THREAT MODELING

SECURE APPLICATIONS

AGENDA

- Introduction
- Software Integrity - Secure SDLC
- Secure SDLC
- Key considerations
- Threat risk modeling
- Deliverables - Threat Model Information
- Security Controls
- Security Control Analysis
- Attack Vectors (Surface)
- Risk awareness
- Standard Methodology
- Conclusion

INTRODUCTION

- A case for achieving security in the Software Development Life Cycle (SDLC).
- Model all activities in the process.
- Identify threats to assets at the earliest indication.
- Reduces the squandering of effort, time, money and useless controls.
- Level an improvement training and appraisal program.
- Thinking to achieve a mature software process across the enterprise.

SOFTWARE INTEGRITY - SECURE SDLC

The use of a framework defines the process used by organizations to build applications from its inception to decommission.

SDLC models:

- Waterfall
- Iterative
- Agile

Process of planning and requirements, architecture and design, test planning, coding, testing and results, and release and maintenance.

SECURE SDLC

A Secure SDLC process ensures that security assurance activities such as penetration testing, code review, and architecture analysis are an integral part of the development effort.

The primary advantages of pursuing a Secure SDLC approach are:

- More secure software as security is a continuous concern
- Awareness of security considerations by stakeholders
- Early detection of flaws in the system
- Cost reduction as a result of early detection and resolution of issues
- Overall reduction of intrinsic business risks for the organization

KEY CONSIDERATIONS

- SDLC Processes (Functional & Non-Functional Requirements)
- System and Component Design & Diagrams
- System Security Authorization Agreement Core
- Information Technology Security Administration and Network Design
- Security Design Document
- Configuration Management Plan
- Patch Management
- Security Features User’s Guide
- System Rules of Behavior
- Incident Response Plan

THREAT RISK MODELING

The threat modeling process can be decomposed into 3 high level steps:

Step 1: Decompose the Application.

Step 2: Determine and rank threats.

Step 3: Determine countermeasures and mitigation.

DELIVERABLES - THREAT MODEL INFORMATION

- Assets
- External dependencies
- Entry points
- Trust levels
- Threat Categorization

STRIDE - A threat categorization such as STRIDE is useful in the identification of threats by classifying attacker goals (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege.)

Cybersecurity Kill Chain – (Reconnaissance, Weaponization, Delivery, Exploitation, Installation, Command & Control, Actions)

SECURITY CONTROLS

Once the basic threat agents and business impacts are understood, security goals are to identify the set of controls that could prevent negative impacts.

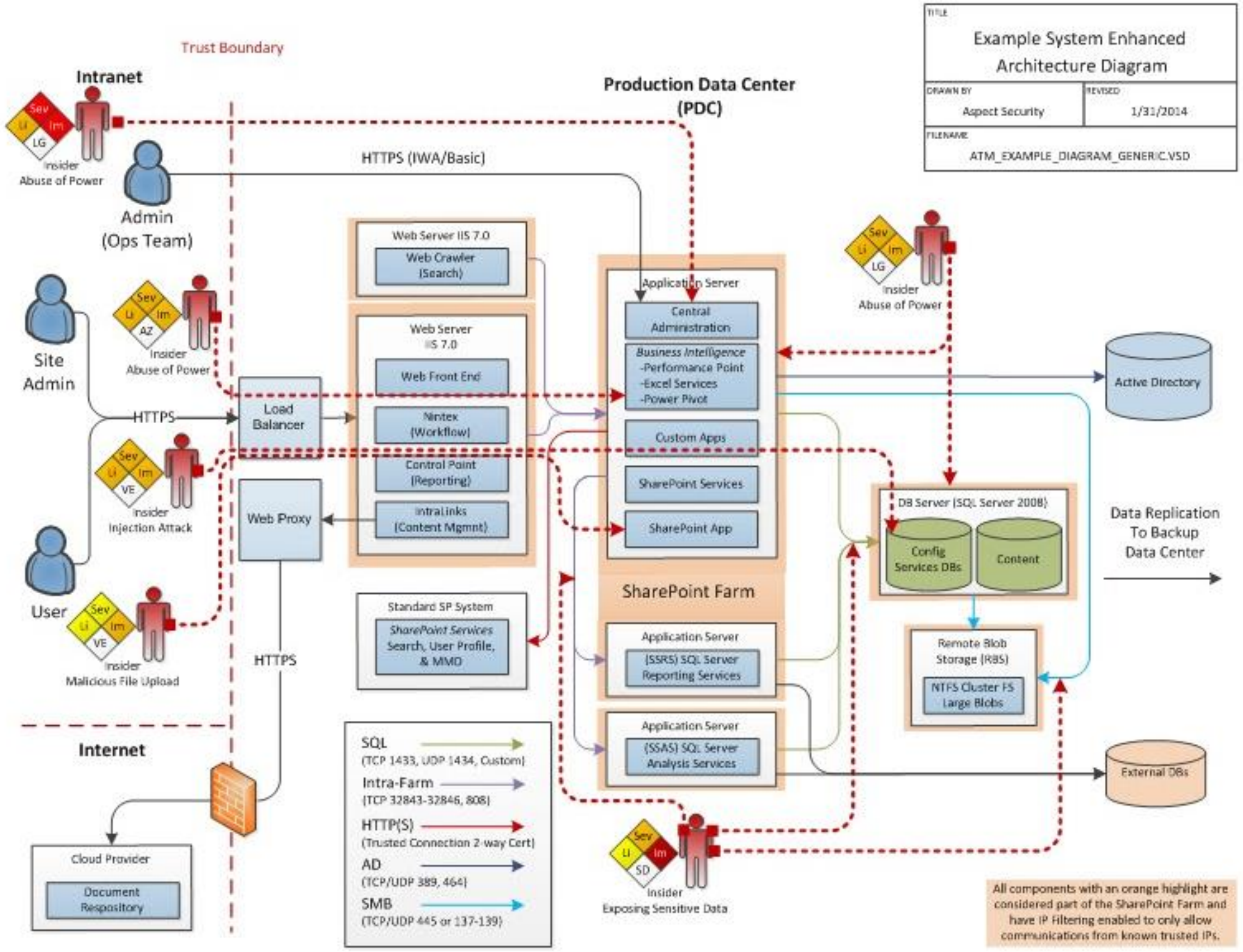
Specific Goals:

- Authentication, Authorization, and Accounting
- Cookie Management (Session Management)
- Data/Input Validation
- Error Handling/Information leakage
- Logging/Auditing
- Cryptography
- Secure Code Environment

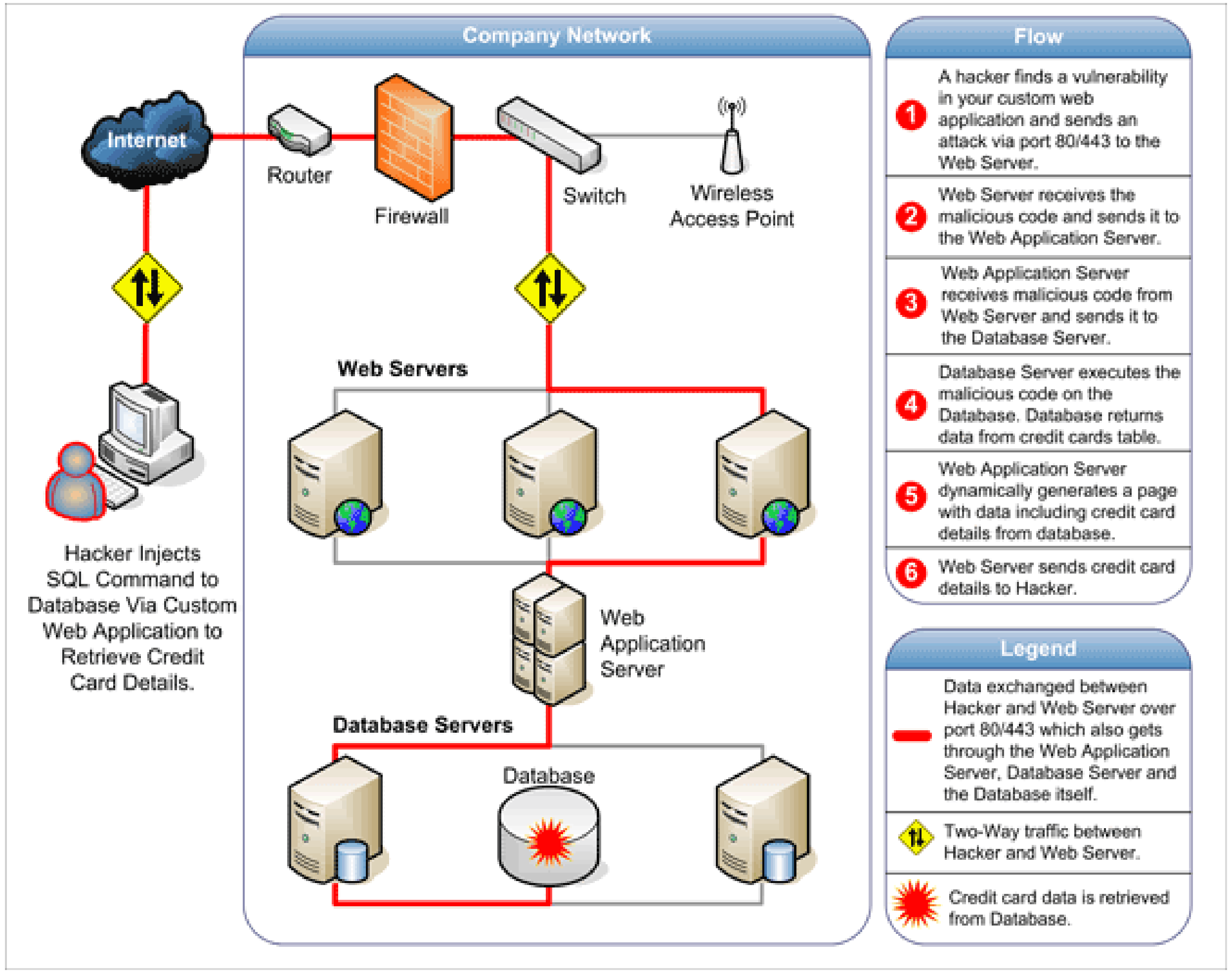
CRITICAL THINKING (MAN VS MACHINE)

1. In each application, should the protection mechanisms in a computer system focus on DATA, OPERATIONS, or USERS?
2. In which layer of the computer system should a security mechanism be placed?
3. Do you prefer simplicity – and higher assurance – to a feature rich security environment?
4. Should the task of defining and enforcing security be given to a central entity or left to individual components?
5. How can you prevent an attacker from getting access to a layer below the protection mechanisms?

SECURITY CONTROL ANALYSIS



ATTACK VECTORS (SURFACE)



Hacker Injects SQL Command to Database Via Custom Web Application to Retrieve Credit Card Details.

Flow

1. A hacker finds a vulnerability in your custom web application and sends an attack via port 80/443 to the Web Server.
2. Web Server receives the malicious code and sends it to the Web Application Server.
3. Web Application Server receives malicious code from Web Server and sends it to the Database Server.
4. Database Server executes the malicious code on the Database. Database returns data from credit cards table.
5. Web Application Server dynamically generates a page with data including credit card details from database.
6. Web Server sends credit card details to Hacker.

Legend

🟥	Data exchanged between Hacker and Web Server over port 80/443 which also gets through the Web Application Server, Database Server and the Database itself.
⇅	Two-Way traffic between Hacker and Web Server.
✸	Credit card data is retrieved from Database.

RISK AWARENESS

- Accidental Discovery
- Automated Malware
- The Curious Attacker
- Script Kiddies
- The Motivated Attacker
- Organized Crime

STANDARD METHODOLOGY

STRIDE

A classification scheme for characterizing known threats according to the kinds of exploit that are used (or motivation of the attacker).

DREAD

A classification scheme for quantifying, comparing and prioritizing the amount of risk presented by each evaluated threat.

RISK_DREAD = (DAMAGE + REPRODUCIBILITY + EXPLOITABILITY + AFFECTED USERS + DISCOVERABILITY) / 5

CONCLUSION

The adoption of a threat modeling process may not fit all organizations. It offers an abstraction to build on a set of principles that facilitate appropriate defense in depth approaches to security at the application level. If STRIDE and DREAD are unacceptable in some cases, it is recommended that the organization "dry run" several other threat risk models against an existing application or design.

NOTE: Read NIST Special Publication 800-154 for more guidance.

QUESTIONS