# Introduction to cryptography, parameter manipulation & exception management

# Unit objectives

After completing this unit, you should be able to:

- Get an overview of cryptography
- Gain insight on parameter manipulation
- Comprehend exception management

# Introduction (1 of 2)

## Introduction to cryptography, parameter manipulation, & exception management

- Cryptography requires the understanding of the following:
  - Basics of cryptography
  - Control objectives of cryptography
  - Requirements of cryptography
  - Insecure cryptographic storage
  - Environment affected for cryptography
  - Cryptographic vulnerability
  - Security verification
  - Manual approach

# Introduction (2 of 2)

## Introduction to cryptography, parameter  manipulation, & exception management

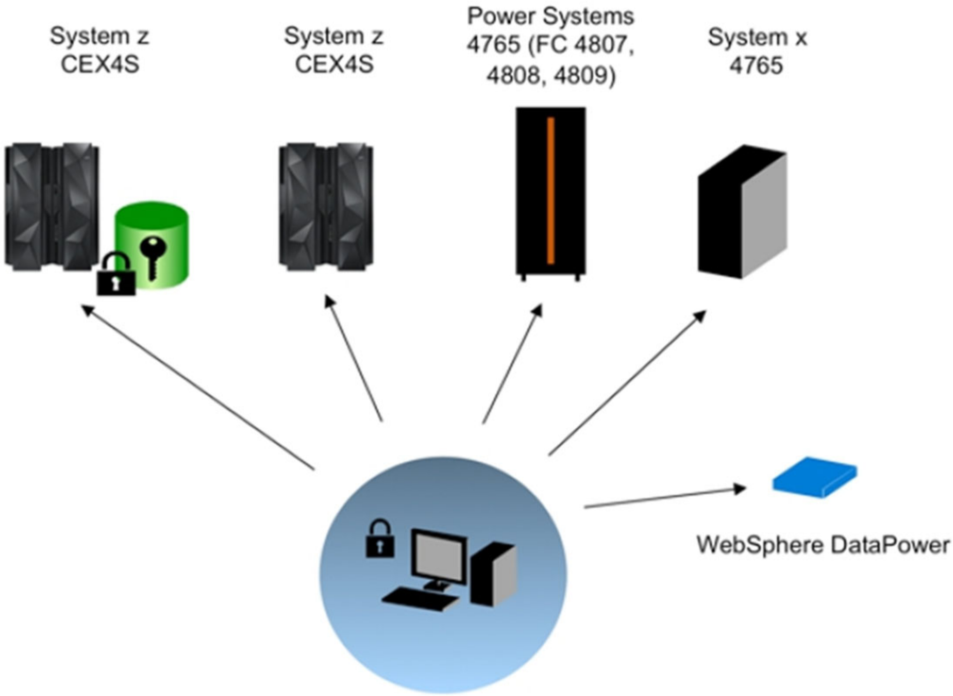
- Prevention from Insecure Cryptographic Storage can be achieved by adopting following techniques:
  - Plan to protect data from insider attack and external user Seeing the threats.
  - Encrypt all data in a manner that protects against these threats.
  - Encrypt all offsite backups but the keys are backed up and managed separately.
  - Use always strong keys and appropriate strong standard algorithms and key management is in place.
  - Hash the passwords with a strong standard algorithm.
  - Passwords and all keys are protected from unauthorized access.

# Poor Key Generation or Key Management (1 of 4)

- Countermeasures to address the threat of key management and poor key contain are as follows:
  - Always use built-in encryption procedures containing secure key management.
  - Store the key in a restricted place Use robust random key generation utilities
  - Use DPAPI to Encrypt the encryption for added security
  - Terminate keys on a regular basis
  - In reality cryptography is a challenging control to implement. Issues range from:
  - False sense of security
  - In-house developed and untested encryption routines
  - Use of untrusted and unsupported encryption routines
  - System performance
  - System/data recovery
  - Key management and recovery
  - Algorithm type/strengths
  - Key lengths
  - Key/random number generation

# Poor Key Generation or Key Management (2 of 4)

- Centralized Key Management

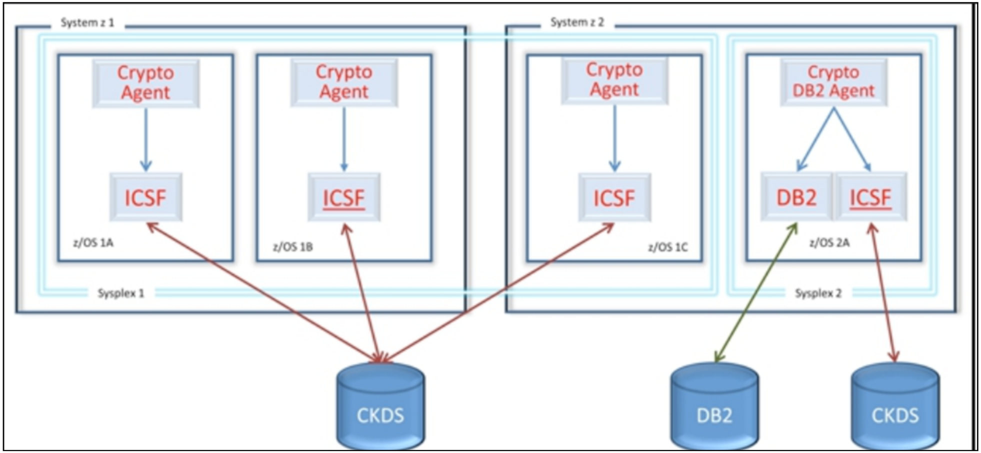


# Poor Key Generation or Key Management (3 of 4)

- Components of Centralized Key Management
  - The workstation
    - All key management operations that involve human interaction are carried out at the workstation
  - The key repository
    - Keys and their metadata are stored in an IBM database
  - The agent
    - The agents are installed on servers where either the key repository resides, or where there are keystores to be managed

# Poor Key Generation or Key Management (4 of 4)

- Utility of Centralized Key Management
  - Many banks today have separated the issuing and the authorization systems by deploying them on separate IBM system logical partitions (LPARs). This requires that the cryptographic keys for the payment cards are available on those systems.
- Key Management with two Parallel Sysplex systems
  - The figure shows a system configuration with two IBM System z® servers, each with two LPARs.
- Installation of the Centralized Key Management
  - In order to deploy a complete IBM Enterprise Key Management Foundation system, both a workstation and a server with the database are required.



# Weak or Custom Encryption

- Example Attack Scenarios
  - To prevent exposure to end users an application encrypts credit cards in a database
  - The system should have been designed to allow to decrypt only back end applications not the front end application.
  - A backup tape is made of encrypted health records, but the encryption key is on the same backup
  - To store everybody’s passwords, the password database uses unsalted hashes
  - Uncertain permits when generating secret key, allowing spoofing

# Basics of Parameter Manipulation

- Query String Manipulation
  - Protection from Query String Manipulation can be sought by
    - Use a session identifier to identify the client and store sensitive items in the session store on the server
    - Select HTTP POST instead of GET to submit forms
    - Encrypt query string parameters

- Form Field Manipulation
  - Protection from Query String Manipulation can be sought by following bellow practices
    - Never rely on client-side validation; always validate input from server-side
    - Avoid hidden fields, using single session token to refer server-side stored cache memory
    - Concatenate the name and value pairs together into a single string and append a secret key to the end of the string

# Cookie Manipulation

- Cookie manipulation refers to the modification made in cookies. Cookies are vulnerable to be manipulated easily by client. There are so many tools an attacker can use to make modification in cookies

- For protection from Cookie Manipulation
  - Do not trust user input for values that are already known to you
  - Use one session token to refer server-side stored cache memory
  - Do not store personal data or financial information in cookies
  - Do not store authentication details in cookies
  - Ensure session IDs are hashed if stored in cookies
  - Encrypt contents of cookies using approved crypto APIs
  - Label cookie as ‘Secure’ to prevent browsers sending it over non-secure connection

# HTTP Header Manipulation

- The client and server pass the data or info through the HTTP headers. The client makes request header and the receiver make the response header to that request. When it comes to make a decision, if your application trust headers then application is vulnerable to attack

- For determining from where the client has requested, do not trust the HTTP headers. HTTP header are used to pass control data from web clients to web servers for HTTP requests and vice versa for HTTP responses. The majority of applications ignore them.

- For availing protection from HTTP Header Manipulation
  - Do not rely on headers without additional security mechanisms

# Basics of Exception Management (1 of 2)

- Exceptions are allowed to propagate to the client can reveal internal implementation details. This details make no sense to the end user, but are very helpful to the attackers

- The basic countermeasure for exception handling is to disclose minimal information following an exception.
  - Information Disclosure
    - Revealing personally identifiable information (PII) such as passwords and credit card data, plus information about the application source and/or its host machines
  - Attacker Reveals Implementation Details
    - Attacker can use number of methods to obtain information that could provide a useful basis on which to perpetrate an attack on websites or the supporting infrastructures

# Basics of Exception Management (2 of 2)

- Protection from Information Disclosure can be sought by:
  - Remove comments (except Copyright comments!) from code before it is migrated to production services.
  - Ensure quality assurance processes include provision to verify that all comments are removed prior to migration to production.
  - Debug Commands: As with source code comments, it is often standard developer practice to include debug switches in HTML to allow them to switch on additional levels of logging or reporting. Permitting this code (and the server-side logic to interpret it) into production services

# Denial of Service (1 of 3)

- Flooding - sending many messages or simultaneous requests to overwhelm a server

- Lockout - sending a surge of requests to force a slow server response by consuming resources or causing the application to restart

- Common causes
  - Placing too many applications on a single server
  - Placing conflicting applications on the same server
  - Neglecting to conduct comprehensive unit testing

- Preventive measures include
  - Filter packets using a firewall
  - Use a load balancer to control the number of requests from a single source
  - Use asynchronous protocols to handle processing-intensive requests and error recovery

# Denial of Service (2 of 3)

- Following is an example of how a typical DoS attack can be carried out against a target:
  - Unsuspecting users are directed by a malicious hyperlink or a phishing email to a website where their systems can get infected by malwares and can be placed under the reason due to which malicious activities are taking place.
  - The machines which have been compromised now a wait for the instructions from the bot controller in order to attack a target. This is not known by the user. These bots remain idle for many days before they come to action and attack the system.
  - The machines which have been compromised now launched DDoS or DoS attacks at the command of the bot controller against the target. This is often carried out by using the UPD ports 53, 80, 443, 514, or the port number 80, 443 and 110 of TCP.
  - The systems which are targeted are often overrun with traffic and are forced offline.

# Denial of Service (3 of 3)

- Affected Environments from Denial of Service
  - The susceptibility of denial of service is present in nearly all identifiable web application, application server, and web server environment.

- Determination of Denial of Service Vulnerability
  - One important test is how many requests per second your application can hold
  - To test from a single IP address is useful as it will give you an idea of how many requests an attacker will have to produce in order to harm your site. To come to a decision about if any resources can be used to make come into existence denial of Service, you should observe each one to see if there is a way to exhaust it

- Protection from Denial of Service Attack
  - The time taken by the database to give response get delayed when server resources are overloaded by database DOS
  - Query rates, rates incurred for connection and various other rates for each user of the database are limited by the server resource. This prevention step is accomplished by Connection Controls.
  - Database servers can be crashed if attackers exploit platform vulnerabilities. Such attacks can be prevented by IPS and protocol validation.
  - Query access control is provided by Dynamic Profiling that can detect any queries beforehand and prevent DOS attacks

# Checkpoint

1. Cryptography is:
   - A. An Art
   - B. A Science
   - C. Both Art as well as Science
   - D. Neither Art nor Science

2. RSA in cryptography stands for:
   - A. Rivest Shamir Adelman
   - B. Real Scalable Algorithm
   - C. Royal Society of America
   - D. Ridley Scott Algorithm

3. Which of the following is considered cryptographic algorithm?
   - A. Symmetric key
   - B. Asymmetric key
   - C. Hash Function
   - D. All of the above

4. Symmetric key cryptographic algorithm uses the same key for encryption as well as decryption:
   - A. True
   - B. False

# Checkpoint

5. Asymmetric key cryptographic algorithm uses:

   A. Public key
   B. Private key
   C. Both Public as well as Private key
   D. Neither Public key nor Private key

6. Which of the following is not a parameter manipulation threat?

   A. Query String Manipulation
   B. Form Field Manipulation
   C. Cookie Manipulation
   D. Encryption Manipulation

7. DDoS stands for:

   A. Dispersed Denial of Service
   B. Distributed Denial of Service
   C. Diluted Denial of Security
   D. Directed Disk Operating System

# Checkpoint

8. Hash collisions are:

A. Failure of a given cryptographic hash function to complete successfully

B. Repetitions within a message digest that indicate weakness in the hash algorithm

C. Matching message digests found during the verification of a digital signature

D. Two different input messages that result in the same message digest value

9. A security architect working for a large financial institution has been asked to evaluate a variety of cryptographic algorithms that are being considered as candidates for a proprietary Electronic Funds Transfer (EFT) application.

A. Recommend a symmetric key cryptographic algorithm to provide confidentiality, integrity, and authenticity protections

B. Recommend that all transactions are digital signed before they are transmitted over an unsafe network

C. Recommend a hybrid solution that combines symmetric and asymmetric cryptography as well as hash functions

D. Recommend a hybrid solution that combines asymmetric cryptography and hash functions

10. Any action that prevents authorized users from executing programs is called

A. Malware

B. Spam

C. Denial of Service

D. Cross-site scripting

# Checkpoint Solutions

1. C
2. A
3. D
4. A
5. C
6. D
7. B
8. D
9. C
10. C

# Unit summary

After completing this unit, you should be able to:

- Get an overview of cryptography
- Gain insight on parameter manipulation
- Comprehend exception management