accessed without authorization by someone who researched answers to two of her security questions
 - Zip code and date of birth and was able to guess the third, where she met her husband!

Effectiveness of Password Guessing

- Another example:
 - Gary McKinnon, accused of perpetrating "biggest military computer hack of all time",
 - Claimed that he was able to get into military's networks simply by using a Perl script that searched for blank passwords
 - His report suggests that there were computers on these networks with no passwords at all!

Effectiveness of Password Cracking

- From a course taught at Penn state in the CS Engineering Department
- Ran John the Ripper on CSE authentications
 - 3500 in all
- In first hour, 25% were recovered
 - About half of these due to dictionary attacks
 - But, half using other heuristics and brute force
- Over 5 days, 35% were recovered
 - Steady state recovery due to brute force

Top Password cracking software listed here

<http://sectools.org/crackers.html>

Password Cracking

Password Length	Numeric only	Alphabetic upper-case only	Alpha-numeric Upper and lower	Alpha-numeric + Special Chars.
4	.001 sec	.046 sec	1.48 sec	8.49 sec
6	.1 sec	30.9 sec	94.7 min	21.7 hrs
7	1 sec	13.4 min	4.08 days	87 days
8	10 sec	5.80 hrs	253 days	22.9 yrs
10	16.7 min	163 days	26.6 centuries	2110 centuries
12	27.8 hrs	303 yrs	102k centuries	19.5m centuries

Note: Table based on being able to generate 10 million cracks per second

Common Password Advice

Should be at least 8 characters

Use characters from each of the following four classes:

- English upper case letters
- English lower case letters
- Westernized Arabic numerals (0,1,2,...)
- Non-alphanumeric (special) characters such as punctuation symbols

Don't use a proper name or any word in the dictionary without misspelling it in some way

Don't reuse a password you have used before

Don't use the same password for different types of systems

How Passwords are Used

- Passwords are not stored, or should not be, on systems their hashed representations are.
- Windows Files: On Windows systems password hashes are stored in the SAM (Security Accounts Manager) database.
- Unix/Linux Files: On Unix/Linux systems the password hashes are stored in the /etc/shadow file
- **Authentication Process**
 - User enters password, Example: catdog
 - Hash is computed, $\text{Hash}(\text{catdog}) =$
sMxYb7\$og4uxH4oHXAVwf
 - The computed hash is compared to stored hash
 - Access granted or denied

Summary

- Vulnerabilities are in all current popular OS's
- Hard to go beyond the “hype” to understand how vulnerable you are given a certain OS
- Try to discover for yourself how secure the OS is that you are using
- Read bulletins, seek opinions of people you trust and try to protect yourself
 - Buy the add-on security products, disable OS features, run with reduced privilege

The End

©2001 Hailmark Licensing, Inc./Dist. by Universal Press Syndicate

*I forgot the password for
the file where I keep all my
passwords.*

