Security Concepts, Backround, and Terminology

Definition of Information Systems Security (INFOSEC):

From the **NSA** (U.S. National Security Agency): Protection of information systems against unauthorized access to or modification of information, whether in storage, processing or transit, and against the denial of service to authorized users, including those measures necessary to detect, document, and counter such threats.

Qu: what do we mean by "*security*"? (Short Answer: CIA) We mean protection of valuable assets, not just from people with malicious intentions ("black hats", or *attackers*) but protection from accidents including environmental disasters. You also must consider how long you need to keep information secret; that can be anywhere from a few minutes to over 70 years.

Attackers work in seven steps: reconnaissance, weaponization, delivery, exploitation, malware installations, command/control, and exfiltration. This model is sometimes known as the *kill chain*. Lockheed-Martin defines these steps this way:

1. **Reconnaissance** - Research, identification and selection of targets, often represented as crawling Internet websites such as conference proceedings and mailing lists for email addresses, social relationships, or information on specific technologies. (This is sometimes called *footprinting*.)
2. **Weaponization** - Coupling a remote access trojan with an exploit into a deliverable payload, typically by means of an automated tool (weaponizer). Increasingly, client application data files such as Adobe Portable Document Format (PDF) or Microsoft Office documents serve as the weaponized deliverable.
3. **Delivery** - Transmission of the weapon to the targeted environment. The three most prevalent delivery vectors for weaponized payloads by APT (*advanced persistent threat*) actors (attackers), as observed by the Lockheed Martin Computer Incident Response Team (LM-CIRT) for the years 2004-2010, are email attachments, websites, and USB removable media.
4. **Exploitation** - After the weapon is delivered to victim host, exploitation triggers intruders' code. Most often, exploitation targets an application or operating system vulnerability, but it could also more simply exploit the users themselves or leverage an operating system feature that auto-executes code.
5. **Installation** - Installation of a remote access trojan or backdoor on the victim system allows the adversary to maintain persistence inside the environment.
6. **Command and Control (C2)** - Typically, compromised hosts must beacon outbound to an Internet controller server to establish a C2 channel. APT malware especially requires manual interaction rather than conduct activity

automatically. Once the C2 channel establishes, intruders have “hands on the keyboard” access inside the target environment.

7. **Actions on Objectives** - Only now, after progressing through the first six phases, can intruders take actions to achieve their original objectives. Typically, this objective is ***data exfiltration***, which involves collecting, encrypting and extracting information from the victim environment; violations of data integrity or availability are potential objectives as well. Alternatively, the intruders may only desire access to the initial victim box for use as a hop point to compromise additional systems and move laterally inside the network.

You can never guarantee 100% safety no matter what you do. So there is a trade-off between how secure you make a system (how hard you make it for attackers to damage or steal your assets, how much protection you provide against power loss, floods, etc.), and how quickly you can recover from an incident, versus how much money you spend, and what sort of procedures you implement. (Your employees and customers won’t accept just any procedure. I doubt a strip-search and full body X-ray for every employee every day would be accepted!)

Security involves protection, detection, and reaction. If there’s no alarm, then sooner or later the protection will be overcome; if there’s no reaction to alarms, you needn’t bother. Protection is only needed for the time it takes to react. In fact, this is how safes are rated: TL-30 means resist a knowledgeable attacker with tools for 30 minutes, and TRTL-60 means resist that attacker with tools and an oxy-acetylene torch for an hour. You buy the safe that you need, depending on (say) security guard schedules.

The balancing of costs and protections makes security an exercise in risk management. First you need to determine what needs protection, and what the threats are that you plan on defending against. This is a *threat assessment* (and produces a *threat matrix*). Next you define *security policy* that, if implemented correctly, will reduce the risk of the identified threats, to an acceptable level. The policy is implemented using various *procedures* and *security mechanisms*.

Security raises issues, not just of protection by technology, but legal, ethical, and professional issues as well. (There is often a difference between personal and professional ethics.) You need to worry about policies, assurance, and security design of software (sadly neglected in programming courses and books!)

For any organization, there are requirements for security. (Here we are concerned only with *Information Systems’ Security* (IS security). Security requirements can be self-imposed or can originate from an external source. For example, companies that process any type of U.S. government information are subject to the provisions of the *Federal Information Security Management Act* (FISMA) and the associated *National Institute of Standards and Technology* (NIST) Special Publications (there are lots of these). Companies that process national security information are subject to security requirements published by the *Committee on National Security Systems* (CNSS). Other U.S. standards are Federal Information Processing Standards (FIPS) and American National Standards Institute (ANSI) standards. Organizations that process credit card data need to comply with the *payment card industry (PCI) Data Security Standard* (PCI DSS). Note that many standards organizations adopt each others' standards, so a given standard may have an ANSI number, a FIPS number, and ISO number, and so on.

There are other requirements at the global, federal, state, or (regulatory) industry level for security of health information, corporate financial data, protection of employees and customers, etc. Nearly all these requirements have the same general security goals: confidentiality, integrity, and availability (CIA), discussed later.

Not all organizations take security seriously. Until a major public breach happens, that is. Some of the organizations who fail to do a proper job get nominated for the annual pwnie awards. Some notable nominees for 2012 include LinkedIn and Yahoo for the loss of emails and passwords, MySQL for accepting any password for any username, and F5 for including the private SSH key in plain text in the device's firmware (thus allowing anyone root access to their Big-IP load balancer).

History of Information Security

Thousands of years old:

- Kings to other rulers (secret agreements)
- Leaders to soldiers (military communications)
- Finance (banking, trading, merchants to partners and branches)
- More recently, expectations of privacy:
  - Medical records (including genetic data)
  - Employment and other records
  - Mail
  - TV viewing, web surfing
  - Computer communications: email, IM, ...
  - Location (GPS and other tracking via cell phones, cars, ...)
- Computer transactions (i.e., buying stuff on-line, auctions, ...)

Early encryption: Julius Caesar' cipher: A->C, B->D, C->E, ...

Early bookkeeping: 8500 B.C.E.!

Double-entry bookkeeping: About 1400 C.E. Every transaction is posted to two separate books, in one as a *credit* and in the other as a *debit*.

Example: Accounts receivable book and the cash account book: A customer pays $100 owed. That is a debit in the Accounts receivable book (the company is now owed $100 less than before, so that account balance goes down), and a credit in the cash accounts book (the company now has $100 more than before).

The two books are maintained by separate clerks. (A slip recording the transaction by some teller has one copy go to each clerk, which is why bank forms used to be multipart carbons).

At the end of the month/week/day (banks are daily), the owner/manager collects the books and compares the totals, which must exactly balance each other.

This system of ***dual control*** prevents any one employee from cheating the firm.

> Sometimes, banks and trading firms get sloppy. In the case of Nick Leeson, who famously brought down Barings Bank in 1995 by losing $1.4bn, there was no proper separation between front and back office. Mr. Leeson was processing the tickets for his own trades. Jerome Kerviel, the Societe Generale rogue trader, is also supposed to have used his intimate knowledge of the bank's systems from his own time working in the back office, and lost his firm 4.9bn euros, the current record. Number two is Yasuo Hamanaka, who lost $2.6bn in metals trading for Sumitomo Corporation in 1996, again by taking advantage of poor controls.
>
> Recently (2011), Kweku Adoboli at Swiss bank UBS, had conducted legitimate derivative transactions, giving the bank heavy exposure to various stock market indexes. But he had then entered "fictitious" hedges against these positions into UBS' risk management system, while in reality he had no hedge in place and was breaching the risk limits that the bank required him to work within. This illegal trading cost $2.3bn.

***Seals* and *Security Printing*:** In 2,000 BC Mesopotamia warehouse keepers would take small marker objects or tables known as bullae, one for each item a customer stored there, bake the bulla into a clay ball known as an envelope and make an official seal on the wet clay. Later a customer could reclaim items by presenting the envelope intact to the warehouse keeper, who would break it open (after inspecting the seals) and allow the customer to take away each item matched by the bulla.

Seals were and are used to authenticate documents. (Ornate seals are supposed to be hard to forge.) For example, a ***signet ring*** would be used to make an impression in some sealing wax melted over a lock. This wax is brittle and if the seal is broken it cannot be resealed without the original signet ring, without detection.

Today seals have evolved into ***security printing***. Examples include currency (the *Intaglio* process and others), watermarks (today, digital watermarks), and price stickers on merchandise that can't be lifted off without ripping.

Tamper resistance means that something can't be changed (or in some cases, examined) easily. ***Tamper evident*** means no changing (or examination) without leaving evidence. Examples include foodstuffs and medicine bottles, and computer cases that warn (or reset) if the case is open (HCC uses these!) Modern versions include ***smart-cards***.

Emission security refers to preventing a system from being attacked using conducted or radiated signals, known as *Van Eck radiation*. Examples include ***power analysis*** (power consumption monitoring): writing a "1" to an EEPROM may consume more power than writing a "0", and analyzing RF signals given off by monitors, cables, etc. allows an attacker to determine what data is written!

In WWI (1914), field phones were used to talk to headquarters from the front lines. These were literally grounded, but it was found the signals could be heard in other field phones hundreds of feet away!

More modern military systems can include TEMPEST hardening, to prevent emission radiation vulnerability. This is not just radiation shielding, but power isolation and timing obfuscation.

> One form of attack is called a *timing attack*. In devices such as computers and smart cards, processing time varies depending on how much of a candidate password is correct. By changing the first character of a password and timing the results, you can see which one takes the least time. Once you have the first character, repeat for the second, and so on.

Together, these non-direct attacks are known as *side-channel attacks*. They can be extremely effective! One such attack known as *padding oracle attacks* was recently (6/2012) used to crack RSA SecurID (and other secure tokens), in about 15 minutes! This attack extracts the private keys stored on devices that use PKCS#11.

Modern information security involves information stored in computers, and thus its history stems from early computer security work done since 1950. Most innovations have been discovered only in the past 30 years or so, making "***InfoSec***" a young discipline.

> The US military has identified electronic communication networks as a new theater of war, and the USAF clearly believes that America should have a robust offensive capability in that theater. They have formed the AFCYBER Command at `www.afcyber.af.mil`. This was absorbed into a larger, joint cyber command at defense.gov. See www.24af.af.mil for the new AFCYBER site.

Early on, computers were not networked, but ran one *batch* job at a time. No security was implemented, but having the next batch job read leftover data stored on disk or RAM was a potential problem.

By the end of the 1960s, multi-tasking computers permitted multiple users to run jobs concurrently. **One of the first information security related publications was in 1968 by Maurice Wilkes, discussing passwords.** Even today, people don't heed that advice!

With the growth of networking computers, security became a more difficult concept to fully understand, let alone implement. For example, Internet protocols evolved from ARPAnet, which was concerned that enemies might crash a vital computer. The protocols invented (which later became TCP/IP) were designed to keep the network functional even if a few nodes were knocked out. However, it was apparently assumed that all users of the network were friendlies, and all nodes and users would "play by the rules". Clearly this assumption is no longer true, if it ever was.

Overview of Security

Define ***Layered Security Architecture***: multiple levels (or layers) of protection, so that if one is compromised there are others to provide protection. An example of this principle is to use proper permissions on files, but also don't allow root to remotely connect to your machine.) This principle is also known as ***rings of security***, and is related to ***defense in depth*** (see below).

Question: Which would you rather have, one virus scanner that is 95% accurate, or three scanners from different providers, where each only offers 80% accuracy?

It turns out that three independent virus scanners, each with only 80% accuracy, is better protection than a single virus scanner offering 98% accuracy! Of course, the same scanner run three times is not any more secure than running it once, so you must be certain that each scanner operates differently to get the protection level you desire.

> The reality is worse than this. Even if a virus scanner were 99% accurate, since the vast majority of files (and network packets) are not malicious, you end up with mostly false positives; maybe one positive in a thousand is real.

A related notion is **defense in depth**. This refers sharing the security burden over many parts of a system, rather than having a single system manage security:

- An application should be written security and deployed in a secure fashion.
- An application/process should manage security of its data.
- A kernel should provide security services to applications, and should enforce security for access to system resources (files, memory, network, GUI, ...).
- Every host should have firewall and other protections, based on policies defined for that specific host.

- Each LAN switch and router should implement security for each LAN in the organization.
- Different collections of LANs in an organization had different security needs and policies. Routers should enforce these.

As you can see, if *defense in depth* is adopted, no single firewall or other security product you can buy is ever going to be sufficient. (No matter what the vendor promises! :-)

Defense in depth applies to the big picture too, the overall *information systems' security* (IS security). **Physical security** controls access into the building, keeps track of who comes and goes, detects and responds to potential intruders. **Personnel security** vets individuals through suitability criteria in accordance with company or government policies, and monitors employee behavior and well-being. (Security and HR often have a close working relationship based on the need to identify potential personnel security risks and mitigate them before they have a chance to materialize.) **Program security** focuses on protecting the key elements of a company's programs: intellectual property ("IP"), customer information, or government secrets. **Security education and training** works to earn true buy-in from employees, converting them into miniature security watchdogs in their own right, dutifully monitoring themselves and their workmates.

Points to keep in mind:

- Install security updates.
- Remove (or don't install) unnecessary software.
- Accept what you can and cannot secure.
- Security management is an balancing act: If you do it wrong users will try to find ways around it. If they *can* get around it you're not doing your job right! If they *can't* get around it they might just give up and leave for another company (or complain to your boss).
- Keep an iron grip on access control. Don't let developers ever touch production servers. Doors that should be locked must have windows so you can see who's in the room.

Many of the security controls designed to protect information systems are directly related to one of the other security disciplines. FIPS Publication 200 establishes 17 general categories of security controls that must be applied to protect information and information systems. They are:

- Access control
- Awareness and training
- Audit and accountability
- Certification, accreditation, and security assessments

- Configuration management
- Contingency planning
- Identification and authentication
- Incident response
- Maintenance
- Media protection
- Physical and environmental protection
- Planning
- Personnel security
- Risk assessment
- System and services acquisition
- System and communications protection
- System and information integrity

Ethics, Laws, and Customs

Laws restrict the availability and use of various security mechanisms, and require the use of others in certain circumstances. Laws and professional standards (*legal and acceptable practices*) may require (or forbid) certain security policies. (Examples: No encryption for email in France, must state privacy policy on commercial U.S. websites, employers cannot require polygraph examinations of employees.) Note it is legal for U.S. companies to require DNA and blood samples from all employees, but it is not socially acceptable.

The U.S. (and the others) government is concerned with criminals and terrorists using encryption technology to defeat surveillance. Unfortunately there is no way to restrict crypto to just non-criminals, so if the government plans succeed crypto may be illegal in the U.S., or be crippled with “back doors” that would allow the security compromised.

The Clipper chip used a data encryption algorithm called Skipjack, invented by the National Security Agency. This algorithm was initially classified SECRET so that it could not be subjected to the peer review that was usual in the encryption research community. The initial cost of the chips was said to be $16 (unprogrammed) or $26 (programmed), with its logic designed by Mykotronx, and fabricated by VLSI Technology, Inc.

But the heart of the concept was ***key escrow***. In the factory, any new telephone or other device with a Clipper chip would be given a “cryptographic key”, that would then be provided to the government in “escrow”. If government agencies “established their authority” to listen to a communication, then the key would be given to those government agencies, who could then decrypt all data transmitted by that particular telephone. In announcing the Clipper chip initiative, the government did not state that it intended to try to make data encryption illegal, but several statements seemed to point in this direction.

Organizations such as the *Electronic Privacy Information Center* and the *Electronic Frontier Foundation* challenged the Clipper chip proposal, but with little effect. Then in 1994, Matt Blaze published the paper *Protocol Failure in the Escrowed Encryption Standard*. It pointed out that the Clipper's escrow system has a serious vulnerability. The Clipper chip was not embraced by consumers or manufacturers and the chip itself was a dead issue by 1996.

The U.S. government continued to press for key escrow by offering incentives to manufacturers, allowing more relaxed export controls if key escrow were part of cryptographic software that was exported. These attempts were largely made moot by the widespread use of strong cryptographic technologies such as PGP, which was not under the control of the U.S. government.

15 years later and the government hasn't given up the idea of outlawing any encryption it can't easily hack. The FBI is currently pushing for encryption backdoors legislation again. See the 9/27/10 New York Times story U.S. Tries to Make It Easier to Wiretap the Internet.

Since the attacks on the U.S. on 9/11/2001, the federal government has passed a large number of laws given them authority to secretly spy on anyone. (See, for example, CALEA, which allows for broad tapping of all communications including Internet use.) Even when a court order is required, you may never know about the surveillance or have any opportunity to challenge it. In a paper published in 2012, A federal judge estimates that his fellow federal judges issue a total of 30,000 secret electronic surveillance orders each year—and the number is probably growing. Though such orders have judicial oversight, few emerge from any sort of adversarial proceeding and many are never unsealed at all. Those innocent of any crime are unlikely to know they have ever been the target of an electronic search.

Security Organizations and Certifications

CERT (cert.org, the *computer emergency response team*, operates a *security coordination center*, CERT/CC (hosted by the Software Engineering Institute of Carnegie-Mellon Univ.). CERT now provides a certification for security incident handling.

www.cybercrime.gov is run by the US Dept. of Justice (US-DOJ) and provides a way to report phishing and other incidents, and advice on responding to incidents.

The **Internet Crime Complaint Center** (IC3) is a partnership between the Federal Bureau of Investigation (FBI), the National White Collar Crime Center (NW3C), and the Bureau of Justice Assistance (BJA). UC3 receives, develops, and refers criminal complaints of cyber crime. The IC3 gives the victims of cyber crime a convenient and easy-to-use reporting mechanism that alerts authorities of suspected criminal or civil violations. (www.ic3.gov)

www.securityfocus.com hosts **Bugtraq**, vulnerability lists, alerts, and job information. A security professional should monitor CERT/CC and Bugtraq (or sign up to the alerting system, mailing lists, or RSS). See also GovInfoSecurity.com, which provides news and other resources related to U.S. Government security issues (run by a private security company).

Disa.mil (and DISA Information Assurance) - The DISA is the U.S. federal agency charged with keeping all federal government computers secure. The standards for this was the DOD Information Assurance Certification and Accreditation Process (DIACAP). In 3/2014, the DoD announced a major change, dropping DIACAP in favor of a family of NIST standards called the RMD (*risk management framework*). (See DoDD 8510.01_2014.)

After 9/11/01, the U.S. DoD (and DISA) spent years drafting new security regulations. One of these, released in 2006, is known as DoD directive 8570. This states (in part) that all military, federal, contractor, and foreign national personnel, in full or part time positions that are for jobs that pose an increased level of IA (***Information Assurance***) risk are "privileged user" positions. Directive 8570 requires these must be filled with personnel who have been trained and hold appropriate certifications with documentation that validates they are qualified for the positions they are hired for. It requires that all users be trained by 2010. It also states that training will be ongoing as positions are filled with new personnel.

Initially, new certifications were created for DoDD 8570, but later, DISA decided to allow ANSI to certify existing IT certifications as 8570-fulfilling, such as from GIAC/SANS. (It has been announced (2013) that this directive will be updated and renumbered to DoDD 8140, the *Cyber Security Workforce*.)

The **DHS** (Department of Homeland Security) runs many security programs (www.dhs.gov). (The DHS, the FBI (Federal Bureau of Investigation), and the military, all play roles in computer and network security for the U.S.)

The U.S. NSA also provides security guides, under their *information assurance* program. See the one for Red Hat 5.

us-cert.gov is run by the DHS, and provides regular advisories and incident reports **which are required reading for all SAs** (subscribe now at www.us-cert.gov/cas/techalerts). (These are summarized in regular bulletins, with links to the National Vulnerability Database for more details including fixes; *show one from a recent bulletin*.) The site is also a rich source of information about security in general.

The ENISA (European Network and Information Security Agency) is roughly the equivalent of US-CERT for the European Union. (They don't produce regular advisories or incident reports however.) In 2012, the EU created CERT-EU.

US-CERT also hosts ICS-CERT, which provides information and alerts for industrial control systems, such as those hacked by Flame and Stuxnet, the 2010 worm designed to cripple Iran's nuclear program (see 6/2012 NY Times story. Stuxnet and Flame were made and deployed by the U.S. (and Israel) as part of a cyber-espionage (or terrorism) program known by the code name *Olympic Games*. More recently (6/2012), ICS-CERT reported a similar attack on the U.S. oil pipelines.

Cyberattacks are a growing threat for all types of devices as they become linked to the Internet or cell phone networks, including automotive systems, security systems, industrial control systems, and medical devices. The security firm iSec demonstrated that they could unlock and start a car by sending text messages to the vehicle's alarm system. A DHS official notes that protecting the devices is especially challenging because they cannot easily be patched on a routine basis.

A computer virus infected the cockpits of America's Predator and Reaper drones in 9/2011, logging pilots' every keystroke as they remotely fly missions over Afghanistan and other war zones. The virus has not prevented pilots at Creech Air Force Base in Nevada from flying their missions. There have been no confirmed incidents of classified information being lost, but the virus has resisted multiple efforts to remove it from Creech's computers.

The **NICC** or *National Infrastructure Coordinating Center*, which replaced the **NIPC** (National Infrastructure Protection Center), is also run by the DHS.

The **FBI** deals with cybersecurity (security of computers and networks) with several initiatives. (www.fbi.gov/cyberinvest/cyberhome.htm)

The FBI also runs Infragard, a group that meets in Tampa and Orlando on alternate months. Coordinated by the FBI, Infragard is a fellowship of federal, state, local, industry, and academic cybercrook catchers and watchers. Infragard has about 33,000 participants in almost 90 cities around the country, and you can apply to become a member yourself. The point is to build an accessible community for the FBI to contact on any given cybercrime problem, especially in the private sector. One cool activity is the cyber war games they conduct.

SANS, the *SysAdmin, Audit, Network, Security* organization (sans.org) is a trusted and the largest source for information security training and certification in the world. It maintains the largest collection of research documents about various aspects of information security, and it operates the Internet's early warning system, the Internet Storm Center. The certification offered by SANS, the GIAC (*Global Information Assurance Certification*, giac.org) is considered by many to be the toughest to get (and hence valuable).

IMPACT, the *International Multilateral Partnership Against Cyber-Terrorism* is an organization that aims to become a platform for international cooperation on cybersecurity. Its advisory board features tech luminaries like Google's Vint Cerf and Symantec CEO John Thompson. The group's forthcoming *World Cyber Security Summit* (WCSS), which will be part of the WCIT (*World Congress of IT*) 2008, is an effort to raise IMPACT's profile as an international platform for responding to and containing cyberattacks. (Quote posted on Ars Technica about this: "*Must be the on-line counterpart organization to the Strategic Homeland Intervention, Enforcement, and Logistics Division. Who'd they hire for names? Stan Lee?*")

CISSP (Certified Information System Security Professional, cissp.com), **ISSEP** (Information System Security Engineering Professional) are certifications granted by *International Information Systems Security Certification Consortium* (ISC^2) isc2.org. Holders of these certifications must know the common body of knowledge that defines the terms and concepts professionals use to communicate, and also includes best practices (including security management), some relevant law (for the U.S. anyway), ethics, and other knowledge and skills.

EC-Council (eccouncil.org) provides "ethical hacking" training, resources, and highly regarded certifications.

There are other **certifications for security** (DoD-8570, security+, ...). HCC now offers AS/AAS/CCC in this. (Prof. Ron Leake runs this program.) Other groups that offer certifications: `iapsc.org`, `ipsa.org.uk`, ...

NIST's ITL (National Institute of Standards and Technology, Information Technology Laboratory) publishes security related information. For example, see **NIST-SP (special publication) 800–27**, which presents a list of 33 basic principles for information technology security. CISSPs need to know many NIST SP 800-X pubs, such as X=12 (Intro to Comp. Sec), 14, 18, 26, 27 (I.T. Security Principles), 30 (risk management). Also OMB *circular A-130*. Another important standard is ISO-27002 (which replaces ISO-17799) *Code of Practice for Information Security Management*.

ISO-27001 is part of a family of ISO/IEC standards (8 so far in 2009) often referred to as ISO27K. This standard deal with InfoSec Management Systems (ISMS). It is often implemented together with ISO-27002, which lists specific security controls. Organizations that claim to have adopted ISO/IEC 27001 can be audited and certified compliant. (ISO/IEC 27002 provides best practice recommendations on InfoSec management for use by those who are responsible for initiating, implementing, or maintaining ISMS, within the standard in the context of the C-I-A triad.)

The National Vulnerability Database (NVD) is a repository maintained by the U.S. government of all known vulnerabilities. This includes security related software flaws, misconfigurations, and product names and versions. The NVD also includes an impact statement for each vulnerability (uses CVSS). It keeps the vulnerabilitity lists for various products (Windows 7, Fedora, etc.), known as *checklists*, in SCAP format, allowing automated vulnerability scanning as well as compliance reporting.

The Common Vulnerabilities and Exposures (CVE) database (at Mitre.org) is the most comprehensive, internationally and freely available database of known vulnerabilities and the malware that exploits them. It assigns a unique identifier to each. These CVE numbers are wildly used everywhere, including SCAP, NVD, US-CERT bulletins, etc. Every new exploit is recorded here.

Security Regulations for Credit/Payment Card Industry (PCI)

Many commercial organizations handle electronic payment information, such as by processing credit (also debit) card payments. In the past each brand of card (Visa, Amex, ...) had security regulations for handling customer names, addresses, and other collected payment info. (For example: an un-encrypted log file is not allowed to store a complete credit card number, but you can store the first 6 and last 4 digits.)

Merchants and payment processors need to file notices of compliance with the PCI *Data Security Standard* (PCI DSS), usually done by an internal audit. Recently all major banks have agreed to a single set of standards. The ***PCI Security Standards Council*** was founded by Amex, Discover, JCB, Visa, and MasterCard. Each organization agreed to adopt the standards that the group decides on. See www.PCISecurityStandards.org for more information.

See also www.linuxsecurity.com (a good site for information, but not for certification), and mybulletins.technet.microsoft.com (Security bulletins from MS, customized for your computer).

The number of payment card data breaches and fraud (stolen card data from self-service gas stations, Walmart, etc.) is huge and growing. Currently, the customer and retailer is protected by law from liability; the bank associated with the card must pay. One technology can help with this, *smart cards*, also known as *chip* cards. The swipe stations that require both a chipped card and a user-entered PIN are known as *chip-and-pin*.

As of October 1, 2015, liability for payment card fraud will shift from card companies to retailers if the retailers have not upgraded their terminals to accept chip-based payment cards. The cards have been used in Europe for 10 years, but the US has been slow to adopt the technology, largely due to the associated costs that merchants will have to bear. The Target breach is what drove the industry to set a timeline for adopting the standard. — ComputerWorld

Live CDs / DVDs

There are a number of bootable Linux live CDs, that contain collections of useful security tools. (A good list can be found at www.securitydistro.com/security-distros.) Examples include Auditor, Phlak, Whoppix, and Pentoo, but most of these haven't been updated in several years. Some more recent examples include BackTrack (a merger of Auditor and Whoppix), Network Security Toolkit (NST), Helix, DEFT, and DVL (a purposely broken distro, designed for learning and training).

The Black Market for Malicious Hackers

[*Adapted 10/17/07 from: en.wikipedia.org/wiki/E-gold and www.cmu.edu/news/archive/2007/October/oct15_internetblackmarkets.shtml.*]

Today there is a thriving market where people can buy stolen credit card numbers, purchase malware or spam software, even hire developers. This market even uses a *reputation* system (think about e-bay or `Amazon.com`'s Marketplace), so buyers can tell if a seller is reliable or just going to take their money and run. The 2007 CSI survey reported that U.S. companies on average lost more than $300,000 to cyber crooks.

"*These troublesome entrepreneurs even offer tech support and free updates for their malicious creations that run the gamut from denial of service attacks designed to overwhelm Web sites and servers to data stealing Trojan viruses*," said Perrig, a professor at CMU. "*...monitoring found that more than 80,000 potential credit card numbers were available through these illicit underground web economies*", said a PhD student at CMU working with Perrig.

You can easily hire hackers these days, say from the Hackers List website.

As of 2012, **LilyJade malware** is available on malware markets for around $1,000. This is a web browser extension that works with IE, Firefox, Chrome, and other browsers, on any platform. LillyJade appears to be focused on click fraud, spoofing ad modules on Yahoo, YouTube, Bing/MSN, AOL, Google and Facebook. It also has a Facebook-based proliferation mechanism, which spams users with a "*Justin Bieber in car crash*" style message, complete with a link to a location where a user can be infected. (Reported in The H 5/23/12.)

Most personal computers that get infected with malware are targeted by pay-per-install (PPI) services, which reportedly (Technology Review 6/9/11) charge up to $180 per 1,000 successful installations. Typical installation schemes involve uploading tainted programs to public file-sharing networks, hacking legitimate Web sites to download automatically the files onto visitors' machines, and quietly running the programs on PCs they have already compromised.

> Another "black" market is called the ***Internet Water Army***. In China, paid posters are known as the Internet Water Army because they will "flood" the Internet for whoever is willing to pay. The flood can consist of comments, gossip and information (or disinformation). Positive recommendations can make a huge difference to a product's sales but can equally drive a competitor out of the market. [From TechnologyReview.com blog post on 11/28/2011.]

Payments on the black market

Whatever the purchases, a buyer will typically contact the black market vendor privately using email, or in some cases, a private instant message. Money generally changes hands through non-bank payment services such as **e-gold**, making the criminals difficult to track. **e-gold.com** (and others such as **e-bullion**) provide accounts backed 100% by gold or silver deposits. These companies allow instant transfers from one account (the buyer's) to another (the seller's). Since these transactions are not part of the international banking system, they are usually impossible to trace. E-gold is also known as *private currency* as it is not issued by governments.

Opening an account at e-gold.com takes only a few clicks of a mouse. Customers can use a false name if they like because no one checks. (Some such as **GoldNow** are more reputable and do require security checks.) With a credit card or wire transfer, a user can buy units of e-gold. These units can then be transferred with a few more clicks to anyone else with an e-gold account. For the recipient, changing e-gold back to regular money is just as convenient and often just as anonymous.

Note that e-gold doesn't convert the deposits to any national currency. E-gold does not sell its e-metal directly to users. Instead *digital currency exchangers* such as **OmniPay** (a sister company of e-gold) and numerous independent companies act as market makers, buying and selling e-metal in exchange for other currencies and a transaction fee. In this manner, e-metals can be converted back and forth to a variety of national currencies.

Bitcoin

Since 2009, a virtual (“crypto”) currency has become popular among hackers. Bitcoin is a pseudonymous (that is, a fake online ID) cryptographic currency, used by the hacker underground to buy and sell everything from servers to drugs to cellphone jammers. Bitcoin is a real currency, once valued about 3 times higher than the US dollar; its value has fallen much lower in recent years (between $4.50 and $5.50, in 2012). It has become the standard currency on *Silk Road* (an underground online market, mostly for drugs) and some porn and gambling sites. (See BitCoincharts or Preev for current values.) Bitcoin is achieving some respectability of late, and some legitimate vendors now accept Bitcoin. (*Show EFF donation page.*) Bitcoin is not the only virtual currency; Litecoin and others are starting to show up (2013).

In 10/2013, the U.S. seized control of Silk Road, shut it down, arrested its sys admin (who called himself “Dread Pirate Roberts”), and seized 3.6 million in Bitcoin. Bitcoin fell 20% in response.

It didn’t take long for Silk Road 2.0 to appear. Many other illicit, or “darknet” sites, including Silk Road 2.0, were seized by a coordinated effort of 16 nations early in 11/2014. This was called “Operation Onymous”.

One of the (now fugitive) site operators has posted server logs, speculating how law enforcement was able to break the Tor security. It is believed that the government operates several Tor nodes, and launched a dDoS against the rest, forcing most Tro traffic through their nodes. That allowed them to see where the traffic originated, and act as a “Man in the Middle” attack to break the encryption. However, this isn’t known for certain. (See Ars Technica story.)

Unlike other currencies, Bitcoin is not tied to any central authority. It is designed to allow people to buy and sell without centralized control by banks or governments, and it allows for pseudonymous transactions that aren’t tied to a real identity. In keeping with the hacker ethos, Bitcoin has no need to trust any central authority; every aspect of the currency is confirmed and secured with strong cryptography. Individuals earn Bitcoins by selling stuff, or by *mining* (donating CPU cycles to organizations for profit) and store them in their Bitcoin wallet — a data file containing private crypto keys. User can cash in their Bitcoins for traditional currency at various exchanges such as mtgox.com, TradeHill, BitFloor (which was hacked in 9/2012), BitInstant, or Bitcoin Centeral.

Bitcoin Central, a Bitcoin exchange that is popular in the Eurozone, says it has secured approval from regulators to operate as a bank under French law.

(Paymium is the French company that operates Bitcoin Centeral.) Euro-denominated funds will be insured by the Garantie des dépôts, the French equivalent to the US FDIC. The accounts will also be integrated with the French banking system, so users can have their paychecks automatically deposited into their accounts and converted to Bitcoins.

> TradeHill, the second-largest Bitcoin exchange at the time, announced that it was closing its doors in 2/2012. In a statement, CEO Jered Kenna cited regulatory problems and the loss of $100,000 in a dispute with one of its payment processors as major factors in the decision. He has pledged to open a new site once these issues have been resolved.

Update: many governments are now beginning to recognize Bitcoin, and are deciding how to deal with it. In 8/2013, Germany decided that Bitcoins were subject to capital gains taxes. At this same time, Bitcoin ATM machines, allowing one to deposit cash (converted to Bitcoin) or withdraw cash (converted from Bitcoin) are showing up; Canada has many of these.

For technical details and information on how to use bitcoins, or become a Bitcoin miner, see Bitcoin.it (especially the FAQ section).

Fighting Back

In January 2006, Business Week reported on the use of the e-gold system by ***Shadow Crew***, a 4000-strong international crime syndicate involved in massive identity theft and fraud. One person reportedly connected to Shadow Crew as an e-gold customer and has moved amounts ranging from $40,000 to $100,000 a week from proceeds of crime through e-gold.

To combat this type of crime CMU researchers propose using a ***slander attack***, in which an attacker eliminates the verified status of a buyer or seller through false defamation. By eliminating the verified status of the honest individuals, an attacker establishes a *lemon market* where buyers are unable to distinguish the quality of the goods or services.

The researchers also propose to undercut the burgeoning black market activity by creating a deceptive sales environment, by a technique to establish fake verified-status identities that are difficult to distinguish from other-verified status sellers making it hard for buyers to identify the “honest” verified-status sellers from dishonest verified-status sellers. "So, when the unwary buyer tries to collect the goods and services promised, the seller fails to provide the goods and services. Such behavior is known as ***ripping***.

I’m not sure if it counts as fighting back, but a new Trojan horse program discovered in mid-2011 seeks out and steals victims’ Bitcoin wallets, the same way other malware goes for their banking passwords or credit card numbers. The malware, Infostealer.Coinbit, is fairly simple: it targets Windows machines and zeroes in on the standard file location for a Bitcoin wallet. It then e-mails the wallet to the attacker by way of a server in Poland, according to Symantec. The first actual theft of Bitcoins was reported in June 2011 by a user who claimed a hacker transferred 25,000 BTC from his machine, theoretically worth about $500,000 at 2011 exchange rates. Another bit of malware called *Stealthcoin* debuted in 2010 that's designed for turning a botnet of compromised computers into a covert parallel Bitcoin mining machine.

Lecture 2 — Security Concept Definitions (*in no particular order*)

MAC and DAC

Access controls are designed to prevent unauthorized use of resources. Once a user is *authenticated*, the system consults access tables to determine if some attempted access is allowed.

Mandatory Access Controls provide a policy based security system. This means that if the policy says some resource (file, device, port number) can only be accessed by XYZ, then not even the root user can change that! Such policy changes must be made to policy files, which are usually human readable text files that in turn get compiled into a more efficient form. The compiled policies are loaded into the OS at boot time and can't be changed without a reboot.

The policy may allow users and administrators some leeway in controlling access. Users typically are allow to grant read, write, and execute (traditional Unix permissions) to their files. An administrator may be granted the privilege of changing the ownership of all but a few files and programs.

Discretionary Access Controls don't have the OS enforce any central policy. Instead, the owner of a resource (or root) is granted total control over access of their resources. This doesn't mean there aren't polices, however! PAM and other ad-hoc mechanisms exist to enforce policy. However since the OS doesn't make you use these mechanisms, an attacker can bypass them.

Multi-Level Security

A ***MLS*** (*multi-level security*) system allows an administrator to assign ***security levels*** to files, such as unclassified, secret, most-secret, top-secret, read-this-and-die, ... A given user is also assigned a security level called their *security clearance*. Then access is denied if your security level is less than that of the file (the *no read-up* property). (Don't confuse MLS with multi-layer (rings of) security.)

The *Bell-LaPadula* model is often used to describe MLS. It focuses on data confidentiality and access to classified information. In this model, the MLS system prevents a user with a high security clearance from creating files with a lower security level (the *no write-down* property). For example, a process with *secret* label can create (write) documents with labels of *secret* or *top-secret*, and read documents with labels of *secret*, *confidential*, or *unclassified*. (Note a process with a *top-secret* label still can't access a document when the DAC or other MAC policy forbids it.)

Special privilege is required to *declassify* (assign a less-restrictive security level to) files. Instead of declassifying documents, a higher-cleared individual can create and share *sanitized* documents with less-cleared individuals.

A *sanitized* (or *redacted*) document is one that has been edited to remove information that the less-cleared individual is not allowed to see.

MLS is common in military and some commercial data systems; each use has, over time, defined a set of such levels. However, sharing documents between organizations or nations is a problem, because each has different rules for labels such as “secret” or “confidential”. MLS also requires a *trusted computing base* (TCB) operating system. It is rarely enforced by operating systems today.

The ***Biba model*** is related to the Bell-LaPadula model. However, instead of confidentiality as the goal, data integrity is. The rules are the opposite: no write up (a less trusted process can’t create/delete/modify higher-security resources; a captain can read orders created by a major but not alter them) and no read down (a high-trust process can’t access low-trust resources, since it shouldn’t believe them anyway).

The Biba model is what Windows UAC enforces. Each process and each file or other resource is assigned a security level of low, medium (the default), high, or system. Protected-mode IE (PMIE) can run at low, preventing it from accessing normal or system files. User processes run at medium, and thus can’t modify system resources. (Admin users have both a low (“filtered” and a normal (“linked”) SID; if the application requires more privilege than the filtered SID, the user is prompted to elevate their permission to the unfiltered level.

OpenBSD has a kernel policy for this, mac_biba.

Related access control techniques designed to prevent enemy access include weighted naval codebooks and lead-lined dispatch cases that could be tossed overboard in the event of immanent capture. Special papers could be used that instantly burn to fine ash, or are water-soluble.

Hardware Based Protection

Access controls are not the only protection mechanisms available (but they are the most useful and most used). Hardware based protection is available as well, limiting access to resources in a hierarchical manner. Most CPUs define such levels and assign a protection *level* (or *domain*, or ***ring***) to blocks of memory and I/O addresses. **Intel x86 CPUs have four rings, 0 through 3.** The lower the number, the more access is allowed. The ring level of running code is stored in a special CPU register. The level currently in use is called the CPU *mode*. In addition to memory and ports, some CPU instructions (about 15) are prohibited in higher numbered rings.

By default, applications run in the highest ring (3) and have no access to resources marked by a lower ring number. Instead, such code must request more privileged code (the kernel) to access the resource on its behalf, allowing the system to determine if access should be allowed, to log the request, to alert the user, or some combination of these. An attempt to access a protected resource or run a restricted instruction from a ring that doesn't allow it, triggers a *general protection fault*.

While different CPUs may have many rings **most OSes only use two: ring 0 (*supervisor mode* or *kernel mode* or *system mode*) and ring 3 (*user mode*).** Although rings 1 and 2 are generally unused, they are considered part of *system mode* if used.

While interest in supporting more than two ring levels has been traditionally low, *virtualization* has caused renewed interest. By running the hypervisor in ring 0 and the various OS kernels in ring 1, the hypervisor can easily intercept resource access attempts and fool the OS into thinking it is accessing the resources directly. Without this feature, each OS requires modified device drivers (and other changes) to invoke the hypervisor.

> To allow fully virtualized (unmodified) OSes to run, newer CPUs have special hardware support. They provide two modes, *root* and *non-root* modes, each with protections rings 0 to 3. The hypervisor runs in root mode and the unmodified guest OSes run in non-root mode.

It may take hundreds of CPU cycles to do a *context switch*; that is, to change the mode. For this reason, there is a security-performance trade-off and some OSes include application code in ring 0 to avoid the frequent context switches. (DOS, but not modern Windows, used ring 0 for all code. Windows runs the GUI in ring 0. Even Unix/Linux systems are not immune to the allure of faster performance; whole web servers can be loaded as "device drivers" and run in ring 0!)

> Many other hardware-based protections can exist such as VTx, used for V12N. Many Intel CPUs now support "TXT" ("Trusted eXecution Technology"), which (along with "Trusted Platform Modules", or "TPM") can be used to validate parts of the OS and boot loader (using public key technology), then lock them down ("seal"). This means safety from many rootkits, but also allows for DRM enforcement. (So some users will need to learn how to "jailbreak" their PCs, the way they do now for smart phones.)
>
> On Linux, the TPM (if present) can only be accessed by a socket; that socket in turn is only accessed by a daemon, `tcsd`. This daemon listens on port 30003 for various commands, and is generally the only way to access the TPM. (If running some host without a TPM, turn this off or you get boot log error messages from it.)

Hostids

A Unix host is assigned a unique 32 bit ID number known as the "host ID". These are supposed to be unique, at least within a single administrative domain. The number could be a serial number of the hosts motherboard or CPU, but today is often (part of) the MAC address of the first Ethernet card. On Solaris boxes the host ID is set in the firmware (NVRAM) when the OS is installed. (Note that re-installing may not result in an identical host ID.)

You can view the host ID with the **`hostid`** command. Although POSIX defines a system call (kernel API) `gethostid()`, it doesn't mandate any user command to return this value. However most systems provide a `hostid` command. (If not it is trivial to write a C program for this.) Linux provides commands to get and set this value (long live open source!)

So why does this matter? For one reason some Unix software requires the host ID registered with the vendor to validate a software license; if none (or a dup) is found software may not install (or work)!

There are far too many software vendors who love proprietary lock-in, hardware lock-in, or a combination of both (e.g., "Windows activation"). This can be called *trusted computing* (in the sense of "we trust you to pay us big bucks if we give you no choice"). A hardware-specific ID with software to query the host ID is usually how it's done, but the host ID is set in the kernel or firmware.

When updating systems or when recreating virtual systems, you may need to "set" the host ID value. Naturally details for doing this are not published (the vendors prefer you to buy another license) but you can find directions in various places for different distros such as in this blog post for Solaris. (Here someone traces the obfuscated Solaris code for setting the hostid.)

[*Much of the following material is from the excellent "Introduction to Computer Security" by Matt Bishop, (C)2005 Addison*-*Wesley*]

CIA

Security requirements, whether self-imposed or mandated by an external agency or customer, are all designed to address the three fundamental objectives of computer security: confidentiality, integrity, and availability. These three concepts are often referred to as the **CIA** triad. FIPS Publication 199 defines these security objectives in precise terms, but we can define them as:

- **Confidentiality** Keeping information, resources secret from those that shouldn't know about them. This related to the concepts of ***authentication*** (who is requesting resources?) and ***authorization*** (what resources is a given person/program allowed to access?). A loss of confidentiality means the unauthorized disclosure of information. You must also consider the duration for which the information must be kept confidential; security measures that can keep secrets for a few weeks or months, may not be good enough to protect military secrets.

Correctly enforced, confidentiality prevents unauthorized disclosure of information. The most common method of enforcing confidentiality is with encryption. (But don’t forget other ways, such as limiting access to the data.)

- **Integrity** Keeping information and resources trustworthy (preventing improper or unauthorized modification, deletion, addition, or replacement, called *corruption*). This relates to the concepts of *credibility*, *authentication*, and *authorization*.

  It is also important to keep data *externally consistent* (data in system accurately reflects reality) and *internally consistent* (ledger books balance, totals match the sums, etc.) A common risk with standard relational databases is losing internal consistency, which is addressed by normalization.

  Integrity mechanisms can prevent corruption or detect corruption (that is, verify data hasn’t been modified). Using message digests (*hashing*) is a very common way to implement integrity.

- **Availability** Keeping information and resources accessible to authorized persons, whenever they are otherwise allowed access. This related to ***reliability***. One key method of availability is by using redundancy: RAID, hot-standby servers (fail-over clusters), multi-homing, and excess network capacity. Another way to keep data available is by using backups.

There are other security concepts however, beyond CIA. Networking router and switch vendors rarely worry about confidentiality or integrity (perhaps they should). *Auditors* need to know your systems correctly implement your security requirements and policies, call *information assurance.* CIA doesn’t address concerns about *authentication* (who you are), *authorization* (who can do what), and *accountability* (who did what and when).

CIA alone just isn’t enough anymore.

AAA

The three concepts of authentication, authorization, and accountability are often implemented in network equipment (such as routers), and is referred to as the **AAA** triad:

- **Authentication** Representing ***identity***: Users, groups, roles, certificates

  Authentication refers to the process of establishing the digital identity of one entity to another entity. An *entity* is a client (a user, process, host, etc.) or a server (service, host, or network). Authentication is accomplished via the presentation of an identity and its corresponding credentials.

Related is the concept of ***identity proofing***. This means verifying identity before issuing credentials. Afterward, the user/process/system can usually authenticate merely by presenting the credentials that were issued. (HTTP authentication works this way, using a cookie. So does Kerberos.) Having an HR manager introduce the new employee to a SA, in person, is another example.

Identify proofing is sometimes used even after credentials have been issued, as an extra check on identity. Zip-code checks at gas stations, using a PIN in addition to your ATM card at the bank, or re-entering your password to run `chfn`, are all examples.

When accessing some protected service or resource of some system you must *authenticate* yourself (prove who you are) before the system can decide if you should be allowed access. There are three common ways, or *factors*, to prove identity. (Additionally, you can have someone known to be trustworthy vouch for you; this is related to the notion of *trust*.) To prove identity, one or more proofs are submitted; usually each is a different factor. A requirement to present two factors to prove identity is called two-factor (or dual-factor or TFA) authentication; with more than two, it is called ***multi-factor authentication*** (MFA). The three factors are:

- Something you know (such as a password or credit-card number)
- Something you possess (such as a key-card, smart-card, dongle, or key-fob)
- Something you are (*biometrics*, such as a fingerprint)

The most common method used is by providing information only you and the system know (i.e. a password, a.k.a. a *shared secret*). Note the system doesn't have to know the secret; it is enough if they can verify you know the secret. (This is why plain-text passwords don't have to be stored on a server.)

Using a phone-based system (i.e. providing a phone number that can be checked) can be used to ***call-back*** the user trying to access a resource to confirm it really is them. Phone calls are made to the (presumably real) user when someone attempts to log in as him or her. The user can then punch in a code on the phone. Phones an also serve as *fraud alerts*. Email call-back ("verification email") systems are also commonly used, especially when resetting passwords.

Another possibility is proving possession of something only you (should) have, such as a smart card swipe card, credit-card (using CVS number), dongle, or RFID. Using a key fob that displays a number that changes every minute, that only the server and the fob know, is another example (RSA SecurID (*show*), YubiKey).

Cell phone apps can take the place of dongles or fobs. Note that since modern cell phones (2012) only run one app at a time, you can't use the AWS-MFA app ("virtual dongle") and the AWS console app!

Gmail has an option for two factor authentication, but it had (1/2013) a back-door built-in that could be used to not only bypass the extra step, but could be used to bypass authentication altogether! Called *application specific passwords*, using these other passwords would disable TFA, useful when fetching mail from, say, Thunderbird. But since the password was sent in plain text, it could be intercepted and used.

RFIDs are subject to a number of security issues. U.S. passwords have RFID tags that transmit sufficient information to enable an attacker with an RFID reader (they are small) to steal your identity and create credit cards in your name. (You can buy metal sleeves for your passport now.) The "low tire pressure" warning you get with newer cars works by having RFID tags in the tires, which each broadcast a unique ID (and the tire's pressure) unencrypted. This can be read up to 40 meters away, so by having readers along highways and outside buildings, any car's location can be tracked at any time.

New types of RFID devices require someone touching them to operate. So your passport, smart card, etc., can't be read while in your pocket.

Increasingly ***biometric identifiers*** such as a fingerprint, eye print, handprint, voiceprint, or even lip prints may be used. (Imagine having to kiss your computer to log in.) Increasingly, ATM machines use vein prints of your hand. Biometrics have some problems; they are not *yes or no*. Like spam scanners, they have problems with false positives and false negatives. Depending on the situation, you can tune the scanner to favor one or the other. Another problem is when the biometric data on file for you has been stolen. The bank can't just issue you a new fingerprint! Another problem is fake biometrics: fake fingers (or in one gruesome case, real fingers cut from a victim) or special latex gloves can defeat advanced fingerprint readers, including ones with "liveness" tests such as sweat, temperature, or pulse.

The next generation of biometrics include continuous monitoring, making sure a living person is still there: pulse, temperature, skin conductivity, heartbeat monitors, even ECGs (brainwave monitoring).

In 2004, a ring of thieves installed tiny cameras in the locker room of a golf club in Japan, to record the PINs people used for their lockers. Then, when those golfers were out playing, the thieves opened their lockers, scanned the magnetic strip of any bank cards found. Later, they copied that data to blank cards, and guessed to use the same PIN as the locker combo. By the time they were caught in 1/2005, they had stolen almost $4 million.

In response, the Japanese government demanded safer ATMs, and the *vein scanners* resulted. This biometric scanner has been very successful, and the technology is being deployed world-wide.

As mentioned above, another technique is to rely on ("trust") another organization for authentication. You present proof (a credential) that the organization has authenticated you, by showing a valid passport, driver's license, major credit-card, Kerberos ticket, etc. Proving you have a valid email address or phone number falls into this category.

Many of the most convenient authentication schemes are weak and easily circumvented (spoofed). When using such weak methods, users typically need to use a two-factor (or multi-factor) authentication system. For example, entering something that only they would know, and use something that only they would physically have on their person. This means entering a login and PIN or password, along with the use of a USB authentication key or smart card, for example.

The patterns that bank customers typically follow when choosing a four-digit PIN code gives hackers a 9 percent chance of correctly guessing their ATM code. Researchers studied 1.7 million "leaked" PINs and found that 23% of users base their PIN on a date, "and nearly a third of these used their own birthday." (InformationWeek.com)

CAPTCHAs — proving you are human

Some systems don't care who you are, as long as you are a person and not some "bot" or program. Services such as web-based guest-books or the WHOIS database are available to anyone, but not to automated programs (*robots* or *bots*). In these systems, you must only prove you are human. Typically this is done by having the use do a task easy for most people but difficult for machines.

A ***CAPTCHA*** (Completely Automated Public Turing test to tell Computers and Humans Apart, if you can believe that) is an obscured graphic of some text, shown on a web form. A human can usually read and then enter the text, but a program has a difficult time reading the text. (Captchas have become an incentive to improve OCR systems!) Other tasks include picture recognition (e.g., "click on the one picture above of a moose"), math problems stated in English (show Ars Technica test, original URL: http://contact.arstechnica.com/spammy/40706/), or audio CAPTCHAs. SANS Institute (Chief Research Officer Johannes Ullrich, Sys Admin, V16 N4 (April 2007), "Minimizing Content Form and Forum Spam" pp.30–ff) reports that

CAPTCHAs are effective at reducing spam, but also reduce legitimate traffic by about 20%.

Ars Technica 9/1/08 - Google's Gmail CAPTCHA was broken in February 2008, followed by that of Hotmail in April.

NetworkWorld.com 11/1/11 - Researchers from Stanford University have developed an automated tool that is capable of deciphering text-based anti-spam tests used by many popular websites with a significant degree of accuracy. With their tool, the tests used by Visa's Authorize.net payment gateway could be beaten 66 percent of the time, while attacks on Blizzard's World of Warcraft portal had a success rate of 70 percent. (The only tested sites where CAPTCHAs couldn't be broken were Google and reCAPTCHA.) Authorize.net and Digg have switched to reCAPTCHA since these tests were performed. These researchers have also successfully broken audio CAPTCHAs on sites like Microsoft, eBay, Yahoo and Digg.

The Stanford researchers came up with several **recommendations to improve CAPTCHA security**. These include randomizing the length of the text string, randomizing the character size, applying a wave-like effect to the output, and using collapsing or lines in the background. Another noteworthy conclusion was that using complex character sets has no security benefits and is bad for usability.

Researchers have fought back by incorporating images into CAPTCHAs but this is only effective against bot-driven CAPTCHA crackers, and while automated attackers may be responsible for a majority of the CAPTCHA-breaking attempts that occur every day, they no longer account for the entirety.

Dancho Danchev (writing for ZDNet) reports on the emergence of CAPTCHA-breaking as an economic model in India. He reports that large CAPTCHA-breaking companies often farm work out to multiple smaller businesses. **CAPTCHA-cracking (referred to as "solving" in marketing parlance) is a booming sector of the Indian tech economy.** Danchev reports that CAPTCHA-crackers can earn more per day than they can as legitimate data processing centers.

It's hard to see how researchers will find a CAPTCHA that legitimate customers can read that remains illegible to humans paid to solve them. One recent (12/2009) attempt is a new mechanism designed to stop computer algorithms programmed to beat current CAPTCHA technology. Danny Cohen-Or led a research team that created video CAPTCHA code that uses an emergence image, an object on a computer screen that only becomes recognizable when it is moving.

For example, an eagle or a lion in a pastoral mountain setting. Humans are very good at identifying these types of images while computers are not.

- **Authorization** Policy stating who is allowed to do which actions on what resources: ACLs, capabilities (tickets). Authorization refers to the granting of specific privileges (including none) to an entity (e.g., process, host, AS, or user). This is based on their authentication (proven identity), what privileges they are requesting, the current system state (e.g. the service is available), or on restrictions such as time-of-day restrictions, physical location restrictions, or restrictions against multiple logins by the same user.
- **Accountability** Keeping track of who does what and when. Log files are an example, as is the tracking of the consumption of network resources by users. The data generated is used for management, planning, billing, auditing, or security (e.g., log monitored by an IDS). Typical information that is gathered includes the identity of the user, the nature of the service delivered, when the service began, and when it ended.

AAA requirements are defined in RFC-2989, and evaluation of some AAA protocols (such as RADIUS) are discussed in **RFC-3127**. A server that provides AAA services is said to implement the AAA architecture.

Information Assurance (IA)

IAS is the set of controls and processes, both technical and policy, intended to protect and defend information and information systems by ensuring their confidentiality, integrity, and availability, and by providing for authentication and non-repudiation (see below). It is also known as ***Information Assurance and Security*** (IAS); in U.S. government/military circles, the new hot term for INFOSEC and SIGINT is “Cybersecurity”, but that is just another term meaning “Information Assurance” (IA). Some U.S. Federal jobs now require IA certification, and IAS is now (2013) a core knowledge area of a four-year computer science degree program.

From an IA perspective, there are five pillars (or core concepts): confidentiality, integrity, availability, authentication, and non-repudiation.

Non-repudiation is related to accounting (and data integrity). This means you can’t convincingly deny some action such as sending a message, receiving a message, entering/changing/deleting data, etc. (Example: ordering an expensive item on-line, then refusing to pay claiming you never ordered the item.)

Other important concepts include:

- **Security Policies** A *security policy* is a statement of what is and what is not allowed. This is also called a ***specification***. Policies can be stated by mathematics and tables that partition the system *states* into allowable (secure) and not allowed (insecure) states. More commonly, vague descriptions are used, which can lead to ambiguity (some states are neither allowed nor disallowed, or both).

When two or more entities communicate or cooperate, the combined entity has a security policy based on the individual policies. If these policies contain inconsistencies the parties involved must resolve them. (Example: proprietary document provided to a university.)

- **Polyinstantiation** is the concept of creating a user or process specific view of a shared resource. (So Process A cannot affect process B by writing malicious code to a shared resource, such as /tmp.) The term refers to a similar concept for databases, where different users get different views of a database (think of a virtual table). (For cryptography, *polyinstantiation* means to have a copy of a secure key in more than one location.)

  The `pam_namespace` module creates a separate *namespace* for users on your system when they login. This separation is enforced by the Linux operating system so that users are protected from several types of security attacks. Using this module, PAM creates a polyinstantiated private `/tmp` directory at login time. This is transparent to the user logging in; the user sees a standard `/tmp` directory and can read and write to it normally. However that user cannot see any other user's (including root's) `/tmp` space or the actual `/tmp` file system.

  The kernel supports polyinstantiation too. `/proc/self` gives a private view of kernel resources. SELinux also supports this, with special mount options.

Security Mechanisms

A *security mechanism* (or security *measure*) is any method, tool, or procedure used to enforce a security policy, or to reduce the impact (and frequency) of threats. Not all mechanisms are tangible.

Security mechanisms can be used to *prevent* an attack, to *detect* an attack has occurred (or is occurring), or to *recover* from an attack. Usually you will used multiple mechanisms to meet all three of these goals.

Qu: what is the goal of a password mechanism? (Ans: prevention, audit; there's overlap.)

Prevention mechanisms keep the system functioning normally and available during any attack. Such prevention mechanisms include ***over-provisioning*** and ***fault-tolerance***, and can be used in safety-critical systems where the high cost is deemed acceptable.

Detection mechanisms are also used to monitor the effectiveness of other mechanisms.

Recovery mechanisms are deployed when detection of an attack has occurred. The first part of recovery is to repair the damage done by the attack and to restore normal service levels. **Recovery is not complete until an assessment of the incident has been made and preventative measures are put into place.** This may include a change to policy, design, or configuration of services, the addition of extra mechanisms, or legal action.

- **False Positives And False Negatives** These occur when checking incidents against a security policy. A *false positive* occurs when the security scanner reports something as suspicious when in fact nothing is wrong or insecure. A *false negative* is when the scanner fails to report a problem when in fact one exists.

  Neither of these can be completely eliminated in any real scanner, and there is something of a trade-off between them. Most scanners opt for more false positives to generate fewer false negatives. However this can lead to the *BWCW* (boy who cried “Wolf!”) syndrome, leading administrators to ignore the scanners reports. Add-on tools (usually Perl scripts) can attempt to *filter* log files and reports to show the most important messages, to summarize, to spot trends (dictionary or DOS attacks in progress), etc. (Example: `logwatch`.)

- **Dual Controls** These are mechanisms designed to prevent any single individual from violating a security policy. Examples include dual-entry bookkeeping, safe deposit boxes at banks, and military missile-launch protocols (require two persons to turn keys more than 10 feet apart at the same time).

  > [Reported in Ars Technica 7/29/10] Malware attacks were discovered on bank ATMs in Eastern Europe last year. Security researchers at Trustwave, based in Chicago, found the malware on 20 machines in Russia and Ukraine that were all running Microsoft’s Windows XP operating system. They said they found signs that hackers were planning on bringing their attacks to machines in the U.S. The malware was designed to attack ATMs made by Diebold and NCR.
  >
  > Those attacks required an insider, such as an ATM technician or anyone else with a key to the machine, to place the malware on the ATM. Once that was done, the attackers could insert a control card into the machine’s card reader to trigger the malware and give them control of the machine through a custom interface and the ATM’s keypad.
  >
  > The malware captured account numbers and PINs from the machine’s transaction application and then delivered it to the thief on a receipt printed from the machine in an encrypted format or to a storage device inserted in the card reader. A thief could also instruct the machine to

eject whatever cash was inside the machine. A fully loaded bank ATM can hold up to $600,000.

Earlier this year, in a separate incident, a Bank of America employee was charged with installing malware on his employer's ATMs that allowed him to withdraw thousands of dollars without leaving a transaction record.

What dual controls would you recommend to prevent this attack?

- **The meaning of *trust* and *Assurance*** ***Trust*** is a measure of *trustworthiness* of a system, relying on sufficient credible evidence that a system will meet a set of given requirements. For example, a system may be considered physically secure if the system is locked up and only authorized people have keys. However, this assumes that those with access to the system and who know how to pick locks (or copy keys), can be trusted not to do so unless authorized. Bank managers are authorize to move funds between accounts (up to some limit), but must be trusted not to move funds into their private accounts. You pay for some goods on amazon.com, but must trust them to actually ship them to you.

  Another example is that you must trust some installed server or other software not to allow violations of system security policy. If the server has exploits (e.g., bugs, back-doors, viruses) that cause the server to fail and break our security, our trust was misplaced.

  Trust relationships exists whenever there is some risk and uncertainty. One must weigh the risks of granting trust against the expected gains. Another way to define trust is "*a positive expectation regarding the behavior of somebody or something in a situation that entails risk to the trusting party*".

  Assurance is the measure of confidence that a system meets its security requirements, based upon specific evidence gathered through various assurance techniques. While *trust* can't be precisely measured or even defined, circumstantial and anecdotal evidence (*assurances*) can be accumulated that can be used to determine how much to trust a system (or determine your insurance rates!)

  Measures that can be taken to provide assurance in a trusted system include security cameras in high security areas to make sure no one is using lock-picks, extensive background checks before hiring a bank manager, following standard best practices, obtaining certifications, using vulnerability scanners, using referrals and recommendations for software, and tamper resistance. For example, don't trust a bank manager if they have stolen before, or buy goods on-line from a vendor with a poor reputation, or install software that has a history of problems (e.g., MS IIS) or is from an unknown source (e.g., acme email server).

Consider installing a strict (SELinux or other) MAC system. Now you don't need to trust software as much, because the (trusted) MAC system provides assurances that broken or malicious software won't compromise the security policy.

- **Design** The *design* of a system is the process of determining a set of components (mechanisms including procedures) and their interactions and configurations, that will be used to implement the security policy (specification). The design that results can be analyzed to determine if it will *satisfy* the specification. While it may be difficult or impossible to fully answer this question, assurance can be generated that the design will satisfy the security policy.

  > The design can also be considered a *security protocol*: exchange keys this way, encrypt data that way, validate the parties this way, etc. Quite often, the individual security mechanisms are secure, but their use in some protocol isn't. (This is what happened to the WEP 802.11 security protocol.)

  Once this is done, the design can be ***implemented***. The implementation must *satisfy* the design, much like the design must satisfy the specification. Proving an implementation satisfies some design is difficult and involves *proofs of correctness*.

- **Identity** is the representation of some unique entity or principal: a person, a website, an authorized meter-reader). Note a given entity may have many identities. (How many login names do you have?)

- **Privacy** is a complex notion, and not always a person's right — it depends on local laws and customs. Each person or entity (corporation or government agency) has their own unique point of view on what information is (or should be) under that person's or entity's control.

  Medical records are a case in point: Doctors don't like to release records to a patient, in case there is later any dispute. By U.S. law, some records must be released if a patient wants them. In addition, medical records can be used for research purposes and medical assurance purposes. In these cases, *identifying information* is supposed to be removed first, in a process known as ***anonymizing*** or ***sanitizing*** or ***blinding*** the data. But this is extremely difficult to do right, and often impossible when only a few records are involved. (The British Medical Association has the most comprehensive laws and rules on patient privacy.)

Safe Harbor Agreement

is an agreement between the United States and the European Union (EU) regarding the transfer of personally identifiable information (PII) from the EU to the United States, which is consistent with Fair Information Practices. Companies that register for Safe Harbor with the U.S. Department of Commerce and abide by the agreement are deemed by the EU to provide adequate data protection for **personally identifiable information** transferred from the EU to the United States.

Personal information may be defined to include a person's first and last name (or first initial and last name) combined with any one or more of the following: A Social Security number, a driver's license number or state-issued identification card, a financial (bank) account number or credit or debit card number, with or without any required security code, access code, personal identification number, or password.

Personal information does not include information that is lawfully obtained from publicly available information, or from federal, state, or local government records lawfully made available to the general public.

Identity theft is merely one problem. Credit reports are not only a pain over which most of us have little control, but the failure of the reporting companies may in fact cost people employment. An abusive ex-spouse, a company reviewing the movies you watch to determine your employment prospects, etc., are all legitimate reasons to desire privacy.

The European Union established new Internet policies in 2009, including a right to Internet access, net neutrality obligations, and strengthened consumer protections. Under the ePrivacy directive, communications service providers will also be required to notify consumers of security breaches, persistent identifiers ("cookies") will become opt-in, there will be enhanced penalties for spammers, and national data protection agencies will receive new enforcement powers. — *reported by EPIC 1/7/2010*

While **privacy laws exist to remove PII from publicly available information**, doing so is very difficult. [The rest of this section was adapted from a post made on Ars Technica by Nate Anderson, 9/8/2009.]

The Massachusetts Group Insurance Commission had a bright idea back in the mid-1990s—it decided to release "anonymized" data on state employees that showed every single hospital visit. The goal was to help researchers, and the state spent time removing all obvious identifiers such as name, address, and Social Security number. At the time GIC released the data, William Weld, then Governor of Massachusetts, assured the public that GIC had protected patient privacy by deleting identifiers. In response, then-graduate student Sweeney started hunting for the Governor's hospital records in the GIC data. She knew that Governor Weld resided in Cambridge, Massachusetts, a city of 54,000 residents and seven ZIP codes. For twenty dollars, she purchased the complete voter rolls from the city of Cambridge, a database containing, among other things, the name, address, ZIP code, birth date, and sex of every voter. By combining this data with the GIC records Sweeney found Governor Weld with ease. Only six people in Cambridge shared his birth date, only three of them men, and of them, only he lived in his ZIP code. Dr. Sweeney sent the Governor's health records (which included diagnoses and prescriptions) to his office.

In 2000, Sweeney showed that 87 percent of all Americans could be uniquely identified using only three pieces of information: ZIP code, birth date, and sex.

When AOL researchers released a massive dataset of search queries (2008?), they first "anonymized" the data by "scrubbing" out user IDs and IP addresses. When Netflix made a huge database of movie recommendations available for study it spent time doing the same thing. **Despite scrubbing the obviously identifiable information from the data, computer scientists were able to identify individual users in both datasets.**

In AOL's case, the problem was that user IDs were scrubbed but were replaced with a number that uniquely identified each user. But those complete lists of search queries were so thorough that individuals could be tracked down simply based on what they had searched for.

As Ohm notes, this illustrates a central reality of data collection: **"Data can either be useful or perfectly anonymous but never both."**

The Netflix case illustrates another principle, which is that the data itself might seem anonymous, but when paired with other existing data, reidentification becomes possible. A pair of computer scientists famously proved this point by combing movie recommendations posted on the Internet Movie Database with the Netflix data, and they learned that people could quite easily be picked from the Netflix data.

Just because of high profile commercial failures does not mean that anonymization is impossible. This is a heavily researched area in computer science and statistics right now (particularly in the database community; search for terms like "K-anonymity" and "L-diversity"). The census bureau generates publically available anonymous data, with no reported breaches of this data. However the data is of limited use because of the intensive data scrubbing.

[*From informit.com article posted on 1/24/10*] In 2007, Massachusetts joined 38 others states and enacted data breach notification laws. **Chapter 93H requires entities that own or license personal information of**

Massachusetts residents to publicly report the unauthorized acquisition or use of compromised data. However Massachusetts went much further; their regulations "Standards for the Protection of Personal Information of Residents of the Commonwealth" (Chapter 93H) are by far the most strident and far-reaching of any information security regulations of any state to date (2010).

Chapter 93H also mandates the adoption of detailed information security regulations for businesses in order to reduce the number of security breaches and thereby the need for data breach notifications. This establishes minimum standards by which a company is required to safeguard the integrity of personal information it handles. **The regulations apply to any business, whether or not operating in Massachusetts**, if such business owns, licenses, receives, maintains, processes or otherwise has access to personal information of Massachusetts residents. Some items include:

Companies must designate one or more employees to maintain and enforce a comprehensive information security program, and the details of the security program must be in writing. Companies must create security policies governing whether and how employees keep, access, and transport records containing personal information outside of business premises. Companies must also impose disciplinary measures for violations of its comprehensive information security program rules. Companies must impose reasonable restrictions upon physical access to records containing personal information. Monitory for compliance must be done, and all incidents must be documented. Companies must take reasonable steps to ensure that third-party service providers with access to personal information have the capacity to protect such information. There are many additional requirements too.

Recently (9/2011), California became the 45th state to enact strict data breach notification laws.

Pandora's Android app transmits personal information to third parties, at least according to an analysis done by security firm Veracode. The company decided to do a follow-up on the news that Pandora—among other mobile app makers—was being investigated by a federal grand jury, and found that data about the user's birth date, gender, Android ID, and GPS information were all being sent to various advertising companies. [Reported by Jacqui Cheng 4/7/2011 on Ars Technica.]

There are some steps you can take to help keep data anonymous:

- Never use any part of the DOB. Use the person's age, rounded to a whole number of years, and modify that by a random factor of +/- two years. The resulting data should still be statistically useful for most purposes.
- Never use zip code. Use the person's county or province. Better approach: use a quad system by breaking the state(s) into 4 or more arbitrary areas. Best approach: don't use a system which targets user locations.
- Never store a user's account numbers, even blinded ones.
- Read the reports on privacy that are available, such as those from the BMA, and use the advice.

- **Risk analysis** (Sometimes referred to as *Cost-Benefit analysis or CBA*, or *trade-off analysis or TOA*.) This means to determine whether some asset should be protected, and if so to what degree. You must balance the cost of an incident should it occur, the likelihood that it will occur, and the cost of preventive measures. You must also consider ***mitigation measures***, which are mechanisms and policies designed to lower the cost and/or likelihood of some type of security incidents.

Stephen Murdoch has published a dissertation that observing the behavior of users of "covert channels", especially anonymity systems, may be enough to discover their intentions or even their identity. The approach is similar to how card players in a game of bridge are able to determine cards by observing the behavior of other players. He adds that collusion between two partners can make the process easier. The strategy can be applied to TCP/IP environments and the simple traffic analysis of an "anonymous" network such as Tor. His findings are

> similar to those presented at the *Black Hat* conference August ’07. Murdoch says that anonymizing technologies might offer protection from casual scans or monitoring but they are unlikely to withstand the scrutiny of dedicated attackers, researchers, or law enforcement officials. He says “[There is] a wealth of practical experience in covert channel discovery that can be applied to find and exploit weaknesses in real-world anonymity systems”.

Risks change with time, and thus a new risk assessment should be made periodically.

- **Audit trails and Logging**   Audit trails provide ***Accountability***, which prevents false repudiation (“I didn’t do it!”).
- **Intrusion Detection**   Host IDSs (e.g. *file alteration/integrity monitor, or FAM/FIM)* and network IDSs seek to detect when an intruder has been attempting (or successful) at compromising integrity.

IPA — Identity, Policy, Audit

For efficiency, compliance, and risk mitigation, organizations need to centrally manage and correlate vital security information including:

- Identity (hosts, virtual machines, users, groups, authentication credentials)
- Policy (configuration settings, access control information)
- Audit (events, logs, analysis thereof)

Mid- and large-scale organizations today need to set up **centralized identity management**. **Policy** means a broad set of things, including access control policy, MAC policy, security configuration settings, which packages and patches are applied and running, and what system configuration settings are set. Organizations want to be able to centrally manage this information applying different policies based on machine group, location, user, and more. Finally, organizations need to be able to gather and analyze **log and audit data** and they need to meaningfully parse that data without getting overwhelmed.

The problem is all three of these aspects of security are inter-related. There are too many “ad-hoc” implementations of each aspect to manage and administer easily. Often custom shell scripts are needed to tie all the different systems together.

FreeIPA (FreeIPA.org) is an integrated security information management solution combining Linux (Fedora), LDAP (Fedora Directory Server), Kerberos, NTP, and DNS. It consists of a web interface and command-line administration tools. "IPA" stands for Identity, Policy and Audit, but as of 2010 only identity management is supported (think "single sign-on").

> Sometimes identity can be a bad thing. You can be tracked through the Internet if the web browser you use has plugins, add-ons, and settings that make it fairly unique. Additionally the IP address you got from your local ISP can be used to place you geographically.
>
> To see how unique your browser is visit https://panopticlick.eff.org/. (The more unique you are, the easier you can be tracked.) You can see what information is available to web servers (and ISPs) at browserspy.dk, and see all HTTP headers from http-header-viewer or web-sniffer.net.
>
> You can mitigate this with various brower add-ons, such as Secret Agent for Firefox.