SEZG566/SSZG566

Secure Software Engineering

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

T V Rao

- *The slides presented here are obtained from the authors of the books, product documentations, and from various other contributors. I hereby acknowledge all the contributors for their material and inputs.*
- *I have added and modified slides to suit the requirements of the course.*

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

Definitions and concepts of security

Security Problem

Organizations store, process, transmit their most sensitive information using software-intensive systems.

Private citizens depend on software to shop, bank, invest, and carry out most personal and social activities

Global connectivity makes the sensitive information and software systems vulnerable to unintentional and unauthorized use.

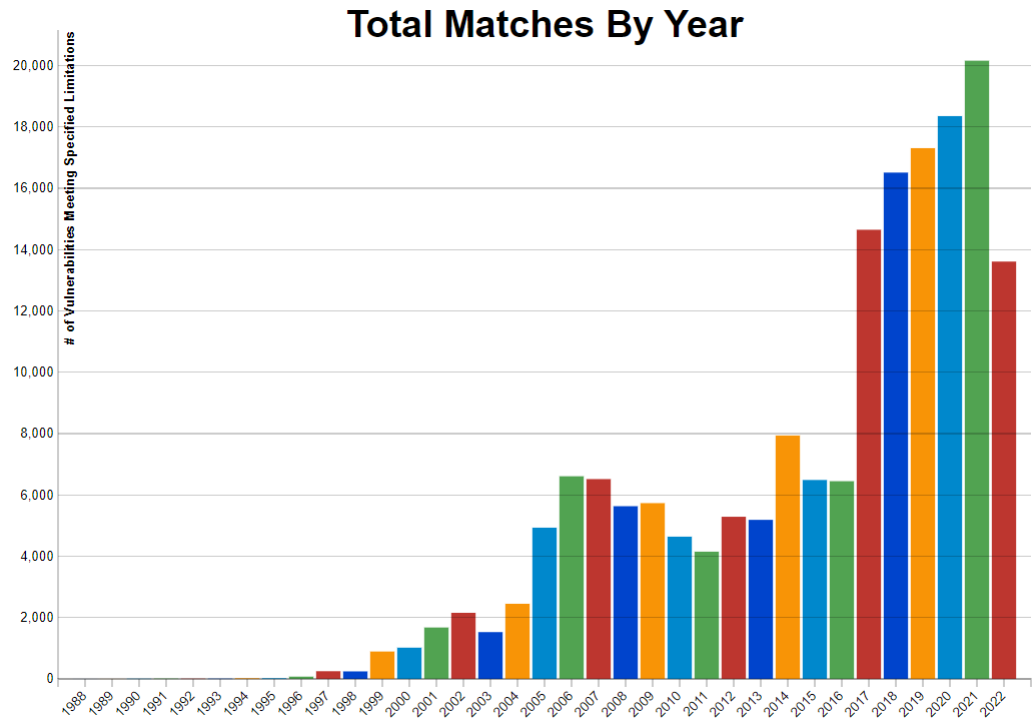
As per some experts, we are in era of

- Information warfare
- Cyber terrorism
- Computer crime

Terrorists, Organized criminals, other criminals are targeting software-intensive systems and are able to gain entry.

- There are many systems which can not resist attacks

Trends of reported vulnerabilities



NVD is the U.S. government repository of standards based vulnerability management data. This data enables automation of vulnerability management, security measurement, and compliance

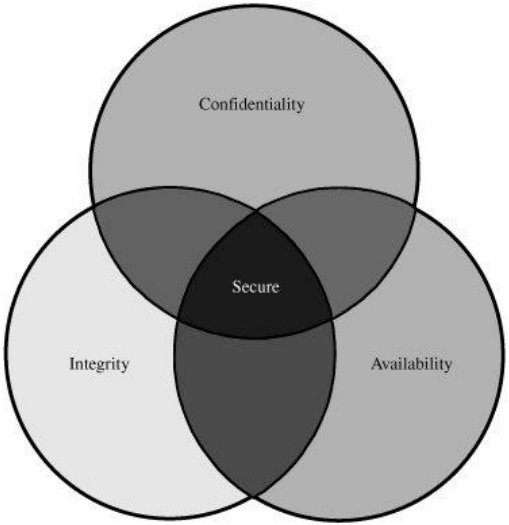
Source : National Vulnerability Database of NIST (National Institute of Standards and Technology)

Security Principles

Saltzer and Schroeder defined security as “techniques that control who may use or modify the computer or the information contained in it”

Described the three main categories of concern:

- Confidentiality
- Integrity
- Availability



Security implies Confidentiality, Integrity, and Availability

Saltzer and Schroeder, “The Protection of Information in Computer Systems.” *Communications of the ACM*, 1974

Key Security Concepts

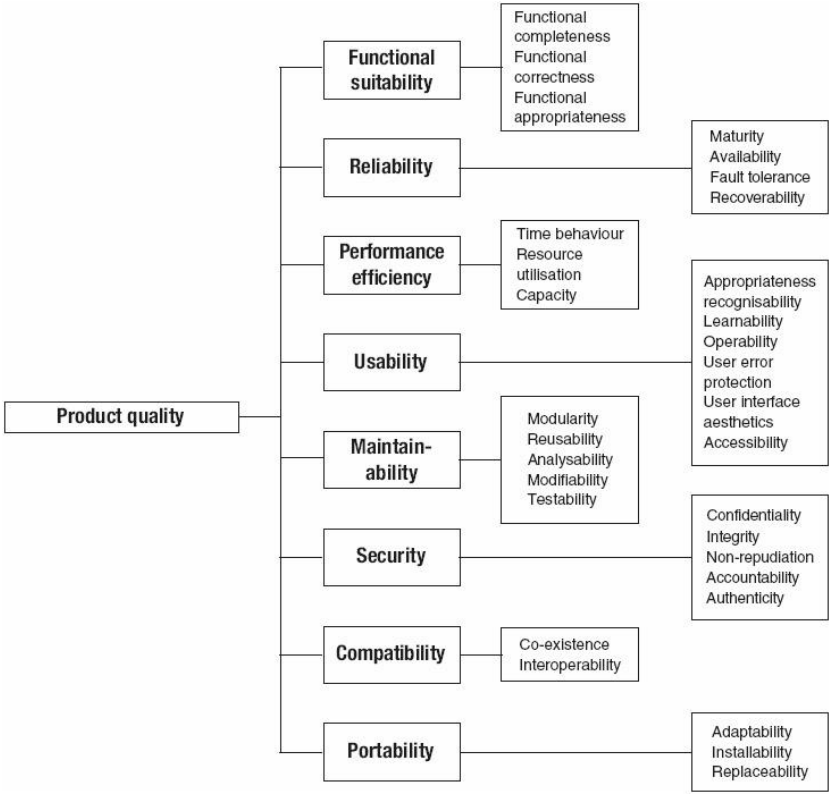
- Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information
- Guarding against improper information modification or destruction, including ensuring information nonrepudiation and authenticity
- Ensuring timely and reliable access to and use of information

Two additional properties

Two additional properties commonly associated with human users are required in software entities that act as users, e.g. proxy agents, web services etc.

- Accountability : All security-relevant actions of the software-as-user must be recorded and tracked with attribution, both while and after the recorded action occurs
- Non-repudiation : Ability to prevent the software-as-user from disproving or denying responsibilities for actions it has performed

Product quality model of ISO/IEC 25010



Software Assurance

The Department of Defense (DoD) defines software assurance as follows:

– *System assurance (SA) is the justified confidence that the system functions as intended and is free of exploitable vulnerabilities, either intentionally or unintentionally designed or inserted as part of the system at any time during the life cycle. This ideal of no exploitable vulnerabilities is usually unachievable in practice, so programs must perform risk management to reduce the probability and impact of vulnerabilities to acceptable levels*

A more practical definition (given by SEI CMM) emphasizes risk management by balancing cost and potential loss

– *the level of confidence we have that a system behaves as expected and the security risks associated with the business use of the software are acceptable*

Software assurance includes software reliability, software safety, and software security

– Software assurance becomes important since critical infrastructure (viz. power, communication etc.) depend on software-intensive systems

Processes for Secure Software

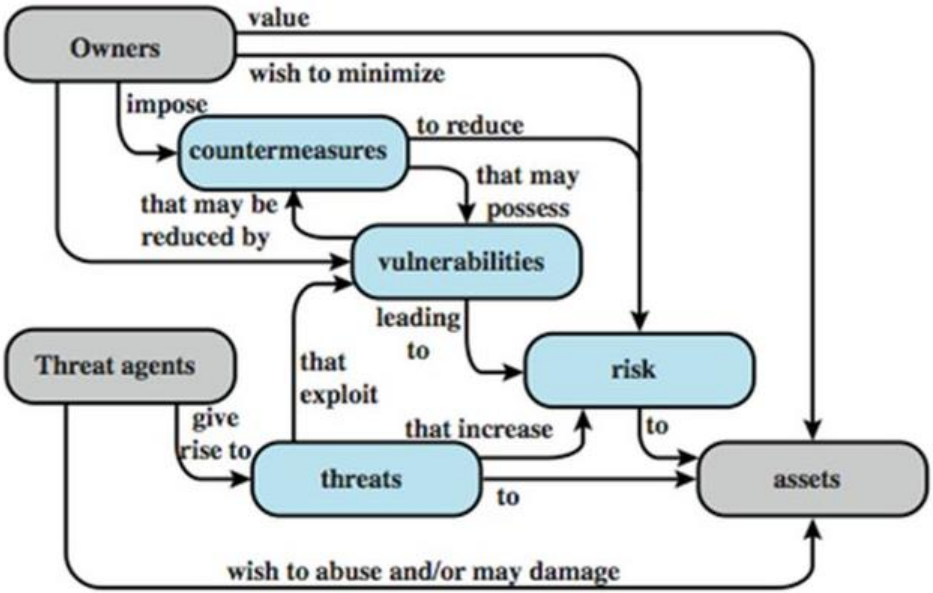
The most critical difference between secure and insecure software lies in the nature of the processes and practices used to specify, design, and develop the software

- Gortzel[2006]

Software vulnerabilities can originate from

- Decisions made by software engineers
- Flaws introduced in specification & design
- Faults from developed code
- Choice of programming language, development tools, operational environment etc.

Security Concepts and Relationships



Security Concepts and Relationships

Threat

- Any circumstance or event with the potential to adversely impact organizational operations (including mission, functions, image, or reputation), organizational assets, individuals, other organizations, or the Nation through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service. [CNSS 2010] 2. Any event that will cause an undesirable impact or loss to an organization if it occurs.

Vulnerability

- Weakness in an information system, system security procedures, internal controls, or implementation that could be exploited by a threat source. [CNSS 2010] 2. The absence or weakness of a safeguard. It can also be described as a weakness in an asset or the methods of ensuring that the asset is survivable.

Risk

- A measure of the extent to which an entity is threatened by a potential circumstance or event, and typically a function of 1) the adverse impacts that would arise if the circumstance or event occurs; and 2) the likelihood of occurrence.

Countermeasure

- Actions, devices, procedures, or techniques that meet or oppose(i.e., counters) a threat, a vulnerability, or an attack by eliminating or preventing it, by minimizing the harm it can cause, or by discovering and reporting it so that corrective action can be taken. NIST SP 800-53: Actions, devices, procedures, techniques, or other measures that reduce the vulnerability of an information system. Synonymous with security controls and safeguards. [CNSS 2010]

Security Concepts and Relationships

The principle of ***defense-in-depth*** is that layered security mechanisms increase security of the system as a whole. If an attack causes one security mechanism to fail, other mechanisms may still provide the necessary security to protect the system.

Social engineering attack is based on deceiving end users or administrators at a target site. Such attacks are typically carried out by email or by contacting users by phone and impersonating an authorized user, in an attempt to gain unauthorized access to a system or application

Security Concepts and Relationships

In computer security, a ***sandbox*** is a security mechanism for separating running programs.

- It is often used to execute untested code, or untrusted programs from unverified third parties, suppliers, untrusted users and untrusted websites.
- A ***sandbox*** typically provides a tightly controlled set of resources for guest programs to run in, such as disk and memory, network access, the ability to inspect the host system or read from input devices (disallowed or heavily restricted).

Sandboxes may be seen as a specific example of virtualization. ***Sandboxing*** is frequently used to test unverified programs that may contain a virus or other malicious code, without allowing the software to harm the host device

Privacy – GDPR

What does GDPR Compliance Look Like for Companies?

- "Data protection will be as significant as antitrust or anti-corruption in terms of compliance risk." − *Hunton & Williams*
- Many small companies including news sites blocking EU users
- Product team compliance includes
  - Changes, often major, to back-end data logging
  - User interface changes
  - New tools for access, correction, portability, etc.
- Legal team compliance includes
  - Data Impact Assessments
  - Internal record-keeping
  - Renegotiating commercial contracts
  - Changing user Terms of Service
  - In some cases appointing Data Protection Officer resident in EU

Cyberlaw.Stanford.edu

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

Threats to Software/Assets

The Asymmetric Problem of Security

Basics of Secure Design Development Test

– www.microsoft.com

Hacking a Politician email

Hacker chooses to hack politician's mail account and possibly impact US elections(2008)

The approach was

- Find the mail id
- Use Forgot Password feature
- The mail provider asks standard personal questions
- The politician biography is well known

Security on Cloud (Example)

Code Spaces kept up their security measures, ensured that their server security was tight, and relied on Amazon for the bulk of their infrastructure -- like thousands of other companies.

The attack that brought Code Spaces under was as simple as gaining access to its AWS control panel.

Code Spaces was built mostly on AWS, using storage and server instances to provide its services. Those server instances weren't hacked, nor was Code Spaces' database compromised or stolen.

Attacker gained access to the company's AWS control panel & deleted resources on the cloud.

The demise of Code Spaces at the hands of an attacker shows that, in the cloud, off-site backups and separation of services could be key to survival

http://www.infoworld.com/article/2608076/data-center/murder-in-the-amazon-cloud.html 23 June 2014

In the space of one hour, my entire digital life was destroyed. First my Google account was taken over, then deleted. Next my Twitter account was compromised, and used as a platform to broadcast racist and homophobic messages. And worst of all, my AppleID account was broken into, and my hackers used it to remotely erase all of the data on my iPhone, iPad, and MacBook.

My accounts were daisy-chained together. Getting into Amazon let my hackers get into my Apple ID account, which helped them get into Gmail, which gave them access to Twitter.

– Mat Honan, Sr. Staff Writer, wired.com

http://www.wired.com/2012/08/apple-amazon-mat-honan-hacking/all/

How Apple and Amazon Security Flaws Led to My Epic Hacking _ WIRED

If it wasn't clear before, it certainly is now: Your username and password are almost impossible to keep safe.

Nearly 443,000 e-mail addresses and passwords for a Yahoo site were exposed late Wednesday. The impact stretched beyond Yahoo because the site allowed users to log in with credentials from other sites -- which meant that user names and passwords for Yahoo (YHOO, Fortune 500), Google's (GOOG, Fortune 500) Gmail, Microsoft's (MSFT, Fortune 500) Hotmail, AOL (AOL) and many other e-mail hosts were among those posted publicly on a hacker forum.

What's shocking about the development isn't that usernames and passwords were stolen -- that happens virtually every day. The surprise is how easily outsiders cracked a service run by one of the biggest Web companies in the world.

The group of seven hackers, who belong to a hacker collective called D33Ds Company, got into Yahoo's Contributor Network database by using a rudimentary attack called a SQL injection.

http://money.cnn.com/2012/07/12/technology/yahoo-hack/

Attack Surfaces

Consist of the reachable and exploitable vulnerabilities in a system

Examples:

- Open ports on outward facing Web and other servers, and code listening on those ports
- Services available on the inside of a firewall
- Code that processes incoming data, email, XML, office documents, and industry-specific custom data exchange formats
- Interfaces, SQL, and Web forms
- An employee with access to sensitive information vulnerable to a social engineering attack

Attack surfaces

Attack surface: the reachable and exploitable vulnerabilities in a system

- Open ports
- Services outside a firewall
- An employee with access to sensitive info
- …

Three categories

- **Network attack surface** (i.e., network vulnerability)
- **Software attack surface** (i.e., software vulnerabilities)
- **Human attack surface** (e.g., social engineering)

Attack analysis: assessing the scale and severity of threats

Security Concepts and Relationships

The ***attack surface*** of a software environment is the sum of the different points (the "attack vectors") where an unauthorized user (the "attacker") can try to enter data to or extract data from an environment.

An ***attack vector*** is a path or means by which a hacker (or cracker) can gain access to a computer or network server in order to deliver a payload or malicious outcome.

If an attack vector is thought of as a guided missile (e.g. email), its payload can be compared to the warhead (e.g. malicious attachment) in the tip of the missile.

Automotive Attack Surface

Modern cars are controlled by complex distributed computer systems comprising millions of lines of code executing on tens of heterogeneous processors with rich connectivity provided by internal networks (e.g., Controller Area Network CAN).

This structure has offers significant benefits to efficiency, safety and cost, but also creates the opportunity for new attacks.

An attacker connected to a car's *internal network* can circumvent *all* computer control systems, including safety critical elements such as the brakes and engine.

The long-range wireless attack surface is that exposed by the remote telematics systems (e.g., Ford's Sync, GM's OnStar, Toyota's SafetyConnect, Lexus' Enform, BMW's BMW Assist, and Mercedes-Benz' mbrace) that provide continuous connectivity via cellular voice and data networks for supporting safety (crash reporting), diagnostics (early alert of mechanical issues), anti-theft (remote track and disable), and convenience (hands-free data access such as driving directions or weather).

Comprehensive Experimental Analyses of Automotive Attack Surfaces Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, and Stefan Savage University of California, San Diego
Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno University of Washington USENIX Security, August 10–12, 2011

Examples of threats

	Availability	Confidentiality	Integrity
Hardware	Equipment is stolen or disabled, thus denying service.	An unencrypted CD-ROM or DVD is stolen.	
Software	Programs are deleted, denying access to users.	An unauthorized copy of software is made.	A working program is modified, either to cause it to fail during execution or to cause it to do some unintended task.
Data	Files are deleted, denying access to users.	An unauthorized read of data is performed. An analysis of statistical data reveals underlying data.	Existing files are modified or new files are fabricated.
Communication Lines and Networks	Messages are destroyed or deleted. Communication lines or networks are rendered unavailable.	Messages are read. The traffic pattern of messages is observed.	Messages are modified, delayed, reordered, or duplicated. False messages are fabricated.

Data in Transit

Attacks on data in networks can be

Passive attacks : eavesdropping on, or monitoring of transmissions

- Release of message contents
- Traffic analysis : Encrypted message can not be read. Location, identity of host, frequency, and length of messages can help opponents make guess

Active attacks

- Replay : passive capture of data & subsequent retransmission to produce an unauthorized effect
- Masquerade : one entity pretends to be another entity.
- Modification of messages : some portion of a legitimate message is altered.
- Denial of service : inhibit normal use of facilities

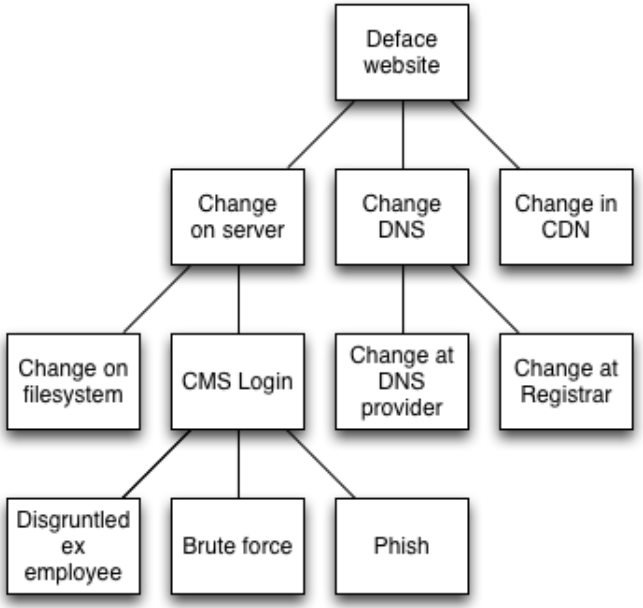
Attack trees

A branching, hierarchical data structure that represents a set of potential vulnerabilities

Objective: to effectively exploit the info available on attack patterns

- published on CERT or similar forums
- Security analysts can use the tree to guide design and strengthen countermeasures

An attack tree (to deface a web site)



http://ertw.com/blog/2015/01/06/thinking-about-cyber-security/

Content delivery network (CDN) : System of distributed servers (network) that deliver webpages and other Web content to user based on the geographic locations

Domain Name System (DNS) : Hierarchical decentralized naming system for computers, services, or any resource connected to the Internet or a private network

CMS : Content Management System

BITS Pilani

Pilani | Dubai | Goa | Hyderabad

Malware Nomenclature

Malware

"A program that is inserted into a system, usually covertly, with the intent of compromising the confidentiality, integrity, or availability of the victim's data, applications, or operating system or otherwise annoying or disrupting the victim."

Malicious software

Programs exploiting system vulnerabilities

Known as malicious software or malware

- program fragments that need a host program
  - e.g. viruses, logic bombs, and backdoors
- independent self-contained programs
  - e.g. worms, bots
- replicating or not

Sophisticated threat to computer systems

Malware Terminology

Virus: *attaches itself to a program*

Worm: *propagates copies of itself to other computers*

Logic bomb: *"explodes" when a condition occurs*

Trojan horse: *fakes/contains additional functionality*

Backdoor (trapdoor): *allows unauthorized access to functionality*

Mobile code: *moves unchanged to heterogeneous platforms*

Auto-rooter Kit (virus generator): *malicious code (virus) generators*

Spammer and flooder programs: *large volume of unwanted "pkts"*

Keyloggers: *capture keystrokes*

Rootkit: *sophisticated hacker tools to gain root-level access*

Zombie: *software on infected computers that launch attack on others (aka bot)*

More terms

Payload: actions of the malware

Crimeware: kits for building malware; include propagation and payload mechanisms

- Zeus, Sakura, Blackhole, Phoenix

APT (advanced persistent threats)

- Advanced: sophisticated
- Persistent: attack over an extended period of time
- Threat: selected targets (capable, well-funded attackers)

https://www.theguardian.com/world/2022/jan/14/ukraine-massive-cyber-attack-government-websites-suspected-russian-hackers

Viruses

Piece of software that infects programs

- modifying them to include a copy of the virus
- so it executes secretly when host program is run

Specific to operating system and hardware

- taking advantage of their details and weaknesses

A typical virus goes through phases of:

- dormant: *idle*
- propagation: *copies itself to other program*
- triggering: *activated to perform functions*
- execution: *the function is performed*

Biological Virus : Tiny scraps of genetic code – DNA or RNA – that can take over a living cell and trick it into making replicas of the original virus

Virus structure

Components:

- infection mechanism: enables replication
- trigger: event that makes payload activate
- payload: what it does, malicious or benign

Prepended/postpended/embedded

- When infected program invoked, executes virus code then original program code

Can block initial infection (difficult) or propagation (with access controls)

Virus structure (abstract)

```
    program V :=

{goto main;
    1234567;

    subroutine infect-executable :=
        {loop:
        file := get-random-executable-file;
        if (first-line-of-file = 1234567)
          then goto loop
          else prepend V to file; }

    subroutine do-damage :=
        {whatever damage is to be done}

    subroutine trigger-pulled :=
        {return true if some condition holds}

main:   main-program :=
        {infect-executable;
        if trigger-pulled then do-damage;
        goto next;}

next:

}
```

Virus classification

By target

- boot sector: *infect a master boot record*
- file infector: *infects executable OS files*
- macro virus: *infects files to be used by an app*
- multipartite: infects multiple ways

By concealment

- encrypted virus: *encrypted; key stored in virus*
- stealth virus: *hides itself (e.g., compression)*
- polymorphic virus: *recreates with diff "signature"*
- metamorphic virus: *recreates with diff signature and behavior*

Virus Variants

Macro and scripting viruses

- Became very common in mid-1990s since
  - platform independent
  - infect documents
  - easily spread
- Exploit macro capability of Office apps
  - executable program embedded in office doc
  - often a form of Basic
- More recent releases include protection
- Recognized by many anti-virus programs

E-Mail Viruses

- More recent development
- Example : Melissa
  - exploits MS Word macro in attached doc
  - if attachment opened, macro activates
  - sends email to all on users address list and does local damage

Worms

Replicating program that propagates over net

- using email, remote exec, remote login

Has phases like a virus:

- dormant, propagation, triggering, execution
- propagation phase: searches for other systems, connects to it, copies self to it and runs

May disguise itself as a system process

Concept seen in Brunner's "Shockwave Rider"

Implemented by Xerox Palo Alto labs in 1980's

Morris worm

One of best know worms

Released by Robert Morris in 1988

- Affected 6,000 computers; cost $10-$100 M

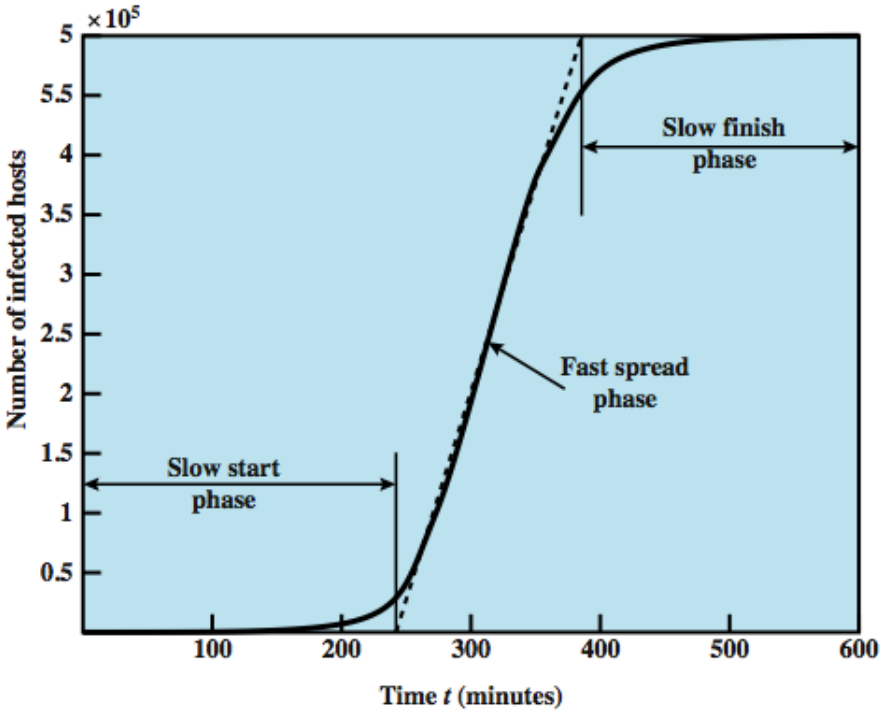
Various attacks on UNIX systems

- cracking password file to use login/password to logon to other systems
- exploiting a bug in the finger protocol
- exploiting a bug in sendmail

If succeed have remote shell access

- sent bootstrap program to copy worm over

Worm Propagation Model (based on recent attacks)



Recent Worm Attacks

Melissa	1998	E-mail worm First to include virus, worm and Trojan in one package
Code Red	July 2001	Exploited Microsoft IIS bug Probes random IP addresses Consumes significant Internet capacity when active
Code Red II	August 2001	Also targeted Microsoft IIS Installs a backdoor for access
Nimda	September 2001	Had worm, virus and mobile code characteristics Spread using e-mail, Windows shares, Web servers, Web clients, backdoors
SQL Slammer	Early 2003	Exploited a buffer overflow vulnerability in SQL server compact and spread rapidly
Sobig.F	Late 2003	Exploited open proxy servers to turn infected machines into spam engines
Mydoom	2004	Mass-mailing e-mail worm Installed a backdoor in infected machines
Warezov	2006	Creates executables in system directories Sends itself as an e-mail attachment Can disable security related products
Conficker (Downadup)	November 2008	Exploits a Windows buffer overflow vulnerability Most widespread infection since SQL Slammer
Stuxnet	2010	Restricted rate of spread to reduce chance of detection Targeted industrial control systems

State of worm technology

Multiplatform: not limited to Windows

Multi-exploit: Web servers, emails, file sharing …

Ultrafast spreading: do a scan to find vulnerable hosts

Polymorphic: each copy has a new code

Metamorphic: change appearance/behavior

Transport vehicles (e.g., for DDoS)

Zero-day exploit of unknown vulnerability (to achieve max surprise/distribution)

Worm countermeasures

Overlaps with anti-virus techniques

Once worm on system A/V can detect

Worms also cause significant net activity

Worm defense approaches include:

- signature-based worm scan filtering: define signatures
- filter-based worm containment (focus on contents)
- payload-classification-based worm containment (examine packets for anomalies)
- threshold random walk scan detection (limit the rate of scan-like traffic)
- rate limiting and rate halting (limit outgoing traffic when a threshold is met)

https://www.gartner.com/en/articles/7-top-trends-in-cybersecurity-for-2022

https://www.gartner.com/en/newsroom/press-releases/2022-02-24-gartner-says-the-cybersecurity-leader-s-role-needs-to

The DarkSide ransomware group was responsible for the Colonial Pipeline Company ransomware incident in May 2021, which led to the company's decision to proactively and temporarily shut down the 5,500-mile pipeline that carries 45 percent of the fuel used on the East Coast of the United States.

Supply Chain Attacks

A software supply chain attack occurs when a cyber threat actor infiltrates a software vendor's network and employs malicious code to compromise the software before the vendor sends it to their customers.

- Newly acquired software may be compromised from the outset, or a compromise may occur through other means like a patch or hotfix.

- The compromised software then compromises the customer's data or system
- These types of attacks affect all users of the compromised software and can have widespread consequences for government, critical infrastructure, and private sector software customers

https://www.cisa.gov/sites/default/files/publications/defending_against_software_supply_chain_attacks_508_1.pdf

Supply Chain Attacks

Lifecycle Phase	Example of Threat
Design	**Hijacked Cellular Devices.** 2016 – A foreign company designed software used by a U.S. cell phone manufacturer. The phones made encrypted records of text and call histories, phone details, and contact information and transmitted that data to a foreign server every 72 hours.
Development & Production	**SolarWinds.** 2020 – An IT management company was infiltrated by a foreign threat actor who maintained persistence in its network for months. The threat actor left the network only after it had compromised the company's build servers and used its update process to infiltrate customer networks.
Distribution	**End-User Device Malware.** 2012 – Researchers from a major U.S. software company investigating counterfeit software found malware preinstalled on 20 percent of devices they tested. The malware was installed in new desktop and laptop computers after they were shipped from a factory to a distributor, transporter, or reseller.
Acquisition & Deployment	**Kaspersky Antivirus.** 2017 – An overseas-based antivirus vendor was being used by a foreign intelligence service for spying. U.S. government customers were directed to remove the vendor's products from networks and disallowed from acquiring future products from that vendor.
Maintenance	**Backdoors Embedded in Routine Maintenance Updates.** 2020 – Thousands of public and private networks were infiltrated when a threat actor used a routine update to deliver a malicious backdoor.
Disposal	**Sensitive Data Spillage.** 2019 – A researcher bought old computers, flash drives, phones and hard drives, and found only two properly wiped devices out of 85 examined. Also found were hundreds of instances of personally identifiable information (PII) spillage, including Social Security numbers, passport numbers, and credit card numbers.

https://www.cisa.gov/sites/default/files/publications/defending_against_software_supply_chain_attacks_508_1.pdf

www.owasp.org

The Open Web Application Security Project (OWASP) is a worldwide not-for-profit charitable organization focused on improving the security of software. “Our mission is to make software security visible, so that individuals and organizations worldwide can make informed decisions about true software security risks.”

There are thousands of active wiki users around the globe who review the changes to the site to help ensure quality. Has a global group of volunteers with thousands of participants.

https://us-cert.cisa.gov/

(CISA is part of the Department of Homeland Security)

Department of Homeland Security (part of US government) offers information and campaigns for awareness of cybersecurity.

The cybersecurity part is available at
https://www.dhs.gov/topic/cybersecurity

CISA – Cybersecurity and Infrastructure Security Agency

www.cert.org

Or https://www.sei.cmu.edu/about/divisions/cert/index.cfm

"At the Computer emergency response teams (CERT) Division of the Software Engineering Institute (SEI) of Carnegie Mellon University(CMU), we study and solve problems with widespread cybersecurity implications, research security vulnerabilities in software products, contribute to long-term changes in networked systems, and develop cutting-edge information and training to help improve cybersecurity."

"We are more than a research organization. Working with software vendors, we help resolve software vulnerabilities. We develop tools, products, and methods to help organizations conduct forensic examinations, analyze vulnerabilities, and monitor large-scale networks. We help organizations determine how effective their security-related practices are".

https://cert-in.org.in/

The Indian Computer Emergency Response Team (CERT-IN) is an office within the Ministry of Electronics and Information Technology of the Government of India. It is the nodal agency to deal with cyber security threats like hacking and phishing. It strengthens security-related defence of the Indian Internet domain.

Software Security Engineering, Julia H. Allen, et al, Pearson, 2008.

Computer Security: Principles and Practice by William Stallings, and Lawrie Brown Pearson, 2018.

Security in Computing by Charles P. Pfleeger, Shari L. Pfleeger, and Deven Shah Pearson Education 2009

Threat Modelling by Adam Shostack, John Wiley 2014

Thank You!