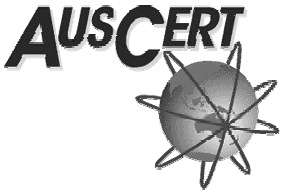


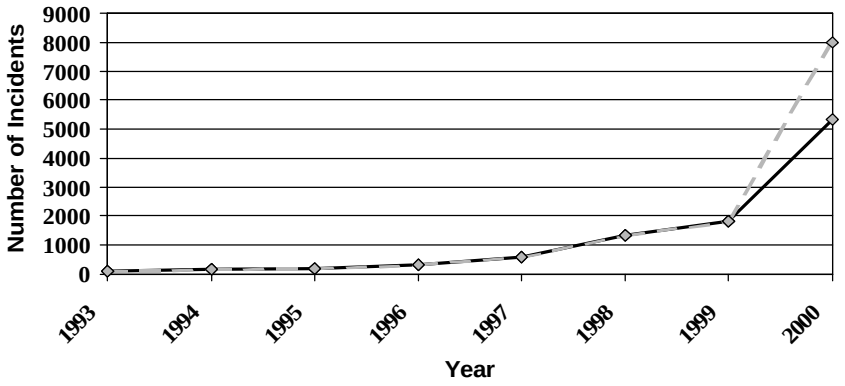
Know Thy Attacker

Australian Computer Emergency Response Team
The University of Queensland
Brisbane, Queensland 4072
AUSTRALIA

Overview

- **Statistics**
- **What the Attackers Do**
- **Root Causes**
- **Industry Developments**
- **Case Study**
- **AusCERT**

Statistics
Recorded Incidents to August 2000



Full line represents actual data.
Dotted line represents projected total.

Overview

- Statistics
- What the Attackers Do
- Root Causes
- Industry Developments
- Case Study
- AusCERT

Evolution (1)

- **Circa 1988:**
  - » **Password attacks**
  - » **Trivial exploits (eg *tftp*)**
- **Circa 1992:**
  - » **Source code examination**
  - » **Exploitation of protocol flaws (eg *NIS*, *NFS*)**
- **Circa 1993:**
  - » **Traffic monitoring (ie packet sniffing)**
  - » **Packages, components and Trojan Horses (eg *rootkit*)**
- **Circa 1995:**
  - » **Multi-phase attacks (eg *IP Spoofing*)**

Evolution (2)

- **Circa 1996:**
  - » **Examination of library calls in real time**
- **Circa 1997: (The Year From Hell)**
  - » **Exploitation of CGI defaults (hardly an advance!)**
  - » **Use of search engines to find victims (eg *phf*)**
  - » **Traffic interception (eg *netcat*)**
  - » **Attacks against crypto systems (eg *NT passwords*)**
  - » **Denial of Service attacks (eg *land*, *teardrop*)**
- **Circa 1998:**
  - » **Advanced Trojan Horses (eg *Back Orifice*)**
  - » **Advanced network mapping (eg *nmap*)**

Evolution (3)

- Circa 1999:
  - » Rapid propagation virii (eg *Melissa*)
  - » Multi-phase DDOS attacks (eg *Trin00* etc)
- Circa 2000: Same ideas, new variations
- Evolutionary invariants:
  - Mechanical implementation errors
  - Exploitation of default configuration

Threat (1): Attacker Profile

- Think *Motive - Opportunity - Means*
- Discriminators:
  - Untargeted (ie *Indiscriminate victim*)
    - » Function of *Opportunity*
  - Specifically targeted:
    - » Function of any of these
    - » Only *Opportunity* can be controlled by the victim
    - » The attacker controls the *Motive* and the *Means*
- Assume motivated attacker
  - » Do NOT assume technical motives only

Threat (2): Motives

- Indiscriminate attack
- Curiosity, vandalism or peer kudos
- Fear, greed or malice
- Hacktivism
- Industrial information operations
- Electronic warfare

Threat (3): Technical Goals

- Intelligence gathering
- Denial of Service
- Read protected information
- Modify information
- Execute arbitrary commands
  - Bonus points for:
    - » Privileges
    - » Expand and ensure further access

Threat (4): Outcomes

- Financial gain
- Media exposure
- Political agendas
- Any other outcome to service *fear* or *greed*

Overview

- Statistics
- What the Attackers Do
- Root Causes
- Industry Developments
- Case Study
- AusCERT

Root Causes: Framework

Software *systems*
implement *properties*
which may be subverted by *attack tools*
that exploit *vulnerability classes*.

Different attack tools may produce similar results
with a totally different set of characteristics.

Root Causes: Example Systems (1)

- **Application layer:**
  - **User applications**
    - » **Electronic mail**
    - » **Web browser**
    - » **Office applications**
  - **Payment systems**
  - **Transaction processors**

Root Causes: Example Systems (2)

- **Common:**
  - **Authentication systems**
  - **Access control systems**
  - **Cryptographic systems**
  - **Auditing systems**
- **Support:**
  - **Domain Name Service (DNS)**
  - **Basic operating system**
  - **Low-level IP stack**

Root Causes: Properties

Core: **(the properties the attacker seeks to modify)**

- **Confidentiality**
- **Integrity**
- **Availability**

Associated issues: **(typical indirect levers)**

- **Accountability**
- **Non-repudiation**
- **Trust**

Root Causes: Attack Tools

Following is a non-exhaustive example list only.

- Network maps
- Packet sniffers
- Password crackers
- Browser based
- Scripting based
- Trojan Horses
- Viruses
- Worms
- Denial of Service (DOS)
- Distributed DOS

Root Causes : Vulnerability Classes

- **People** (eg social engineering)
- **Practices** (eg default configuration)
- **Implementation** (eg buffer overflows)
- **Protocols** (eg IP Spoofing)

The remedy (if any) depends on the class (type) of vulnerability.

Implementation Vulnerabilities: Reasons (1)

Following is a non-exhaustive example list only.

- **Insufficient testing**
  - » **Test all exceptions as well as core functionality**
  - » **Crash testing: Core dump content? Cached files?**
  - » **Buffer overflows**
- **Dependencies on flawed resources**
  - » **Race conditions in kernels**
  - » **Buggy shared libraries**

Implementation Vulnerabilities: Reasons (2)

Following is a non-exhaustive example list only.

- **Trusting untrustworthy data**
  - » **Forged packets**
  - » **Tainted data entry**
- **Inappropriate use of resources**
  - » **Using *emacs* or *vi* in a captive account**
- **Incompatible or incomplete design specs**
  - » **Differences in use of saved setuids by different kernels**
  - » **Incompatible library interfaces**

Example 1: Virus - *Love Bug*

- Summary: A script-based email-borne virus
- Location (system):
  - » User application
- Property affected:
  - » Confidentiality (second wave)
  - » Availability (effect of propagation technique)
- Vulnerability class:
  - » People (social engineering aspect)
    - Remedy: Educate users
  - » Practices (default scripting configurations)
    - Remedy: Adjust scripting configs or filter content
    - Remedy: Adjust mail filtering rules

Example 2: DoS - *teardrop*

- Summary: A DoS attack exploiting coding error
  - » Specifically, the reconstruction of packet fragments
- Location (system):
  - » Low-level IP stack
- Property affected:
  - » Availability (Blue Screen of Death)
- Vulnerability class:
  - » Implementation
    - Remedy: Vendor patch to ensure precise fragment size

Example 3: DoS - *smurf*

- Summary: A DoS attack exploiting configuration
  - » Specifically, replies to forged broadcast requests
- Location (system):
  - » Low-level IP stack
- Property affected:
  - » Availability
- Vulnerability class:
  - » Practices
    - Remedy: Intermediary to adjust router and kernel configs
    - Remedy: *Victim has limited defence*...

Addressing the Issue

Security is a business problem, not a technical problem

- Physical security
- Capacity planning
- Change control
- Procedures
  - Continuity of service
  - Recovery of service
- Purchasing criteria
- Service Level Agreements
- Administrative controls
- Technical defences

Overview

- Statistics
- What the Attackers Do
- Root Causes
- Industry Developments
- Case Study
- AusCERT

Industry Developments

- Standardisation
  - For example: AS4444, CMM, Common Criteria etc
- Penetration testing
- Legal environment
- Insurance
- Open source software
  - Nice counterview at SecurityFocus
- Critical Infrastructure Protection

Overview

- Statistics
- What the Attackers Do
- Root Causes
- Industry Developments
- Case Study
- AusCERT

Background

- Notable media events during 1999 - 2000:
  - » Apr 1999 - *Melissa*
  - » Jan 2000 - Y2K
  - » Feb 2000 - Distributed Denial of Service (DDOS) attacks
  - » May 2000 - The *Love Bug* (ie *Herbie*)
- Wild speculation raises important questions:
  - » How many sites were really affected?
  - » To what extent?
  - » Does media coverage accurately reflect this?
  - » What is the true cost of damage?

Survey Results

- Infection rate 23% (*Melissa*), 50% (*Herbie*)
  - However, generally low impact
- Estimated losses not huge
- Mixed view of media exposure
  - Some value in raising awareness
  - Hype didn't help
- User awareness increasing

Major Conclusions

- Sophisticated users buy time
- The market gets the security it asks for
  - Feature-rich and convenience are great, but...
  - Security and vendor responsiveness are critical
- Forearm with:
  - Prepared response plans
  - Technical diversity
- A cool head is invaluable
- Rapid notice and intelligence is valuable

Overview

- Statistics
- What the Attackers Do
- Root Causes
- Industry Developments
- Case Study
- AusCERT

Core Roles

- Coordination centre
- Knowledgeable point of contact
  - Statistical reference
  - Trusted reference *("the voice of reason")*
- Training and Education
- Research and Publishing
- Advocacy

References

- **Lessons Learned from Loving Melissa**
  - **http://www.auscert.org.au/Information/Auscert_info/Papers/loving-melissa.html**
- **A Lab Engineer's Checklist for Writing Secure Unix Code**
  - **ftp://ftp.auscert.org.au/pub/auscert/papers/secure_programming_checklist**
- **Wide Open Source**
  - **http://www.securityfocus.com/templates/article.html?id=19**

AusCERT Contact Information

24 Hour Hotline: (07) 3365 4417 (After Hours for Emergencies)
International: +61 7 3365 4417 (GMT+1000)

Facsimile: (07) 3365 7031
International: +61 7 3365 7031

Electronic Mail: auscert@auscert.org.au
Anonymous ftp: ftp://ftp.auscert.org.au/pub/
World Wide Web: http://www.auscert.org.au/

Postal: AusCERT
The University of Queensland
Brisbane Qld. 4072
Australia