# Blue Team course notes:

**Security Information Management,** also known as**SIM**is specialized security software that helps with the collection, monitoring, and analysis of data and event logs generated from all security devices in a network (IDS, IPS, Antivirus Software, Firewalls). Basically a centralized information service. Some of the actions carried out by**SEM**:

* ***Real-time events monitoring.***
* ***Obtaining security events in devices and applications within the system.***
* ***Correlation of events to provide a clear picture of the information system.***
* ***Analyze logs according to their level of importance.***
* ***Real-time incident response.***

**Security Event Management,**also known as**SEM** is a security software specialized in the identification, collection, monitoring, evaluation, notification and correlation in real-time of events and alerts of a computer system (network devices, security systems (IDS, IPS, Firewall), specialized software (Antivirus), etc.)

### **Benefits**:

* ***Centralization of information from different devices and network elements.***
* ***Reduction of false positives and false negatives.***
* ***Considerable improvement in response time to internal and external threats.***

### **BUT**

* ***They are hard to deploy.***
* ***They have a high market cost***
* ***As they are automated systems, they can present failures that allow for false positives and negatives.***

## **Security Information and Event Management (SIEM) It's a software solution that aggregates and analyzes activity from different resources across an organization’s entire IT infrastructure. SIEM is a combination of security information management (SIM) and security event management (SEM). It uses rules and statistical correlations to help organizations detect threats and turn log entries, and events from security systems, into actionable information.**

## **Stuff 2 know about:**

#### Log // A log is data that is produced by systems, applications, services, and other processes, generating an output that states what actions have been taken. An example of a log is when a user logs into a Windows system, the operating system will make a note that a user has logged in and record it.

#### **WEL** // Windows Event Logs – Windows Event logs or Event Logs are files in binary format (with .evtx extension) stored locally in the Windows directory of a computer. These logs keep a detailed record of anything that happens on a Windows system, from users logging in to program execution. We can use these logs to monitor what happens on Windows endpoints.

#### **Sysmon** // System Monitor – Sysmon is a Windows system service and device driver that monitors and logs system activity to the Windows event log, but can provide more valuable information than standard Windows Event logs

#### **Regex** // Regular Expression – Regex is a string of text that allows you to create patterns that help match, locate, and manage text. Regular expressions can also be used from the command line to find and sort specific files or data on a system.

## **Syslog:**

System Logging Protocol (Syslog) is a standard protocol used to convey event or system log notification messages to a designated server, known as a Syslog server. The Syslog server centralizes data collection from various devices for analysis, review, and intervention. The protocol can be enabled on most network equipment such as switches, routers and firewalls, and even endpoint devices. Windows systems use their own by default as opposed to Syslog (Windows Event Manager). Syslog uses **UDP 514** by default; TCP**514** can be used for more reliability. It isn't encrypted.

A Syslog message is made of three components; a **Priority Value (PRI)**, a **Header**, and a **Message**.

### **Priority Value (PRI)**

The Priority Value is derived from both the **Facility Code** and the **Severity Level.**We can use the simple equation to calculate PRI:  
(facility code \* 8) + Severity value = PRI.

| **Facility Code** | **Description** |
| --- | --- |
| 0 | kernel messages |
| 1 | user-level messages |
| 2 | mail system |
| 3 | system daemons |
| 4 | security/authorization messages |
| 5 | messages generated internally by syslogd |
| 6 | line printer subsystem |
| 7 | network news subsystem |
| 8 | UUCP subsystem |
| 9 | clock daemon |
| 10 | security/authorization messages |
| 11 | FTP daemon |
| 12 | NTP subsystem |
| 13 | log audit |
| 14 | log alert |
| 15 | scheduling daemon |
| 16 | local use 0 (local0, local1...) |

| **Severity Code** | **Description** |
| --- | --- |
| 0 | Emergency: system is unusable |
| 1 | Alert: action must be taken immediately |
| 2 | Critical: critical conditions |
| 3 | Error: error conditions |
| 4 | Warning: warning conditions |
| 5 | Notice: normal but significant condition |
| 6 | Informational :informational messages |
| 7 | Debug: debug-level messages |

## **Header**

This contains identifying information, such as; Timestamp, Hostname, Application name, Message ID. This is useful for understanding where the system message has come from.

## **Message**

This could be simple readable text or only machine-readable. Each message sent to the Syslog server has two labels associated with it that make the message easier to handle. The first label describes the function (facility) of the application that generated it. For example, mail servers typically log using the **mail** facility. The second label specifies the severity level. After these two labels, the action is specified. The action is usually a filename in the **/var/log** directory tree, in which the messages will be stored.

## **Cringedows Event logs:**

Windows Event logs or “Event Logs” are files in binary format (with .evtx extension) stored locally in the Windows directory of a computer with that operating system:

* **Windows Server 2008 to 2019, and Windows Vista to Win10**:  
  %WinDir%\system32\WinEVT\Logs\*.evtx
* **Windows 2000 to WinXP/Windows Server 2003**:  
  %WinDir%\system32\Config\*.evt

These logs keep a detailed record of the vast majority of events that have occurred on the system (hardware events, user logins, program execution and installation, etc.), allowing system administrators to keep track of everything that happens within a system during its execution and being able to diagnose and foresee potential issues. Some security professionals believe that Windows event logs are.. well.. terrible, and that **Sysmon** is a **much**better way to log information on Windows endpoints. As usual, windows is just worse.

**Normalization merges events containing different data into a reduced format that contains common event attributes. Categorization involves adding meaning to events – identifying log data related to system events, authentication, local/remote operations, etc.**

## **Log Enrichment**

Log enrichment involves adding important information that can make the overall data more beneficial for security analysts when investigating alerts or unusual activity e.g. geographical location of an IP to improve general analytics.

## **Log Indexing**

By indexing attributes that are shared by a large number of logs, it can make searching for specific attributes across large data faster**.**

## **Normalization**

SIEM log normalization is the process of changing log formats into a format that is as similar as possible across all devices and log sources, because not all logs are created equally. When writing rules, Sigma can be used to convert from/to any log format.

## **Search queries(Splunk):**

**We can combine text strings, file names, process names, IP addresses, operators and lots more to look for specific data in our index. Some examples:**

* **Searching With Fields** (Selected Fields, Interesting Fields)
* **Field / Value Pairs**(AND, OR, NOT operators)
* **Wildcards**
* **Processes** (Sysmon Image field)

The simplest search we can conduct is for a field and a value, for example, searching against our data for the source IP field (src) and the IP address value 10.10.10.50.

search src="10.10.10.50"

search src="10.10.10.50" OR dst="10.10.10.50" - sql like syntax

search pass\* AND fail\* - searches for any and every thing that contains "pass" and "fail" (password failure, pass fail etc.)

**index="some\_thing" earliest=0 Image="\*\\cmd.exe" | stats values(CommandLine) by host - searches for the executable that has created a process (cmd) and retrieves values(commands) that were used(CommandLine)**

**Suricata logs come from the Suricata NIDS, xmlwineventlogs represent Sysmon logs, Fortigate\_UTM represents the Fortigate Unified Threat Management firewall, and wineventlogs represent standard Windows event logging.**

**This search looks up a process creation event referencing a malicious file**:

index=main source="xmlwineventlog:microsoft-windows-sysmon/operational" EventCode=1

| search CommandLine="\*Salaries.xls\*"

| table \_time host user ProcessId CommandLine

**This search is a little more in-depth and displays additional fields such as the parent process name, parent process ID, path etc.:**

index=main source="xmlwineventlog:microsoft-windows-sysmon/operational" EventCode=1

| search CommandLine="\*Salaries.xls\*"

| table parent\_process\_name parent\_process\_id parent\_process\_path process\_path

