

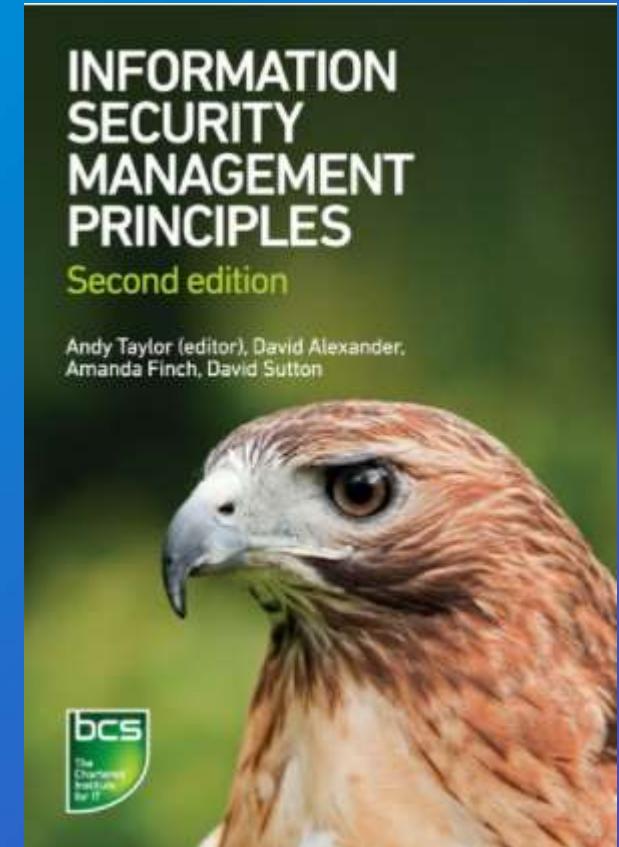


الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432

Introduction to Cyber Security and Digital Crime

Information Security Management Principles

by David Alexander, David Sutton, Andy Taylor, Amanda Finch



Week 1

Chapter 1: INFORMATION SECURITY PRINCIPLES



Contents

1. Concepts and definitions
2. The need for, and benefits of, Information security



Objectives

- Define and explain the Information security policy concepts.
- Explain and justify the importance of information security as part of a business model and to describe its appropriate use as applicable.
- Understand the effect of the rapidly changing business environment.
- Explain the balancing cost and impact of security with the reduction in risk.
- Understand the role of the policy, standards, guidelines and procedures documentation.
- Define corporate governance and related areas of risk management.



Required Reading

1. Chapter 1 from Information Security Management Principles.



Concepts and definitions



Concepts and definitions

- Information assurance management has its own language.
- It is limited in scope and complexity to allow the entire business population to appreciate the concepts with little difficulty.
- Definitions are taken from the General Information Assurance Products and Services Initiative (GIPSI).
- GIPSI took the definitions from:
 - BS ISO / IEC 27001: 2005 where the definition exists,
 - other ISO standards where there was no 27001 definition
 - SC27 or SD6 where ISO standards do not provide any definition.



Confidentiality

- Property that information is not made available or disclosed to unauthorized persons, entities or processes (ISO 27001).
- The information is only applicable to a limited number of people because of its nature, its content or because its wider dissemination will lead to undesirable effects, in particular legal or financial penalties, or inconvenience to one either party.
- Restricting access to information to those with a “need to know” is good practice and is based on the principle of confidentiality.
- Controls to ensure confidentiality are an important part of the larger aspects of managing information assurance.



Integrity

- The property of safeguarding the accuracy and completeness of assets (ISO 27001).
- Information is only useful if it is complete and accurate, and so on.
- Maintaining these aspects of information (its integrity) is often critical and ensuring that only certain people have the appropriate authority to modify, update or delete information is another basic tenet of information assurance.



Availability

- The property of being accessible and usable on demand by an authorized entity (ISO 27001).
- Information that is not available when and as required is not information at all, but irrelevant data.
- Availability is an area where technological developments have dramatically increased the challenges for the information assurance professional.
- The information could be locked in a very secure vault in one form or another and never be allowed to access it - almost perfect assurance but, of course, totally impractical.



Assets and asset types....1

- Asset is anything that has value to the organization, its business operations and its continuity (ISO 27001).
- Assets come in as wide a variety of types as the mechanisms for using them.
- In information assurance, three main types of assets are considered, although the sub-categories that fall under each of these main types can be numerous:
 - Pure information (in whatever format).
 - physical assets such as buildings and computer systems.
 - software used to process or otherwise manage information.



Assets and asset types....2

- When assets are factored into any aspect of information assurance, the impact on these three types of assets should be considered.
- The value of an asset is typically calculated through a Business Impact Assessment, which estimates the cost or value of its loss or unavailability to the business.
- Other aspects to consider including, but not limited to:
 - The value to a competitor.
 - The cost of recovery or reconstruction.
 - The damage to other operations.
 - The impact on such intangibles as reputation, brand awareness and customer loyalty.



Threat, vulnerability, risk and impact....1

- Threat:

- A potential cause of an incident that may result in harm to a system or organization (ISO 27002).
- A threat is something that can happen and that can have undesirable consequences.
- Example: If we see clouds in the sky that look big and dark, we are talking about the threat of rain.
- Threats for one organization may well be opportunities for another - It all depends a lot on the point of view, the environment and the situation under consideration.



Threat, vulnerability, risk and impact....2

- Vulnerability:

- A weakness of an asset or group of assets that can be exploited by one or more threats (ISO 27002).
- A vulnerability is a weakness, something that, if exploited, could cause some unwanted effect(s).
- Example: if someone were to venture into the cloudy environment without an umbrella, it could be considered a vulnerability.



Threat, vulnerability, risk and impact....3

- Risk:

- The potential that a given threat will exploit vulnerabilities of an asset or group of assets and thereby cause harm to the organization (ISO 27002).
- Example: If there is a threat (of rain) and a vulnerability (of not carrying an umbrella), then there is a risk that the affected person will get wet and damage their expensive clothes.
- It is also important to recognize that there can sometimes be a combination of circumstances which also lead to additional and more serious risks:
 - the delay in attending an appointment, combined with a number of other similar events, could result in termination of employment.



Threat, vulnerability, risk and impact....4

- Impact:

- The result of an information security incident, caused by a threat, which affects assets (ISO 27005).
- Perhaps the most important concept of all to grasp is the impact of risk that actually occurs.
- It is the potential impact that must be considered and managed in information assurance.
- If the impact is small and insignificant, then it may be quite appropriate to accept the risk and take no action other than to monitor it.
- When it comes to businesses, the impact on the organization and its day-to-day operations is usually the critical consideration and will often justify taking additional action.



Information security policy concepts

- Every organization should have a policy for its management of information assurance.
 - Recognize the business risks resulting from poor information assurance and take appropriate action to address them.
 - It should include statements clearly indicating that the organization considers the risk to be a serious problem, which should be discussed at all appropriate meetings.
- It is common for organizations to form an Information Assurance Working Group to lead the activities necessary to ensure appropriate levels of assurance within the organization.



The purpose of controls

- Controls, in the sense of information assurance, are activities that are undertaken to manage identified risks.
- Eliminate: Risk avoidance – Decision not to get involved or action to withdraw from a risk situation (ISO Guide 73).
- Reduce: Risk reduction – Action taken to reduce the likelihood, negative consequences, or both, associated with the risk (ISO Guide 73).
- Transfer: Risk transfer – Share the burden of the loss, or the benefit of the gain, with another party for a risk (ISO Guide 73).
- Accept: Risk acceptance – Decision to accept a risk (ISO Guide 73).



Identity, authentication and authorization

- Identity: Properties of an individual or resource that can be used to uniquely identify an individual or resource (Authors).
- Authentication: Ensuring that the identity of a subject or resource is the one claimed (Authors derived from Authenticity in ISO 13335).
- Authorization: The process of verifying the authentication of an individual or resource to establish and confirm their authorized use or access to information or other assets (Authors)



Accountability, audit and compliance....1

- Accountability is the responsibility for actions and processes (authors):
 - when an action is performed on an information system or as part of the information assurance management system, a person must be responsible for this action.
- Auditing is a formal or informal examination of actions, processes, policies and procedures (authors):
 - it is the verification (formal or informal) of a system's records to ensure that the activities that were planned actually took place.



Accountability, audit and compliance....2

- Compliance operates in accordance with established actions, processes, policies and procedures without necessarily having independent reviews (authors).
- Ensuring that a system or process conforms to defined or expected operational procedures is compliance.
- This could cover a major operation:
 - An entire organization conforming to a recognized national standard for information assurance.
 - It could be much more limited with just certain aspects of the operation.
 - Individual users of a specific system, being compliant.
- In general, compliance must be independently audited to obtain certification against a standard, legal or regulatory framework



Information security professionalism and ethics....2

- The core of all assurance is trust and without it it is impossible to function in the world as it is today.
- The degree of trust is where there is leeway and it is often the degree of trust in staff, customers, suppliers, shareholders, etc. which will determine the measures to be put in place.
- It is crucial that the trust placed in information assurance professionals is by no means misplaced.



Information security professionalism and ethics....1

- The general awareness of the work performed by information assurance professionals (as opposed to IT security professionals) is gradually increasing as organizations become more and more complex.
- People are the most important asset of an organization and must effectively use the information an organization holds.
- The Institute of Information Security Professionals (IISP) has developed a competency framework for the information assurance professional.
- An information assurance professional will inevitably become a stakeholder in some of the most important information an organization might hold.



Information security management system (ISMS) concepts

- Part of the overall management system, based on a business risk approach, to establish, implement, operate, monitor, review, maintain and improve information security (ISO 27001).
- The main principle of ISMS is that there should be a “one stop shop” for all information relevant to information assurance within an organization.
- It is essential that organizations make their information as freely and readily available as possible, convenient and necessary and this also applies to the security regulations that control them.



The national and international security standards....1

- Information assurance is the subject of several international and national standards and these should be taken into account when considering for the examination.
- The questions asked in the examination will never be specific to a standard, but will be generic to all best practices, where applicable.
- Knowledge of the appropriate standards required for the examination is therefore limited to a general understanding of the principles involved as long as they reflect best practice.



The national and international security standards....2

- Other aspect of this: When an information assurance professional works in an organization to provide a secure and efficient information management system, the relevant standards should always be considered as the achievable goal for that system.
- Whether it is necessary to achieve simple compliance or to take an extra step to achieve certification is an arbitrary decision often based on other factors.
- It is considered good practice to base an effective information assurance management system on the principles of the relevant standards.
- The use of an internationally accepted standard such as the ISO/IEC 27000 series makes sense in the global nature of operations today.



The need for, and benefits of, Information security



The need for, and benefits of, Information security

- Every business has information that is essential to its continued smooth operation.
- Taking care of this information properly is not free but comes at a price which can be, in some circumstances, very important.
- It is essential that information assurance professionals are able to justify their recommendations for appropriate security measures in a reasonable but pragmatic manner, which must take into account the specific environment in which the business is based.



The importance of information security as part of a business model...1

- Information security:

- Preservation of confidentiality, integrity and availability of information; in addition, other properties such as authenticity, accountability, non-repudiation and reliability can also be involved (ISO 27001)
- Neither information nor assurance operate in a vacuum.
- Both must consider the environment in which they operate and resolve the issues that environment causes.
- It is essential that any information assurance system be firmly rooted in the business world. This means that information assurance is not an issue for the IT manager or security manager, but for the entire organization.
- As soon as one part of the organization is given the task of managing insurance, the rest of the organization will care less.
- All staff in an organization, regardless of nature, activity, location or any other factor, should be concerned with information assurance.



The importance of information security as part of a business model...1

- Information assurance (IA):

- Confidence that information systems will protect the information they carry and perform as they need it, when they need it, under the control of legitimate users (UK Cabinet Office).
- Physical, technical and administrative controls are needed to accomplish these tasks.
- Although primarily focused on information in digital form, the full spectrum of AI encompasses not only digital form, but also analog or physical form.
- These protections apply to data in transit, in physical and electronic form, as well as to data at rest in various types of physical and electronic storage facilities.
- Information systems include any means of storing, processing or disseminating information, including computer systems, media and paper systems.



Different business models and their impact on security

- Over the past 30 years, the business world has changed dramatically - perhaps more than in the previous 50 or 100 years.
- One of the main reasons for this is the increased use of technology which has made it possible to do business remotely rather than in person.
- One of the consequences of this is that more people are able to conduct business transactions on their own rather than expecting others to act as intermediaries.
- All of these transactions and many more can be done directly with the supplier (often over the internet) or with a merchant in another part of the country or the world who may offer a better deal.
- As global organizations now move highly sensitive information or other assets around the world at all times, the need to ensure this is done securely and with proof of receipt, integrity and authority has also increased.



The effect of the rapidly changing business environment

- Dominant factor in society today: It is change, continuing change, inevitable change,
- This quote is from Isaac Asimov and it is now understood that for a business to survive in today's climate of change, it must adapt and be able to adapt quickly:
 - What was acceptable as a business practice last week may no longer be acceptable this week.
- Any insurance system put in place must reflect this changing climate and be flexible enough to deal with it:
 - This does not mean that the assurance can be relaxed or reduced in any way.
 - On the contrary, flexibility should produce a higher level of security and the assurance that risks are managed effectively.



Balancing cost and impact of security with the reduction in risk

- Life can never be risk-free: it is often considered that life is all about risk and its effective management.
- The steps taken in an organization to reduce risk to an acceptable level can sometimes become excessively expensive.
- A fair balance must be struck between the cost or business impact of a risk if it occurs and the cost of actions taken to reduce its likelihood or impact.
- A typical example is insurance:
 - An insurance policy can help offset the cost of the occurrence of a risk by providing the necessary financial support to be used to deal with the occurrence of a risk.
 - If the cost of the insurance policy is too high, it may simply be cheaper to accept that the risk may occur and pay the lower amount to deal with its consequences.
 - It should also be remembered that whilst it may be possible to transfer part of the impact of the occurrence of a risk to a third party.



Information security as part of company policy....1

- Assurance is not an add-on: It is not possible to adequately treat insurance by viewing it as an additional expense to be avoided as much as possible.
- The most effective way to deal with it is to include it from the beginning, in all areas of the organization.
- To this end, the inclusion of assurance as part of the operational policy of the organization is the only cost-effective way of covering the issues adequately.



Information security as part of company policy....2

- There are clear similarities between information assurance and health and safety issues:
 - As soon as health and safety is seen as the concern of one person (that of the health and safety manager), the battle for a safe working environment is lost.
 - Assurance is not the concern solely of the information security manager, but of the whole organization.
- It is essential also that this involvement is from the top of the organization to the bottom.
- Senior management has a critical role to play in ensuring that it creates a work environment where information assurance is the norm and is accepted by all.



Policy, standards, guidelines and procedures documentation

- It needs to be fully supported by a series of other documents covering expected standards, guidelines on how to do things right and procedures on what needs to be done to maintain assurance of the information in question.
- This documentation should be complete in its cover, should be written in a style understandable to the intended audience.
- Example: Procedures are also required for the management of physical assets such as filing cabinets including how they should be erased prior to disposal to avoid inadvertent inclusion of a confidential file for the Filing Cabinet Market.



Corporate governance and related areas of risk management

- In recent years, the advent of some high-profile commercial criminal investigations has resulted in much stricter and invasive legislation regarding risk-taking in businesses.
- It is no longer effective or acceptable (even if it ever was) to delegate the responsibility for risk management down to the manager of the IT section.
- The successful implementation of effective information assurance must be at the heart of all organizations, regardless of their industry, size or activity.
- Properly implemented, secure information management can provide assurance that risk is being managed effectively in at least one area and can form a solid basis for further risk management in related areas.
- If all information is covered by the implemented measures, then financial, operational, intellectual property rights and a host of other areas of risk can be managed through the establishment of a single framework.



Security as an enabler

- In the information economy we all live in today, the cost of information loss, corruption, unavailability or unauthorized dissemination can be very high.
- The effective implementation of information assurance measures can have a very beneficial effect on the potential costs of such events.
- It is easy to develop a compelling business case for effective information management through the use of an approved standard and associated processes.
- The use of appropriate countermeasures and contingency plans can also have the very beneficial effect of making the work done by an organization much more orderly by being based on best working practices.



The role of information security in countering hi-tech and other crime

- Crime is steadily rising and growing, often a little faster than the law enforcement agencies established to fight it.
- The high-tech industry (spanning computers, internet, digitization, communications and related fields) over the past 30 years or so has provided criminals with ever-increasing opportunities for more advanced and profitable crime in a world.
- Some crimes are the old ones that had actually been taken out of the criminals handbook.
- Example: Fraud, which has been hit hard by the introduction of sophisticated security features in banknotes, passports and the like.
- With the ever increasing use of the Internet, it has now come back with much more “efficiency”.
- The growth of these crimes has increased the importance of forensic investigations and in particular the requirement to preserve evidence based on computer systems.



Summary

- The Concepts and definitions of the information security principles are presented.
- The business model is based on the importance of information security.
- The variety of business models and the quick changing business environment have an impact to the information security.
- A fair balance must be struck between the cost or business impact of a risk if it occurs and the cost of actions taken to reduce its likelihood or impact.



Main Reference

1. Chapter 1 from Information Security Management Principles.

Additional References

1. Chapter 1 from Information Security: Principles and Practice, 2nd Edition
2. The cyber security principles. <https://www.cyber.gov.au/acsc/view-all-content/guidance/cyber-security-principles>



Thank You



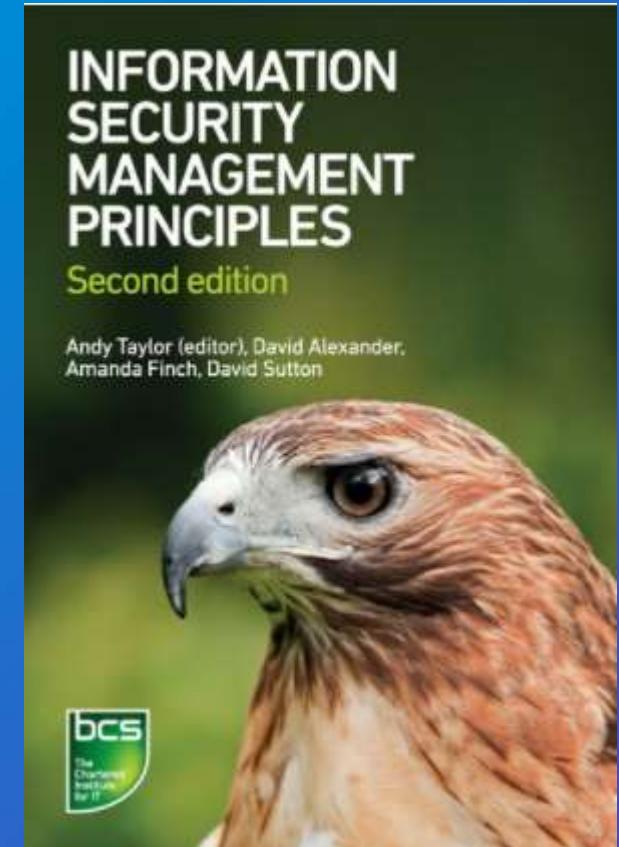


الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432

Introduction to Cyber Security and Digital Crime

Information Security Management Principles

by David Alexander, David Sutton, Andy Taylor, Amanda Finch



Week 2

Chapter 2: INFORMATION RISK



Contents

1. Threats
2. Vulnerabilities
3. Assets
4. Impact
5. Risk management
6. Information classification policies
7. Assessing the risks in business terms



Objectives

- Define and explain each of the key concepts of information risk management and have a thorough understanding of the terminology used.
- Understand the overall process of risk management, and the appropriate use of controls to enable them to manage risk in a cost-effective and appropriate manner for their organization.



Required Reading

1. Chapter 2 from Information Security Management Principles.



THREATS TO, AND VULNERABILITIES OF, INFORMATION SYSTEMS



Threats

- A threat is something that may happen, and that may cause some undesirable consequence.
- Example: a feasible threat is that an unauthorized person might discover your username and password to a system or service.
- Threats must be realistic: Records of such incidents to support the validity of the threat.
- What might be a threat to one person may well be an opportunity to another.



Threat categorization....1

- Threats can be categorized into two main areas :
 - Accidental threats
 - Deliberate threats.
- Each of these areas may contain two further choices:
 - Internal threats
 - External threats.
- In information assurance, accidental threats include a number of conditions, such as human error, system malfunctions, fire and floods.



Threat categorization....2

- Deliberate threats occur when someone sets out with every intention of carrying out the threat.
- This type of threat includes :
 - Hacking
 - Malicious software
 - Sabotage
 - Cyber terrorism
 - Hi-tech crime



Threat categorization....3

- Insider threats originate from within the organization itself, or from business partners and vendors who have some degree of access to the organization's information systems environment.
- Sources of internal threats include permanent and contract employees, trusted partners and managed-service organizations.
- External threats arise from outside the organization and its less closely linked business partners and suppliers.
- Typical external threats may arise from hackers (of various kinds), competitors and protest groups.
- Hazards can be either internal or external in origin (A fire or flood may originate within the organization's building or may originate from outside)



Vulnerabilities

- A vulnerability is a weakness; something which, if exploited, could lead to undesirable consequences.
- Example: Write a password on a Post-It note and stick it underneath the computer's keyboard.
- Whether or not a vulnerability might be exploited will depend on likelihood or probability.
- Example: Software packages and operating systems which are most vulnerable to attack as they present a more easily available or inviting target for malicious software writers and hackers.



Vulnerability categorization

- Vulnerabilities fall into two distinct categories:
 - general vulnerabilities including basic weaknesses in software (including poor design), hardware, buildings or facilities, people, processes and procedures.
 - information-specific vulnerabilities including such areas as unsecured computers (including personal computers, hand-held devices and memory sticks), unsecured servers, unpatched operating systems and applications, unsecured network boundary devices, unsecured wireless systems, unsecured web servers, unsecured email systems, unlocked filing cabinets, etc.
- Threats are said to take advantage of, or exploit, vulnerabilities in order to succeed in achieving their goal.



Assets

- An information asset can vary considerably in form. It can be a/an:
 - System
 - Database
 - Building.
 - Intellectual property
 - Business service
 - Organization's brand
 - Reputation of the organization's chief executive.
- What is important about assets is that if they are lost, stolen or damaged in any way, the organization will almost certainly suffer as a result, and if that damage is sufficiently serious, it might never recover.



Impact....1

- The impact (or potential impact) of the risk is an important concept.
- It is usually this potential impact that has to be considered and managed in information assurance.
- If the impact is small and insignificant, then it may be entirely appropriate to accept the risk and to take no further action other than to monitor it periodically.
- Example: Failure of ‘hole-in-the-wall’ cash dispensers:
 - if just one machine in the bank’s network fails, the impact would generally be very low.



Impact....2

- While the potential impact could be the loss of vital business information, more appropriate countermeasures should be considered.
- When it comes to businesses, the impact on the organization and its day-to-day operations is usually the critical consideration and will often justify taking further action.
- The business impacts of realized threats include loss of confidentiality, integrity and availability, and they frequently result in financial loss, inability to negotiate, brand damage, loss of customer trust, etc.



Likelihood or probability....1

- Some things are very likely to happen, while some are very unlikely to happen. Most others lie somewhere in the grey area in between.
- It is generally accepted that the greater the vulnerability, the more likely an incident is to take place – meaning the threat is carried out.
- There are two basic ways in which likelihood can be assessed:
 - Quantitatively - there will be clear metrics to calculate the likelihood. These may be derived from previously recorded information including statistical data.
 - Qualitatively – the work is more subjective and relies on opinions rather than facts.



Likelihood or probability....1

- Example: companies that produce anti-virus software are able to point to the large number of viruses that their products can scan for and remove, from which one can conclude that without anti-virus software, the risk of infection is high.
- On the other hand, one does not need to know the exact number of incidents to be aware that, without proper password protection, the likelihood of a breach of confidentiality or integrity is high.
- Both methods of assessment (quantitatively, qualitatively) have their place – the important thing is that likelihood assessments are carried out according to agreed criteria.



Risk

- The result of having a vulnerability, which is exploited by a threat, results in an impact or consequence.
- The assessment of the risk for any particular threat is considered to be a combination of the impact and the likelihood that the threat can be carried out.
- Sometimes there may be a combination of circumstances that lead to further, more serious risks as well.
- For an unauthorized person to discover the username and password combination is one thing. If the files include a list of other usernames and their passwords, this would lead to further (and potentially more serious) security breaches.



RISK MANAGEMENT

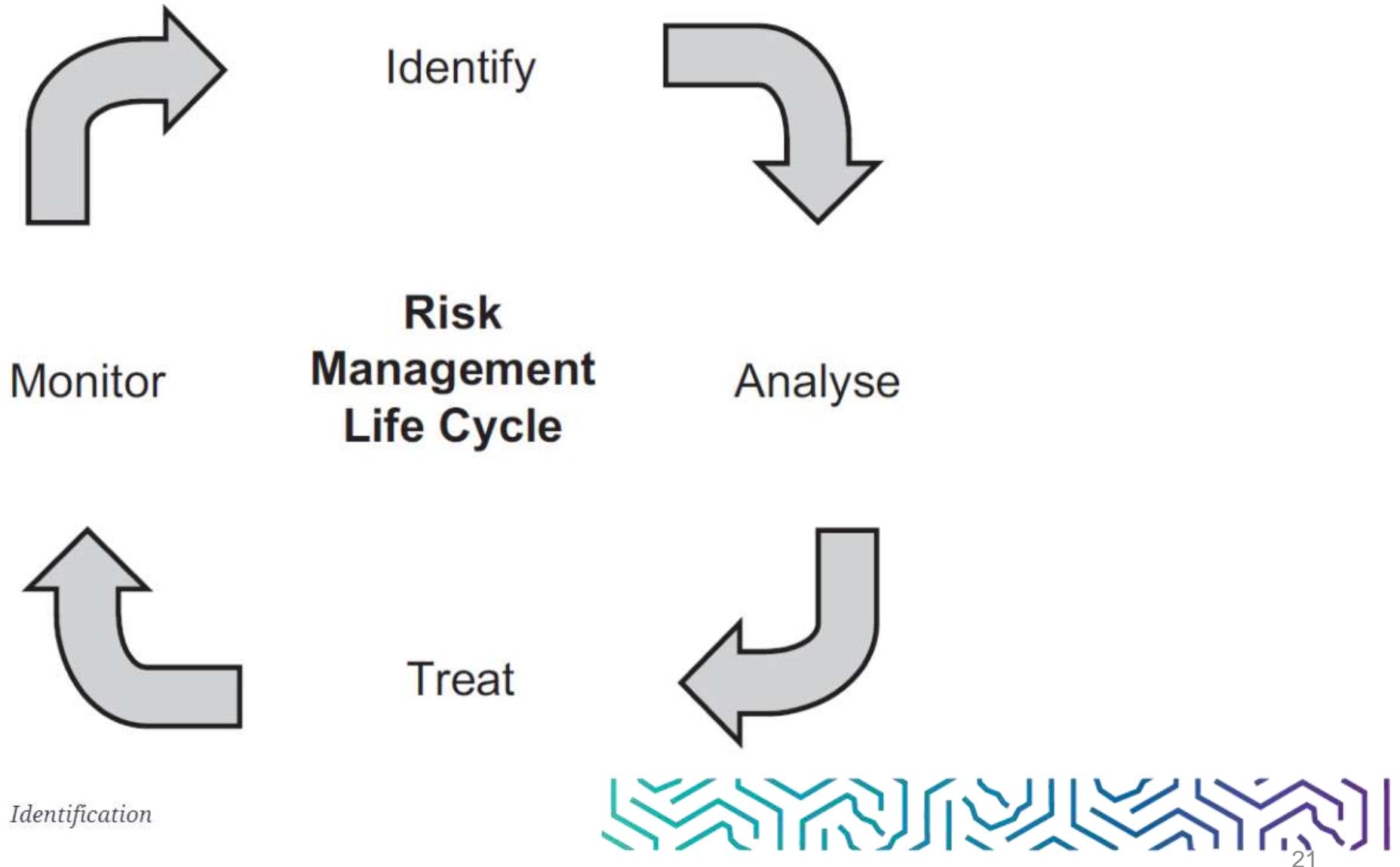


Risk management process....1

- Risk management consists of four distinct areas:
 - Identification of the threats.
 - Impact analysis and risk assessment.
 - Treatment of the risks.
 - Monitoring of the results.
- Risk assessments may take place at a number of levels, for example across a corporation, a business system or process, or a physical location.
- While these are somewhat different types of risk assessment, the way in which they are conducted and the way in which the results will be used are essentially the same.



Risk management process....2



Identification

- This should be carried out in conjunction with an understanding of any known vulnerabilities.
- Example: if the assessment is looking at the threat of possible hacking attacks on a web server, operating system and web server software vulnerabilities should be considered.
- Sometimes this will result in the identification of more than one threat, whilst at other times it will become clear that a number of different vulnerabilities will all be covered by a single threat.
- Once each threat has been identified (often more will appear during the process of the work), it should be considered in the light of its impact on the asset concerned.
- An alternative approach might be to start with a list of the assets that are critical to the organization, and then to determine the potential threats to those assets.

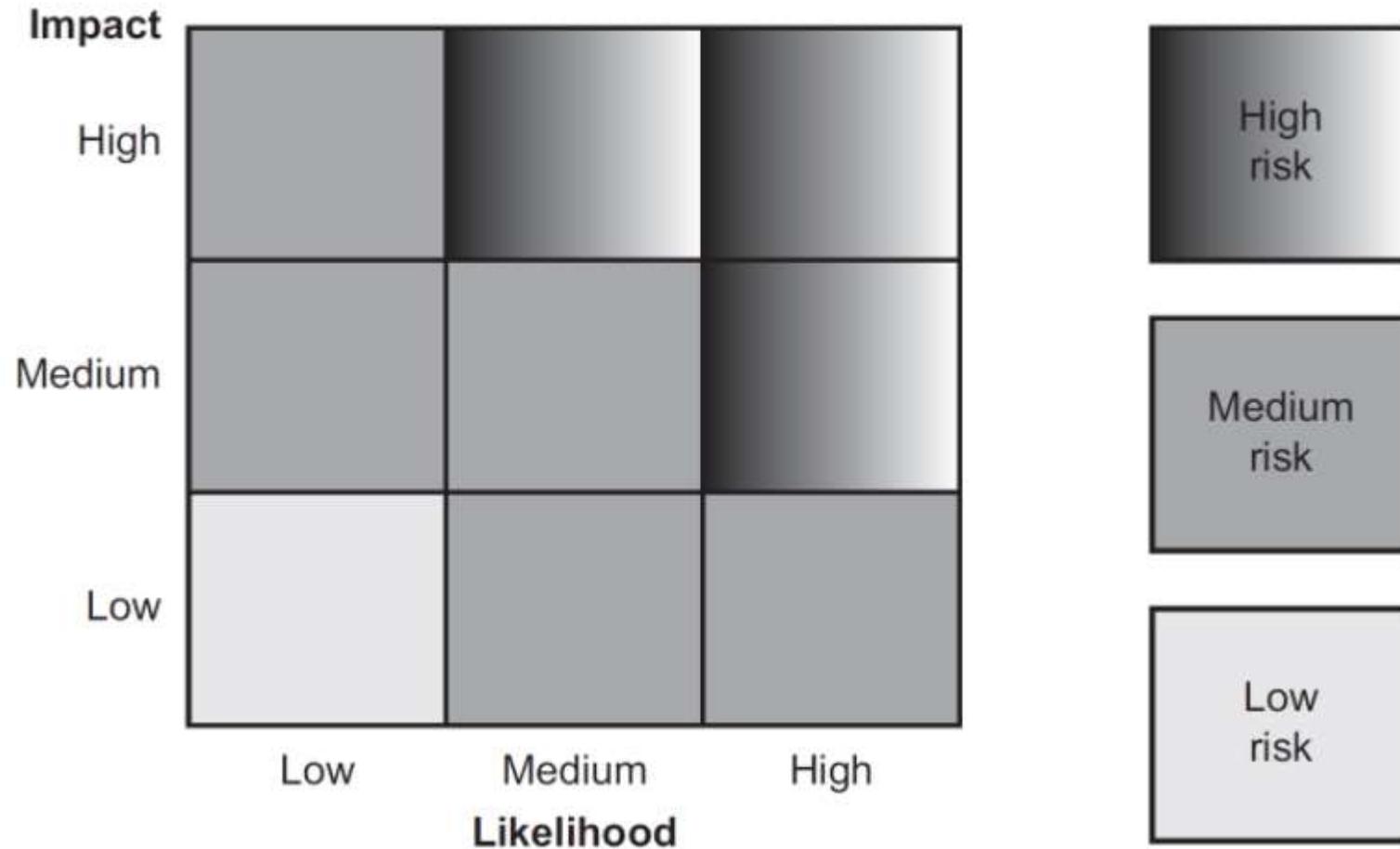


Analysis

- Having identified the impact (or impacts) for each threat, the next task is to assess the likelihood of it occurring.
- It is tempting at this point to assume that, because the system might be fully up to date with its security patches, there is a low likelihood of a threat being realized.
- It must be remembered that this is ongoing work and that, if the patching falls behind, the likelihood of an attack being successful will increase.
- It should also be remembered that new vulnerabilities are continually being discovered.



A typical risk matrix



A typical risk matrix....1

- The matrix can be drawn in many ways:
 - Three-by-three matrix (simplest form) with high, medium and low ratings for both impact and likelihood, as shown above.
 - Five-by-five matrices are very common and provide a greater level of granularity in the results.
- There is no reason why the matrix cannot contain more ratings for either axis and, equally, there is no reason to have the same number of ratings for each.
- It is entirely up to the organization or the individual to decide what size and shape of matrix is appropriate.
- Once a particular matrix has been chosen, it is recommended that it is used throughout the organization, so that all risk assessments are carried out on the same basis.



A typical risk matrix....2

- The output from the matrix will be a number of risk levels.
- These are arbitrary, and can be agreed based upon the organization's 'risk appetite'. (This is the degree of risk an organization is prepared to accept.)
- Organizations that have a low risk appetite include some aspects of the work of pharmaceutical companies, where the introduction of new drugs sometimes requires years of rigorous testing before a product is considered safe enough to launch.
- Organizations that display a high-risk appetite include some of the petrochemical companies, who will spend tens of millions of pounds in searching for scarce oil reserves, and will often drill many 'dry' holes before finally finding a rich source.
- The highest combination of impact and likelihood give the highest level of risk, and these are risks that should be treated as soon as possible.



A typical risk matrix....3

- Whatever the risk, the assessment for each threat should be recorded on a risk register, which will include details of the impact and likelihood, the level of risk calculated, possible treatment options, who should be responsible for carrying out the risk treatment, and a date by which the work should be completed.
- It is considered good practice to note a review date for each risk, as ongoing monitoring will show whether either the impact or likelihood of any threat has changed since the last assessment.
- It may also show whether other factors have had an effect; either increasing or reducing the threat, likelihood or impact of the risk.



Options for treating risks....1

- The output of the risk matrix will determine one of four courses of action in order to treat the risks. These will be:
 - Accept or tolerate the risk: When the level of risk assessed is very low, an organization may decide that it is willing to live with, tolerate and accept that risk.
 - Reduce or modify the risk: There are basically three possibilities – to reduce the threat, the vulnerability (and thereby the likelihood) or the impact of a risk. Actions that take place in reducing the risk are usually referred to as controls.



Options for treating risks....2

- Transfer or share the risk:
 - Transferring risk is to move it to a third party when the relevant expertise to manage the risk is not available within the organization.
 - Risk transfer can be achieved in a number of ways, but typically an insurance policy is an appropriate method when the impact of the risk can be measured as a purely financial one.
- Monitor:
 - The final stage of the risk management cycle is to monitor the results of the risk treatment plan.
 - The frequency of this process may vary according to the type of threat
 - Some threats may change very quickly and will require monitoring at frequent intervals, while others will change little over long periods of time and will only need occasional monitoring.



Approaches to risk assessment...1

- Qualitative: The main thing to agree is what constitutes 'high', 'medium' and 'low' for example, so that any assessment will have a degree of rationality about it, making it easy to understand and straightforward to justify later

Rating	Impact	Likelihood
1	Insignificant	Negligible
2	Minor	Rare
3	Moderate	Unlikely
4	Significant	Possible
5	Catastrophic	Probable

One possible rating framework for risk assessment



Approaches to risk assessment...2

- Quantitative:

- Takes a much more factual approach and can use statistical evidence to support both impact and likelihood assessments.
- Example, when assessing the risk of virus attacks, there will be plenty of numeric information available on the websites of anti-virus vendors to provide the basis for supporting a metric-based assessment.
- Statistical information to support likelihood assessments is very likely to be widely available, but should also be treated with caution.



Software tools

- There are a number of software tools available that will help in carrying out risk assessments.
- It is very easy to become obsessed with choosing the right software tool and working through a complex set of analyses, only to find that the answers are not as you would hope.
- Very often a simple analysis tool can be created using a spreadsheet and is therefore very much easier to tailor to the needs of the organization.
- Try to keep the work as simple as possible and reduce the impression of a ‘black art’ by making the results understandable to as wide an audience as possible.



Questionnaires....1

- To conduct a risk assessment it will be necessary to visit various areas of the organization to seek information from people who understand far more about their particular area of the business.
- Spend some time in preparing a questionnaire containing a series of questions designed specifically to discover the exact information required in order to carry out the risk assessment.
- Working with a questionnaire also helps to ensure that there is a level of consistency across the answers provided.



Questionnaires....2

- It is usually best to begin with open questions.
- Example:
 - asking people to describe the processes and procedures by which things happen, as this information will often point to the need for further questions.
 - It may help to begin by asking for an explanation of what the person's department does.
 - What are the inputs and outputs;
 - What processes are involved;
 - Who carries out the work.
- Closed questions can then follow, drilling down into the detail and uncovering facts and figures that will help the auditor to build up a more detailed picture and allow him/her to provide a detailed analysis of what might go wrong and how likely this might be.



Identifying and accounting for the value of information assets

- Before carrying out any form of risk assessment on the organization's information, it will be required to identify and document each of these 'information assets'.
- The value of each of these information assets will depend very much on its function, how long the business can manage without it, how long it would take (or how difficult it would be) to recover or restore it and how frequently the information changes.
- One of the key questions to ask when assessing the information value is 'how much will the organization lose (or not make) if the asset is not available?'.
- Example: Loss of access to a human resources database for a short period of time should not pose a serious threat and the impact would be low, but loss of a database holding online customer orders on an ecommerce website, even for a few minutes, would have a much higher impact.



Information classification policies

- Some information held by an organization (for example a product list) will be considered to be public domain information and will be allocated a low classification – often referred to as ‘unmarked’ or ‘unrestricted’.
- Other information will be more strictly controlled – for example, a list of customer accounts and their annual spend must be kept within the organization and will therefore have a higher level of classification such as ‘confidential’.
- More critical information will have a higher level of classification again. Example: documents relating to a merger or acquisition will not be available to many people within the organization – perhaps only at board level and for a very few senior managers. These might be graded as ‘highly confidential’ or ‘secret’.



Assessing the risks in business terms

- While it is very straightforward (after some practice) to carry out risk assessments, there will be a great temptation to describe and document these in risk management terminology.
- Terminology that is alien to the recipient will diminish the effectiveness of the risk assessment and may make it more difficult to convince the reader that appropriate action must be taken.
- It is always advisable for the risk assessor to be able to express the outcome in terms that are readily understood by managers within the business – to talk their language, in other words.
- A number of risk assessments must be ‘translated’ for the benefit of different departments in the organization.
- Example: different terminology is used in an organization's production and marketing departments, therefore the output of the risk assessments must be adjusted so that the language used reflects their own specific terminology.



Balancing the cost of information security against the potential losses

- Once the results of the risk assessments have been made available, there will be recommendations as to how the higher-level risks should be mitigated and have a rough idea at least of the order of cost.
- It is possible to present the results of the risk assessments in a more balanced way so that the decision-makers can take a more objective view.
- Example: If the anticipated losses as the result of a threat being carried out are £50,000, the overall risk may be deemed as medium. If the costs of reducing this to a lower level will amount to £25,000, the decision might well be to accept the risk rather than reduce it as the cost of the control is high in comparison to the possible impact.



The role of management in accepting risk

- Many organizations are unable to differentiate between accepting risk and ignoring risk (which is never an option).
- If the recommendation is to accept a risk, then the decision to do this must be a conscious one and should be fully documented.
- It is common practice for a single manager to ‘sign off’ a risk, it is better practice to have a second manager sign off as well if the impact is high – preferably one who is more remote from the risk itself but, nevertheless, one who has a good understanding of the potential impact of the risk materializing.



Contribution to risk registers

- Risk registers are a vital part of the overall risk management process.
- They achieve a number of objectives:
 - They permit all risks identified in the risk assessment process to be documented in a formal manner.
 - They allow an authorized observer (for example an auditor) to have visibility of the impact and likelihood of the risk and all the associated details and to assess the suitability of the responses selected.
 - They allow ongoing monitoring of the status of the risk and can be used as management reports on the progress of risk mitigation and of any variation in the risks.
- A risk register should contain as a minimum the details of the threat:
 - Its assessed impact and likelihood
 - The overall risk calculated from these
 - The recommended treatment (accept or tolerate, avoid or terminate, reduce or modify, transfer or share), and the actual action(s) to be taken
 - The person or department responsible for carrying out this work and the date by which it is expected to be completed.



Summary

- Threats, vulnerabilities, assets and impact are presented as the key concepts of information risk management.
- The risk management life cycle consist of identifying, analyzing, treating, and monitoring the risk.
- The conclusion of the risk management is to accept or tolerate the risk, reduce or modify the risk, transfer or share the risk, or monitor the risk.
- The risk assessment can be with qualitative or quantitative approaches.



Main Reference

1. Chapter 1 from Information Security Management Principles.

Additional References

1. Chapter 1 from Information Security: Principles and Practice, 2nd Edition
2. The cyber security principles. <https://www.cyber.gov.au/acsc/view-all-content/guidance/cyber-security-principles>



Thank You



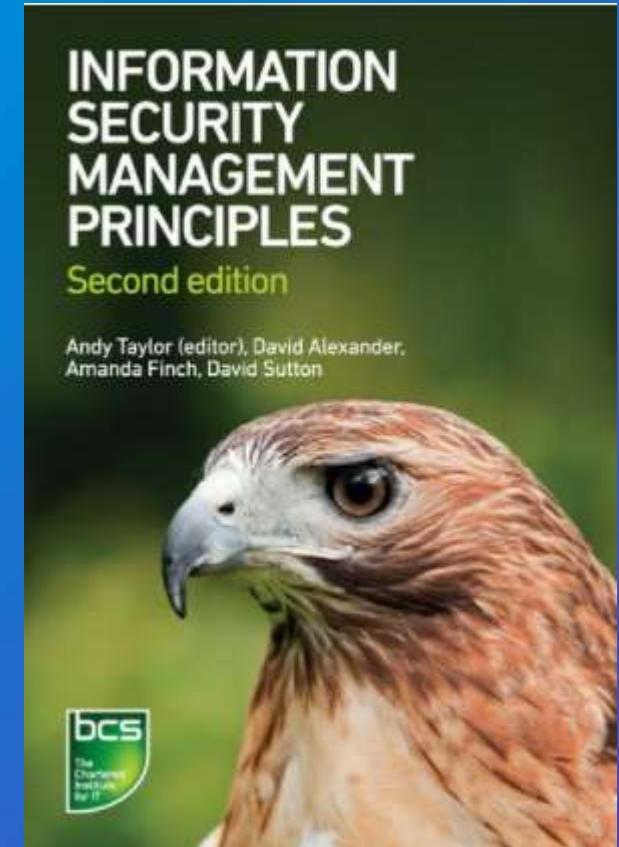


الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432

Introduction to Cyber Security and Digital Crime

Information Security Management Principles

by David Alexander, David Sutton, Andy Taylor, Amanda Finch



Week 3

Chapter 3: INFORMATION SECURITY FRAMEWORK



Contents

1. Organization and responsibilities
2. Organizational policy standards and procedures
3. Information security governance
4. Information security implementation
5. Security incident management
6. Legal framework
7. Security standards and procedures



Objectives

- Understand the principles for organizing information assurance across the enterprise.
- Define and explain the main concepts and also draft documents to meet the general requirements in the organization's management of security, the information security roles within the enterprise, the placement in the enterprise structure, and the responsibilities across the organization.
- Develop, write and gain user commitment for assurance policies, standards, operating procedures and guidelines.
- Explain and justify the main concepts and draft documents to meet the general requirements in developing, writing, and getting commitment to security policies.



Objectives (Cont.)

- Understand the principles of information assurance governance.
- Explain and justify main concepts and also establish procedures and draft documents to meet the general requirements in the review, evaluation, revision and audit of security policy.
- Understand the principles of how to implement information assurance measures within an enterprise.
- Manage assurance incidents and plan and conduct a forensic investigation.
- Understand the general principles of law, legal jurisdiction and associated topics and how they affect information security management.



Required Reading

1. Chapter 3 from Information Security Management Principles.



ORGANISATIONS AND RESPONSIBILITIES



The organization's management of security

- Establishing an organizational structure :
 - manage information assurance, and
 - provide a framework to ensure the understanding of the assurance requirements of the enterprise.
- Accountabilities need to be clearly defined
- Assurance activities need to be coordinated appropriately across the organization
- Ensure that the accountabilities and assurance activities are being managed effectively.



Information security roles within the enterprise....1

- A responsible should be nominated for the day-to-day management of information assurance issues.
- This is to ensure that good information assurance practice is applied properly and effectively across the enterprise and all assurance activities are coordinated.
- In larger organizations, this function should be a full-time role and the manager of this function is often referred to as the head of information assurance, the information security manager or the chief information security officer (CISO).
- In smaller organizations the role may be combined with other responsibilities.



Information security roles within the enterprise....2

- The information security manager needs to:
 - understand the information security risks that the enterprise may face,
 - what controls are in place,
 - where the enterprise may be vulnerable.
- This information must be communicated effectively to senior management (who have ultimate responsibility for information assurance).
- This is to ensure that they understand the status of assurance within the enterprise so that the appropriate safeguards are put into place.
- The main activities of the information security manager are:



Placement in the enterprise structure....1

- Placement of the various assurance roles within an organization will normally depend on the structure, the particular requirements and the culture of the enterprise.
- There are no definite hard and fast rules as to where the roles should specifically sit, how they should be organized or what their scope should include.
- In some enterprises, the information assurance function is located within the corporate compliance area.
- This is common in enterprises or industries that have a strong compliance culture such as banking or manufacturing.



Placement in the enterprise structure....2

- In other enterprises the function is based in the information technology group because many (but rarely all) of the controls to protect the enterprise are reliant on computer technology.
- Sometimes the function can be placed within a central facilities group since assurance responsibilities often span a number of management areas within an enterprise.
- To work effectively reporting structures should include dotted-line responsibilities to the chief risk officer (CRO), chief information security officer (CISO) or the chief finance officer (CFO).



Board/director responsibility....1

- One senior individual within the organization should be given the overall responsibility for protecting the assurance of the enterprise's information assets, and should be formally held accountable.
- This role should be performed by a board member, or equivalent, to demonstrate the enterprise's management commitment to information assurance.
- In some organizations, the CISO is a board member.



Board/director responsibility....2

- Their main responsibility is to ensure that appropriate assurance controls are implemented across the enterprise and to:
 - provide a single point of accountability for information assurance;
 - ensure that assurance goals are identified and meet the enterprise's needs;
 - ensure that adequate assurance resources are made available to protect the enterprise to an acceptable and agreed level of risk;
 - assign specific assurance roles and responsibilities across the enterprise;
 - provide clear direction, commitment and visible support for assurance initiatives, for example by approving and providing sign off for high-level security policies, strategies and requisite architectures.



Responsibilities across the organization

- Achieving good information assurance requires teamwork and a wide variety of skills, ranging from managerial to technical and administrative.
- The roles need to be delegated to the appropriate teams or to specific individuals with the necessary skills.
- For instance the skill sets required to maintain an enterprise's anti-virus systems are different from those required for administering user IDs.
- All those involved need to have a proper understanding of accountabilities and be given clear direction and support from senior management to achieve what is required of them.
- In many cases, individuals may be working together as a 'virtual team' that spans across separate management responsibility areas.
- Their activities require coordination and monitoring from a central information assurance function to ensure they are successful.



Statutory, regulatory and advisory requirements....1

- External factors can influence how an enterprise's information assurance should be managed, and these requirements need to be understood so that the appropriate assurance controls can be adopted to enable the business to fulfil its responsibilities.
- Requirements can arise from a variety of organizations such as the police, utility companies, government, trade regulatory bodies or telecommunications suppliers.
- They may be statutory, regulatory or advisory.



Statutory, regulatory and advisory requirements....2

- Statutory requirements are legal requirements that must be fulfilled.
- Law enforcement agencies must be contacted should certain laws be broken or are suspected of being broken.
- Compliance with these requirements may influence how an enterprise's incident reporting procedures are organized.
- Example: How, when and by whom should the authorities be contacted?
- Privacy legislation such as the Data Protection Act will influence how information is stored and managed within the enterprise and how resources are deployed to ensure that it complies with this legislation.



Provision of specialist information security advice and expertise....1

- Those involved in the security function should provide specialist security information advice and expertise to the enterprise.
- A high degree of current knowledge on information assurance matters should be maintained on topics such as awareness of industry trends, changes to organizational threats, new control measures, analysis of risk, legislation and compliance requirements and the latest technological developments.
- It is not necessary to have all the answers, but it is essential to be in a position to know where to find this information or to have access to someone with this specialist knowledge as and when needed.



Provision of specialist information security advice and expertise....2

- One way of achieving this aim is to keep in regular contact with special interest groups and websites or to network with information assurance peers in other enterprises via professional associations or security forums.
- Bulletin boards, websites and news groups also can provide early warnings of possible alerts, attacks and vulnerabilities and it is important to identify which ones may relate to the enterprise.
- A certain amount of ongoing self-education is needed to maintain this level of competency and to gain knowledge and understanding.
- Training courses are available to develop this knowledge with courses that range from specific topics on information assurance to training that covers a wider focus such as security management.



Creating a culture of good information security practice....1

- Information assurance needs the cooperation and collaboration of everyone with access to the enterprise's information.
- Involving everyone in the assurance process will help to develop a culture of good information security practice.
- It is important that information assurance is taken seriously by senior management within the enterprise and that they provide sponsorship and support for assurance initiatives.
- Their support and commitment will then cascade down through the organization.
- Line managers will proactively take responsibility for adopting information assurance measures within their teams, and likewise end users will know that they must take their own responsibilities seriously.



Creating a culture of good information security practice....2

- Positive reinforcement of good assurance behavior by the information assurance function and management helps to cement good behavior and some organizations even include feedback on assurance behavior in their performance reviews.
- A key factor for success is ensuring that everyone with access to the enterprise's information knows what is expected of them.
- Having in place clearly defined assurance roles and responsibilities, up-to-date security policies and standards and procedures will eliminate any ambiguities, but they also need to be clearly communicated and readily accessible.
- Example: Assurance responsibilities should be included in employee job descriptions and for third parties they should form part of their contract conditions.



ORGANISATIONAL POLICY, STANDARDS AND PROCEDURES



Developing, writing and getting commitment to security policies

- One senior person within the organization should be given the overall responsibility for protecting the assurance of the organization's information assets.
- They should be formally held accountable to ensure that appropriate security controls are implemented across the business.
- This director should be supported by a working group to ensure that adequate assurance measures have been put in place to protect the organization to an acceptable level of risk.
- Involving senior management will help to endorse the governance process.
- It will also ensure that adequate resources are made available, that controls are implemented effectively and that any identified security gaps are addressed.



Developing policies, standards, guidelines and operating procedures..1

- There is often confusion concerning the definition of policies, standards, procedures and guidelines so this should be clarified first of all.
- A policy is a high-level statement of an organization's values, goals and objectives in a specific area, and the general approach to achieving them.
- They should be regularly reviewed, policies should hold good for some time as they are not intended to provide either detailed or specific guidance on how to achieve these goals.
- Example: A policy might say that each user is responsible for creating and maintaining their system passwords, even if it doesn't say exactly how to do this.
- Compliance with policies is obligatory.



Developing policies, standards, guidelines and operating procedures..2

- A standard is more prescriptive than a policy.
- It quantifies what needs to be done and provides consistency in controls that can be measured.
- Example: passwords must contain a minimum of eight characters, be a mix of numbers and letters and be changed every 30 days.
- Standards should support policy and state what ‘must’ be done and how it should be achieved.
- Standards can be either general (for example handling sensitive information) or technical (for example encryption of data) but they should always relate to a specific subject.
- Standards are obligatory.



Balance between physical, procedural and technical controls....1

- Physical, procedural and technical controls can provide very effective security mechanisms and do much to reduce the likelihood of incidents occurring.
- Users need to access information systems in order to carry out their tasks and this inevitably introduces a level of risk to the information.
- They may need to share this data with colleagues or external suppliers and make value judgements as to whether it should be released to them.
- Reducing this kind of risk is difficult to achieve through technical controls alone. Technical controls introduced by a documental security system, for example, may well provide a good level of security.
- Formal policies and procedures can be used to make users aware of their responsibilities and the risks relating to the data to which they have access.



Balance between physical, procedural and technical controls....2

- Occasionally, due to time pressures or perhaps because of expediency, policy rules may be circumvented or ignored.
- Ignorance or failure to properly understand the policy will prevent compliance, and in these instances users won't understand the risks to their information assets and are very unlikely to be fully aware of the threats to them.
- Policies and procedures rely on individuals knowing that the policy exists and understanding what the policy expects of them as well as gaining their agreement to comply with it.
- There needs to be a sensible balance between using physical, procedural and technical controls to manage the risks associated with information assets.
- All three elements should be used to complement one another in a layered approach to manage risk to an acceptable level.



End-user code of practice

- The development of a high-level security policy should be bolstered by an end-user code of practice or acceptable use policy that provides a readily accessible way of communicating requirements to end users.
- An acceptable use policy demonstrates the organization's commitment to information assurance and must be approved by the director responsible for information assurance.
- It should be published to all users that need to access the organization's information management systems and include all employees (permanent and temporary, full and part time), contractors and third parties.
- The acceptable use policy should detail what is expected from users to protect the organization's information assets.



Consequences of policy violation

- Anyone accessing the enterprise's information assets needs to know and understand the consequences of a policy violation, and this should be clearly stated in the policy, standard or procedure.
- Appropriate processes should be established for reporting and dealing with violations so that they are managed in a consistent manner.
- These processes should be documented and agreed with the relevant stakeholders when the documents are produced.
- Violation of a policy may, in severe cases, lead to an employee disciplinary process being instigated, termination of supplier contract or the need to report the behavior to the appropriate law enforcement agency.



INFORMATION SECURITY GOVERNANCE



Review, evaluation and revision of security policy

- Reviews should take place after any significant changes to either systems or resources or as part of a regular review schedule (for example annually).
- A management review process should be established to ensure that policy reviews take place in an organized and timely manner.
- The review schedule should identify all the persons to be involved and a formal record kept of any revisions made, with an explanation as to why content has been incorporated or removed.
- Senior management should then approve the final version of any amended documentation.
- The review should involve all the main stakeholders including external parties and, where applicable, regulatory authorities.
- Once the review has been completed the revised policy should be communicated effectively to the relevant users, both internal and external to the organization.



Security audits and reviews

- Audits and reviews provide a good opportunity to understand how well things are working within the enterprise and they should provide senior management with valuable information on the assurance of their environment.
- Regular independent assurance audits and reviews should be carried out across the business to ensure that its information systems are compliant with existing security policies, standards and controls.
- Possible vulnerabilities to these systems can be checked and the effectiveness of existing controls can be tested.
- Audits and reviews should be carried out periodically or when a significant change (for example a system upgrade) has occurred.



Checks for compliance with security policy

- Regular checks should be carried out to measure compliance with security policies, standards and procedures.
- Carrying out compliance checks helps to identify whether controls are still adequate and relevant.
- Compliance checks also help to gauge the level of user understanding and awareness of their assurance responsibilities and whether or not these are being taken seriously.
- If regular checks are not carried out, then over time there can be a tendency for users to show less regard for them.
- Assurance is weakened as users become aware that monitoring does not take place and that they are not likely to be challenged.



Reporting on compliance status

- The finance industry has a long history of regulation and most stock exchanges have their own regulatory controls to prevent financial malpractice, but governance controls have gradually extended to other operating spheres.
- Many countries have produced their own codes of ethics, often in response to large corporate failures or in response to public pressure.
- The Sarbanes–Oxley Act was introduced in 2002, following a number of high-profile financial accounting scandals in the USA.
- The European Union's governance legislation was revised in 2004 via the Companies (Audit, Investigations and Community Enterprise) Act, which at the time of writing, is being implemented across the member states and will replace most of their local company legislation.
- Senior management and any regulatory or compliance bodies need to have access to sufficient information to be able to demonstrate compliance.



INFORMATION SECURITY IMPLEMENTATION



Planning – ensuring effective program implementation

- Good planning is the foundation of any successful information assurance program implementation.
- It can be used as a powerful tool for gaining support from both senior management and key stakeholders and to demonstrate how the assurance program is helping to reduce risk within the enterprise.
- This builds support for further initiatives.
- When planning an implementation, consideration should be given to how long it will take to implement controls, how easy the implementation will be, what the associated costs are and how important the resolution of the issues is to the organization.



Security architecture and strategy

- Information security strategy and architecture are two relatively new concepts in information assurance implementation.
- An information security strategy is a plan to take the assurance function within an organization from the reality of where it is now, with all its problems and issues, to an improved state in the future.
- It provides a road map or vision as to how this can be achieved and how it will support the organization going forward.
- A strategy should normally cover a period of time that is long enough to implement a significant level of change, but short enough to be able to predict changes in technology and organizational objectives (three- to five-year period).



SECURITY INCIDENT MANAGEMENT



Security incident reporting, recording and management

- Having an incident response plan that has been worked out and tested in advance is like having a good insurance policy.
- The first priority is to ensure that all the people within the organization know how to recognize an incident and who they should report it to.
- This can be done in a number of ways including awareness training, a section on a company intranet or portal and by carrying out exercises.
- There are normally five phases in the management of an incident:
 - reporting;
 - investigation;
 - assessment;
 - corrective action;
 - review.



Incident Response Teams and procedures

- An Incident Response Team (IRT) must be appointed in advance and all members of that team should be properly briefed and prepared.
- The members need to come from a cross section of the business to ensure that there is sufficient breadth of knowledge to deal effectively with the situation.
- They need to be senior and experienced enough to have the authority to make decisions on the spot.
- They must also be empowered to call upon additional resources (internal and external) as they see fit, to use in resolving the incident.
- There needs to be a documented escalation process for the team to reach the most senior members of the organization as and when necessary.
- It is advisable to give each of the team a copy of the incident response plan documentation and make sure they have a pager or mobile phone so they can be contacted immediately if the decision to activate the plan is made.



Processes for involving law enforcement

- There are times when it will be necessary to involve law enforcement or other similar organizations in the response to an incident.
- If there is any likelihood of criminal activity or other deliberate action, the appropriate authorities should be notified.
- It is important that senior management has a good understanding of the legal requirements for reporting certain events.
- One last possibility is that an organization may be visited by law enforcement officers conducting an enquiry into activities of which management has no knowledge.
- They may have a warrant to search the premises and remove items, or they may simply be conducting enquiries.



LEGAL FRAMEWORK



Protection of personal data

- Privacy laws exist to protect the rights of the individual.
- Most organizations hold and process information about people such as employee or customer information.
- Organizations need to be aware of the legal restrictions placed on them to protect this information and how it may be used and monitored.
- The last few years have seen an increase in privacy legislation and this should be considered when processing personal information.
- Many countries have legislation to protect the individual and to restrict and control not only the amount of information held, but how it should be used and monitored.
- They do share common principles there are significant differences in the legislative approaches and this can cause difficulties when working across different legal jurisdictions.



Employment issues and employee rights

- Depending on the legal jurisdiction, employees have certain rights when using the enterprise's information systems – such as the right to privacy and the right to know what information is held about them by the enterprise.
- Rights may also extend to monitoring controls.
- The enterprise must communicate this information to employees.
- The easiest way to do this is to include a statement about the extent of monitoring in the enterprise's information assurance policies or employment contracts.
- If this is not done it may be necessary to gain specific consent from individuals to allow their information to be monitored.
- An assessment of the monitoring strategy should be carried out to demonstrate that the monitoring techniques that are being used are justified, not excessive and meet legal requirements.



Common concepts of computer misuse

- Much of the legislation that currently applies to the misuse of computers has not been written specifically to address computer crime.
- It can be said that crime is crime and criminals simply use whatever means are available to carry it out.
- Legislation has been produced to target crimes that are committed using computers.
- The misuse of computers can include:
 - illegal access (hacking) to computer systems;
 - illegal interception of information;
 - interference with information and systems;
 - computer-related fraud and forgery;
 - commercial infringement of copyrights;
 - download of illegal material such as child pornography;
 - trafficking in passwords, digital signatures and encryption keys.



Requirements for the retention of records

- Certain documents or records need to be retained by an organisation for legal or regulatory purposes for a period of time.
- These can include company board minutes, financial reports and accounts or technical specifications.
- The duration for which documents need to be retained varies by the document type and the legislation of the country in which it is being used.
- In multinational organizations, records may be passed over to other countries within the same enterprise; meaning that the same data is then subject to different legislation requirements, which might even conflict with one another.
- Most retention requirements state a minimum length of time for keeping data, some legislation conversely states when a record must be destroyed.



Intellectual property rights

- Individuals and enterprises invest a lot of time, money and effort in creating original works, products, methodologies and ideas.
- They can be significantly out of pocket if they are unable to realize the benefit of their investment because other parties have used their ideas without compensating them.
- ‘Intellectual property rights’ (IPR) is the term given to the legal rights that protect creative works and most countries have legislation in place to protect such intellectual property.
- Copyright law was originally designed to protect original artistic works such as pieces of music, but its use can also be applied to software programs, computer games, documents, books or other types of work made using a computer or generated by a computer.
- Copyright is automatically associated with the piece of work upon its publication and has to be deemed as original.



Contractual safeguards

- When developing contracts with third parties it is important to put in place controls to protect the information assets of the enterprise to an acceptable level.
- The types of safeguards required will vary depending on the type of service being provided and the sensitivity of the enterprise data.
- Contract conditions should include clauses to ensure that proper assurance controls are in place.
- Security conditions are often handled via a security schedule within the contract.



Collection of admissible evidence

- There are a number of rules and processes that need to be followed when collecting evidence so that it can meet certain criteria when used in a court of law (described as admissible evidence).
- If legal guidelines are not followed, the evidence may be excluded as being inadmissible.
- This could result in a court case being lost, adverse publicity, loss of face and financial penalties to the prosecuting party.
- Organizations need to be able to demonstrate that evidence is authentic, has not been tampered with and has been gathered in an acceptable manner that meets legislative requirements.
- This includes being able to retain and document the state and integrity of items at the crime scene.



Securing digital signatures

- Traditionally, a handwritten signature on an original document proves who signed it and any alterations can be detected.
- In the electronic world the original is indistinguishable from a copy and this means that there is potential for fraud.
- Digital signatures are a form of electronic signature that addresses this problem.
- A digital signature electronically binds the sender of a message to the contents of the actual message to prove that it is genuine.
- It also proves when it was sent, to whom it was sent, that it has not been tampered with, that it has been kept confidential and that neither party can deny its transmission.
- Enterprises are increasingly using digital signatures to conduct their business and legislation has been developed to facilitate and control their use.
- What is acceptable varies across legal jurisdictions so it is important that you should obtain legal advice before adopting the use of digital signatures.



Restrictions on the purchase, use and movement of cryptography technology

- Cryptography is a powerful tool for protecting privacy that can be used by businesses, governments, criminals and individuals to protect confidential information.
- Governments argue that it is in the national interest for them to control cryptographic activity in order to protect the individual and to prevent and track criminal or terrorist activity.
- There are numerous controls in place over its use. Cryptography legislation varies greatly from country to country.
- In some countries, the controls are quite draconian, especially where repressive political regimes are in government.
- It is important that organizations that operate internationally understand the local operating restrictions as penalties can be extremely harsh.



SECURITY STANDARDS AND PROCEDURES



National and international standards

- There are many standards that apply to information assurance.
- These standards typically define a set of requirements for products, processes or procedures.
- Standard bodies collaborate with industry experts in different areas, whether representing vendors, scientific research agencies or government departments, to produce good practices that can be applied by others.
- The jurisdiction of a standards body may be international, or it may extend to a specific industry sector or a particular country alone.
- The standards that apply to an enterprise will vary depending on a number of factors; these may include the actual country in which the enterprise is based, whether it works internationally, the industry sector in which it operates, or perhaps engagement in government contracts.
- Most standards are produced by non-profit making organizations and are funded by the various parties that have a vested interest in their existence.



Certification of information security management systems

- Gaining information assurance certification is a means of demonstrating that the information assurance is conducted seriously and that good assurance processes and controls have been implemented.
- An increasing number of organizations now look for certification in their trading partners and, for some, certification can be a prerequisite for doing business.
- Certifications can apply across the enterprise, or to a specific set of processes within the organization.
- Certification usually involves the enterprise undergoing an external audit by an accredited third party.



Product certification to recognized standards

- Prior to being launched onto the market, many products require independent testing and certification to ensure that they conform to safety requirements, technical specifications or other compliance regulations.
- It is useful to have an independent third party to verify that a new product does meet expectations and that it can be trusted.
- This particularly applies to security products as it is often difficult for the consumer to be able to test the security of the product for themselves.
- Certificates provide customers with the assurance that the security features offer the level of protection that is claimed by the vendor.
- It is helpful to know that a standards-based approach has been used to do this evaluation, as this will aid understanding of how rigorous it has been.
- Test results produced in a standardized format will enable straightforward comparison with other competing products.



Awareness of key technical standards

- There are a number of technical standards that are applicable to information assurance management.
- The Internet Engineering Task Force (IETF) is a large open international community that develops and promotes standards for the internet.
- Its governing body meets two or three times a year.
- Standards are developed by working groups of interested parties such as network designers, operators, vendors and researchers that each focus on a particular topic.
- The standards generated are known as RFCs (Request for Comments), and upon production are subsequently issued to the IETF community as draft RFCs for comment and review.
- The published RFC documents have a status of either a proposed standard or an informational statement.



Summary

- The principles for organizing information assurance across the enterprise are based on the organization's management of security and the associated roles and responsibilities.
- The organizational policy, standards and procedures are focusing on how to develop them and how to balance between physical, procedural and technical controls.
- Information security governance includes the review, the evaluation and the revision of security policy.
- The information security implementation specify the techniques used to ensure effective program implementation.
- The security incident management starts from detecting incidents until applying the necessary response measures.
- The legal framework includes the protection of personal data, the employment issues and the employee rights and the intellectual property rights.



Main Reference

1. Chapter 3 from Information Security Management Principles.

Additional References

1. CYBERSECURITY FRAMEWORK, NIST, 2021,
<https://www.nist.gov/cyberframework>
2. A Proposed Best-practice Framework for Information Security, Governance,
Ghada Gashgari, Robert Walters and Gary Wills, 2017, DOI:
10.5220/0006303102950301
3. In Proceedings of the 2nd International Conference on Internet of Things,
Big Data and Security (IoTBDS 2017), pages 295-301,

<https://www.scitepress.org/papers/2017/63031/63031.pdf>

Thank You



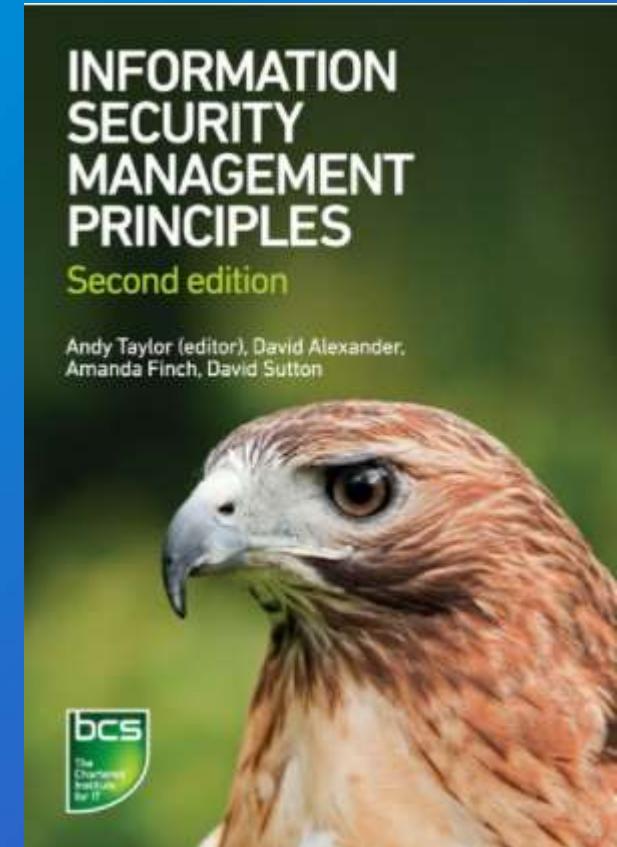


الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432

Introduction to Cyber Security and Digital Crime

Information Security Management Principles

by David Alexander, David Sutton, Andy Taylor, Amanda Finch



Week 4

Chapter 5: TECHNICAL SECURITY CONTROLS



Contents

1. Protection from malicious software
2. Networks and communications
3. External services
4. Cloud computing
5. IT infrastructure



Objectives

- Define how to put in place effective controls to manage the risks from malicious software.
- Understand the issues that organizations should take into consideration when identifying and managing the security risks to their networks and communications links.
- Understand the security issues surrounding services that use the network, which are often bought in from external suppliers.
- Understand the information security issues faced when utilizing cloud computing facilities.
- Understand the security issues surrounding the security of the IT infrastructure and the content of the associated documentation.



Required Reading

1. Chapter 5 from Information Security Management Principles.



PROTECTION FROM MALICIOUS SOFTWARE



Types of malicious software....1

- Malware (from MALicious softWARE), as it is often known, is one of the largest threats to the users and managers of information systems.
- An understanding of the capabilities of malware and those who write it, and the controls necessary to counter this threat, are essential for most information assurance practitioners.
- An unauthorized piece of code that installs and runs itself on a computer without the knowledge or permission of the owner.
- The traditional idea of malware is the virus that infects the computer, tries to spread to others, then destroys the contents of the hard drive or displays a message to show that it has successfully infected the machine.



Types of malicious software....2

- Modern malware can be split into the following major categories depending on their payload:
 - Viruses: These cannot spread on their own. They need to be attached to another piece of data or program to reach and infect another computer.
 - Worms: The difference between a worm and a virus is that worms contain the code needed to spread themselves without any user action.
 - Rootkits: These are complex software packages that hijack the operating system and attempt to make themselves invisible both to the user and to the software designed to find and remove malware.



Types of malicious software....3

- Back doors: Provides a means for a third party to access the computer and use it for their own purposes without having to carry out the normal authentication checks.
- Spyware: A common example of this is the use of cookies by websites. Some are designed to be permanent and to track and report the web usage back to a third party without the knowledge of the user.
- Trojans: Often disguised as another piece of software or are hidden inside compromised copies or other programs that users are lured into downloading and running.
- Another very successful infection route is through compromised websites.



Zero day exploits

- No matter how good and how comprehensive the defenses are, there is always the possibility that a new form of attack could pass through them.
- 'zero day exploits' : can bypass scan engines because they are not on the "stop" list in updates.
- Some products are better at detecting types of behavior than others, and their analysis tools can identify many new versions of malware because they exhibit behavior that is known to be unacceptable or similar code to that found in other malware.
- There is even a zero day exploit business, with hackers selling the knowledge to others.



Routes of infection

- Infected media: Any media that has been out of your control or supervision should be considered suspicious - CD, DVD, USB stick, etc.
- The most common routes today are through :
 - email, as an attachment or macro in a document,
 - websites,
 - unprotected systems.
- It is also possible that malware can infect the system through a wireless network connection, Bluetooth port, or infrared.
- With an increase in the number of employees allowed to “bring your own device” (BYOD) there is an increased risk to the company's IT infrastructure.



Routes of infection

- Infected media: Any media that has been out of your control or supervision should be considered suspicious - CD, DVD, USB stick, etc.
- The most common routes today are through :
 - email, as an attachment or macro in a document,
 - websites,
 - unprotected systems.
- It is also possible that malware can infect the system through a wireless network connection, Bluetooth port, or infrared.
- With an increase in the number of employees allowed to “bring your own device” (BYOD) there is an increased risk to the company's IT infrastructure.



Malware countermeasures....1

- The countermeasures required to detect and defeat malware depend on the configuration of the systems and networks to be defended and must be continually updated to face the latest threats.
- A single computer, connected to a broadband connection at home, is very different from a global corporate network or a small organization.
- Even for a single user, due to the different possible infection routes, a basic antivirus package is not enough:
 - needs a personal firewall package
 - profiling and access control tool



Malware countermeasures....1

- Example of malware countermeasures:

- content scanning for web traffic and some means of controlling web access to stop prohibited sites from being accessed;
- email content and source checking software;
- firewalls that block ports and check content;
- network intrusion detection or prevention systems;
- ‘Sheep-dip’ malware scanners for untrusted media;
- personal firewall or application control software on individual systems including checking files when they are accessed;
- use of managed services providers to scan mail and web traffic – inbound and outbound.



Methods of control

- Several approaches to controlling malware and reduce the risk:
 - Patches and upgrades are released quite frequently, and every organization should test and install fixes as soon as possible.
 - User awareness: Avoid to click on a suspect link or fall for a social engineering attack that tries to trick them into loading malware.
 - Harden the operating system by not installing unnecessary features or applications.
 - Ensure that default passwords and open configurations are not used.
 - Use of anti-virus and personal firewall software.
 - Harden the settings in the web browser in use.



NETWORKS AND COMMUNICATIONS



Entry points in networks and principles of authentication techniques..1

- The absence of a network would reduce the security requirements by a factor of ten.
- The network and communication links exist to make the systems connected to authorized/not authorized users.
- If there is an Internet connection somewhere, then there are over two and a half billion potential unauthorized users:
 - Some of them do not prepare anything good and will try to compromise the network
- Any location, logical or physical, from which a user or device can access a network is considered an entry point.



Entry points in networks and principles of authentication techniques..2

- The principle of authentication on a network is very similar to user access controls for identification and connection to a computer.
- A single sign-on system may even be used to authenticates the identity of the user on the network.
- There are protocols designed specifically for centralized access control (eg Radius, TACACS, Kerberos, and Diameter) that work well for networks.
- It could be just a username and password or some sort of token and code entry or possibly a challenge-response mechanism.



Partitioning networks....1

- Partitioning a network is another way to protect critical systems by limiting the amount of data that can be seen and makes an attacker's job much more difficult.
- Same principle as the physical access control to limit access to sensitive areas.
- Corporate governance and separation of roles rules within certain industries, including finance, require complete data separation to defend against insider trading and accusations of market manipulation.
- By using a network "sniffer", an attacker can potentially log all traffic passing through a segment.
- The sniffer can be:
 - hardware module or software
 - server as a Trojan horse to capture data and send it to the attacker



Partitioning networks....2

- Any connection to the outside, such as the Internet, must be protected by at least one firewall.
- For remote access, such as dial-up, ADSL, or web access, the servers must be located in a demilitarized zone (DMZ), located between two firewalls.
- Different approaches to partitioning networks:
 - Physically separation cabling
 - Virtual private networks (VPNs) configured in network hardware or even protocols like CISCO MPLS.



Cryptography in networking

- There are two common mistakes that a lot of people make when they think about cryptography:
 - Always preventing people from seeing the content of data.
 - To think that cryptography is only used to ensure privacy.
- The four main uses of cryptography are:
 - secrecy : nobody else can see the plain text;
 - data integrity : the data has not been changed, deleted or inserted;
 - user verification : this is the person they claim to be;
 - non-repudiation : the sender cannot later deny sending the message or its content.



Control of third-party access

- A remote connection from a vendor, used to support hardware or software remotely.
- Allow some form of electronic data interchange (EDI) to improve efficiency or speed up business processes.
- Example: A customer using a just-in-time manufacturing approach, placing electronic orders with suppliers for carefully scheduled component deliveries.
- Can be on the Internet through some sort of VPN, or through a private link.
- Can be a business partner, but they don't need to know a lot about your organization that isn't in the public domain.
- A good design will normally have the link to the third party located in a DMZ, protected by a firewall from the outside world.



Network usage policy

- The network usage policy document exists to define the purposes for which the network can and cannot be used.
- It will also define the individuals and roles that are authorized to use it and the official line on access control.
- Includes user profile definitions for each role :
 - privileges,
 - password lengths and strengths,
 - renewal period,
 - etc.



Intrusion monitoring and detection

- Detect intrusion from :
 - outside by unauthorized users,
 - authorized users within the organization who attempt to perform tasks for which they are not authorized.
- Ensure that all relevant log data is saved securely and in such a way that an attacker cannot modify or delete the information in order to cover their tracks with investigators and auditors.
- The data should be reviewed periodically to identify any unauthorized activity.
- Look for patterns of behavior that indicate some kind of attack.



Vulnerability analysis and penetration testing

- Vulnerability scan is the process of examining the network for any vulnerabilities that could increase the frequency or impact of any threat.
- Only the most qualified and reliable specialists should be allowed to perform this type of work.
- There are important legal issues to consider before undertaking any form of "penetration testing".
- Penetration testing is sometimes referred to as "ethical hacking" because testers will use many techniques that would be used by a hacker to identify weaknesses in the network.



Secure network management

- The task of managing a network securely is one of the most crucial aspects of providing IT services.
- Certain industries require minimum standards through legal and regulatory controls.
- Other sectors choose to implement them to comply with standards such as ISO 27001.
- Network management can play a major role in managing risk and improving resilience for business continuity.
- Any organization needs to have information about its physical and logical infrastructure by following the Plan–Do–Check–Act model.



EXTERNAL SERVICES



Securing real-time services

- The rapid increase in the popularity of services has added another dimension to the challenges faced by information security officials.
- Example in instant messaging:
 - to extract data;
 - to insert malware onto networks;
 - as a channel for phishing attacks;
 - for unauthorized purposes leading to legal action against the perpetrators.
- Other real-time services :
 - ordinary telephony,
 - Voice Over IP (VOIP),
 - Closed-Circuit TV (CCTV)
- VOIP is especially vulnerable if it is integrated into a single messaging system.



Securing data exchange

- Data exchange over the network must be protected against threats to confidentiality, integrity and availability.
- Cryptography and security protocols can be used to perform this function for data in transit.
- The key issue is to ensure that all parties protect the data to the same standard.
- Once the data arrives, it should be checked for any signs of malware or compromise before it is allowed to access it or gives it any credibility as legitimate traffic.



The protection of web services and ecommerce

- In business-to-business relationships, there is normally a lower degree of risk when Electronic Data Interchange (EDI) occurs.
- Users of web and e-commerce services are often members of the public, and organizations therefore have no control over the configuration and integrity of the PC used to access the service provided.
- Protection must be provided to prevent attackers from :
 - extracting data,
 - entering fake data,
 - adding their own code to the site.
- The most obvious form of cryptography that most people see and use is Secure Sockets Layer (SSL).



Protection of mobile and telecommuting services

- In the modern world, more and more people are spending time away from the office traveling or working from home.
- Mobile phone companies also provide services (GPRS, GSM, HSDPA, EDGE, and LTE) for a high-speed connection to the office and to the Internet.
- The three main problems facing assurance practitioners are:
 - The connection uses network infrastructure that does not belong to the company, so traffic can be viewed, altered or deleted by an attacker.
 - The users take their IT and communications equipment away from company premises, where it is more vulnerable to theft, loss or compromise.
 - Ensuring that connections are only used by authorized employees.



EXTERNAL SERVICES



Introduction

- Cloud computing is an umbrella term used to describe on-demand, off-site, and location-independent IT services.
- Cloud computing can be provided in a variety of ways :
 - Infrastructure As A Service (IAAS)
 - Software As A Service (SAAS)
 - Platform As A Service (PAAS)
 - Others
- Businesses are enthusiastically taking advantage of cloud environments that allow them to quickly implement technical solutions to meet business needs.



Legal implications for cloud computing

- It can be relatively easy for a business or end user to contract for cloud services.
- When a business or end user signs up for a cloud service, the organization has accepted the terms and conditions and entered into a formal contract, which may limit the legal rights of the organization.
- Most countries have legislation controlling the storage of personally identifiable information and it is essential that information held in the cloud meets these legal and regulatory requirements.
- Some providers reserve the right to change their terms and conditions without prior consent, which may infringe an organization's rights and control over the information held.



Security issues when selecting a cloud supplier

- A cloud service provider is a third-party supplier and third-party security best practices should be followed when engaging with them.
- The organization must understand the financial and operational impacts if the cloud service is suddenly withdrawn or becomes unavailable, or if its information is compromised or leaked.
- When choosing a cloud provider, the organization should ensure that the provider can meet their security requirements and fully understand how the service will be delivered.
- The organization must consider all stages of the information lifecycle and obtain explicit assurance that key security issues are being addressed at an adequate level.

Distinguishing between supplier commercial risk and purchaser risk

- The main risks for the cloud service provider will be largely commercial.
- If they do not provide the contractually agreed service, their business model will fail and serious business problems will ensue.
- The organization may choose to outsource its data to a cloud provider, but it must still retain the responsibility of ensuring that the data is properly protected (risk sharing).
- Risk assessment and business impact analysis for purchasing such services should be undertaken with as much (if not more) rigor as if the service were provided in-house.
- When entering into a relationship with a cloud provider, it is essential to have a good understanding of what will be provided and how it will be delivered, as discussed previously.



IT INFRASTRUCTURE



Separation of systems to reduce risk

- A simple, but very effective way to manage risk and provide insurance is to separate systems.
- In some cases, it may be decided that the risks outweigh the benefits and this should not be done.
- An alternative is to allow very limited functionality to pass between systems through an inter-domain connector (IDC) or to allow data to pass only one way, through some form of data diode or a specially configured router.
- Separate systems has the advantage that they are less complex to manage and easier to assess for risk due to the reduced complexity which always increases the possibility of error in computer systems.



Conformance with security policy, standards and guidelines

- There is no point in having standards for the design, implementation and operation of systems if they are not followed.
- Accreditation to the ISO/IEC 27000 series will require that all the relevant controls have been identified, documented, implemented and then followed.
- The policy defines the overall information assurance objectives of the organization and must be supported by the board of directors and the CEO to provide authority.
- The standards define the minimum acceptable criteria for achieving this policy in key areas.
- The guidelines indicate how to design and implement workable procedures and countermeasures to meet standards and enable the business to manage risk.



Control of privileged access and correctness of input and stored data

- Many organizations use additional guarantees for accounts that can grant the privileged access:
 - Example: use longer passwords (eg 12 characters instead of 9)
- Attackers and malware can subvert stored data and can exploit incorrect user input or errors in the design or coding of the software.
- There are several ways to promote data accuracy :
 - Make sure the design of the software and database is correct
 - Use proven code review techniques when developing and testing
 - Use defensive coding, which checks for values within acceptable ranges
 - Train the users in how to use the application properly
 - Audit the system regularly to look for anomalies
 - Try to identify how the errors occurred and then how to stop them happening again



Principles of recovery capability

- The capability of recovering data in case of loss or errors.
- It has been shown that any organization that loses access to its data for more than 10 days is very likely to go bankrupt.
- It is absolutely essential that backups exist for all data, and not just a current backup.
- Consider a disaster recovery contract or the option of moving the data center to another company location in an emergency.
- It is required to keep a copy of the backup in a secure offsite location.



Intrusion monitoring and detection methods

- Intrusion Detection and Prevention Systems (IDS and IPS) use automated tools to analyze log data, system activity, and network traffic for the purpose of identifying a security breach.
- The analysis is performed either by an application or by a hardware device.
- Use statistical and SNORT techniques to analyze data for changes in system configuration or operation, or for known types of behavior.
- IPS solutions are the most problematic because they tend to prevent authorized users from working when they block a false positive.
- It is recommended to configure them correctly and to educate the system to understand what is, and is not, normal activity.



Installation of baseline controls to secure systems and applications

- Baseline controls standards are used to define how systems should be configured and managed.
- The contents will include details on:
 - which versions of operating systems to use;
 - which parts of the operating system to install;
 - the patches required;
 - additional applications such as anti-virus software, intrusion detection agents and so on;
 - settings for password length, access control lists and so on;
 - network configuration.



Configuration management and operational change control

- The process of monitoring and controlling the configuration of devices and documentation within the infrastructure.
- The configuration documentation should describe the baseline that is in place and it can then be used to identify changes made.
- Managing change control requires the efficient configuration management process as a core element.
- Documentation can be used to help assess change requirements and the impacts those changes may have before granting change approval.
- Documentation can also be used as part of the audit process, for quality, assurance and operational purposes.



Configuration management and operational change control

- The process of monitoring and controlling the configuration of devices and documentation within the infrastructure.
- The configuration documentation should describe the baseline that is in place and it can then be used to identify changes made.
- Managing change control requires the efficient configuration management process as a core element.
- Documentation can be used to help assess change requirements and the impacts those changes may have before granting change approval.
- Documentation can also be used as part of the audit process, for quality, assurance and operational purposes.



Protection and promotion of security documentation

- Organization and third parties (managed service providers or outsourced operations) are required to :
 - work to the same standards of information assurance and reporting,
 - adopt the same working practices,
- The use of work protocol documents and contractual clauses may require them to conduct an audit to ensure compliance.
- After producing a set of security documents, it is very important that they are protected against unauthorized access and loss.
- They can be physical, electronic, or both, and all must be protected.



Summary

- Protection from malicious software is explained by understanding the origin of malwares and the routes of infections.
- Some techniques are used to secure real-time services, web services, mobile and telecommuting services.
- As external services, the cloud computing is considered a promoting technology which requires advanced security methods and legal implications.
- Protection of the IT infrastructure is linked to the separation of systems, the control of privileged access and correctness of input and stored data.



Main Reference

1. Chapter 5 from Information Security Management Principles.

Additional References

1. Critical Security Controls for Effective Cyber Defense. Part 1: The Critical Security Controls, ETSI, 2018.
https://www.etsi.org/deliver/etsi_tr/103300_103399/10330501/03.01.01_60/tr_10330501v030101p.pdf
2. Chapter 9 and chapter 11 from Information Security: Principles and Practice, 2nd Edition



Thank You



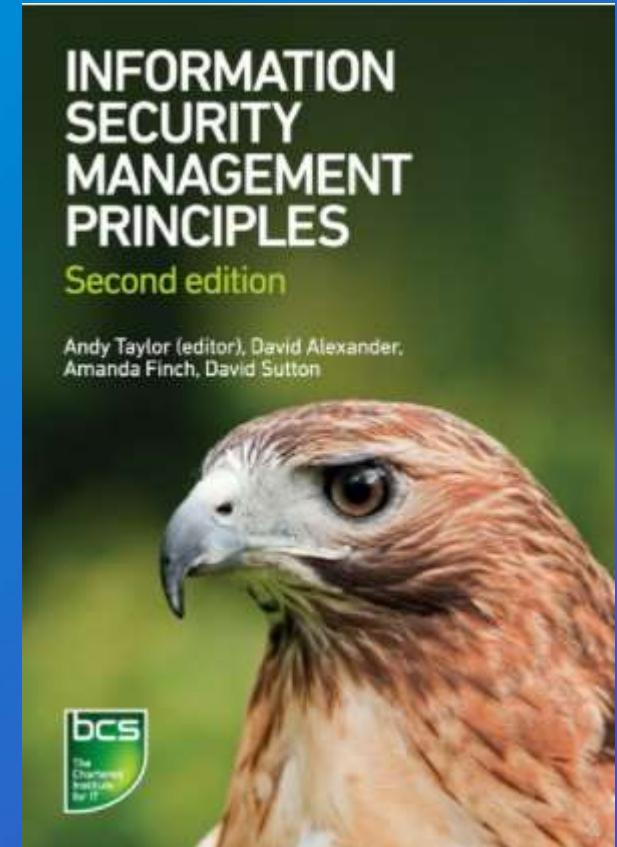


الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432

Introduction to Cyber Security and Digital Crime

Information Security Management Principles

by David Alexander, David Sutton, Andy Taylor, Amanda Finch



Week 5

Chapter 7: PHYSICAL AND ENVIRONMENTAL SECURITY



Contents

1. General controls
2. Physical security
3. Technical security
4. Procedural security
5. Protection of equipment
6. Processes to handle intruder alerts
7. Clear screen and desk policy
8. Moving property on and off site
9. Procedures for secure disposal
10. Security requirements in delivery and loading areas



Objectives

- Understand the environmental risks to information in terms of the need for appropriate power supplies
- Understand the environmental risks to information in terms of the need for protection from natural risks (fire, flood and so on)
- Understand the environmental risks to information in the everyday operations of an organization.



Required Reading

1. Chapter 7 from Information Security Management Principles



1. General controls



General controls

- There are three types of controls:
 - Physical
 - Procedural
 - Technical
- Each has its own role.
- If they combined appropriately, they will enhance the assurance of an organization significantly and effectively.
- However, if they are used inappropriately, they can actually end up reducing overall security.



2. Physical security



Physical security

- It is the first line of defense in many organizations.
- It prevents intrusion, or at least make it very difficult for an intruder to gain access.
- Main entrance needs to be secured to ensure unauthorized is not entering the building.
- Other entrances; Fire escapes, back doors; must also be suitably protected.
- climbing walls and fences to gain access to roof lights, upper-floor windows or stairs is also a source of a security risk that must be assessed and managed.



Physical security

- Inside a building:
 - offices, server rooms, cabinets, desks and other document storage facilities, and other sensitive areas need to be secured.
- Equipment needs to be appropriately protected.
 - In high-risk and high-security, crash-proof barriers outside the server locations is needed to prevent vehicles being driven into the building and destroying the equipment.



3. Technical security



Technical security

- It employs technology in some way to offer security.
- It is related to computers and the software techniques that can be employed:
 - Technical locks, using tokens or fingerprints
 - Hardware, through the ‘locking’ or disabling of ports
 - or to some other technological solution for a specific application



Technical security

- One of the main concerns about such measures is the ease with which they can be overcome.
- For example, if there is a power failure in an electronic locks we have no protection at all.



3. Procedural Security



Procedural Security

- It covers the rules, regulations and policies that an organization puts in place to help reduce the risk.
- For example, including clauses in employment contracts that legally bind employees to obeying the security policy.
- If employees violate the security policy, they might be disciplined or dismissed.
 - This will strengthen the procedural controls, and have them more effective.



Procedural Security

- Background needs to be done when hiring new employees.
 - This ensures that they don't have any convictions or other incidents in their background that might jeopardize the security of the organization.
- Part of procedural control is to conduct orientation.
 - It will ensure that all staff are fully aware of their responsibilities.



Combined controls

- Combination will provide a high level of assurance and should prevent most incidents happening.
- For example:
 - controls getting into the site (procedural and physical);
 - further checks for specific high-risk buildings (procedural, physical and technical);
 - specific logons to computer systems containing or processing classified data (technical and procedural);
 - a set of well-drafted and effectively policed policies which staff are well aware of and have signed contracts to that effect (procedural).



4. PROTECTION OF EQUIPMENT



PROTECTION OF EQUIPMENT

- Stolen equipment or loss of the use of some critical equipment, due to fault, power failure is a serious issue.
- This might result in loss of revenue, loss of credibility, reputational losses.
- Some measure is necessary to reduce the impact or remove the cause of such a problem.
 - Wiring PCs to prevent them from being stolen.
 - having a stand-by power generator
 - This is frequently used in hospital life support systems.



PROTECTION OF EQUIPMENT

- having appropriate maintenance contracts and service level agreements can be used to enhance the security of the assets.
 - But it comes with its own price.
 - Organization needs to balance security with cost.
- Business impact analysis, risk assessment and cost benefit analysis to determine what is appropriate.



5. PROCESSES TO HANDLE INTRUDER ALERTS



PROCESSES TO HANDLE INTRUDER ALERTS

- For each of the three of control there may be one of three uses:
 - Preventative action
 - Detective action
 - Reactive action.



PROCESSES TO HANDLE INTRUDER ALERTS

- Physical Controls:
 - As a preventative action:
 - stop unauthorized people getting into a building.
 - As detective action:
 - intruder alarms.
 - As reactive controls:
 - electrified fences
- A potential intruder will be detected, arrested and hopefully prevented from trying it again.



PROCESSES TO HANDLE INTRUDER ALERTS

- Technical Controls:
 - Anti-virus software is an example of all three uses.
 - Preventative:
 - Any malware being loaded on a computer system.
 - Detective
 - Routinely run checks to ensure there is none installed
 - Reactive
 - Provides a system for virus removal.



PROCESSES TO HANDLE INTRUDER ALERTS

- Procedural Controls:
 - Non-disclosure agreement to protect the intellectual property rights of an organization is an example of all three uses.
 - Preventative:
 - Prevent unauthorized disclosure by warning of the consequences.
 - Detective
 - Disabling the copying of large database records of clients.
 - Reactive
 - Possible actions that can be taken in the event of a breach – dismissal, legal action.



PROCESSES TO HANDLE INTRUDER ALERTS

- link between the different types of controls and their uses:
 - The procedures may define the technological protection measures that are used behind some physical barrier.
 - If a preventive control fails, a detective control must be in place. Also, a determination of consequences must be in place as well.



PROCESSES TO HANDLE INTRUDER ALERTS

- Disgruntled employees need to be handled carefully.
 - They might put malware onto computer systems, delete or copy important files.
 - Their access needs to be disabled immediately.
 - They need to be escorted off the premises and forward their personal belongings.



6. CLEAR SCREEN AND DESK POLICY



CLEAR SCREEN AND DESK POLICY

- Employees might leave their disk unattended.
 - open-plan offices allow other employees to see sensitive information being displayed on screens.
 - Third-party companies to carry out routine tasks such as cleaning is another threat.
- A clear screen policy must be enforced.
 - The time default of the clear screen needs to be balanced.
 - Not too short so employee needs to reenter password.
 - Not too long that allows other to see sensitive information.



CLEAR SCREEN AND DESK POLICY

- In a cluttered desk, sensitive documents may become covered by other papers.
 - This becomes potential for security breach.
- Clear desk:
 - A clear desk is much easier to check for important documents that should be locked away.
 - This prevents intruders from gaining a chance to use or expose sensitive information.



7. MOVING PROPERTY ON AND OFF SITE



MOVING PROPERTY ON AND OFF SITE

- New models of cars being released before the official announcement, classified data being found at the roadside are examples of security incidents.
- The control of an organization's property both on and off site is needed to increase assurance and security.
- To reduce the risk of such incidents:
 - All assets must be uniquely identified.
 - Inventory needs to be include all assets.
 - Only authorized is allowed to move equipment around.



MOVING PROPERTY ON AND OFF SITE

- Taking equipment off site should also be controlled.
- Laptops and smartphone are essential for mobile business.
 - Bring your own device' or BYOD is business need for individuals to use their own devices for work purposes.
 - The security of the devices and, the prevention of spreading malware from these devices to the corporate environment must be considered and managed
 - The data and access need to be controlled.
- A clear procedure on how to use equipment's, where to store them if not in use is a good practice.



9. PROCEDURES FOR SECURE DISPOSAL



ROCEDURES FOR SECURE DISPOSAL

- The disposal of equipment or other information assets might contain:
 - Confidential files
 - Valuable or sensitive files on hard disks in computers;
 - Classified data
- A good start is to have a policy and procedure to ensure that any equipment is properly checked before it is allowed off site.
- If a contractor engaged in the process of destruction , it is necessary to follow up to ensure a secure destruction has been done.



ROCEDURES FOR SECURE DISPOSAL

- For electronic media, delete key on a computer will not remove the information completely from the system.
- ways to completely delete the information:
 - writing random data multiple times
 - physically destroying the media itself – cutting-up or shredding hard drives.



10. SECURITY REQUIREMENTS IN DELIVERY AND LOADING AREAS



SECURITY REQUIREMENTS IN DELIVERY AND LOADING AREAS

- Delivery and loading areas are often remote from the main buildings of organizations. But it still has a security concerns.
- It is common for those working in such areas to have lower security clearances.
- They are responsible for the receipt and dispatch of goods. They have access to a lot of business-critical information.
- example: If the inbound goods dealt with are new computers, then they need as much protection in the receipt and dispatch areas as they receive in all other parts of the organization



Summary

- Physical and environmental security is important to ensure the security of the premises.
- Three principal types of control can be applied. If combined, it will increase the security.
- Equipment is an asset of the organization, and it needs to be protected.
- A secure and proper disposal of equipment will reduce security threat of leaking data.



Thank You



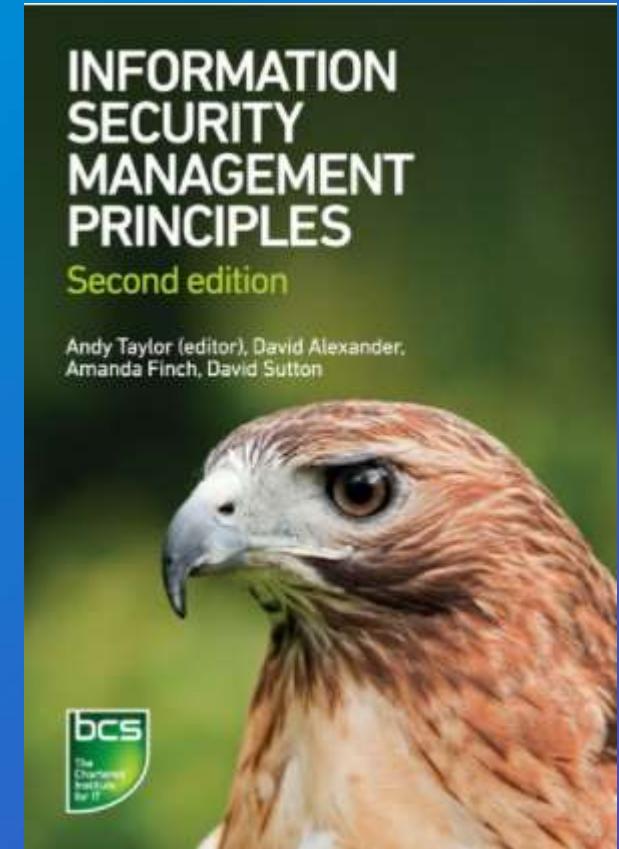


الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432

Introduction to Cyber Security and Digital Crime

Information Security Management Principles

by David Alexander, David Sutton, Andy Taylor, Amanda Finch



Week 6

Chapter 8: DISASTER RECOVERY AND BUSINESS CONTINUITY MANAGEMENT



Contents

1. DR/BCP, risk assessment and impact analysis
2. Writing and implementing plans
3. Documentation, maintenance and testing
4. Links to managed service provision and outsourcing
5. Secure off-site storage of vital material
6. Involvement of personnel, suppliers and IT systems providers
7. Security incident management
8. Compliance with standards



Objectives

- Define and explain DR/BCP, risk assessment and impact analysis and be able to describe their appropriate use as applicable.
- Understand how to write and implement plans.
- Define the links to managed service provision and outsourcing.
- Understand how secure off-site storage of vital material.
- Examine the involvement of personnel, suppliers and IT systems providers.
- Conduct security incident management.
- Define compliance with standards.



Required Reading

1. Chapter 8 from Information Security Management Principles



1. DR/BCP, risk assessment and impact analysis



DR/BCP, risk assessment and impact analysis

- Business Continuity Plan (BCP) is:
 - maintaining the continuity of business operations.
- Printer running out of toner or more serious, such as a power outage are problems will adversely affect the operational capability.
- An approach that maintain normal operations as effectively as possible while resolve these issues must be taken.
- The main objective of BCP is to ensure availability of information and resources.



DR/BCP, risk assessment and impact analysis

- Disaster recovery (DR) is:
 - restoration of ‘normal operations.
- If normal operations can not be restored, DR plan may become the new ‘normal operations’.
- the risk assessment and business impact analysis (BIA) are the keys for DR.
- DR might include: hot, warm or cold backup sites, and significant investment.
 - But if the disaster is very rare, it is not worth to invest significant amounts on a disaster recovery plan.



DR/BCP, risk assessment and impact analysis

- BIA is the key factor to develop a disaster plan.
 - If the impact is too severe and the organization not being able to operate effectively ever again, then a DR must be developed.
 - If the impact might cause minor disruption that can be resolved within a few hours, then DR might not be necessary.
 - But if the minor event occurs each week then the cumulative effect of the event may raise its importance and impact.



DR/BCP, risk assessment and impact analysis

- The distinction between DR and BCP:
 - If the plan calls for minor adjustments to normal working practices or a comparatively small change in normal operations, then this should form part of the BCP.
 - DR on the other hand is generally focused on contingency planning for ICT systems, and may be part of a larger BCP.



2. Writing and implementing plans



Writing and implementing plans

- It is important to write BCP and DR that work for the organization itself.
- A complete and effective risk assessment needs to be conducted.
 - Unlikely events need to be considered in this assessment.
- Plans need to be developed to address what to do if the resources became unavailable for significant period.



Writing and implementing plans

- The procedures for implementation is essential to know which part of the DR/BCP plan should be implemented.
- It is good practice to involve key staff members to determine what event would really have a major effect.
- DR/BCP need to consider problems happened elsewhere that might impact the business.
 - Lack of access is an example. If a building next door is experiencing an issue, how that might affect your business?



Writing and implementing plans

- Writing and disseminating a plan is not enough.
- Implementation requires awareness and education program to ensure that all staff end up being fully aware of the plans.
- Practicing the developed plan ensures what actions they need to take in the event of an issue arising.



3. Documentation, maintenance and testing



Documentation, maintenance and testing

- Documentation is vital and can make the difference between a successful conclusion to an event and a disaster.
- Documenting the expected actions and procedures is not enough.
 - The availability of the document at the right time and location is important to have successful plan.
 - The integrity of the secured plan is another factor that needs to be considered.



Documentation, maintenance and testing

- Maintenance of the plans is another area that can cause problems.
- A plan needs to be updated regularly.
 - If not checked, it becomes invalid and wouldn't work properly and there is high chance to create more problems.



Documentation, maintenance and testing

- Routines for testing and checking the details of the plans must be comprehensive.
 - Testing a plan needs to be balanced with other factor.
 - For example, closing a factory to test the BCP might provide an excellent validity, but might significantly affect the profitability.



Documentation, maintenance and testing

- Testing types:
 1. Desk check: key people sitting round a desk pretending to do the activities required of them in the plans.
 - Major issues with plans will be discovered and will enable updating.
 2. Full-scale test: the entire plan needs to be examined at the same time not in parts.
 - it needs significant planning and must be coordinated by incident management team.



Documentation, maintenance and testing

- Testing types:
 - 3. Brown envelope' technique: simulate possible emergencies in some way.
 - Setting up a scenario of a major incident.
 - then drafting a number of instructions to relevant staff in 'brown envelopes'.
 - The relevant staff members are instructed to open the envelope at the appropriate time and to take the necessary actions in accordance with the information supplied.
 - This could be to make a telephone call, to invoke a particular element of a plan or some other action including reacting in some specific manner.



Documentation, maintenance and testing

- 'Brown envelope' technique can be used to test the plans for one specific location.
 - This may need to be done when the normal work is unlikely to be severely affected.
- The objective of testing is to test the appropriateness, effectiveness and comprehensiveness of the overall planning.
- Other factors, weather, rush-hour travel, pandemics, need to be considered when practicing a plan.



4. Links to managed service provision and outsourcing



Links to managed service provision and outsourcing

- Plans need to consider the services supplied by a third party.
- A contract should include the expected level of service in the event of an emergency.
- Contractors must be involved with the development and testing of any set of BCP or DR plans.
 - The services they provide may be critical to the overall success of the DR.



Links to managed service provision and outsourcing

- Access to BCP and DR plans is critical.
 - It would be useless to have a huge investment on a plan that are not available at the crucial time.
 - A plan needs to be available to those who need them whenever and wherever they are.
 - A plan needs to be consistent and with latest version.
 - Contact details for all the key players needs to be included.
 - A plan needs to be secured.
 - For example, on an encrypted memory stick that can be used anywhere.



Links to managed service provision and outsourcing

- If there is a need to secure a plan in another place, it is important to consider other factors, such as distance.
- It might also be necessary to consider that critical information might need to be stored elsewhere to enable ‘normal business’ to be maintained.
 - Taking backup tapes from computer systems off site each night to ensure their availability.



5. Secure off-site storage of vital material



Secure off-site storage of vital material

- Stolen equipment or loss of the use of some critical equipment, due to fault, power failure is a serious issue.
- This might result in loss of revenue, loss of credibility, reputational losses.
- Some measures are necessary to reduce the impact or remove the cause of such a problem.
 - Wiring PCs to prevent them from being stolen.
 - having a stand-by power generator
 - This is frequently used in hospital life support systems.



Secure off-site storage of vital material

- having appropriate maintenance contracts and service level agreements can be used to enhance the security of the assets.
 - But it comes with its own price.
 - Organization needs to balance security with cost.
- Business impact analysis, risk assessment and cost benefit analysis to determine what is appropriate.



5. Involvement of personnel, suppliers and IT systems providers



Involvement of personnel, suppliers and IT systems providers

- It is important that all staff, full and part time, are fully aware of the workings of the BCP and DR plans.
- Ongoing education and training program are important.
 - Having an orientation course covering all the basic requirements and highlighting responsibility in the event of an incident is a good practice.
 - That needs to be supported by exercises and reminder sessions.



Involvement of personnel, suppliers and IT systems providers

- Education and training program need to consider other key players, suppliers, outsource companies.
- Staff who routinely work within the establishment of the client, manning an IT helpdesk, must be involved and actively engaged.
- The need for suppliers of crucial services to be involved is also paramount.
 - it is critical that those who have to work with the contracts on a day-to-day basis must understand the detail of how any SLA will work in the event of a major problem arising.



Involvement of personnel, suppliers and IT systems providers

- Supply chain needs to be address as well. In case of security incidents, options need to be considered:
 - Holding stocks of critical elements elsewhere in an alternative location.
 - Ensuring that all the suppliers themselves have up-to-date and workable BCP and DR plans;
 - Ensuring the contracts cover the more likely eventualities.



7. Security incident management



Security incident management

- Incident management is the term used to describe the work done to deal with the incident itself.
- A team should be specifically trained to handle such incidents.
- Forensics team may involve to secure evidence such as temporary files on a computer, fingerprints (electronic or human), audit logs.



Security incident management

- The relationship between the teams responsible for dealing with the incident, the BCP and the DR plan must be a very close one.
- Prioritization is important aspect that needs to be considered.
 - Decide which business function to continue requires someone in a position of authority to make the necessary and, most importantly, timely decisions.



7. Compliance with standards



Compliance with standards

- There are a number of standards that cover some, or most, aspects of the management of BCP and DR.
 - IT Service Continuity Management.
 - The ITIL guide on service management best practice
 - ISO/IEC 27031:2011 as part of the ISO/IEC27000 series of information security management standards.
 - Business Continuity Institute produces its own Good Practice Guidelines that are an excellent source of information



Compliance with standards

- It would be well advised to consult the appropriate documentation that is available to ensure that what they implement in this area is based on good practice and therefore likely to be successful.
- information assurance is based on best practices, thus experiencing them is the key factor to know how good they are.



Summary

- BCP is an important plan to ensure the availability and continuity of a business.
- DR is essential plan that ensures going back to normality.
- Risk assessment needs to be conducted to build a successful plan.
- Writing a plan would facilitate familiarity with it.
- Training and testing are the key factor to measure the effectiveness of BCP/DR plans.



Thank You



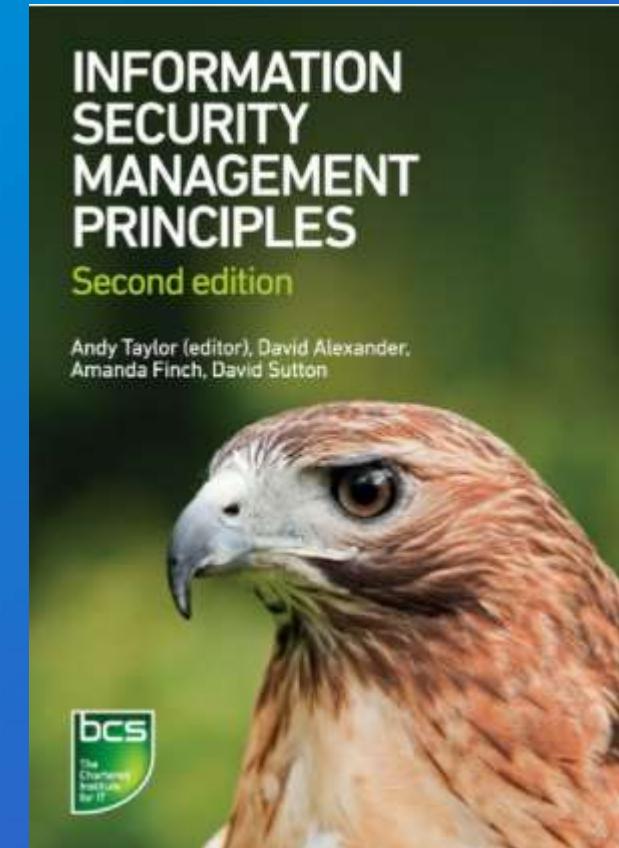


الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432

Introduction to Cyber Security and Digital Crime

Information Security Management Principles

by David Alexander, David Sutton, Andy Taylor, Amanda Finch



Week 6

Chapter 9: OTHER TECHNICAL ASPECTS

Contents

1. Investigations and forensics
2. Role of cryptography



Objectives

- Understand the important aspects of incident investigation and how forensic evidence may be preserved.
- Define the role that cryptography plays in protecting systems and assets, including awareness of the relevant standards and practices.



Required Reading

1. Chapter 9 from Information Security Management Principles



1. Investigations and forensics



Investigations and forensics

- In some occasions, it is necessary to investigate activity and use forensic techniques to discover and preserve evidence for later use.



Common processes, tools and techniques for conducting investigations

- If evidence is not handled carefully, it is very easy to render evidence inadmissible.
- Observe high standards when investigating an incident will be a great help in winning the case.
- The UK's Police and Criminal Evidence Act (PACE) defines very strict standards of conduct in order to allow the police to demonstrate that the evidence is valid and admissible in court.



Common processes, tools and techniques for conducting investigations

- **Evidence custody officer:**

- This person is responsible for:
 - Collecting
 - Securely storing evidence
 - While maintaining a good documentary record to preserve what is often referred to as the ‘chain of evidence’.



Common processes, tools and techniques for conducting investigations

- There are forensic tools available to collect and examine evidence from IT systems.
 - These tools need to be used by skilled and properly trained investigators.
 - If the tools used inappropriately it might destroy the evidence.
 - A well trained third party might be contracted since many organizations will not have this kind of resource in-house.



Common processes, tools and techniques for conducting investigations

- To achieve accountability, policies and procedures should contain part on investigating incidents and attributing responsibility to an individual.
- The investigation must be conducted in a manner that preserves the evidence in a form that is compliant with legal procedures.
- In Security incident management, there is a need to practice incident response and investigations to help identify any flaws in the plans.
- Final tip: Understand the rules, do not break the evidence.



Relations with law enforcement

- If an organization is considered important to the ‘national interest’, many law enforcement agencies will cooperate with it to help improve the level of security.
 - Such as the Centre for the Protection of National Infrastructure (CPNI) in the UK and the Department for Homeland Security (DHS) in America.
 - Emergency response teams, such as CERT in America, GovCertUK in the UK and in other countries, and the Forum for Incident Response and Security Teams (FIRST).



Relations with law enforcement

- Law enforcement agencies often have specialist staff who can offer advice and guidance to any organization that feels at risk from logical or physical attacks.
- It is always worth contacting them for any material they can provide.
- It might be necessary to involve law enforcement or other similar organizations in the response to an incident



Relations with law enforcement

- It is, for instance, mandatory in the UK to inform the police if there is a suspicion of terrorist activity or that child pornography has been viewed or processed through the IT systems of an organization.
- UK legislation also requires the reporting of suspicious financial activities.
- Similarly in Saudi Arabia, any suspicious activities need to be reported to law enforcement.



Relations with law enforcement

- If there is any likelihood of criminal activity or other deliberate action:
 - the appropriate authorities should be notified.
 - It is important that senior management has a good understanding of the legal requirements for reporting certain events.



Security issues when procuring forensic services and support from third parties

- One of the reasons why organizations must develop policy, processes and procedures is the need to be ready to investigate incidents and possible criminal offences.
- An incident response policy needs to be developed and checked with a specialist.
 - Any mistake can render the findings inadmissible in a court.



Security issues when procuring forensic services and support from third parties

- When an investigation starts:
 - It is vital that the incident team consults senior management to decide involvement of criminal justice agencies.
 - It also determines the level of effort required to gather the evidence.
 - There might be a need to get external specialists on site to image hardware drives for evidence.



Security issues when procuring forensic services and support from third parties

- Outsourcing forensics and investigations to a third party is an option for organization:
 - that do not have their own skills in-house.
 - Or it is not considered to be a major risk.



Security issues when procuring forensic services and support from third parties

- Companies that do decide to have their own in-house resources:
 - Must make sure that the products and skills they acquire are sufficient.
 - It is very important that the used product meet standard.
 - All the above requirements will provide legally admissible evidence.
 - One simple mistake can invalidate everything else you do correctly.



Security issues when procuring forensic services and support from third parties

- If there is a need to contract a third party to complete the investigation, the followings need to be considered:
 - Signed non-disclosure agreement:
 - This document provides legally binding confidentiality.
 - The third party is required under obligation to provide the same level of confidentiality to the information seen.
 - The third party is legally required to notify law enforcement of any suspected child pornography or terrorist activity, even if they have signed an NDA.



Security issues when procuring forensic services and support from third parties

- The investigation contract should have also the following:
 - Standards required in preserving evidence and accompanying documentation for legal admissibility
 - The handover, assured destruction or secure erasure of all materials obtained by the third party at the end of the incident;
 - Participation in any review at the end of the incident to improve the response process.



Security issues when procuring forensic services and support from third parties

- Evidence must be collected as soon as possible.
 - In order to preserve evidence, devices must not be used until forensically investigated.
 - This allows to return to normal operations quickly.
 - Approval is needed when conducting the investigation.



2. ROLE OF CRYPTOGRAPHY



ROLE OF CRYPTOGRAPHY

- Cryptography plays an important role in protecting systems and assets.
- It establishes a partnership or chain of trust between two or more remote parties.



Basic cryptographic theory, techniques and algorithm types

- Cryptography should provide:
 - Confidentiality: it is kept secret, assuming that this is a key requirement of the relationship
 - Integrity: it is not changed by third parties while in transit.
 - Authentication :the origin of the information is assured.
 - Non-repudiation :the originator cannot deny having sent the information.



Basic cryptographic theory, techniques and algorithm types

- The needs to provide confidentiality:
 - Secure information stored in a system against unauthorized access.
 - File or disk encryption are examples of this.
 - Secure information while in transit between sender and recipient so that unauthorized parties are unable to understand the information even if they are able to intercept it.



Basic cryptographic theory, techniques and algorithm types

- To provide confidentiality:
 - Information or ‘plain text’ may be encrypted – changed into ‘cipher text’.
 - The original plain text cannot be read or inferred.
 - Then sent to the recipient who reverses the process by decrypting the message to recover the original plain text.



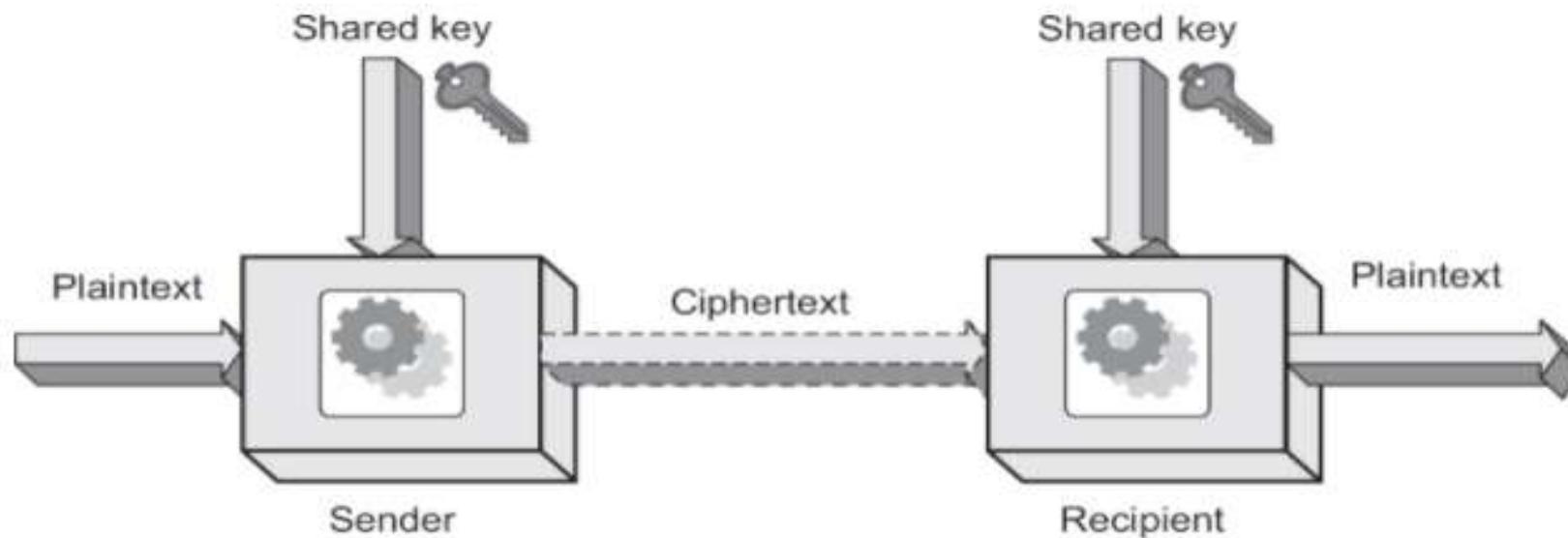
Basic cryptographic theory, techniques and algorithm types

- Cryptography may be used in several ways:
 1. It may be used to encrypt information during transfer from one computer to another.
 2. It may be used to encrypt a number files on computer media.
 3. It may be used to encrypt an entire hard disk drive including the operating system, applications and configuration information as well as the data.



Secret (or symmetric) key cryptography

Figure 9.1 Symmetric key encryption



Secret (or symmetric) key cryptography

- There are two methods of encrypting information:
 1. stream cipher: information is encrypted effectively a bit (as in binary digit).
 - Example of this is of GSM mobile phones
 2. block cipher: information is encrypted in bulk as one or more blocks of data and the entire message are sent to the recipient.
 - Examples of block ciphers are IDEA, RC5, DES, Triple-DES and AES.



Secret (or symmetric) key cryptography

- The processes of encryption or decryption require two things:
 1. Algorithm: a computational method.
 - Only algorithms deliver strong encryption can be used with any degree of certainty.
 2. Key: a string of binary digits or bits.
 - Both sender and recipient keep the encryption key as a shared secret.



Secret (or symmetric) key cryptography

- Symmetric encryption :
 - It uses the same key for both the encryption and decryption processes.
 - Once the recipient has received the encrypted message:
 - it can be decrypted using the same secret key and the same algorithm as that with which it was originally encrypted.



Secret (or symmetric) key cryptography

- Cryptanalysis attacks on keys:
 - Brute force attack:
 - each possible combination of bits that make up the key are tried in turn, eventually one will work.
 - But by making the key greater in length:
 - The number of possible permutations will be increased.
 - This will result in it taking longer for an attacker to identify the valid.



Secret (or symmetric) key cryptography

- Cover time:
 - The minimum time for which the information must remain secret.
 - If an attacker can recover the key by brute force in less than the cover time, a stronger key is needed.



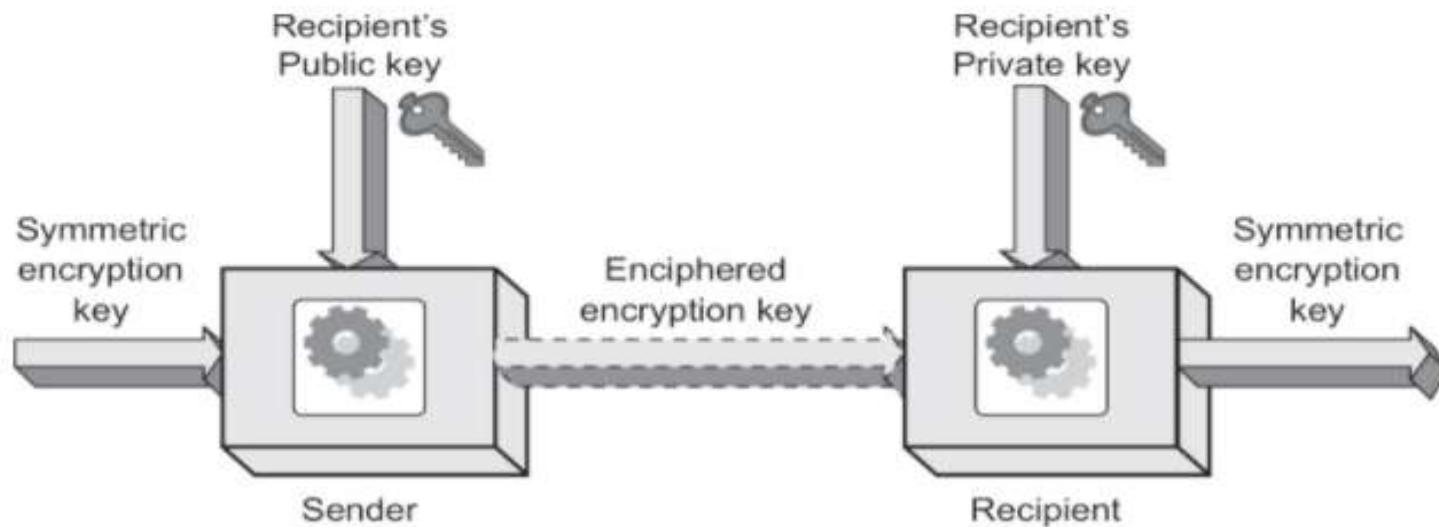
Secret (or symmetric) key cryptography

- If the shared key has been compromised, then a new key needs to be generated and shared.
 - One of the attack is ‘man-in-the-middle’ attack. It intercepts the communication and can use the recovered key.



Public key (or asymmetric) cryptography

Figure 9.2 Asymmetric key encryption



Public key (or asymmetric) cryptography

- It solves key exchange problem.
 - it is possible to exchange secret keys between sender and receiver without them being compromised.
 - the recipient can be assured that the new key has originated from a trusted source and not from a ‘man-in- the-middle’ attacker.



Public key (or asymmetric) cryptography

- There are two entirely different keys:
 1. Public key: is intended to be used by anybody – it is not secret and is shared with anyone who needs to use it
 2. Private key: is intended to be kept secret by its owner
- They are mathematically linked, but in such a way that it is virtually impossible to deduce the private key from the public key.



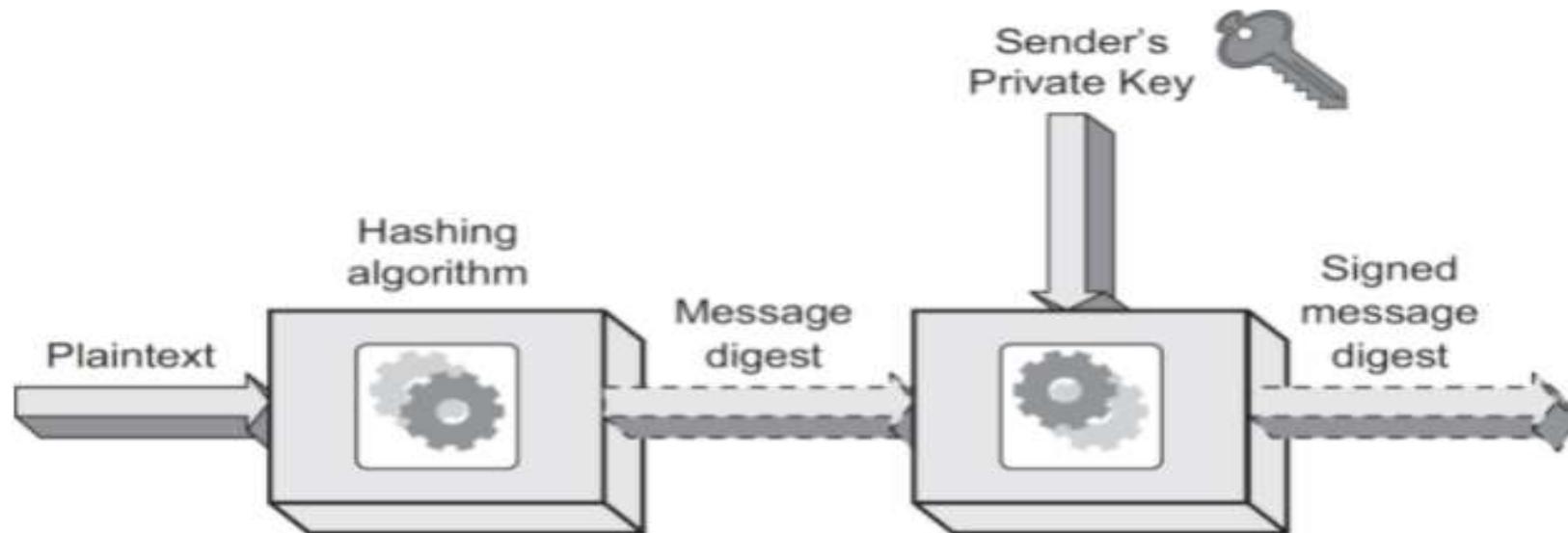
Public key (or asymmetric) cryptography

- Anyone can encrypt data to send to a recipient using the recipient's public key.
- That data can only be decrypted using the recipient's private key.
- This means there is no need to agree and exchange a secret encryption key in advance



Hashing techniques

Figure 9.3 Producing a signed message digest



Hashing techniques

- Aka message digest.
- It produce a numerical value.
- Hashing provides verification of the integrity of a message.
- There is no way to recover the original text from a strong message digest



Hashing techniques

- Aka message digest.
- It produce a numerical value.
- Hashing provides verification of the integrity of a message.
- There is no way to recover the original text from a strong message digest



Hashing techniques

- Ensuring the integrity of a message:
 - a message digest is produced from the original message.
 - Encrypted with the sender's private key and sent to the recipient.
 - They will be able to decrypt this encrypted or 'signed' message digest using the sender's public key.
 - They can then produce their own message digest and compare that with the received message digest.
 - If the two are identical, integrity of the message has been proven



Digital Certificate

- It is authenticating the sender's public key. :
 - The sender obtaining a digital certificate from a certification authority or CA (for example Verisign).
 - The digital certificate signed with their private key authenticates the entity's public key.
 - Most or all of the above components comprise what is generally referred to as a Public Key Infrastructure (PKI).



Summary

- Evidence needs to be handled carefully.
- Employing standards will help having evidence admissible.
- if there is a need to outsource the investigation process, it is important to have the third party to sign the NDA.
- Cryptography provides protection to organization assets.
- Two types of encryption, symmetric and asymmetric, can be used.
- Integrity can be achieved via hashing.



Thank You



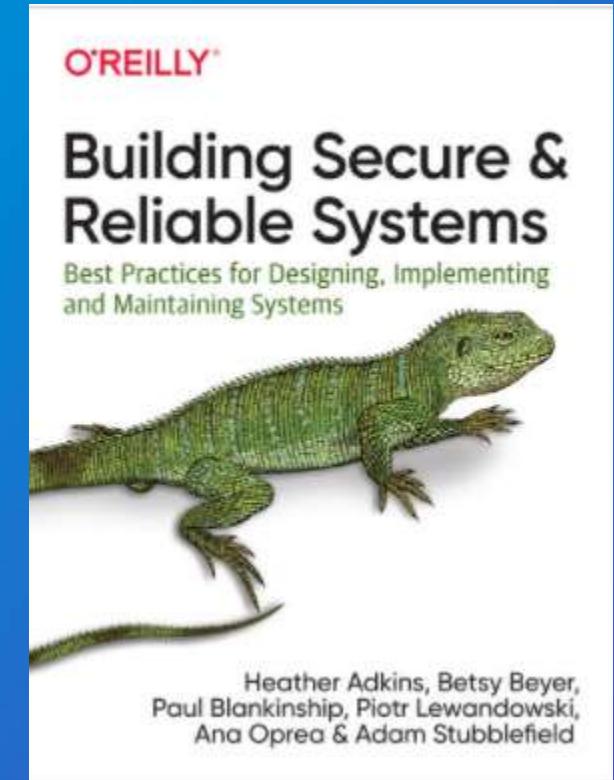


الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432

Introduction to Cyber Security and Digital Crime

Building Secure and Reliable Systems

by Heather Adkins, Betsy Beyer, Paul Blankinship, Piotr Lewandowski, Ana Oprea, Adam Stubblefield



Week 8

Chapter 1: The Intersection of Security and Reliability



Contents

1. Reliability Versus Security: Design Considerations
2. Confidentiality, Integrity, Availability
3. Reliability and Security: Commonalities



Objectives

- Consider risk when designing a reliable system.
- Apply fail safe/ fail secure to secure a system
- Describe and examine the commonalities of Reliability and Security when designing a system
- Understand the concept of Confidentiality, Integrity, Availability



Required Reading

1. Chapter 1 from Building Secure and Reliable Systems



1. Reliability Versus Security: Design Considerations



Reliability Versus Security: Design Considerations

- In designing for reliability and security, you must consider different risks
- **Reliability risks** are non-malicious in nature.
- For example, a bad software update or a physical device failure.



Reliability Versus Security: Design Considerations

- **Security risks** come from adversaries who are actively trying to exploit system vulnerabilities.
- When designing for reliability, you assume that some things will go wrong at some point.
- When designing for security you must assume that an adversary could be trying to make things go wrong at any point.



Reliability Versus Security: Design Considerations

- Different systems are designed to respond to failures in quite different ways.
- In the absence of an adversary, systems often *fail safe* (or open): for example, an electronic lock is designed to remain open in case of power failure, to allow safe exit through the door.



Reliability Versus Security: Design Considerations

- Fail safe/open behavior can lead to obvious security vulnerabilities.
- To defend against an adversary who might exploit a power failure, you could design the door to fail secure and remain closed when not powered.



Reliability and Security Tradeoff: Redundancy

- In designing for reliability, you often need to add redundancy to systems.
- For example , if an electronic locks fail, you need a physical key to unlock the door during power failures.
- Redundancy increases reliability, however, it increases the attack surface.
 - An adversary need only find a vulnerability in one path to be successful.



Confidentiality, Integrity, Availability

- Confidentiality, integrity, and availability are considered the fundamental attributes of secure systems and are referred to as the CIA triad.
- Both security and reliability are concerned with the confidentiality, integrity, and availability of systems.



Confidentiality

- A reliable system must not breach confidentiality accidentally.
- For example, In the aviation industry, having a push-to-talk microphone stuck in the transmit position is a confidentiality problem.
 - it might broadcast private conversations between pilots in the cockpit, which represents a breach of confidentiality.
- In this case, no malicious adversary is involved:
 - a hardware reliability flaw causes the device to transmit when the pilot does not intend it to.



Integrity

- A secure system must prevent an active adversary from accessing, tampering with confidential data.
- Data integrity compromise need not involve an active adversary.
- For example, Google Site Reliability Engineers (SREs) noticed that the end-to-end cryptographic integrity checks on a few blocks of data were failing due to memory errors,
- They write a software that compute the integrity for every version and they detect all the errors.



Availability

- Availability is both a reliability and a security concern.
- An adversary might exploit a system's weak spot to bring the system to a halt or impair its operation for authorized users.
- They might control a large number of devices spread around the world to perform a classic distributed denial-of-service (DDoS).
- Denial-of-service (DoS) attacks are an interesting case.
- It is hard to distinguish a malicious attack from a design flaw or a legitimate spike in traffic



Reliability and Security: Commonalities

- Reliability and security are emergent properties of a system's design.
- It would be hard to fix them after the fact.
- Ideally, you need to take both into account from the earliest design stages.
- They also require ongoing testing throughout the entire system lifecycle,
- In a complex system, reliability and security properties are often determined by the interaction of many components.



Reliability and Security: Invisibility

- Reliability and security are mostly invisible when everything is going well.
- Since both are invisible, they might be seen as a cost that can be reduced without consequences.
- The costs of reliability and security failures can be severe.
- For example, a power failure caused key computer systems to shut down at Delta Airlines and resulted in almost 700 flight cancellations and thousands of delays.



Reliability and Security: Assessment

- It is not practical to achieve perfect reliability or security, a risk-based approaches is used to estimate the costs of negative events.
- The probability of negative events for reliability and security should be measured differently.



Reliability and Security: Assessment

- For reliability of systems, it can be assessed according to desired error budgets or include metrics such as fault tolerance levels of the system.
- For security, a simulated attack can also be used to evaluate a system's resistance to particular kinds of attacks.



Reliability and Security: Simplicity

- A simple design is one of the best ways to improve the ability to assess the reliability and the security of a system.
- A simpler design:
 1. reduces the attack surface.
 2. decreases the potential for unanticipated system interactions.
 3. makes it easier to understand the system and help responders mitigate symptoms quickly and reduce mean time to repair (MTTR).



Reliability and Security: Evolution

- Systems are expected to change over the time regardless of the simplicity of the initial design.
 - Adding new feature requirements, and changes in scale will introduce complexity.
 - On the security side, evolving attacks and new adversaries can also increase system complexity.
- Complexity often accumulates unintentionally.
- A small change may lead to major consequences for a system's reliability or security.



Reliability and Security: Evolution

- An example of a small change that leads to a major problem is:
- In the Debian GNU/Linux version of the OpenSSL library:
 - A standard tool for debugging memory problems was reporting a warning.
 - A developer removed two lines from code to eliminate the warnings.
 - This caused OpenSSL's pseudo-random number generator to only be seeded with a process ID, which on Debian at the time defaulted to a number between 1 and 32,768.
 - Brute force could then easily break cryptographic keys.



Reliability and Security: Resilience

- Systems should be designed to be resilient under adverse or unexpected circumstances.
- For example, in a memory utilization problem:
 - From the reliability perspective; to address component failures, system design should incorporate redundancy and distinct failure domains so that you can limit the impact of failures by rerouting requests.
 - In complex system, defense in depth and distinct failure domains can be used.



Reliability and Security: Resilience

- *Defense in depth* is the application of multiple, sometimes redundant, defense mechanisms.
- *Distinct failure domains* limits an adversary's ability to exploit a compromised host or stolen credentials in order to escalate privilege and affect other parts of the system.



Reliability and Security: Resilience

- Distinct failure domains can be implemented by categorizing permissions or restricting the scope of credentials.
- For example, Google's internal infrastructure supports credentials that are explicitly scoped to a geographic region.
- These types of features can limit the ability of an attacker who compromises a server in one region to move laterally to servers in other regions.



Reliability and Security: Resilience

- for defense in depth, using independent encryption layers for sensitive data is a common mechanism.
- For example, even though disks provide device-level encryption, it's often a good idea to also encrypt the data at the application layer.
- This way, even a flawed implementation of an encryption algorithm in a drive controller won't be sufficient to compromise the confidentiality of protected data if an attacker gains physical access to a storage device.



Reliability and Security: Resilience

- Also, we need to consider insider attackers.
- The *principle of least privilege* can mitigate insider threats.
 - It dictates that a user should have the minimal set of privileges required to perform their job at a given time.
 - For example, mechanisms like Unix's sudo support fine-grained policies that specify which users can run which commands as which role.



Reliability and Security: Resilience

- The *multi-party authorization* is used by Google to ensure that sensitive operations are reviewed and approved by specific sets of employees.
- This multi-party mechanism both protects against malicious insiders and reduces the risk of innocent human error, a common cause of reliability failures.



From Design to Production

- Security and reliability considerations should be kept in mind.
- Starting with the development of the code, opportunities exist to spot potential security and reliability issues.
- Common frameworks and libraries are used reviews, and prevent such issues.
- Before deploying a system, you can use testing to ensure that it functions correctly.
- Edge cases need to be considered as well.



From Design to Production

- Testing can be used to:
 - Understand the behavior of a system under a flood of queries,
 - fuzzing to explore the behavior on potentially unexpected inputs,
- Specialized tests are used to ensure that cryptographic libraries aren't leaking information.
- Testing plays a critical role in gaining assurance that the system you've actually built matches your design intentions.



From Design to Production

- Some approaches to actually deploying code can limit security and reliability risk.
 - For example, canaries and slow rollouts can prevent you from breaking the system for all users simultaneously.
- Similarly, a deployment system that accepts only code that's been properly reviewed can help to mitigate the risk of an insider pushing a malicious binary to production.



Investigating Systems and Logging

- It is usually impractical or too expensive to achieve perfect reliability or security.
- When failures occur in preventive mechanisms:
 - a plan that detects and recovers from such failures is needed.
- Good logging is the foundation of detection and failure preparedness.



Investigating Systems and Logging

- The more complete and detailed logs are better.
- Log volume can have reliability and security issues.
 - Log volume poses a significant cost, and analyzing logs effectively can become difficult.
 - Security logs pose an additional challenge: logs typically should not contain sensitive information, such as authentication credentials or personally identifiable information (PII).
 - lest the logs themselves become attractive targets for adversaries.



Crisis Response

- During an emergency, teams must work together quickly and smoothly because problems can have immediate consequences.
- For example, in 2014:
 - An attacker put the code-hosting service out of business. All data including backups had been deleted.
- Well-rehearsed collaboration and good incident management are critical for timely responses in these situations.



Crisis Response

- It is best to have a plan in place before an emergency occurs.
- Responders are operating under stress and time pressure, with limited situational awareness.
- Security incidents also typically cause tension.
- The investigation might grow beyond company boundaries or involve law enforcement agencies.
- During a crisis, it is essential to have a clear chain of command and a solid set of checklists, playbooks, and protocols.



Recovery

- Recovering from a security failure often requires patching systems to fix a vulnerability.
- Recovery needs to happen as quickly as possible, this process needs to be reliable.
- The capability to push changes quickly is a double-edged sword:
 - It can close vulnerabilities quickly.
 - However, it can also introduce bugs or performance issues that cause a lot of damage.



Recovery

- The pressure to push patches quickly is greater if the vulnerability is widely known or severe.
- The choice of whether to push fixes slowly to have more assurance that no other vulnerabilities have been introduced or to do so quickly depends on:
 1. a risk assessment
 2. a business decision.
- For example, it may be acceptable to lose some performance or increase resource usage to fix a severe vulnerability.



Summary

- Security and reliability have a lot in common.
- Both are inherent properties of all information systems that are tempting to initially sacrifice in the name of velocity, but costly to fix after the fact.
- Recover from incidents requires fixing a vulnerability and ensuring no others have been introduced.



Thank You



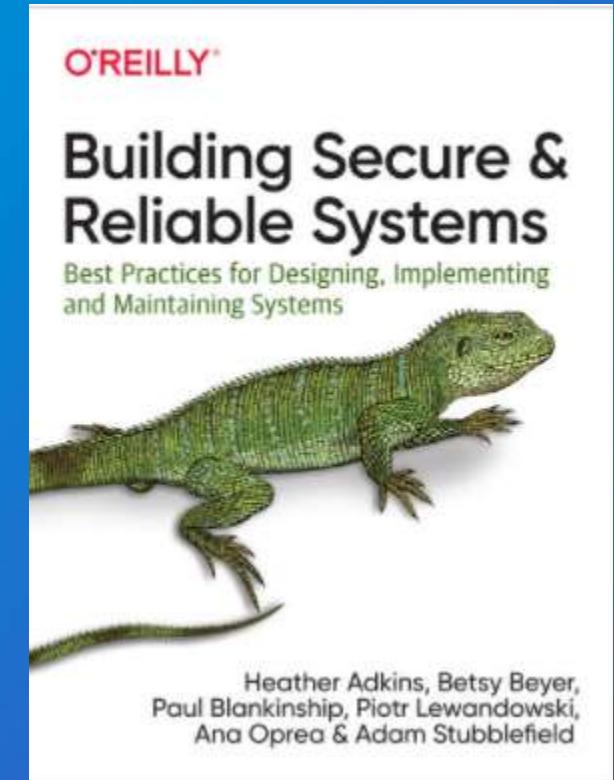


الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432

Introduction to Cyber Security and Digital Crime

Building Secure and Reliable Systems

by Heather Adkins, Betsy Beyer, Paul Blankinship, Piotr Lewandowski, Ana Oprea, Adam Stubblefield



Week 9

Chapter 2: Understanding Adversaries



Contents

1. Attacker Motivations
2. Attacker Profiles
3. Attacker Methods
4. Risk Assessment Considerations



Objectives

- Identify attacker motivations
- Outline profiles of attackers
- Describe how attackers carry out their attacks
- Assessing risk posed by various attackers



Required Reading

1. Chapter 2 from Building Secure and Reliable Systems



Attacker Motivations

- Security adversaries are first and foremost human (at least for the time being).
- we can consider the purpose of attacks through the eyes of the people who carry them out.
- Doing so may better equip us to understand how we should respond:
 - proactively (during system design)
 - and reactively (during incidents)



Attacker Motivations

- To understand attacker motivations, we need to take into account the people themselves :
 - who they are,
 - whether they perform attacks for themselves or for someone else,
 - and their general interests.



Attacker Motivations

- Attack motivations:
 1. Fun
 - To undermine the security of a system for the sheer joy of knowing it can be done.
 2. Fame
 - To gain notoriety for showing off technical skills.
 3. Activism
 - To make a point or broadcast a message—typically, a political viewpoint—widely.
 4. Financial gain
 - To make money.



Attacker Motivations

5. Coercion

- To get a victim to knowingly do something they don't want to do.

6. Manipulation

- To create an intended outcome or change behavior—for example, by publishing false data (misinformation).

7. Espionage

- To gain information that might be valuable (spying, including industrial espionage). These attacks are often performed by intelligence agencies.

8. Destruction

- To sabotage a system, destroy its data, or just take it offline.



Attacker Motivations

- An attacker might be:
 - a financially motivated vulnerability researcher, government espionage agent, and criminal actor all at the same time!
- For example, in 2018 North Korean citizen accused of participating in a wide variety of activities on behalf of his government, including:
 - creating the infamous 2017 WannaCry Ransomware (used for financial gain),
 - the 2014 compromise of Sony Pictures (intended to coerce Sony into not releasing a controversial movie),
 - and the compromise of electric utilities (presumably for espionage or destructive purposes).



Attacker Motivations

- When designing systems, it's important to keep these diverse motivations in mind.
- Consider an organization that is processing money transfers on behalf of its customers.
 - If we understand why an attacker might be interested in this system, we can design the system more securely.



Attacker Motivations

- A good example of possible motivations in this case:
 - The activities of a group of North Korean government attackers.
 - Allegedly attempted to steal millions of dollars by breaking into banking systems and exploiting the SWIFT transaction system to transfer money out of customer accounts.



2. Attacker Profiles



Attacker Profiles

- Profiles of attackers:
 - indicating how they relate to a system.



Attacker Profiles: Hobbyists

- *Hobbyists*: curious technologists who wanted to understand how systems worked.
 - Taking computers apart or debugging their programs.
 - They discovered flaws that the original system designers hadn't noticed.
- Hobbyists are motivated by their thirst for knowledge;
- They hack for fun, and can be allies to developers looking to build resilience into a system.



Attacker Profiles: Vulnerability Researchers

- *Vulnerability researchers* use their security expertise professionally.
 - They enjoy finding security flaws as full-time employees, part-time freelancers, or even accidentally as average users who stumble across bugs.
 - Many researchers participate in Vulnerability Reward Programs, also known as *bug bounties*.



Attacker Profiles: Vulnerability Researchers

- Vulnerability researchers are typically motivated to make systems better.
 - they are important allies to organizations seeking to secure their systems.
 - They tend to operate within a set of predictable disclosure norms that set expectations between system owners and researchers about how vulnerabilities are discovered, reported, fixed, and discussed.



Attacker Profiles: Vulnerability Researchers

- Researchers operating under these norms avoid inappropriately accessing data, causing harm, or breaking the law.
- Typically, operating outside these norms invalidates the possibility of getting a reward and may qualify as criminal behavior.



Attacker Profiles: Vulnerability Researchers

- *Red Teams and penetration testers* attack targets with the permission of the system owner.
 - They look for ways to defeat system security with a focus on improving security and operate within a set of ethical guidelines.



Attacker Profiles: Governments and Law Enforcement

- Government organizations (for example, law enforcement agencies and intelligence agencies) may hire security experts to gather:
 - intelligence, police domestic crime, commit economic espionage, or complement military operations.
- In some cases, governments may turn to talented students fresh out of school, reformed attackers who have spent time in jail, or notable luminaries in the security industry.



Attacker Profiles: Governments and Law Enforcement

- Intelligence gathering
 - Intelligence gathering employs people who know how to break into systems.
 - In the past few decades, traditional spying techniques, including signals intelligence (SIGINT) and human intelligence (HUMINT), have modernized with the advent of the internet.



Attacker Profiles: Governments and Law Enforcement

- Intelligence gathering
 - For example, The security company RSA was compromised by experts associate with China's intelligence apparatus.
 - The attackers compromised RSA to steal cryptographic seeds for their popular two-factor authentication tokens.
 - Once they had these seeds, the attackers didn't need physical tokens to generate one-time authentication to log in to the systems of Lockheed Martin.



Attacker Profiles: Governments and Law Enforcement

- Military purposes
 - Governments may break into systems for military purposes, AKA cyber warfare or information warfare.
 - If a government wants to invade another country: Could they somehow attack the target's air defense systems and trick them into not recognizing an inbound air force?
 - Could they shut down their power, water, or banking systems?
 - Also, if a government wants to prevent another country from building or obtaining a weapon.
 - Could they remotely and stealthily disrupt their progress?



Attacker Profiles: Governments and Law Enforcement

- Military purposes
 - Governments may break into systems for military purposes, AKA cyber warfare or information warfare.
 - If a government wants to invade another country: Could they somehow attack the target's air defense systems and trick them into not recognizing an inbound air force?
 - Could they shut down their power, water, or banking systems?
 - Also, if a government wants to prevent another country from building or obtaining a weapon.
 - Could they remotely and stealthily disrupt their progress?



Attacker Profiles: Activists

- *Hacktivism* is the act of using technology to call for social change.
 - For the purpose of thinking about how to design systems, Hacktivists have been known to deface websites—that is, replace normal content with a political message.
 - This kind of attack can be very embarrassing for website owners and can undermine user trust in the site.



Attacker Profiles: Activists

- Other hacktivist attacks may be far more destructive.
 - They might took numerous websites offline through denial-of-service attacks.
 - As a result, anyone visiting the affected websites experienced slow service or an error.
 - On the more serious end of the spectrum, attackers may even threaten to destroy or sabotage systems entirely, inspiring some researchers to label them cyberterrorists..



Attacker Profiles: Criminal Actors

- Attack techniques are used to carry out crimes that closely resemble their nondigital cousins—for example, committing identity fraud, stealing money, and blackmail.
- Criminal actors have a wide range of technical abilities.
 - Some may be sophisticated and write their own tools.
 - Others may purchase or borrow tools that other people build, relying on their easy, click-to-attack interfaces.



Attacker Profiles: Criminal Actors

- **Social engineering** the act of tricking a victim into aiding you in the attack—is highly effective despite being at the lowest end of difficulty.
- The only barriers to entry for most criminal actors are:
 - a bit of time, a computer, and a little cash.



Attacker Profiles: Criminal Actors

- attackers have also realized that victims will hand over money when their sensitive data is threatened.
- Ransomware is software that holds a system or its information hostage (usually by encrypting it) until the victim makes a payment.
- Commonly, attackers infect victim machines with this software by exploiting vulnerabilities, by packaging the ransomware with legitimate software, or by tricking the user into installing it themselves



Protecting your systems from criminal actors:

- consider which systems they might target, and how to make their attacks expensive.
- The evolution of Completely Automated Public Turing test (**CAPTCHA**) systems is a good example of how to increase the cost of attacks over time.
- **CAPTCHAs** are used to determine whether a human or an automated bot is interacting with a website.



Attacker Profiles: Insiders

- Every organization has insiders: current or former employees who are trusted with internal access to systems.
- A person becomes an insider threat when they are able to perform actions that could result in harm to the organization.



Attacker Profiles: Insiders

Table 2-1. General categories of insiders and examples

First-party insiders	Third-party insiders	Related insiders
Employees	Third-party app developers	Friends
Interns	Open source contributors	Family
Executives	Trusted content contributors	Roommates
Board directors	Commercial partners	
	Contractors	
	Vendors	
	Auditors	



Attacker Profiles: Designing for insider risk:

1. Least privilege

- Granting the fewest privileges necessary to perform job duties, both in terms of scope and duration of access.

2. Zero trust

- Designing automated or proxy mechanisms for managing systems so that insiders don't have broad access that allows them to cause harm.

3. Multi-party authorization

- Using technical controls to require more than one person to authorize sensitive actions.



Attacker Profiles: Designing for insider risk:

4. **Auditing and detection**

- Reviewing all access logs and justifications to make sure they're appropriate.

5. **Recoverability**

- The ability to recover systems after a destructive action, like a disgruntled employee deleting critical files or systems.



Attacker Methods :

- Trying to predict what any particular attacker might do on any given day is unfeasible because of the variety of attack methods available.
- A few frameworks for studying attacker methods:
 1. *Threat intelligence*,
 2. *Cyber kill chains*,
 3. *Tactics, Techniques, and Procedures (TTPs)*.



Attacker Methods: Threat Intelligence

- *threat intelligence* can help system defenders understand how real attackers are working every day and how to repel them.
- Threat intelligence comes in multiple forms, each serving a different purpose:
 1. *Written reports*
 2. *Indicators of compromise*
 3. *Malware reports*



Attacker Methods: Cyber Kill Chains

- Lay out all the possible steps that an attacker may have to take to achieve their goals.
- This frameworks can help you plot the formal progression of an attack alongside defensive controls to consider



Attacker Methods: Cyber Kill Chains

Table 2-3. Cyber Kill Chain of a hypothetical attack

Attack stage	Attack example	Example defenses
<i>Reconnaissance:</i> Surveilling a target victim to understand their weak points.	Attacker uses a search engine to find the email addresses of employees at a target organization.	Educate employees about online safety.
<i>Entry:</i> Gaining access to the network, systems, or accounts necessary to carry out the attack.	Attacker sends phishing emails to employees that lead to compromised account credentials. The attacker then signs in to the organization's virtual private network (VPN) service using those credentials.	Use two-factor authentication (such as security keys) for the VPN service. Only permit VPN connections from organization-managed systems.
<i>Lateral movement:</i> Moving between systems or accounts to gain additional access.	Attacker remotely logs in to other systems using the compromised credentials.	Permit employees to log in to only their own systems. Require two-factor authentication for login to multiuser systems.



Attacker Methods: Tactics, Techniques, and Procedures

- TTPs is a common way of cataloging attack methods
- The ATT&CK framework expands each stage of the cyber kill chain into detailed steps and provides formal descriptions of how an attacker could carry out each stage of an attack.
- The ATT&CK framework lays out hundreds of ways attackers can operate so that defenders can build defenses against each attack method.



Risk Assessment Considerations

- The following considerations important when assessing the risk posed by various attackers:
 1. You may not realize you're a target.
 2. Attack sophistication is not a true predictor of success
 3. Don't underestimate your adversary.
 4. Attribution is hard.
 5. Attackers aren't always afraid of being caught.



Summary

- All security attacks can be traced back to a motivated person
- Assess who might want to target you. What are your assets? Who buys your products or services?
- Stay current on the threat intelligence issued by security firms
- Be mindful of complex attack strategies



Thank You



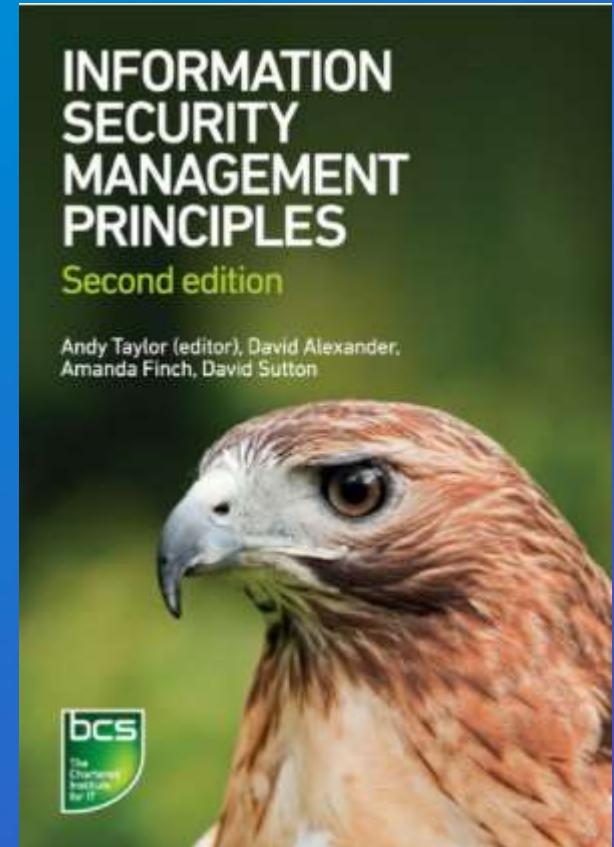


الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432

Introduction to Cyber Security and Digital Crime

Information Security Management Principles

by David Alexander, David Sutton, Andy Taylor, Amanda Finch



Week 10

Chapter 6: SOFTWARE DEVELOPMENT AND LIFE CYCLE



Contents

1. Testing, audit and review
2. Systems development and support



Objectives

- Understanding of the issues surrounding security of the IT infrastructure.
- Gain an understanding of the importance and appropriate audit and review processes and of effective change control and configuration management.
- Learn about the differences in security between open source and proprietary solutions, commercial off-the-shelf software and bespoke systems.
- learn about some of the techniques involved in reducing the security risks in the development of code



Required Reading

1. Chapter 6 from Information Security Management Principles



1. Testing, audit and review



Methods and strategies for security testing systems

- A single test after completion is not sufficient, as threats and business requirements are constantly changing.
- Tests and reviews should be repeated at periodically.
- Some of this requires expert testing by a professional penetration test team.
- Some of it requires a review by a combination of business and security analysts.
- An independent external consultant can help to identify areas that may have been overlooked or about which the internal team have limited knowledge.



Methods and strategies for security testing systems

- The continued review and analysis of vulnerabilities in systems is a critical element.
- Vulnerabilities are uncovered through a number of mechanisms including:
 - The results of penetration tests
 - The analysis of viruses
 - Specialist vulnerability analysts



Methods and strategies for security testing systems

- Vulnerabilities are uncovered through a number of mechanisms including:
 - The results of penetration tests
 - The analysis of viruses
 - Specialist vulnerability analysts
- A policy should be implemented to direct the actions for dealing with vulnerabilities.
- A poor patching policy will lead to a major security issues.



Correct reporting of testing and reviews

- The test and review process requires accurate and comprehensive reporting.
- The report must be an open and honest ‘warts and all’ report.
- it needs to highlight any shortcomings in the security architecture.
- Any attempt to hide or downplay problems may lead to vulnerabilities being left in place that can be exploited successfully.



Correct reporting of testing and reviews

- The report must include:
 - detailed technical content
 - an executive summary.
 - This summary must contain the ‘take-away’ messages and important conclusions, along with a brief justification for further action and expenditure.
 - It is recommended to give the report some level of protective marking to prevent unauthorized access.

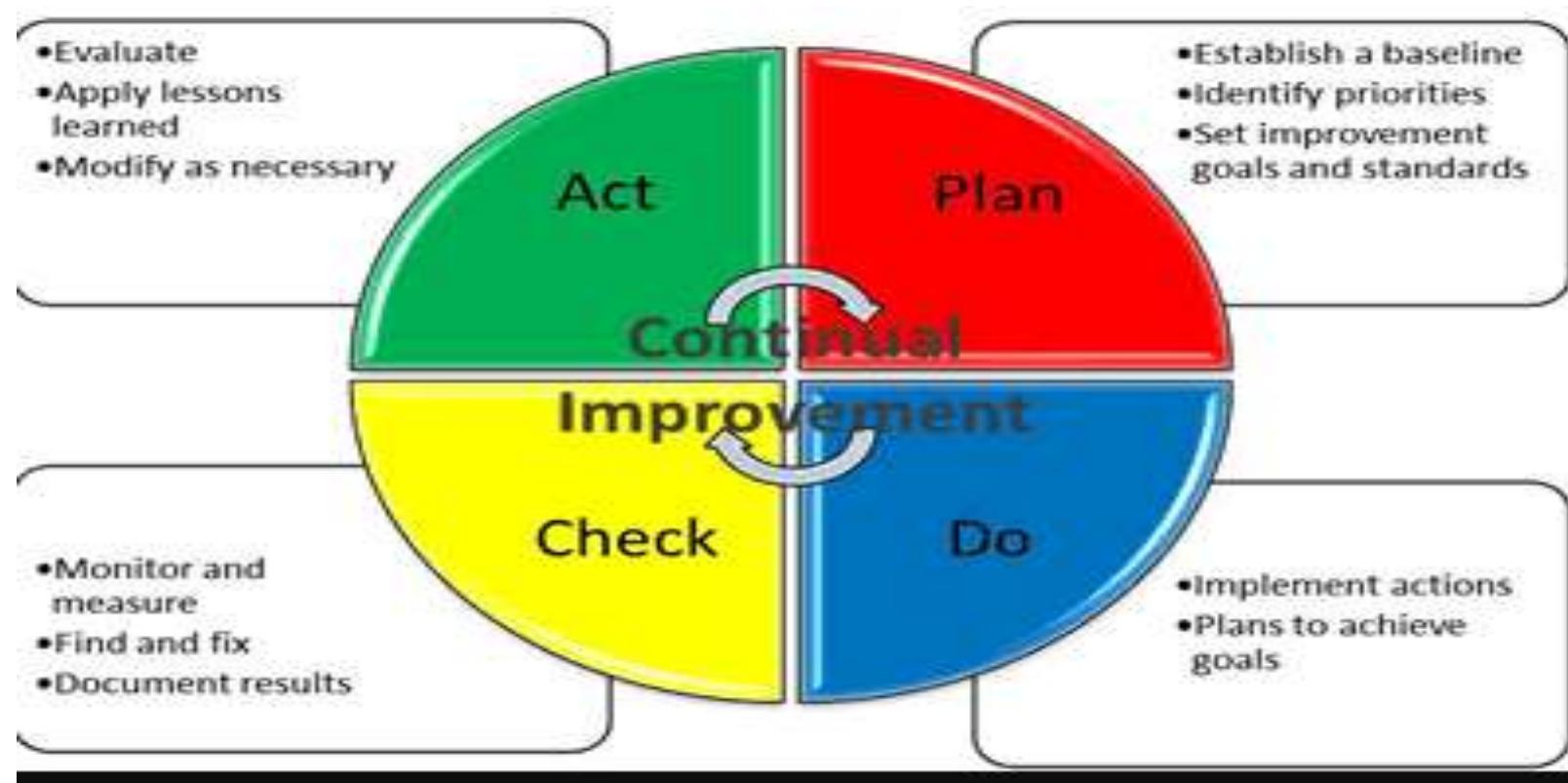


Verifying the links between IT and clerical processes

- It is important to align the information security architecture, policy and procedures with the needs of the business and its primary operational processes.
- ‘*Plan–Do–Check–Act*’ –PDCA- cycle is where to check that the task has been done properly and that the basis for the original design has not changed since the last review.
 - PDCA cycle show if people are following the procedures.
 - It checks that the procedures are correct and current.



Plan–Do–Check–Act



<http://iso14001certification.com/plan-check-act/>



Monitoring system and network access or usage

- There is a need to collect event log data from a whole range of systems, and devices.
- Monitoring the traffic passing over the network and any external data links, such as the internet is essential.
 - There are commercial devices and software applications that can be used to perform this role.
 - Many organizations will keep six months' of event log data for just such eventualities.



Monitoring system and network access or usage

- The collected data can be analyzed to detect unusual patterns of behavior, malware and signatures of known attacks.
- It can also be *reviewed forensically* to gather evidence of wrongdoing and abuse that can be used in an internal disciplinary case or provided to criminal justice.
- The analysis needs to be performed by well-trained and skilled individuals.
 - The training must not only cover the technical side of recognizing unusual activity, but also how to collect and preserve data in such a way that it is legally admissible in court.



Monitoring system and network access or usage

- Audit will be done for both internal and third party.
- This audit is for both the technical services and the quality of service against defined service level agreements (SLAs).
- It is also important to check that the work done does meet the requirements of the organization, so some internal auditing must be done.



2. SYSTEMS DEVELOPMENT AND SUPPORT



Security requirement specification

- The design of any system must meet the operational requirements and also be aligned with the information security architecture of the organization.
- It is most important that the assurance requirements are captured at the start of any project in order to ensure that they are effective.
- If the project team is trying to reduce the security requirements to save time and money, it is important to sign off acceptance of the increased risk that results from any changes.



Security requirement specification

- Security is not to defend against improper access and misuse only, but it also means:
 1. defensive coding to make sure that only valid and accurate data is processed by the system;
 2. proper functional testing to ensure it behaves as expected and within the design criteria;
 3. methods to back up and secure data against loss or damage;
 4. adequate assurance of availability;
 5. compliance with any legal and regulatory requirements;
 6. security of communications;
 7. effective auditing of activity.



Security involvement in system and product assessment

- All new systems should have to go through some form of appropriate acceptance testing before being used in production.
- It does not matter if they were developed in-house or purchased;
- They should be assessed for acceptable and appropriate levels of security.
- For example, a product bought from a reputable supplier should be given more trust than a piece of freeware written by someone you have never heard of, downloaded from a website.



Security involvement in system and product assessment

- Every product should be considered for its potential effects on confidentiality, integrity and availability.
- Many ‘best practice’ organizations maintain a separate test environment that replicates the live systems to allow assessments to be conducted without risk of adverse impact.
- Another approach is to examine the source code (not always practical) by eye or with automated tools.
- Use of a malware scanner is always recommended for new code.



Security issues associated with commercial off-the-shelf products

- The most obvious threat is of rogue code hidden within an application that performs activity against the best interests of the organization.
- It could also be that there are ‘bugs’ that, while not intentionally malicious, have a serious adverse impact.
- It is recommended to have *separate test environment*:
 - To help find any such code by identifying its behavior before it affects production assets.



Security issues associated with commercial off-the-shelf products

- Sometimes dishonest people will advertise cheap copies of applications
 - Because they have altered the code to include malware.
- The reduced price means it is more likely to be purchased and their malware installed.



Security issues associated with commercial off-the-shelf products

- The security issues don't just mean checking for rogue code.
- It is also important to check that the product is a legal copy and not pirated.
- Make sure that the supplier is reputable, not some dubious market stall selling cheap copies.
- Failure to buy genuine copies can lead to financial penalties, impact on operations and loss of reputation.



Separation of development and live systems areas

- The main reason for keeping the live and development systems separate is to protect the live data from any unintended actions that might compromise it.
- Work to develop new systems and applications almost always contains mistakes in coding or design.
- Any attempt to run unproven and incomplete code against a live database could have a major impact on the ability of the organization to function.
- It is often considered best practice to have three separate systems for
 1. development,
 2. test
 3. live.



Security of acceptance processes and authorization for use

- Once a deliverable system has completed development and is ready for deployment, it must be tested to make sure it does exactly what the requirements specify.
- If the product is an update of an existing product, there must also be regression testing to make sure that no unexpected changes to existing functionality.
- This includes testing the security aspects of the product and also ensuring that the testing is conducted securely.



Security of acceptance processes and authorization for use

- Security testing to consider includes:
 1. effectiveness of defensive coding;
 2. protection against malware and code injection through interfaces;
 3. backup and recovery of data;
 4. access control;
 5. auditing and behavioral analysis;
 6. communications security;
 7. resilience.



Accreditation for new and modified systems

- accreditor, who is responsible for ensuring that any changes or additions to information systems and networks are of a required standard from a security standpoint specifically.
- This person has to approve the information security architecture, policy and procedures before the product(s) can be deployed and used.
- Normally this process is supported by formal documentation to standards defined in government documents.



Accreditation for new and modified systems

- An alternative approach is where a new system needs to be capable of accreditation to a standard such as the ISO/IEC 27000 series.
- It may be that the organization already has the accreditation, or is working towards it,
- The organization wants to ensure that the new system is capable of meeting the required standards for controls and countermeasures so that they will pass audit without remedial action.



Accreditation for new and modified systems

- The same principle applies to existing systems that are modified or updated.
- All changes should go through the same review process to make sure that the standards defined when the system was new are being maintained in the latest work.
- Many organizations also require periodic review and re-accreditation even if there does not appear to have been any change.
- Sometimes users will make changes in design or working practices without permission, or the environment changes (e.g. new threats and technology).
- Periodic review will help to identify these and formal processes can then be used to take remedial action.



Change control for software integrity

- Any change to a software application, while designed to enhance its functionality, can introduce unintended problems.
- Every organization should implement and enforce an effective formal change control process to manage the risks to their information assets and reputation.



Change control for software integrity

- RFC requirements:
 - submission of an outline of the proposed changes
 - Assessments of the benefits against the risks
 - specify certain conditions and approaches to be used in order to manage the risks
 - the new version must undergo regression and functionality testing



Security issues arising from outsourcing software development

- The practice of outsourcing has become more widespread.
- It drives down costs, but it can also introduce new risks to the process.
 - Some of these risks carry security implications, such as the introduction of malicious code.
 - There is also the risk that there will be a loss of intellectual property through the information that has to be given to the third party.
 - A similar risk applies to any data sent to the third party.
 - The laws on the protection of data apply to anything sent to a third party as part of the development process.



Security patching

- Every software application and operating system contains bugs.
- The complexity and length of the code makes it impossible to test every single execution path through it completely.
- These bugs can have different impacts, ranging from incorrect values being stored in a database to allowing unauthorized access to the system.
- They will have some form of adverse impact on the confidentiality, integrity or availability



Security patching

- When bugs are found, the supplier will normally issue a patch that can be installed in order to remove the vulnerability.
- These patches need to be ***tested and installed*** at the earliest opportunity.
- Hackers will also download the patches and attempt to reverse-engineer them in order to exploit the vulnerability if they can.



Use of certified products and systems

- When use software products (e.g. firewalls), hardware devices (e.g. network switches) and operating systems:
 - They have to be formally certified as providing a minimum standard of security, safety, reliability or a combination of these.



Use of certified products and systems

- Common Criteria' assessment scheme provides a scale of product assurance, ranging from EAL (Evaluation Assurance Level) 1 to 7.
 - The higher the number, the greater the level of assurance.
 - The concept is that an assured product can help to formally reduce risk in a quantifiable way when designing security architectures.
 - Each product will have a 'security target' of the features and functions that have been assessed.
 - It is very important to make sure that the features you plan to use are included within that target, otherwise the certification is of no value.



Use of escrow to reduce risks of loss of source code

- If source code has been provided by a third-party, the customer is dependent on that supplier for support, updates and changes to their software.
- If the supplier went out of business, or was sold to a competitor:
 - This may force the client to spend extra money to resolve or get support if something went wrong.
 - The solution is, the supplier and customer agree on a neutral third party (often a firm of lawyers or a bank) who will hold a copy of the source code and development materials.
 - There is a legally binding agreement that specifies the circumstances under which the third party will release the material to the customer.



Summary

- We discuss security issues that arise from the development, testing and implementation of new software.
- The ongoing life cycle of software is also a concern and is addressed.
- The design of any system must meet the operational requirements
- Every software application and operating system contains bugs, a patch can be installed in order to remove the vulnerability.
- separate test environment can help find any bad code by identifying its behavior before it affects production assets.



Thank You



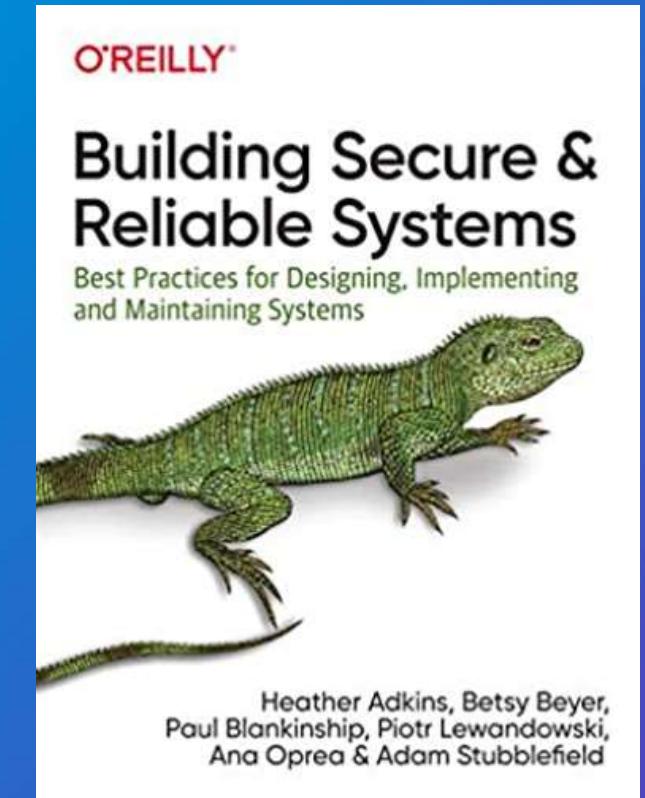


الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432

Introduction to Cyber Security and Digital Crime

Building Secure & Reliable Systems Best Practices for Designing, Implementing and Maintaining Systems

by Heather Adkins, Betsy Beyer, Paul
Blankinship, Piotr Lewandowski, Ana
Oprea & Adam Stubblefi



Week 11

Chapters 4 & 5: Design Tradeoffs and Design for Least Privilege



Contents

1. Design Objectives and Requirements
2. Balancing Requirements
3. Managing Tensions and Aligning Goals
4. Initial Velocity Versus Sustained Velocity
5. Concepts and Terminology
6. Classifying Access Based on Risk
7. Best Practices
8. A Policy Framework for Authentication and
9. Authorization Decisions
10. Advanced Controls
11. Tradeoffs and Tensions



Objectives

- Recognize the importance of reviewing system's security and reliability needs as early as possible in the software design phase.
- Define functional requirements that satisfy particular needs
- Define requirements that are related to security and reliability
- Mitigating security concerns when developing an application
- Understanding the security concepts related to access controls
- Applying classification to mitigate risk associated with access controls
- Applying best practices when designing a system



Required Reading

1. Chapters 4&5 from Building Secure & Reliable Systems Best Practices for Designing, Implementing and Maintaining Systems.



Chapter 4 : Design Tradeoffs



Design Objectives and Requirements



Feature Requirements1

- Functional requirements:
 - Identify the primary function of a service or application.
 - Describe how a user can accomplish a particular task or satisfy a particular need.
 - Expressed in terms of use cases, user stories, or user journeys—sequences.
 - Critical requirements are the subset of feature requirements.
- If a design does not satisfy a critical requirement or critical user story: No viable product.



Feature Requirements2

- Nonfunctional Requirements: Focus on general attributes or behaviors of the system
- Nonfunctional requirements are relevant to :
 - focus—security
 - Reliability
- Example:
 - What are the exclusive circumstances under which someone may have access to certain data?
 - What are the service level objectives (SLOs) for metrics?
 - How does the system respond under load above a certain threshold?



Feature Requirements3

- Broader areas of nonfunctional Requirements include:

- Development efficiency and velocity
- Selected implementation language
- Application frameworks
- Testing processes
- Build processes
- How to iterate on new features efficiently ?
- How to understand and modify or debug existing code efficiently?



Feature Requirements....4

■ Features Vs. Emergent Properties:

- Feature requirements usually exhibit :
 - A fairly straightforward connection between the requirements,
 - The code that satisfies those requirements
 - Tests that validate the implementation.
- Example:
 - Specification - User story or requirement
 - Implementation - A web or mobile application based on the specification
 - Validation - An integration test that walks through the specified user story step by step.
- Nonfunctional requirements—like reliability and security requirements : More difficult to pin down



Balancing Requirements....1

- Interact both with implementations of feature requirements and with each other.
- Difficult to reason about tradeoffs involving security and reliability as a standalone topic.
- Examining examples :
 - Illustrates the kinds of tradeoffs to consider.
 - Delve quite deeply into technical details,
 - Illustrate the complex interdependencies between requirements.
- The objective is to think about the process that goes into designing a system with complex security and reliability requirements.



Balancing Requirements....2

- Example: Payment Processing
 - build an online service that sells widgets to consumers.
 - Online widgets catalog by using a mobile or web application
 - Purchase the chosen widgets, which requires that they provide details for a payment method.
 - Security and reliability considerations:
 - Accepting payment information introduces significant security and reliability considerations (Names, addresses, and credit card numbers, etc.)
 - Compliance with industry-level or regulatory security standards such as PCI DSS.
 - A compromise of the sensitive user information can have serious consequences for the project.



Managing Tensions and Aligning Goals....1

- Satisfy important nonfunctional requirements: security and reliability
- Nonfunctional requirements are very much aligned with general software quality attributes.
- Example: Microservices and the Google Web Application Framework
 - Define strong contracts between the various microservices.
 - Allow for independent deployment cycles, including rollback.
 - Minimize test automation and quality-assurance overhead.
 - Improve clarity of logging and monitoring.
 - Provide fine-grained cost accounting.
 - Increase overall application scalability and reliability.



Managing Tensions and Aligning Goals....2

▪ Initial Velocity Versus Sustained Velocity

- Not introducing critical requirements like security, reliability, and maintainability early in the project cycle may indeed increase the project's velocity.
- The late-stage cost of updating a design to accommodate the requirements manifested by emerging properties can be very important.
- Making invasive late-stage changes to address security and reliability risks can in itself introduce even more security and reliability risks.
- Example : Lack of security in IP, TCP, DNS, HTTP and other Internet protocols initiates the design of secure versions like Ipsec, SSL, HTTPS,etc.



Chapter 5 : Design for Least Privilege



Design for Least Privilege

- Companies often want to assume their engineers have the best intentions and perform Herculean tasks. This is a wrong expectation.
- Possibility of damage and the right expectation is to ask :
 - What could they do?
 - How would they do it?
 - What's the worst mistake they could make?
 - etc.
- Because we can't rely on human perfection, we must assume that any possible bad action or outcome can happen.
 - It is recommended to design the system to minimize or eliminate the impact of these bad actions
 - Needs to limit the human privilege and the trust we place in their credentials.



Concepts and Terminology



Least Privilege

- Least privilege is a broad concept that's well established in the security industry.
- Assign the least privilege for any given task or action path.
- This goal applies to :
 - Humans
 - Automated tasks,
 - Individual machines from a distributed system
- The objective of least privilege should extend through all authentication and authorization layers of the system.



Zero Trust Networking

- Eliminates the concepts of trust based on network location within a perimeter.
- Leverages device and user claims to get access to data and resources.
- Identify provider to keep track of users and user-related information.
- Policy evaluation service to determine if a user or device conforms to the policy set forth by security administrators.
- Access proxy that utilizes the above signals to grant or deny access to an organizational resource.



Zero Touch

- Build upon the concept of least privilege through automation, with the goal of moving to what we call Zero Touch interfaces.
- The specific goal of these interfaces:
 - Make the application/system safer and reduce outages by removing direct human access to production roles.
 - humans have indirect access to production through tooling and automation that make predictable and controlled changes to production infrastructure.
- This approach requires extensive automation, new safe APIs, and resilient multi-party approval systems.



Classifying Access Based on Risk....1

- Any risk reduction strategy comes with tradeoffs.
- Reducing the risk introduced by human actors likely entails additional controls or engineering work.
 - Can introduce tradeoffs to productivity;
 - It may increase engineering time, process changes, operational work, or opportunity cost.
- Limit these costs by clearly scoping and prioritizing what you want to protect.
- Not all data or actions are created equal, and the makeup of the access may differ dramatically depending on the nature of your system.
 - Protect all access to the same degree.
 - Classify access based on impact, security risk, and/or criticality.



Classifying Access Based on Risk....2

- Example :

- Likely need to handle access differently to different types of data :
 - publicly available data
 - company data
 - user data
 - cryptographic secrets
 - Likely need to treat administrative APIs that can delete data differently than service-specific read APIs.



Classifying Access Based on Risk....3

- Classifications should be clearly defined, consistently applied, and broadly understood so people can design systems and services that “speak” that language.
- Classification framework will vary based on the size and complexity of the system:
 - Need only two or three types that rely on ad hoc labeling
 - Need a robust and programmatic system for classifying parts of the system (API groupings, data types) in a central inventory.
- Ensure that the framework can handle the most important entities within the systems.



Classifying Access Based on Risk....4

- After establishing a foundation of classification, consider several dimensions:
 - Who should have access?
 - How tightly should that access be controlled?
 - What type of access does the user need (read/write)?
 - What infrastructure controls are in place?
- Example:
 - A company may need three classifications: public, sensitive, and highly sensitive.
 - It might categorize security controls as: low risk, medium risk, or high risk by the level of damage the access can allow if granted inappropriately.



Classifying Access Based on Risk....5

- Example access classifications based on risk:

	Description	Read access	Write access	Infrastructure access ^a
Public	Open to anyone in the company	Low risk	Low risk	High risk
Sensitive	Limited to groups with business purpose	Medium/high risk	Medium risk	High risk
Highly sensitive	No permanent access	High risk	High risk	High risk



Best practices



Small Functional APIs

- Unix culture centers around small and simple tools that can be combined.
- To adapt this culture to the current computing environment :Make each API endpoint do one thing well.
- When building systems with an eye toward security and reliability:
 - avoid open-ended interactive interfaces—instead
 - design around small functional APIs
- This approach allows to apply the classic security principle of least privilege and grant the minimum permissions necessary to perform a particular function.



Breakglass

- A breakglass mechanism provides access to your system in an emergency situation and bypasses your authorization system completely.
- This can be useful for recovering from unforeseen circumstances.
- When employing a breakglass mechanism, consider the following guidelines:
 - It should be highly restricted.
 - It should be available only from specific locations.
 - All uses of a breakglass mechanism should be closely monitored.
 - It should be tested regularly by the team(s) responsible for production services to make sure it functions when you need it



Auditing

- Serves to detect incorrect authorization usage.
- Examples:
 - Malicious system operator abusing their powers.
 - A compromise of a user's credentials by an external actor.
 - Rogue software taking unexpected actions against another system.
- The ability to audit dependents on the design of the systems:
 - How granular is the access control decision being made or bypassed?
 - How clearly can you capture the metadata associated with the request?



Testing and Least Privilege

- Testing has two important dimensions with regard to least privilege:
 - Ensure that access is properly granted only to necessary resources
 - Ensure that the infrastructure for testing has only the access it needs
- Least privilege: well-defined user profiles have enough privileges to perform their role, but no more.
- The infrastructure allows to do the following:
 - Describe what a specific user profile needs to be able to do in their job role.
 - Describe a set of scenarios in which the user profile attempts an action on the system.
 - Run these scenarios and compare the actual result/impact against the expected result/impact.



Diagnosing Access Denials

- Least privilege policy is enforced at multiple levels (multi-party authorization)
- In case of authorization system denies access, one of three possible outcomes might occur:
 - The client was correctly denied and your system behaved appropriately.
 - The client was correctly denied, but can use an advanced control to obtain temporary access.
 - The client believes they were incorrectly denied, and potentially files a support ticket with your security policy team.



Worked Example: Configuration Distribution



Worked Example: Configuration Distribution

- The best practices for managing a configuration file are to:
 - Store the configuration file in a version control system.
 - Code review changes to the file.
 - Automate the distribution

	POSIX API via OpenSSH	Software update API	Custom OpenSSH ForceCommand	Custom HTTP receiver
API surface	Large	Various	Small	Small
Preexisting^a	Likely	Yes	Unlikely	Unlikely
Complexity	High	High	Low	Medium
Ability to scale	Moderate	Moderate, but reusable	Difficult	Moderate
Auditability	Poor	Good	Good	Good
Can express least privilege	Poor	Various	Good	Good

APIs that update web server configuration and their tradeoffs



POSIX API via OpenSSH

- Automation to connect to the web server host via OpenSSH.
- The automation can then write the configuration file and restart the web server process.
- Leveraging the large preexisting administrative API introduces several risks:
 - The role running the automation can :
 - Stop the web server permanently
 - Start another binary in its place
 - Read any data it has access to
 - Bugs in the automation implicitly have enough access to cause a coordinated outage of all of the web servers.
 - A compromise of the automation's credentials is equivalent to a compromise of
 - all of the web servers.



Software Update API

- Many ways to package and trigger binary updates, using APIs of varying sizes.
- Examples:
 - Debian package (.deb) pulled from a central repository by a periodic apt-get called from cron (simple).
 - Build an application using one of the patterns instead of cron: reuse for both the configuration and the binary (more complex).
- Sometimes the needs of binary and configuration update systems don't align:
 - Use two distribution mechanisms: one for binaries, another for configuration updates.



Custom OpenSSH ForceCommand

- Short script should be written to perform the following steps:
 - Receive the configuration from STDIN.
 - Sanity check the configuration
 - Restart the web server to update the configuration
- Expose this command via OpenSSH by tying particular entries in an authorized_keys file with the ForceCommand option.
- Implement as many of these unique key/ForceCommand combinations:
 - This pattern can be hard to scale to many unique administrative actions.



Custom HTTP Receiver

- **Sidecar :**

- Write a small sidecar daemon—much like the ForceCommand solution, but using another AAA mechanism that accepts a config.
 - This approach doesn't require modifying the serving binary and is very flexible.

- **In-Process:**

- Modify the web server to expose an API to update its config directly, receiving the config and writing it to disk.
 - One of the most flexible approaches, and bears a strong similarity to the way we manage configuration at Google.
 - It requires incorporating the code into the serving binary.



Tradeoffs

- An attacker can compromise the web server's role by pushing an arbitrary config;
- Choosing a smaller API means that the push mechanism won't implicitly allow that compromise.
- Further design for least privilege by signing the config independently from the automation that pushes it.
- This strategy segments the trust between roles:
 - If the automation role pushing the configuration is compromised, the automation cannot also compromise the web server by sending a malicious config.
 - Solution : Design each piece of the system to perform one task to isolate trust.



A Policy Framework for Authentication and Authorization Decisions



Using Advanced Authorization Controls

- An access control list is a common way to implement an authorization decision.
- The simplest ACL is a string matching the authenticated role.
- More complex authorization requirements:
 - Multi-Factor Authorization (MFA)
 - Multi-Party Authorization (MPA)
- Advised to separate the complexities of authorization decisions from core API design and business logic with frameworks.
- The security policy framework allows the code to make simple checks.
 - Example: Can X access resource Y?



Investing in a Widely Used Authorization Framework

- Enable authentication and authorization changes at scale by using:
 - A shared library to implement authorization decisions
 - A consistent interface as widely as possible.
- Applying this classic modular software design advice in the security sphere yields surprising benefits.
- Example:
 - Add support for MFA or MPA to all service endpoints with a single library change.
 - Implement this support for a small percentage of the actions or resources in all services with a single configuration change.
 - Improve reliability by requiring MPA for all actions that allow a potentially unsafe action (code review system)
- Improve security against insider risk threats :
 - Facilitating fast incident response
 - Don't allow broad unilateral access.
- A uniform authorization framework facilitates team mobility.



Avoiding Potential Pitfalls

- Designing a complex authorization policy language is difficult.
 - If the policy language is too simplistic it won't achieve its goal
 - If the policy language is too general, it can be very hard to reason about.
- To mitigate these concerns: Apply standard software API design practices: iterative design approach
- Application developers will need assistance with the policy decisions that will be encoded in this language.
 - Require collaboration between application developer to craft the right balance between security and functionality



Advanced Controls



Multi-Party Authorization (MPA)

- Involving another person is one classic way to ensure a proper access decision, fostering a culture of security and reliability.
- This strategy offers several benefits:
 - Preventing mistakes or unintentional violations of policy that may lead to security or privacy issues.
 - Discouraging bad actors from attempting to perform malicious changes.
- Includes both employees, who risk disciplinary action, and external attackers, who risk detection.
 - Increasing the cost of the attack
 - Auditing past actions for incident response or postmortem analysis
 - Providing customer comfort.



Three-Factor Authorization (3FA)

- MPA often has one key weakness that can be exploited by a determined and persistent attacker.
- Mitigating the risk that a single compromised platform can undermine all authorization requires the following:
 - Maintaining at least two platforms
 - The ability to approve requests on two platforms
 - (Preferably) The ability to harden at least one platform
- Another option is to require authorization from a hardened mobile platform for certain very risky operations.
- 3FA protects against broad compromise of internal workstations, but does not provide any protection against insider threats when used in isolation.



Temporary Access

- Used to limit the risk of an authorization decision
- Can be useful when fine-grained controls are not available for every action, but still grantee the least privilege with the available tooling.
- Can be used in a structured and scheduled way or in an on-demand fashion.
- Combine temporary access with a request for :
 - Multi-party authorization,
 - Business justification,
 - Authorization control.
- Temporary access also creates a logical point for auditing.
- Temporary access also reduces ambient authority.



Proxies

- A monitored and restricted proxy machine (or bastion) can be used if fine-grained controls for backend services are not available.
- Only requests from specified proxies are allowed to access sensitive services.
- A proxy can :
 - Restrict dangerous action
 - Rate limit actions
 - Perform more advanced logging.
- Introduce restrictions or additional controls that mitigate the risk.
- Implementing any of these controls comes with an integration and operational cost.



Tradeoffs and Tensions



Increased Security Complexity

- A highly granular security posture is a very powerful tool, but it's also complex and therefore challenging to manage.
- Comprehensive set of tooling and infrastructure :
 - Define, manage, analyze, push, and debug security policies.
- Answer these foundational questions:
 - “Does a given user have access to a given service/piece of data?”
 - “For a given service/piece of data, who has access?”



Impact on Collaboration and Company Culture

- While a strict model of least privilege is likely appropriate for sensitive data and services, a more relaxed approach in other areas can provide tangible benefits.
- Example: providing software engineers with broad access to source code carries a certain amount of risk.
- Including source code and related artifacts in the data classification:
 - Protect sensitive assets
 - Visibility into less sensitive assets



Quality Data and Systems That Impact Security

- In a zero trust environment every granular security decision depends on two things:
 - the policy being enforced
 - the context of the request.
- Example - The data might include :
 - the role of the user,
 - the groups the user belongs to,
 - the attributes of the client making the request,
 - the training set fed into a machine learning model
 - the sensitivity of the API being accessed.
- Low-quality data will result in incorrect security decisions.



Impact on User Productivity

- The users need to be able to accomplish their workflows as efficiently as possible.
- The best security posture is ignored by the end user.
- Introducing new three-factor and multi-party authorization steps may impinge on user productivity, especially if users must wait to be granted authorization.
- Minimize user pain by making sure the new steps are easy to navigate.
- End users need a simple way to make sense of access denials.



Impact on Developer Complexity

- Developers must conform to model for least privilege.
- Provide developers easy and fast access to security engineers for security reviews and general consulting.
- Deploying third-party software in this environment requires particular care.



Summary

- The design process should involve, in the early stages of a project, numerous tradeoffs between security, reliability, and feature requirements.
- “Deal with them later”—but doing so often comes at significant cost and risk to the project.
- When designing a complex system, the least privilege model is the most secure way to ensure that clients have the ability to accomplish what they need to do, but no more.
- This is a powerful design paradigm to protect the systems and the data from malicious or accidental damage caused by known or unknown users.



Thank You



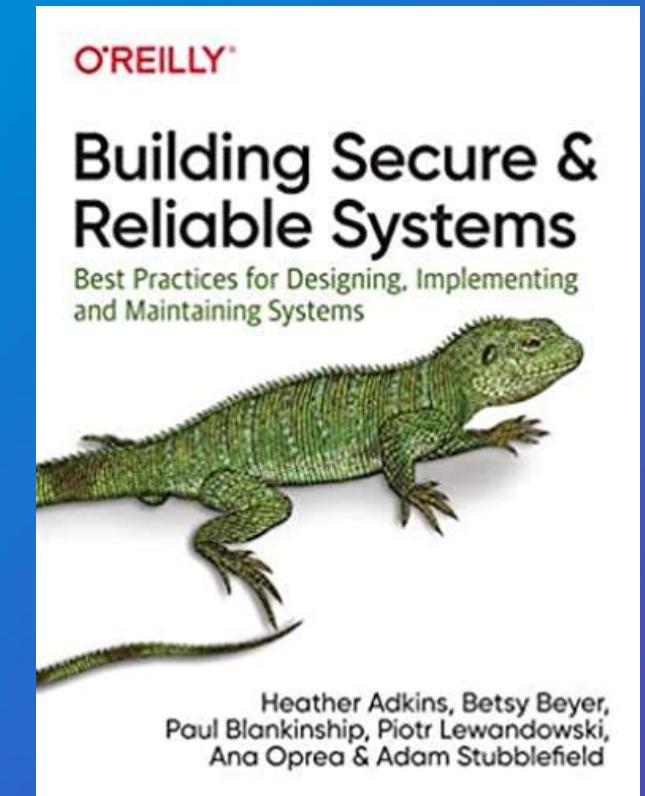


الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432

Introduction to Cyber Security and Digital Crime

Building Secure & Reliable Systems Best Practices for Designing, Implementing and Maintaining Systems

by Heather Adkins, Betsy Beyer, Paul
Blankinship, Piotr Lewandowski, Ana
Oprea & Adam Stubblefi



Week 12

Chapter 12: Writing Code



Contents

1. Frameworks to Enforce Security and Reliability
2. Common Security Vulnerabilities
3. Lessons for Evaluating and Building Frameworks
4. Simplicity Leads to Secure and Reliable Code
5. Security and Reliability by Default



Objectives

- Identify code vulnerability
- Apply software development patterns to enforce desired security properties
- Apply and mitigate typical reliability anti-patterns
- Know how to simplify code
- Select the right tools ,language, framework, and libraries



Required Reading

1. Chapter 12 from Building Secure & Reliable Systems Best Practices for Designing, Implementing and Maintaining Systems.



Frameworks to Enforce Security and Reliability



Benefits of Using Frameworks....1

- Most applications have similar building blocks for security :
 - authentication
 - authorization
 - logging
 - data encryption)
 - reliability (rate limiting, load balancing, retry logic).
- Developing and maintaining from scratch for every service is expensive.
- Frameworks enable code reuse: customize a specific building block
- Example: Specify which information from the incoming request credentials is important for authorization.



Benefits of Using Frameworks....2

- Frameworks leads to increased productivity for all developers in an organization.
- Workload distribution for more efficiency :
 - Design and develop the framework building blocks by a team of domain experts.
 - Implement security and reliability features itself by each individual team.
- Example: If the security team handles cryptography, all other teams benefit from their knowledge.
- Frameworks increase productivity by providing tools that are easy to integrate with.
 - Example: provide tools that automatically export basic operational metrics
- Using frameworks makes reasoning about the code easy by clearly separating business logic from common functions.
 - Enables developers to make assertions about the security or reliability of a service with more confidence.

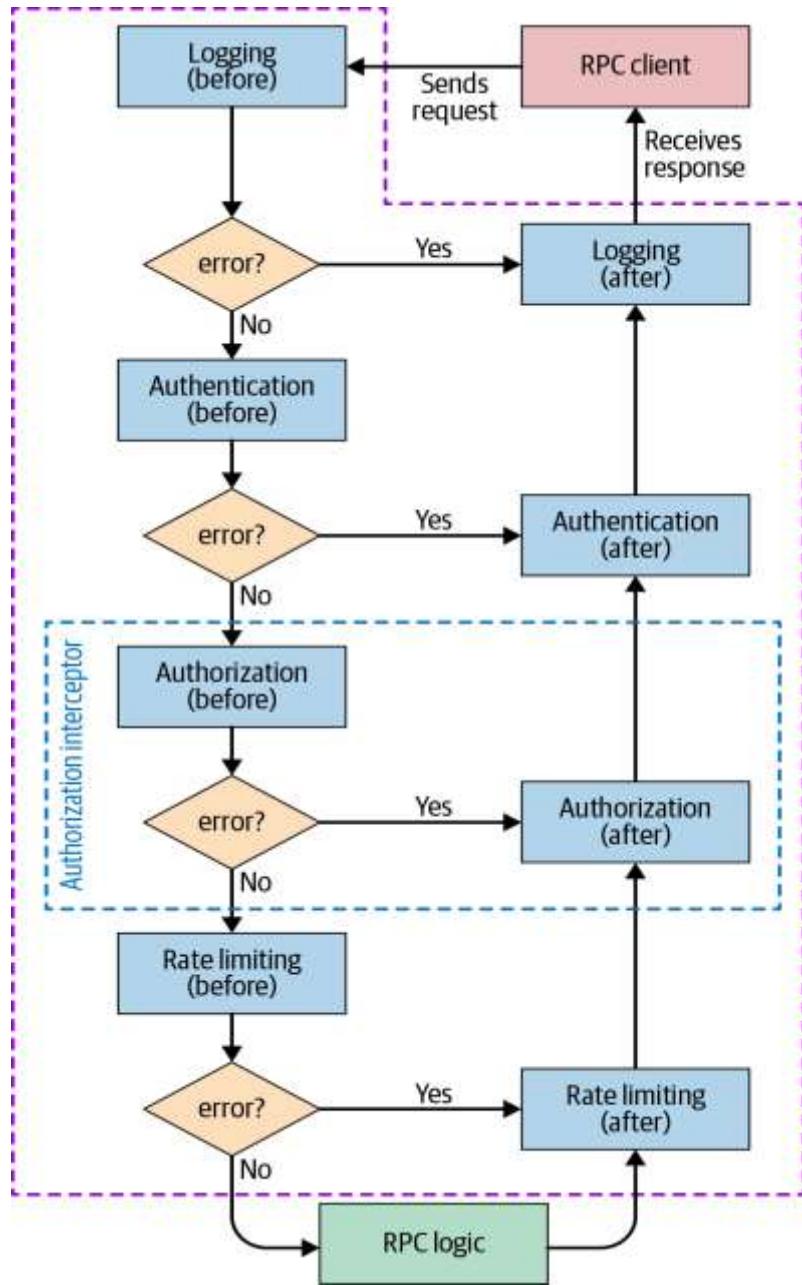


Example: Framework for RPC Backends....1

- Most RPC backends follow a similar structure.
- They handle request-specific logic and typically also perform the following:
 - Logging
 - Authentication
 - Authorization
 - Throttling (rate limiting)
- No need to reimplementing specific functionality for every single RPC backend.
 - Use a framework that can hide the implementation details of their building blocks
 - Developers just need to customize each step to accommodate their service's needs.



Example: Framework for RPC Backends....2



- Framework architecture based on predefined interceptors that are responsible for each of the previously mentioned steps.
- Each interceptor defines an action to be performed before and after the actual RPC logic executes.
- Each stage can report an error condition, which prevents further interceptors from executing.
- The before stage of the logging interceptor could log the call, and the after stage could log the status of the operation.



Example: Framework for RPC Backends....3

```
type Request struct {
    Payload proto.Message
}

type Response struct {
    Err error
    Payload proto.Message
}

type Interceptor interface {
    Before(context.Context, *Request) (context.Context, error)
    After(context.Context, *Response) error
}

type CallInfo struct {
    User string
    Host string
    ...
}
```

“Initial type definitions (the before stage of an interceptor can modify the context; for example, the authentication interceptor can add verified information about the caller)”



Example: Framework for RPC Backends...4

```
type authzInterceptor struct {
    allowedRoles map[string]bool
}

func (ai *authzInterceptor) Before(ctx context.Context, req *Request) (context.Context, error) {
    // callInfo was populated by the framework.
    callInfo, err := FromContext(ctx)
    if err != nil { return ctx, err }

    if ai.allowedRoles[callInfo.User] { return ctx, nil }
    return ctx, fmt.Errorf("Unauthorized request from %q", callInfo.User)
}

func (*authzInterceptor) After(ctx context.Context, resp *Response) error {
    return nil // Nothing left to do here after the RPC is handled.
}
```

“Example authorization interceptor that allows only requests from whitelisted users”



Example: Framework for RPC Backends...4

```
type logInterceptor struct {
    logger *LoggingBackendStub
}

func (*logInterceptor) Before(ctx context.Context,
                             req *Request) (context.Context, error) {
    // callInfo was populated by the framework.
    callInfo, err := FromContext(ctx)
    if err != nil { return ctx, err }
    logReq := &pb.LogRequest{
        timestamp: time.Now().Unix(),
        user: callInfo.User,
        request: req.Payload,
    }
    resp, err := logger.Log(ctx, logReq, WithAttemptCount(3))
    return ctx, err
}

func (*logInterceptor) After(ctx context.Context, resp *Response) error {
    if resp.Err == nil { return nil }

    logErrorReq := &pb.LogErrorRequest{
        timestamp: time.Now().Unix(),
        error: resp.Err.Error(),
    }
    resp, err := logger.LogError(ctx, logErrorReq, WithAttemptCount(3))
    return err
}
```

“Example logging interceptor that logs every incoming request (before stage) and then logs all the failed requests with their status (after stage);”



Common Security Vulnerabilities



SQL Injection Vulnerabilities: TrustedSqlString....1

- SQL injection is a common class of security vulnerability.
- If untrustworthy string fragments are inserted into a SQL query: Possibility to inject database commands.
- The following is a simple password reset web form:
 - db.query("UPDATE users SET pw_hash = " + request["pw_hash"] + " WHERE reset_token = " + request.params["reset_token"] + "")
- User's request is directed to a backend with an unguessable reset_token specific to their account.
- With a string concatenation a malicious user could craft a custom reset_token with extra SQL commands (such as ' or username='admin) and inject this token into the backend.
 - Reset the password hash of a different user—

SQL Injection Vulnerabilities: TrustedSqlString....2

- SQL injection vulnerabilities can be harder to spot in more complicated codebases.
- The database engine can help to prevent SQL injection vulnerabilities by providing bound parameters and prepared statements:
 - Query q = db.createQuery()
 - "UPDATE users SET pw_hash = @hash WHERE token = @token";
 - q.setParameter("hash", request.params["hash"]);
 - q.setParameter("token", request.params["token"]);
 - db.query(q);



SQL Injection Vulnerabilities: TrustedSqlString....2

- SQL injection vulnerabilities can be harder to spot in more complicated codebases.
- The database engine can help to prevent SQL injection vulnerabilities by providing bound parameters and prepared statements:
 - Query q = db.createQuery()
 - "UPDATE users SET pw_hash = @hash WHERE token = @token";
 - q.setParameter("hash", request.params["hash"]);
 - q.setParameter("token", request.params["token"]);
 - db.query(q);
- Create a separate type called TrustedSqlString and enforce by construction that all SQL query strings are created from developer-controlled input.



Preventing XSS: SafeHtml

- XSS vulnerabilities occur when a web application renders untrustworthy input without appropriate sanitization.
- Example: An attacker-controlled \$address value into an HTML snippet such as <div> \$address</div>, which is shown to another user.
 - Set \$address to <script>exfiltrate_user_data();</script> and execute arbitrary code in the context of another user's page.
- HTML does not have the equivalent of binding query parameters.
- Different HTML attributes and elements have different semantics
 - Application developers have to treat values differently depending on the context in which they appear.



Lessons for Evaluating and Building Frameworks



Simple, Safe, Reliable Libraries for Common Tasks....1

- Building a safe library:
 - Covers all possible use cases
 - Handles each reliably can be very challenging.
- Example: Application developer working on an HTML template:
 - Show profile">
- Three different layers of context to be XSS-proof:
 - a single-quoted string,
 - inside JavaScript,
 - inside an attribute in an HTML element.



Simple, Safe, Reliable Libraries for Common Tasks....2

- Create a template system that can handle all possible combinations of corner cases is complicated.
 - Business needs might dictate complex rules.
 - Difficult to meet all developer needs for general-purpose programming language.
- Simple libraries are easier to explain, document, and use.
 - Reduce developer friction
 - Adopt the secure-by-design library.
- Offer different libraries optimized for different use cases.
- Example: Both HTML templating systems for complicated pages and building libraries for short snippets.



Rollout Strategy

- Using types for security properties is very useful for new code.
- Applications in Google-internal web framework developed with safe types for HTML have few reported XSS vulnerabilities.
- It is more challenging to adapt existing code to use safe types.
- Need a strategy for migrating legacy code:
 - Discover new classes of security and reliability issues.
 - Be able to access and modify the entire source code of the application.
- Most of Google's source code is stored in a single repository with centralized processes for making, building, testing, and submitting changes.



Simplicity Leads to Secure and Reliable
Code



Avoid Multilevel Nesting....1

- Multilevel nesting is a common anti-pattern that can lead to simple mistakes.
- If the error is in the most common code path, it will likely be captured by the unit tests.
- Unit tests don't always check error handling paths in multilevel nested code.
- The error might result in :
 - Decreased reliability - If the service crashes when it mishandles an error)
 - Security vulnerability - like a mishandled authorization check error



Avoid Multilevel Nesting....1

- Multilevel nesting is a common anti-pattern that can lead to simple mistakes.
- If the error is in the most common code path, it will likely be captured by the unit tests.
- Unit tests don't always check error handling paths in multilevel nested code.
- The error might result in :
 - Decreased reliability - If the service crashes when it mishandles an error)
 - Security vulnerability - like a mishandled authorization check error



Avoid Multilevel Nesting....2

Two equivalent codes

```
response = stub.Call(rpc, request)

if rpc.status.ok():
    if response.GetAuthorizedUser():
        if response.GetEnc() == 'utf-8':
            if response.GetRows():
                vals = [ParseRow(r) for r in
                        response.GetRows()]
                avg = sum(vals) / len(vals)
                return avg, vals
            else:
                raise ValueError('no rows')
        else:
            raise AuthError('unauthorized')
    else:
        raise ValueError('wrong encoding')
else:
    raise RpcError(rpc.ErrorText())
```

```
response = stub.Call(request, rpc)

if !rpc.status.ok():
    raise RpcError(rpc.ErrorText())

if not response.GetAuthorizedUser():
    raise ValueError('wrong encoding')

if response.GetEnc() != 'utf-8':
    raise AuthError('unauthorized')

if not response.GetRows():
    raise ValueError('no rows')

vals = [ParseRow(r) for r in
        response.GetRows()]
avg = sum(vals) / len(vals)
return avg, vals
```

- Errors are often harder to spot in code with multiple levels of nesting



Eliminate YAGNI Smells....1

- overengineer solutions by adding functionality that may be useful in the future, “just in case.”
- YAGNI (You Aren’t Gonna Need It) :Implement only the needed code.
- YAGNI code needs to be documented, tested, and maintained (adds unnecessary complexity).
- Example:

```
class Mammal { ...
virtual Status Sleep(bool hibernate) = 0;};
class Human : public Mammal { ...
virtual Status Sleep(bool hibernate) {
age += hibernate ? kSevenMonths : kSevenHours;
return OK;}
};
```



Eliminate YAGNI Smells....1

- overengineer solutions by adding functionality that may be useful in the future, “just in case.”
- YAGNI (You Aren’t Gonna Need It) :Implement only the needed code.
- YAGNI code needs to be documented, tested, and maintained (adds unnecessary complexity).
- Example:

```
class Mammal { ...
virtual Status Sleep(bool hibernate) = 0;};
class Human : public Mammal { ...
virtual Status Sleep(bool hibernate) {
age += hibernate ? kSevenMonths : kSevenHours;
return OK;}
};
```



Eliminate YAGNI Smells....2

- The Human::Sleep code must handle the case when hibernate is true, even though all callers should always pass false.
- **Callers must handle the returned status, even though that status should always be OK.**
- The code can be simplified to the following:

```
class Human { ...
void Sleep() { age += kSevenHours; }
};
```
- **It will be easier to create a Mammal interface with a better common API generalized based on several existing classes.**



Repay Technical Debt

- **Common practice for developers to mark places that require further attention with TODO or FIXME annotations.**
- Advantage :
 - this habit can accelerate the delivery velocity for the most critical functionality.
- Disadvantage:
 - Can incur technical debt.
- Technical debt can be resolved in many ways:
 - Keeping dashboards with code health metrics.
 - Creating notifications when code health metrics drop below predefined thresholds or when the number of automatically detected issues is too high.



Refactoring

- Refactoring is the most effective way to keep a codebase clean and simple.
- Change the backend even a healthy codebase occasionally needs to be refactored when you extend the existing feature set.
- Useful when working with old, inherited codebases.
- The first step of refactoring is measuring code coverage and increasing that coverage to a sufficient level.
 - Higher the coverage, higher the confidence in the safety of refactoring.
- 100% test coverage can't guarantee success because the tests may not be meaningful.



Security and Reliability by Default



Choose the Right Tools....1

- **Complex task to select a language, framework, and libraries:**
 - **Integration with the existing codebase**
 - **Availability of libraries**
 - **Skills or preferences of the developer team**
- The language choice can have enormous impact on the security and reliability of the project:
 - Use memory-safe languages
 - Microsoft's Matt Miller claimed that around 70% of all security vulnerabilities are due to memory safety issues.
 - Nick Kralevich from Google reported that 85% of all bugs in Android were caused by memory management errors.
 - Use strong typing and static type checking:
 - Enforce type checking either during compilation (static type checking) or at runtime (dynamic type checking).



Choose the Right Tools....2

- **The benefits of strong typing and static type checking are especially noticeable when working on large codebases with multiple developers.**
- Deduce about the code only in case of 100% test coverage with dynamic type checking (for example, in Python):
 - Great in principle, but rarely observed in practice.
- Reasoning about the code becomes even harder in weakly typed languages, often leading to surprising behavior.
- Example: In JavaScript, every literal is by default treated as a string:
 - [9, 8, 10].sort() ->[10, 8, 9]
- Use extensions like the following to improve the reliability of the code (incrementally add them to existing codebases):
 - Pytype for Python
 - TypeScript for JavaScript



Use Strong Types....1

- Using untyped primitives (such as strings or integers) can lead to the following issues:
 - Passing conceptually invalid parameters to a function
 - Unwanted implicit type conversions
 - Difficult-to-understand type hierarchy
 - Confusing measurement units
- Passing conceptually invalid parameters to a function occurs if the primitive type of a function parameter does not have enough context.
- Examples:
 - AddUserToGroup(string, string), it's unclear whether the group name is provided as the first or the second argument.
 - What is the order of height and width in the Rectangle (3.14, 5.67) constructor call?
 - Does Circle(double) expect a radius or diameter?



Use Strong Types....2

- Documentation can correct for ambiguity, but developers are still bound to make mistakes.
- Use strong types can catch mistakes at compilation time.
- The required calls would look like the following:
 - Add(User("alice"), Group("root-users"))
 - Rectangle(Width(3.14), Height(5.67))
 - Circle(Radius(1.23))
- Implicit type conversions may lead to the following:
 - Truncation when converting from larger to smaller integer types
 - Precision loss when converting from larger to smaller floating-point types
 - Unexpected object creation



Use Strong Types....3

- Using strong types protects the code from errors that a compiler doesn't capture.
- Example - difficult-to-understand type hierarchy:

```
class Bar {  
public:  
    Bar(bool is_safe) {...}  
};  
• void Foo(const Bar& bar) {...}  
• Foo(false); // Likely OK, but is the developer aware a Bar object was created?  
• Foo(5); // Will create Bar(is_safe := true), but likely by accident.  
• Foo(NULL); // Will create Bar(is_safe := false), again likely by accident.
```

- The three calls will compile and execute, but the outcome of the operation doesn't match developers' expectations.



Sanitize Your Code....1

- It's very useful to automatically validate that the code is not experiencing any typical memory management or concurrency pitfalls.
 - Run these checks as a pre-submit action for each change list or as part of a continuous build and test automation harness.
- Memory management errors are a leading cause of security issues, and can result in the following failure scenarios:
 - Reading unallocated memory (before new or after delete)
 - Reading outside of the allocated memory (buffer overflow attack scenario)
 - Reading uninitialized memory
 - Memory leaks when a system loses the address of allocated memory or doesn't deallocate unused memory



Sanitize Your Code....2

- Valgrind is a popular framework that allows developers to catch those sorts of errors.
- Valgrind has the benefit of providing a virtual machine that interprets a user's binary.
- The Valgrind tool Helgrind can additionally detect common synchronization errors such as these:
 - Misuses of the POSIX pthreads API (e.g., unlocking a not-locked mutex, or a mutex held by another thread)
 - Potential deadlocks arising from lock ordering problems
 - Data races caused by accessing memory without adequate locking or synchronization



Sanitize Your Code....3

- Google Sanitizers suite offers various components that can detect all the same issues that Valgrind's Callgrind can detect:
 - AddressSanitizer (ASan) detects memory errors (buffer overflows, use after free, incorrect initialization order).
 - LeakSanitizer (LSan) detects memory leaks.
 - MemorySanitizer (MSan) detects when a system is reading uninitialized memory.
 - ThreadSanitizer (TSan) detects data races and deadlocks.
 - UndefinedBehaviorSanitizer (UBSan) detects situations that have undefined behavior.
- The main advantage of the Google Sanitizers suite is speed: it's up to 10 times faster than Valgrind.



Summary

- Several principles that guide developers toward designing and implementing more secure and reliable code.
- It will be recommended to use frameworks as a powerful strategy, as they reuse proven building blocks for sensitive areas of code prone to reliability and security issues: authentication, authorization, logging, etc.
- Frameworks also tend to improve developer productivity.
- Additional strategies for writing secure and reliable code include aiming for simplicity, choosing the right tools, using strong rather than primitive types, and continuously sanitizing the codebase.



Thank You



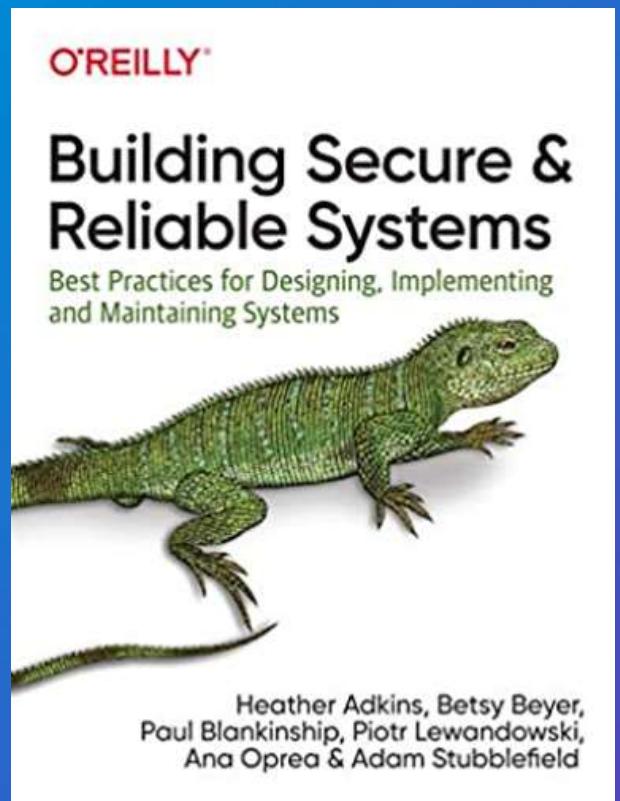


الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432

Introduction to Cyber Security and Digital Crime

Building Secure & Reliable Systems Best Practices for Designing, Implementing and Maintaining Systems

by Heather Adkins, Betsy Beyer, Paul
Blankinship, Piotr Lewandowski, Ana
Oprea & Adam Stubblefi



Week 12

Chapter 13: Testing Code



Contents

1. Unit Testing
2. Integration Testing
3. Writing Effective Integration Tests
4. Fuzz Testing
5. Static Program Analysis



Objectives

- Apply different testing techniques to mitigate failure risks
- Apply and strengthen software's resilience by applying static and dynamic program analysis
- Validate users input



Required Reading

1. Chapter 13 from Building Secure & Reliable Systems Best Practices for Designing, Implementing and Maintaining Systems.



Unit Testing



Writing Effective Unit Tests....1

- *Increase system security and reliability by pinpointing a wide range of bugs in individual software components before a release.*
 - Involves breaking software components into smaller, self-contained “units” that have no external dependencies, and then testing each unit.
- **Consist of code that exercises a given unit with different inputs selected by the engineer writing the test.**
- Popular unit test frameworks are available for many languages.
 - Systems based on the xUnit architecture are very common.



Writing Effective Unit Tests....2

- Frameworks following the xUnit paradigm allow common setup and teardown code to execute with each individual **test method**.
- Popular examples include **JUnit for Java**, **GoogleTest for C++**, **go2xunit for Golang**, and the built-in **unittest module in Python**.
- Example : **Unit test for a function that checks whether the provided argument is a prime number**

```
TEST(IsPrimeTest, Trivial) {  
    EXPECT_FALSE(IsPrime(0));  
    EXPECT_FALSE(IsPrime(1));  
    EXPECT_TRUE(IsPrime(2));  
    EXPECT_TRUE(IsPrime(3));  
}
```



When to Write Unit Tests....1

- **A common strategy is to write tests shortly after writing the code.**
- **These tests typically accompany the new code in the same commit** (the cases when the code is checked manually).
- Example: Only billing administrators for the group that owns the service can request more quota” in case of storage management application.
- **A peer reviewer can double-check the tests to ensure they’re sufficiently robust to maintain the quality of the codebase.**



When to Write Unit Tests....2

- **Test-Driven Development (TDD) methodologies encourage engineers to write unit tests (not necessary after writing code).**
- **When testing new features or bug fixes, the tests will fail until the behavior is completely implemented.**
- **Once a feature is implemented and the tests pass, engineers progress to the next feature.**
- **For existing projects (not in TDD model): slowly integrate and improve test coverage in response to bug reports.**
- Achieving full coverage, doesn't mean the project is bug-free
- Write unit tests in response to internal manual testing or code review efforts.



How Unit Testing Affects Code....1

- To improve the comprehensiveness of the tests:
 - **Need to design new code to include testing provisions.**
 - **Refactor older code to make it more testable.**
- refactoring involves providing a way to intercept calls to external systems.
- The code can be tested in a variety of ways: invokes the interceptor the correct number of times , or with the correct arguments.



How Unit Testing Affects Code....2

- Test a piece of code that opens tickets in a remote issue tracker when certain conditions are met:
 - **Creating a real ticket every time the unit test runs would generate unnecessary noise.**
- Even worse, this testing strategy may fail randomly if the issue tracker system is :
 - Unavailable
 - Violating the goal of quick
 - Reliable test results.
- Refactor this code:
 - Remove direct calls to the issue tracker service
 - Replace those calls with an abstraction (Ex. an interface for an IssueTrackerService object.)



How Unit Testing Affects Code....3

- **The implementation for testing could record data when it receives calls such as “Create an issue”.**
- **This refactor dramatically reduces the “flakiness” of a test that depends on real-world systems.**
- It’s easy to fall into the trap of over-abstraction, where tests assert mechanical facts about the order of function calls or their arguments.
- To help avoid constant test rewrites:
 - Consider asking engineers familiar with the service to provide suitable fake implementations for any nontrivial testing needs.
- Advantage: the team that owns the abstraction can ensure it tracks the feature set of the service as it evolves.



Integration Testing



Writing Effective Integration Tests....1

- **Integration tests may be influenced by design choices in the code.**
- Example: a unit test mock may simply assert that the method was invoked to file a ticket with the remote service.
- **An integration test would more likely use a real client library.**
- Rather than creating spurious bugs in production, the integration test would communicate with a QA endpoint.
- **Test cases would exercise the application logic with inputs that trigger calls to the QA instance.**
- Supervising logic could then query the QA instance to verify that externally visible actions took place successfully from an end-to-end perspective.



Writing Effective Integration Tests....2

- Understanding why integration tests fail when all unit tests pass can require a lot of time and energy.
- **Good logging at key logical junctures of the integration tests can help to debug and understand where breakdowns occur.**
- For integration tests:
 - Go beyond individual units by examining interactions between components,
 - Only a limited amount about how well those units will conform to the expectations in other scenarios.
- One of the many reasons using each type of testing in the development lifecycle adds value:
 - One form of testing is often not a substitute for another.



Dynamic Program Analysis



Dynamic Program Analysis....1

- Program analysis allows users to carry out a number of useful actions.
- Example:
 - Performance profiling.
 - Checking for security-related correctness.
 - Code coverage reporting.
 - Dead code elimination.
- Perform program analysis statically to investigate software without executing it.
- Dynamic program analysis analyzes software by running programs (in virtualized or emulated environments).
- Best-known types of dynamic analysis:
 - Performance profilers
 - Code coverage report generators



Dynamic Program Analysis....2

- Role of compilers and dynamic program analysis tools :
 - Configure instrumentation to collect runtime statistics like:
 - Performance
 - Profiling information
 - Code coverage information,
 - Profile-based optimizations.
- The compiler inserts additional instructions and callbacks to a backend runtime library that surfaces and collects the relevant information when the binary is executed.
- The Google Sanitizers suite provides compilation-based dynamic analysis tools.
- Example: AddressSanitizer (ASan) finds a number of common memory related bugs (out-of-bounds memory accesses, in C/C++ programs).



Fuzz Testing



How Fuzz Engines Work....1

- Fuzz engines can vary in complexity and sophistication.
- At the low end of the spectrum:
 - Dumb fuzzing simply reads bytes from a random number generator and passes them to the fuzz target in an attempt to find bugs.
- Fuzz engines have grown increasingly smart through integration with compiler toolchains.
 - Generate more interesting and meaningful samples by taking advantage of the compiler instrumentation features.



How Fuzz Engines Work....2

- A good industry practice: Use the maximum of fuzz engines into the build toolchain, and to monitor metrics like the percentage of code covered.
- Some fuzz engines accept dictionaries of interesting keywords from the specifications or grammars:
 - Well-specified protocols
 - languages
 - formats (like HTTP, SQL, and JSON).
- The fuzz engine can generate input that's likely to be accepted by the program under test.



How Fuzz Engines Work....3

- Peach Fuzzer allows a fuzz driver author to programmatically define the format of the input and the expected relationships between fields.
 - Generate test cases that violate those relationships.
 - Accept a set of sample input files, referred to as a seed corpus, that are representative of what the code being fuzzed expects.
 - Mutates these seed inputs, in addition to carrying out any other supported input generation strategies.
- Some software packages come with sample files as part of their existing test suites (ex. MP3s or JPEGs)
- Security researchers publish seed corpora for popular file formats:
 - OSS-Fuzz
 - The Fuzzing Project
 - American Fuzzy Lop (AFL)



How Fuzz Engines Work....4

- In recent years, improvements to compiler toolchains have resulted in significant advancements toward making smarter fuzz engines.
- C/C++ - LLVM Clang can instrument the code to allow the fuzz engine to observe what code is executed while processing a specific sample input.
- Other languages - fuzz engines may require a specific compiler to properly trace the execution paths to increase code coverage:
 - afl-gcc for AFL
 - go-fuzz-build for the go-fuzz engine
- A fuzz engine is most effective at detecting bugs if encountering them triggers consistent and well-defined events.



Writing Effective Fuzz Drivers....1

- To make these fuzzing concepts more concrete:
 - Use the framework provided by LLVM's libFuzzer engine, which is included with the Clang compiler.
- As a fuzzer author: Write a single driver that implements the function prototype:

```
int LLVMFuzzerTestOneInput(const uint8_t *data, size_t size);
```
- The respective fuzz engines will then generate byte sequences and invoke your driver, which can pass the input to the code you want to test.
- The goal of the fuzz engines is to execute the fuzz target via the driver as quickly as possible with as many unique and interesting inputs as they can generate.



Writing Effective Fuzz Drivers....2

- To enable reproducible crashes and quick fuzzing, try to avoid the following in the fuzz drivers:
 - Nondeterministic behavior, such as relying on random number generators or specific multithreading behavior.
 - Slow operations, like console logging or disk I/O.
 - Crashing intentionally.
- The fuzz engine can't disambiguate intentional crashes.
- The fuzz engine is unlikely to ever produce a valid checksum and pass the integrity check without specialized logic.
- A common convention is to use compiler preprocessor flags like `DFUZZING_BUILD_MODE_UNSAFE_FOR_PRODUCTION` to enable this fuzzer-friendly behavior and to help reproduce crashes identified through fuzzing.



An Example Fuzzer....1

- Knusperli is a JPEG decoder that might see a wide range of input if it's encoding user uploads or processing images from the web.
- It provides a function that accepts
 - A sequence of bytes (the JPEG)
 - Size parameter
 - A parameter that controls which sections of the image to parse.
- Example fuzz driver targets this function:
 - `bool ReadJpeg(const uint8_t* data, const size_t len,
JpegReadMode mode, JPEGData* jpg);`



An Example Fuzzer....2

- Example “jpeg_decoder_fuzzer.cc”

```
1 #include <cstddef>
2 #include <cstdint>
3 #include "jpeg_data_decoder.h"
4 #include "jpeg_data_reader.h"
5
6 extern "C" int LLVMFuzzerTestOneInput(const uint8_t *data, size_t sz) {
7 knusperli::JPEGData jpg;
8 knusperli::ReadJpeg(data, sz, knusperli::JPEG_READ_HEADER, &jpg);
9 return 0;
10 }
```

Build and run the fuzz driver with these commands:

```
$ CC=clang-6.0 CXX=clang++-6.0 bazel build --config=asan :fuzzer
$ mkdir synthetic_corpus
$ ASAN_SYMBOLIZER_PATH=/usr/lib/llvm-6.0/bin/llvm-symbolizer bazel-bin/fuzzer \
-max_total_time 300 -print_final_stats synthetic_corpus/
```



Continuous Fuzzing

- Running fuzzers regularly over a codebase as it's developed can provide a valuable feedback loop to engineers.
- A continuous build pipeline can generate daily builds of fuzzers in the codebase to be consumed by a system that runs the fuzzers, collects crash information, and files bugs in an issue tracker.
- Engineering teams can use the results to focus on identifying vulnerabilities or eliminating root causes that make the service miss its SLO.
- Example: ClusterFuzz and OSSFuzz



Continuous Fuzzing - ClusterFuzz

- ClusterFuzz is an open source implementation of a scalable fuzzing infrastructure released by Google.
- It manages pools of virtual machines that run fuzzing tasks and provides a web interface to view information about the fuzzers.
- ClusterFuzz does not build fuzzers, but instead expects a continuous build/integration pipeline to push fuzzers to a Google Cloud Storage bucket.
- It provides services like :
 - corpus management,
 - crash deduplication
 - lifecycle management for the crashes that it identifies.



Continuous Fuzzing - OSS-Fuzz

- OSS-Fuzz combines modern fuzzing techniques with a scalable distributed execution of ClusterFuzz that's hosted on the Google Cloud Platform.
- It uncovers security vulnerabilities and stability issues, and reports them directly to developers.
- within five months of its launch in December 2016, OSS-Fuzz had discovered over a thousand bugs, and since then it has found tens of thousands more.
- Once a project is integrated with OSS-Fuzz, the tool uses continuous and automated testing to find issues only hours after modified code is introduced into the upstream repository.



Static Program Analysis



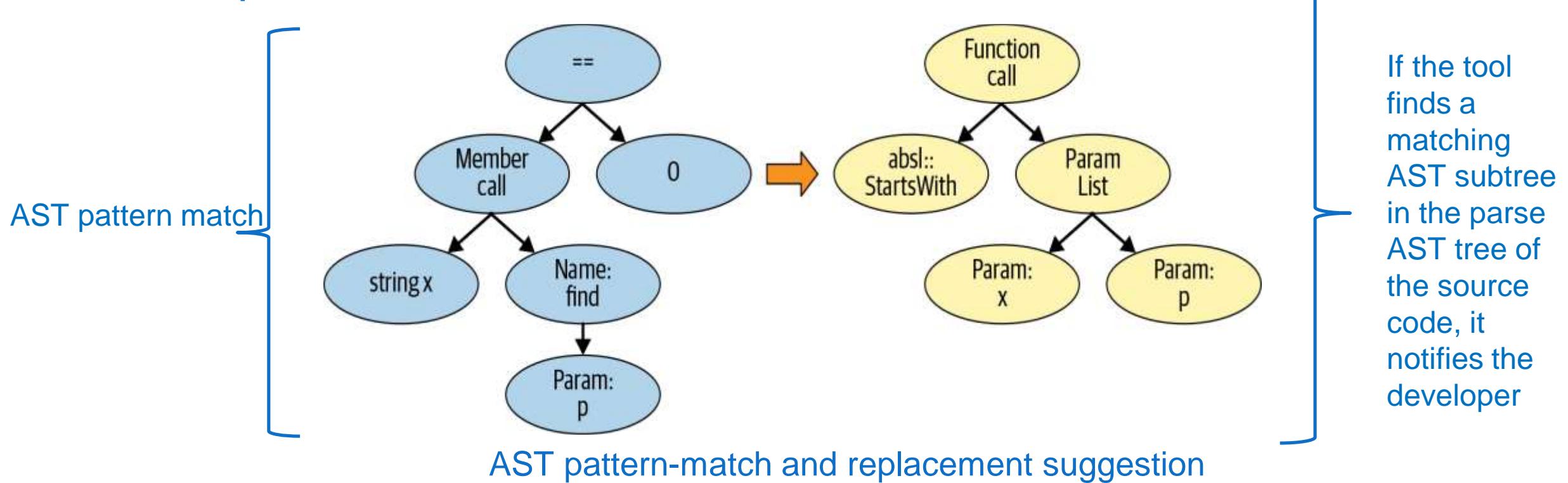
Automated Code Inspection Tools....1

- Automated code inspection tools perform a syntactic analysis of source code with respect to language features and usage rules.
- Code inspection tools are also easily extensible:
 - Simply add new rules that cover many types of bugs, especially bugs related to language features.
- Many organizations enforce style and format checks by default in order to maintain a cohesive codebase that's easier to manage across large developer teams.
- An AST is a tree representation of a program's source code based on the syntactic structure of the programming language.
- Example: AST may contain a node representing an if-then-else construct, which has three child nodes:
 - One node for the condition of the if statement,
 - One node representing the subtree for the then branch,
 - One node representing the subtree of the else branch.



Automated Code Inspection Tools....2

- The Clang-Tidy infrastructure provides the tooling to find AST subtree patterns.



Integration of Static Analysis in the Developer Workflow....1

- Finding bugs early is important because the cost of fixing them increases substantially if they're pushed into the source code repository or deployed to users.
- Google developed the Tricorder program analysis platform.
- Shipshape is an open-source version of Tricorder.
- Tricorder performs static analysis of approximately 50,000 code review changes per day.
- The platform runs many types of program analysis tools and surfaces warnings to developers during code review.



Integration of Static Analysis in the Developer Workflow....2

```
package com.google.devtools.staticanalysis;

public class Test {
    ▾ Lint      Missing a Javadoc comment.
    Java
    1:02 AM, Aug 21
    Please fix
    Not useful

    public boolean foo() {
        return getString() == "foo".toString();
    }

    ▾ ErrorProne  String comparison using reference equality instead of value equality
    StringEquality
    1:03 AM, Aug 21
    Please fix
    Suggested fix attached: show
    Not useful

    }

    public String getString() {
        return new String("foo");
    }
}
```

Screenshot of static analysis results during code review provided via Tricorder

```
//depot/google3/java/com/google/devtools/staticanalysis/Test.java
package com.google.devtools.staticanalysis;
public class Test {
    public boolean foo() {
        return getString() == "foo".toString();
    }

    public String getString() {
        return new String("foo");
    }
}

package com.google.devtools.staticanalysis;
import java.util.Objects;
public class Test {
    public boolean foo() {
        return Objects.equals(getString(), "foo".toString());
    }

    public String getString() {
        return new String("foo");
    }
}
```

Apply Cancel

Screenshot of the preview fix view for the Error Prone warning from



Abstract Interpretation....1

- Abstract interpretation-based tools statically perform a semantic analysis of program behaviors.
- This technique has been used successfully to verify safety-critical software, such as flight control software.
- Example: program that generates the 10 smallest positive even integers.
- During its regular execution: generates the integer values 2, 4, 6, 8, 10, 12, 14, 16, 18, and 20.
- The objective of the abstract interpretation is to verify that the program covers all observed values with the abstract representation.



Abstract Interpretation....2

- A number of tools rely on abstract interpretation for a variety of languages and properties.
- Example: Frama-C tool allows to :
 - Find common runtime errors
 - assertion violations (buffer overflows, segmentation faults, division by zero).
- The types of bugs related memory can have security implications.
- The AbsInt tool can perform worst-case execution time analysis of tasks in real-time systems.
- The App Security Improvement (ASI) program performs a sophisticated interprocedural analysis on every Android app for safety and security.



Formal Methods

- Formal methods allow users to specify properties of interest for software or hardware systems.
- Most of these are so-called safety properties that specify that a certain bad behavior should never be observable.
- Example:
 - “bad behavior” can include assertions in programs.
 - Others include liveness properties, which allow users to specify a desired outcome
- Users of formal methods can verify these properties for particular systems or models, and even develop such systems using correct-by-construction–based approaches.



Summary

- Scratch the surface of testing software for security and reliability.
- The testing strategies combined with practices around writing secure code to eliminate entire bug classes have been key in minimizing outages and security problems.
- It's important to build software with testability in mind from the earliest stages of development, and to engage in comprehensive testing throughout the development lifecycle.
- Emphasize the value of fully integrating all of these testing and analysis methods into the engineering workflows and CI/CD pipelines.



Thank You

