Best Practices Against Insider Threats

Objectives:

At the end of this episode, I will be able to:

Understand what the best practices against insider threats are.

Explain what the challenges in handling and responding to insider threats are.

Identify what the incident handling responsibilities for insider threats are.

Define the steps necessary to contain, eradicate, and recover from insider threats.

External Resources:

Handling and Responding to Insider Threats

What is an insider?

Any employee (trusted person or persons) having access to critical assets of an organization.

What is an insider attack?

Using privileged access to intentionally violate rules or cause threat to the organization's information or information systems in any form.

There are four types of insider threats:

```
▪ Malicious Insider
▪ Negligent Insider
▪ Professional Insider
▪ Compromised Insider
```

What are the Preparation steps for Handling Insider Threats?

```
    ▪ Security Awareness Training
    ▪ Policies prohibiting disclosing of confidential information
    ▪ Principle of least privilege
    ▪ Separation of Duties
    ▪ Background Checks
    ▪ Employee Monitoring
    ▪ Regular Auditing
```

How can I Detect and Analyze Insider Threats?

```
    ▪ Look for indicators of insider threat activity - UNUSUAL BEHAVIOR !!
    ▪ Mole Detection/Profiling
    ▪ Behavioral Analysis
    ▪ Physical (facility) security Analysis
    ▪ Log Analysis
    ▪ System Analysis - (Removeable Media & Web Browsers)
```

Removeable Media Usage:

```
    Windows systems store history of connected USB drives in the following
registry key: HKEY_LOCAL_MACHINE\SYSTEM \CurrentControlSet\Enum\USB

    Mac O/S: Click the Apple menu button and select About This Mac option;
Click the System Report In the System Information window, go to the Hardware
section on the left side and select the USB option

    Linux: Open the command console & run the usb-devices command to list all
the connected USB devices
```

Browser Data:

```
Mozilla Firefox stores cache, cookies, and history in the following system
```

locations:

Cache: C:\Users\user_name\AppData\Local\Mozilla\Firefox\Profiles\XXXXXXXX.default\cache2

Cookies: C:\Users\user_name\AppData\Roaming\Mozilla\Firefox\Profiles\XXXXXXXX.default\cookies.sqlite

History: C:\Users\user_name\AppData\Roaming\Mozilla\Firefox\Profiles\XXXXXXXX.default\formhistory.sqlite

NOTE: You can use MZCacheView and MZHistoryView tools to analyze the cache folder and history data files respectively.

```
Chrome stores the cache, cookies, and history in the following system
```

locations:

Cache: C:\Users\user_name\AppData\Local\Google\Chrome\User Data\Default\Cache

Cookies: C:\Users\user_name\AppData\Local\Google\Chrome\User Data\Profile 1

History: C:\Users\user_name\AppData\Local\Google\Chrome\User Data\Default Use

NOTE: You can use ChromeCacheView and ChromeHistoryView tools to examine the cache folder and history data files respectively.

```
Microsoft Edge stores cache, cookies and history in the following system
```

locations:

Cache: C:\Users\Admin\AppData\Local\Microsoft\Windows\INetCache

Cookies: C:\Users\Admin\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe \AC\MicrosoftEdge

History: C:\Users\Admin\AppData\Local\Microsoft\Windows\History Use

NOTE: You can use EdgeCookiesView and BrowsingHistoryView tools to analyze the cookies folder and history data files respectively.

```
Database Analysis:

        ▪ Transaction logs
        ▪ Error logs
        ▪ Trace files
        ▪ Link files
        ▪ Volatile database information
        ▪ DBCC logs
        ▪ Database plan cache
```

Examine Error Logs:

Navigate to C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\LOG and open ERRORLOG

Examine Trace Files:

Navigate to C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\LOG and double-click log_n.trc file

Examine Volatile database information:

Collect and analyze the database files (.mdf) and log files (.ldf) from C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\DATA

The fn_dblog() function allows you to retrieve the active portion of the transaction log file

The fn_dblog () function filter transactions by:

```
    ▪ Target database object
    ▪ Specific columns
    ▪ SPID and/or date/time range
```

Use/Examine the DBCC Log Command:

The DBCC LOG command allows you to retrieve the active transaction log files for the specified database

```
Syntax: DBCC LOG(<databasename>, <output>)
```

The output parameter specifies the level of information a incident handler wants to retrieve:

0 = minimal information of each operation such as the Current LSN, Operation, Transaction ID, etc.

1 = slightly more info than 0, such as Flag Bits, Previous LSN, etc.

2 = detailed information, including (AllocUnitId, page id, slot id, etc.)

3 = full information about each operation

4 = full information on each operation along with the hex dump of current transaction row

What about Containment & Eradication of Insider Threats?

```
    ▪ Access Control
    ▪ Encryption
    ▪ Change Passwords
    ▪ Data Centric Audit and Protection (DCAP)
    ▪ HR Oversight
    ▪ Network Security
    ▪ Privileged Users
    ▪ Audit Trails and Log Monitoring
    ▪ Physical Security
```

What about Recovery after an Insider Threat/Attack?

```
BCDR Plans !!!
```