CSE331: Introduction to Networks and Security

Lecture 33
Fall 2002
Announcements

• Final Exam Location
  – Moore 212
  – Tues. 17 Dec.
  – 8:30 – 10:30 AM
Recap

• Malicious Programs
  – Computer Virus Defenses

• Today:
  – Computer Worms
  – Other malicious code
Internet Worms

• November 2, 1988
• Robert T. Morris Jr. unleashed Internet worm  
  – Graduate student at Cornell University  
  – Convicted in 1990 of violating Computer Fraud and Abuse Act  
  – $10,000 fine, 3 yr. Suspended jail sentence, 400 hours of community service  

  – Son of the chief scientist at the National Computer Security Center -- part of the National Security Agency  
  – Today he’s a professor at MIT
Morris Worm Transmission

• Find user accounts on the target machine
  – Dictionary attack on /etc/passwd
  – If it found a match, it would log in and try the same username/password on other local machines

• Exploit bug in fingerd
  – Classic buffer overflow attack

• Exploit trapdoor in sendmail
  – Programmer left DEBUG mode in sendmail, which allowed sendmail to execute an arbitrary shell command string.
Morris Worm Infection

- Sent a small loader to target machine
  - 99 lines of C code
  - It was compiled on the remote platform (cross platform compatibility)
  - The loader program transferred the rest of the worm from the infected host to the new target.
  - Used authentication! To prevent sys admins from tampering with loaded code.
  - If there was a transmission error, the loader would erase its tracks and exit.
Morris Worm Stealth

• When loader obtained full code
  – It put into main memory and encrypted
  – Original copies were deleted from disk
  – (Even memory dump wouldn’t expose worm)

• Worm periodically changed its name and process ID
Effects

• Resource exhaustion
  – Denial of service
  – There was a bug in the loader program that caused many copies of the worm to be spawned per host

• System administrators cut their network connections
  – Couldn’t use internet to exchange fixes!

• 6,000 networks were shut down or disconnected
  – Down for several days
  – Damage estimates: $100,000 — $97 Million
Worms (In General)

• Self-contained running programs
  – Unlike viruses

• Infection strategy more active
  – Exploit buffer overflows
  – Exploit bad password choice

• Defenses
  – Monitor system resources
  – Proper access control
Trapdoors

• A *trapdoor* is a secret entry point into a module
  – Affects a particular system

• Inserted during code development
  – Accidentally (forget to remove debugging code)
  – Intentionally (maintenance)
  – Maliciously (an insider creates a hole)
Trojan Horse

• A program that pretends to be do one thing when it does another
  – Or does more than advertised

• Login Prompts
  – Trusted path

• Accounting software

• Example:
  – Game that doubles as a sshd process.
Salami Attacks

• Programs that compute money are targets of this attack
  – Use subtle rounding errors
  – Often distribute errors among many transactions
  – Accumulate small amounts of money

• Example:
  – 6.5% (annual) interest on $102.87 for 31 days is $0.5495726
  – Instead of rounding to $0.55, round to $0.54
  – Funnel the extra penny somewhere else
Covert Channels

• Program that leaks confidential information intentionally via secret channels.
• Not that hard to leak a small amount of data
  – A 64 bit shared key is quite small!

• Example channels
  – Adjust the formatting of output: use the “\t” character for “1” and 8 spaces for “0”
  – Vary timing behavior based on key
Differential Power Analysis

- Read the value of a DES password off of a smartcard by watching power consumption!

- This figure shows simple power analysis of DES encryption. The 16 rounds are clearly visible.
Defenses for Covert Channels

• Code review
  – This is an audit!

• Automated program analysis
  – Type systems that let programmers specify confidentiality labels

• But, not much you can do against a determined insider
Multilevel Security

• Department of Defense
• Trusted Computer System Evaluation Criteria (TCSEC)
  – Known as the *Orange Book*
  – Circa 1985

• Multiple levels of security
  – Public < Classified < Secret < Top Secret
TCSEC Ratings

• **Division (D): Minimal Protection**
  – *This division contains only one class. It is reserved for those systems that have been evaluated but that fail to meet the requirements for a higher evaluation class.*

• **Division (C): Discretionary Protection**
  – *Classes in this division provide for discretionary (need-to-know) protection and, through the inclusion of audit capabilities, for accountability of subjects and the actions they initiate.*
TCSEC Ratings

• **Division (B): Mandatory Protection**
  – The notion of a TCB that preserves the integrity of sensitivity labels and uses them to enforce a set of mandatory access control rules is a major requirement in this division. Systems in this division must carry the sensitivity labels with major data structures in the system. The system developer also provides the security policy model on which the TCB is based and furnishes a specification of the TCB. Evidence must be provided to demonstrate that the reference monitor concept has been implemented.
TCSEC Ratings

• **Division (A): Verified Protection**
  
  – *This division is characterized by the use of formal security verification methods to assure that the mandatory and discretionary security controls employed in the system can effectively protect classified or other sensitive information stored or processed by the system. Extensive documentation is required to demonstrate that the TCB meets the security requirements in all aspects of design, development and implementation.*