Malla Reddy College of Engineering & Technology

(UGC Autonomous Institution)

SECURITY ASSESSMENT & RISK ANALYSIS

Prepared by

Dr M V Kamal

Professor & HoD

Department of Emerging Technologies | MRCET

IV Year B.Tech. CSE (CYS)- I Sem L/T/P/C 3/-/-/3

(R20A6208) SECURITY ASSESSMENT and RISK ANALYSIS

SYLLABUS

COURSE OBJECTIVES

1. The course takes a software development perspective to the challenges of engineering software systems that are secure.
2. This course addresses design and implementation issues critical to producing secure software systems.
3. The course deals with the question of how to make the requirements for confidentiality, integrity, and availability integral to the software development process.
4. Secure software requirements gathering to design, development, configuration, deployment, and ongoing maintenance
5. Security of enterprise information systems.

UNIT-1: Defining computer security, the principles of secure software, trusted computing base, etc, threat modeling, advanced techniques for mapping security requirements into design specifications. Secure software implementation, deployment and ongoing management.

UNIT-2: Software design and an introduction to hierarchical design representations. Difference between high-level and detailed design. Handling security with high-level design. General Design Notions. Security concerns designs at multiple levels of abstraction, Design patterns, quality assurance activities and strategies that support early vulnerability detection, Trust models, security Architecture & design reviews .

UNIT-3 Software Assurance Model: Identify project security risks & selecting risk management strategies, Risk Management Framework, Security Best practices/ Known Security Flaws, Architectural risk analysis, Security Testing & Reliability (Penn testing, Risk-Based Security Testing

UNIT-4: Software Security in Enterprise Business: Identification and authentication, Enterprise Information Security, Symmetric and asymmetric cryptography, including public key cryptography, data encryption standard (DES), advanced encryption standard (AES), algorithms for hashes and message digests. Authentication, authentication schemes , access control models, Kerberos protocol, public key infrastructure (PKI), protocols specially designed for e-commerce and web applications, firewalls and VPNs.

UNIT-5: Security development frameworks. Security issues associated with the development and deployment of information systems, including Internet-based e-commerce, e-business, and e-service systems.

TEXT BOOKS:

1. W. Stallings, Cryptography and network security: Principles and practice, 5 th Edition, Upper Saddle River, NJ: Prentice Hall., 2011
2. C. Kaufman, r. Perlman, & M. Speciner, Network security: Private communication in a public world, 2 nd Edition, Upper Saddle River, NJ:Prentice HalL, 2002
3. C. P. Pfleeger, S. L. Pfleeger, Security in Computing, 4 th Edition, Upper Saddle River, NJ:Prentice Hall, 2007
4. T4. M. Merkow, & J. Breithaupt, Information security: Principles and practices. Upper Saddle River, NJ:Prentice Hall, 2005

REFERENCE BOOKS:

1. Gary McGraw, Software Security: Building Security In, Addison-Wesley, 2006

COURSE OUTCOMES:

1. Understand various aspects and principles of software security.
2. Devise security models for implementing at the design level.
3. Identify and analyze the risks associated with s/w engineering and use relevant models to mitigate the risks.
4. Understand the various security algorithms to implement for secured computing and computer networks
5. Explain different security frameworks for different types of systems including electronic systems.

UNIT - I

SECURITY ASSESSMENT & RISK ANALYSIS

Dr M V Kamal

Unit-I Content

Defining computer security, the principles of secure software, trusted computing base, etc, threat modeling, advanced techniques for mapping security requirements into design specifications. Secure software implementation, deployment and ongoing management.

What is Computer Security?

Computer security basically is the protection of computer systems and information from harm, theft, and unauthorized use. It is the process of preventing and detecting unauthorized use of your computer system.

Or

computer security, also called **cybersecurity**, the protection of computer systems and information from harm, theft, and unauthorized use. Computer hardware is typically protected by the same means used to protect other valuable or sensitive equipment—namely, serial numbers, doors and locks, and alarms. The protection of information and system access, on the other hand, is achieved through other tactics, some of them quite complex.

As per the wiki: **Computer security**, **cyber security** , **digital security** or **information technology security** (**IT security**) is the protection of computer systems and networks from attack by malicious actors that may result in unauthorized information disclosure, theft of, or damage to hardware, software, or data, as well as from the disruption or misdirection of the services they provide

There are various types of computer security which is widely used to protect the valuable information of an organization.

What is Computer Security and its types?

One way to ascertain the similarities and differences among Computer Security is by asking what is being secured. For example,

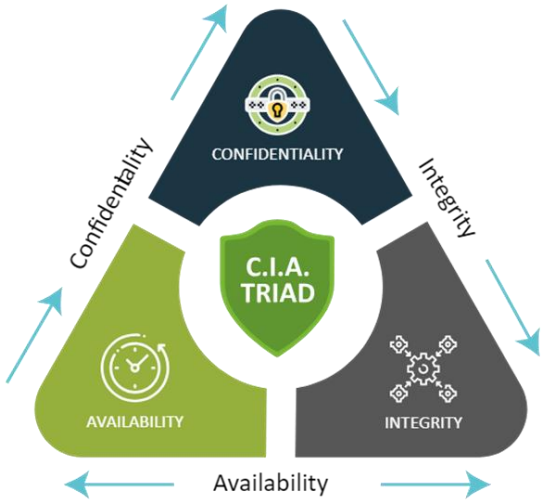
- **Information security** is securing information from unauthorized access, modification & deletion
- **Application Security** is securing an application by building security features to prevent from Cyber Threats such as SQL injection, DoS attacks, data breaches and etc.
- Computer Security means securing a standalone machine by keeping it updated and patched

- **Network Security** is by securing both the software and hardware technologies
- **Cyber security** is defined as protecting computer systems, which communicate over the computer networks

The CIA Triad

Computer security is mainly concerned with three main areas:

- **Confidentiality** is ensuring that information is available only to the intended audience
- **Integrity** It involves making sure your data is trustworthy and free from tampering. The integrity of your data is maintained only if the data is authentic, accurate, and reliable.
- **Availability** is protecting information from being modified by unauthorized parties



Computer security threats

Computer security threats are possible dangers that can possibly hamper the normal functioning of your computer. In the present age, cyber threats are constantly increasing as the world is going digital. The most harmful types of computer security are:

Viruses

A computer virus is a malicious program which is loaded into the user's computer without user's knowledge. It replicates itself and

infects the files and programs on the user's PC. The ultimate goal of a virus is to ensure that the victim's computer will never be able to operate properly or even at all.

Computer Worm

A computer worm is a software program that can copy itself from one computer to another, without human interaction. The potential risk here is that it will use up your computer hard disk space because a worm can replicate in greate volume and with great speed.

Phishing

Disguising as a trustworthy person or business, phishers attempt to steal sensitive financial or personal information through fraudulent email or instant messages. Phishing in unfortunately very easy to execute. You are deluded into thinking it's the legitimate mail and you may enter your personal information.

Botnet

A botnet is a group of computers connected to the internet, that have been compromised by a hacker using a computer virus. An individual computer is called 'zombie computer'. The result of this threat is the victim's computer, which is the bot will be used for malicious activities and for a larger scale attack like DDoS.

Rootkit

A rootkit is a computer program designed to provide continued privileged access to a computer while actively hiding its presence. Once a rootkit has been installed, the controller of the rootkit will be able to remotely execute files and change system configurations on the host machine.

Keylogger

Also known as a keystroke logger, keyloggers can track the real-time activity of a user on his computer. It keeps a record of all the keystrokes made by user keyboard. Keylogger is also a very powerful threat to steal people's login credential such as username and password.

Apart from the above, the following are few more types of cyber attacks....

1. Denial of service attack or DOS: A denial of service attack is a kind of cyber attack in which the attackers disrupt the services of the particular network by sending infinite requests and temporary or permanently making the network or machine resources unavailable to the intended audience.

2. Backdoor: In a backdoor attack, malware, trojan horse or virus gets installed in our system and start affecting it's security along with the main file. Consider an example: suppose you are installing free software from a certain website on the Internet. Now, unknowingly, along with this software, a malicious file also gets installed, and as soon as you execute the installed software that file's malware gets affected and starts affecting your computer security. This is known as a backdoor.

3.Eavesdropping: Eavesdropping refers to secretly listening to someone's talk without their permission or knowledge. Attackers try

to steal, manipulate, modify, hack information or systems by passively listening to network communication, knowing passwords etc. A physical example would be, suppose if you are talking to another person of your organization and if a third person listens to your private talks then he/ she is said to eavesdrop on your conversation. Similarly, your conversation on the internet maybe eavesdropped by attackers listening to your private conversation by connecting to your network if it is insecure.

4. Phishing: Phishing is pronounced as "fishing" and working functioning is also similar. While fishing, we catch fish by luring them with bait. Similarly, in phishing, a user is tricked by the attacker who gains the trust of the user or acts as if he is a genuine person and then steals the information by ditching. Not only attackers but some certain websites that seem to be genuine, but actually they are fraud sites. These sites trick the users and they end up giving their personal information such as login details or bank details or card number etc. Phishing is of many types: Voice phishing, text phishing etc.

5. Spoofing: Spoofing is the act of masquerading as a valid entity through falsification of data(such as an IP address or username), in order to gain access to information or resources that one is otherwise unauthorized to obtain. Spoofing is of several types- email spoofing, IP address spoofing, MAC spoofing , biometric spoofing etc.

6. Malware: Malware is made up of two terms: Malicious + Software = Malware. Malware intrudes into the system and is designed to damage our computers. Different types of malware are adware, spyware, ransomware, Trojan horse, etc.

7. Social engineering: Social engineering attack involves manipulating users psychologically and extracting confidential or sensitive data from them by gaining their trust. The attacker generally exploits the trust of people or users by relying on their cognitive basis.

8. Polymorphic Attacks: Poly means "many" and morph means "form", polymorphic attacks are those in which attacker adopts multiple forms and changes them so that they are not recognized easily. These kinds of attacks are difficult to detect due to their changing forms.

Why is Computer Security Important?

In this digital era, we all want to keep our computers and our personal information secure and hence computer security is important to keep our personal information protected. It is also important to maintain our

computer security and its overall health by preventing viruses and malware which would impact on the system performance.

Computer Security Practices

Computer security threats are becoming relentlessly inventive these days. There is much need for one to arm oneself with information and resources to safeguard against these complex and growing computer security threats and stay safe online. Some preventive steps you can take include:

- Secure your computer physically by:
    - Installing reliable, reputable security and anti-virus software
    - Activating your firewall, because a firewall acts as a security guard between the internet and your local area network
- Stay up-to-date on the latest software and news surrounding your devices and perform software updates as soon as they become available
- Avoid clicking on email attachments unless you know the source
- Change passwords regularly, using a unique combination of numbers, letters and case types
- Use the internet with caution and ignore pop-ups, drive-by downloads while surfing
- Taking the time to research the basic aspects of computer security and educate yourself on evolving cyber-threats
- Perform daily full system scans and create a periodic system backup schedule to ensure your data is retrievable should something happen to your computer.

Apart from these, there are many ways you can protect your computer system. Aspects such as encryption and computer cleaners can assist in protecting your computers and its files.

Unfortunately, the number of cyber threats are increasing at a rapid pace and more sophisticated attacks are emerging. So, having a good foundation in cybersecurity concepts will allow you to protect your computer against ever-evolving cyber threats.

Steps to ensure Computer Security

In order to protect our system from the above-mentioned attacks, users should take certain steps to ensure system security:

1. Always keep your Operating System up to date. Keeping it up to date reduces the risk of their getting attacked by malware, viruses, etc.

2. Always use a secure network connection. One should always connect to a secure network. Public wi-fi's and unsecured networks should be avoided as they are at risk of being attacked by the attacker.

3. Always install an Antivirus and keep it up to date. An antivirus is software that scans your PC against viruses and isolates the infected file from other system files so that they don't get affected. Also, we should try to go for paid anti-viruses as they are more secure.

4. Enable firewall. A firewall is a system designed to prevent unauthorized access to/from a computer or even to a private network of computers. A firewall can be either in hardware, software or a combination of both.

5. Use strong passwords. Always make strong passwords and different passwords for all social media accounts so that they cannot be key logged, brute forced or detected easily using dictionary attacks. A strong password is one that has 16 characters which are a combination of upper case and lower case alphabets, numbers and special characters. Also, keep changing your passwords regularly.

6. Don't trust someone easily. You never know someone's intention, so don't trust someone easily and end up giving your personal information to them. You don't know how they are going to use your information.

7. Keep your personal information hidden. Don't post all your personal information on social media. You never know who is spying on you. As in the real world, we try to avoid talking to strangers and sharing anything with them. Similarly, social media also have people whom you don't know and if you share all your information on it you may end up troubling yourself.

8. Don't download attachments that come along with e-mails unless and until you know that e-mail is from a genuine source. Mostly, these attachments contain malware which, upon execution infect or harms your system.

9. Don't purchase things online from anywhere. Make sure whenever you are shopping online you are doing so from a well-known website. There are multiple fraud websites that may steal your card information as soon as you checkout and you may get bankrupt by them.

10. Learn about computer security and ethics. You should be well aware of the safe computing and ethics of the computing world. Gaining appropriate knowledge is always helpful in reducing cyber-crime.

11. If you are attacked, immediately inform the cyber cell so that they may take appropriate action and also protect others from getting attacked by the same person. Don't hesitate to complain just because you think people may make your fun.

12. Don't use pirated content. Often, people try to download pirated movies, videos or web series in order to get them for free. These pirated content are

at major risk of being infected with viruses, worms, or malware, and when you download them you end up compromising your system security.

Trusted Computing Base

- A trusted computing base (TCB) is everything in a computing system that provides a secure environment for operations. This includes its hardware, firmware, software, operating system, physical locations, built-in security controls, and prescribed security and safety procedures.
- It's consists of multiple components. All these components work together to secure the computing system in expected and desired ways. As a result, if any one trusted component is compromised, the entire system may be compromised and fail to behave as expected.

The Major Components of TCB are:

- Security Assessment.
- Security-by-design.
- Trusted Computing Base.
- Security Life-cycle.

The **TCB achieves system security** by means of:

- provisioning methods like controlling access
- giving privileges only to specific applications or processes
- enforcing authorization to access specific resources
- enforcing user authentication
- taking regular data backups
- installing antivirus and antimalware software

Characteristics or guiding principles of a trusted computing base

An effective TCB has the following characteristics:

- **Tamperproof.** No external part of the computing system should be able to modify or tamper with the TCB's code or state. This will ensure that the TCB's integrity is maintained.
- **Not bypassable.** There should be no way to bypass the TCB to breach the system's security.

- **Verifiable.** Admins should be able to verify the TCB's correctness to ensure that its features and subsystems are secure.
- **Simple.** A simple TCB is easier to verify and maintain than a complex trusted computing implementation.

Threat Modeling

- Threat modeling is a structured process with these objectives: identify security requirements, pinpoint security threats and potential vulnerabilities, quantify threat and vulnerability criticality, and prioritize remediation methods.

Threat modeling methods create these artifacts:

- An abstraction of the system
- Profiles of potential attackers, including their goals and methods
- A catalog of threats that could arise

How does threat modeling work?

Threat modeling works by identifying the types of threat agents that cause harm to an application or computer system. It adopts the perspective of malicious hackers to see how much damage they could do. When conducting threat modeling, organizations perform a thorough analysis of the software architecture, business context, and other artifacts (e.g., functional specifications, user documentation). This process enables a deeper understanding and discovery of important aspects of the system. Typically, organizations conduct threat modeling during the design stage (but it can occur at other stages) of a new application to help developers find vulnerabilities and become aware of the security implications of their design, code, and configuration decisions.

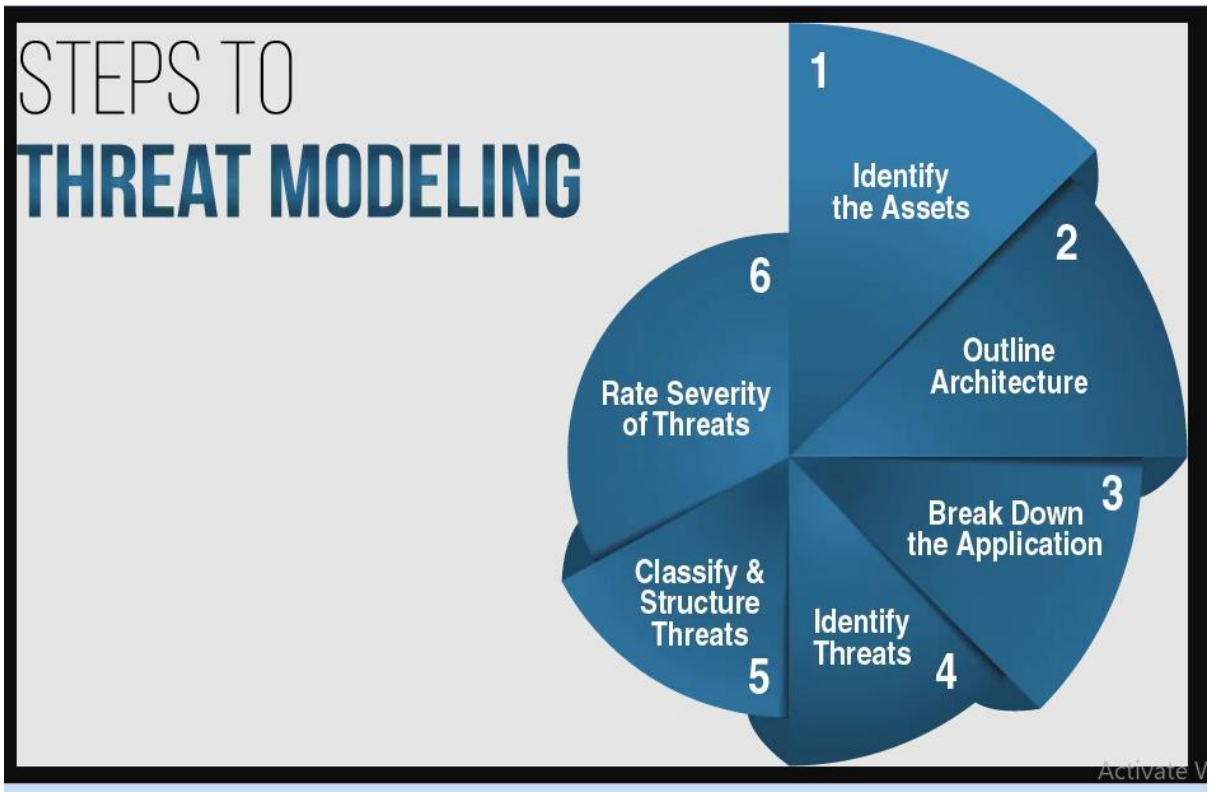


Figure: **Process in Threat Modeling**

Advantages of Threat Modeling

When performed correctly, threat modeling can provide a clear line of sight across a software project, helping to justify security efforts. The threat modeling process helps an organization document knowable security threats to an application and make rational decisions about how to address them. Otherwise, decision-makers could act rashly based on scant or no supporting evidence.

Overall, a well-documented threat model provides assurances that are useful in explaining and defending the security posture of an application or computer system. And when the development organization is serious about security, threat modeling is the most effective way to do the following:

- Detect problems early in the software development life cycle (SDLC)—even before coding begins.
- Spot design flaws that traditional testing methods and code reviews may overlook.
- Evaluate new forms of attack that you might not otherwise consider.
- Maximize testing budgets by helping target testing and code review.
- Identify security requirements.
- Remediate problems before software release and prevent costly recoding post-deployment.
- Think about threats beyond standard attacks to the security issues unique to your application.

- Keep frameworks ahead of the internal and external attackers relevant to your applications.
- Highlight assets, threat agents, and controls to deduce components that attackers will target.
- Model the location of threat agents, motivations, skills, and capabilities to locate potential attackers in relation to the system architecture.

BEST PRACTICES OF THREAT MODELING

- The killer application of threat modeling is promoting security understanding across the whole team. It's the first step toward making security everyone's responsibility. Conceptually, threat modeling is a simple process. So consider these five basic best practices when creating or updating a threat model:

1. Define the scope and depth of analysis. Determine the scope with stakeholders, then break down the depth of analysis for individual development teams so they can threat model the software.

2. Gain a visual understanding of what you're threat modeling. Create a diagram of the major system components (e.g., application server, data warehouse, thick client, database) and the interactions among those components.

3. Model the attack possibilities. Identify software assets, security controls, and threat agents and diagram their locations to create a security model of the system (see Figure 1). Once you've have modeled the system, you can identify what could go wrong (i.e., the threats) using methods like STRIDE.

4. Identify threats. To produce a list of potential attacks, ask questions such as the following:

- Are there paths where a threat agent can reach an asset without going through a control?
- Could a threat agent defeat this security control?
- What must a threat agent do to defeat this control?

5. Create a traceability matrix of missing or weak security controls. Consider the threat agents and follow their control paths. If you reach the software asset without going through a security control, that's a potential attack. If you go through a control, consider whether it would halt a threat agent or whether the agent would have methods to bypass it.

Threat Modeling Methods And Tools

CIA method
As a starting point, use the CIA (confidentiality, integrity, availability) method to define what needs protecting in the organization. For example, there may be sensitive customer information (confidentiality), company operational or proprietary data (integrity), or reliability of a service such as a web portal (availability).

Attack trees
Attack trees are a graphic representation of systems and possible vulnerabilities. The trunk of the attack tree is the asset, while entry points and threats are branches or roots. Attack trees are often combined with other methods.

STRIDE
Developed by Microsoft, STRIDE (spoofing, tampering, repudiation, information disclosure, denial of service, elevation of privilege) is one of the oldest and most widely used frameworks for threat modeling. STRIDE is a free tool that will produce DFDs and analyze threats.

PASTA
PASTA (process for attack simulation and threat analysis) is a framework designed to elevate threat modeling to the strategic level, with input from all stakeholders, not just IT or security teams. PASTA is a seven-step process that begins with defining objectives and scope. It includes vulnerability checks, weakness analysis, and attack modeling, and ends with risk and impact analysis expressed through scoring.

Trike
An open-source tool available as a spreadsheet template or stand-alone program, Trike consists of a matrix combining assets, actors, actions, and rules. When parameters and data are entered in this matrix, the program produces a score-based analysis of risks and probabilities.

VAST
VAST (visual, agile, and simple threat) modeling consists of methods and processes that can be easily scaled and adapted to any scope or part of an organization. The results produce benchmarks that can be used to make reliable comparisons and measurements of effective risk across a whole organization.

Persona non grata

This method is similar to criminal profiling in law enforcement. To anticipate attacks in more detail, brainstorming exercises are performed to create a detailed picture of a hypothetical attacker, including their psychology, motivations, goals, and capabilities.

LINDDUN

The LINDDUN framework focuses on analysis of privacy threats, based on the categories that form its acronym: linkability, identifiability, non-repudiation, detectability, disclosure of information, unawareness, and non-compliance. It uses threat trees to help users choose the relevant privacy controls to apply.

Advanced Techniques for Mapping Security Requirements into Design Specifications

Introduction

Nowadays, the information security is demanding a great attention due to a large number of discovered vulnerabilities in the applications/systems announced as secure. It is well known it is very hard to build an application with no bugs and/or security breaches, nevertheless, the companies cannot give up improving development processes and adapting them to the current scenarios.

Besides the several publications of academic researchers and industries about the importance of security practices in the System Development Life Cycle (SDLC), there is a paradox for implementation. Most of development centers fail to apply the recommendations due to resistance to new processes and mindset adaptation. It also common the engineers and developers are resistant to accept their applications are subject to security flaws.

Even the development teams understand the importance of new security paradigm for SDLC, although, they are concerned about it, unfortunately, this is not enough. In order to reach the security goals, it is necessary a detailed and deeply knowledge on security procedures and techniques. A comprehensive Security Policy (SP) is the most important reference to guide a security development and all hardware engineers, developers, application architects, software engineers, testers and project leaders must see it as a law. It establishes rules for all development phases: requirements, design/architecture, implementation, testing and maintenance, and shall define the responsibilities for all roles involved in development process. Also, establish rules for requirements phase must attend the security principles, such as information security, integrity, privacy, confidentiality, Information availability, continuity, based on environment and public threats to the system. Certainly, process and its implementation require preparation time and a detailed planning.

This presentation will cover the Security Aspects on Requirements Analysis, the first step of SDLC. This is the proper step to identify and define the security specification targets, the necessary methods for the system and how important these activities are. Next, a discussion about security principles that support Requirements Elicitation, analysis and specification disciplines is brought to the audience. Of course, a model must be flexible to fit all needs and it will be presented hints on how to adapt the Security Requirements Modeling and, finally, a practical case of study will be presented to demonstrate the concepts in practical way.

Security Principles

A considerable amount of applications and systems have been faced serious security threats due to the large number of new available technologies and the lack of knowledge and investigation about them. In the past, security concerns were basically around network infrastructure layers. Currently, due to the growing use of networks and the Internet concept dominance, such as cloud computing, Software as a Service (SaaS), serious vulnerabilities are being discovered by attackers in the application layer. Therefore, the concept of application security layer emerged as an essential task in the development process.

According to Federal Information Processing Standard (FIPS) (The National Institute of Standards and Technology (NIST), 2010) there are three security core principles that guide the information security area:

- **Confidentiality:** preserve the access control and disclosure restrictions on information. Guarantee that no one will be break the rules of personal privacy and proprietary information;
- **Integrity:** avoid the improper (unauthorized) information modification or destruction. Here is included ensure the non-repudiation and information authenticity;
- **Availability:** the information must be available to access and use all the time and with reliable access. Certainly, it just must be true for those who have right of access.

When these principles are broken, it is defined the level of impact that they can generate for the organizational information, organizational assets or individuals. Generally, the impact is qualified as:

- **Low:** generate a limited adverse effect;
- **Moderate:** generate a serious or critical adverse effect;
- **High:** generate a severe or catastrophic adverse effect.

Bringing to application SDLC, these principles can be sum up in four verbs with a simple explanation:

- **Identify:** allow users tell to application who they are;
- **Authenticate:** verify the credentials of users;
- **Authorize:** define the rights and permissions for users/third parts;
- **Audit:** do evaluation for users usability.

The core security principles are the pillar of security and must be whole understood until apply security for SDLC. It is recommended the book Information Security: Principles and Practice (Stamp, 2011) for readers who need additional security knowledge.

Security into Requirements Model

The analysis is the most essential activity to obtain the understanding between the development team and the business team. It maps the information to develop systems/applications in accordance to business and user needs, provided by stakeholders as high-level statements on features and functionalities.

In order to cover security aspects it is necessary to bring together business, development and security teams to understand the key sensitivities and business consequences caused by risk of security flaws. Since SDLC is a feed forward process and errors introduced in this phase will be spread in the next development phases, it is important to analyze security risks at very early stages.

Currently security requirements are categorized as nonfunctional requirements, which are usually defined as attributes of the software, skipping to be mapped to proper functional requirements and therefore, may not built into the software neither tested appropriately. However, what is the security impact if the security requirements are not captured or defined? Probably the result application may not be assessed for success or failure prior to implementation. Map the nonfunctional to functional requirements, the security requirements become a part of the overall requirements analysis process and, in this case, if conflicts are inevitable than they need to be identified and treated properly.

To show how this discussion has high security impact for application, the IATAC published the report Software Security Assurance (Goertzel, et al., 2007) where the authors met the main security-specific issues involved in requirements engineering, among them:

- The people involved are not likely to know or care about nonfunctional requirements. Stakeholders have a tendency to take for granted nonfunctional security needs;
- Traditional techniques and guidelines tend to be more focused on functional requirements;

- Security controls are perceived to limit functionality or interfere with usability;
- Stakeholders must understand the threats facing a system in order to build defenses against them;
- The users who help define the system are not typically the abusers from whom the system must be protected;
- It is more difficult to specify what a system should not do than what it should do.

Note that there is a high level of complexity to deal with this change, mainly for the stakeholders. In fact, they think about what their system need to do, instead of what it should not do.

After this brief discussion, all security requirements shall be captured by requirements analyst and analyzed by security team as part of functional requirements and added in the Security Requirements Specification (SecRS) document, which may be a section in the System Requirements or a Software Requirements Specification. Exhibit 1 show some expected information for the SecRS.

Exhibit 1 - Items waited on the SecRS document.

To reach good results during the security specification, the requirements analyst needs to spend special attention with the Stakeholders. He/she shall consider they have not enough security experience and so, there is a big chance to security be the last thinking. To avoid it, elaborating a questionnaire it is a good approach. The following example set of questions can help to establish the questionnaire roadmap:

- Secure against what?
- Secure against whom?
- What is to be secure?
- Who or what should provide the security?
- In which way will be the security provided?

Yet, knowing the security risks involved to the application and its impacts on the business can also facilitate the work. In this way, some examples of risks are listed in the Exhibit 2.

Exhibit 2 - Risk categories and the possible impact.

After the risk classification, the Analyst has two main activities:

- Define what need to be protected – Analyze assets and their value. A correct analysis of the asset values can facilitate the definition of the security goals and avoid unnecessary efforts;

- About what should be protected from – Analyst should have a good knowledge about the potential threats that can damage assets and the business. This activity helps to avoid mistakes when defining the controls to protect the assets and information.

With this information and going back to the security principles discussed in the previous section makes possible defining the actions to reach the security goals. Following, Exhibit 3 includes some examples based on OWASP Application Threat Modeling (OWASP Foundation, 2014).

Exhibit 3 - Actions to be performed to reach the security goals.

Summarizing, the security requirements must cover areas such as:

- Authentication and password management
- Authorization and role management
- Audit logging and analysis
- Network and data security
- Code integrity and validation testing
- Cryptography and key management
- Data validation and sanitization
- Third party component analysis

Tasks for Security Requirements Development

Certainly, some security models were proposed to treat the security concepts into the requirements phase. There are some well know security models, such as CLASP (Comprehensive, Lightweight Application Security Process) (OWASP Foundation, 2014), Secure Development Lifecycle (SDL) (Microsoft Corporation, n.d.), Security Quality Requirements Engineering (SQUARE) (Mead, Hough, & Stehney II, 2005), Secure Requirements Engineering Process (SREP) (Mellado, Fernandez-Medina, & Piattini, 2006), etc.

Looking at these models, CLASP model was chosen as base for the proposed set of tasks that can help the requirements analyst because it is a model developed "after years of extensive field work in which system resources of many development lifecycles were methodically decomposed in order to create a comprehensive set of security requirements". Below, there are six basic tasks describing the proposed adapted model.

1. Specify Operational Environment

This task focuses on understanding the operational context and its relationship to the secure system building. In order to map the operation environment, the Analyst must perform the following activities:

- **Identify requirements related to individual hosts** – Identify anything that could be potentially security-relevant. For example, if the project is expected to interact with important system components or libraries bundled into the Operating System (OS).
- **Identify requirements related to network architecture** – The network environment brings concerns like topology, configuration details, available services, single-sign-on mechanisms and type of protection configured.

2. Identify Resources and Trust Boundaries

This activity gives an architectural vision on system security requirements. According to CLASP model, this task is composed by two activities:

- **Identify network-level design** – When it comes to identify network-level design, the most difficult work is to identify all product/software parts that depend on network environment. For example, client software middleware and any database should be identified. When identifying components, it is important denote trust boundaries, such as firewalls. A network-level design developed through diagram coding can facilitate communication.
- **Identify data resources** – For identifying data resources, the Analyst needs to identify data resources that might be used by the application as much granular as possible. Some example of resources include: databases and database tables, ACLs, cryptographic key stores, audit logs, configuration files, web pages, registry keys, etc.

3. Identifying User Roles and Resource Capabilities

Everything users may be able to do into systems needs a good understanding. It is common restricting sensitive operations applying the principle of the least privilege by binding capabilities to roles only when necessary. The Analyst must perform the following activities:

- **Identify the Resources** – Thinking in security vulnerabilities, threats could be originated in any place of system/product, therefore it is important to identify each resource and respective capabilities.
- **Identify the User Roles** – An important task for the Analyst is mapping the capabilities to the classes of users. In addition, it is necessary to define one role for every set of resource capabilities

one might want to allow. It is recommended mapping roles to static sets of capabilities and specifying the default set of capabilities for the role as well as the maximum set of capabilities for the role.

- **Establish the Access Control mechanisms** – All the operations on resources shall be mediated via access control mechanisms such as reading, writing and executing privileges on a file system. Nevertheless, it is also necessary to control other operations that could be considered "meta-operations" are often overlooked as setting file ownership or reading and writing file attributes. Remembering the system, itself can have an implicit role (or set of roles) with all capabilities and mediates access to them (client-server application).

4. Document Security-Relevant Requirements

The Analyst must document the following requirements categories:

- **Goals** – This document must cover all information about security risks and additionally, determines risk mitigations for each resource – i.e., which risks on individual resources need to be addressed.
- **Content** – Risks on capabilities may differ depending on the lifetime of a system and, when specifying functional requirements for protecting data, it should be explicitly considered. If data-flow diagrams are available for the system, one should trace each core security service at each step, particularly assessing whether currently identified controls are valid at each trust level. Functional requirements should specify what mechanisms should be put in place to provide security services on resources. Such mechanisms address particular risks and if the chosen mechanisms may not address all risks, every time a new risk is identified, the security expert needs to insert it into requirements document and the mitigation plan must be updated. When a system has multiple levels of requirements, it is a good practice to divide in global, business and functional requirements.

5. Detail Misuse Cases

Sindre and Opdahl firstly proposed the misuse cases (Sindre & Opdahl, Eliciting security requirements by misuse cases, 2000), (Sindre & Opdahl, 2005), however, John McDermott and Chris Fox (McDermott & Fox, 1999) was the pioneers to show the concept of abuse cases. Although both means the same, the name misuse case seems to be more common and used. According to authors, "a misuse case is the inverse of a use case, i.e., a function that the system should not allow". They also defined the misusers or mis-actors as the inverse of actor, i.e. "an actor that one does not want the system to support, an actor who initiates misuse cases".

Based on those works and others from Ian Alexander, (Alexander, 2003), (Alexander, 2003), the requirement Analyst must identify the possible misuse cases using the following activities:

- **Motivation** – Sometimes potential risks may be lost during discussions with stakeholders or, they may not be comprehended. So, one interesting activity and recommended by several process, is the application of misuse cases.
- **Goals** – This is a specific technique to help the requirements Analyst to communicate potential risks to stakeholders. They are almost identical to use cases; notwithstanding, they are developed to detail common attempted abuses of the system.
- **Content** – Like use cases, abuse cases require understanding the system actors. Whenever possible, those actors should be mapped to capabilities. For each one should be designed misuse cases starting with high-level and refining them after a good understanding. It is possible to develop misuse cases recursively, going from system to subsystem levels or lower as necessary. Lower-level cases can highlight aspects not considered at higher levels, possibly forcing another analysis. It is recommended the important misuse cases should be represented visually through a shaded background. For when misuse cases were not depicted visually, and they are still important, they should be documented explicitly.

- Security threats are rarely neutralized completely by mitigation measures. Thieves pick locks and break into systems through unsuspected access paths. Partial mitigations are still useful as long as they afford a realistic increase in protection at reasonable cost. The goal of neutralizing all possible threats is of course wishful thinking and cannot be stated as a requirement.
- According to Lan Alexander (Alexander, 2003), "there is a clear relationship between the hostile actors who initiate misuse cases and exception classes (i.e. exception classes are simply named categories of exception, situations that cause system to fail)". The author shows devising threats and malign agents with misuse cases sometimes are a more powerful technique than simply stepping through a template or thinking about exceptions. By inverting the problem from use to misuse, it opens a new avenue of exploration, helping to find requirements, which might have been missed.
- Defense mechanisms should map directly to a functional requirement, or, in case of defense mechanisms be user-dependent, it should be inserted in an operational security guide.

- Besides, they help to discover exceptions classes, misuse cases may be used to help in other phases as test and design. For example, for test phase, it can help tester engineer to define the test plan and for design phase, to design trade-offs. Nevertheless, the main and most important contribution of misuse cases is probably related to the risk analysis process.
- Some kind of diagram tools can help to create misuse cases. It is also possible to use simple text tools with graphic capability as, for example, UMLsec or SecureUML (Jürjens, 2002).

6. Review Security Requirements

Reviews are good activities for security analysis. Therefore, although the resistance from Analysts, the review can avoid flaws in the understanding of security requirements. It is recommended that both Analyst and the Security Team make the review. Many times, even after two or more review from Analyst it is common the security team find bad description about the security requirements. Furthermore, generally the security team has more security skills that requirements Analyst and acts as a third party review.

A good practice for review security requirements is the confrontation of analysis between Analyst and the Security Team analysis. This approach can be seen as a trust checking and can avoid misunderstanding.

End of Unit-I | The above Material is Prepared by **Dr. M V Kamal**

UNIT - II

SECURITY ASSESSMENT

&

RISK ANALYSIS

UNIT-II

By

Dr. M V Kamal

Professor & HoD

Dept. of Emerging Technologies

Unit-II

Syllabus

Instructor: Dr M V Kamal | Professor | CSE (ET) Dept. | MRCET

Unit_II

Topics…

- Software design and an introduction to hierarchical design representations. Difference between high-level and detailed design. Handling security with high-level design. General Design Notions. Security concerns designs at multiple levels of abstraction, Design patterns, quality assurance activities and strategies that support early vulnerability detection, Trust models, security Architecture & design reviews .

Types of Assessments

Secure software

- Secure software is defined as software developed or engineered in such a way that its operations and functionalities continue as normal even when subjected to malicious attacks.
- The systems and resources in its environment remain safe and the attacks detected and removed.

Secure Software Development

- Secure software development includes enabling software security (security requirements planning, designing a software architecture from a security perspective, adding security features, etc.) and maintaining the security of software and the underlying infrastructure.

Stages of Secure Software Development

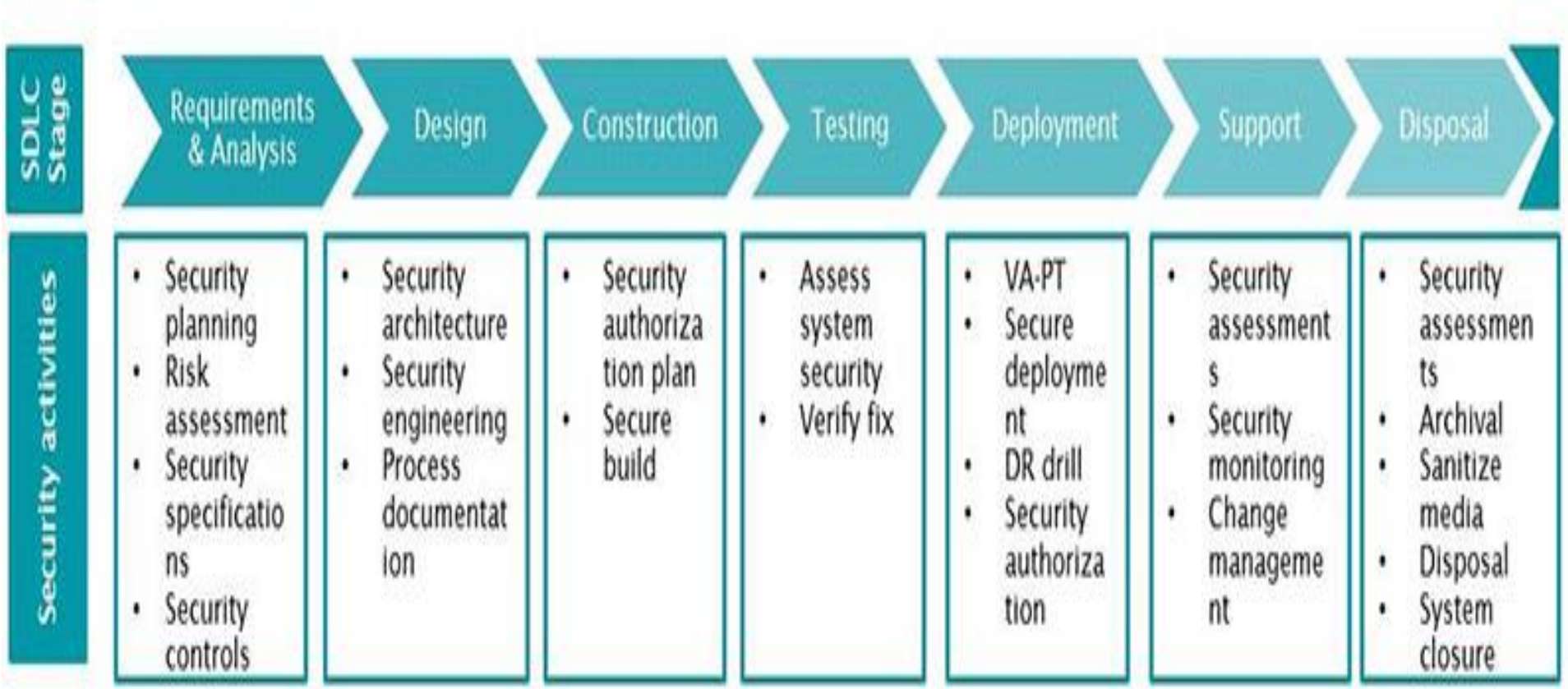
- Requirements gathering, prioritization and analysis: mapping security requirements.
  - Identification requirements
  - Authentication requirements
  - Authorization requirements
  - Integrity requirements
  - Non-repudiation requirements
  - Privacy requirements
  - Survivability requirements
- Software design: threat modelling, secure architecture, planning security features.

Stages of Secure Software Development

Contd..

- Software development: secure coding practices, static analysis, and regular peer review.
- Software deployment and support: penetration testing, final security review, and an incident response plan.

SDLC Stages


SDLC Stage
Requirements & Analysis
Design
Construction
Testing
Deployment
Support
Disposal
Security activities
• Security planning
• Risk assessment
• Security specifications
• Security controls
• Security architecture
• Security engineering
• Process documentation
• Security authorization plan
• Secure build
• Assess system security
• Verify fix
• VA-PT
• Secure deployment
• DR drill
• Security authorization
• Security assessments
• Security monitoring
• Change management
• Security assessments
• Archival
• Sanitize media
• Disposal
• System closure

Hierarchical Design Representation

Instructor: Dr M V Kamal | Professor | CSE (ET) Dept. | MRCET

Hierarchical Architecture

- Hierarchical architecture views the whole system as a hierarchy structure, in which the software system is decomposed into logical modules or subsystems at different levels in the hierarchy.

- This approach is typically used in designing system software such as network protocols and operating systems.

Hierarchical Architecture (contd..)

- In system software hierarchy design, a low-level subsystem gives services to its adjacent upper level subsystems, which invoke the methods in the lower level.
- The lower layer provides more specific functionality such as I/O services, transaction, scheduling, security services, etc.
- The middle layer provides more domain dependent functions such as business logic and core processing services. And, the upper layer provides more abstract functionality in the form of user interface such as GUIs, shell programming facilities, etc.

High-level and Detailed Design

Instructor: Dr M V Kamal | Professor | CSE (ET) Dept. | MRCET

High-level and Detailed Design

- **High-level design** or **HLD** refers to the overall system, a design that consists description of the system architecture and design and is a generic system design that includes:
  - System architecture
  - Database design
  - Brief description of systems, services, platforms, and relationships among modules.

- **High-level design** or **HLD** is also known as **macro level designing**

Detailed Design

- Which creates a full definition of every aspect of a project development.
- Low Level Design in short LLD is like detailing HLD means it refers to component-level design process.
- It describes detailed description of each and every module means it includes actual logic for every system component and it goes deep into each modules specification.
- It is also known as micro level/detailed design. It is created by designers and developers.
- It converts the High Level Solution into Detailed solution.

Difference between
High-level and Detailed Design

Instructor: Dr M V Kamal | Professor | CSE (ET) Dept. | MRCET

S.No.	HIGH LEVEL DESIGN	DETAILED / LOW LEVEL DESIGN
01.	High Level Design is the general system design means it refers to the overall system design.	Low Level Design is like detailing HLD means it refers to component-level design process.
02.	High Level Design in short called as HLD.	Low Level Design in short called as LLD.
03.	It is also known as macro level/system design.	It is also known as micro level/detailed design.
04.	It describes the overall description/architecture of the application.	It describes detailed description of each and every module.
05.	High Level Design expresses the brief functionality of each module.	Low Level Design expresses details functional logic of the module.

S.No.	HIGH LEVEL DESIGN	DETAILED / LOW LEVEL DESIGN
06.	It is created by solution architect.	It is created by designers and developers.
07.	Here in High Level Design the participants are design team, review team, and client team.	Here in Low Level Design participants are design team, Operation Teams, and Implementers.
08.	It is created first means before Low Level Design.	It is created second means after High Level Design.
09.	In HLD the input criteria is Software Requirement Specification (SRS).	In LLD the input criteria is reviewed High Level Design (HLD).
10.	High Level Solution converts the Business/client requirement into High Level Solution.	Low Level Design converts the High Level Solution into Detailed solution.
11.	In HLD the output criteria is data base design, functional design and review record.	In LLD the output criteria is program specification and unit test plan.

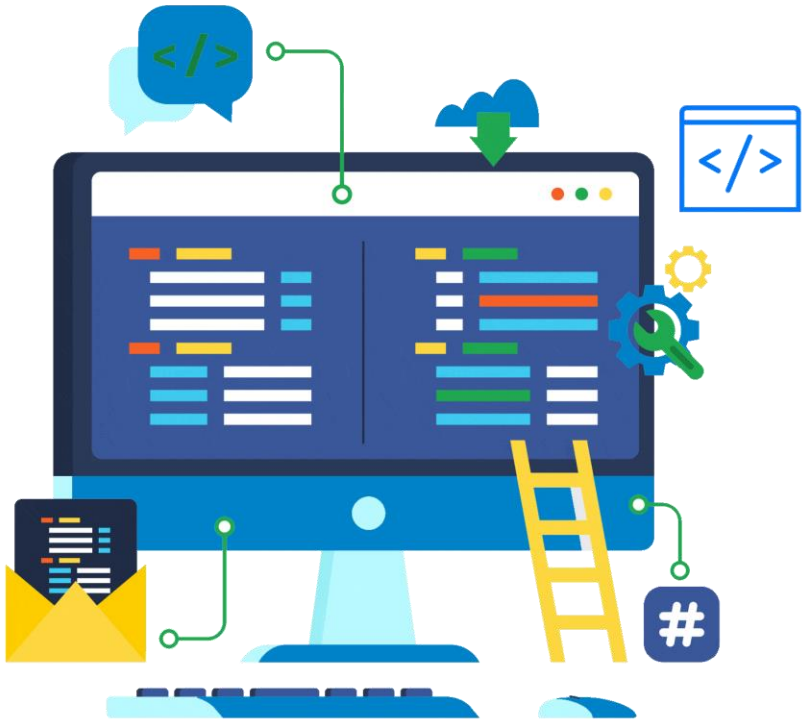
Handling Security With High-level Design

Instructor: Dr M V Kamal | Professor | CSE (ET) Dept. | MRCET

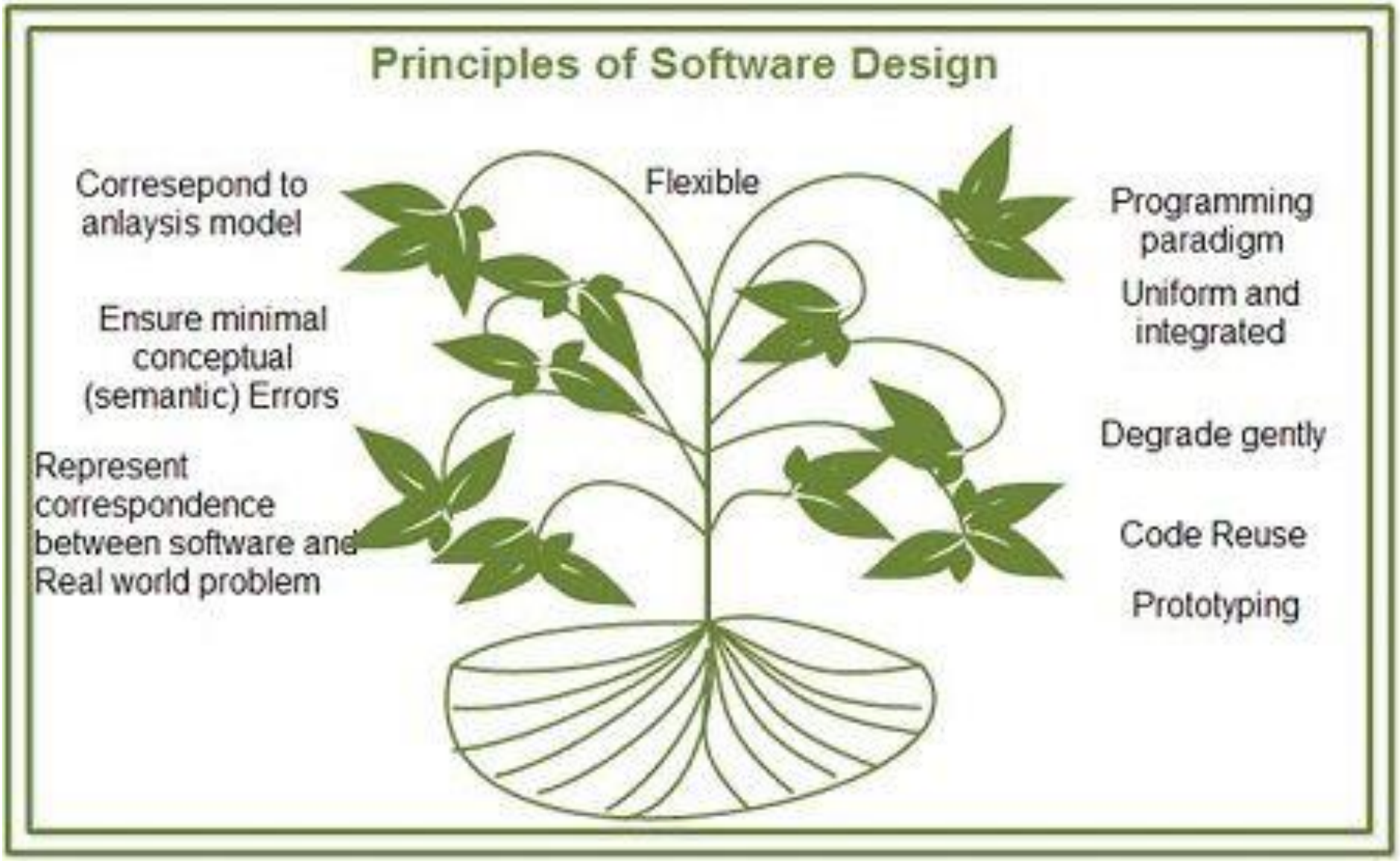
Handling Security With High-level Design

- Use best practices for secure design.
- Perform threat analysis.
- Check the Coding Standards and do the reviews
- Perform security assessment from abstract level onwards..
- Assess attacker scenarios or consequences and even plan for incident response
- Plan of action for Security in High-Level Design.
- Build the Security Plan
- Study the input data and do the validation.
- Check the data flow in the HLD.
- Identify Design Techniques that Mitigate Risks
- Identify Components Essential to Security
- Defect Management & DR Management

General Design Notions



General Design Notions



Unit_II

Topics…

- Software design and an introduction to hierarchical design representations. Difference between high-level and detailed design. Handling security with high-level design. General Design Notions. Security concerns designs at multiple levels of abstraction, Design patterns, quality assurance activities and strategies that support early vulnerability detection, Trust models, security Architecture & design reviews .

Security concerns designs at multiple levels of abstraction

What is the abstraction

Multi-level Abstraction

Instructor: Dr M V Kamal | Professor | CSE (ET) Dept. | MRCET

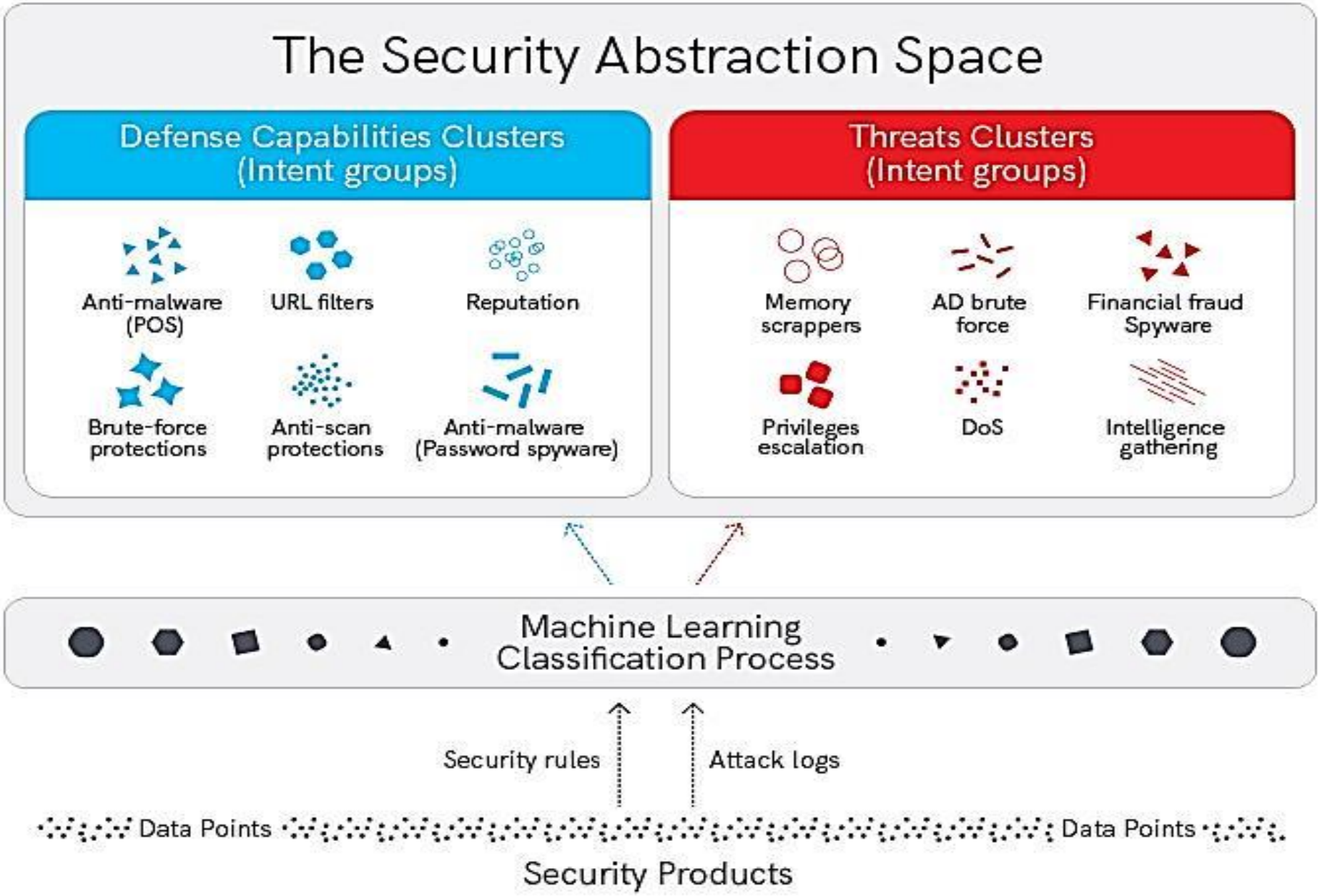
Multi-level Abstraction

- Employing multi-level abstraction in modeling refers to representing objects at multiple levels of one or more abstraction hierarchies, mainly classification, aggregation and generalization.

- Multiple representation, however, leads to accidental complexity, complicating modeling and extension

Security Abstraction

- Security Abstraction enables the generalization of complex cyber security models.
- The goal is to break down the cybersecurity ecosystem into abstract components in a way that clearly defines the security role of each one – its pros and cons –all using one common language.
- This can provide us with a simplified view of the complex security infrastructures.


The Security Abstraction Space
Defense Capabilities Clusters (Intent groups)
Anti-malware (POS)
URL filters
Reputation
Brute-force protections
Anti-scan protections
Anti-malware (Password spyware)
Threats Clusters (Intent groups)
Memory scrappers
AD brute force
Financial fraud Spyware
Privileges escalation
DoS
Intelligence gathering
Machine Learning Classification Process
Security rules
Attack logs
Data Points
Data Points
Security Products

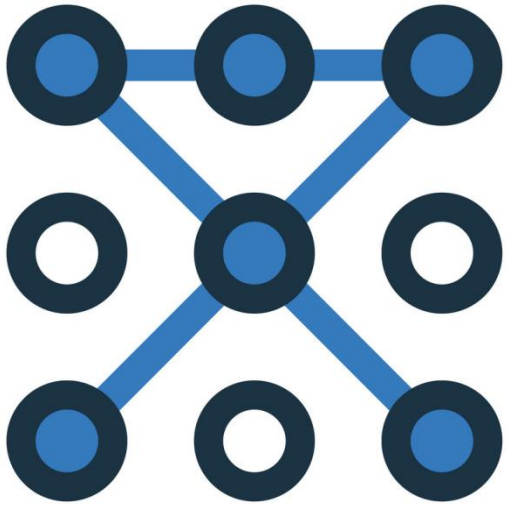
Refer the **pdf document** for more detailed explanation…

SECURE DESIGN PATTERNS

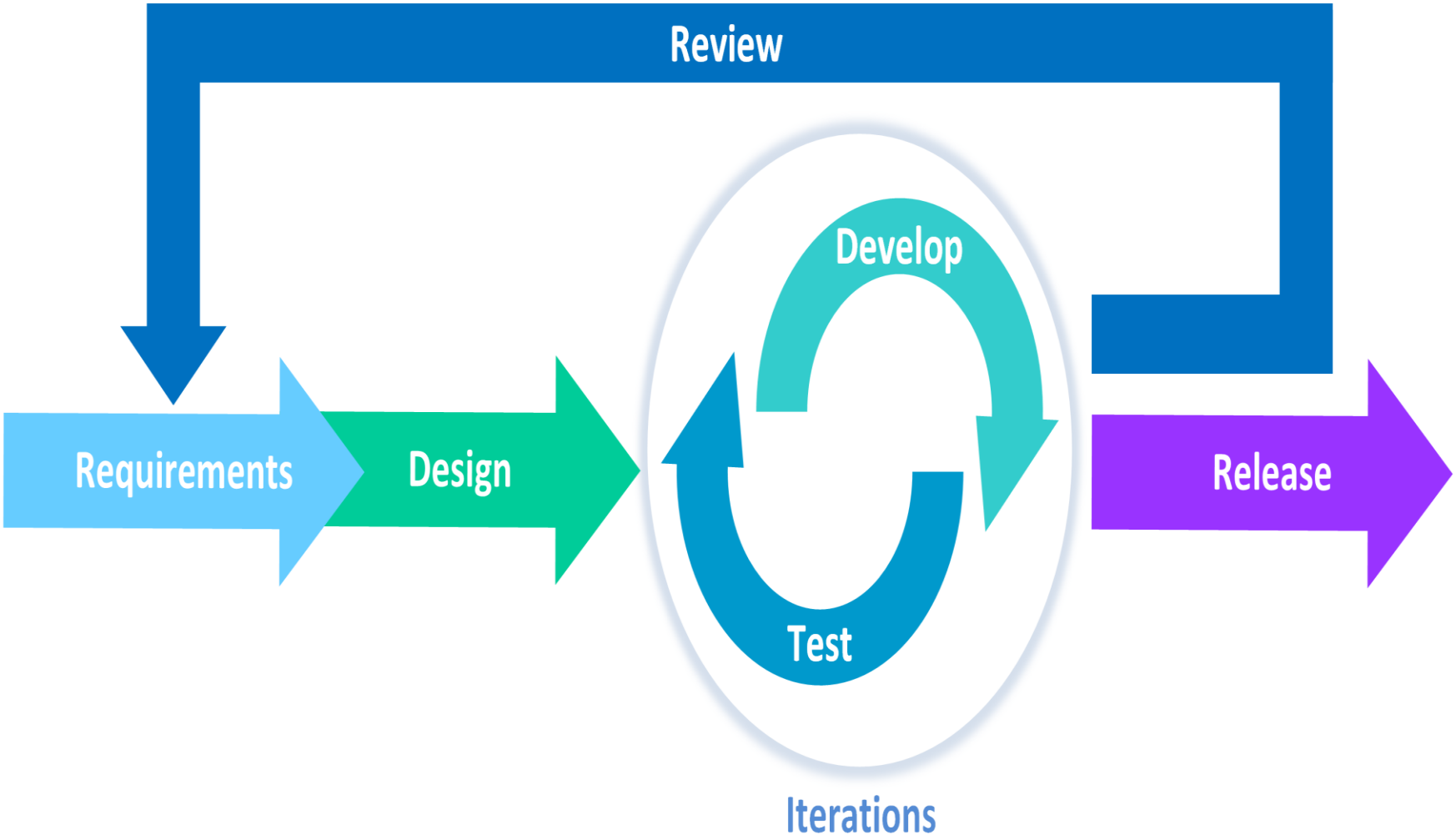
Instructor: Dr M V Kamal | Professor | CSE (ET) Dept. | MRCET

SECURE DESIGN PATTERNS

- Secure design patterns are meant to eliminate the accidental insertion of vulnerabilities into code and to mitigate the consequences of these vulnerabilities



Secure Design Patterns Contd..



Secure Design Patterns Contd..

- A pattern is a general reusable solution to a commonly occurring problem in design.
- Note that a design pattern is not a finished design that can be transformed directly into code.
- It is a description or template for how to solve a problem that can be used in many different situations.
- Algorithms are not thought of as design patterns because they solve computational problems rather than design problems

Secure Design Patterns Contd..

- Secure design patterns address security issues at widely varying levels of specificity ranging from architectural-level patterns involving the high-level design of the system down to implementation-level patterns providing guidance on how to implement portions of functions or methods in the system.

- Patterns address high-level security concerns, such as how to handle communication with untrusted third-party systems and the importance of multi-layered security.

Secure Design Patterns Contd..

- Address high-level process issues such as the use of white-hat penetration testing and addressing simple, high-impact security issues early in the system development and configuration process.

- Secure design patterns differ from security patterns in that they do not describe specific security mechanisms (such as access control, authentication, and authorization (AAA) and logging), define secure development processes, or provide guidance on the configuration of existing secure systems.

Secure Design Patterns Contd..

- Three general classes of patterns are presented in this document:
  - Architectural-level patterns
  - Design-level patterns
  - Implementation-level patterns

How to write a security pattern..

You can check from the below link…

- https://securitypatterns.io/docs/how-to-write-a-security-pattern/

Unit_II

Topics…

- Software design and an introduction to hierarchical design representations. Difference between high-level and detailed design. Handling security with high-level design. General Design Notions. Security concerns designs at multiple levels of abstraction, Design patterns, quality assurance activities and strategies that support early vulnerability detection, Trust models, Security Architecture & Design Reviews .

Quality Assurance Activities

Quality Assurance Activities


Risk Management
Change Management
Causal Analysis Resolution
Project Closure
Decision Analysis, Resolution
Intergroup Co-ordination
Training Development
Resource Management
Process Improvement
Configuration Management
Reviews, Audits
Project Planning
Estimation
Project Monitoring
Project Management Activities
Discovery
Planning
Design
Execute
Improve
Agile
Iterative
TDD
V-Model
QA Process
Horizontal Activities

Strategies that Support Early **Vulnerability Detection**

5 Ways to Detect Application Security Vulnerabilities Sooner to Reduce Costs and Risk

- Security testing has always been an important step in the application development process.
- Yet, traditional measures often occur too late in the process to effectively find and fix vulnerabilities before causing costly production delays, or worse, putting organizations at risk for potential security breaches.
- To minimize security-related costs and risks, testing needs to occur sooner and more frequently throughout the development process.

5 Ways to Detect Vulnerabilities…

1. Static Application Security Testing (SAST)

- It allowing you to identify potential issues at the coding stage, so you can resolve problems without breaking builds or allowing vulnerabilities to get passed to the final application release
- Commercial solutions such as Checkmarx help to identify hundreds of security vulnerabilities and weaknesses in custom code. You can also leverage many open source linters for your specific platforms to detect various vulnerability patterns that can compromise code security.

5 Ways to Detect Vulnerabilities...(contd..)

2. Detecting Open Source Software Vulnerabilities

- Most applications use a large number of dependencies, or third-party open source software (OSS) components. These may have various security vulnerabilities and put your application at risk.
- Tools such as Whitesource Bolt and Black Duck can scan all of your projects, not only to detect OSS components, but also identify and provide fixes for any known vulnerabilities

5 Ways to Detect Vulnerabilities...(contd..)

3. Identifying Compromising Credentials

- Human error is always a security concern, especially when it comes to credentials. Just consider how many times you've heard of developers committing code only to later realize they'd accidentally included a password. These errors can lead to high-cost consequences for organizations.
- There are many tools that scan for secrets and credentials that can be accidentally committed to a source code repository. One example is Microsoft Credential Scanner (CredScan). Perform this scan in the PR/CI build to identify the issue as soon as it happens so they can be changed before this becomes a problem.
- Once an application is deployed, you can continue to scan for vulnerabilities through the following automated continuous delivery pipeline capabilities.

5 Ways to Detect Vulnerabilities...(contd..)

4. Dynamic Application Security Testing

- Unlike SAST, which looks for potential security vulnerabilities by examining an application from the inside—at the source code—Dynamic Application Security Testing (DAST) looks at the application while it is running to identify any potential vulnerabilities that a hacker could exploit.
- OWASP Zed Attack Proxy (ZAP) is an open source tool for performing pen testing on web applications and APIs. Pen testing helps ensure that there are no security vulnerabilities hackers can manipulate. It can be installed as a client application or come configured on a docker container. OWASP ZAP scans can be incorporated into your pipeline to check every deployment for security vulnerabilities.

5 Ways to Detect Vulnerabilities...(contd..)

5. Verifying Cloud Infrastructure Security

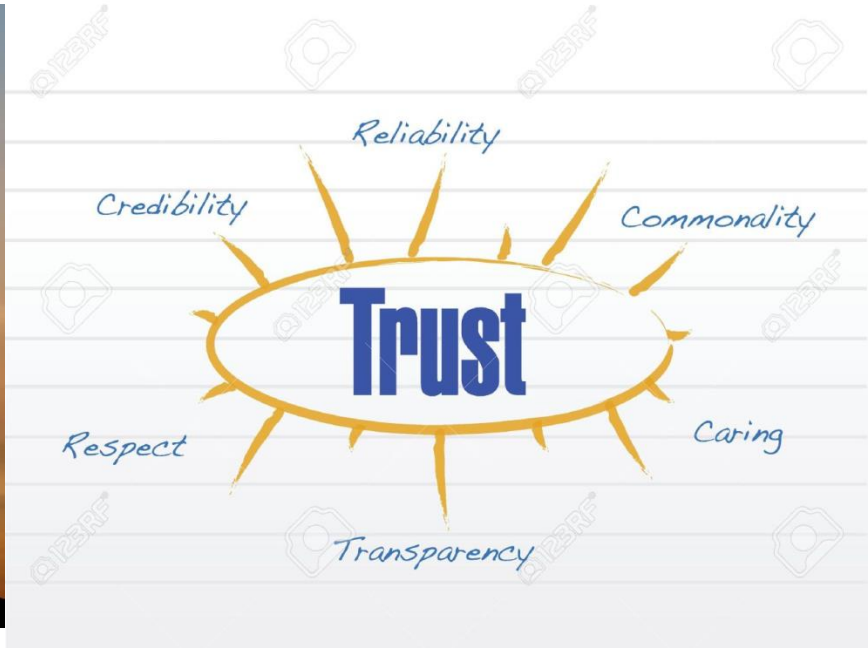
- The infrastructure should be validated to check for vulnerabilities. When using a public cloud, deploying the application and shared infrastructure is easy, so it's important to validate that everything has been done securely.
- Each public cloud includes tools to help verify that the infrastructure has been provisioned securely. APIs can be leveraged to check immediately after deployment in lower environments to help ensure any infrastructure security issues are caught before they get to production.
- Additionally, tools such as InSpec provide compliance-as-code to enforce the intent of provisioned infrastructure is always being met.

Trust Model

Instructor: Dr M V Kamal | Professor | CSE (ET) Dept. | MRCET

Trust Model

- A trust model measures the security strength and computes a trust value. A trust value comprises of various parameters that are necessary dimensions along which security of cloud services can be measured.

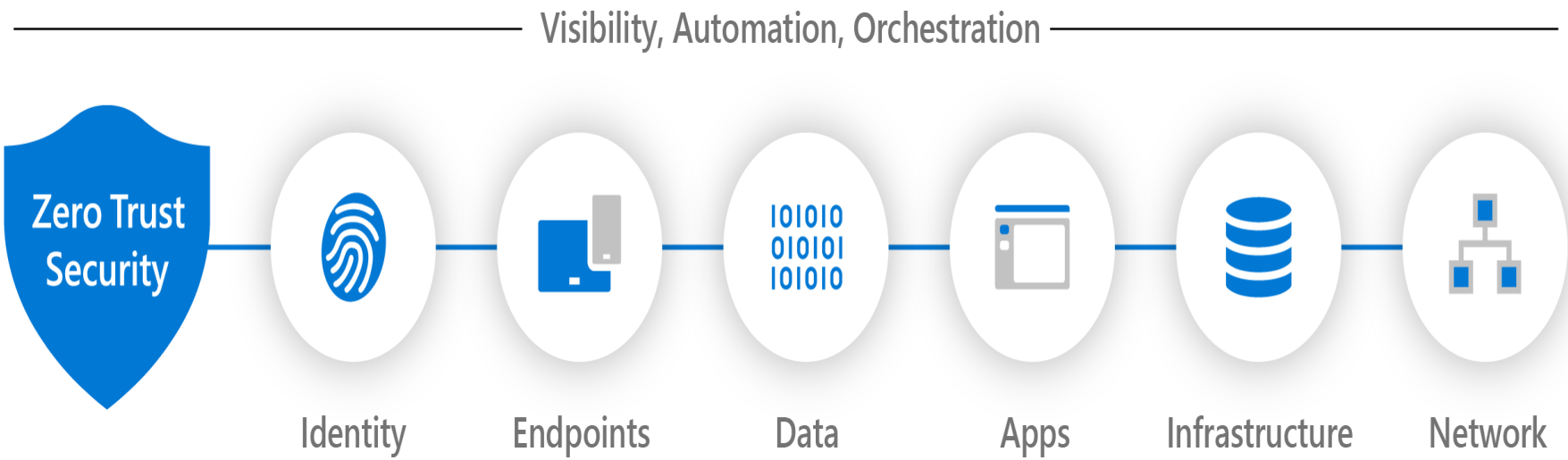


Trust Model

- A **trust model** is a collection of rules that ensure the legitimacy of the digital certificates used by the organizational components.
- It will focus on authenticity, confidentiality, integrity and non-repudiation of the information.
- Different trust models are available based on different trust anchor models and different rules to create, manage, distribute, store and revoke the digital certificates.

Zero Trust

- **Zero Trust** is a strategic approach to cybersecurity that secures an organization by eliminating implicit trust and continuously validating every stage of a digital interaction.



Zero Trust

- Rooted in the principle of "never trust, always verify," Zero Trust is designed to protect modern environments and enable digital transformation by using strong authentication methods, leveraging network segmentation, preventing lateral movement, providing Layer 7 threat prevention, and simplifying granular, "least access" policies.

Zero Trust

- Zero Trust was created based on the realization that traditional security models operate on the outdated assumption that everything inside an organization's network should be implicitly trusted.

- This implicit trust means that once on the network, users – including threat actors and malicious insiders – are free to move laterally and access or exfiltrate sensitive data due to a lack of granular security controls.

Zero Trust

Why Zero Trust…?

- Today's organizations need a new security model that more effectively adapts to the complexity of the modern environment, embraces the hybrid workplace, and protects people, devices, apps, and data wherever they're located.

Zero Trust Principles

Verify explicitly

- Always authenticate and authorize based on all available data points, including user identity, location, device health, service or workload, data classification, and anomalies.

Use least-privilege access

- Limit user access with just-in-time and just-enough access (JIT/JEA), risk-based adaptive polices, and data protection to help secure both data and productivity.

Assume breach

- Minimize blast radius and segment access. Verify end-to-end encryption and use analytics to get visibility, drive threat detection, and improve defenses.

Security Architecture

Instructor: Dr M V Kamal | Professor | CSE (ET) Dept. | MRCET

Security Architecture

- While there are various definitions of security architecture, it is ultimately a collection of security concepts, procedures, and models that balance opportunity and threat.
- Security architecture is the process of evaluating information security controls and implementing the right business process and tools into IT systems to protect the data being used and stored by an organization.
- When it comes down to it, security architecture is only the first step–security comes from implementation and operations.

Security Architecture Contd..

- When good security architecture is not properly established, an organization's teams are stuck hunting for designs and implementations that randomly protect against threats.

- Most 'cyber security architecture' is reactive and threat-oriented. However, robust security architecture will provide preventative measures that will ensure the enterprise is secured.

Security Architecture Contd..

- A good security architecture match with your benefit and risk objectives. Essentially ensuring you stay secure enough from cyber attacks.

- Business needs are translated into actionable security requirements through security architecture.

- Most 'cyber security architecture' is reactive and threat-oriented. However, robust security architecture will provide preventative measures that will ensure the enterprise is secured.

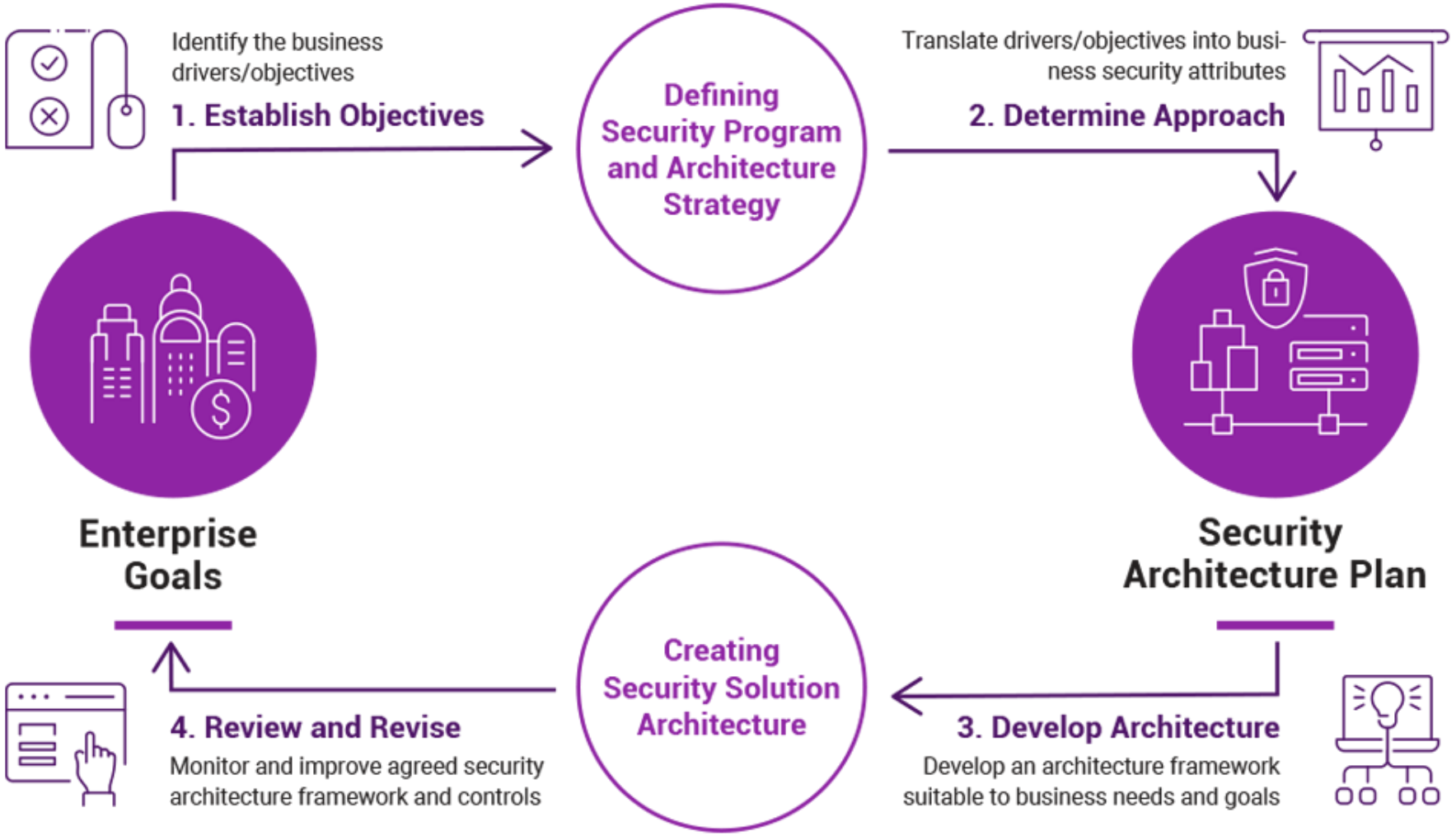
Security Architecture Contd..

- The job of a security architect is quite similar to that of a house, school, or business building architect. They analyze the property, consider aspects such as customer preferences, soil type, terrain, and climate (the land's existing condition), and then devise a strategy to attain the desired result (the blueprint). Others construct the structure, in this case, builders and contractors,

Attributes of Security Architecture

- The key attributes of security architecture..

The elements of Security Architecture



Design Reviews / Security Design Reviews

Instructor: Dr M V Kamal | Professor | CSE (ET) Dept. | MRCET

Design Reviews

- The Design Review (DR) practice is focused on assessment of software design and architecture for security-related problems.
- This allows an organization to detect architecture-level issues early in software development and thereby avoid potentially large costs from refactoring later due to security concerns.
- Pentesting is a limited form of security control. It doesn't identify the majority of your application's security issues. For that, you need a Security Design Review and Security Code Review.

Design Reviews

- Security Design Reviews are a great way to identify threat scenarios that can result in the compromise of your application.
- Investing in Security Design Reviews early can save you a lot of money, time, and resources.

Purpose of a design review

- A design review can be triggered by a range of requirements but it's common as part of the service planning and design phases to consider reviewing services and solutions to ensure:
  - They are fit for purpose
  - That the risks are known and understood
  - That the solution meets the business needs
  - That it has suitable controls regarding risks

Security Design Reviews

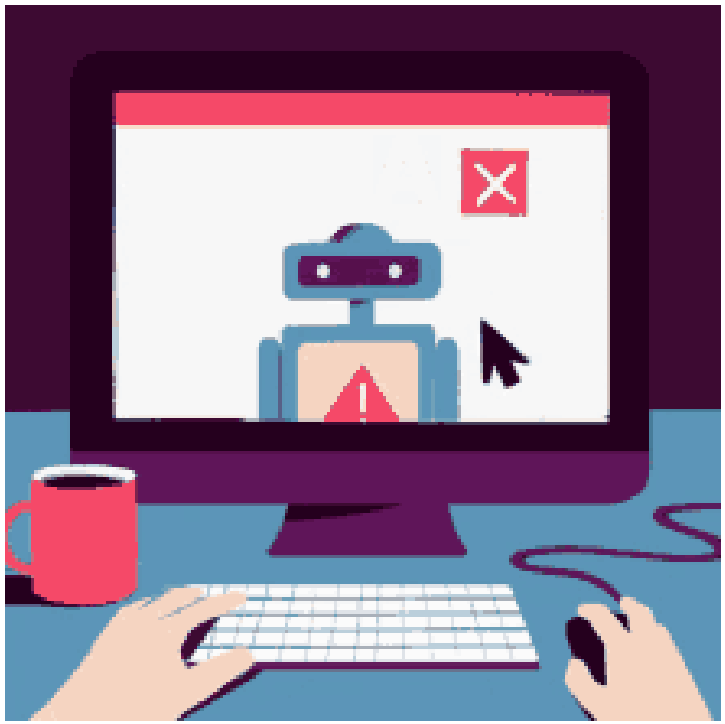
- Security design reviews have no dependency on the application being built or run in an environment.
- They can also be applied early in the SDLC and provide significant cost savings due to avoidance of costly fixes later on in the application life-cycle.
- *-Refer my document about this topic*

  (Document Name: Design Review.pdf)

Unit_II

Topics Covered…

- Software design and an introduction to hierarchical design representations. Difference between high-level and detailed design. Handling security with high-level design. General Design Notions. Security concerns designs at multiple levels of abstraction, Design patterns, quality assurance activities and strategies that support early vulnerability detection, Trust models, Security Architecture & Design Reviews .



End of Unit-II

Principles Of Software Design :

1. **Should not suffer from “Tunnel Vision” –**
   While designing the process, it should not suffer from “tunnel vision” which means that is should not only focus on completing or achieving the aim but on other effects also.

2. **Traceable to analysis model –**
   The design process should be traceable to the analysis model which means it should satisfy all the requirements that software requires to develop a high-quality product.

3. **Should not “Reinvent The Wheel” –**
   The design process should not reinvent the wheel that means it should not waste time or effort in creating things that already exist. Due to this, the overall development will get increased.

4. **Minimize Intellectual distance –**
   The design process should reduce the gap between real-world problems and software solutions for that problem meaning it should simply minimize intellectual distance.

5. **Exhibit uniformity and integration –**
   The design should display uniformity which means it should be uniform throughout the process without any change. Integration means it should mix or combine all parts of software i.e. subsystems into one system.

6. **Accommodate change –**
   The software should be designed in such a way that it accommodates the change implying that the software should adjust to the change that is required to be done as per the user’s need.

7. **Degrade gently –**
   The software should be designed in such a way that it degrades gracefully which means it should work properly even if an error occurs during the execution.

8. **Assessed or quality –**
   The design should be assessed or evaluated for the quality meaning that during the evaluation, the quality of the design needs to be checked and focused on.

9. **Review to discover errors –**
   The design should be reviewed which means that the overall evaluation should be done to check if there is any error present or if it can be minimized.

10. **Design is not coding and coding is not design –**
    Design means describing the logic of the program to solve any problem and coding is a type of language that is used for the implementation of a design.

Security Abstraction

Security Abstraction enables the generalization of complex cyber security models. The goal is to break down the cybersecurity ecosystem into abstract components in a way that clearly defines the security role of each one – its pros and cons –all using one common language. This can provide us with a simplified view of the complex security infrastructures.

How will that help?

Understanding the role of each capability (security particle) leads to instant understanding of the security event that the same capability can generate – which is the intent of the attack. Security Abstraction enables the intent of each event to be instantly identified, allowing organizations to deploy the best security particles with the relevant capabilities to handle the threat. This abstraction process makes the work of determining if a real threat exists and effectively orchestrating the proper security "troops" to best deal with it much faster and more accurate.

Some people will say this is impossible. Well, it is certainly an enormous challenge – but not impossible. This is because of some significant technology shifts have occurred in the past few years:

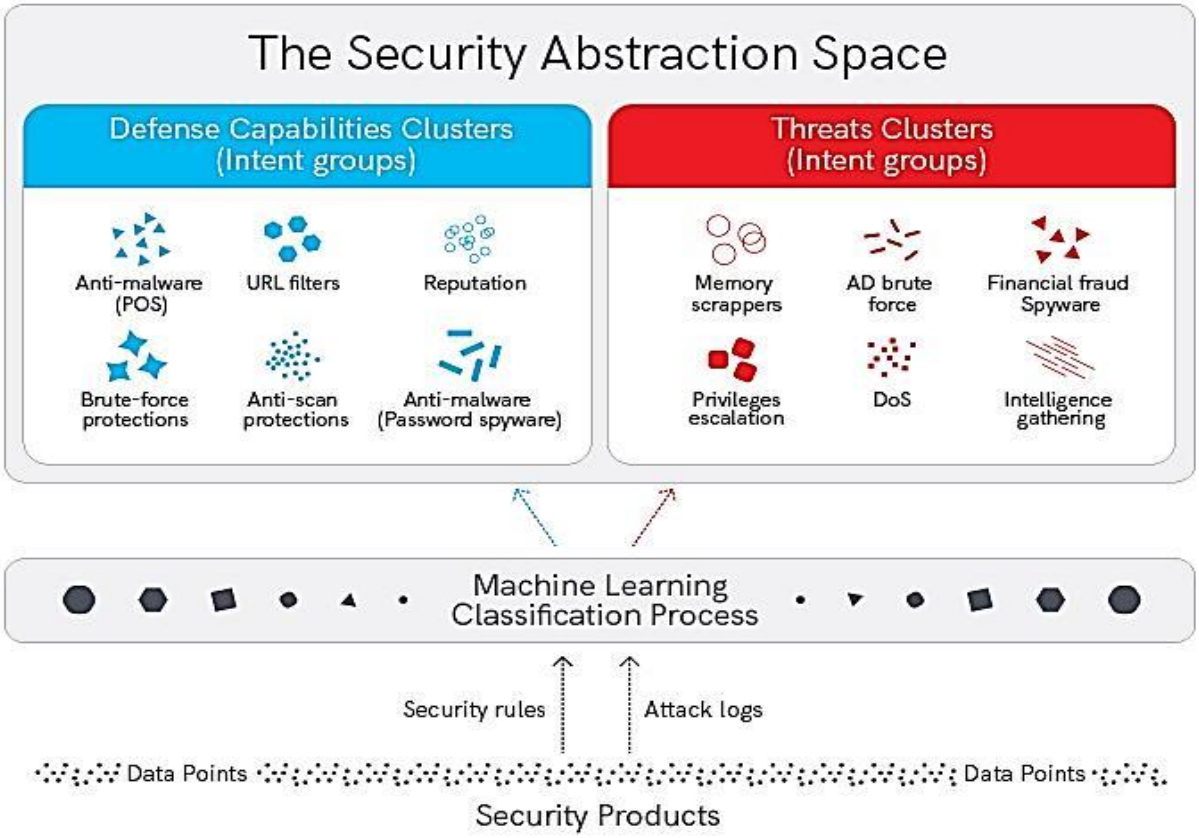
- **API** – Based on demand from significant customers, vendors are designing and developing APIs that are more predictable and stable than ever.
- **SDDC & NFV** – Customers demand more "openness" and "programmability" in order to integrate, customize, and automate the underlying networking technologies and better fit them to their business needs. This results in more "transparency" of APIs internal functions, which allows for better control over these products.
- Artificial Intelligence technologies have advanced significantly, especially Machine Learning technologies that can artificially understand meaning and context, i.e., classification algorithms.

  All these have created a foundation which makes security abstraction a reality.

Abstraction of complex security infrastructure means two things:

- Analyzing and breaking down security tools and services into their various detection, investigation and mitigation or remediation capabilities. This should be done continuously, because security products are constantly updated with new security data and hence, new capabilities (e.g., weekly updates of intrusion attack DBs, new malware hashes, etc.).
- Abstracting threats by analyzing the huge number of logs and classifying them into groups of security intents.

Machine learning is all about solving the problem of how to assign a new data point into a cluster based on previous learning processes. In the world of cyber-security, these data points can be attack signatures (in IDS/IPS attack databases), malware names and descriptions, or hashes, security rules etc., and the clusters represent groups of security intent. Each cluster has a different security meaning (a threat intent or a defense intent). This process is illustrated below:



Per the illustration above, it is clear that by understanding the threat intent of the security logs (the threat clusters on the right side), it becomes simple to correlate them with the best security defense capabilities (the defense clusters on the left side).

Machine learning plays a major role in creating this abstract view. It facilitates the creation of an adaptive security taxonomy which provides an understanding of both the security capabilities and of the threats themselves (which is the other side of the same coin), thus enhancing the ability to analyze a very large amount of security data. This enables instant data correlation, because when the intent is clear, correlation becomes simple. Lastly, it allows one to correlate the threat (per its intent) with the best response capability.

The value of machine learning classification algorithms goes beyond just clustering. The fact that these algorithms can work on almost any source of data points and group them into security clusters, i.e., into security intent groups, makes them agnostic to the underlying security products. This means that the security taxonomy created can be considered an abstract layer that sits on top of any medium to large organization's existing security configuration, and provides the required clarity (discussed above), no matter which set of security vendors the organization decides to use.

Quality Assurance Activities

Quality Assurance Activities or QAA is designed for product evaluation and process monitoring. They also assure that the product development and associated processes are correctly carried out as per the process control plan. Products are monitored for conformance to standards and operations are monitored for conformance to procedures. Quality Assurance Activities not only assure the existence of clear and achievable standards but also evaluate the compliance of the products to the established standards.

In Quality Assurance Activities, audits are the key technique used to perform product evaluation and process monitoring. At the same time, they also assure that appropriate steps to carry out the process are being followed. Moreover, management plan review ensures that appropriate methods are regularly supplement to these processes. Audit report to management consists of the findings and recommendations to bring the development into conformance with standards and/or procedures. QAA assures that:

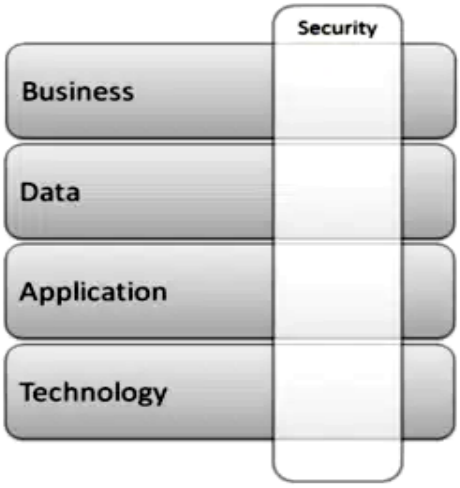
- ✓ All the operation and production activities are performed in accordance with the quality plans, standards and procedures.
- ✓ Configuration control is maintained in critical phases of testing, acceptance and delivery.
- ✓ Authentication is established by a series of reviews that exhibit the performance required by the standard or contractual specification.
- ✓ Verification and validation activities are done by monitoring technical reviews, inspections and walkthrough.
- ✓ Formal testing is done in accordance with plans and procedures.
- ✓ Requirements are complete, testable and properly expressed.

What is Security Architecture?

While there are various definitions of security architecture, it is ultimately a collection of security concepts, procedures, and models that balance opportunity and threat. Everything your business does creates the possibility of benefit and threats. You will have a different risk tolerance in different parts of your business. A good security architecture match with your benefit and risk objectives. Essentially ensuring you stay secure enough from cyber attacks. Business needs are translated into actionable security requirements through security architecture.

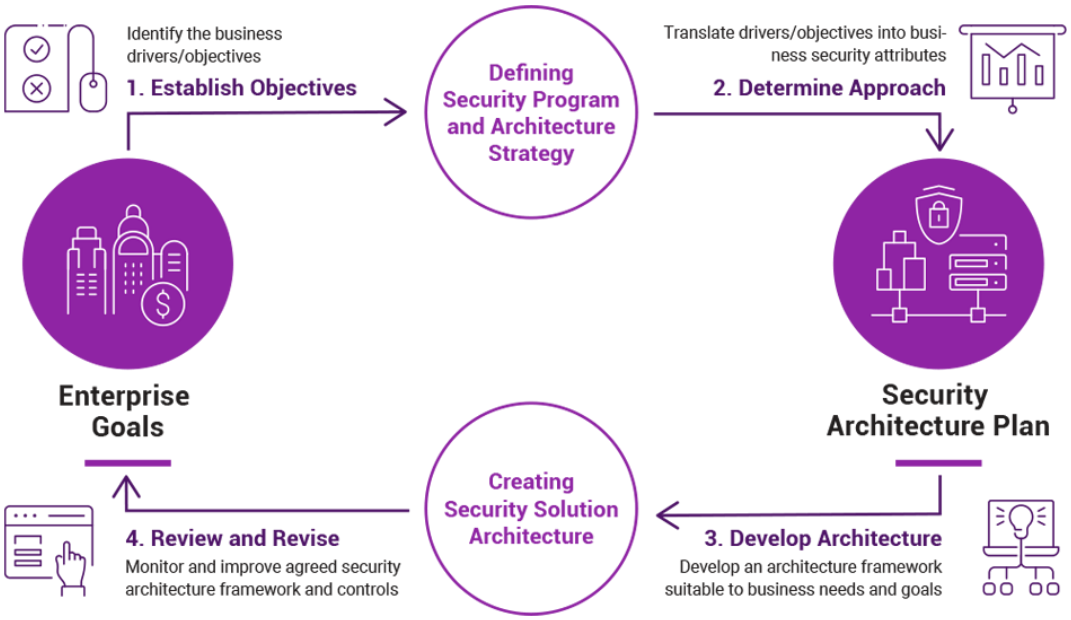
One way to immediately grasp it is to compare it to traditional architecture. The job of a security architect is quite similar to that of a house, school, or business building architect. They analyze the property, consider aspects such as customer preferences, soil type, terrain, and climate (the land's existing condition), and then devise a strategy to attain the desired result (the blueprint). Others construct the structure, in this case, builders and contractors, under the supervision of the architect to guarantee it achieves the goal (Architecture Governance).
The goal of most security architectures is to safeguard the enterprise against cyber threats. Security Architects will work with other Enterprise Architects in your business for a period in order to discover what makes your organization unique. They'll speak with your executives, staff, and business architects to learn about your company objectives, system requirements, consumer wants, and other essential aspects. They may then create a strategy and advice that is tailored to your company's goals and meets your stated cyber security risk appetite.

The Key attributes of the Security Architecture:



The **key phases** in the security architecture process are as follows:

- Architecture Risk Assessment: Evaluates the business influence of vital business assets, and the odds and effects of vulnerabilities and security threats.
- Security Architecture and Design: The design and architecture of security services, which facilitate business risk exposure objectives.
- Implementation: Security services and processes are implemented, operated and controlled. Assurance services are designed to ensure that the security policy and standards, security architecture decisions, and risk management are mirrored in the real runtime implementation.
- Operations and Monitoring: Day-to-day processes, such as threat and vulnerability management and threat management. Here, measures are taken to supervise and handle the operational state in addition to the depth and breadth of the systems security.



Common security architecture frameworks

1. **TOGAF:** The Open Group Architecture Framework, or TOGAF, helps determine what problems a business wants to solve with security architecture. It focuses on the preliminary phases of security architecture, an organization's scope and goal, setting out the

problems a business intends to solve with this process. However, it does not give specific guidance on how to address security issues.

2. **SABSA:** Sherwood Applied Business Security Architecture, or SABSA, is a quite policy driven framework that helps define key questions that must be answered by security architecture: who, what, when and why. Its aim is to ensure that security services are designed, delivered and supported as an integral part of the enterprise's IT management. However, while often described as a 'security architecture method', it does not go into specifics regarding technical implementation.
3. **OSA:** Open Security Architecture, or OSA, is a framework related to functionality and technical security controls. It offers a comprehensive overview of key security issues, principles, components and concepts underlying architectural decisions that are involved when designing effective security architectures. That said, it can typically only be used once the security architecture is already designed.

WHAT IS THE BENEFIT OF SECURITY ARCHITECTURE?

1. STRONG SECURITY ARCHITECTURE LEADS TO FEWER SECURITY BREACHES

Modern businesses need to have a robust security architecture framework for protecting their most important information assets. By strengthening your security architecture to close common weaknesses, you can drastically reduce the risk of an attacker succeeding in breaching your systems.

One of the top benefits of security architecture is its ability to translate each organization's unique requirements into executable strategies to develop a risk-free environment up and down the business, aligned with business needs and the latest security standards.

As an added benefit, with these measures in place organizations can demonstrate their trustworthiness to potential partners, potentially helping them put their business ahead of competitors.

This will ultimately deliver an architecture that is of long-term benefit to the organization.

Security Design Reviews

The Design Review (DR) practice is focused on assessment of software design and architecture for security-related problems. This allows an organization to detect architecture-level issues early in software development and thereby avoid potentially large costs from refactoring later due to security concerns.

Beginning with lightweight activities to build understanding of the security-relevant details about an architecture, an organization evolves toward more formal inspection methods that verify completeness in provision of security mechanisms.

At the organization level, design review services are built and offered to stakeholders. In a sophisticated form, provision of this practice involves detailed, data-level inspection of designs, and enforcement of baseline expectations for conducting design assessments and reviewing findings before releases are accepted.

Security Design Reviews are a great way to identify threat scenarios that can result in the compromise of your application. Investing in Security Design Reviews early can save you a lot of money, time, and resources.

- Security design reviews have no dependency on the application being built or run in an environment.
- They can also be applied early in the SDLC and provide significant cost savings due to avoidance of costly fixes later on in the application life-cycle.

Purpose of a design review

A design review can be triggered by a range of requirements but it's common as part of the service planning and design phases to consider reviewing services and solutions to ensure:

- They are fit for purpose
- That the risks are known and understood
- That the solution meets the business needs
- That it has suitable controls regarding risks

Design Review Considerations

A major consideration here is the scope of activity and exercise. In addition to this it is wide to review things with a view on:

- Risk Appetite

- Risk Tolerances
- Business Outcomes
- Security Outcomes
- Constraints

A custom web application holding sensitive data may warrant a greater level of review than a simple component or COTS application. It should be risk aligned.

Review Perspectives

We should consider a range of views and perspectives which include:

- Customer
- Supply Chain
- Operations
- Financials
- Legal and Regulatory
- Corporate Policy
- Procurement and Support
- Value
- Risk
- Controls
- Supply Chain

Supplier Considerations

- Supply Chain Security
- Accreditations
    - ISO27001:2013
    - Cyber E
    - IASME
    - SOC2
    - ISO9001
- Risks
- Location
- Countries of Operation

Product/Service Considerations

- Requirements

- Governance
    - Policies
- Compliance & Assurance
    - Common Criteria/EAL
    - Penetration Testing
    - Code Review
    - Product Assurance Certifications
- Operations
    - Security Monitoring
    - Reporting
    - Operator Requirements
        - Skills
        - Certifications
        - Training
        - Documentation
    - Configuration
        - Vendor Hardening Guidance
    - Vulnerabilities/Vulnerability History
    - Digital Forensics/Chain of Custody Considerations
- Integration
- Features
    - Product/Service Specific Features
- Financials/Costs
- Interfaces
    - User
    - Administrator
- Authentication
    - IAM, PAM, RBAC
- Authorization
- Data Protection
    - Data Location
    - Access to services/data
    - Encryption at Rest
    - Encryption in Transit
- Firmware

- Frequency of Firmware Updates (historic)
- Patching and Updates
- Physical Security
- Audit Logs
- Reporting
- Alerting
- Product Roadmap
- Product Lifecycle and Supportability
- Warranty
- Performance
- Availability/HA Options
- Backup and Recoverability
- Supply Chain
- Secure Wipe/Destruction

There's quite a bit to think about. Whilst you can silo out some of the areas and say some are IT and some are Security that to me is an odd approach. It would be better from my point of view to just consider leveraging different skills, tools, and techniques. If we look at security in isolation without the context, we will almost certainly not be able to understand the solution and therefore we won't be able to understand the risks etc.

Advantages and Disadvantages of Security Design Reviews

Advantages	How you can benefit from these advantages
Requires no supporting technology	The absence of supporting technology can be an advantage because it enables our staff to rely on their skill and experience to manually review the code to ensure there are no security issues.
Early in the SDLC can help avoid costly fixes later on	Identifying and addressing issues early in the Software Development Life Cycle (SDLC) can help avoid costly and time-consuming fixes later on in the pro
Encourages a security mindset	Having regular security audits and assessments in place encourages a security mindset among team members, which is an advantage in terms of identifying and mitigating potential security threats and vulnerabilities.

While Security Design Reviews come with some disadvantages, we can actually address each of them in a unique way

Disadvantages	How we address these disadvantages
Can be time-consuming	To accelerate your time to value, you can leverage our experience and expertise.
Supporting material not always	Our process involves a series of workshops to capture all the required

available	information from your team where previously undocumented material is not available.
Encourages a security mindset	Having regular security audits and assessments in place encourages a security mindset among team members, which is an advantage in terms of identifying and mitigating potential security threats and vulnerabilities.

By

Dr M V Kamal

UNIT-III

Prepared By
Dr M V Kamal

WHAT IS AN IT SECURITY RISK ASSESSMENT?

A security risk assessment is the process of assessing an organization's IT infrastructure to identify potential security risks and vulnerabilities. The goal is to improve the organization's security posture and increase its overall level of protection.

Some of the vulnerabilities that an SRA might find are:

- **Weak passwords**: Weak passwords can be easily guessed by hackers, resulting in credential theft.
- **Unpatched systems**: Outdated software can increase the risk of malicious exploits and cyber-attacks.
- **Weak user access controls**: Unrestricted user access can allow employees or potential threat actors to access data they would otherwise not have access to.
- **Insufficiently secure networks**: Weak encryption or un-encrypted Wi-Fi networks can be vulnerable to attacks.
- **Insecure data storage**: Storing or transmitting data without proper encryption can lead to data breaches which may result in data theft or loss.

WHAT PROBLEMS DOES A SECURITY RISK ASSESSMENT SOLVE?

Security risk assessments can help organizations identify and address potential threats before they become a problem. By performing a security risk assessment, organizations can identify weak points in their security posture and create a plan of action to address those weaknesses.

Some of these problems and threats may be things like:

- **Data breaches**: An SRA can help to identify weak points in an organization's IT infrastructure that could lead to data breaches.
- **Reputational damage**: A security breach can lead to significant reputational damage and could even lead to legal action.
- **Lost productivity**: Security incidents can lead to downtime which can result in loss of productivity.
- **Regulatory fines**: Regulatory bodies may impose fines or other punishments on organizations that do not take sufficient measures to protect their customer's data.

WHO SHOULD PERFORM THE IT SECURITY RISK ASSESSMENT?

The security risk assessment process should be carried out by a qualified security professional who is experienced in assessing security risks in IT environments. This person should have a thorough understanding of IT security and be familiar with the

organization's IT infrastructure. They should be able to identify potential security vulnerabilities and recommend ways to mitigate them.

Some of the recommended solutions might be :

- **Strengthen password security**: Installing strong password policies, using multi-factor authentication, and requiring periodic password changes can help to protect against credential theft.
- **Patch systems**: Installing system patches regularly can help to reduce the risk of malicious attacks.
- **Restrict user access**: Implementing role-based access control systems can help to ensure that only authorized users can access sensitive data.
- **Encrypt data**: Encrypting data both in transit and at rest can help to prevent data breaches.

HOW IS AN IT RISK ASSESSMENT DONE?

A security risk assessment involves looking at the system architecture, network security, user access controls, and all other security measures. Based on the results, the security professional should provide recommendations for improving the organization's security posture.

Some other ways to perform security risk assessments are:

- **Conduct interviews**: Interviewing key personnel can help to identify potential threats and vulnerabilities that are not obvious from a technical standpoint.
- **Perform penetration testing**: Penetration testing can help to identify potential vulnerabilities that could be exploited by malicious actors.
- **Run periodic scans**: Periodic vulnerability and malware scans can help to detect emerging threats and vulnerabilities.
- **Conduct security audits**: A security audit can help to identify any weaknesses in an organization's security policies and procedures.

By performing an IT security risk assessment regularly, organizations can reduce the risk of a security breach or incident while also further protecting their customer data. It is an important part of any organization's security strategy and can help to ensure that its IT infrastructure is secure and resilient against potential threats.

WHAT INDUSTRIES REQUIRE A SECURITY RISK ASSESSMENT FOR COMPLIANCE?

Many industries require organizations to perform security risk assessments as part of their compliance requirements. Organizations in these industries must perform security risk assessments regularly to maintain their compliance and ensure that their IT environments are secure.

A few of these industries include:

- **Healthcare**
- **Finance**

- **Government**
- **Retail**
- **Education**

By performing a security risk assessment, organizations in these industries can identify potential security vulnerabilities and reduce their overall risk of a security breach or incident. This helps them to stay compliant with industry regulations and protect their customers' data.

HOW LONG DOES IT TAKE?

The length of time required to complete a security risk assessment can vary depending on the size and complexity of the organization's IT environment. Generally, the process can take anywhere from a few days to several weeks, depending on the complexity of the organization's IT environment and the number of consultants involved in the assessment.

The factors that may change this timeframe include:

- **Size of the organization**: Organizations with larger IT environments may require a longer assessment time.
- **Number of consultants**: The number of consultants involved in the assessment can also impact the length of the assessment.
- **Depth of the assessment**: The depth of the assessment can also impact the time required to complete the assessment.
- **Experience of the consultants**: If the consultants involved in the assessment have limited experience, it can take longer to complete the assessment.

Regardless of the time required to complete the assessment, the organizations must take the necessary time to ensure that the assessment is thorough and that all potential security risks are identified and addressed.

Cyber Security Framework

Cybersecurity framework is a powerful tool to organize and improve your cybersecurity program. It is a set of guidelines and best practices to help organizations build and improve their cybersecurity posture. The framework puts forth a set of recommendations and standards that enable organizations to be better prepared in identifying and detecting cyber-attacks, and also provides guidelines on how to respond, prevent, and recover from cyber incidents.

This framework addresses the lack of standards when it comes to cybersecurity and provides a uniform set of rules, guidelines, and standards for organizations to use across industries.

The framework categorizes all cybersecurity capabilities, projects, processes, daily activities into these 5 core functions:

Capability	Description
Identify	What processes and assets need protection?
Protect	Implement appropriate safeguards to ensure protection of the enterprise's assets
Detect	Implement appropriate mechanisms to identify the occurrence of cybersecurity incidents
Respond	Develop techniques to contain the impacts of cybersecurity events
Recover	Implement the appropriate processes to restore capabilities and services impaired due to cybersecurity events

5 Core Functions of Cybersecurity Framework

IDENTIFY

The Identify function is focused on laying the groundwork for an effective cybersecurity program. This function assists in developing an organizational understanding to manage cybersecurity risk to systems, people, assets, data, and capabilities. To enable an organization to focus and prioritize its efforts, consistent with its risk management strategy and business needs, this function stressed the importance of understanding the business context, the resources that support critical functions, and the related cybersecurity risks. Essential activities in this group include:

- Identifying physical and software assets to establish the basis of an asset management program
- Identifying the organization's business environment including its role in the supply chain
- Identifying established cybersecurity policies to define the governance program as well as identifying legal and regulatory requirements regarding the cybersecurity capabilities of the organization
- Identifying asset vulnerabilities, threats to internal and external organizational resources, and risk response activities to assess risk
- Establishing a risk management strategy including identifying risk tolerance
- Identifying a supply chain risk management strategy including priorities, constraints, risk tolerances, and assumptions used to support risk decisions associated with managing supply chain risks

PROTECT

The Protect function outlines appropriate safeguards to ensure delivery of critical infrastructure services and supports the ability to limit or contain the impact of a potential cybersecurity event. Critical activities in this group include:

- Implementing protections for Identity Management and Access Control within the organization including physical and remote access
- Empowering staff through security awareness training including role based and privileged user training
- Establishing data security protection consistent with the organization's risk strategy to protect the confidentiality, integrity, and availability of information
- Implementing processes and procedures to maintain and manage the protections of information systems and assets
- Protecting organizational resources through maintenance, including remote maintenance activities
- Managing technology to ensure the security and resilience of systems, consistent with organizational policies, procedures, and agreements

DETECT

Detecting potential cybersecurity incidents is critical and this function defines the appropriate activities to identify the occurrence of a cybersecurity event in a timely manner. Activities in this function include:

- Ensuring anomalies and events are detected, and their potential impact is understood
- Implementing continuous monitoring capabilities to monitor cybersecurity events and verify the effectiveness of protective measures including network and physical activities

RESPOND

The Respond function focuses on appropriate activities to take action in case of a detected cybersecurity incident and supports the ability to contain the impact of a potential cybersecurity incident. The essential activities for this function include:

- Ensuring response planning process are executed during and after an incident
- Managing communications with internal and external stakeholders during and after an event
- Analyzing the incident to ensure effective response and supporting recovery activities including forensic analysis and determining the impact of incidents
- Performing mitigation activities to prevent expansion of an event and to resolve the incident
- Implementing improvements by incorporating lessons learned from current and previous detection / response activities

RECOVER

The Recover function identifies appropriate activities to renew and maintain plans for resilience and to restore any capabilities or services that were impaired due to a cybersecurity incident. Timely recovery to normal operations is impressed upon, to reduce the impact from a cybersecurity incident. Essential activities for this function somewhat overlap with those of Respond and include:

- Ensuring the organization implements recovery planning processes and procedures to restore systems and/or assets affected by cybersecurity incidents
- Implementing improvements based on lessons learned and reviews of existing strategies
- Internal and external communications are coordinated during and following the recovery from a cybersecurity incident

Why should use Cybersecurity Framework?

The framework can help you with the following challenges…

1. You worry about unseen risks and vulnerabilities.
2. You do not have an accurate inventory of assets that need to be protected.
3. Your team spends much effort chasing items that will not have impact, while you would like them to focus on real risk
4. You want to know how to address risk items given your current tools and what's available in the marketplace
5. Your colleagues outside the security team do not understand cyber risk and therefore fail to "own" critical mitigation tasks
6. Your board is beginning to ask you about quantifying the risk reduction outcomes from the strategic cybersecurity plan that your team has been executing.

Common Software Security Flaws

There’s no telling which software security issues pose the most threat to your business. That’s why it’s essential to address all software security design flaws within your development cycle so that software security flaws are not included within your next deployment. Here are the most common software security issues and solutions to ward off cybercriminals from malicious attacks.

- **Broken Authentication and Session Management**

  Despite best intentions, passwords and user authentication are often security risks for software that hackers commonly seek to exploit. Because when there are errors with the functionality of authentication - that is, the process of users confirming that they are who they say they are when connecting with a software application - it doesn't take long for an attacker to get inside your software system.

  A cybercriminal can take advantage of broken authentication to compromise user passwords and session tokens when there is:

  - No implementation of Strong Password mechanism
  - No effective Password Policy for your application
  - No definition of session timeout duration within your application
  - No reset of default generated credentials (such as passwords) upon login

  Prevent malicious attacks by revisiting your Password Policy and refining all security policies that best protect user accounts, including defining session timeout duration within your software app. Implement a Strong Password mechanism so that your users sign in with a complex passcode and trigger a reset of default generated credentials when a user successfully logs in.

- **URL Manipulation**

  A hacker manipulates a URL simply by changing parts of the URL for a web-based application to test if they can gain access. A trial-and-error approach in manipulating URL values can reveal easy access to user accounts and invoices for gathering sensitive (and valuable) data, such as credit card information and bank routing numbers. Even worse, many hackers have tools that automate this process for finding vulnerabilities within your URLs.

  URL manipulation can pose a threat to your system if your application:

  - Features important ID and keys within any URL, including session tokens, cookies, hidden fields and session IDs.
  - Allows access to other user data by tampering with URLs

- Prevent malicious attacks by restructuring how your URLs pull information from your servers and databases. Ensure your web-based application is patched with the latest security updates, including encryptions and latest threat definitions. Confirm URLs cannot be manipulated for unauthorized user access during QA testing.

- **Broken User Access Control**

  Without proper account configurations or missing account restrictions, any user can access sensitive data for accounts not associated with their log-in criteria. Most users are concerned with only their user data and will not notice this broken user access control. Unfortunately, cybercriminals are trained to spot these software security flaws within your system and even modify access rights or user data to suit their needs.

  Broken user access control can pose a threat to your application if your system:

  - Enables authorized access to account data for users who should not have authorization
  - Exposes any user permission or access control to unauthorized user accounts

  Prevent malicious attacks by restricting authorization for user permissions and access control to admin accounts only. Require user verification when requesting access to sensitive account data within the application, even when signed in.

- Sensitive Data Exposure

  Health information, financial data, passwords and usernames all qualify as sensitive data for an application to safeguard. However, this information is appealing to cybercriminals who want to commit fraud and steal people's identities. So whenever sensitive data is not properly protected within a software system, attackers are the first to find ways to retrieve this information from your software app.

  Sensitive data exposure can pose a threat to your network if your application:

  - Does not mask sensitive information, including passwords, credit card details and payment activities
  - Does not prevent unauthorized users from accessing sensitive data, including personal information, medical records and account history

  Prevent malicious attacks by applying extra protection for sensitive data through encryption, whether users are at rest or in transit. Trigger user verification whenever requesting access to sensitive account data within the software system, even when logged in.

- **Cross-Site Scripting**

  In a cross-site scripting attack - commonly referred as XSS - a hacker executes malicious scripts on legitimate, trusted websites within a web-based software application. These scripts allow attackers to bypass access controls in order to harm users within the app, be it to conduct phishing schemes or to steal their identities. For example, a user may submit personal information within a contact form request, only for that data to be sent directly to the cybercriminal.

  Cross-site scripting can pose a threat to your application if your system:

  - Supports untrusted data on a webpage without proper validation
  - Applies a browser API to create HTML or JavaScript on any webpage updated with user-supplied data

  .