Status of Security in Computing

Brian Woodward

STATUS OF SECURITY IN COMPUTING

STATUS OF SECURITY IN COMPUTING

Brian Woodward

BIBLIOTEX
Digital Library

Status of Security in Computing
by Brian Woodward

www.bibliotex.com

Ebook ISBN: 9781984664150

Published by:
Bibliotex
Canada
Website: www.bibliotex.com

Contents

1

Computer Security Design

Computer security is a branch of computer technology known as information security as applied to computers and networks. The objective of computer security includes protection of information and property from theft, corruption, or natural disaster, while allowing the information and property to remain accessible and productive to its intended users. The term computer system security means the collective processes and mechanisms by which sensitive and valuable information and services are protected from publication, tampering or collapse by unauthorized activities or untrustworthy individuals and unplanned events respectively. The strategies and methodologies of computer security often differ from most other computer technologies because of its somewhat elusive objective of preventing unwanted computer behaviour instead of enabling wanted computer behaviour.

SECURITY BY DESIGN

The technologies of computer security are based on logic. As security is not necessarily the primary goal of most computer applications, designing a programme with security in mind often imposes restrictions on that program's behaviour. There are 4 approaches to security in computing, sometimes a combination of approaches is valid:

1. Trust all the software to abide by a security policy but the software is not trustworthy (this is computer insecurity).
2. Trust all the software to abide by a security policy and the software is validated as trustworthy (by tedious branch and path analysis for example).
3. Trust no software but enforce a security policy with mechanisms that are not trustworthy (again this is computer insecurity).
4. Trust no software but enforce a security policy with trustworthy hardware mechanisms.

Many systems have unintentionally resulted in the first possibility. Since approach two is expensive and non-deterministic, its use is very limited. Approaches one and three lead to failure. Because approach number four is often based on hardware mechanisms and avoids abstractions and a multiplicity of degrees of freedom, it is more practical. Combinations of approaches two and four are often used in a layered architecture with thin layers of two and thick layers of four. There are various strategies and techniques used to design security systems. However there are few, if any, effective strategies to enhance security

after design. One technique enforces the principle of least privilege to great extent, where an entity has only the privileges that are needed for its function. That way even if an attacker gains access to one part of the system, fine-grained security ensures that it is just as difficult for them to access the rest. Furthermore, by breaking the system up into smaller components, the complexity of individual components is reduced, opening up the possibility of using techniques such as automated theorem proving to prove the correctness of crucial software subsystems. This enables a closed form solution to security that works well when only a single well-characterized property can be isolated as critical, and that property is also assessible to math. Not surprisingly, it is impractical for generalized correctness, which probably cannot even be defined, much less proven. Where formal correctness proofs are not possible, rigorous use of code review and unit testing represent a best-effort approach to make modules secure.

The design should use "defense in depth", where more than one subsystem needs to be violated to compromise the integrity of the system and the information it holds. Defense in depth works when the breaching of one security measure does not provide a platform to facilitate subverting another. Also, the cascading principle acknowledges that several low hurdles does not make a high hurdle. So cascading several weak mechanisms does not provide the safety of a single stronger mechanism. Subsystems should default to secure settings, and wherever possible should be designed to "fail secure" rather than "fail insecure". Ideally, a secure system should require a deliberate, conscious, knowledgeable and

free decision on the part of legitimate authorities in order to make it insecure. In addition, security should not be an all or nothing issue. The designers and operators of systems should assume that security breaches are inevitable. Full audit trails should be kept of system activity, so that when a security breach occurs, the mechanism and extent of the breach can be determined. Storing audit trails remotely, where they can only be appended to, can keep intruders from covering their tracks. Finally, full disclosure helps to ensure that when bugs are found the "window of vulnerability" is kept as short as possible.

SECURITY ARCHITECTURE

Security Architecture can be defined as the design artifacts that describe how the security controls (security countermeasures) are positioned, and how they relate to the overall information technology architecture. These controls serve the purpose to maintain the system's quality attributes, among them confidentiality, integrity, availability, accountability and assurance." Hardware mechanisms that protect computers and data Hardware based or assisted computer security offers an alternative to software-only computer security. Devices such as dongles may be considered more secure due to the physical access required in order to be compromised.

SECURE OPERATING SYSTEMS

One use of the term computer security refers to technology to implement a secure operating system. Much of this

technology is based on science developed in the 1980s and used to produce what may be some of the most impenetrable operating systems ever. Though still valid, the technology is in limited use today, primarily because it imposes some changes to system management and also because it is not widely understood. Such ultra-strong secure operating systems are based on operating system kernel technology that can guarantee that certain security policies are absolutely enforced in an operating environment. An example of such a Computer security policy is the Bell-LaPadula model. The strategy is based on a coupling of special microprocessor hardware features, often involving the memory management unit, to a special correctly implemented operating system kernel. This forms the foundation for a secure operating system which, if certain critical parts are designed and implemented correctly, can ensure the absolute impossibility of penetration by hostile elements. This capability is enabled because the configuration not only imposes a security policy, but in theory completely protects itself from corruption. Ordinary operating systems, on the other hand, lack the features that assure this maximal level of security. The design methodology to produce such secure systems is precise, deterministic and logical.

Systems designed with such methodology represent the state of the art of computer security although products using such security are not widely known. In sharp contrast to most kinds of software, they meet specifications with verifiable certainty comparable to specifications for size, weight and power. Secure operating systems designed this way are used primarily to protect national security

information, military secrets, and the data of international financial institutions. These are very powerful security tools and very few secure operating systems have been certified at the highest level (Orange Book A-1) to operate over the range of "Top Secret" to "unclassified" (including Honeywell SCOMP, USAF SACDIN, NSA Blacker and Boeing MLS LAN.) The assurance of security depends not only on the soundness of the design strategy, but also on the assurance of correctness of the implementation, and therefore there are degrees of security strength defined for COMPUSEC. The Common Criteria quantifies security strength of products in terms of two components, security functionality and assurance level (such as EAL levels), and these are specified in a Protection Profile for requirements and a Security Target for product descriptions. None of these ultra-high assurance secure general purpose operating systems have been produced for decades or certified under Common Criteria.

In USA parlance, the term High Assurance usually suggests the system has the right security functions that are implemented robustly enough to protect DoD and DoE classified information. Medium assurance suggests it can protect less valuable information, such as income tax information. Secure operating systems designed to meet medium robustness levels of security functionality and assurance have seen wider use within both government and commercial markets. Medium robust systems may provide the same security functions as high assurance secure operating systems but do so at a lower assurance level (such as Common Criteria levels EAL4 or EAL5). Lower

levels mean we can be less certain that the security functions are implemented flawlessly, and therefore less dependable. These systems are found in use on web servers, guards, database servers, and management hosts and are used not only to protect the data stored on these systems but also to provide a high level of protection for network connections and routing services.

SECURE CODING

If the operating environment is not based on a secure operating system capable of maintaining a domain for its own execution, and capable of protecting application code from malicious subversion, and capable of protecting the system from subverted code, then high degrees of security are understandably not possible. While such secure operating systems are possible and have been implemented, most commercial systems fall in a 'low security' category because they rely on features not supported by secure operating systems (like portability, et al.). In low security operating environments, applications must be relied on to participate in their own protection. There are 'best effort' secure coding practices that can be followed to make an application more resistant to malicious subversion.

In commercial environments, the majority of software subversion vulnerabilities result from a few known kinds of coding defects. Common software defects include buffer overflows, format string vulnerabilities, integer overflow, and code/command injection. It is to be immediately noted that all of the foregoing are specific instances of a general

class of attacks, where situations in which putative "data" actually contains implicit or explicit, executable instructions are cleverly exploited. Some common languages such as C and C++ are vulnerable to all of these defects (see Seacord, *"Secure Coding in C and C++"*). Other languages, such as Java, are more resistant to some of these defects, but are still prone to code/command injection and other software defects which facilitate subversion.

Recently another bad coding practice has come under scrutiny; dangling pointers. The first known exploit for this particular problem was presented in July 2007. Before this publication the problem was known but considered to be academic and not practically exploitable. Unfortunately, there is no theoretical model of "secure coding" practices, nor is one practically achievable, insofar as the variety of mechanisms are too wide and the manners in which they can be exploited are too variegated. It is interesting to note, however, that such vulnerabilities often arise from archaic philosophies in which computers were assumed to be narrowly disseminated entities used by a chosen few, all of whom were likely highly educated, solidly trained academics with naught but the goodness of mankind in mind. Thus, it was considered quite harmless if, for (fictitious) example, a FORMAT string in a FORTRAN programme could contain the J format specifier to mean "shut down system after printing." After all, who would use such a feature but a well-intentioned system programmer? It was simply beyond conception that software could be deployed in a destructive fashion. It is worth noting that, in some languages, the distinction between code (ideally, read-only) and data

(generally read/write) is blurred. In LISP, particularly, there is no distinction whatsoever between code and data, both taking the same form: an S-expression can be code, or data, or both, and the "user" of a LISP programme who manages to insert an executable LAMBDA segment into putative "data" can achieve arbitrarily general and dangerous functionality. Even something as "modern" as Perl offers the eval() function, which enables one to generate Perl code and submit it to the interpreter, disguised as string data.

CAPABILITIES AND ACCESS CONTROL LISTS

Within computer systems, two security models capable of enforcing privilege separation are access control lists (ACLs) and capability-based security. The semantics of ACLs have been proven to be insecure in many situations, e.g., the confused deputy problem. It has also been shown that the promise of ACLs of giving access to an object to only one person can never be guaranteed in practice. Both of these problems are resolved by capabilities. This does not mean practical flaws exist in all ACL-based systems, but only that the designers of certain utilities must take responsibility to ensure that they do not introduce flaws. Capabilities have been mostly restricted to research operating systems and commercial OSs still use ACLs. Capabilities can, however, also be implemented at the language level, leading to a style of programming that is essentially a refinement of standard object-oriented design. An open source project in the area is the E language. First the

Plessey System 250 and then Cambridge CAP computer demonstrated the use of capabilities, both in hardware and software, in the 1970s. A reason for the lack of adoption of capabilities may be that ACLs appeared to offer a 'quick fix' for security without pervasive redesign of the operating system and hardware. The most secure computers are those not connected to the Internet and shielded from any interference. In the real world, the most security comes from operating systems where security is not an add-on.

APPLICATIONS

Computer security is critical in almost any technology-driven industry which operates on computer systems. Computer security can also be referred to as computer safety. The issues of computer based systems and addressing their countless vulnerabilities are an integral part of maintaining an operational industry.

CLOUD COMPUTING SECURITY

Security in the cloud is challenging, due to varied degree of security features and management schemes within the cloud entitites. In this connection one logical protocol base need to evolve so that the entire gamet of components operates synchronously and securely.

IN AVIATION

The aviation industry is especially important when analyzing computer security because the involved risks include human life, expensive equipment, cargo, and

transportation infrastructure. Security can be compromised by hardware and software malpractice, human error, and faulty operating environments. Threats that exploit computer vulnerabilities can stem from sabotage, espionage, industrial competition, terrorist attack, mechanical malfunction, and human error. The consequences of a successful deliberate or inadvertent misuse of a computer system in the aviation industry range from loss of confidentiality to loss of system integrity, which may lead to more serious concerns such as data theft or loss, network and air traffic control outages, which in turn can lead to airport closures, loss of aircraft, loss of passenger life. Military systems that control munitions can pose an even greater risk. A proper attack does not need to be very high tech or well funded; for a power outage at an airport alone can cause repercussions worldwide. One of the easiest and, arguably, the most difficult to trace security vulnerabilities is achievable by transmitting unauthorized communications over specific radio frequencies.

These transmissions may spoof air traffic controllers or simply disrupt communications altogether. These incidents are very common, having altered flight courses of commercial aircraft and caused panic and confusion in the past. Controlling aircraft over oceans is especially dangerous because radar surveillance only extends 175 to 225 miles offshore. Beyond the radar's sight controllers must rely on periodic radio communications with a third party. Lightning, power fluctuations, surges, brown-outs, blown fuses, and various other power outages instantly disable all computer systems, since they are dependent on an electrical source.

Other accidental and intentional faults have caused significant disruption of safety critical systems throughout the last few decades and dependence on reliable communication and electrical power only jeopardizes computer safety.

Notable System Accidents

In 1994, over a hundred intrusions were made by unidentified crackers into the Rome Laboratory, the US Air Force's main command and research facility. Using trojan horse viruses, hackers were able to obtain unrestricted access to Rome's networking systems and remove traces of their activities. The intruders were able to obtain classified files, such as air tasking order systems data and furthermore able to penetrate connected networks of National Aeronautics and Space Administration's Goddard Space Flight Center, Wright-Patterson Air Force Base, some Defense contractors, and other private sector organizations, by posing as a trusted Rome center user.

Secure by Design

Secure by design, in software engineering, means that the software has been designed from the ground up to be secure. Malicious practices are taken for granted and care is taken to minimize impact when a security vulnerability is discovered or on invalid user input. Generally, designs that work well do not rely on being secret. It is not mandatory, but proper security usually means that everyone is allowed to know and understand the design *because it is secure*. This has the advantage that many people are looking at the

code, and this improves the odds that any flaws will be found sooner (Linus's law). Of course, attackers can also obtain the code, which makes it easier for them to find vulnerabilities as well. Also, it is very important that everything works with the least amount of privileges possible (principle of least privilege). For example a Web server that runs as the administrative user (root or admin) can have the privilege to remove files and users that do not belong to itself. Thus, a flaw in that programme could put the entire system at risk. On the other hand, a Web server that runs inside an isolated environment and only has the privileges for required network and filesystem functions, cannot compromise the system it runs on unless the security around it is in itself also flawed. A perfect authentication system for logins does not allow anyone to log in at all, because the user could be a threat to the system. However, some designs can never be perfect. Passwords, biometrics, and such are never perfect.

SECURITY BY DESIGN IN PRACTICE

Many things, especially input, should be distrusted by a secure design. A fault-tolerant programme could even distrust its own internals. Two examples of insecure design are allowing buffer overflows and format string vulnerabilities. The following C programme demonstrates these flaws:

```
int main()
{
char buffer[100];
printf("What is your name?\n");
gets(buffer);
```

```
printf("Hello, ");
printf(buffer);
printf("!\n");
return 0;
}
```

Because the gets function in the C standard library does not stop writing bytes into buffer until it reads a newline character or EOF, typing more than 99 characters at the prompt constitutes a buffer overflow. Allocating 100 characters for buffer with the assumption that almost any given name from a user is no longer than 99 characters doesn't prevent the user from actually *typing* more than 99 characters. This can lead to arbitrary machine code execution. The second flaw is that the programme tries to print its input by passing it directly to the printf function. This function prints out its first argument, replacing conversion specifications (such as "%s", "%d", et cetera) sequentially with other arguments from its call stack as needed.

Thus, if a malicious user entered "%d" instead of his name, the programme would attempt to print out a non-existent integer value, and undefined behaviour would occur. A related mistake in Web programming is for an online script not to validate its parameters. For example, consider a script that fetches an article by taking a filename, which is then read by the script and parsed. Such a script might use the following hypothetical URL to retrieve an article about dog food:

http://www.example.net/cgi-bin/article.sh?name=dogfood.html

If the script has no input checking, instead trusting that the filename is always valid, a malicious user could forge

a URL to retrieve configuration files from the webserver:

```
http://www.example.net/cgi-bin/article.sh?name=../../../../../
etc/passwd
```

Depending on the script, this may expose the /etc/passwd file, which on Unix-like systems contains (among others) user IDs, their login names, home directory paths and shells.

SERVER/CLIENT ARCHITECTURES

In server/client architectures, the programme at the other side may not be an authorised client and the client's server may not be an authorised server. Even when they are, a man-in-the-middle attack could compromise communications. Often the easiest way to break the security of a client/server system is not to go head on to the security mechanisms but instead to go around them. A man in the middle attack is a simple example of this, because you can use it to collect details to impersonate a user. Which is why it is important to consider encryption, hashing, and other security mechanisms in your design to ensure that information collected from a potential attacker won't allow access.

Another key feature to client-server security design is general good-coding practices. For example, following a known software design structure such as client and broker can help in designing a well built structure with a solid foundation. Further more that if the software is modified in the future it is even more important that it follows a logical foundation of separation between the client and

server. This is because if a programmer comes in and can not clearly understand the dynamics of the programme they may end up adding or changing something that can add a security flaw. Even with the best design this is always a possibility, but the better standardized the design the less chance there is of this occurring.

2

Computer Security Model

A computer security model is a scheme for specifying and enforcing security policies. A security model may be founded upon a formal model of access rights, a model of computation, a model of distributed computing, or no particular theoretical grounding at all. For a more complete list of available articles on specific security models, see Category:Computer security models.

CYBER SECURITY STANDARDS

Cyber security standards are security standards which enable organizations to practice safe security techniques to minimize the number of successful cyber security attacks. These guides provide general outlines as well as specific techniques for implementing cyber security.

For certain specific standards, cyber security certification by an accredited body can be obtained. There

are many advantages to obtaining certification including the ability to get cyber security insurance.

HISTORY

Cyber security standards have been created recently because sensitive information is now frequently stored on computers that are attached to the Internet. Also many tasks that were once done by hand are carried out by computer; therefore there is a need for Information Assurance (IA) and security. Cyber security is important in order to guard against identity theft. Businesses also have a need for cyber security because they need to protect their trade secrets, proprietary information, and personally identifiable information (PII) of their customers or employees.

The government also has the need to secure its information. One of the most widely used security standards today is ISO/IEC 27002 which started in 1995. This standard consists of two basic parts. BS 7799 part 1 and BS 7799 part 2 both of which were created by (British Standards Institute) BSI. Recently this standard has become ISO 27001. The National Institute of Standards and Technology (NIST) has released several special publications addressing cyber security. Three of these special papers are very relevant to cyber security: the 800-12 titled "Computer Security Handbook;" 800-14 titled "Generally Accepted Principles and Practices for Securing Information Technology;" and the 800-26 titled "Security Self-Assessment Guide for Information Technology Systems". The International Society of Automation (ISA) developed cyber security standards for

industrial automation control systems (IACS) that are broadly applicable across manufacturing industries. The series of ISA industrial cyber security standards are known as ISA-99 and are being expanded to address new areas of concern.

ISO 27002

ISO 27002 incorporates both parts of the BS 7799 standard. Sometimes ISO/IEC 27002 is referred to as BS 7799 part 1 and sometimes it refers to part 1 and part 2. BS 7799 part 1 provides an outline for cyber security policy; whereas BS 7799 part 2 provides a certification. The outline is a high level guide to cyber security. It is most beneficial for an organization to obtain a certification to be recognized as compliant with the standard.

The certification once obtained lasts three years and is periodically checked by the BSI to ensure an organization continues to be compliant throughout that three year period. ISO 27001 (ISMS) replaces BS 7799 part 2, but since it is backward compatible any organization working toward BS 7799 part 2 can easily transition to the ISO 27001 certification process.

There is also a transitional audit available to make it easier once an organization is BS 7799 part 2-certified for the organization to become ISO 27001-certified. ISO/IEC 27002 states that information security is characterized by integrity, confidentiality, and availability. The ISO/IEC 27002 standard is arranged into eleven control areas; security policy, organizing information security, asset management, human resources security, physical and environmental

security, communication and operations, access controls, information systems acquisition/development/maintenance, incident handling, business continuity management, compliance.

STANDARD OF GOOD PRACTICE

In the 1990s, the Information Security Forum (ISF) published a comprehensive list of best practices for information security, published as the *Standard of Good Practice* (SoGP). The ISF continues to update the SoGP every two years; the latest version was published in February 2007. Originally the *Standard of Good Practice* was a private document available only to ISF members, but the ISF has since made the full document available to the general public at no cost. Among other programmes, the ISF offers its member organizations a comprehensive benchmarking programme based on the SoGP.

NERC

The North American Electric Reliability Corporation (NERC) has created many standards. The most widely recognized is NERC 1300 which is a modification/update of NERC 1200. The newest version of NERC 1300 is called CIP-002-1 through CIP-009-2 (CIP=Critical Infrastructure Protection). These standards are used to secure bulk electric systems although NERC has created standards within other areas. The bulk electric system standards also provide network security administration while still supporting best practice industry processes.

NIST

1. Special publication 800-12 provides a broad overview of computer security and control areas. It also emphasizes the importance of the security controls and ways to implement them. Initially this document was aimed at the federal government although most practices in this document can be applied to the private sector as well. Specifically it was written for those people in the federal government responsible for handling sensitive systems.
2. Special publication 800-14 describes common security principles that are used. It provides a high level description of what should be incorporated within a computer security policy. It describes what can be done to improve existing security as well as how to develop a new security practice. Eight principles and fourteen practices are described within this document.
3. Special publication 800-26 provides advice on how to manage IT security. This document emphasizes the importance of self assessments as well as risk assessments.
4. Special publication 800-37, updated in 2010 provides a new risk approach: "Guide for Applying the Risk Management Framework to Federal Information Systems"
5. Special publication 800-53 rev3, "Guide for Assessing the Security Controls in Federal Information Systems", updated in August 2009, specifically addresses the 194 security controls that are applied to a system to make it "more secure."

ISO 15408

This standard develops what is called the "Common Criteria". It allows many different software applications to be integrated and tested in a secure way.

RFC 2196

RFC 2196 is memorandum published by Internet Engineering Task Force for developing security policies and procedures for information systems connected on the Internet.

The RFC 2196 provides a general and broad overview of information security including network security, incident response or security policies. The document is very practical and focusing on day-to-day operations.

ISA-99

ISA99 is the Industrial Automation and Control System Security Committee of the International Society for Automation (ISA). The committee is developing a multi-part series of standards and technical reports on the subject, several of which have been publicly released. Work products from the ISA99 committee are also submitted to IEC as standards and specifications in the IEC 63443 series.

- ISA-99.01.01 (formerly referred to as "Part 1") (ANSI/ ISA 99.00.01) is approved and published.
- ISA-TR99.01.02 is a master glossary of terms used by the committee. This document is still a working

draft but the content is available on the committee Wiki site (http://isa99.isa.org/ISA99%20Wiki/Master%20Glossary.aspx)

- ISA-99.01.03 identifies a set of compliance metrics for IACS security. This document is currently under development.
- ISA-99.02.01 (formerly referred to as "Part 2") (ANSI/ISA 99.02.01-2009) addresses how to establish an IACS security programme. This standard is approved and published. It has also been approved and published by the IEC as IEC 62443-2-1
- ISA-99.02.02 addresses how to operate an IACS security programme. This standard is currently under development.
- ISA-TR99.02.03 is a technical report on the subject of patch management. This report is currently under development.
- ISA-TR99.03.01 ()is a technical report on the subject of suitable technologies for IACS security. This report is approved and published.
- ISA-99.03.02 addresses how to define security assurance levels using the zones and conduits concept. This standard is currently under development.
- ISA-99.03.03 defines detailed technical requirements for IACS security. This standard is currently under development.
- ISA-99.03.04 addresses the requirements for the development of secure IACS products and solutions. This standard is currently under development.

- Standards in the ISA-99.04.xx series address detailed technical requirements at the component level. These standards are currently under development.

ISA SECURITY COMPLIANCE INSTITUTE

Related to the work of ISA 99 is the work of the ISA Security Compliance Institute. The ISA Security Compliance Institute (ISCI) has developed compliance test specifications for ISA99 and other control system security standards. They have also created an ANSI accredited certification programme called ISASecure for the certification of industrial automation devices such as programmable logic controllers (PLC), distributed control systems (DCS) and safety instrumented systems (SIS). These types of devices provided automated control of industrial processes such as those found in the oil & gas, chemical, electric utility, manufacturing, food & beverage and water/wastewater processing industries. There is growing concern from both governments as well as private industry regarding the risk that these systems could be intentionally compromised by "evildoers" such as hackers, disgruntled employees, organized criminals, terrorist organizations or even state-sponsored groups. The recent news about the industrial control system malware known as Stuxnet has heightened concerns about the vulnerability of these systems.

High Technology Crime Investigation Association

The High Technology Crime Investigation Association or as known by the abbreviation HTCIA Inc., Roseville,

California, is devoted to digital forensics for investigation of crimes. Members of HTCIA Inc. are made up of a professional body of investigators, prosecutors and security professionals. HTCIA is designed to promote, aid encourage and effect the voluntary interchange of data, experience, information, knowledge and ideas about processes, procedures, methods, and techniques relating to investigations and security in advanced technologies, and new technologies introduced into the field of forensic investigation for crime and the law. The HTCIA also promote uniformity in investigative methods, and develop matters of mutual interest It is also one of the most successful collaborative efforts between law enforcement and private industry working together. The HTCIA serve a common theme within each member state and internationally. This is to foster the growth in knowledge of investigation methods, processes and techniques amongst their Chapter's within their own national capital region in the US and Internationally. The HTCIA hold annual conferences. The HTCIA are primarily intended for Computer forensic analysts, Cybercrime Investigators, Mobile forensic analysts, IT Security, Security Managers, CIOs, Lawyers, Prosecutors, Police officers, Judiciary, and Incident Response specialists. Training opportunities do exist for law enforcement personnel, and investigative professionals. Which is seen today of paramount importance within the organization The HTCIA has local chapters that sponsor meetings. These meetings attract law enforcement as well as their public sector counterparts and academia. Generally, topics of current interest are presented, allowing members to obtain

valuable knowledge. One of the clearest benefits of becoming a member is the ability of members to network &meet other investigators in similar fields. The ability to contact others who have faced the same problems is invaluable. It is today one of the most respected organizations for professional, in-service training of law officials interested in computers and their role in criminal activity. Some high profile members include Howard Schmidt, Matthew Blake and James Lance.

HISTORY

The HTCIA had its beginnings in the early 1980s when security managers in Silicon Valley saw the need for law enforcement investigators to understood the importance of investigation with high technology and computer crime. In 1984, the Santa Carla District Attorney, Leo Himmelsbach was approached by members of the Santa Carla County Industrial Security Managers Group. Including security manager at Intel and later Sun Microsystems John Callaghan, and Pete Kostner security manager at AMD, they discussed the need for having law enforcement officers trained in the field of high technology crime. This was seen as quiet visionary for its time. Mr. Himmelsbach then applied to the state of California and received a grant from the Office of Criminal Justice Planning Project approved by the California State Assembly, State Assembly Bill 1078 passed into law August 31, 1984, Penal Code Section 13970 (GrntProjSummry) called " SANTA CLARA COUNTY DISTRICT ATTORNEY'S HIGH TECHNOLOGY CRIME PREVENTION PROGRAM" The HTCIA started as part of the 1985 OCJB

grant "Project Objectives and Activities. The main objective at that time was to train San Francisco Bay Area investigators and prosecutors in high-technology theft investigation. During that time, law enforcement officers from the Los Angeles area attended DATTA training and wanted to start their own organization in southern California. A second objective was then conceived "to establish a base that will provide the nucleus for the development of a regional high technology theft prevention effort"(OCJB Grant Contract Feb 1985).

In 1990, the District Attorney's Theft Technology Association of Santa Carla County affiliated with the other HTCIA chapters in existence in the Silicon Valley Chapter. It was then decided to allow private investigators to become members. From then on in the HTCIA began growing from strength to strength, with national seminars, setting standards for training and guidelines. The HTCIA now cover considerable areas throughout the US and Internationally across the World primarily focusing on High Technology and Crime.

HTCIA CORE VALUES

HTCIA Core Values are defined as follows: I) The HTCIA values the Truth uncovered effective techniques used to uncover that Truth, and within digital information and the so wrongful convictions are avoided! II) The HTCIA values the security of their society and its citizens through the enforcement of our laws and the protection of our infrastructure and economies. III) The HTCIA values the

ethical concept of its members and the evidence they expose through investigative procedures and computer forensic best practices including specialized techniques used to gather digital evidence. IV) The HTCIA values the trusted network of forensic and investigative professionals within private and public businesses including law enforcement who share our values and our vision. V) The HTCIA values the confidentiality of its membership and the information, skills and techniques they share within the association

MEMBERSHIP

I) Membership is open to prosecuting attorneys, and investigators engaged in the investigation of criminal activity associated with computers or technology. II) Senior security specialists, and managerial level professionals engaged in professions covering computers or advanced technology environments III) On October 2008, the international board of directors approved the bylaw provision creating the student membership. The purpose of this membership class is to promote and encourage the study of criminal investigations involving advanced technologies and security by the academic community. This would be composed of students studying in areas such as criminal justice, law enforcement computer science; forensics, corrections, accounting, a minimum Grade Point Average (GPA)is established by the International Executive Committee (IEC). The IEC will establish general application procedures and requirements for Student Members which are not in conflict with these bylaws. Scholarship for Service (SFS) provides scholarships that fully fund the typical costs that students pay for books,

tuition, and room and board while attending an approved institution of higher learning. The scholarships are funded through grants awarded by the National Science Foundation (NSF) HTCIA is not affiliated with the SFS.

CASE OF THE YEAR AWARD

Case of the year award The HTCIA also offer a case of the year award to recognize new technology or techniques which were expended to resolve the case. A synopsis of factors which must be used to evaluate the nominees are either the case was international, national, or regional in scope, it resolved a particularly violent offense & it established an important legal precedent

TRAINING EXAMPLE

Presentations may include discussing the benefits for using live computer forensic investigation techniques, and outline the situations where these techniques may be most appropriate; an example would be the ability to capture encryption passwords. Members or attendees would be introduced to the components of a live computer forensic investigation, shown tools for identifying the machine state to help mitigate the “trojan defense”

3

Internet Privacy

Internet privacy involves the desire or mandate of personal privacy concerning transactions or transmission of data via the Internet. It also involves the exercise of control over the type and amount of information revealed about a person on the Internet and who may access said information. Internet privacy forms a subset of computer privacy. A number of experts within the field of Internet security and privacy believe that privacy doesn't exist; "Privacy is dead – get over it" This should be more encouraged according to Steve Rambam, private investigator specializing in Internet privacy cases. In fact, it has been suggested that the "appeal of online services is to broadcast personal information on purpose." On the other hand, in his essay *The Value of Privacy,* security expert Bruce Schneier says, "Privacy protects us from abuses by those in power, even if we're doing nothing wrong at the time of surveillance."

LEVELS OF PRIVACY

People with only a casual concern for Internet privacy need not achieve total anonymity. Internet users may achieve an adequate level of privacy through controlled disclosure of personal information. The revelation of IP addresses, non-personally-identifiable profiling, and similar information might become acceptable trade-offs for the convenience that users could otherwise lose using the workarounds needed to suppress such details rigorously. On the other hand, some people desire much stronger privacy. In that case, they may try to achieve *Internet anonymity* to ensure privacy — use of the Internet without giving any third parties the ability to link the Internet activities to personally-identifiable information (P.I.I.) of the Internet user. In order to keep your information private, people need to be careful on what they submit and look at online. When filling out forms and buying merchandise, that becomes tracked and because your information was not private, companies are now sending you spam and advertising on similar products. The Sanders decision relied heavily on another California decision from a year ago. In Shulman v. Group W Productions, the court concluded that two people injured in a car accident could sue for invasion of privacy because a cameraman recorded emergency care given in a rescue helicopter. According to the court, while the accident victims could not claim a reasonable expectation of privacy at the accident scene (where they were recorded by the same cameraman), they could claim a reasonable expectation of privacy in the rescue helicopter, even if they expected that their conversations in the helicopter would be overheard.

Because these cases make it more difficult to determine under what circumstances undercover reporting would violate a reasonable expectation of privacy-thus exposing journalists to liability-news organizations may think twice about their approach to investigative reporting. In California at least, as a result of these recent decisions, trial judges will be reluctant to throw out cases before trial, allowing them to go before a jury. And because media lawyers are uncertain about whether jurors would think that a privacy invasion was justified by a legitimate need to gather news, they are likely to offer conservative advice and deter stations from engaging in certain investigations. Resulting, once again, in a chilling effect on the media. Related State Laws Privacy of Personal Information: Nevada and Minnesota require Internet Service Providers to keep information private regarding their customers. This is only unless a customer approves their information being given out. According to the National Conference of State Legislator, the following states have certain laws on the personal privacy of its citizens.

Minnesota Statutes §§ 325M.01 to .09 -Prohibits Internet service providers from disclosing personally identifiable information, including a consumer's physical or electronic address or telephone number; Internet or online sites visited; or any of the contents of a consumer's data storage devices. Provides for certain circumstances under which information must be disclosed, such as to a grand jury; to a state or federal law enforcement officer acting as authorized by law; pursuant to a court order or court action. Provides for civil damages of $500 or actual damages and attorney fees for violation of the law. Nevada Revised Statutes § 205.498 -

In addition, California and Utah laws, although not specifically targeted to on-line businesses, require all nonfinancial businesses to disclose to customers, in writing or by electronic mail, the types of personal information the business shares with or sells to a third party for direct marketing purposes or for compensation. Under the California law, businesses may post a privacy statement that gives customers the opportunity to choose not to share information at no cost. There are also certain laws for employees and businesses and privacy policies for websites. California, Connecticut, Nebraska and Pennsylvania all have specific privacy policies regarding websites, these include:

> "California (Calif. Bus. & Prof. Code §§ 22575-22578) California's Online Privacy Protection Act requires an operator, defined as a person or entity that collects personally identifiable information from California residents through an Internet Web site or online service for commercial purposes, to post conspicuously its privacy policy on its Web site or online service and to comply with that policy. The bill, among other things, would require that the privacy policy identify the categories of personally identifiable information that the operator collects about individual consumers who use or visit its Web site or online service and third parties with whom the operator may share the information.

Connecticut (Conn. Gen Stat. § 42-471) Requires any person who collects Social Security numbers in the course of business to create a privacy protection policy. The policy must be "publicly displayed" by posting on a web page and the policy must (1) protect the confidentiality of Social Security numbers, (2) prohibit unlawful disclosure of Social Security numbers, and (3) limit access to Social Security numbers. Nebraska (Nebraska Stat. § 87-302(14)) Nebraska prohibits knowingly making a false or misleading statement in a privacy policy, published on the Internet or otherwise

distributed or published, regarding the use of personal information submitted by members of the public. Pennsylvania (18 Pa. C.S.A. § 4107(a)(10)) Pennsylvania includes false and misleading statements in privacy policies published on Web sites or otherwise distributed in its deceptive or fraudulent business practices statute." There are also at least 16 states that require government websites to create privacy policies and procedures or to include machine-readable privacy policies into their websites. These states include Arizona, Arkansas, California, Colorado, Delaware, Iowa, Illinois, Maine, Maryland, Michigan, Minnesota, Montana, New York, Sourth Carolina, Texas, Utah, and Virginia.

RISKS TO INTERNET PRIVACY

In today's technological world, millions of individuals are subject to privacy threats. Companies are hired not only to watch what you visit online, but to infiltrate the information and send advertising based on your browsing history. People set up accounts for Facebook; enter bank and credit card information to various websites. Those concerned about Internet privacy often cite a number of *privacy risks* — events that can compromise privacy — which may be encountered through Internet use. These methods of compromise can range from the gathering of statistics on users, to more malicious acts such as the spreading of spyware and various forms of bugs (software errors) exploitation. Privacy measures are provided on several social networking sites to try to provide their users with protection

for their personal information. On Facebook for example privacy settings are available for all registered users. The settings available on Facebook include the ability to block certain individuals from seeing your profile, the ability to choose your "friends," and the ability to limit who has access to your pictures and videos. Privacy settings are also available on other social networking sites such as E-harmony and MySpace. It is the user's prerogative to apply such settings when providing personal information on the internet. In late 2007 Facebook launched the Beacon programme where user rental records were released on the public for friends to see. Many people were enraged by this breach in privacy, and the *Lane v. Facebook, Inc.* case ensued.

HTTP COOKIES

An HTTP cookie is data stored on a user's computer that assists in automated access to websites or web features, or other state information required in complex web sites. It may also be used for user-tracking by storing special usage history data in a cookie. Cookies are a common concern in the field of privacy. As a result, some types of cookies are classified as a *tracking cookie*. Although website developers most commonly use cookies for legitimate technical purposes, cases of abuse occur. In 2009, two researchers noted that social networking profiles could be connected to cookies, allowing the social networking profile to be connected to browsing habits. Systems do not generally make the user explicitly aware of the storing of a cookie. (Although some users object to that, it does not properly relate to Internet privacy. It does however have implications for computer

privacy, and specifically for computer forensics. The original developers of cookies intended that only the website that originally distributed cookies to users so they could retrieve them, therefore returning only data already possessed by the website. However, in practice programmers can circumvent this restriction. Possible consequences include:

- the placing of a personally-identifiable tag in a browser to facilitate web profiling, or,
- use of cross-site scripting or other techniques to steal information from a user's cookies.

Some users choose to disable cookies in their web browsers – as of 2000 a Pew survey estimated the proportion of users at 4%. Such an action eliminates the potential privacy risks, but may severely limit or prevent the functionality of many websites. All significant web browsers have this disabling ability built-in, with no external programme required. As an alternative, users may frequently delete any stored cookies. Some browsers (such as Mozilla Firefox and Opera) offer the option to clear cookies automatically whenever the user closes the browser. A third option involves allowing cookies in general, but preventing their abuse. There are also a host of wrapper applications that will redirect cookies and cache data to some other location. The process of *profiling* (also known as "tracking") assembles and analyzes several events, each attributable to a single originating entity, in order to gain information (especially patterns of activity) relating to the originating entity. Some organizations engage in the profiling of people's web browsing, collecting the URLs of sites visited. The

resulting profiles can potentially link with information that personally identifies the individual who did the browsing.

Some web-oriented marketing-research organizations may use this practice legitimately, for example: in order to construct profiles of 'typical Internet users'. Such profiles, which describe average trends of large groups of Internet users rather than of actual individuals, can then prove useful for market analysis. Although the aggregate data does not constitute a privacy violation, some people believe that the initial profiling does. Profiling becomes a more contentious privacy issue when data-matching associates the profile of an individual with personally-identifiable information of the individual. Governments and organizations may set up honeypot websites – featuring controversial topics – with the purpose of attracting and tracking unwary people. This constitutes a potential danger for individuals.

FLASH COOKIES

Flash cookies, also known as Local Shared Objects, work the same ways as normal cookies and are used by the Adobe Flash Player to store information at the user's computer. They exhibit a similar privacy risk as normal cookies, but are not as easily blocked, meaning that the option in most browsers to not accept cookies does not affect flash cookies. One way to view and control them is with browser extensions or add-ons.

EVERCOOKIES

An Evercookie is a JavaScript-based application which produces cookies in a web browser that actively "resist"

deletion by redundantly copying themselves in different forms on the user's machine (e.g.: Flash Local Shared Objects, various HTML5 storage mechanisms, window.name caching, etc.), and resurrecting copies are missing or expired.

PHOTOGRAPHS ON THE INTERNET

Today many people have digital cameras and post their photos online. The people depicted in these photos might not want to have them appear on the Internet. Some organizations attempt to respond to this privacy-related concern. For example, the 2005 Wikimania conference required that photographers have the prior permission of the people in their pictures. Some people wore a 'no photos' tag to indicate they would prefer not to have their photo taken. The Harvard Law Review published a short piece called "In The Face of Danger: Facial Recognition and Privacy Law," much of it explaining how "privacy law, in its current form, is of no help to those unwillingly tagged." Any individual can be unwillingly tagged in a photo and displayed in a manner that might violate them personally in some way, and by the time Facebook gets to taking down the photo, many people will have already had the chance to view, share, or distribute it. Furthermore, traditional tort law does not protect people who are captured by a photograph in public because this is not counted as an invasion of privacy.

The extensive Facebook privacy policy covers these concerns and much more. For example, the policy states that they reserve the right to disclose member information or share photos with companies, lawyers, courts, government

entities, etc. if they feel it absolutely necessary. The policy also informs users that profile pictures are mainly to help friends connect to each other.

However, these, as well as other pictures, can allow other people to invade a person's privacy by finding out information that can be used to track and locate a certain individual. In an article featured in ABC news, it was stated that two teams of scientists found out that Hollywood stars could be giving up information about their private whereabouts very easily through pictures uploaded to the Internet. Moreover, it was found that pictures taken by iPhones automatically attach the latitude and longitude of the picture taken through metadata unless this function is manually disabled.

SEARCH ENGINES

Search engines have the ability to track a user's searches. Personal information can be revealed through searches including search items used, the time of the search, and more. Search engines have claimed a necessity to retain such information in order to provide better services, protect against security pressure, and protect against fraud.

DATA LOGGING

Many programmes and operating systems are set up to perform data logging of usage. This may include recording times when the computer is in use, or which web sites are visited. If a third party has sufficient access to the computer, legitimately or not, the user's privacy may be compromised. This could be avoided by disabling logging, or by clearing logs regularly.

PRIVACY WITHIN SOCIAL NETWORKING SITES

Prior to the social networking site explosion over the past decade, there were early forms of social network technologies that included online multiplayer games, blog sites, news groups, mailings lists and dating services. These all created a backbone for the new modern sites, and even from the start of these older versions privacy was an issue. In 1996, a young woman in New York City was on a first date with an online acquaintance and later sued for sexual harassment as they went back to her apartment after when everything became too real. This is just an early example of many more issues to come regarding internet privacy. Social networking sites have become very popular within the last five years. With the creation of Facebook and the continued popularity of MySpace many people are giving their personal information out on the internet. These social networks keep track of all interactions used on their sites and save them for later use. Most users are not aware that they can modify the privacy settings and unless they modify them, their information is open to the public.

On Facebook privacy settings can be accessed via the drop down menu under account in the top right corner. There users can change who can view their profile and what information can be displayed on their profile. In most cases profiles are open to either “all my network and friends” or “all of my friends.” Also, information that shows on a user’s profile such as birthday, religious views, and relationship status can be removed via the privacy settings. If a user is under 13 years old they are not able to make a Facebook or a MySpace account, however, this is not regulated. Social

networking has redefined the role of Internet privacy. Since users are willingly disclosing personal information online, the role of privacy and security is somewhat blurry. Sites such as Facebook, Myspace, and Twitter have grown popular by broadcasting status updates featuring personal information such as location. Facebook "Places," in particular, is a Facebook service, which publicizes user location information to the networking community. Users are allowed to "check-in" at various locations including retail stores, convenience stores, and restaurants. Also, users are able to create their own "place," disclosing personal information onto the Internet. This form of location tracking is automated and must be turned off manually. Various settings must be turned off and manipulated in order for the user to ensure privacy.

According to epic.org, Facebook users are recommended to: (1) disable "Friends can check me in to Places," (2) customize "Places I Check In," (3) disable "People Here Now," and (4) uncheck "Places I've Visited.". Moreover, the Federal Trade Commission has received two complaints in regards to Facebook's "unfair and deceptive" trade practices, which are used to target advertising sectors of the online community. "Places" tracks user location information and is used primarily for advertising purposes. Each location tracked allows third party advertisers to customize advertisements that suit one's interests. Currently, the Federal Trade Commissioner along with the Electronic Privacy Information Center are shedding light on the issues of location data tracking on social networking sites. Recently, Facebook has been scrutinized for having a variety of

applications that are considered to be invasive to user privacy. "The Breakup Notifier" is an example of a Facebook "cyberstalking" app that has recently been taken down. Essentially, the application notifies users when a person breaks up with their partner through Facebook, allowing users to instantly become aware of their friend's romantic activities. The concept became very popular, with the site attracting 700,000 visits in the first 36 hours; people downloaded the app 40,000 times. Just days later, the app had more than 3.6 million downloads and 9,000 Facebook likes.

There are other applications that border on "cyberstalking." An application named "Creepy" can track a person's location on a map using photos uploaded to Twitter or Flickr. When a person uploads photos to a social networking site, others are able to track their most recent location. Some smart phones are able to embed the longitude and latitude coordinates into the photo and automatically send this information to the application. Anybody using the application can search for a specific person and then find their immediate location. This poses many potential threats to users who share their information with a large group of followers. Facebook recently updated its profile format allowing for people who are not "friends" of others to view personal information about other users, even when the profile is set to private. However, As of January 18, 2011 Facebook changed its decision to make home addresses and telephone numbers accessible to third party members, but it is still possible for third party members to have access to less exact personal information, like one's hometown and

employment, if the user has entered the information into Facebook. EPIC Executive Director Marc Rotenberg said "Facebook is trying to blur the line between public and private information.

And the request for permission does not make clear to the user why the information is needed or how it will be used." Similar to Rotenberg's claim that Facebook users are unclear of how or why their information has gone public, recently the Federal Trade Commission and Commerce Department have become involved. The Federal Trade Commission has recently released a report claiming that Internet companies and other industries will soon need to increase their protection for online users. Because online users often unknowingly opt in on making their information public, the FTC is urging Internet companies to make privacy notes simpler and easier for the public to understand, therefore increasing their option to opt out. Perhaps this new policy should also be implemented in the Facebook world. The Commerce Department claims that Americans, "have been ill-served by a patchwork of privacy laws that contain broad gaps,". Because of these broad gaps, Americans are more susceptible to identity theft and having their online activity tracked by others. Spokeo Spokeo is a "people-related" search engine with results compiled through data aggregation. The site contains information such as age, relationship status, estimated personal wealth, immediate family members and home address of individual people. This information is compiled through what is already on the internet or in other public records, but the website does not guarantee accuracy.

Spokeo has been faced with potential class action law suits from people who claim that the organization breaches the Fair Credit Reporting Act. In September, 2010, Jennifer Purcell claimed that the FCRA was violated by Spokeo marketing her personal information. Her case is pending in court. Also in 2010, Thomas Robins claimed that his personal information on the website was inaccurate and he was unable to edit it for accuracy. The case was dismissed because Robins did not claim that the site directly caused him actual harm. On February 15, 2011, Robins filed another suit, this time stating Spokeo has caused him "imminent and ongoing" harm. Twitter Case - In January 2011, the government recently obtained a court order to force the social networking site, Twitter, to reveal information applicable surrounding certain subscribers involved in the WikiLeaks cases. This outcome of this case is questionable because it deals with the user's First Amendment rights. Twitter moved to reverse the court order, and supported the idea that internet users should be notified and given an opportunity to defend their constitutional rights in court before their rights are compromised.

Facebook Friends Study - A study was conducted at Northeastern University by Alan Mislove and his colleagues at the Max Planck Institute for Software Systems, where an algorithm was created to try and discover personal attributes of a Facebook user by looking at their friend's list. They looked for information such as high school and college attended, major, hometown, graduation year and even what dorm a student may have lived in. The study revealed that only 5% of people thought to change their friend's list to

private. For other users, 58% displayed university attended, 42% revealed employers, 35% revealed interests and 19% gave viewers public access to where they were located. Due to the correlation of Facebook friends and universities they attend, it was easy to discover where a Facebook user was based on their list of friends. This fact is one that has become very useful to advertisers targeting their audiences but is also a big risk for the privacy of all those with Facebook accounts.

Law enforcement prowling the networks - The FBI has dedicated undercover agents on Facebook, Twitter, MySpace, LinkedIn. The rules and guidelines to the privacy issue is internal to the Justice Department and details aren't released to the public. Agents can impersonate a friend, a long lost relative, even a spouse and child. This raises real issues regarding privacy. Although people who use Facebook, Twitter, and other social networking sites are aware of some level of privacy will always be compromised, but, no one would ever suspect that the friend invitation might be from a federal agent whose sole purpose of the friend request was to snoop around. Furthermore, Facebook, Twitter, and MySpace have personal information and past posts logged for up to one year; even deleted profiles, and with a warrant, can hand over very personal information. One example of investigators using Facebook to nab a criminal is the case of Maxi Sopo. Charged with bank fraud, and having escaped to Mexico, he was nowhere to be found until he started posting on Facebook. Although his profile was private, his list of friends were not, and through this vector, they eventually caught him.

In recent years, some state and local law enforcement agencies have also begun to rely on social media websites as resources. Although obtaining records of information not shared publicly by or about site users often requires a subpoena, public pages on sites such as Facebook and MySpace offer access to personal information that can be valuable to law enforcement. Police departments have reported using social media websites to assist in investigations, locate and track suspects, and monitor gang activity. Teachers and MySpace - Teachers' privacy on MySpace has created controversy across the world. They are forewarned by The Ohio News Association that if they have a MySpace account, it should be deleted. Eschool News warns, "Teachers, watch what you post online." The ONA also posted a memo advising teachers not to join these sites. Teachers can face consequences of license revocations, suspensions, and written reprimands. The *Chronicle of Higher Education* wrote an article on April 27, 2007, entitled "A MySpace Photo Costs a Student a Teaching Certificate" about Stacy Snyder. She was a student of Millersville University of Pennsylvania who was denied her teaching degree because of an unprofessional photo posted on MySpace, which involved her drinking with a pirate's hat on and a caption of "Drunken Pirate". As a substitute, she was given an English degree. Internet privacy and Blizzard Entertainment - On July 6, 2010, Blizzard Entertainment announced that it would display the real names tied to user accounts in its game forums.

On July 9, 2010, CEO and cofounder of Blizzard Mike Morhaime announced a reversal of the decision to force

posters' real names to appear on Blizzard's forums. The reversal was made in response to subscriber feedback. Internet privacy and Google Maps - In Spring 2007, Google improved their Google Maps to include what is known as "Street View". This feature gives the user a 3-D, street level view with real photos of streets, buildings, and landmarks. In order to offer such a service, Google had to send trucks with cameras mounted on them and drive through every single street snapping photos. These photos were eventually stitched together to achieve a near seamless photorealistic map. However, the photos that were snapped included people caught in various acts, some of which includes a man urinating on the street, nude people seen through their windows, and apparently, a man trying to break into someone's apartment, etc; although some images are up to interpretation. This prompted a public outburst and sometime after, Google offered a "report inappropriate image" feature to their website. Internet privacy and Facebook advertisements The illegal activities on Facebook are very wild, especially "phishing attack" which is the most popular way of stealing other people's passwords.

The Facebook users are led to land on a page where they are asked for their login information, and their personal information is stolen in that way. According to the news from *PC World Business Center* which was published on April 22, 2010, we can know that a hacker named Kirllos illegally stole and sold 1.5 million Facebook IDs to some business companies who want to attract potential customers by using advertisements on the Facebook. Their illegal approach is that they used accounts which were bought

from hackers to send advertisements to friends of users. When friends see the advertisements, they will have opinion about them, because "People will follow it because they believe it was a friend that told them to go to this link," said Randy Abrams, director of technical education with security vendor Eset. There were 2.2232% of the population on Facebook that believed or followed the advertisements of their friends. Even though the percentage is small, the amount of overall users on Facebook is more than 400 million worldwide. The influence of advertisements on Facebook is so huge and obvious. According to the blog of Alan who just posted advertisement son the Facebook, he earned $300 over the 4 days. That means he can earn $3 for every $1 put into it. The huge profit attracts hackers to steal users' login information on Facebook, and business people who want to buy accounts from hackers send advertisements to users' friends on Facebook.

INTERNET SERVICE PROVIDERS

Internet users obtain Internet access through an Internet service provider (ISP). All data transmitted to and from users must pass through the ISP. Thus, an ISP has the potential to observe users' activities on the Internet. However, ISPs are usually prevented from participating in such activities due to legal, ethical, business, or technical reasons. Despite these legal and ethical restrictions, some ISPs, such as British Telecom (BT), are planning to use deep packet inspection technology provided by companies such as Phorm in order to examine the contents of the pages that people visit. By doing so, they can build up a profile of a person's

web surfing habits, which can then be sold on to advertisers in order to provide targeted advertising. BT's attempt at doing this will be marketed under the name 'Webwise'. Normally ISPs do collect at least *some* information about the consumers using their services. From a privacy standpoint, ISPs would ideally collect only as much information as they require in order to provide Internet connectivity (IP address, billing information if applicable, etc). Which information an ISP collects, what it does with that information, and whether it informs its consumers, pose significant privacy issues. Beyond the usage of collected information typical of third parties, ISPs sometimes state that they will make their information available to government authorities upon request. In the US and other countries, such a request does not necessarily require a warrant.

An ISP cannot know the contents of properly-encrypted data passing between its consumers and the Internet. For encrypting web traffic, https has become the most popular and best-supported standard. Even if users encrypt the data, the ISP still knows the IP addresses of the sender and of the recipient. (However, see the IP addresses section for workarounds.) An Anonymizer such as I2P – The Anonymous Network or Tor can be used for accessing web services without them knowing your IP address and without your ISP knowing what the services are that you access. General concerns regarding Internet user privacy have become enough of a concern for a UN agency to issue a report on the dangers of identity fraud. While signing up for internet services, each computer contains a unique IP, Internet Protocol address. This particular address will not give away

private or personal information, however, a weak link could potentially reveal information from your ISP. Social networking has redefined the role of Internet privacy. Since users are willingly disclosing personal information online, the role of privacy and security is somewhat blurry.

Sites such as Facebook, Myspace, and Twitter have grown popular by broadcasting status updates featuring personal information such as location. Facebook "Places," in particular, is a Facebook service, which publicizes user location information to the networking community. Users are allowed to "check-in" at various locations including retail stores, convenience stores, and restaurants. Also, users are able to create their own "place," disclosing personal information onto the Internet. This form of location tracking is automated and must be turned off manually. Various settings must be turned off and manipulated in order for the user to ensure privacy. According to epic.org, Facebook users are recommended to: (1) disable "Friends can check me in to Places," (2) customize "Places I Check In," (3) disable "People Here Now," and (4) uncheck "Places I've Visited.". Moreover, the Federal Trade Commission has received two complaints in regards to Facebook's "unfair and deceptive" trade practices, which are used to target advertising sectors of the online community. "Places" tracks user location information and is used primarily for advertising purposes. Each location tracked allows third party advertisers to customize advertisements that suit one's interests. Currently, the Federal Trade Commissioner along with the Electronic Privacy Information Center are shedding light on the issues of location data tracking on social networking sites.

LEGAL THREATS

Use by government agencies of an array of technologies designed to track and gather Internet users' information are the topic of much debate between privacy advocates, civil libertarians and those who believe such measures are necessary for law enforcement to keep pace with rapidly changing communications technology.

SPECIFIC EXAMPLES

- Following a decision by the European Union's council of ministers in Brussels, in January, 2009, the UK's Home Office adopted a plan to allow police to access the contents of individuals' computers without a warrant. The process, called "remote searching", allows one party, at a remote location, to examine another's hard drive and Internet traffic, including email, browsing history and websites visited. Police across the EU are now permitted to request that the British police conduct a remote search on their behalf. The search can be granted, and the material gleaned turned over and used as evidence, on the basis of a senior officer believing it necessary to prevent a serious crime. Opposition MPs and civil libertarians are concerned about this move toward widening surveillance and its possible impact on personal privacy. Says Shami Chakrabarti, director of the human rights group Liberty, "The public will want this to be controlled by new legislation and judicial authorisation. Without those safeguards it's a devastating blow to any notion of personal privacy."

- The FBI's Magic Lantern software programme was the topic of much debate when it was publicized in November, 2001. Magic Lantern is a Trojan Horse programme that logs users' keystrokes, rendering encryption useless.

LAWS FOR INTERNET PRIVACY PROTECTION

USA Patriot Act

The purpose of this act, enacted on October 26, 2001 by former President Bush, was to enhance law enforcement investigatory tools, investigate online activity, as well as to discourage terrorist acts both within the United States and around the world.

This act reduced restrictions for law enforcement to search various methods and tools of communication such as telephone, e-mail, personal records including medical and financial, as well as reducing restrictions with obtaining of foreign intelligence.

Electronic Communications Privacy Act (ECPA)

This act makes it unlawful under certain conditions for an individual to reveal the information of electronic communication and contains a few exceptions. One clause allows the ISP to view private e-mail if the sender is suspected of attempting to damage the internet system or attempting to harm another user. Another clause allows the ISP to reveal information from a message if the sender or recipient

allows to its disclosure. Finally, information containing personal information may also be revealed for a court order or law enforcement's subpoena.

Employees and Employers Internet Regulations

When considering the rights between employees and employers regarding internet privacy and protection at a company, different states have their own laws. Connecticut and Delaware both have laws that state an employer must create a written notice or electronic message that provides understanding that they will regulate the internet traffic. By doing so, this relates to the employees that the employer will be searching and monitoring emails and internet usage. Delaware charges $100 for a violation where Connecticut charges $500 for the first violation and then $1000 for the second. When looking at public employees and employers, California and Colorado created laws that would also create legal ways in which employers controlled internet usage. The law stated that a public company or agency must create a prior message to the employees stating that accounts will be monitored. Without these laws, employers could access information through employees accounts and use them illegally. In most cases, the employer is allowed to see whatever he or she pleases because of these laws stated both publicly and privately.

OTHER POTENTIAL INTERNET PRIVACY RISKS

- Malware is a term short for "malicious software" and is used to describe software to cause damage to a

single computer, server, or computer network whether that is through the use of a virus, trojan horse, spyware, etc.

- Spyware is a piece of software that obtains information from a user's computer without that user's consent.
- A web bug is an object embedded into a web page or email and is usually invisible to the user of the website or reader of the email. It allows checking to see if a person has looked at a particular website or read a specific email message.
- Phishing is a criminally fraudulent process of trying to obtain sensitive information such as user names, passwords, credit card or bank information. Phishing is an internet crime in which someone masquerades as a trustworthy entity in some form of electronic communication.
- Pharming is hackers attempt to redirect traffic from a legitimate website to a completely different internet address. Pharming can be conducted by changing the hosts file on a victim's computer or by exploiting a vulnerability on the DNS server.
- Social engineering
- Malicious proxy server (or other "anonymity" services)

SPECIFIC CASES

JASON FORTUNY AND CRAIGSLIST

In early September 2006, Jason Fortuny, a Seattle-area freelance graphic designer and network administrator, posed

as a woman and posted an ad to Craigslist Seattle seeking a casual sexual encounter with men in that area. On September 4, he posted to the wiki website Encyclopædia Dramatica all 178 of the responses, complete with photographs and personal contact details, describing this as the Craigslist Experiment and encouraging others to further identify the respondents Although some online exposures of personal information have been seen as justified for exposing malfeasance, many commentators on the Fortuny case saw no such justification here. "The men who replied to Fortuny's posting did not appear to be doing anything illegal, so the outing has no social value other than to prove that someone could ruin lives online," said law professor Jonathan Zittrain, while *Wired* writer Ryan Singel described Fortuny as "sociopathic". The Electronic Frontier Foundation indicated that it thought Fortuny might be liable under Washington state law, and that this would depend on whether the information he disclosed was of legitimate public concern. Kurt Opsahl, the EFF's staff attorney, said "As far as I know, they (the respondents) are not public figures, so it would be challenging to show that this was something of public concern."

According to Fortuny, two people lost their jobs as a result of his Craigslist Experiment and another "has filed an invasion-of-privacy lawsuit against Fortuny in an Illinois court." Fortuny did not enter an appearance in the Illinois suit, secure counsel, or answer the complaint after an early amendment. Mr. Fortuny had filed a motion to dismiss, but he filed it with the Circuit Court of Cook County, Illinois, and he did not file proof that he had served the plaintiff.

As a result, the court entered a default judgment against Mr. Fortuny and ordered a damages hearing for January 7, 2009. After failing to show up at multiple hearings on damages, Fortuny was ordered to pay $74,252.56 for violation of the Copyright Act, compensation for Public Disclosure of Private Facts, Intrusion Upon Seclusion, attorneys fees and costs.

USA vs. Warshak

The case *United States v. Warshak*, decided December 14, 2010 by the Sixth Circuit Court of Appeals, maintained the idea that an ISP actually is allowed access to private e-mail. However, the government must get hold of a search warrant before obtaining such e-mail. This case dealt with the question of emails hosted on an isolated server. Due to the fact that e-mail is similar to other forms of communication such as telephone calls, e-mail requires the same amount of protection under the 4th amendment.

SEARCH ENGINE DATA AND LAW ENFORCEMENT

Data from major Internet companies, including Yahoo! and MSN (Microsoft), have already been subpoenaed by the United States and China. AOL even provided a chunk of its own search data online, allowing reporters to track the online behaviour of private individuals. In 2006, a wireless hacker pled guilty when his Google searches were used as evidence against him. The defendant ran a Google search over the network using the following search terms: “how to broadcast interference over wifi 2.4 GHZ,” “interference over wifi 2.4 Ghz,” “wireless networks 2.4 interference,” and

"make device interfere wireless network." While court papers did not describe how the FBI obtained his searches (e.g. through a seized hard-drive or directly from the search-engine), Google has indicated that it can provide search terms to law enforcement if given an Internet address or Web cookie.

US V. ZEIGLER

In the United States many cases discuss whether a private employee (i.e., not a government employee) who stores incriminating evidence in workplace computers is protected by the Fourth Amendment's reasonable expectation of privacy standard in a criminal proceeding. Most case law holds that employees do not have a reasonable expectation of privacy when it comes to their work related electronic communications. See, e.g. *US v. Simons*, 206 F.3d 392, 398 (4th Cir., Feb. 28, 2000). However, one federal court held that employees can assert that the attorney-client privilege with respect to certain communications on company laptops. See *Curto v. Medical World Comm.*, No. 03CV6327, 2006 U.S. Dist. LEXIS 29387 (E.D.N.Y. May 15, 2006). Another recent federal case discussed this topic. On January 30, 2007, the Ninth Circuit court in US v. Ziegler, reversed its earlier August 2006 decision upon a petition for rehearing. In contrast to the earlier decision, the Court acknowledged that an employee has a right to privacy in his workplace computer. However, the Court also found that an employer can consent to any illegal searches and seizures. See *US v. Ziegler*, ___F.3d 1077 (9th Cir. Jan. 30, 2007, No. 05-30177). Cf. US v. Ziegler, 456 F.3d 1138 (9th Cir. 2006).

In Ziegler, an employee had accessed child pornography websites from his workplace. His employer noticed his activities, made copies of the hard drive, and gave the FBI the employee's computer. At his criminal trial, Ziegler filed a motion to suppress the evidence because he argued that the government violated his Fourth Amendment rights. The Ninth Circuit allowed the lower court to admit the child pornography as evidence. After reviewing relevant Supreme Court opinions on a reasonable expectation of privacy, the Court acknowledged that Ziegler had a reasonable expectation of privacy at his office and on his computer. That Court also found that his employer could consent to a government search of the computer and that, therefore, the search did not violate Ziegler's Fourth Amendment rights.

STATE V. REID

The New Jersey Supreme Court has also issued an opinion on the privacy rights of computer users, holding in *State v. Reid* that computer users have a reasonable expectation of privacy concerning the personal information they give to their ISPs. In that case, Shirley Reid was indicted for computer theft for changing her employer's password and shipping address on its online account with a supplier. The police discovered her identity after serving the ISP, Comcast, with a municipal subpoena not tied to any judicial proceeding. The lower court suppressed the information from Comcast that linked Reid with the crime on grounds that the disclosure violated Reid's constitutional right to be protected from unreasonable search and seizure. The

appellate court affirmed, as did the New Jersey Supreme Court, which ruled that ISP subscriber records can only be disclosed to law enforcement upon the issuance of a grand jury subpoena. As a result, New Jersey offers greater privacy rights to computer users than most federal courts. This case also serves as an illustration of how case law on privacy regarding workplace computers is still evolving.

ROBBINS V. LOWER MERION SCHOOL DISTRICT

In *Robbins v. Lower Merion School District* (U.S. Eastern District of Pennsylvania 2010), the federal trial court issued an injunction against the school district after plaintiffs charged two suburban Philadelphia high schools violated the privacy of students and others when they secretly spied on students by surreptitiously and remotely activating webcams embedded in school-issued laptops the students were using at home. The schools admitted to secretly snapping over 66,000 webshots and screenshots, including webcam shots of students in their bedrooms.

4

Capability-Based Security

Capability-based security is a concept in the design of secure computing systems, one of the existing security models. A capability (known in some systems as a key) is a communicable, unforgeable token of authority. It refers to a value that references an object along with an associated set of access rights. A user programme on a capability-based operating system must use a capability to access an object. Capability-based security refers to the principle of designing user programmes such that they directly share capabilities with each other according to the principle of least privilege, and to the operating system infrastructure necessary to make such transactions efficient and secure. Although most operating systems implement a facility which resembles capabilities, they typically do not provide enough support to allow for the exchange of capabilities among possibly mutually untrusting entities to be the primary

means of granting and distributing access rights throughout the system. A capability-based system, in contrast, is designed with that goal in mind. Capabilities as discussed in this article should not be confused with POSIX 1e/2c "Capabilities". The latter are coarse-grained privileges that cannot be transferred between processes.

CAPABILITIES AND CAPABILITY-BASED SECURITY

Capabilities achieve their objective of improving system security by being used in place of forgeable references. A forgeable reference (for example, a path name) identifies an object, but does not specify which access rights are appropriate for that object and the user programme which holds that reference. Consequently, any attempt to access the referenced object must be validated by the operating system, typically via the use of an access control list (ACL). Instead, in a system with capabilities, the mere fact that a user programme possesses that capability entitles it to use the referenced object in accordance with the rights that are specified by that capability. In theory, a system with capabilities removes the need for any access control list or similar mechanism by giving all entities all and only the capabilities they will actually need.

A capability is typically implemented as a privileged data structure that consists of a section that specifies access rights, and a section that uniquely identifies the object to be accessed. In practice, it is used much like a file descriptor in a traditional operating system, but to access every object

on the system. Capabilities are typically stored by the operating system in a list, with some mechanism in place to prevent the programme from directly modifying the contents of the capability (so as to forge access rights or change the object it points to). Some systems have also been based on capability-based addressing (hardware support for capabilities), such as Plessey System 250. Programmes possessing capabilities can perform functions on them, such as passing them on to other programmes, converting them to a less-privileged version, or deleting them. The operating system must ensure that only specific operations can occur to the capabilities in the system, in order to maintain the integrity of the security policy.

INTRODUCTION TO CAPABILITY-BASED SECURITY

(The following introduction assumes some basic knowledge of Unix systems.) A capability is defined to be a protected object reference which, by virtue of its possession by a user process, grants that process the capability (hence the name) to interact with an object in certain ways. Those ways might include reading data associated with an object, modifying the object, executing the data in the object as a process, and other conceivable access rights. The capability logically consists of a reference that uniquely identifies a particular object and a set of one or more of these rights. Suppose that, in a user process's memory space, there exists the following string:

```
/etc/passwd
```

Although this identifies a unique object on the system, it does not specify access rights and hence is not a capability. Suppose there is instead the following two values:

```
/etc/passwd
O_RDWR
```

This identifies an object along with a set of access rights. It, however, is still not a capability because the user process's *possession* of these values says nothing about whether that access would actually be legitimate. Now suppose that the user programme successfully executes the following statement:

```
int fd = open("/etc/passwd", O_RDWR);
```

The variable fd now contains the index of a file descriptor in the process's file descriptor table. This file descriptor *is* a capability. Its existence in the process's file descriptor table is sufficient to know that the process does indeed have legitimate access to the object. A key feature of this arrangement is that the file descriptor table is in kernel memory and cannot be directly manipulated by the user programme.

SHARING OF CAPABILITIES BETWEEN PROCESSES

In traditional operating systems, programmes often communicate with each other and with storage using references like those in the first two examples. Path names are often passed as command-line parameters, sent via sockets, and stored on disk. These references are not capabilities, and must be validated before they can be used.

In these systems, a central question is "on whose *authority* is a given reference to be evaluated?" This becomes a critical issue especially for processes which must act on behalf of two different authority-bearing entities. They become susceptible to a programming error known as the confused deputy problem, very frequently resulting in a security hole. In a capability-based system, the capabilities themselves are passed between processes and storage using a mechanism that is known by the operating system to maintain the integrity of those capabilities.

Although many operating systems implement facilities very similar to capabilities through the use of file descriptors or file handles — for example, in UNIX, file descriptors can be discarded (closed), inherited by child processes, and even sent to other processes via sockets — there are several obstacles that prevent all of the benefits of a capability-based addressing system from being realized in a traditional operating system environment. Chief among these obstacles is the fact that entities which might hold capabilities (such as processes and files) cannot be made persistent in such a way that maintains the integrity of the secure information that a capability represents. The operating system cannot trust a user programme to read back a capability and not tamper with the object reference or the access rights, and has no built-in facilities to control such tampering. Consequently, when a programme wishes to regain access to an object that is referenced on disk, the operating system must have some way of validating that access request, and an access control list or similar mechanism is mandated.

One novel approach to solving this problem involves the

use of an orthogonally persistent operating system. (This was realised in the Flex machine. See Ten15). In such a system, there is no need for entities to be discarded and their capabilities be invalidated, and hence require an ACL-like mechanism to restore those capabilities at a later time. The operating system maintains the integrity and security of the capabilities contained within all storage, both volatile and nonvolatile, at all times; in part by performing all serialization tasks by itself, rather than requiring user programmes to do so, as is the case in most operating systems. Because user programmes are relieved of this responsibility, there is no need to trust them to reproduce only legal capabilities, nor to validate requests for access using an access control mechanism.

POSIX CAPABILITIES

POSIX draft 1003.1e specifies a concept of permissions called “capabilities”. However POSIX capabilities differ from capabilities in this article — POSIX capability is not associated with any object — a process having CAP_NET_BIND_SERVICE capability can listen on any TCP port under 1024.

RESEARCH AND COMMERCIAL SYSTEMS

- Tahoe-LAFS - Open Source capability-based filesystem
- KeyKOS
  - o EROS - The Extremely Reliable Operating System - KeyKOS successor

  - CapROS - EROS successor, project to further develop EROS code base for commercial use
  - Coyotos - EROS successor, for research
- kaneton
- Cambridge CAP computer
- Carnegie Mellon University C.mmp with Hydra (operating system)
- Carnegie Mellon University CM* with StarOS
- IBM System/38 and AS/400
- Intel iAPX 432
- Plessey System 250
- Symbian
- Flex
- L4 microkernel - Open Kernel Labs - OKL4 and NICTA - seL4, TU-Dresden - Fiasco.OC
- Amoeba distributed operating system

Cloud Computing Security

Cloud computing security (sometimes referred to simply as "cloud security") is an evolving sub-domain of computer security, network security, and, more broadly, information security. It refers to a broad set of policies, technologies, and controls deployed to protect data, applications, and the associated infrastructure of cloud computing.

SECURITY ISSUES ASSOCIATED WITH THE CLOUD

There are a number of security issues/concerns associated with cloud computing but these issues fall into

two broad categories: Security issues faced by cloud providers (organizations providing Software-, Platform-, or Infrastructure-as-a-Service via the cloud) and security issues faced by their customers. In most cases, the provider must ensure that their infrastructure is secure and that their clients' data and applications are protected while the customer must ensure that the provider has taken the proper security measures to protect their information.

DIMENSIONS OF CLOUD SECURITY

While cloud security concerns can be grouped into any number of dimensions (Gartner names seven while the Cloud Security Alliance identifies fifteen areas of concern) these dimensions have been aggregated into three general areas: Security and Privacy, Compliance, and Legal or Contractual Issues.

SECURITY AND PRIVACY

In order to ensure that data is secure (that it cannot be accessed by unauthorized users or simply lost) and that data privacy is maintained, cloud providers attend to the following areas:

DATA PROTECTION

To be considered protected, data from one customer must be properly segregated from that of another; it must be stored securely when "at rest" and it must be able to move securely from one location to another. Cloud providers have systems in place to prevent data leaks or access by

third parties. Proper separation of duties should ensure that auditing and/or monitoring cannot be defeated, even by privileged users at the cloud provider.

IDENTITY MANAGEMENT

Every enterprise will have its own identity management system to control access to information and computing resources. Cloud providers either integrate the customer's identity management system into their own infrastructure, using federation or SSO technology, or provide an identity management solution of their own.

PHYSICAL AND PERSONNEL SECURITY

Providers ensure that physical machines are adequately secure and that access to these machines as well as all relevant customer data is not only restricted but that access is documented.

AVAILABILITY

Cloud providers assure customers that they will have regular and predictable access to their data and applications.

APPLICATION SECURITY

Cloud providers ensure that applications available as a service via the cloud are secure by implementing testing and acceptance procedures for outsourced or packaged application code.

It also requires application security measures (application-level firewalls) be in place in the production environment.

PRIVACY

Finally, providers ensure that all critical data (credit card numbers, for example) are masked and that only authorized users have access to data in its entirety. Moreover, digital identities and credentials must be protected as should any data that the provider collects or produces about customer activity in the cloud.

COMPLIANCE

Numerous regulations pertain to the storage and use of data, including Payment Card Industry Data Security Standard (PCI DSS), the Health Insurance Portability and Accountability Act (HIPAA), the Sarbanes-Oxley Act, among others. Many of these regulations require regular reporting and audit trails. Cloud providers must enable their customers to comply appropriately with these regulations.

BUSINESS CONTINUITY AND DATA RECOVERY

Cloud providers have business continuity and data recovery plans in place to ensure that service can be maintained in case of a disaster or an emergency and that any data lost will be recovered. These plans are shared with and reviewed by their customers.

LOGS AND AUDIT TRAILS

In addition to producing logs and audit trails, cloud providers work with their customers to ensure that these logs and audit trails are properly secured, maintained for as long as the customer requires, and are accessible for the purposes of forensic investigation (e.g., eDiscovery).

UNIQUE COMPLIANCE REQUIREMENTS

In addition to the requirements to which customers are subject, the data centers maintained by cloud providers may also be subject to compliance requirements.

LEGAL AND CONTRACTUAL ISSUES

Aside from the security and compliance issues enumerated above, cloud providers and their customers will negotiate terms around liability (stipulating how incidents involving data loss or compromise will be resolved, for example), intellectual property, and end-of-service (when data and applications are ultimately returned to the customer

PUBLIC RECORDS

Legal issues may also include records-keeping requirements in the public sector, where many agencies are required by law to retain and make available electronic records in a specific fashion. This may be determined by legislation, or law may require agencies to conform to the rules and practices set by a records-keeping agency. Public agencies using cloud computing and storage must take these concerns into account.

5

Separation of Protection and Security

In computer sciences the separation of protection and security is a design choice. Wulf et al. identified protection as a mechanism and security as a policy, therefore making the protection-security distinction a particular case of the separation of mechanism and policy principle.

The adoption of this distinction in a computer architecture, usually means that protection is provided as a fault tolerance mechanism by hardware/firmware and kernel, whereas the operating system and applications implement their security policies. In this design, security policies rely therefore on the protection mechanisms and on additional cryptography techniques. The major hardware approach for security or protection is the use of hierarchical protection domains. Prominent example of this approach is ring architecture with “supervisor mode” and “user mode”).

Such approach adopts a policy already at the lower levels (hardware/firmware/kernel), restricting the rest of the system to rely on it. Therefore, the choice to distinguish between protection and security in the overall architecture design implies rejection of the hierarchical approach in favour of another one, the capability-based addressing.

DESIGN MODELS WITH THE SEPARATION

The models with the protection and security separation are: access matrix, UCLA Data Secure Unix, take-grant and filter.

DESIGN MODELS WITHOUT THE SEPARATION

The models without such separation are: high-water mark, Bell–LaPadula (original and revisited), information flow, strong dependency and constraints.

Computer Insecurity

Many current computer systems have only limited security precautions in place. This computer insecurity article describes the current battlefield of computer security exploits and defenses. Please see the computer security article for an alternative approach, based on security engineering principles.

SECURITY AND SYSTEMS DESIGN

Many current real-world computer security efforts focus on external threats, and generally treat the computer system

itself as a trusted system. Some knowledgeable observers consider this to be a disastrous mistake, and point out that this distinction is the cause of much of the insecurity of current computer systems — once an attacker has subverted one part of a system without fine-grained security, he or she usually has access to most or all of the features of that system. Because computer systems can be very complex, and cannot be guaranteed to be free of defects, this security stance tends to produce insecure systems.

FINANCIAL COST

Serious financial damage has been caused by computer security breaches, but reliably estimating costs is quite difficult. Figures in the billions of dollars have been quoted in relation to the damage caused by malware such as computer worms like the Code Red worm, but such estimates may be exaggerated. However, other losses, such as those caused by the compromise of credit card information, can be more easily determined, and they have been substantial, as measured by millions of individual victims of identity theft each year in each of several nations, and the severe hardship imposed on each victim, that can wipe out all of their finances, prevent them from getting a job, plus be treated as if *they* were the criminal. Volumes of victims of phishing and other scams may not be known. Individuals who have been infected with spyware or malware likely go through a costly and time-consuming process of having their computer cleaned. Spyware is considered to be a problem specific to the various Microsoft Windows operating systems, however this can be partially explained by the fact

that Microsoft controls a major share of the PC market and thus represents the most prominent target.

REASONS

There are many similarities (yet many fundamental differences) between computer and physical security. Just like real-world security, the motivations for breaches of computer security vary between attackers, sometimes called hackers or crackers. Some are thrill-seekers or vandals (the kind often responsible for defacing web sites); similarly, some web site defacements are done to make political statements. However, some attackers are highly skilled and motivated with the goal of compromising computers for financial gain or espionage. An example of the latter is Markus Hess (more diligent than skilled), who spied for the KGB and was ultimately caught because of the efforts of Clifford Stoll, who wrote a memoir, The Cuckoo's Egg, about his experiences. For those seeking to prevent security breaches, the first step is usually to attempt to identify what might motivate an attack on the system, how much the continued operation and information security of the system are worth, and who might be motivated to breach it. The precautions required for a home PC are very different for those of banks' Internet banking system, and different again for a classified military network. Other computer security writers suggest that, since an attacker using a network need know nothing about you or what you have on your computer, attacker motivation is inherently impossible to determine beyond guessing. If true, blocking all possible attacks is the only plausible action to take.

VULNERABILITIES

To understand the techniques for securing a computer system, it is important to first understand the various types of "attacks" that can be made against it. These threats can typically be classified into one of these seven categories:

EXPLOITS

An exploit (from the same word in the French language, meaning "achievement", or "accomplishment") is a piece of software, a chunk of data, or sequence of commands that take advantage of a software 'bug' or 'glitch' in order to cause unintended or unanticipated behaviour to occur on computer software, hardware, or something electronic (usually computerized). This frequently includes such things as gaining control of a computer system or allowing privilege escalation or a denial of service attack. Many development methodologies rely on testing to ensure the quality of any code released; this process often fails to discover unusual potential exploits.

The term "exploit" generally refers to small programmes designed to take advantage of a software flaw that has been discovered, either remote or local. The code from the exploit programme is frequently reused in trojan horses and computer viruses.

In some cases, a vulnerability can lie in certain programmes' processing of a specific file type, such as a non-executable media file. Some security web sites maintain lists of currently known unpatched vulnerabilities found in common programmes.

EAVESDROPPING

Eavesdropping is the act of surreptitiously listening to a private conversation, typically between hosts on a network. Even machines that operate as a closed system (i.e., with no contact to the outside world) can be eavesdropped upon via monitoring the faint electro-magnetic transmissions generated by the hardware such as TEMPEST. The FBI's proposed Carnivore programme was intended to act as a system of eavesdropping protocols built into the systems of internet service providers.

SOCIAL ENGINEERING AND HUMAN ERROR

A computer system is no more secure than the human systems responsible for its operation. Malicious individuals have regularly penetrated well-designed, secure computer systems by taking advantage of the carelessness of trusted individuals, or by deliberately deceiving them, for example sending messages that they are the system administrator and asking for passwords. This deception is known as Social engineering.

DENIAL-OF-SERVICE ATTACK

Unlike other exploits, denial of service attacks are not used to gain unauthorized access or control of a system. They are instead designed to render it unusable. Attackers can deny service to individual victims, such as by deliberately entering a wrong password 3 consecutive times and thus causing the victim account to be locked, or they may overload the capabilities of a machine or network and block all users at once. These types of attack are, in practice, very hard

to prevent, because the behaviour of whole networks needs to be analyzed, not only the behaviour of small pieces of code. Distributed denial of service (DDoS) attacks are common, where a large number of compromised hosts (commonly referred to as "zombie computers", used as part of a botnet with, for example; a worm, trojan horse, or backdoor exploit to control them.) are used to flood a target system with network requests, thus attempting to render it unusable through resource exhaustion. Another technique to exhaust victim resources is through the use of an attack amplifier — where the attacker takes advantage of poorly designed protocols on 3rd party machines, such as FTP or DNS, in order to instruct these hosts to launch the flood. There are also commonly found vulnerabilities in applications that cannot be used to take control over a computer, but merely make the target application malfunction or crash. This is known as a denial-of-service exploit.

INDIRECT ATTACKS

An indirect attack is an attack launched by a third party computer. By using someone else's computer to launch an attack, it becomes far more difficult to track down the actual attacker. There have also been cases where attackers took advantage of public anonymizing systems, such as the tor onion router system.

BACKDOORS

A backdoor in a computer system (or cryptosystem or algorithm) is a method of bypassing normal authentication, securing remote access to a computer, obtaining access to

plaintext, and so on, while attempting to remain undetected. The backdoor may take the form of an installed programme (e.g., Back Orifice), or could be a modification to an existing programme or hardware device. A specific form of backdoors are rootkits, which replaces system binaries and/or hooks into the function calls of the operating system to hide the presence of other programmes, users, services and open ports. It may also fake information about disk and memory usage. The threat of backdoors surfaced when multiuser and networked operating systems became widely adopted. Petersen and Turn discussed computer subversion in a paper published in the proceedings of the 1967 AFIPS Conference. They noted a class of active infiltration attacks that use "trapdoor" entry points into the system to bypass security facilities and permit direct access to data. The use of the word trapdoor here clearly coincides with more recent definitions of a backdoor. However, since the advent of public key cryptography the term trapdoor has acquired a different meaning. More generally, such security breaches were discussed at length in a RAND Corporation task force report published under ARPA sponsorship by J.P. Anderson and D.J. Edwards in 1970.

A backdoor in a login system might take the form of a hard coded user and password combination which gives access to the system. A famous example of this sort of backdoor was as a plot device in the 1983 film WarGames, in which the architect of the "WOPR" computer system had inserted a hardcoded password (his dead son's name) which gave the user access to the system, and to undocumented parts of the system (in particular, a video game–like

simulation mode and direct interaction with the artificial intelligence). An attempt to plant a backdoor in the Linux kernel, exposed in November 2003, showed how subtle such a code change can be.

In this case, a two-line change appeared to be a typographical error, but actually gave the caller to the sys_wait4 function root access to the system.

DIRECT ACCESS ATTACKS

Someone who has gained access to a computer can install any type of devices to compromise security, including operating system modifications, software worms, key loggers, and covert listening devices. The attacker can also easily download large quantities of data onto backup media, for instance CD-R/DVD-R, tape; or portable devices such as keydrives, digital cameras or digital audio players. Another common technique is to boot an operating system contained on a CD-ROM or other bootable media and read the data from the harddrive(s) this way. The only way to defeat this is to encrypt the storage media and store the key separate from the system.

REDUCING VULNERABILITIES

Computer code is regarded by some as a form of mathematics. It is theoretically possible to prove the correctness of certain classes of computer programmes, though the feasibility of actually achieving this in large-scale practical systems is regarded as small by some with practical experience in the industry — see Bruce Schneier et al. It's also possible to protect messages in transit (i.e.,

communications) by means of cryptography. One method of encryption — the one-time pad — is unbreakable when correctly used.

This method was used by the Soviet Union during the Cold War, though flaws in their implementation allowed some cryptanalysis. The method uses a matching pair of key-codes, securely distributed, which are used once-and-only-once to encode and decode a single message. For transmitted computer encryption this method is difficult to use properly (securely), and highly inconvenient as well. Other methods of encryption, while breakable in theory, are often virtually impossible to directly break by any means publicly known today. Breaking them requires some non-cryptographic input, such as a stolen key, stolen plaintext (at either end of the transmission), or some other extra cryptanalytic information.

Social engineering and direct computer access (physical) attacks can only be prevented by non-computer means, which can be difficult to enforce, relative to the sensitivity of the information. Even in a highly disciplined environment, such as in military organizations, social engineering attacks can still be difficult to foresee and prevent. In practice, only a small fraction of computer programme code is mathematically proven, or even goes through comprehensive information technology audits or inexpensive but extremely valuable computer security audits, so it's usually possible for a determined hacker to read, copy, alter or destroy data in well secured computers, albeit at the cost of great time and resources. Few attackers would audit applications for vulnerabilities just to attack a single specific system. It is possible to reduce an attacker's chances by keeping systems

up to date, using a security scanner or/and hiring competent people responsible for security. The effects of data loss/ damage can be reduced by careful backing up and insurance.

SECURITY MEASURES

A state of computer "security" is the conceptual ideal, attained by the use of the three processes:

1. Prevention
2. Detection
3. Response

- User account access controls and cryptography can protect systems files and data, respectively.
- Firewalls are by far the most common prevention systems from a network security perspective as they can (if properly configured) shield access to internal network services, and block certain kinds of attacks through packet filtering.
- Intrusion Detection Systems (IDS's) are designed to detect network attacks in progress and assist in post-attack forensics, while audit trails and logs serve a similar function for individual systems.
- "Response" is necessarily defined by the assessed security requirements of an individual system and may cover the range from simple upgrade of protections to notification of legal authorities, counter-attacks, and the like. In some special cases, a complete destruction of the compromised system is favoured, as it may happen that not all the compromised resources are detected.

Today, computer security comprises mainly "preventive" measures, like firewalls or an Exit Procedure. A firewall can be defined as a way of filtering network data between a host or a network and another network, such as the Internet, and can be implemented as software running on the machine, hooking into the network stack (or, in the case of most UNIX-based operating systems such as Linux, built into the operating system kernel) to provide realtime filtering and blocking. Another implementation is a so called physical firewall which consists of a separate machine filtering network traffic.

Firewalls are common amongst machines that are permanently connected to the Internet. However, relatively few organizations maintain computer systems with effective detection systems, and fewer still have organized response mechanisms in place.

DIFFICULTY WITH RESPONSE

Responding forcefully to attempted security breaches (in the manner that one would for attempted physical security breaches) is often very difficult for a variety of reasons:

- Identifying attackers is difficult, as they are often in a different jurisdiction to the systems they attempt to breach, and operate through proxies, temporary anonymous dial-up accounts, wireless connections, and other anonymising procedures which make backtracing difficult and are often located in yet another jurisdiction. If they successfully breach security, they are often able to delete logs to cover their tracks.

- The sheer number of attempted attacks is so large that organizations cannot spend time pursuing each attacker (a typical home user with a permanent (e.g., cable modem) connection will be attacked at least several times per day, so more attractive targets could be presumed to see many more). Note however, that most of the sheer bulk of these attacks are made by automated vulnerability scanners and computer worms.
- Law enforcement officers are often unfamiliar with information technology, and so lack the skills and interest in pursuing attackers. There are also budgetary constraints. It has been argued that the high cost of technology, such as DNA testing, and improved forensics mean less money for other kinds of law enforcement, so the overall rate of criminals not getting dealt with goes up as the cost of the technology increases. In addition, the identification of attackers across a network may require logs from various points in the network and in many countries, the release of these records to law enforcement (with the exception of being voluntarily surrendered by a network administrator or a system administrator) requires a search warrant and, depending on the circumstances, the legal proceedings required can be drawn out to the point where the records are either regularly destroyed, or the information is no longer relevant.

6

Information Security

Information security means protecting information and information systems from unauthorized access, use, disclosure, disruption, modification, perusal, inspection, recording or destruction. The terms information security, computer security and information assurance are frequently incorrectly used interchangeably. These fields are interrelated often and share the common goals of protecting the confidentiality, integrity and availability of information; however, there are some subtle differences between them. These differences lie primarily in the approach to the subject, the methodologies used, and the areas of concentration. Information security is concerned with the confidentiality, integrity and availability of data regardless of the form the data may take: electronic, print, or other forms. Computer security can focus on ensuring the availability and correct operation of a computer system without concern for the information stored or processed by the computer.

Governments, military, corporations, financial institutions, hospitals, and private businesses amass a great deal of confidential information about their employees, customers, products, research, and financial status. Most of this information is now collected, processed and stored on electronic computers and transmitted across networks to other computers. Should confidential information about a business' customers or finances or new product line fall into the hands of a competitor, such a breach of security could lead to lost business, law suits or even bankruptcy of the business. Protecting confidential information is a business requirement, and in many cases also an ethical and legal requirement. For the individual, information security has a significant effect on privacy, which is viewed very differently in different cultures. The field of information security has grown and evolved significantly in recent years. There are many ways of gaining entry into the field as a career. It offers many areas for specialization including: securing network(s) and allied infrastructure, securing applications and databases, security testing, information systems auditing, business continuity planning and digital forensics science, etc. This article presents a general overview of information security and its core concepts.

HISTORY

Since the early days of writing, heads of state and military commanders understood that it was necessary to provide some mechanism to protect the confidentiality of written correspondence and to have some means of detecting tampering. Julius Caesar is credited with the invention of

the Caesar cipher ca. 50 B.C., which was created in order to prevent his secret messages from being read should a message fall into the wrong hands. World War II brought about many advancements in information security and marked the beginning of the professional field of information security. The end of the 20th century and early years of the 21st century saw rapid advancements in telecommunications, computing hardware and software, and data encryption. The availability of smaller, more powerful and less expensive computing equipment made electronic data processing within the reach of small business and the home user. These computers quickly became interconnected through a network generically called the Internet or World Wide Web. The rapid growth and widespread use of electronic data processing and electronic business conducted through the Internet, along with numerous occurrences of international terrorism, fueled the need for better methods of protecting the computers and the information they store, process and transmit. The academic disciplines of computer security, information security and information assurance emerged along with numerous professional organizations – all sharing the common goals of ensuring the security and reliability of information systems.

BASIC PRINCIPLES

KEY CONCEPTS

For over twenty years, information security has held confidentiality, integrity and availability (known as the CIA triad) to be the core principles of information security.

There is continuous debate about extending this classic trio. Other principles such as Accountability have sometimes been proposed for addition – it has been pointed out that issues such as Non-Repudiation do not fit well within the three core concepts, and as regulation of computer systems has increased (particularly amongst the Western nations) Legality is becoming a key consideration for practical security installations. In 2002, Donn Parker proposed an alternative model for the classic CIA triad that he called the six atomic elements of information. The elements are confidentiality, possession, integrity, authenticity, availability, and utility. The merits of the Parkerian hexad are a subject of debate amongst security professionals.

Confidentiality

Confidentiality is the term used to prevent the disclosure of information to unauthorized individuals or systems. For example, a credit card transaction on the Internet requires the credit card number to be transmitted from the buyer to the merchant and from the merchant to a transaction processing network. The system attempts to enforce confidentiality by encrypting the card number during transmission, by limiting the places where it might appear (in databases, log files, backups, printed receipts, and so on), and by restricting access to the places where it is stored. If an unauthorized party obtains the card number in any way, a breach of confidentiality has occurred. Breaches of confidentiality take many forms. Permitting someone to look over your shoulder at your computer screen while you have confidential data displayed on it could be a breach of

confidentiality. If a laptop computer containing sensitive information about a company's employees is stolen or sold, it could result in a breach of confidentiality. Giving out confidential information over the telephone is a breach of confidentiality if the caller is not authorized to have the information. Confidentiality is necessary (but not sufficient) for maintaining the privacy of the people whose personal information a system holds.

Integrity

In information security, integrity means that data cannot be modified undetectably. This is not the same thing as referential integrity in databases, although it can be viewed as a special case of Consistency as understood in the classic ACID model of transaction processing. Integrity is violated when a message is actively modified in transit. Information security systems typically provide message integrity in addition to data confidentiality.

Availability

For any information system to serve its purpose, the information must be available when it is needed. This means that the computing systems used to store and process the information, the security controls used to protect it, and the communication channels used to access it must be functioning correctly. High availability systems aim to remain available at all times, preventing service disruptions due to power outages, hardware failures, and system upgrades. Ensuring availability also involves preventing denial-of-service attacks.

Authenticity

In computing, e-Business and information security it is necessary to ensure that the data, transactions, communications or documents (electronic or physical) are genuine. It is also important for authenticity to validate that both parties involved are who they claim they are.

Non-repudiation

In law, non-repudiation implies one's intention to fulfill their obligations to a contract. It also implies that one party of a transaction cannot deny having received a transaction nor can the other party deny having sent a transaction. Electronic commerce uses technology such as digital signatures and encryption to establish authenticity and non-repudiation.

RISK MANAGEMENT

A comprehensive treatment of the topic of risk management is beyond the scope of this article. However, a useful definition of risk management will be provided as well as some basic terminology and a commonly used process for risk management. The CISA Review Manual 2006 provides the following definition of risk management: *"Risk management is the process of identifying vulnerabilities and threats to the information resources used by an organization in achieving business objectives, and deciding what countermeasures, if any, to take in reducing risk to an acceptable level, based on the value of the information resource to the organization."* There are two things in this definition that may need some clarification. First, the *process* of risk

management is an ongoing iterative process. It must be repeated indefinitely. The business environment is constantly changing and new threats and vulnerability emerge every day. Second, the choice of countermeasure (computer)s (controls) used to manage risks must strike a balance between productivity, cost, effectiveness of the countermeasure, and the value of the informational asset being protected.

Risk is the likelihood that something bad will happen that causes harm to an informational asset (or the loss of the asset). A vulnerability is a weakness that could be used to endanger or cause harm to an informational asset. A threat is anything (man made or act of nature) that has the potential to cause harm. The likelihood that a threat will use a vulnerability to cause harm creates a risk. When a threat does use a vulnerability to inflict harm, it has an impact. In the context of information security, the impact is a loss of availability, integrity, and confidentiality, and possibly other losses (lost income, loss of life, loss of real property). It should be pointed out that it is not possible to identify all risks, nor is it possible to eliminate all risk. The remaining risk is called *residual risk*. A risk assessment is carried out by a team of people who have knowledge of specific areas of the business. Membership of the team may vary over time as different parts of the business are assessed. The assessment may use a subjective qualitative analysis based on informed opinion, or where reliable dollar figures and historical information is available, the analysis may use quantitative analysis. The research has shown that the most vulnerable point in most information systems is the

human user, operator, designer, or other human The ISO/IEC 27002:2005 Code of practice for information security management recommends the following be examined during a risk assessment:

- security policy,
- organization of information security,
- asset management,
- human resources security,
- physical and environmental security,
- communications and operations management,
- access control,
- information systems acquisition, development and maintenance,
- information security incident management,
- business continuity management, and
- regulatory compliance.

In broad terms, the risk management process consists of:

1. Identification of assets and estimating their value. Include: people, buildings, hardware, software, data (electronic, print, other), supplies.
2. Conduct a threat assessment. Include: Acts of nature, acts of war, accidents, malicious acts originating from inside or outside the organization.
3. Conduct a vulnerability assessment, and for each vulnerability, calculate the probability that it will be exploited. Evaluate policies, procedures, standards, training, physical security, quality control, technical security.

4. Calculate the impact that each threat would have on each asset. Use qualitative analysis or quantitative analysis.
5. Identify, select and implement appropriate controls. Provide a proportional response. Consider productivity, cost effectiveness, and value of the asset.
6. Evaluate the effectiveness of the control measures. Ensure the controls provide the required cost effective protection without discernible loss of productivity.

For any given risk, Executive Management can choose to accept the risk based upon the relative low value of the asset, the relative low frequency of occurrence, and the relative low impact on the business. Or, leadership may choose to mitigate the risk by selecting and implementing appropriate control measures to reduce the risk. In some cases, the risk can be transferred to another business by buying insurance or out-sourcing to another business. The reality of some risks may be disputed. In such cases leadership may choose to deny the risk. This is itself a potential risk.

CONTROLS

When Management chooses to mitigate a risk, they will do so by implementing one or more of three different types of controls.

Administrative

Administrative controls (also called procedural controls) consist of approved written policies, procedures, standards and guidelines. Administrative controls form the framework

for running the business and managing people. They inform people on how the business is to be run and how day to day operations are to be conducted. Laws and regulations created by government bodies are also a type of administrative control because they inform the business. Some industry sectors have policies, procedures, standards and guidelines that must be followed – the Payment Card Industry (PCI) Data Security Standard required by Visa and Master Card is such an example. Other examples of administrative controls include the corporate security policy, password policy, hiring policies, and disciplinary policies. Administrative controls form the basis for the selection and implementation of logical and physical controls. Logical and physical controls are manifestations of administrative controls. Administrative controls are of paramount importance.

Logical

Logical controls (also called technical controls) use software and data to monitor and control access to information and computing systems. For example: passwords, network and host based firewalls, network intrusion detection systems, access control lists, and data encryption are logical controls. An important logical control that is frequently overlooked is the principle of least privilege. The principle of least privilege requires that an individual, programme or system process is not granted any more access privileges than are necessary to perform the task. A blatant example of the failure to adhere to the principle of least privilege is logging into Windows as user

Administrator to read Email and surf the Web. Violations of this principle can also occur when an individual collects additional access privileges over time. This happens when employees' job duties change, or they are promoted to a new position, or they transfer to another department. The access privileges required by their new duties are frequently added onto their already existing access privileges which may no longer be necessary or appropriate.

Physical

Physical controls monitor and control the environment of the work place and computing facilities. They also monitor and control access to and from such facilities. For example: doors, locks, heating and air conditioning, smoke and fire alarms, fire suppression systems, cameras, barricades, fencing, security guards, cable locks, etc. Separating the network and work place into functional areas are also physical controls. An important physical control that is frequently overlooked is the separation of duties. Separation of duties ensures that an individual can not complete a critical task by himself. For example: an employee who submits a request for reimbursement should not also be able to authorize payment or print the check. An applications programmer should not also be the server administrator or the database administrator – these roles and responsibilities must be separated from one another.

DEFENSE IN DEPTH

Information security must protect information throughout the life span of the information, from the initial creation of the information on through to the final disposal of the

information. The information must be protected while in motion and while at rest. During its lifetime, information may pass through many different information processing systems and through many different parts of information processing systems. There are many different ways the information and information systems can be threatened. To fully protect the information during its lifetime, each component of the information processing system must have its own protection mechanisms. The building up, layering on and overlapping of security measures is called defense in depth. The strength of any system is no greater than its weakest link. Using a defence in depth strategy, should one defensive measure fail there are other defensive measures in place that continue to provide protection. Recall the earlier discussion about administrative controls, logical controls, and physical controls. The three types of controls can be used to form the basis upon which to build a defense-in-depth strategy. With this approach, defense-in-depth can be conceptualized as three distinct layers or planes laid one on top of the other. Additional insight into defense-in- depth can be gained by thinking of it as forming the layers of an onion, with data at the core of the onion, people the next outer layer of the onion, and network security, host-based security and application security forming the outermost layers of the onion. Both perspectives are equally valid and each provides valuable insight into the implementation of a good defense-in-depth strategy.

SECURITY CLASSIFICATION FOR INFORMATION

An important aspect of information security and risk management is recognizing the value of information and

defining appropriate procedures and protection requirements for the information. Not all information is equal and so not all information requires the same degree of protection. This requires information to be assigned a security classification. The first step in information classification is to identify a member of senior management as the owner of the particular information to be classified. Next, develop a classification policy. The policy should describe the different classification labels, define the criteria for information to be assigned a particular label, and list the required security controls for each classification. Some factors that influence which classification information should be assigned include how much value that information has to the organization, how old the information is and whether or not the information has become obsolete. Laws and other regulatory requirements are also important considerations when classifying information. The type of information security classification labels selected and used will depend on the nature of the organisation, with examples being:

- In the business sector, labels such as: Public, Sensitive, Private, Confidential.
- In the government sector, labels such as: Unclassified, Sensitive But Unclassified, Restricted, Confidential, Secret, Top Secret and their non-English equivalents.
- In cross-sectoral formations, the Traffic Light Protocol, which consists of: White, Green, Amber and Red.

All employees in the organization, as well as business partners, must be trained on the classification schema and understand the required security controls and handling procedures for each classification. The classification a

particular information asset has been assigned should be reviewed periodically to ensure the classification is still appropriate for the information and to ensure the security controls required by the classification are in place.

ACCESS CONTROL

Access to protected information must be restricted to people who are authorized to access the information. The computer programmes, and in many cases the computers that process the information, must also be authorized. This requires that mechanisms be in place to control the access to protected information. The sophistication of the access control mechanisms should be in parity with the value of the information being protected – the more sensitive or valuable the information the stronger the control mechanisms need to be. The foundation on which access control mechanisms are built start with identification and authentication. Identification is an assertion of who someone is or what something is. If a person makes the statement *"Hello, my name is John Doe"* they are making a claim of who they are. However, their claim may or may not be true. Before John Doe can be granted access to protected information it will be necessary to verify that the person claiming to be John Doe really is John Doe. Authentication is the act of verifying a claim of identity. When John Doe goes into a bank to make a withdrawal, he tells the bank teller he is John Doe (a claim of identity). The bank teller asks to see a photo ID, so he hands the teller his driver's license. The bank teller checks the license to make sure it has John Doe printed on it and compares the photograph

on the license against the person claiming to be John Doe. If the photo and name match the person, then the teller has authenticated that John Doe is who he claimed to be.

There are three different types of information that can be used for authentication: something you know, something you have, or something you are. Examples of *something you know* include such things as a PIN, a password, or your mother's maiden name. Examples of *something you have* include a driver's license or a magnetic swipe card. *Something you are* refers to biometrics. Examples of biometrics include palm prints, finger prints, voice prints and retina (eye) scans. Strong authentication requires providing information from two of the three different types of authentication information. For example, something you know plus something you have. This is called two factor authentication. On computer systems in use today, the Username is the most common form of identification and the Password is the most common form of authentication. Usernames and passwords have served their purpose but in our modern world they are no longer adequate. Usernames and passwords are slowly being replaced with more sophisticated authentication mechanisms. After a person, programme or computer has successfully been identified and authenticated then it must be determined what informational resources they are permitted to access and what actions they will be allowed to perform (run, view, create, delete, or change). This is called authorization.

Authorization to access information and other computing services begins with administrative policies and procedures. The policies prescribe what information and computing

services can be accessed, by whom, and under what conditions. The access control mechanisms are then configured to enforce these policies. Different computing systems are equipped with different kinds of access control mechanisms - some may even offer a choice of different access control mechanisms. The access control mechanism a system offers will be based upon one of three approaches to access control or it may be derived from a combination of the three approaches. The non-discretionary approach consolidates all access control under a centralized administration. The access to information and other resources is usually based on the individuals function (role) in the organization or the tasks the individual must perform. The discretionary approach gives the creator or owner of the information resource the ability to control access to those resources. In the Mandatory access control approach, access is granted or denied basing upon the security classification assigned to the information resource.

Examples of common access control mechanisms in use today include Role-based access control available in many advanced Database Management Systems, simple file permissions provided in the UNIX and Windows operating systems, Group Policy Objects provided in Windows network systems, Kerberos, RADIUS, TACACS, and the simple access lists used in many firewalls and routers. To be effective, policies and other security controls must be enforceable and upheld. Effective policies ensure that people are held accountable for their actions. All failed and successful authentication attempts must be logged, and all access to information must leave some type of audit trail.

CRYPTOGRAPHY

Information security uses cryptography to transform usable information into a form that renders it unusable by anyone other than an authorized user; this process is called encryption. Information that has been encrypted (rendered unusable) can be transformed back into its original usable form by an authorized user, who possesses the cryptographic key, through the process of decryption. Cryptography is used in information security to protect information from unauthorized or accidental disclosure while the information is in transit (either electronically or physically) and while information is in storage. Cryptography provides information security with other useful applications as well including improved authentication methods, message digests, digital signatures, non-repudiation, and encrypted network communications. Older less secure application such as telnet and ftp are slowly being replaced with more secure applications such as ssh that use encrypted network communications. Wireless communications can be encrypted using protocols such as WPA/WPA2 or the older (and less secure) WEP. Wired communications (such as ITU-T G.hn) are secured using AES for encryption and X.1035 for authentication and key exchange. Software applications such as GnuPG or PGP can be used to encrypt data files and Email.

Cryptography can introduce security problems when it is not implemented correctly. Cryptographic solutions need to be implemented using industry accepted solutions that have undergone rigorous peer review by independent experts in cryptography. The length and strength of the encryption

key is also an important consideration. A key that is weak or too short will produce weak encryption. The keys used for encryption and decryption must be protected with the same degree of rigor as any other confidential information. They must be protected from unauthorized disclosure and destruction and they must be available when needed. PKI solutions address many of the problems that surround key management.

PROCESS

The terms reasonable and prudent person, due care and due diligence have been used in the fields of Finance, Securities, and Law for many years. In recent years these terms have found their way into the fields of computing and information security. U.S.A. Federal Sentencing Guidelines now make it possible to hold corporate officers liable for failing to exercise due care and due diligence in the management of their information systems. In the business world, stockholders, customers, business partners and governments have the expectation that corporate officers will run the business in accordance with accepted business practices and in compliance with laws and other regulatory requirements. This is often described as the "reasonable and prudent person" rule. A prudent person takes due care to ensure that everything necessary is done to operate the business by sound business principles and in a legal ethical manner. A prudent person is also diligent (mindful, attentive, and ongoing) in their due care of the business. In the field of Information Security, Harris offers the following definitions of due care and due diligence:

> *"Due care are steps that are taken to show that a company has taken responsibility for the activities that take place within the corporation and has taken the necessary steps to help protect the company, its resources, and employees."* And, [Due diligence are the] *"continual activities that make sure the protection mechanisms are continually maintained and operational."* Attention should be made to two important points in these definitions. First, in due care, steps are taken to *show* - this means that the steps can be verified, measured, or even produce tangible artifacts. Second, in due diligence, there are continual activities - this means that people are actually doing things to monitor and maintain the protection mechanisms, and these activities are ongoing.

SECURITY GOVERNANCE

The Software Engineering Institute at Carnegie Mellon University, in a publication titled "Governing for Enterprise Security (GES)", defines characteristics of effective security governance. These include:

- An enterprise-wide issue
- Leaders are accountable
- Viewed as a business requirement
- Risk-based
- Roles, responsibilities, and segregation of duties defined
- Addressed and enforced in policy
- Adequate resources committed
- Staff aware and trained
- A development life cycle requirement
- Planned, managed, measurable, and measured
- Reviewed and audited

INCIDENT RESPONSE PLANS

1 to 3 paragraphs (non technical) that discuss:

- Selecting team members
- Define roles, responsibilities and lines of authority
- Define a security incident
- Define a reportable incident
- Training
- Detection
- Classification
- Escalation
- Containment
- Eradication
- Documentation

CHANGE MANAGEMENT

Change management is a formal process for directing and controlling alterations to the information processing environment. This includes alterations to desktop computers, the network, servers and software. The objectives of change management are to reduce the risks posed by changes to the information processing environment and improve the stability and reliability of the processing environment as changes are made. It is not the objective of change management to prevent or hinder necessary changes from being implemented. Any change to the information processing environment introduces an element of risk. Even apparently simple changes can have unexpected effects. One of Managements many responsibilities is the management of risk. Change management is a tool for managing the risks introduced by changes to the information processing environment. Part of the change management process

ensures that changes are not implemented at inopportune times when they may disrupt critical business processes or interfere with other changes being implemented.

Not every change needs to be managed. Some kinds of changes are a part of the everyday routine of information processing and adhere to a predefined procedure, which reduces the overall level of risk to the processing environment. Creating a new user account or deploying a new desktop computer are examples of changes that do not generally require change management. However, relocating user file shares, or upgrading the Email server pose a much higher level of risk to the processing environment and are not a normal everyday activity. The critical first steps in change management are (a) defining change (and communicating that definition) and (b) defining the scope of the change system. Change management is usually overseen by a Change Review Board composed of representatives from key business areas, security, networking, systems administrators, Database administration, applications development, desktop support and the help desk. The tasks of the Change Review Board can be facilitated with the use of automated work flow application. The responsibility of the Change Review Board is to ensure the organizations documented change management procedures are followed. The change management process is as follows:

- Requested: Anyone can request a change. The person making the change request may or may not be the same person that performs the analysis or implements the change. When a request for change is received,

it may undergo a preliminary review to determine if the requested change is compatible with the organizations business model and practices, and to determine the amount of resources needed to implement the change.

- Approved: Management runs the business and controls the allocation of resources therefore, Management must approve requests for changes and assign a priority for every change. Management might choose to reject a change request if the change is not compatible with the business model, industry standards or best practices. Management might also choose to reject a change request if the change requires more resources than can be allocated for the change.
- Planned: Planning a change involves discovering the scope and impact of the proposed change; analyzing the complexity of the change; allocation of resources and, developing, testing and documenting both implementation and backout plans. Need to define the criteria on which a decision to back out will be made.
- Tested: Every change must be tested in a safe test environment, which closely reflects the actual production environment, before the change is applied to the production environment. The backout plan must also be tested.
- Scheduled: Part of the change review board's responsibility is to assist in the scheduling of changes by reviewing the proposed implementation date for

potential conflicts with other scheduled changes or critical business activities.

- Communicated: Once a change has been scheduled it must be communicated. The communication is to give others the opportunity to remind the change review board about other changes or critical business activities that might have been overlooked when scheduling the change. The communication also serves to make the Help Desk and users aware that a change is about to occur. Another responsibility of the change review board is to ensure that scheduled changes have been properly communicated to those who will be affected by the change or otherwise have an interest in the change.
- Implemented: At the appointed date and time, the changes must be implemented. Part of the planning process was to develop an implementation plan, testing plan and, a back out plan. If the implementation of the change should fail or, the post implementation testing fails or, other "drop dead" criteria have been met, the back out plan should be implemented.
- Documented: All changes must be documented. The documentation includes the initial request for change, its approval, the priority assigned to it, the implementation, testing and back out plans, the results of the change review board critique, the date/time the change was implemented, who implemented it, and whether the change was implemented successfully, failed or postponed.

- Post change review: The change review board should hold a post implementation review of changes. It is particularly important to review failed and backed out changes. The review board should try to understand the problems that were encountered, and look for areas for improvement.

Change management procedures that are simple to follow and easy to use can greatly reduce the overall risks created when changes are made to the information processing environment.

Good change management procedures improve the over all quality and success of changes as they are implemented. This is accomplished through planning, peer review, documentation and communication. ISO/IEC 20000, The Visible OPS Handbook: Implementing ITIL in 4 Practical and Auditable Steps (Full book summary), and Information Technology Infrastructure Library all provide valuable guidance on implementing an efficient and effective change management programme information security.

BUSINESS CONTINUITY

Business continuity is the mechanism by which an organization continues to operate its critical business units, during planned or unplanned disruptions that affect normal business operations, by invoking planned and managed procedures. Unlike what most people think business continuity is not necessarily an IT system or process, simply because it is about the business. Today disasters or disruptions to business are a reality. Whether the disaster

is natural or man-made (the TIME magazine has a website on the top 10), it affects normal life and so business. So why is planning so important? Let us face reality that "all businesses recover", whether they planned for recovery or not, simply because business is about earning money for survival. The planning is merely getting better prepared to face it, knowing fully well that the best plans may fail. Planning helps to reduce cost of recovery, operational overheads and most importantly sail through some smaller ones effortlessly. For businesses to create effective plans they need to focus upon the following key questions. Most of these are common knowledge, and anyone can do a BCP.

1. Should a disaster strike, what are the first few things that I should do? Should I call people to find if they are OK or call up the bank to figure out my money is safe? This is Emergencey Response. Emergency Response services help take the first hit when the disaster strikes and if the disaster is serious enough the Emergency Response teams need to quickly get a Crisis Management team in place.
2. What parts of my business should I recover first? The one that brings me most money or the one where I spend the most, or the one that will ensure I shall be able to get sustained future growth? The identified sections are the critical business units. There is no magic bullet here, no one answer satisfies all. Businesses need to find answers that meet business requirements.
3. How soon should I target to recover my critical business units? In BCP technical jargon this is called

Recovery Time Objective, or RTO. This objective will define what costs the business will need to spend to recover from a disruption. For example, it is cheaper to recover a business in 1 day than in 1 hour.

4. What all do I need to recover the business? IT, machinery, records...food, water, people...So many aspects to dwell upon. The cost factor becomes clearer now...Business leaders need to drive business continuity. Hold on. My IT manager spent $200000 last month and created a DRP (Disaster Recovery Plan), whatever happened to that? a DRP is about continuing an IT system, and is one of the sections of a comprehensive Business Continuity Plan. Look below for more on this.
5. And where do I recover my business from... Will the business center give me space to work, or would it be flooded by many people queuing up for the same reasons that I am.
6. But once I do recover from the disaster and work in reduced production capacity, since my main operational sites are unavailable, how long can this go on. How long can I do without my original sites, systems, people? this defines the amount of business resilience a business may have.
7. Now that I know how to recover my business. How do I make sure my plan works? Most BCP pundits would recommend testing the plan at least once a year, reviewing it for adequacy and rewriting or updating the plans either annually or when businesses change.

DISASTER RECOVERY PLANNING

While a business continuity plan (BCP) takes a broad approach to dealing with organizational-wide effects of a disaster, a disaster recovery plan (DRP), which is a subset of the business continuity plan, is instead focused on taking the necessary steps to resume normal business operations as quickly as possible. A disaster recovery plan is executed immediately after the disaster occurs and details what steps are to be taken in order to recover critical information technology infrastructure.

LAWS AND REGULATIONS

Below is a partial listing of European, United Kingdom, Canadian and USA governmental laws and regulations that have, or will have, a significant effect on data processing and information security. Important industry sector regulations have also been included when they have a significant impact on information security.

- UK Data Protection Act 1998 makes new provisions for the regulation of the processing of information relating to individuals, including the obtaining, holding, use or disclosure of such information. The European Union Data Protection Directive (EUDPD) requires that all EU member must adopt national regulations to standardize the protection of data privacy for citizens throughout the EU.
- The Computer Misuse Act 1990 is an Act of the UK Parliament making computer crime (e.g. cracking - sometimes incorrectly referred to as hacking) a

criminal offence. The Act has become a model upon which several other countries including Canada and the Republic of Ireland have drawn inspiration when subsequently drafting their own information security laws.

- EU Data Retention laws requires Internet service providers and phone companies to keep data on every electronic message sent and phone call made for between six months and two years.
- The Family Educational Rights and Privacy Act (FERPA) (20 U.S.C. § 1232 g; 34 CFR Part 99) is a USA Federal law that protects the privacy of student education records. The law applies to all schools that receive funds under an applicable programme of the U.S. Department of Education. Generally, schools must have written permission from the parent or eligible student in order to release any information from a student's education record.
- Health Insurance Portability and Accountability Act (HIPAA) of 1996 requires the adoption of national standards for electronic health care transactions and national identifiers for providers, health insurance plans, and employers. And, it requires health care providers, insurance providers and employers to safeguard the security and privacy of health data.
- Gramm-Leach-Bliley Act of 1999 (GLBA), also known as the Financial Services Modernization Act of 1999, protects the privacy and security of private financial information that financial institutions collect, hold, and process.

- Sarbanes-Oxley Act of 2002 (SOX). Section 404 of the act requires publicly traded companies to assess the effectiveness of their internal controls for financial reporting in annual reports they submit at the end of each fiscal year. Chief information officers are responsible for the security, accuracy and the reliability of the systems that manage and report the financial data. The act also requires publicly traded companies to engage independent auditors who must attest to, and report on, the validity of their assessments.
- Payment Card Industry Data Security Standard (PCI DSS) establishes comprehensive requirements for enhancing payment account data security. It was developed by the founding payment brands of the PCI Security Standards Council, including American Express, Discover Financial Services, JCB, MasterCard Worldwide and Visa International, to help facilitate the broad adoption of consistent data security measures on a global basis. The PCI DSS is a multifaceted security standard that includes requirements for security management, policies, procedures, network architecture, software design and other critical protective measures.
- State Security Breach Notification Laws (California and many others) require businesses, nonprofits, and state institutions to notify consumers when unencrypted "personal information" may have been compromised, lost, or stolen.

- Personal Information Protection and Electronics Document Act (PIPEDA) – An Act to support and promote electronic commerce by protecting personal information that is collected, used or disclosed in certain circumstances, by providing for the use of electronic means to communicate or record information or transactions and by amending the Canada Evidence Act, the Statutory Instruments Act and the Statute Revision Act.

SOURCES OF STANDARDS

International Organization for Standardization (ISO) is a consortium of national standards institutes from 157 countries, coordinated through a secretariat in Geneva, Switzerland. ISO is the world's largest developer of standards. ISO 15443: "Information technology - Security techniques - A framework for IT security assurance", ISO/IEC 27002: "Information technology - Security techniques - Code of practice for information security management", ISO-20000: "Information technology - Service management", and ISO/IEC27001: "Information technology - Security techniques - Information security management systems - Requirements" are of particular interest to information security professionals.

The USA National Institute of Standards and Technology (NIST) is a non-regulatory federal agency within the U.S. Department of Commerce.

The NIST Computer Security Division develops standards, metrics, tests and validation programmes as well as publishes

standards and guidelines to increase secure IT planning, implementation, management and operation. NIST is also the custodian of the USA Federal Information Processing Standard publications (FIPS). The Internet Society is a professional membership society with more than 100 organization and over 20,000 individual members in over 180 countries.

It provides leadership in addressing issues that confront the future of the Internet, and is the organization home for the groups responsible for Internet infrastructure standards, including the Internet Engineering Task Force (IETF) and the Internet Architecture Board (IAB). The ISOC hosts the Requests for Comments (RFCs) which includes the Official Internet Protocol Standards and the RFC-2196 Site Security Handbook.

The Information Security Forum is a global nonprofit organization of several hundred leading organizations in financial services, manufacturing, telecommunications, consumer goods, government, and other areas. It undertakes research into information security practices and offers advice in its biannual Standard of Good Practice and more detailed advisories for members. The IT Baseline Protection Catalogs, or IT-Grundschutz Catalogs, ("IT Baseline Protection Manual" before 2005) are a collection of documents from the German Federal Office for Security in Information Technology (FSI), useful for detecting and combating security-relevant weak points in the IT environment ("IT cluster"). The collection encompasses over 3000 pages with the introduction and catalogs.

PROFESSIONALISM

Information security professionalism is the set of knowledge that people working in Information security and similar fields (Information Assurance and Computer security) should have and eventually demonstrate through certifications from well respected organizations. It also encompasses the education process required to accomplish different tasks in these fields. Information technology adoption is always increasing and spread to vital infrastructure for civil and military organizations. Everybody can get involved in the Cyberwar. It is crucial that a nation can have skilled professional to defend its vital interests.

CONCLUSION

Information security is the ongoing process of exercising due care and due diligence to protect information, and information systems, from unauthorized access, use, disclosure, destruction, modification, or disruption or distribution. The never ending process of information security involves ongoing training, assessment, protection, monitoring & detection, incident response & repair, documentation, and review. This makes information security an indispensable part of all the business operations across different domains.

7

Computer Security Policy

UNITED STATES

Cybersecurity Act of 2010

On April 1, 2009, Senator Jay Rockefeller (D-WV) introduced the "Cybersecurity Act of 2009 - S. 773" (full text) in the Senate; the bill, co-written with Senators Evan Bayh (D-IN), Barbara Mikulski (D-MD), Bill Nelson (D-FL), and Olympia Snowe (R-ME), was referred to the Committee on Commerce, Science, and Transportation, which approved a revised version of the same bill (the "Cybersecurity Act of 2010") on March 24, 2010. The bill seeks to increase collaboration between the public and the private sector on cybersecurity issues, especially those private entities that own infrastructures that are critical to national security interests (the bill quotes John Brennan, the Assistant to the President for Homeland Security and Counterterrorism: "our

nation's security and economic prosperity depend on the security, stability, and integrity of communications and information infrastructure that are largely privately-owned and globally-operated" and talks about the country's response to a "cyber-Katrina".), increase public awareness on cybersecurity issues, and foster and fund cybersecurity research. Some of the most controversial parts of the bill include Paragraph 315, which grants the President the right to "order the limitation or shutdown of Internet traffic to and from any compromised Federal Government or United States critical infrastructure information system or network." The Electronic Frontier Foundation, an international non-profit digital rights advocacy and legal organization based in the United States, characterized the bill as promoting a "potentially dangerous approach that favours the dramatic over the sober response".

International Cybercrime Reporting and Cooperation Act

On March 25, 2010, Representative Yvette Clarke (D-NY) introduced the "International Cybercrime Reporting and Cooperation Act - H.R.4962" (full text) in the House of Representatives; the bill, co-sponsored by seven other representatives (among whom only one Republican), was referred to three House committees. The bill seeks to make sure that the administration keeps Congress informed on information infrastructure, cybercrime, and end-user protection worldwide. It also "directs the President to give priority for assistance to improve legal, judicial, and enforcement capabilities with respect to cybercrime to countries with low

information and communications technology levels of development or utilization in their critical infrastructure, telecommunications systems, and financial industries" as well as to develop an action plan and an annual compliance assessment for countries of "cyber concern".

Protecting Cyberspace as a National Asset Act of 2010 ("*Kill Switch Bill*")

On June 19, 2010, United States Senator Joe Lieberman (I-CT) introduced a bill called "Protecting Cyberspace as a National Asset Act of 2010 - S.3480" (full text in pdf), which he co-wrote with Senator Susan Collins (R-ME) and Senator Thomas Carper (D-DE). If signed into law, this controversial bill, which the American media dubbed the "*Kill switch bill*", would grant the President emergency powers over the Internet. However, all three co-authors of the bill issued a statement claiming that instead, the bill "[narrowed] existing broad Presidential authority to take over telecommunications networks".

TERMINOLOGY

The following terms used in engineering secure systems are explained below.

- Authentication techniques can be used to ensure that communication end-points are who they say they are.
- Automated theorem proving and other verification tools can enable critical algorithms and code used in secure systems to be mathematically proven to meet their specifications.

- Capability and access control list techniques can be used to ensure privilege separation and mandatory access control. This section discusses their use.
- Chain of trust techniques can be used to attempt to ensure that all software loaded has been certified as authentic by the system's designers.
- Cryptographic techniques can be used to defend data in transit between systems, reducing the probability that data exchanged between systems can be intercepted or modified.
- Firewalls can provide some protection from online intrusion.
- A microkernel is a carefully crafted, deliberately small corpus of software that underlies the operating system *per se* and is used solely to provide very low-level, very precisely defined primitives upon which an operating system can be developed. A simple example with considerable didactic value is the early '90s GEMSOS (Gemini Computers), which provided extremely low-level primitives, such as "segment" management, atop which an operating system could be built. The theory (in the case of "segments") was that—rather than have the operating system itself worry about mandatory access separation by means of military-style labeling—it is safer if a low-level, independently scrutinized module can be charged solely with the management of individually labeled segments, be they memory "segments" or file system "segments" or executable text "segments." If software below the visibility of the operating system is (as in

this case) charged with labeling, there is no theoretically viable means for a clever hacker to subvert the labeling scheme, since the operating system *per se* does not provide mechanisms for interfering with labeling: the operating system is, essentially, a client (an "application," arguably) atop the microkernel and, as such, subject to its restrictions.

- Endpoint Security software helps networks to prevent data theft and virus infection through portable storage devices, such as USB drives.

Some of the following items may belong to the computer insecurity article:

- Access authorization restricts access to a computer to group of users through the use of authentication systems. These systems can protect either the whole computer – such as through an interactive logon screen – or individual services, such as an FTP server. There are many methods for identifying and authenticating users, such as passwords, identification cards, and, more recently, smart cards and biometric systems.
- Anti-virus software consists of computer programmes that attempt to identify, thwart and eliminate computer viruses and other malicious software (malware).
- Applications with known security flaws should not be run. Either leave it turned off until it can be patched or otherwise fixed, or delete it and replace it with some other application. Publicly known flaws

are the main entry used by worms to automatically break into a system and then spread to other systems connected to it. The security website Secunia provides a search tool for unpatched known flaws in popular products.

- Backups are a way of securing information; they are another copy of all the important computer files kept in another location. These files are kept on hard disks, CD-Rs, CD-RWs, and tapes. Suggested locations for backups are a fireproof, waterproof, and heat proof safe, or in a separate, offsite location than that in which the original files are contained. Some individuals and companies also keep their backups in safe deposit boxes inside bank vaults. There is also a fourth option, which involves using one of the file hosting services that backs up files over the Internet for both business and individuals.
  - o Backups are also important for reasons other than security. Natural disasters, such as earthquakes, hurricanes, or tornadoes, may strike the building where the computer is located. The building can be on fire, or an explosion may occur. There needs to be a recent backup at an alternate secure location, in case of such kind of disaster. Further, it is recommended that the alternate location be placed where the same disaster would not affect both locations. Examples of alternate disaster recovery sites being compromised by the same disaster that affected the primary site include having had a primary site in World Trade Center

I and the recovery site in 7 World Trade Center, both of which were destroyed in the 9/11 attack, and having one's primary site and recovery site in the same coastal region, which leads to both being vulnerable to hurricane damage (e.g. primary site in New Orleans and recovery site in Jefferson Parish, both of which were hit by Hurricane Katrina in 2005). The backup media should be moved between the geographic sites in a secure manner, in order to prevent them from being stolen.

- Encryption is used to protect the message from the eyes of others. Cryptographically secure ciphers are designed to make any practical attempt of breaking infeasible. Symmetric-key ciphers are suitable for bulk encryption using shared keys, and public-key encryption using digital certificates can provide a practical solution for the problem of securely communicating when no key is shared in advance.
- Firewalls are systems which help protect computers and computer networks from attack and subsequent intrusion by restricting the network traffic which can pass through them, based on a set of system administrator defined rules.
- Honey pots are computers that are either intentionally or unintentionally left vulnerable to attack by crackers. They can be used to catch crackers or fix vulnerabilities.
- Intrusion-detection systems can scan a network for people that are on the network but who should not

be there or are doing things that they should not be doing, for example trying a lot of passwords to gain access to the network.

- Pinging The ping application can be used by potential crackers to find if an IP address is reachable. If a cracker finds a computer, they can try a port scan to detect and attack services on that computer.
- Social engineering awareness keeps employees aware of the dangers of social engineering and/or having a policy in place to prevent social engineering can reduce successful breaches of the network and servers.
- File Integrity Monitors are tools used to detect changes in the integrity of systems and files.

Security-Focused Operating System

This is an alphabetical list of operating systems with a sharp security focus. Their order does not imply rank. In our context, "Security-focused" means that the project is devoted to increasing the security as a major goal. As such, something can be secure without being "security-focused." For example, almost all of the operating systems mentioned here are faced with security bug fixes in their lifetime; however, they do all strive to consistently approach all generic security flaws inherent in their design with new ideas in an attempt to create a secure computing environment.

BSD

BSD is a family of Unix variants derived from a code base originating at the University of California, Berkeley. All

derived BSD operating systems are released under the terms of a BSD-style license. There are several BSD variants, with only one being heavily focused on security.

OPENBSD

OpenBSD is an open source BSD operating system that is known to be concerned heavily with security. The project has completed rigorous manual reviews of the code and addressed issues most systems have not. OpenBSD also supplies an executable space protection scheme known as WX (memory is writable xor executable), as well as a ProPolice compiled executable base.

TRUSTEDBSD

TrustedBSD is a sub-project of FreeBSD designed to add trusted operating system extensions, targeting the Common Criteria for Information Technology Security Evaluation. Its main focuses are working on access control lists, event auditing, extended attributes, mandatory access controls, and fine-grained capabilities. Since access control lists are known to be confronted with the confused deputy problem, capabilities are a different way to avoid this issue. As part of the TrustedBSD project, there is also a port of the NSA's FLASK/TE implementation to run on FreeBSD. Many of these trusted extensions have been integrated into the main FreeBSD branch starting at 5.x.

LINUX

Linux itself is inherently security-focused; however, many distributions and projects attempt to make Linux more secure.

ANNVIX

Annvix was originally forked from Mandriva to provide a security-focused server distribution that employs ProPolice protection, hardened configuration, and a small footprint. There have been plans to include full support for the RSBAC Mandatory access control system. However, Annvix seems to be a dormant operating system with the last version being released December 30, 2007.

ENGARDE SECURE LINUX

EnGarde Secure Linux is a secure platform designed for servers. It has boasted a browser-based tool for MAC using SELinux since 2003. Additionally, it can be accompanied with Web, DNS, and Email enterprise applications, specifically focusing on security without any unnecessary software. The community platform of EnGarde Secure Linux is the bleeding-edge version freely available for download.

FEDORA

Fedora is a free, Red Hat sponsored community developed Linux distribution. It is one of those mainstream Linux distribution, with a concentrated effort to improve system security, as a consequence it boasts a fully integrated SELinux MAC and fine-grained executable memory permission system (Exec Shield) and all binaries compiled with GCC's standard stack-smashing protection, as well as focusing on getting security updates into the system in a timely manner.

HARDENED GENTOO

Hardened Gentoo is a subproject of the Gentoo Linux project. Hardened Gentoo offers a ProPolice protected and

Position Independent Executable base using exactly the same package tree as Gentoo. Executable space protection in Hardened Gentoo is handled by PaX. The Hardened Gentoo project is an extremely modular project, and also provides subprojects to integrate other intrusion-detection and Mandatory access control systems into Gentoo. All of these can be optionally installed in any combination, with or without PaX and a ProPolice base.

HARDENED LINUX

Hardened Linux is a small distribution for firewalls, intrusion detection systems, VPN-gateways and authentication jobs that is still under heavy development. It includes GRSecurity, PaX and GCC stack smashing protection.

IMMUNIX

Immunix is a commercial distribution of Linux focused heavily on security. They supply many systems of their own making, including StackGuard; cryptographic signing of executables; race condition patches; and format string exploit guarding code. Immunix traditionally releases older versions of their distribution free for non-commercial use. Note that the Immunix distribution itself is licensed under two licenses: The Immunix commercial and non-commercial licenses. Many tools within are GPL, however; as is the kernel.

OPENWALL PROJECT

Owl by a developer known as Solar Designer was the first distribution to have a non-executable userspace stack, /tmp race condition protection and access control restrictions

to /procdata, by way of a kernel patch. It also features a per-user tmp directory via the pam_mktemp PAM module, and supports Blowfish password encryption.

RED HAT ENTERPRISE LINUX

Red Hat Enterprise Linux - offers the same security benefits as Fedora with the additional support of back-porting security fixes to the released versions of the packages (particularly the kernel) so the sys-admin does not have to perform a significant (and risky) upgrade to get a security fix.

UBUNTU

Like Fedora and Red Hat Enterprise Linux, Ubuntu provides swift security fixes for stable releases. It also has AppArmor installed by default and supports SELinux. Ubuntu locks the root account by default.

SOLARIS

Solaris is a Unix variant created by Sun Microsystems. Solaris itself is not inherently security-focused. Majority of Solaris source code has been released via the OpenSolaris project, mostly under the Common Development and Distribution License. Enhancements to OpenSolaris, both security related and others, are backported to the official Solaris when Sun certifies their quality.

TRUSTED SOLARIS

Trusted Solaris is a security-focused version of the Solaris Unix operating system. Aimed primarily at the government

computing sector, Trusted Solaris adds detailed auditing of all tasks, pluggable authentication, mandatory access control, additional physical authentication devices, and fine-grained access control. Trusted Solaris is Common Criteria certified. The most recent version, Trusted Solaris 8, received the EAL4 certification level augmented by a number of protection profiles.

SOLARIS 10 AND TRUSTED FUNCTIONALITY

Trusted Solaris functionality has now been added to the mainstream version of Solaris. In the 11/06 update to Solaris 10, the *Solaris Trusted Extensions* feature adds mandatory access control and labelled security. Introduced in the same update, the *Secure by Default Networking* feature implements less services on by default compared to most previous releases which had most services enabled. RBAC, found in both mainstream Solaris and Trusted Solaris, dramatically lessens the need for using root directly by providing a way for fine grained control over various administrative tasks.

Security Architecture

Security provided by IT Systems can be defined as the IT system's ability to be able to protect confidentiality and integrity of processed data, as well as to be able to provide availability of the system and data. "IT Architecture" may be defined as a set of design artifacts, that are relevant for describing an object such that it can be produced to requirements (quality) as well as maintained over the period of its useful life (change). The design artifact describe the

structure of components, their inter-relationships, and the principles and guidelines governing their design and evolution over time. Consequently the definition of "IT Security Architecture" may be considered as:

> The design artifacts that describe how the security controls (= security countermeasures) are positioned and how they relate to the overall IT Architecture. These controls serve the purpose to maintain the system's quality attributes, among them confidentiality, integrity and availability.

Security qualities are often considered as Non-functional requirements when systems are designed. In other words they are not required for the system to meet its functional goals such as processing financial transactions, but are needed for a given level of assurance that the system will perform to meet the functional requirements that have been defined. In recent years there has been a trend towards a hierarchy of control objectives, controls and specific technical implementations of controls, which are implemented within a given security architecture in order to meet the security requirements.

8

Data Security

Data security is the means of ensuring that data is kept safe from corruption and that access to it is suitably controlled.

Thus data security helps to ensure privacy. It also helps in protecting personal data.

DATA SECURITY TECHNOLOGIES

DISK ENCRYPTION

Disk encryption refers to encryption technology that encrypts data on a hard disk drive. Disk encryption typically takes form in either software or hardware. Disk encryption is often referred to as on-the-fly encryption ("OTFE") or transparent encryption.

HARDWARE BASED MECHANISMS FOR PROTECTING DATA

Software based security solutions encrypt the data to prevent data from being stolen. However, a malicious programme or a hacker may corrupt the data in order to make it unrecoverable or unusable. Similarly, encrypted operating systems can be corrupted by a malicious programme or a hacker, making the system unusable. Hardware-based security solutions can prevent read and write access to data and hence offers very strong protection against tampering and unauthorized access. Hardware based or assisted computer security offers an alternative to software-only computer security. Security tokens such as those using PKCS#11 may be more secure due to the physical access required in order to be compromised. Access is enabled only when the token is connected and correct PIN is entered. However, dongles can be used by anyone who can gain physical access to it. Newer technologies in hardware based security solves this problem offering fool proof security for data.

Working of Hardware based security: A hardware device allows a user to login, logout and to set different privilege levels by doing manual actions. The device uses biometric technology to prevent malicious users from logging in, logging out, and changing privilege levels. The current state of a user of the device is read by controllers in peripheral devices such as harddisks. Illegal access by a malicious user or a malicious programme is interrupted based on the current state of a user by harddisk and DVD controllers making illegal access to data impossible. Hardware based access

control is more secure than protection provided by the operating systems as operating systems are vulnerable to malicious attacks by viruses and hackers. The data on harddisks can be corrupted after a malicious access is obtained. With hardware based protection, software cannot manipulate the user privilege levels, it is impossible for a hacker or a malicious programme to gain access to secure data protected by hardware or perform unauthorized privileged operations. The hardware protects the operating system image and file system privileges from being tampered. Therefore, a completely secure system can be created using a combination of hardware based security and secure system administration policies.

BACKUPS

Backups are used to ensure data which is lost can be recovered

DATA MASKING

Data Masking of structured data is the process of obscuring (masking) specific data within a database table or cell to ensure that data security is maintained and sensitive information is not exposed to unauthorized personnel.

This may include masking the data from users (for example so banking customer representatives can only see the last 4 digits of a customers national identity number), developers (who need real production data to test new software releases but should not be able to see sensitive financial data), outsourcing vendors, etc.

DATA ERASURE

Data erasure is a method of software-based overwriting that completely destroys all electronic data residing on a hard drive or other digital media to ensure that no sensitive data is leaked when an asset is retired or reused.

INTERNATIONAL LAWS AND STANDARDS

INTERNATIONAL LAWS

In the UK, the Data Protection Act is used to ensure that personal data is accessible to those whom it concerns, and provides redress to individuals if there are inaccuracies. This is particularly important to ensure individuals are treated fairly, for example for credit checking purposes. The Data Protection Act states that only individuals and companies with legitimate and lawful reasons can process personal information and cannot be shared.

INTERNATIONAL STANDARDS

The International Standard ISO/IEC 17799 covers data security under the topic of information security, and one of its cardinal principles is that all stored information, i.e. data, should be owned so that it is clear whose responsibility it is to protect and control access to that data. The Trusted Computing Group is an organization that helps standardize computing security technologies.

Database Model

A '*database model*' is the theoretical foundation of a database and fundamentally determines in which manner

data can be stored, organized and manipulated in a database system. It thereby defines the infrastructure offered by a particular database system. The most popular example of a database model is the relational model.

OVERVIEW

A database model is a theory or specification describing how a database is structured and used. Several such models have been suggested. Common models include:

- Hierarchical model
- Network model
- Relational model
- Entity-relationship
- Object-relational model
- Object model

A data model is not just a way of structuring data: it also defines a set of operations that can be performed on the data. The relational model, for example, defines operations such as select, project, and join. Although these operations may not be explicit in a particular query language, they provide the foundation on which a query language is built.

MODELS

Various techniques are used to model data structure. Most database systems are built around one particular data model, although it is increasingly common for products to offer support for more than one model. For any one logical model various physical implementations may be possible,

and most products will offer the user some level of control in tuning the physical implementation, since the choices that are made have a significant effect on performance. An example of this is the relational model: all serious implementations of the relational model allow the creation of indexes which provide fast access to rows in a table if the values of certain columns are known.

FLAT MODEL

The flat (or table) model consists of a single, two-dimensional array of data elements, where all members of a given column are assumed to be similar values, and all members of a row are assumed to be related to one another. For instance, columns for name and password that might be used as a part of a system security database. Each row would have the specific password associated with an individual user. Columns of the table often have a type associated with them, defining them as character data, date or time information, integers, or floating point numbers. This may not strictly qualify as a data model, as defined above.

HIERARCHICAL MODEL

In a hierarchical model, data is organized into a tree-like structure, implying a single upward link in each record to describe the nesting, and a sort field to keep the records in a particular order in each same-level list. Hierarchical structures were widely used in the early mainframe database management systems, such as the Information Management System (IMS) by IBM, and now describe the structure of

XML documents. This structure allows one 1:N relationship between two types of data. This structure is very efficient to describe many relationships in the real world; recipes, table of contents, ordering of paragraphs/verses, any nested and sorted information. However, the hierarchical structure is inefficient for certain database operations when a full path (as opposed to upward link and sort field) is not also included for each record. Parent–child relationship: Child may only have one parent but a parent can have multiple children. Parents and children are tied together by links called "pointers". A parent will have a list of pointers to each of their children.

NETWORK MODEL

The network model (defined by the CODASYL specification) organizes data using two fundamental constructs, called *records* and *sets*. Records contain fields (which may be organized hierarchically, as in the programming language COBOL). Sets (not to be confused with mathematical sets) define one-to-many relationships between records: one owner, many members. A record may be an owner in any number of sets, and a member in any number of sets.

The network model is a variation on the hierarchical model, to the extent that it is built on the concept of multiple branches (lower-level structures) emanating from one or more nodes (higher-level structures), while the model differs from the hierarchical model in that branches can be connected to multiple nodes. The network model is able to represent redundancy in data more efficiently than in the hierarchical model.

The operations of the network model are navigational in style: a programme maintains a current position, and navigates from one record to another by following the relationships in which the record participates. Records can also be located by supplying key values. Although it is not an essential feature of the model, network databases generally implement the set relationships by means of pointers that directly address the location of a record on disk. This gives excellent retrieval performance, at the expense of operations such as database loading and reorganization.

Most object databases use the navigational concept to provide fast navigation across networks of objects, generally using object identifiers as "smart" pointers to related objects. Objectivity/DB, for instance, implements named 1:1, 1:many, many:1 and many:many named relationships that can cross databases. Many object databases also support SQL, combining the strengths of both models.

RELATIONAL MODEL

The relational model was introduced by E.F. Codd in 1970 as a way to make database management systems more independent of any particular application. It is a mathematical model defined in terms of predicate logic and set theory. The products that are generally referred to as relational databases in fact implement a model that is only an approximation to the mathematical model defined by Codd. Three key terms are used extensively in relational database models: *relations*, *attributes*, and *domains*. A relation is a table with columns and rows. The named

columns of the relation are called attributes, and the domain is the set of values the attributes are allowed to take. The basic data structure of the relational model is the table, where information about a particular entity (say, an employee) is represented in rows (also called tuples) and columns. Thus, the "relation" in "relational database" refers to the various tables in the database; a relation is a set of tuples. The columns enumerate the various attributes of the entity (the employee's name, address or phone number, for example), and a row is an actual instance of the entity (a specific employee) that is represented by the relation. As a result, each tuple of the employee table represents various attributes of a single employee.

All relations (and, thus, tables) in a relational database have to adhere to some basic rules to qualify as relations. First, the ordering of columns is immaterial in a table. Second, there can't be identical tuples or rows in a table. And third, each tuple will contain a single value for each of its attributes. A relational database contains multiple tables, each similar to the one in the "flat" database model. One of the strengths of the relational model is that, in principle, any value occurring in two different records (belonging to the same table or to different tables), implies a relationship among those two records. Yet, in order to enforce explicit integrity constraints, relationships between records in tables can also be defined explicitly, by identifying or non-identifying parent-child relationships characterized by assigning cardinality (1:1, (0)1:M, M:M). Tables can also have a designated single attribute or a set of attributes that can act as a "key", which can be used to uniquely identify

each tuple in the table. A key that can be used to uniquely identify a row in a table is called a primary key. Keys are commonly used to join or combine data from two or more tables. For example, an *Employee* table may contain a column named *Location* which contains a value that matches the key of a *Location* table. Keys are also critical in the creation of indexes, which facilitate fast retrieval of data from large tables.

Any column can be a key, or multiple columns can be grouped together into a compound key. It is not necessary to define all the keys in advance; a column can be used as a key even if it was not originally intended to be one. A key that has an external, real-world meaning (such as a person's name, a book's ISBN, or a car's serial number) is sometimes called a "natural" key. If no natural key is suitable (think of the many people named *Brown*), an arbitrary or surrogate key can be assigned (such as by giving employees ID numbers). In practice, most databases have both generated and natural keys, because generated keys can be used internally to create links between rows that cannot break, while natural keys can be used, less reliably, for searches and for integration with other databases. (For example, records in two independently developed databases could be matched up by social security number, except when the social security numbers are incorrect, missing, or have changed.)

DIMENSIONAL MODEL

The dimensional model is a specialized adaptation of the relational model used to represent data in data warehouses

in a way that data can be easily summarized using OLAP queries. In the dimensional model, a database consists of a single large table of facts that are described using dimensions and measures. A dimension provides the context of a fact (such as who participated, when and where it happened, and its type) and is used in queries to group related facts together. Dimensions tend to be discrete and are often hierarchical; for example, the location might include the building, state, and country. A measure is a quantity describing the fact, such as revenue. It's important that measures can be meaningfully aggregated - for example, the revenue from different locations can be added together. In an OLAP query, dimensions are chosen and the facts are grouped and added together to create a summary. The dimensional model is often implemented on top of the relational model using a star schema, consisting of one table containing the facts and surrounding tables containing the dimensions. Particularly complicated dimensions might be represented using multiple tables, resulting in a snowflake schema. A data warehouse can contain multiple star schemas that share dimension tables, allowing them to be used together. Coming up with a standard set of dimensions is an important part of dimensional modeling.

OBJECTIONAL DATABASE MODELS

In recent years, the object-oriented paradigm has been applied to database technology, creating a new programming model known as object databases. These databases attempt to bring the database world and the application programming world closer together, in particular by ensuring that the

database uses the same type system as the application programme. This aims to avoid the overhead (sometimes referred to as the *impedance mismatch*) of converting information between its representation in the database (for example as rows in tables) and its representation in the application programme (typically as objects). At the same time, object databases attempt to introduce the key ideas of object programming, such as encapsulation and polymorphism, into the world of databases. A variety of these ways have been tried for storing objects in a database. Some products have approached the problem from the application programming end, by making the objects manipulated by the programme persistent.

This also typically requires the addition of some kind of query language, since conventional programming languages do not have the ability to find objects based on their information content. Others have attacked the problem from the database end, by defining an object-oriented data model for the database, and defining a database programming language that allows full programming capabilities as well as traditional query facilities.

Object databases suffered because of a lack of standardization: although standards were defined by ODMG, they were never implemented well enough to ensure interoperability between products. Nevertheless, object databases have been used successfully in many applications: usually specialized applications such as engineering databases or molecular biology databases rather than mainstream commercial data processing. However, object

database ideas were picked up by the relational vendors and influenced extensions made to these products and indeed to the SQL language.

Object Database

An object database (also object-oriented database) is a database model in which information is represented in the form of objects as used in object-oriented programming. Object databases are a niche field within the broader database management system (DBMS) market dominated by relational database management systems. Object databases have been considered since the early 1980s and 1990s but they have made little impact on mainstream commercial data processing, though there is some usage in specialized areas.

OVERVIEW

When database capabilities are combined with object-oriented programming language capabilities, the result is an object-oriented database management system (OODBMS). Today's trend in programming languages is to utilize objects, thereby making OODBMS ideal for object-oriented programmers because they can develop the product, store them as objects, and can replicate or modify existing objects to make new objects within the OODBMS. Information today includes not only data but video, audio, graphs, and photos which are considered complex data types. Relational DBMS are not natively capable of supporting these complex data types. By being integrated with the programming language, the programmer can maintain consistency within

one environment because both the OODBMS and the programming language will use the same model of representation. Relational DBMS projects using complex data types would have to be divided into two separate tasks: the database model and the application.

As the usage of web-based technology increases with the implementation of Intranets and extranets, companies have a vested interest in OODBMS to display their complex data. Using a DBMS that has been specifically designed to store data as objects gives an advantage to those companies that are geared towards multimedia presentation or organizations that utilize computer-aided design (CAD). Some object-oriented databases are designed to work well with object-oriented programming languages such as Ruby, Python, Perl, Java, C#, Visual Basic .NET, C++, Objective-C and Smalltalk; others have their own programming languages. OODBMSs use exactly the same model as object-oriented programming languages.

HISTORY

Object database management systems grew out of research during the early to mid-1970s into having intrinsic database management support for graph-structured objects. The term "object-oriented database system" first appeared around 1985. Notable research projects included Encore-Ob/Server (Brown University), EXODUS (University of Wisconsin–Madison), IRIS (Hewlett-Packard), ODE (Bell Labs), ORION (Microelectronics and Computer Technology Corporation or MCC), Vodak (GMD-IPSI), and Zeitgeist (Texas

Instruments). The ORION project had more published papers than any of the other efforts. Won Kim of MCC compiled the best of those papers in a book published by The MIT Press. Early commercial products included Gemstone (Servio Logic, name changed to GemStone Systems), Gbase (Graphael), and Vbase (Ontologic). The early to mid-1990s saw additional commercial products enter the market. These included ITASCA (Itasca Systems), Jasmine (Fujitsu, marketed by Computer Associates), Matisse (Matisse Software), Objectivity/DB (Objectivity, Inc.), ObjectStore (Progress Software, acquired from eXcelon which was originally Object Design), ONTOS (Ontos, Inc., name changed from Ontologic), O_2 (O_2 Technology, merged with several companies, acquired by Informix, which was in turn acquired by IBM), POET (now FastObjects from Versant which acquired Poet Software), Versant Object Database (Versant Corporation), VOSS (Logic Arts) and JADE (Jade Software Corporation).

Some of these products remain on the market and have been joined by new open source and commercial products such as InterSystems CACHÉ. Object database management systems added the concept of persistence to object programming languages. The early commercial products were integrated with various languages: GemStone (Smalltalk), Gbase (LISP), Vbase (COP) and VOSS (Virtual Object Storage System for Smalltalk). For much of the 1990s, C++ dominated the commercial object database management market. Vendors added Java in the late 1990s and more recently, C#. Starting in 2004, object databases have seen a second growth period when open source object

databases emerged that were widely affordable and easy to use, because they are entirely written in OOP languages like Smalltalk, Java or C#, such as db4o (db4objects), DTS/S1 from Obsidian Dynamics and Perst (McObject), available under dual open source and commercial licensing.

TIMELINE

- 1985 – Term Object Database first introduced
- 1988
  - o Versant Corporation started (as Object Sciences Corp)
  - o Objectivity, Inc. founded
- Early 1990s
  - o Gemstone (Smalltalk)
  - o GBase (LISP)
  - o VBase (O2- ONTOS – INFORMIX)
  - o Objectivity/DB launched
- Mid 1990's
  - o Versant Object Database
  - o ObjectStore
  - o Poet
  - o Jade
  - o Matisse
- 2000's
  - o Cache'
  - o db4o project started by Carl Rosenberger
  - o ObjectDB for Java
- 2001

  - o IBM acquires Informix (Illustra) integrates with DB2
  - o db4o shipped to first pilot customer
- 2004 - db4o's commercial launch as db4objects, Inc.
- 2008 - db4o acquired by Versant Corporation

ADOPTION OF OBJECT DATABASES

Object databases based on persistent programming acquired a niche in application areas such as engineering and spatial databases, telecommunications, and scientific areas such as high energy physics and molecular biology. They have made little impact on mainstream commercial data processing, though there is some usage in specialized areas of financial services. It is also worth noting that object databases held the record for the World's largest database (being the first to hold over 1000 terabytes at Stanford Linear Accelerator Center) and the highest ingest rate ever recorded for a commercial database at over one Terabyte per hour. Another group of object databases focuses on embedded use in devices, packaged software, and real-time systems.

TECHNICAL FEATURES

Most object databases also offer some kind of query language, allowing objects to be found by a more declarative programming approach. It is in the area of object query languages, and the integration of the query and navigational interfaces, that the biggest differences between products are found. An attempt at standardization was made by the

ODMG with the Object Query Language, OQL. Access to data can be faster because joins are often not needed (as in a tabular implementation of a relational database). This is because an object can be retrieved directly without a search, by following pointers. (It could, however, be argued that "joining" is a higher-level abstraction of pointer following.) Another area of variation between products is in the way that the schema of a database is defined. A general characteristic, however, is that the programming language and the database schema use the same type definitions. Multimedia applications are facilitated because the class methods associated with the data are responsible for its correct interpretation.

Many object databases, for example VOSS, offer support for versioning. An object can be viewed as the set of all its versions. Also, object versions can be treated as objects in their own right. Some object databases also provide systematic support for triggers and constraints which are the basis of active databases. The efficiency of such a database is also greatly improved in areas which demand massive amounts of data about one item. For example, a banking institution could get the user's account information and provide them efficiently with extensive information such as transactions, account information entries etc. The Big O Notation for such a database paradigm drops from O(n) to O(1), greatly increasing efficiency in these specific cases.

STANDARDS

The Object Data Management Group (ODMG) was a consortium of object database and object-relational mapping

vendors, members of the academic community, and interested parties. Its goal was to create a set of specifications that would allow for portable applications that store objects in database management systems. It published several versions of its specification. The last release was ODMG 3.0. By 2001, most of the major object database and object-relational mapping vendors claimed conformance to the ODMG Java Language Binding. Compliance to the other components of the specification was mixed.

In 2001, the ODMG Java Language Binding was submitted to the Java Community Process as a basis for the Java Data Objects specification. The ODMG member companies then decided to concentrate their efforts on the Java Data Objects specification. As a result, the ODMG disbanded in 2001. Many object database ideas were also absorbed into SQL:1999 and have been implemented in varying degrees in object-relational database products. In 2005 Cook, Rai, and Rosenberger proposed to drop all standardization efforts to introduce additional object-oriented query APIs but rather use the OO programming language itself, i.e., Java and .NET, to express queries. As a result, Native Queries emerged.

Similarly, Microsoft announced Language Integrated Query (LINQ) and DLINQ, an implementation of LINQ, in September 2005, to provide close, language-integrated database query capabilities with its programming languages C# and VB.NET 9. In February 2006, the Object Management Group (OMG) announced that they had been granted the right to develop new specifications based on the ODMG 3.0 specification and the formation of the Object Database

Technology Working Group (ODBT WG). The ODBT WG planned to create a set of standards that would incorporate advances in object database technology (e.g., replication), data management (e.g., spatial indexing), and data formats (e.g., XML) and to include new features into these standards that support domains where object databases are being adopted (e.g., real-time systems).

The work of the ODBT WG was suspended in March 2009 when, subsequent to the economic turmoil in late 2008, the ODB vendors involved in this effort decided to focus their resources elsewhere. In January 2007 the World Wide Web Consortium gave final recommendation status to the XQuery language. XQuery uses XML as its data model. Some of the ideas developed originally for object databases found their way into XQuery, but XQuery is not intrinsically object-oriented. Because of the popularity of XML, XQuery engines compete with object databases as a vehicle for storage of data that is too complex or variable to hold conveniently in a relational database.

COMPARISON WITH RDBMSS

An object database stores complex data and relationships between data directly, without mapping to relational rows and columns, and this makes them suitable for applications dealing with very complex data. Objects have a many to many relationship and are accessed by the use of pointers. Pointers are linked to objects to establish relationships. Another benefit of OODBMS is that it can be programmed with small procedural differences without affecting the entire

system. This is most helpful for those organizations that have data relationships that are not entirely clear or need to change these relations to satisfy the new business requirements.

Database Storage Structures

Database tables/indexes are typically stored on hard disk in one of many forms, ordered/unordered Flat files, ISAM, Heaps, Hash buckets or B+ Trees.

These have various advantages and disadvantages discussed in this topic. The most commonly used are B+trees and ISAM.

UNORDERED

Unordered storage typically stores the records in the order they are inserted. While having good insertion efficiency (), it may seem that it would have inefficient retrieval times (), but this is usually never the case as most databases use indexes on the primary keys, resulting in or for keys that are the same as database row offsets within the database file storage system, efficient retrieval times.

ORDERED

Ordered storage typically stores the records in order and may have to rearrange or increase the file size in the case a record is inserted, this is very inefficient. However is better for retrieval as the records are pre-sorted, leading to a complexity of.

STRUCTURED FILES

HEAPS

- simplest and most basic method
  - o insert efficient, records added at end of file – 'chronological' order
  - o retrieval inefficient as searching has to be linear
  - o deletion – deleted records marked requires periodic reorganization if file is very volatile
- advantages
  - o good for bulk loading data
  - o good for relatively small relations as indexing overheads are avoided
  - o good when retrievals involve large proportion of records
- disadvantages
  - o not efficient for selective retrieval using key values, especially if large
  - o sorting may be time-consuming
- not suitable for 'volatile' tables

HASH BUCKETS

- Hash functions calculate the address of the page in which the record is to be stored based on one or more fields in the record
  - o Hashing functions chosen to ensure that addresses are spread evenly across the address space
  - o 'occupancy' is generally 40% – 60% of total file size

  - unique address not guaranteed so collision detection and collision resolution mechanisms are required
- open addressing
- chained/unchained overflow
- pros and cons
  - efficient for exact matches on key field
  - not suitable for range retrieval, which requires sequential storage
  - calculates where the record is stored based on fields in the record
  - hash functions ensure even spread of data
  - collisions are possible, so collision detection and restoration is required

B+ TREES

These are the most used in practice.

- the time taken to access any tuple is the same because same number of nodes searched
- index is a full index so data file does not have to be ordered
- Pros and cons
  - versatile data structure – sequential as well as random access
  - access is fast
  - supports exact, range, part key and pattern matches efficiently
  - 'volatile' files are handled efficiently because index is dynamic – expands and contracts as table grows and shrinks

- o less well suited to relatively stable files – in this case, ISAM is more efficient

Index (Database)

Database index is a data structure that improves the speed of data retrieval operations on a database table at the cost of slower writes and increased storage space. Indexes can be created using one or more columns of a database table, providing the basis for both rapid random lookups and efficient access of ordered records. The disk space required to store the index is typically less than that required by the table (since indices usually contain only the key-fields according to which the table is to be arranged, and exclude all the other details in the table), yielding the possibility to store indices in memory for a table whose data is too large to store in memory. In a relational database, an index is a copy of one part of a table. Some databases extend the power of indexing by allowing indices to be created on functions or expressions. For example, an index could be created on upper(last_name), which would only store the upper case versions of the last_name field in the index. Another option sometimes supported is the use of "filtered" indices, where index entries are created only for those records that satisfy some conditional expression. A further aspect of flexibility is to permit indexing on user-defined functions, as well as expressions formed from an assortment of built-in functions. Indices may be defined as unique or non-unique. A unique index acts as a constraint on the table by preventing duplicate entries in the index and thus the backing table.

INDEX ARCHITECTURE

Index architectures can be classified as clustered or nonclustered.

NON-CLUSTERED

The data is present in random order, but the logical ordering is specified by the index. The data rows may be randomly spread throughout the table. The non-clustered index tree contains the index keys in sorted order, with the leaf level of the index containing the pointer to the page and the row number in the data page. In non-clustered index:

- The physical order of the rows is not the same as the index order.
- Typically created on column used in JOIN, WHERE, and ORDER BY clauses.
- Good for tables whose values may be modified frequently.

Microsoft SQL Server creates non-clustered indices by default when CREATE INDEX command is given. There can be more than one non-clustered index on a database table. There can be as many as 249 nonclustered indexes per table. It also creates a clustered index on a primary key by default.

CLUSTERED

Clustering alters the data block into a certain distinct order to match the index, resulting in the row data being stored in order. Therefore, only one clustered index can be created on a given database table. Clustered indices can

greatly increase overall speed of retrieval, but usually only where the data is accessed sequentially in the same or reverse order of the clustered index, or when a range of items is selected. Since the physical records are in this sort order on disk, the next row item in the sequence is immediately before or after the last one, and so fewer data block reads are required.

The primary feature of a clustered index is therefore the ordering of the physical data rows in accordance with the index blocks that point to them. Some databases separate the data and index blocks into separate files, others put two completely different data blocks within the same physical file(s). Create an object where the physical order of rows is same as the index order of the rows and the bottom(leaf) level of clustered index contains the actual data rows. They are known as "index organized tables" under Oracle database.

COLUMN ORDER

The order in which columns are listed in the index definition is important. It is possible to retrieve a set of row identifiers using only the first indexed column. However, it is not possible or efficient (on most databases) to retrieve the set of row identifiers using only the second or greater indexed column. For example, imagine a phone book that is organized by city first, then by last name, and then by first name. If you are given the city, you can easily extract the list of all phone numbers for that city. However, in this phone book it would be very tedious to find all the phone numbers for a given last name. You would have to look

within each city's section for the entries with that last name. Some databases can do this, others just won't use the index.

APPLICATIONS AND LIMITATIONS

Indices are useful for many applications but come with some limitations. Consider the following SQL statement: SELECT first_name FROM people WHERE last_name = 'Smith';. To process this statement without an index the database software must look at the last_name column on every row in the table (this is known as a full table scan). With an index the database simply follows the B-tree data structure until the Smith entry has been found; this is much less computationally expensive than a full table scan. Consider this SQL statement: SELECT email_address FROM customers WHERE email_address LIKE '%@yahoo.com';. This query would yield an email address for every customer whose email address ends with "@yahoo.com", but even if the email_address column has been indexed the database still must perform a full table scan.

This is because the index is built with the assumption that words go from left to right. With a wildcard at the beginning of the search-term, the database software is unable to use the underlying b-tree data structure (in other words, the WHERE-clause is *not sargable*). This problem can be solved through the addition of another index created on reverse(email_address) and a SQL query like this: SELECT email_address FROM customers WHERE reverse(email_address) LIKE reverse('%@yahoo.com');. This

puts the wild-card at the right-most part of the query (now moc.oohay@%) which the index on reverse(email_address) can satisfy.

TYPES

BITMAP INDEX

A bitmap index is a special kind of index that stores the bulk of its data as bit arrays (bitmaps) and answers most queries by performing bitwise logical operations on these bitmaps. The most commonly used index, such as B+trees, are most efficient if the values it indexes do not repeat or repeat a smaller number of times. In contrast, the bitmap index is designed for cases where the values of a variable repeat very frequently. For example, the gender field in a customer database usually contains two distinct values: male or female. For such variables, the bitmap index can have a significant performance advantage over the commonly used trees.

DENSE INDEX

A dense index in databases is a file with pairs of keys and pointers for every record in the data file. Every key in this file is associated with a particular pointer to *a record* in the sorted data file. In clustered indices with duplicate keys, the dense index points *to the first record* with that key.

SPARSE INDEX

A sparse index in databases is a file with pairs of keys and pointers for every block in the data file. Every key in

this file is associated with a particular pointer *to the block* in the sorted data file. In clustered indices with duplicate keys, the sparse index points *to the lowest search key* in each block. primary key is a sparse index.

REVERSE INDEX

A reverse key index reverses the key value before entering it in the index. E.g., the value 24538 becomes 83542 in the index. Reversing the key value is particularly useful for indexing data such as sequence numbers, where new key values monotonically increase.

INDEX IMPLEMENTATIONS

Indices can be implemented using a variety of data structures. Popular indices include balanced trees, B+ trees, Fractal Tree™ indexes and hashes. In Microsoft SQL Server, the leaf node of the clustered index corresponds to the actual data, not simply a pointer to data that resides elsewhere, as is the case with a non-clustered index. Each relation can have a single clustered index and many unclustered indices.

INDEX CONCURRENCY CONTROL

An index is typically being accessed concurrently by several transactions and processes, and thus needs concurrency control. While in principle indexes can utilize the common database concurrency control methods, specialized concurrency control methods for indexes exist, which are applied in conjunction with the common methods for a substantial performance gain.

COVERING INDEX

In most cases, an index is used to quickly locate the data record(s) from which the required data is read. In other words, the index is only used to locate data records in the table and not to return data. A covering index is a special case where the index itself contains the required data field(s) and can return the data. Consider the following table (other fields omitted):

ID	Name	Other Fields
12	Plug...	
13	Lamp	...
14	Fuse	...

To find the Name for ID 13, an index on (ID) will be useful, but the record must still be read to get the Name. However, an index on (ID, Name) contains the required data field and eliminates the need to look up the record. A covering index can dramatically speed up data retrieval but may itself be large due to the additional keys, which slow down data insertion & update. To reduce such index size, some systems allow non-key fields to be included in the index. Non-key fields are not themselves part of the index ordering but only included at the leaf level, allowing for a covering index with less overall index size.

STANDARDIZATION

There is no standard about creating indexes because the ISO SQL Standard does not cover physical aspects, and indexes are one of the pysical part of databse conception

among others like storage (tablespace or filegroups). However RDBMS vendors all give a CREATE INDEX syntax with some specific options which depends on functionalities they provide to customers.

Database Transaction

A database transaction comprises a unit of work performed within a database management system (or similar system) against a database, and treated in a coherent and reliable way independent of other transactions. Transactions in a database environment have two main purposes:

1. To provide reliable units of work that allow correct recovery from failures and keep a database consistent even in cases of system failure, when execution stops (completely or partially) and many operations upon a database remain uncompleted, with unclear status.
2. To provide isolation between programmes accessing a database concurrently. Without isolation the program's outcomes are possibly erroneous.

A database transaction, by definition, must be atomic, consistent, isolated and durable. Database practitioners often refer to these properties of database transactions using the acronym ACID. Transactions provide an "all-or-nothing" proposition, stating that each work-unit performed in a database must either complete in its entirety or have no effect whatsoever. Further, the system must isolate each transaction from other transactions, results must conform to existing constraints in the database, and transactions that complete successfully must get written to durable storage.

PURPOSE

Databases and other data stores which treat the integrity of data as paramount often include the ability to handle transactions to maintain the integrity of data. A single transaction consists of one or more independent units of work, each reading and/or writing information to a database or other data store. When this happens it is often important to ensure that all such processing leaves the database or data store in a consistent state. Examples from double-entry accounting systems often illustrate the concept of transactions. In double-entry accounting every debit requires the recording of an associated credit. If one writes a check for €100 to buy groceries, a transactional double-entry accounting system must record the following two entries to cover the single transaction:

1. Debit €100 to Groceries Expense Account
2. Credit €100 to Checking Account

A transactional system would make both entries — or both entries would fail. By treating the recording of multiple entries as an atomic transactional unit of work the system maintains the integrity of the data recorded. In other words, nobody ends up with a situation in which a debit is recorded but no associated credit is recorded, or vice versa.

TRANSACTIONAL DATABASES

A '*transactional database* is a DBMS where write transactions on the database are able to be rolled back if they are not completed properly (e.g. due to power or

connectivity loss). Most modern relational database management systems fall into the category of databases that support transactions. In a database system a transaction might consist of one or more data-manipulation statements and queries, each reading and/or writing information in the database. Users of database systems consider consistency and integrity of data as highly important. A simple transaction is usually issued to the database system in a language like SQL wrapped in a transaction, using a pattern similar to the following:

1. Begin the transaction
2. Execute several data manipulations and queries
3. If no errors occur then commit the transaction and end it
4. If errors occur then rollback the transaction and end it

If no errors occurred during the execution of the transaction then the system commits the transaction. A transaction commit operation applies all data manipulations within the scope of the transaction and persists the results to the database.

If an error occurs during the transaction, or if the user specifies a rollback operation, the data manipulations within the transaction are not persisted to the database. In no case can a partial transaction be committed to the database since that would leave the database in an inconsistent state. Internally, multi-user databases store and process transactions, often by using a transaction ID or XID.

IN SQL

SQL is inherently transactional, and a transaction is automatically started when another ends. Some databases extend SQL and implement a START TRANSACTION statement, but while seemingly signifying the start of the transaction it merely deactivates autocommit. The result of any work done after this point will remain invisible to other database-users until the system processes a COMMIT statement. A ROLLBACK statement can also occur, which will undo any work performed since the last transaction. Both COMMIT and ROLLBACK will end the transaction, and start a new. If autocommit was disabled using START TRANSACTION, autocommit will often also be reenabled. Some database systems allow the synonyms BEGIN, BEGIN WORK and BEGIN TRANSACTION, and may have other options available.

DISTRIBUTED TRANSACTIONS

Database systems implement distributed transactions as transactions against multiple applications or hosts. A distributed transaction enforces the ACID properties over multiple systems or data stores, and might include systems such as databases, file systems, messaging systems, and other applications. In a distributed transaction a coordinating service ensures that all parts of the transaction are applied to all relevant systems.

As with database and other transactions, if any part of the transaction fails, the entire transaction is rolled back across all affected systems.

TRANSACTIONAL FILESYSTEMS

The Namesys Reiser4 filesystem for Linux supports transactions, and as of Microsoft Windows Vista, the Microsoft NTFS filesystem supports distributed transactions across networks.

Concurrency Control

In information technology and computer science, especially in the fields of computer programming, operating systems, multiprocessors, and databases, concurrency control ensures that correct results for concurrent operations are generated, while getting those results as quickly as possible. Computer systems, both software and hardware, consist of modules, or components. Each component is designed to operate correctly, i.e., to obey to or meet certain consistency rules. When components that operate concurrently interact by messaging or by sharing accessed data (in memory or storage), a certain component's consistency may be violated by another component.

The general area of concurrency control provides rules, methods, design methodologies, and theories to maintain the consistency of components operating concurrently while interacting, and thus the consistency and correctness of the whole system. Introducing concurrency control into a system means applying operation constraints which typically result in some performance reduction. Operation consistency and correctness should be achieved with as good as possible efficiency, without reducing performance below reasonable.

CONCURRENCY CONTROL IN DATABASES

Comments:

1. This section is applicable to all transactional systems, i.e., to all systems that use *database transactions* (*atomic transactions*; e.g., transactional objects in Systems management and in networks of smartphones which typically implement private, dedicated database systems), not only general-purpose database management systems (DBMSs).

2. DBMSs need to deal also with concurrency control issues not typical just to database transactions but rather to operating systems in general. These issues (e.g., see *Concurrency control in operating systems* below) are out of the scope of this section.

Concurrency control in Database management systems (DBMS; e.g., Bernstein et al. 1987, Weikum and Vossen 2001), other transactional objects, and related distributed applications (e.g., Grid computing and Cloud computing) ensures that *database transactions* are performed concurrently without violating the data integrity of the respective databases. Thus concurrency control is an essential element for correctness in any system where two database transactions or more, executed with time overlap, can access the same data, e.g., virtually in any general-purpose database system. Consequently a vast body of related research has been accumulated since database systems have emerged in the early 1970s. A well established concurrency control theory for database systems is outlined in the references mentioned above: serializability theory,

which allows to effectively design and analyze concurrency control methods and mechanisms. An alternative theory for concurrency control of atomic transactions over abstract data types is presented in (Lynch et al. 1993), and not utilized below. This theory is more refined, with a wider scope, but has been less utilized in the Database literature than the classical theory above. Each theory has its pros and cons, emphasis and insight. To some extent they are complementary, and their merging may be useful.

To ensure correctness, a DBMS usually guarantees that only *serializable* transaction schedules are generated, unless *serializability* is intentionally relaxed to increase performance, but only in cases where application correctness is not harmed. For maintaining correctness in cases of failed (aborted) transactions (which can always happen for many reasons) schedules also need to have the *recoverability* (from abort) property. A DBMS also guarantees that no effect of *committed* transactions is lost, and no effect of *aborted* (rolled back) transactions remains in the related database. Overall transaction characterization is usually summarized by the ACID rules below. As databases have become distributed, or needed to cooperate in distributed environments (e.g., Federated databases in the early 1990, and Cloud computing currently), the effective distribution of concurrency control mechanisms has received special attention.

DATABASE TRANSACTION AND THE ACID RULES

The concept of a *database transaction* (or *atomic transaction*) has evolved in order to enable both a well

understood database system behaviour in a faulty environment where crashes can happen any time, and *recovery* from a crash to a well understood database state. A database transaction is a unit of work, typically encapsulating a number of operations over a database (e.g., reading a database object, writing, acquiring lock, etc.), an abstraction supported in database and also other systems. Each transaction has well defined boundaries in terms of which program/code executions are included in that transaction (determined by the transaction's programmer via special transaction commands). Every database transaction obeys the following rules (by support in the database system; i.e., a database system is designed to guarantee them for the transactions it runs):

- Atomicity - Either the effects of all or none of its operations remain ("all or nothing" semantics) when a transaction is completed (*committed* or *aborted* respectively). In other words, to the outside world a committed transaction appears (by its effects) to be indivisible, atomic, and an aborted transaction does not leave effects at all, as if never existed.
- Consistency - Every transaction must leave the database in a consistent (correct) state, i.e., maintain the predetermined integrity rules of the database (constraints upon and among the database's objects). A transaction must transform a database from one consistent state to another consistent state (it is the responsibility of the transaction's programmer to make sure that the transaction itself is correct, i.e., performs correctly what it intends to perform while maintaining

the integrity rules). Thus since a database can be normally changed only by transactions, all the database's states are consistent. An aborted transaction does not change the state.

- Isolation - Transactions cannot interfere with each other. Moreover, usually the effects of an incomplete transaction are not visible to another transaction. Providing isolation is the main goal of concurrency control.
- Durability - Effects of successful (committed) transactions must persist through crashes (typically by recording the transaction's effects and its commit event in a non-volatile memory).

WHY IS CONCURRENCY CONTROL NEEDED?

If transactions are executed *serially*, i.e., sequentially with no overlap in time, no transaction concurrency exists. However, if concurrent transactions with interleaving operations are allowed in an uncontrolled manner, some unexpected, undesirable result may occur. Here are some typical examples:

1. The lost update problem: A second transaction writes a second value of a data-item (datum) on top of a first value written by a first concurrent transaction, and the first value is lost to other transactions running concurrently which need, by their precedence, to read the first value. The transactions that have read the wrong value end with incorrect results.
2. The dirty read problem: Transactions read a value written by a transaction that has been later aborted.

This value disappears from the database upon abort, and should not have been read by any transaction ("dirty read"). The reading transactions end with incorrect results.

3. The incorrect summary problem: While one transaction takes a summary over the values of all the instances of a repeated data-item, a second transaction updates some instances of that data-item. The resulting summary does not reflect a correct result for any (usually needed for correctness) precedence order between the two transactions (if one is executed before the other), but rather some random result, depending on the timing of the updates, and whether certain update results have been included in the summary or not.

CONCURRENCY CONTROL MECHANISMS

Categories

The main categories of concurrency control mechanisms are:

- Optimistic - Delay the checking of whether a transaction meets the isolation and other integrity rules (e.g., serializability and recoverability) until its end, without blocking any of its (read, write) operations ("...and be optimistic about the rules being met..."), and then abort a transaction to prevent the violation, if the desired rules are to be violated upon its commit. An aborted transaction is immediately restarted and re-executed, which incurs an obvious overhead (versus

executing it to the end only once). If not too many transactions are aborted, then being optimistic is usually a good strategy.

- Pessimistic - Block an operation of a transaction, if it may cause violation of the rules, until the possibility of violation disappears. Blocking operations is typically involved with performance reduction.
- Semi-optimistic - Block operations in some situations, if they may cause violation of some rules, and do not block in other situations while delaying rules checking (if needed) to transaction's end, as done with optimistic.

Different categories provide different performance, i.e., different average transaction completion rates (*throughput*), depending on transaction types mix, computing level of parallelism, and other factors. If selection and knowledge about trade-offs are available, then category and method should be chosen to provide the highest performance. The mutual blocking between two transactions (where each one blocks the other) or more results in a deadlock, where the transactions involved are stalled and cannot reach completion. Most non-optimistic mechanisms (with blocking) are prone to deadlocks which are resolved by an intentional abort of a stalled transaction (which releases the other transactions in that deadlock), and its immediate restart and re-execution. The likelihood of a deadlock is typically low.

Methods

Many methods for concurrency control exist. Most of them can be implemented within either main category above.

The major methods, which have each many variants, and in some cases may overlap or be combined, are:

1. Locking (e.g., Two-phase locking - 2PL) - Controlling access to data by locks assigned to the data. Access of a transaction to a data item (database object) locked by another transaction may be blocked (depending on lock type and access operation type) until lock release.
2. Serialization graph checking (also called Serializability, or Conflict, or Precedence graph checking) - Checking for cycles in the schedule's graph and breaking them by aborts.
3. Timestamp ordering (TO) - Assigning timestamps to transactions, and controlling or checking access to data by timestamp order.
4. Commitment ordering (or Commit ordering; CO) - Controlling or checking transactions' chronological order of commit events to be compatible with their respective precedence order.

Other major concurrency control types that are utilized in conjunction with the methods above include:

- Multiversion concurrency control (MVCC) - Increasing concurrency and performance by generating a new version of a database object each time the object is written, and allowing transactions' read operations of several last relevant versions (of each object) depending on scheduling method.
- Index concurrency control - Synchronizing access operations to indexes, rather than to user data. Specialized methods provide substantial performance gains.

The most common mechanism type in database systems since their early days in the 1970s has been *Strong strict Two-phase locking* (SS2PL; also called *Rigorous scheduling* or *Rigorous 2PL*) which is a special case (variant) of both Two-phase locking (2PL) and Commitment ordering (CO). It is pessimistic. In spite of its long name (for historical reasons) the idea of the SS2PL mechanism is simple: "Release all locks applied by a transaction only after the transaction has ended." SS2PL (or Rigorousness) is also the name of the set of all schedules that can be generated by this mechanism, i.e., these are SS2PL (or Rigorous) schedules, have the SS2PL (or Rigorousness) property.

MAJOR GOALS OF CONCURRENCY CONTROL MECHANISMS

Concurrency control mechanisms firstly need to operate correctly, i.e., to maintain each transaction's integrity rules while transactions are running concurrently, and thus the integrity of the entire transactional system. Correctness needs to be achieved with as good performance as possible. In addition, increasingly a need exists to operate effectively while transactions are distributed over processes, computers, and computer networks. Other subjects that may affect concurrency control are recovery and replication.

Correctness

Serializability

For correctness, a common major goal of most concurrency control mechanisms is generating schedules with the *Serializability* property. Without serializability

undesirable phenomena may occur, e.g., money may disappear from accounts, or be generated from nowhere. Serializability of a schedule means equivalence (in the resulting database values) to some *serial* schedule with the same transactions (i.e., in which transactions are sequential with no overlap in time, and thus completely isolated from each other: No concurrent access by any two transactions to the same data is possible). Serializability is considered the highest level of isolation among database transactions, and the major correctness criterion for concurrent transactions.

In some cases compromised, relaxed forms of serializability are allowed for better performance (e.g., the popular *Snapshot isolation* mechanism) or to meet availability requirements in highly distributed systems, but only if application's correctness is not violated by the relaxation (e.g., no relaxation is allowed for money transactions, since by relaxation money can disappear, or appear from nowhere). Almost all implemented concurrency control mechanisms achieve serializability by providing *Conflict serializablity*, a broad special case of serializability (i.e., it covers, enables most serializable schedules, and does not impose significant additional delay-causing constraints) which can be implemented efficiently.

Recoverability

Comment: While in the general area of systems the term "recoverability" may refer to the ability of a system to recover from failure, within concurrency control of database systems this term has received a specific meaning.

Concurrency control typically also ensures the *Recoverability* property of schedules for maintaining correctness in cases of aborted transactions (which can always happen for many reasons). Recoverability (from abort) means that no committed transaction in a schedule has read data written by an aborted transaction. Such data disappear from the database (upon the abort) and are parts of an incorrect database state. Reading such data violates the consistency rule of ACID. Unlike Serializability, Recoverability cannot be compromised, relaxed at any case, since any relaxation results in quick database integrity violation upon aborts. The major methods listed above provide serializability mechanisms. None of them in its general form automatically provides recoverability, and special considerations and mechanism enhancements are needed to support recoverability. A commonly utilized special case of recoverability is *Strictness*, which allows efficient database recovery from failure (but excludes optimistic implementations; e.g, Strict CO (SCO) cannot have an optimistic implementation, but has semi-optimistic ones).

Comment: Note that the *Recoverability* property is needed even if no database failure occurs and no database *recovery* from failure is needed. It is rather needed to correctly automatically handle transaction aborts, which may be unrelated to database failure and recovery from it.

Distribution

With the fast technological development of computing the difference between local and distributed computing over low latency networks is blurring. Thus the quite effective

utilization of local techniques in such distributed environments is common, e.g., in computer clusters. However for a large-scale distribution local concurrency control techniques typically do not scale well.

Distributed Serializability and Commitment Ordering

As database systems have become distributed, or started to cooperate in distributed environments (e.g., Federated databases in the early 1990s, and nowadays Grid computing, Cloud computing, and networks with smartphones), some transactions have become distributed. A distributed transaction means that the transaction spans processes, and may span computers and geographical sites. This generates a need in effective distributed concurrency control mechanisms. Achieving the Serializability property of a distributed system's schedule effectively poses special challenges typically not met by most of the regular serializability mechanisms, originally designed to operate locally. This is especially due to a need in costly distribution of concurrency control information amid communication and computer latency. The only known general effective technique for distribution is Commitment ordering, which was disclosed publicly in 1991 (after being patented). Commitment ordering (Commit ordering, CO; Raz 1992) means that transactions' chronological order of commit events is kept compatible with their respective precedence order.

CO does not require the distribution of concurrency control information and provides a general effective solution

(reliable, high-performance, and scalable) for both distributed and global serializability, also in a heterogeneous environment with database systems (or other transactional objects) with different (any) concurrency control mechanisms. CO is indifferent to which mechanism is utilized, since it does not interfere with any transaction operation scheduling (which most mechanisms control), and only determines the order of commit events.

Thus, CO enables the efficient distribution of all other mechanisms, and also the distribution of a mix of different (any) local mechanisms, for achieving distributed and global serializability. The existence of such a solution has been considered "unlikely" until 1991, and by many experts also later, due to misunderstanding of the CO solution. An important side-benefit of CO is automatic distributed deadlock resolution. Contrary to CO, virtually all other techniques (when not combined with CO) are prone to distributed deadlocks (also called global deadlocks) which need special handling. CO is also the name of the resulting schedule property: A schedule has the CO property if the chronological order of its transactions' commit events is compatible with the respective transactions' precedence (partial) order.

SS2PL mentioned above is a variant (special case) of CO and thus also effective to achieve distributed and global serializability. It also provides automatic distributed deadlock resolution (a fact overlooked in the research literature even after CO's publication), as well as Strictness and thus Recoverability. Possessing these desired properties together with known efficient locking based implementations explains

SS2PL's popularity. SS2PL has been utilized to efficiently achieve Distributed and Global serializability since the 1980, and has become the de-facto standard for it. However, SS2PL is blocking and constraining (pessimistic), and with the proliferation of distribution and utilization of systems different from traditional database systems (e.g., as in Cloud computing), less constraining types of CO (e.g., Optimistic CO) may be needed for better performance.

Comments

1. The *Distributed conflict serializability* property in its general form is difficult to achieve efficiently, but it is achieved efficiently via its special case *Distributed CO*: Each local component (e.g., a local DBMS) needs both to provide some form of CO, and enforce a special *voting strategy* for the *Two-phase commit protocol* (2PC: utilized to commit distributed transactions). Differently from the general Distributed CO, *Distributed SS2PL* exists automatically when all local components are SS2PL based (in each component CO exists, implied, and the voting strategy is now met automatically). This fact has been known and utilized since the 1980s (i.e., that SS2PL exists globally, without knowing about CO) for efficient Distributed SS2PL, which implies Distributed serializability and strictness (e.g., see Raz 1992, page 293; it is also implied in Bernstein et al. 1987, page 78). Less constrained Distributed serializability and strictness can be efficiently achieved by Distributed Strict CO (SCO), or by a mix of SS2PL based and SCO based local components.

2. About the references and Commitment ordering: (Bernstein et al. 1987) was published before the discovery of CO in 1990. CO is called *Dynamic atomicity* in (Lynch et al. 1993, page 201; see The History of Commitment Ordering). CO is described in (Weikum and Vossen 2001, pages 102, 700), but the description is partial and misses CO's essence. (Raz 1992) was the first refereed and accepted for publication article about CO. Other CO articles followed.

Distributed Recoverability

Unlike Serializability, *Distributed recoverability* and *Distributed strictness* can be achieved efficiently in a straightforward way, similarly to the way Distributed CO is achieved: In each database system they have to be applied locally, and employ a voting strategy for the Two-phase commit protocol (2PC; Raz 1992, page 307).

OTHER SUBJECTS OF ATTENTION

The design of concurrency control mechanisms is often influenced by the following subjects:

Recovery

All systems are prone to failures, and handling *recovery* from failure is a must. The properties of the generated schedules, which are dictated by the concurrency control mechanism, may have an impact on the effectiveness and efficiency of recovery. For example, the Strictness property is often desirable for an efficient recovery.

Replication

For high availability database objects are often *replicated*. Updates of replicas of a same database object need to be kept synchronized. This may affect the way concurrency control is done.

9

Network Security

In the field of networking, the area of network security consists of the provisions and policies adopted by the network administrator to prevent and monitor unauthorized access, misuse, modification, or denial of the computer network and network-accessible resources.

THE FIRST STEP TO INFORMATION SECURITY

The term network security and information security are often used interchangeably. Network security is generally taken as providing protection at the boundaries of an organization by keeping out intruders (hackers). Information security, however, explicitly focuses on protecting data resources from malware attack or simple mistakes by people within an organization by use of data loss prevention (DLP)

techniques. One of these techniques is to compartmentalize large networks with internal boundaries.

NETWORK SECURITY CONCEPTS

Network security starts from authenticating the user, commonly with a username and a password. Since this requires just one thing besides the user name, i.e. the password which is something you 'know', this is sometimes termed one factor authentication. With two factor authentication something you 'have' is also used (e.g. a security token or 'dongle', an ATM card, or your mobile phone), or with three factor authentication something you 'are' is also used (e.g. a fingerprint or retinal scan). Once authenticated, a firewall enforces access policies such as what services are allowed to be accessed by the network users. Though effective to prevent unauthorized access, this component may fail to check potentially harmful content such as computer worms or Trojans being transmitted over the network. Anti-virus software or an intrusion prevention system (IPS) help detect and inhibit the action of such malware.

An anomaly-based intrusion detection system may also monitor the network and traffic for unexpected (i.e. suspicious) content or behaviour and other anomalies to protect resources, e.g. from denial of service attacks or an employee accessing files at strange times. Individual events occurring on the network may be logged for audit purposes and for later high level analysis. Communication between two hosts using a network could be encrypted to maintain

privacy. *Honeypots*, essentially decoy network-accessible resources, could be deployed in a network as surveillance and early-warning tools as the honeypot will not normally be accessed. Techniques used by the attackers that attempt to compromise these decoy resources are studied during and after an attack to keep an eye on new exploitation techniques. Such analysis could be used to further tighten security of the actual network being protected by the honeypot.

SECURITY MANAGEMENT

Security Management for networks is different for all kinds of situations. A small home or an office would only require basic security while large businesses will require high maintenance and advanced software and hardware to prevent malicious attacks from hacking and spamming.

SMALL HOMES

- A basic firewall or a unified threat management system.
- For Windows users, basic Antivirus software. An anti-spyware programme would also be a good idea. There are many other types of antivirus or anti-spyware programmes out there to be considered.
- When using a wireless connection, use a robust password. Also try to use the strongest security supported by your wireless devices, such as WPA2 with AES encryption.
- If using Wireless: Change the default SSID network name, also disable SSID Broadcast; as this function

is unnecessary for home use. (However, many security experts consider this to be relatively useless. http://blogs.zdnet.com/Ou/index.php?p=43)

- Enable MAC Address filtering to keep track of all home network MAC devices connecting to your router.
- Assign STATIC IP addresses to network devices.
- Disable ICMP ping on router.
- Review router or firewall logs to help identify abnormal network connections or traffic to the Internet.
- Use passwords for all accounts.
- Have multiple accounts per family member, using non-administrative accounts for day-to-day activities. Disable the guest account (Control Panel> Administrative Tools> Computer Management> Users).
- Raise awareness about information security to children.

MEDIUM BUSINESSES

- A fairly strong firewall or Unified Threat Management System
- Strong Antivirus software and Internet Security Software.
- For authentication, use strong passwords and change it on a bi-weekly/monthly basis.
- When using a wireless connection, use a robust password.
- Raise awareness about physical security to employees.
- Use an optional network analyzer or network monitor.
- An enlightened administrator or manager.

LARGE BUSINESSES

- A strong firewall and proxy to keep unwanted people out.
- A strong Antivirus software package and Internet Security Software package.
- For authentication, use strong passwords and change it on a weekly/bi-weekly basis.
- When using a wireless connection, use a robust password.
- Exercise physical security precautions to employees.
- Prepare a network analyzer or network monitor and use it when needed.
- Implement physical security management like closed circuit television for entry areas and restricted zones.
- Security fencing to mark the company's perimeter.
- Fire extinguishers for fire-sensitive areas like server rooms and security rooms.
- Security guards can help to maximize security.

SCHOOL

- An adjustable firewall and proxy to allow authorized users access from the outside and inside.
- Strong Antivirus software and Internet Security Software packages.
- Wireless connections that lead to firewalls.
- Children's Internet Protection Act compliance.
- Supervision of network to guarantee updates and changes based on popular site usage.

- Constant supervision by teachers, librarians, and administrators to guarantee protection against attacks by both internet and sneakernet sources.

LARGE GOVERNMENT

- A strong firewall and proxy to keep unwanted people out.
- Strong antivirus software and Internet Security Software suites.
- Strong encryption.
- Whitelist authorized wireless connection, block all else.
- All network hardware is in secure zones.
- All host should be on a private network that is invisible from the outside.
- Put web servers in a DMZ, or a firewall from the outside and from the inside.
- Security fencing to mark perimeter and set wireless range to this.

Proactive Cyber Defence

Proactive Cyber Defence means acting in anticipation to oppose an attack against computers and networks. It represents the dynamic between purely offensive and defensive action; interdicting and disrupting an attack or a threat's preparation to attack, either pre-emptively or in self-defence. Proactive cyber defence will most often require operationalizing upstream security (security from the Cloud) mechanisms of the telecommunications/Internet providers. Some of the compelling reasons for a proactive defence

strategy are about cost and choice. Decisionmakers have few choices after an impact and that all of them are costly. Proactive defence is key to mitigating operational risk.

BACKGROUND

In the Fifth century, B.C., Sun Tzu advocated "foreknowledge" or predictive analysis as part of a winning strategy. He warned that planners must have a precise understanding of the active threat and not "remain ignorant of the enemy's condition." The thread of proactive defence is spun throughout his teachings. Psychiatrist Viktor Frankl was likely the first to use of the term proactive in his 1946 book *Man's Search for Meaning* to distinguish the act of taking responsibility for one's own circumstances rather than attributing one's condition to external factors. Later in 1982, the United States Department of Defense (DoD) used "proactive" as a contrary concept to "reactive' in assessing risk. In the framework of risk management 'proactive" meant taking initiative by acting rather than reacting to threat events. Conversely "reactive" measures respond to a stimulus or past events rather than predicting the event. In military science, then and now considers defence is the science-art of thwarting an attack.

Furthermore doctrine poses that if a party attacks an enemy who is about to attack this could be called active-defence. Defence is also a euphemism for war but does not carry the negative connotation of an offensive war. Usage in this way has broadened the term to include most military issues including offensive, which is implicitly referred to as

active-defence. Politically the concept of national self-defence to counter a war of aggression refers to a defensive war involving pre-emptive offensive strikes and is one possible criterion in the 'Just War Theory'. Proactive defence has moved beyond theory. It has been put into practice in theatres of operation. In 1989, Stephen Covey's *The Seven Habits of Highly Effective People*, published by Free Press, transformed the meaning "to act before a situation becomes a source of confrontation or crisis." From that day "proactive" has been placed in opposition to the words "reactive" or "passive." Cyber is derived from "Cybernetics", a word originally coined by a group of scientists led by Norbert Wiener and made popular by Wiener's book of 1948, *Cybernetics or Control and Communication in the Animal and the Machine*. Cyberspace typically refers to the vast and growing logical domain composed of public and private networks; independently managed networks linked together through the lingua franca of the Internet, the Internet Protocol (IP).

The definition of Cyberspace has been extended to include all network-space which at some point, through some path, may have eventual access to the public internet. Under this definition, cyberspace becomes virtually every networked device in the world, which is not devoid of a network interface entirely. There is no air-gap anymore between networks. The origins of cyber defence undoubtedly evolved from the original purpose of the Internet which was to harden military networks against the threat of a nuclear strike. Later cyber defence was coveted by the tenets of information warfare and information operations. The rapid evolution of

information warfare operations doctrine in the 1990's embraced a proactive pre-emptive cyber defence strategy. "Information Warfare is an emergent reality that comes from a self-organization process that has never seen before. The problem is that we talk about it using terms that have well known connotations. And it is difficult to talk about something completely new using words that bring with them specific understanding and expectancies. The early period of the automobile faced a similar situation. At one time it was called a "horseless carriage" as this was the only way to define its essential quality. The car is more than a carriage without a horse. This is the dilemma we face when we discuss Information Warfare.

The danger is that the uses of familiar words misrepresent and mask the true extend of the revolution that will have to take place if we are to be able to retain a military capacity in a new physical, social and cognitive space." - Dr. Garigue, 1994. The National Strategy to Secure Cyberspace was published in February 2003 to outline an initial framework for both organizing and prioritizing efforts to secure the cyberspace. It highlighted the necessity for public private partnerships. Proactive threads include the call to deter malicious activity and prevent cyber attacks against America's critical infrastructures. The hype-cycle of discussion reached its peak in 1994. Present-day proactive cyber defence strategy was conceived within the context of the rich discussion that preceded it, existing doctrine and real proactive cyber defence programmes that have evolved globally over the past decade. Dr. Robert John Garigue, a computational epistemologist and father of information

warfare in Canada, published Information Warfare, Developing a Conceptual Framework. This was a landmark document in 1994 and genesis for proactive cyber defensive theory in Canada.

"Effective cyber defenses ideally prevent an incident from taking place. Any other approach is simply reactive. FedCIRC, the NIPC, the NSIRC, the Department of Defense and industry components realize that the best [action] is a pre-emptive and proactive approach." - Sallie McDonald, the Assistant Commissioner for the Office Of Information Assurance and Critical Infrastructure Protection, Federal Technology Service and General Services Administration; in offering testimony with regard to the National Infrastructure Protection Center (NIPC) and the Federal Computer Incident Response Center or FedCIRC; before The Subcommittee on Terrorism Technology and Government Information Committee on Judiciary and the United States Senate July 25, 2001. The notion of a Proactive Pre-emptive Operations Group (P2OG) emerged from a report of the Defense Science Board (DSB), 2002 briefing. The briefing was reported by Dan Dupont in Inside the Pentagon on September 26, 2002 and was also discussed by William M. Arkin in the Los Angeles Times on October 27, 2002. The Los Angeles Times has subsequently quoted US Secretary of Defence Donald Rumsfeld revealing the creation of the 'Proactive, Pre-emptive Operations Group.' The mission of the P2OG is reportedly to conduct Aggressive, Proactive, Pre-emptive Operations to interdiction and disruption the threat using: Psychological operations, Managed Information Dissemination, Precision Targeting, Information Warfare Operations, and SIGINT...

The proactive defence strategy is meant to improves information collection by stimulating reactions of the threat agents, provide strike options and to enhance operational preparation of the real or virtual battle space. The P2OG has been recommended to be constituted of "one hundred 'highly specialized people with unique technical and intelligence skills such as information operations, PSYOPS, network attack, covert activities, SIGINT, HUMINT, SOF, influence warfare/deception operations and to report to the National Security Council with an annual budget of $100 million." The group would be overseen by the White House's deputy national security adviser and would carry out missions coordinated by the secretary of defense or the CIA director. "The proposal is the latest sign of a new assertiveness by the Defense Department in intelligence matters, and an indication that the cutting edge of intelligence reform is not to be found in Congress but behind closed doors in the Pentagon." - Steven Aftergood of the Federation of American Scientists. DoD doctrinally would initiate a 'pre-emptive' attack on the basis of evidence that an enemy attack is imminent. Proactive measures, according to DoD are those actions taken directly against the preventive stage of an attack by the enemy. Strike back doctrine aligns with pre-emptive and counter-attack tactics of a proactive cyber defence strategy. The notion of 'proactive defence' has a rich history. The hype of 'Proactive cyber defence' reached its zenith around 1994. This period was marked by intense 'hype' discussions under the auspices of Information Warfare. Much of the current doctrine related to proactive cyber defence was fully developed by 1995.

A number of programmes were initiated then, and advanced to full operation by 2005 including those of hostile states. Meanwhile the public discussions diminished until the most recent resurgence in proactive cyber defence 2004-2008. Now most of the discussions around proactive defence in the literature are much less 'proactive' than the earlier discussions in 1994 or existing operational programmes. 'Proactive' is often used to hype marketing of security products or programmes, in much the same way that 'extreme' or 'quality' adjectives have been misused.

Security Architecture

Security provided by IT Systems can be defined as the IT system's ability to be able to protect confidentiality and integrity of processed data, as well as to be able to provide availability of the system and data. "IT Architecture" may be defined as a set of design artifacts, that are relevant for describing an object such that it can be produced to requirements (quality) as well as maintained over the period of its useful life (change). The design artifact describe the structure of components, their inter-relationships, and the principles and guidelines governing their design and evolution over time. Consequently the definition of "IT Security Architecture" may be considered as:

> The design artifacts that describe how the security controls (= security countermeasures) are positioned and how they relate to the overall IT Architecture. These controls serve the purpose to maintain the system's quality attributes, among them confidentiality, integrity and availability.

Security qualities are often considered as Non-functional requirements when systems are designed. In other words they are not required for the system to meet its functional

goals such as processing financial transactions, but are needed for a given level of assurance that the system will perform to meet the functional requirements that have been defined. In recent years there has been a trend towards a hierarchy of control objectives, controls and specific technical implementations of controls, which are implemented within a given security architecture in order to meet the security requirements.