# Security Engineering Integration into System Engineering

12/1/18

Mahasa Zahirnia

# Agenda

- Purpose
- Abstract
- What is The Problem
- Types of Security
- Stages of Requirement and Integration of Security
  - Early Development
  - Late Development
  - Security Architecture
  - Detail Design
- Cyber Attack Actors
- Security CONOPS
  - Typical Network Attack and Counter Attack
- Security Network Diagram
- Kill Chain Prevention
- Security Modeling
- Deliverables
- Conclusion

# Purpose

- Define the integration of Cyber Security into the various stage of requirement and design
- Provide a simple example and various artifacts that show the integration of cyber requirements
- Define sample attack onto the system and environment

# Abstract


- Security is crucial issue for information systems.  Traditionally, security is considered after system definition, however this approach often leads to issues which translates into vulnerabilities.
- Through integration of the security and system engineering, we can eliminate the security vulnerabilities
- How can we effectivity integrate security into the requirement process such that the developers can drive the required software?
- How can we verify that our assumptions are valid?

# What is The Problem

- System Engineering considers security as a non-functional requirement. Non-functional requirements introduce quality characteristics, but they also represents constraints under which the system must operate
- System Engineering recognizes non-functional requirements such as reliability and performance into software requirements however security is introduced after the system requirements definition.
- The problem is in four areas:
  - Security is difficult to analyze and model
  - Security requirement are often written in separate section so it is hard to coordinate between functional and non-functional requirements
  - Developers lack expertise and are often interested in building fun capabilities
  - Simple attack scenarios are not developed at early stages of the program

# Types of Security

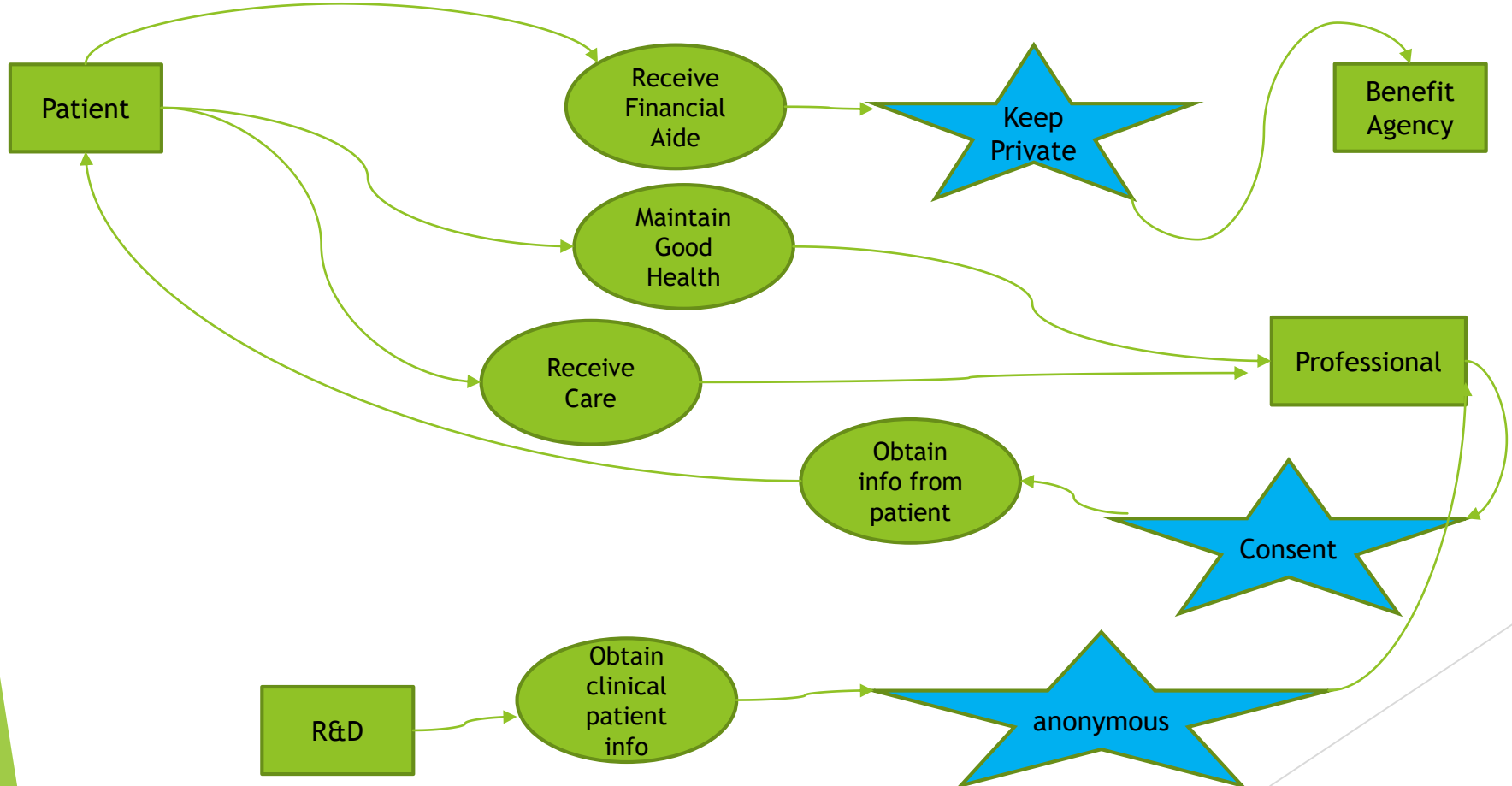
- There are three different types of Security:
  - Security constrains are defined as constrains that is related to security of the system such as access privileges
  - Secure entitles are resources /tasks of the system that needs to be secured or act in a secure way
  - Secure goal is what needs to be met to achieve security constrains defined as: Confidentiality, Integrity, and Availability

# Stages of Requirement and the Integration of Security

- Early Requirements: Understanding of the requirement by studying the existing settings and products.  The output is an organizational model with actor and imposed security constrains
- Late Requirements: Designing the system within it's operational mode, deriving its functionally and security requirements.  A small set of actors define dependencies and security constrains
- Architectural Design: Define system of system and its sub-components, data flow, interfaces and all of their security dependencies
- Detail design: future defines the input/output, controls and security dependencies

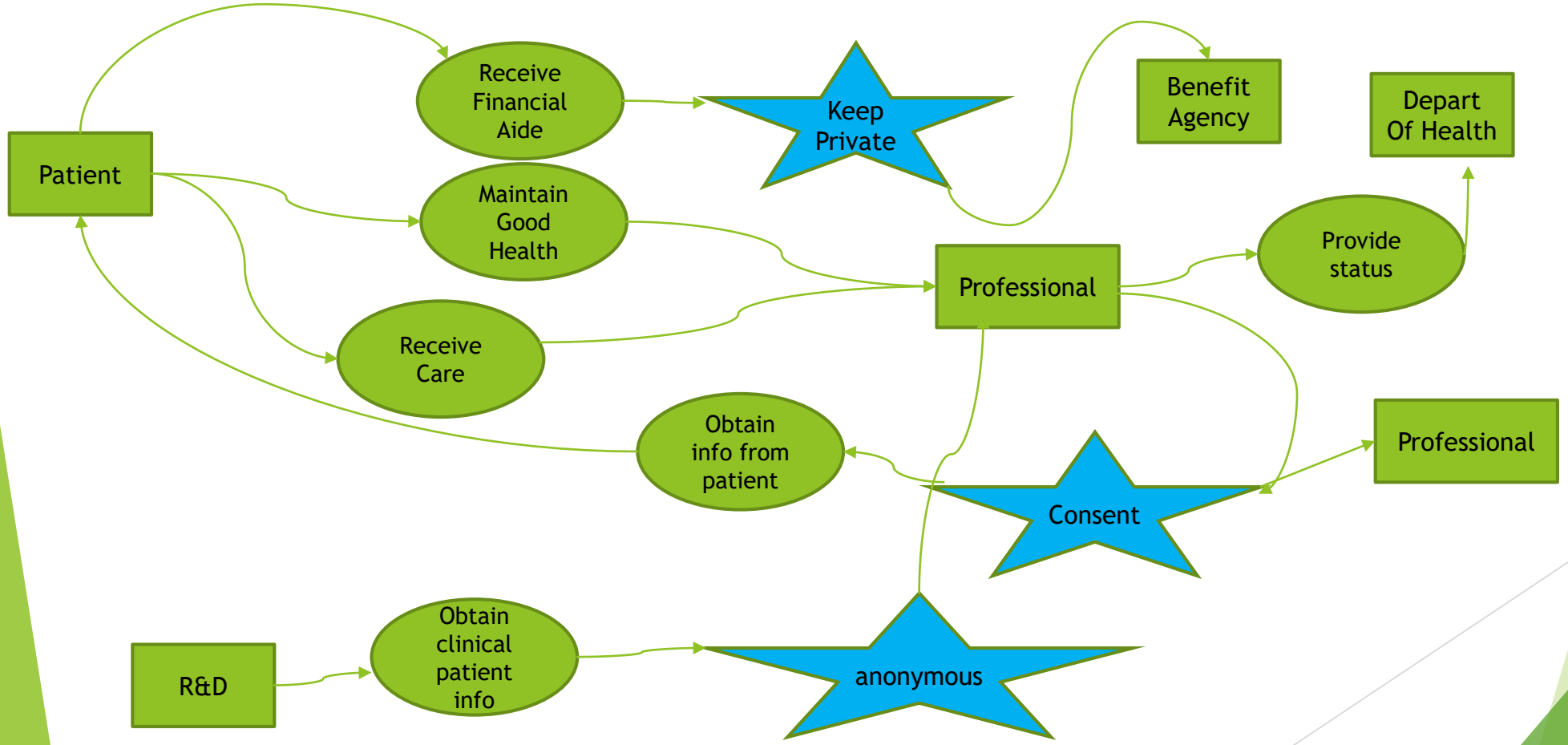
# Early Development

- During Early Development the actors are modeled in Use Cases to explain their interdependencies and the security constrains (stars in the diagram) that must be met for the actors accomplish their goals



# Late Requirement

- We define the system to be by adding/deleting more functionality and security onto the existing environment

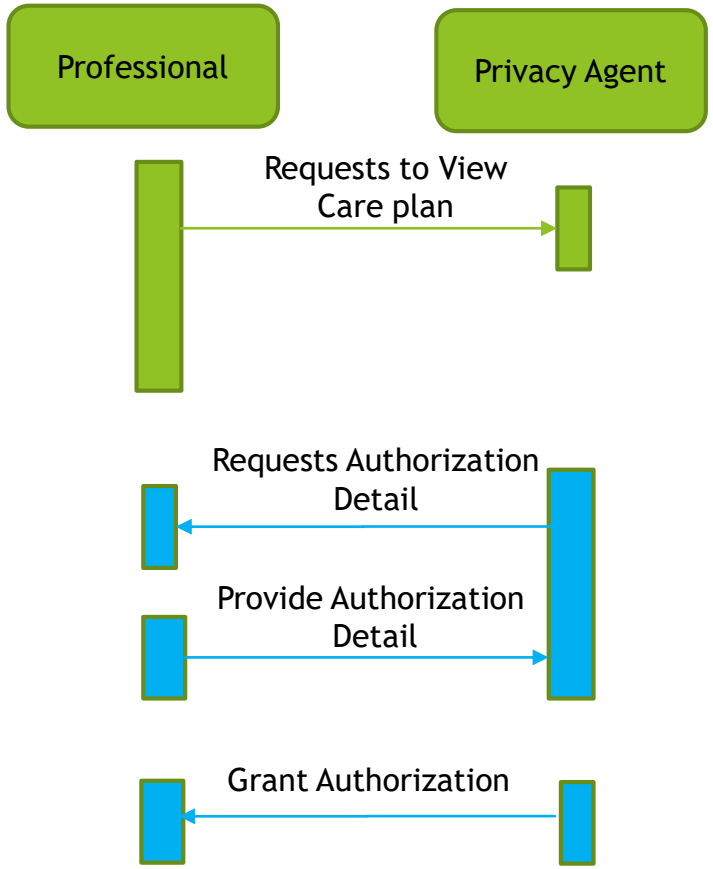


# Security Architecture

A multi phase approach of embedding security requirements into the functional architecture

# Detail Design

- Detail design is derived from specifications.  During this stage the developers define detail interaction and take into account security aspects. A Unified Modeling Language (UML) diagram can be developed to future explain the design



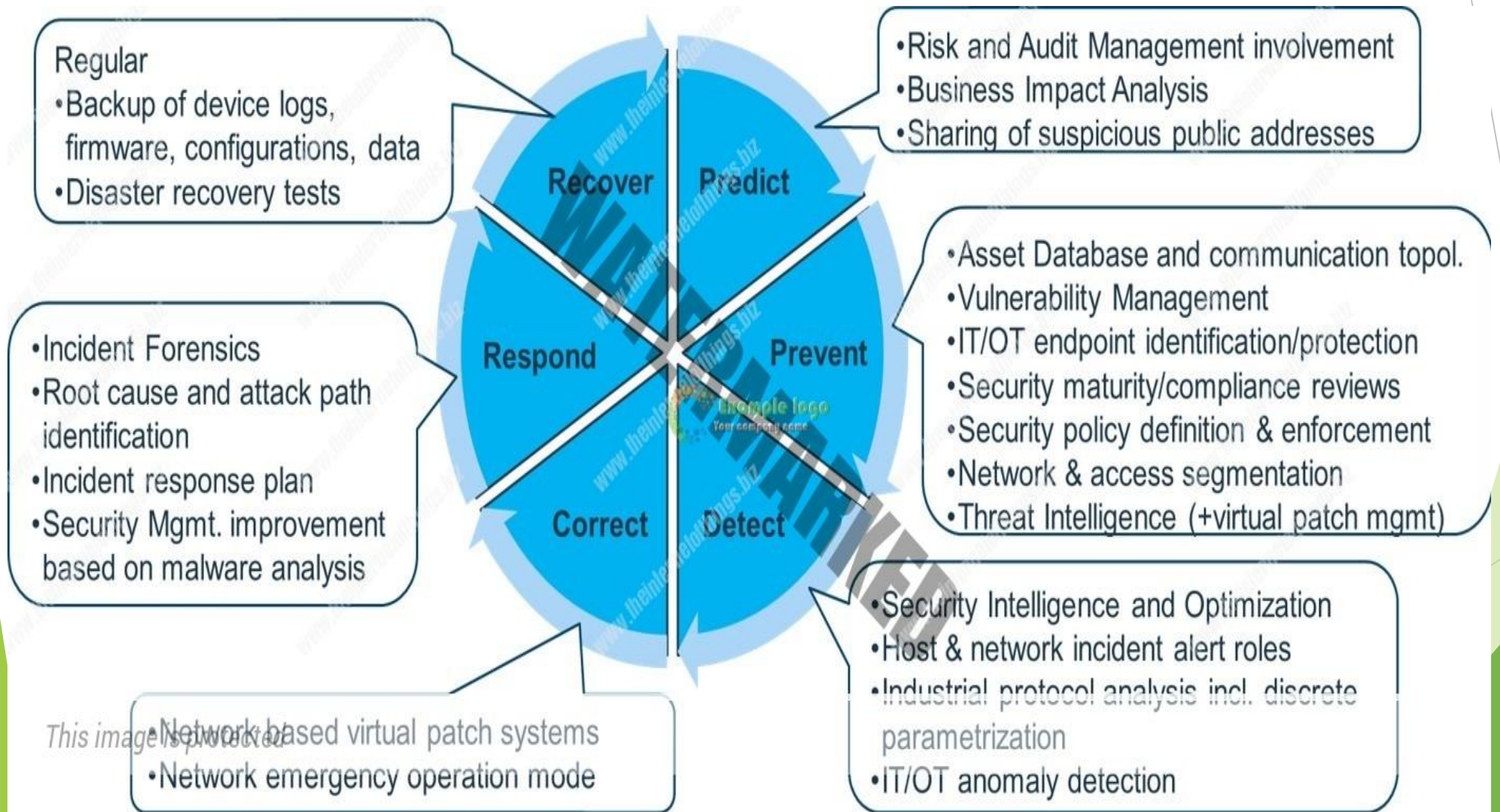

Privacy Agent is looking for multiple set of criteria's before allowing authorization and not just user name and password (build a better mouse trap) ex: retina scan

# Cyber Attack Actors

- Identify adversaries as one the main actors of the environment and not just the end users
- Identify intent and usage of the environment of adversary actors.  Typical adversary actors are:
  - Small groups
  - Political Groups
  - Criminal Organization
  - Terrorist
- Build Security CONOPS, Attack Scenario and additional artifacts such as kill chain prevention to encompass attacks and counter attacks into product delivery

# Security CONOPS Layout

- Define specific activity that would include below parameters that are not generic diagram/sentences:



# Typical Network Attack Scenario

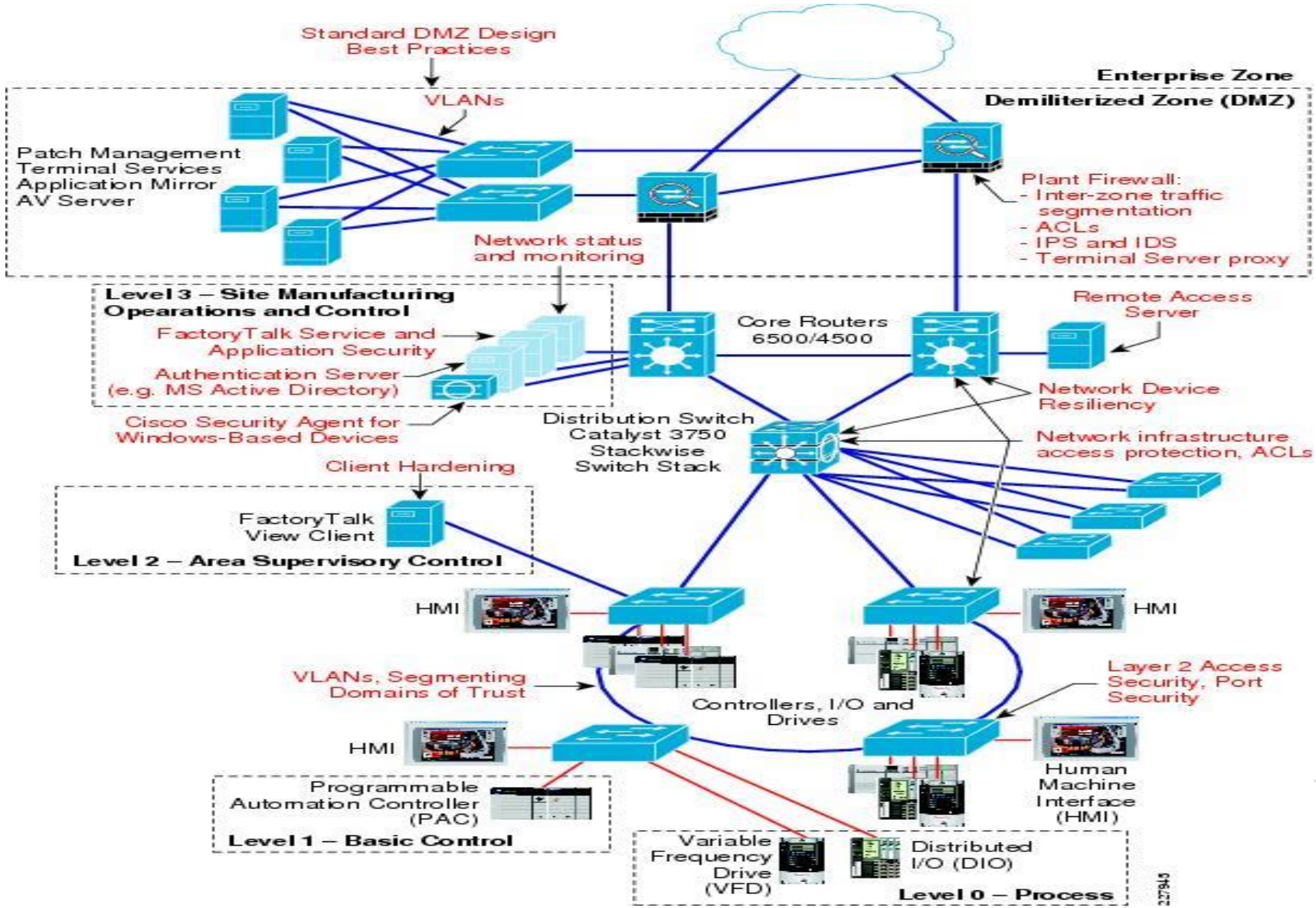
These attack are quite similar and usually show the following pattern:

- Hackers gain access to the facility information system using diverse methods: physical presence (e.g. USB drive), exploitation of vulnerable and expired software, theft of staff's mobile devices and even phishing or malicious emails.
- Once hackers have access to the IS, they use a special virus that holds the system hostage by encrypting the data it contains. Therefore, it becomes completely inaccessible and unusable until hackers are paid a ransom - usually in Bitcoin to make it untraceable - as the virus remains in the system and prevents anyone from using it.
- What makes hospitals such easy targets is their time sensitivity. Indeed, without quick access to patients' health record, their care may be delayed, which could result in serious consequences on their health - even death - and lawsuits for the hospital. Thus, facilities usually do not take any additional risks and they directly pay the ransom.

# Typical Network Counter Attack Scenario

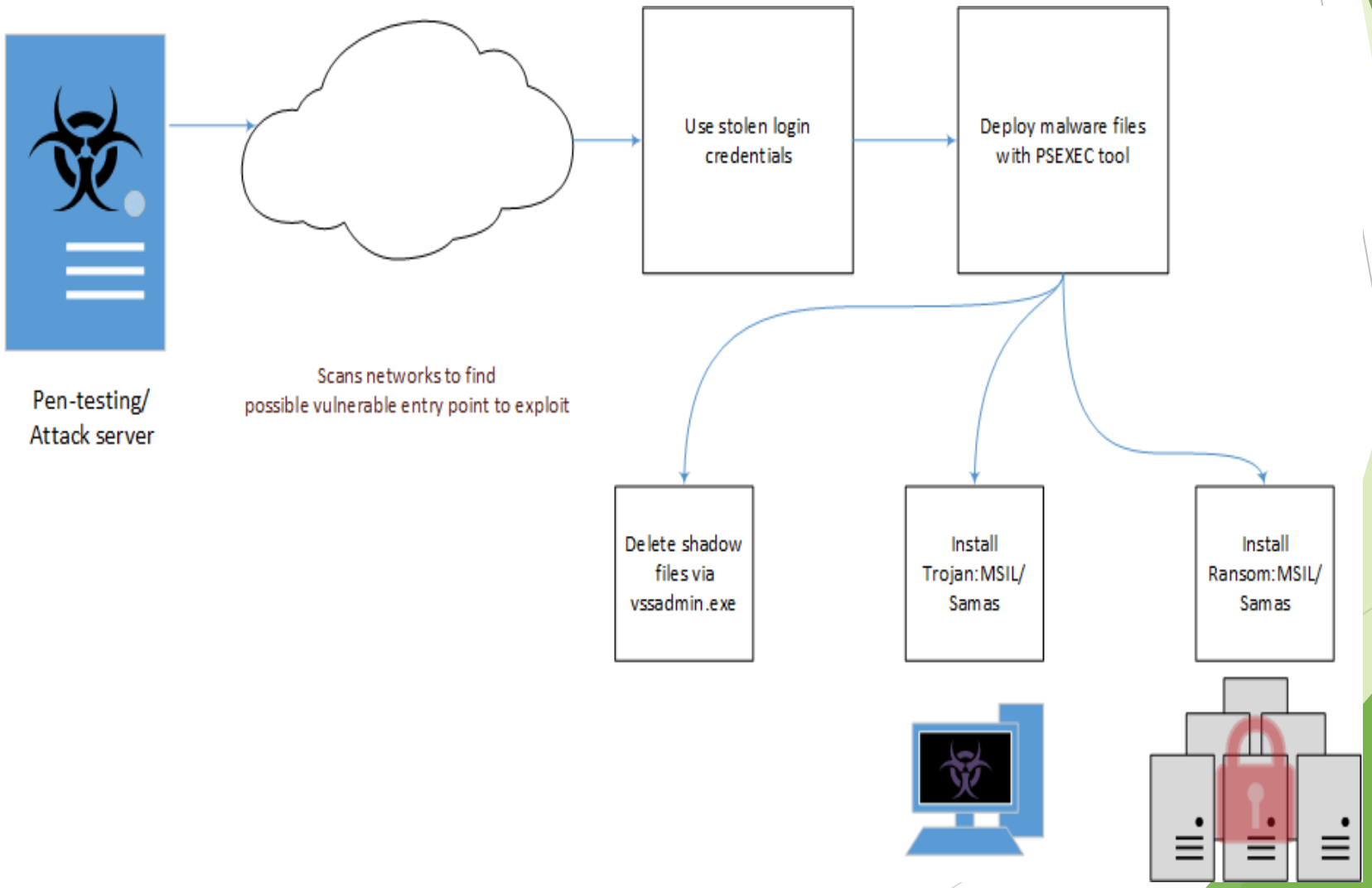
- Security Camera and badge access to all area
- Turn off all USB ports
- Shutdown the system when detecting a USB
- BitLocker on all hardware
- Dual factor authentication
- Encrypted VPN and remote access
- No external devices (laptops, Mobil,…)
- Training all staff
- Hire experts for defensive design and penetration testers for offensive attacks before and after the product has been installed
- Continuous monitoring of the environment for updates of design, network and attack enhancement

# Security Network

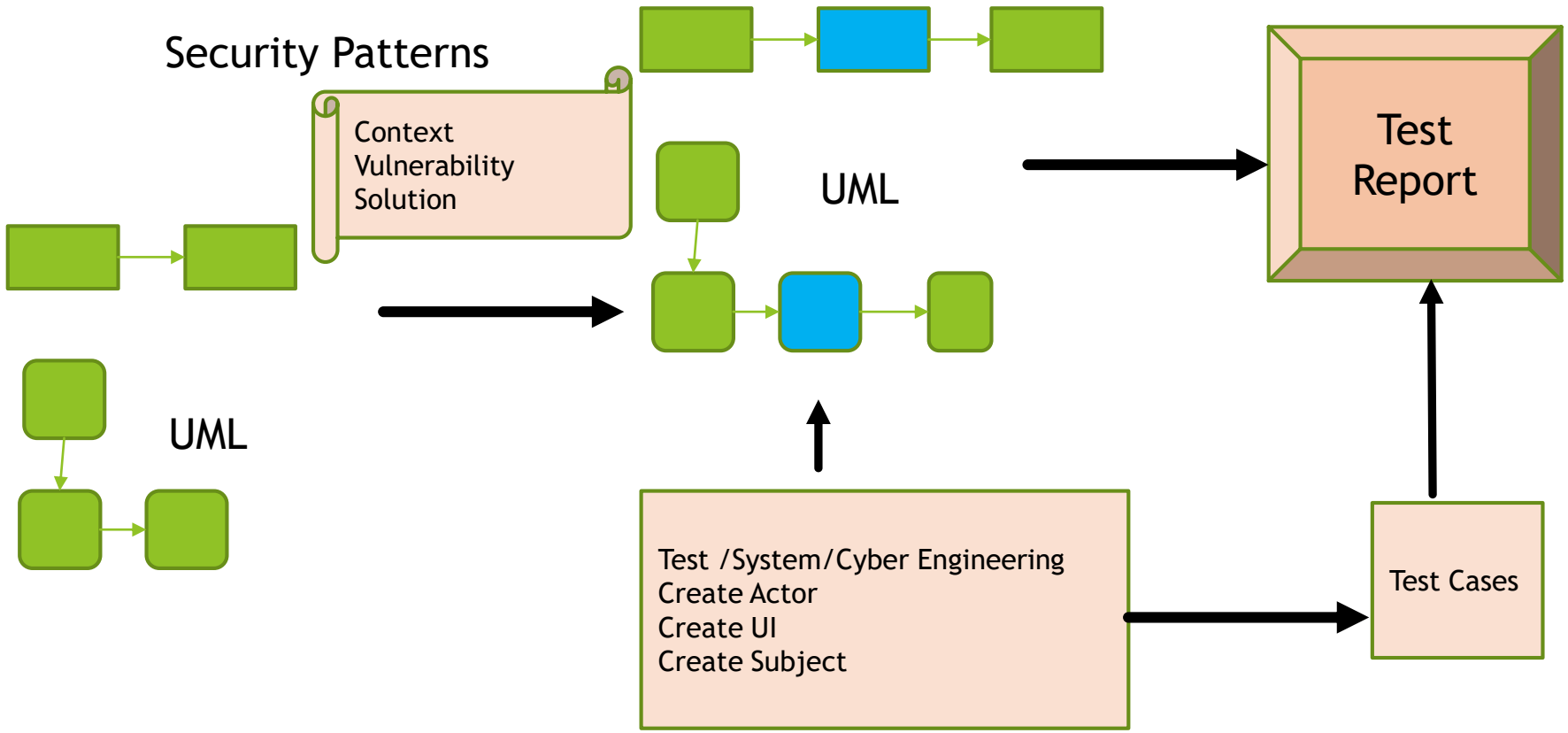


# Kill Chain

- Define the parameters and requirement (i.e. Splunk) that would prevent the Kill Chain such as:
  - Find
  - Fix
  - Track
  - Target
  - Engage
  - Assess



# Security Modeling



# Deliverables

- **Standard Documents**: Use existing Use Cases and UMLs and incorporate IA controls
- **Security Patterns**: Develop Uses Cases with adversity actors, estimate impact, define possible solution and estimate cost. With this diagram we are looking at bad actors intent and how they can achieve their goals. For example: In most cases once the bad actor invades the operating system he can SSH into any workstation on the network. Another example: If the Database is not password protected then he can access the database, or if the password is in a flat file then he can easily read the file and access the database
- **Security Network**: Create Network diagram that defines all security severs and their roles and responsibility
- **Product Security.** Product Security defines how the product should behave and manage the flow of data when security agent is in control. DISA Controls do not dictate product security, and only control commercial products such as the OS, Database,….At this level we looking at the product to protect itself after adversity has gained access. For example: At what level of data mining do we want to incorporate additional security? Or when can we tell if the operator is an adversary or standard user? How can the product be smarter at detecting patterns of corruptions?

Note: Security Patterns and Product Security are basically looking at the same condition but from different angles (aggressive and invasive vs. protection).

# Conclusion

- Security requirements and design must be an integrated part of the overall System of System approach. The stakeholders and developers must examine the varies phases of project execution and integrate cyber security requirement into the project for a holistic approach.  Attaching DoD DISA STIG requirements at the end of the documents is not a valid approach to incorporating Cyber into the project.  We must implement solutions at the earliest stage of the project

- Security is not just a part of our solution — it is embedded in all phases of the design to ensure that the data is protected and networks are available.

# Resources

- Department of Justice System Development Life Cycle Guidance, Chapter 6: www.usdoj.gov/jmd/irm/lifecycle/table.htm

- *Program Manager's Guide for Management Software*,0.6, 29 June 2001 Chapter 6: www.geia.org/sstc/G47/SW MgmtGuide%20Rev%/200.4.doc

- Requirements Generation System Joint Chief of Staff: www.jsc.mil/jsec3/EMCSLSA/stdlib/cd/added/3170_01.pdf