Team 6 - Team 404
Alejandro Zamora, Jacob Torres, Jorge Felix, Eddy Todd, Matt Montoya

CRC

Concrete Class: Splunk	
Superclass: None	
Subclasses: None	
Description: A third party software tool for transforming log files into normalized log entries	
Responsibilities: 1. Transform a log file into normalized log entries. 2. Store log entries in the normalized data files. 3. Process incoming data into individual activities according to the nature of the data.	**Collaborations:**
Attributes:	

Concrete Class: Maltego	
Superclass: None	
Subclasses: None	
Description: A third party software tool for graphing.	
Responsibilities: 1. Create graphs with nodes and relationships between nodes.	**Collaborations:**
Attributes:	

Concrete Class: Log File	

Superclass: None

Subclasses: None

Description: A computer data object that stores logs.

Responsibilities:	**Collaborations:**
1. Contain transcription with timestamps in one-minute interval if the log file is of type “audio” or “video”. 2. Contain extracted text if the log file is of type “image” or “pdf”. 3. Contain a timestamp per line, bounded by the start data, end date, start time, and end time specified in the event configuration. 4. When a timestamp property of a previously saved event is changed, the impact of the change shall be restricted to the “not-validated” log files.	1. Use Splunk to transform log files into normalized log entries[Splunk (1)]. 2. Will be stored in the Vector DB [Vector Database (1)] 3. Create an Enforcement action report for errors in log files[Enforcement Action Report (3)]

Attributes:

1. Log File Name
2. Cleansing Status
3. Validation Status
4. Ingestion Status
5. Acknowledgement Status

Concrete Class: Log Entry

Superclass: None

Subclasses: Log Entry

Description: A log entry is the output of the log file. Log entries contain information from the log file and are recorded details of an ingested log.

Responsibilities:	**Collaborations:**
1. Contain sanitized log information	1. Will be represented as nodes [Node (1)] 2. Will be retrieved from splunk [Splunk (2)] 3. Use OCR to transform image to

<table>
<tr><td></td><td>text[OCR (3)]</td></tr>
<tr><td colspan="2"><b>Attributes:</b></td></tr>
</table>

<table>
<tr><td colspan="2"><b>Concrete Class:</b> Significant Log Entry</td></tr>
<tr><td colspan="2"><b>Superclass:</b> Log Entry</td></tr>
<tr><td colspan="2"><b>Subclasses:</b> None</td></tr>
<tr><td colspan="2"><b>Description:</b> Logs that are important to the overall cyber-attack scenario between the Blue Team and the Red Team.</td></tr>
<tr><td><b>Responsibilities:</b>
1. Contain relevant information for a vector.</td><td><b>Collaborations:</b>
1. Will be part of at least one vector [Vector (1)]</td></tr>
<tr><td colspan="2"><b>Attributes:</b>
1. Log Entry Number
2. Log Entry Timestamp
3. Log Entry Content
4. Host
5. Source
6. Source Type</td></tr>
</table>

<table>
<tr><td colspan="2"><b>Concrete Class:</b> Node</td></tr>
<tr><td colspan="2"><b>Superclass:</b> None</td></tr>
<tr><td colspan="2"><b>Subclasses:</b> None</td></tr>
<tr><td colspan="2"><b>Description:</b> Visual representation of a significant event that was marked for the current vector.</td></tr>
<tr><td><b>Responsibilities:</b>
1. Represent a log entry as part of the graph</td><td><b>Collaborations:</b>
1. Will be part of at least one graph[Graph (1)]
2. Will use relationships to link to other nodes [Relationship (1)]
3. Will be displayed with an icon [Icon (3)]</td></tr>
</table>

Attributes:
1. Node ID
2. Node Name
3. Node Timestamp
4. Node Description
5. Log Entry Reference
6. Log Creator
7. Event Type
8. Icon Type
9. Source
10. Node Visibility

Concrete Class: Relationship

Superclass: None

Subclasses: None

Description: Represent the relationship between nodes.

Responsibilities:
1. Correlate nodes to one another.

Collaborations:

Attributes:
1. Relationship ID
2. Parent ID
3. Child ID
4. Label

Concrete Class: Vector Database

Superclass: None

Subclasses: None

Description: A database that shows cleansed logs of vectors ingested.

Responsibilities:
1. Store the cleansed log files in permanent storage.
2. Store significant log entries in

Collaborations:

permanent storage. 3. Notify the lead when a record is pushed 4. Provide vectors to analysts. 5. Allow the lead to approve pushes. 6. Get vectors from analysts	
Attributes:	

Concrete Class: Analyst	
Superclass: None	
Subclasses: Lead	
Description: A user who has the privilege to perform all functionalities of the system except verify sync from analysts.	
Responsibilities: 1. Use the PICK tool for log analysis	**Collaborations:** 1. Get vectors from Vector DB[Vector Database (4)] 2. Push vectors to the Vector DB [Vector Database (6)]
Attributes: 1. Human	

Concrete Class: Lead	
Superclass: Analyst	
Subclasses: None	
Description: A user who has the privilege to perform all functionalities of the system.	
Responsibilities: 1. Use the Pick tool for log analysis 2. Version control 3. Database control	**Collaborations:** 1. Approve pushes made to the Vector DB [Vector Database (1)]
Attributes:	

<table>
<tr><td>1. Human
2. Superior of analyst</td></tr>
</table>

<table>
<tr><td colspan="2"><b>Concrete Class:</b> Graph</td></tr>
<tr><td colspan="2"><b>Superclass:</b> None.</td></tr>
<tr><td colspan="2"><b>Subclasses:</b> None.</td></tr>
<tr><td colspan="2"><b>Description:</b> A visual representation of a vector.</td></tr>
<tr><td><b>Responsibilities:</b>
1. A graph shall comprise of at least one node.
2. Allow the analyst to add nodes, edit nodes, delete nodes, add relationships, edit relationships, and delete relationships.
3. Create a PNG of the graph.</td><td><b>Collaborations:</b>
1. Use Maltego to generate the graph [Maltego (1)]</td></tr>
<tr><td colspan="2"><b>Attributes:</b>
1. Export Format
2. Orientation
3. Internal Units
4. Interval
5. Position of Nodes
6. Position of Relationships</td></tr>
</table>

<table>
<tr><td colspan="2"><b>Concrete Class:</b> Vector</td></tr>
<tr><td colspan="2"><b>Superclass:</b> None</td></tr>
<tr><td colspan="2"><b>Subclasses:</b> None</td></tr>
<tr><td colspan="2"><b>Description:</b> A description of a significant event.</td></tr>
<tr><td><b>Responsibilities:</b>
1. Correlate log entries that correspond to an event.</td><td><b>Collaborations:</b>
1. Will be composed of at least one significant log entry[Significant Log Entry (1)]</td></tr>
<tr><td colspan="2"><b>Attributes:</b></td></tr>
</table>

1. Vector Name
2. Vector Description
3. Log Entries

Concrete Class: OCR

Superclass: None

Subclasses: None

Description: Allows the user to make any type of document into a word searchable document.

Responsibilities:

1. Read documents
2. Recognize character
3. Translate images to text

Collaborations:

Attributes:

1. 3rd party software

Concrete Class: Event Configuration

Superclass: None

Subclasses: None

Description: This class enables the user to configure the events.

Responsibilities:

1. Keep track of event names
2. Keep track of start and end times
3. Organize team folders
4. Track description of events

Collaborations:

Attributes:

1. Event Name
2. Event Description
3. Event Start Timestamp
4. Event End Timestamp
5. Root Directory
6. Red Team Folder
7. White Team Folder

8. Blue Team Folder
9. Lead
10. Lead’s IP Address
11. Connection Established

Concrete Class: Enforcement Action Report	
Superclass: None	
Subclasses: None	
Description: Handles errors when anomalous events happen.	
Responsibilities: 1. Track error codes 2. Display error code 3. Track description of error codes 4. Track the state of the system	**Collaborations:**
Attributes: 1. Line Number 2. Error Message	

Concrete Class: Icon	
Superclass: None	
Subclasses: None	
Description: An image that helps represent what the event is in the graph.	
Responsibilities: 1. Track event 2. Allow the change of icon 3. Display as part of a node	**Collaborations:**
Attributes: 3. Icon Name 4. File Path	

Concrete Class: Directory	
Superclass: None	
Subclasses: None	
Description: Path and structure of the log files.	
Responsibilities: 1. Perform structure check when data ingestion starts. 2. Generate root directory structure error if it fails to contain three folders or if the folder names specified in the event configuration don't match. 3. Store the log files in their corresponding folder	**Collaborations:**
Attributes: 1. Red Team Folder 2. Blue Team Folder 3. White Team Folder	

Concrete Class: Search and Filter	
Superclass: None	
Subclasses: None	
Description: Gives the user to search for specific criteria	
Responsibilities: 1. Read logs. 2. Display search results 3. When a search operation is complete, the system shall return result that matches the searched keyword with the searched keyword highlighted in the search result. 4. Perform logical searching. 5. Perform wildcard searching.	**Collaborations:**
Attributes:	

Concrete Class: Team Configuration UI

Superclass: None

Subclasses: None

Description: This class will contain a Panel that will display the Team Configuration.

Responsibilities:

1. Displays the attributes of the Team Configuration Panel UI.

Collaborations:

Attributes: The team configuration shall have the following components:

1. A label labeled as “Team configuration”
2. A checkbox labeled as “Lead”
3. A text field labeled as “Lead IP address
4. A label labeled as “No. of established connections to the lead’s IP address”
5. A label labeled with the number of connections to the lead’s IP address.
6. A button labeled as “Connect”.
7. A label labeled as “Team configuration”
8. A checkbox labeled as “Lead”
9. A text text field labeled as “Lead IP address”
10. A label label labeled as “No. of established connections to the lead’s IP address”
11. A label labeled with the number of connections to the lead’s IP address
12. A button labeled as “Connect”

Concrete Class: Event Configuration UI

Superclass: None

Subclasses: None

Description: This class will contain a Panel that will display the Event Configuration.

Responsibilities:

1. Displays the attributes of the Event configuration UI.

Collaborations:

Attributes:

1. A label labeled as “Event Configuration”
2. A text field labeled as “Event Name”
3. A text Field labeled as “Event description”

4. A text field labeled as “Event start timestamp”
5. A text Field labeled as “Event end timestamp”
6. A button labeled as “Save Event”

Concrete Class: Directory Configuration UI

Superclass: None

Subclasses: None

Description: This class will contain a Panel that will display the Directory Configuration.

Responsibilities:

1. Displays the attributes of the Directory configuration panel UI.

Collaborations:

Attributes:

1. A label labeled as “Directory configuration”
2. A text field labeled as “Root directory”
3. A text field labeled as “Red team folder”
4. A text field labeled as “Blue team folder”
5. A text field labeled as “White team folder”
6. A button labeled as “Start data ingestion”.

Concrete Class: Vector Configuration UI

Superclass: None

Subclasses: None

Description: This class will contain a Panel that will display the Vector configuration.

Responsibilities:

1. Displays the attributes of the Directory configuration panel UI.
2. The Vector configuration will display the vector table.

Collaborations:

Attributes:

1. A label labeled as “Vector configuration”
2. A vector table
3. A button labeled as “Add vector”

4. A button labeled as “Delete vector”
5. A button labeled as “Edit vector”.
6. A column of checkboxes
7. A column of text fields with the column header labeled as “Vector name”
8. An upward/downward arrow within the column header labeled as “Vector name”
9. A column of text fields with the column header labeled as “Vector description”
10. An upward/downward arrow within the column header labeled as “Vector description”.

Concrete Class: Log File Configuration UI

Superclass: None

Subclasses: None

Description: This class will contain a Panel that will display the Log File configuration.

Responsibilities:	**Collaborations:**
1. Display the attributes of the Log File configuration UI. 2. The Log file Configuration shall display the Log File Table. 3. The Log file Configuration shall display an enforcement action report table.	

Attributes:

1. A column of labels with the column header labeled as “File name”
2. An upward/downward arrow within the column header labeled as “File name”
3. A column of labels with the column header labeled as “Source”
4. An upward/downward arrow within the column header labeled as “Source”
5. A column of labels with the column header labeled as “Cleansing status”
6. An upward/downward arrow within the column header labeled as “Cleansing status”
7. A column of labels with the column header labeled as “Validation status”
8. An upward/downward arrow within the column header labeled as “Validation status”
9. A column of labels with the column header labeled as “Ingestion status”
10. An upward/downward arrow within the column header labeled as “Ingestion status”
11. A column of buttons with the column header labeled as “View enforcement action report”.
12. A column of labels with the column header labeled as “File name”
13. An upward/downward arrow within the column header labeled as “File name”
14. A column of labels with the column header labeled as “Source”
15. An upward/downward arrow within the column header labeled as “Source”
16. A column of labels with the column header labeled as “Cleansing status”

17. An upward/downward arrow within the column header labeled as “Cleansing status”
18. A column of labels with the column header labeled as “Validation status”
19. An upward/downward arrow within the column header labeled as “Validation status”
20. A column of labels with the column header labeled as “Ingestion status”
21. An upward/downward arrow within the column header labeled as “Ingestion status”
22. A column of buttons with the column header labeled as “View enforcement action report”.

Concrete Class: Filter Configuration UI

Superclass: None

Subclasses: None

Description: Allows users to input filter parameters

Responsibilities:

1. Track user input
2. Search through system for search criteria
3. Display search criteria

Collaborations:

Attributes:
The filter configuration shall have the following components:

1. A label labeled as “Filter configuration”
2. A text field labeled as “Keyword search”
3. A label labeled as “Creator”
4. A checkbox labeled as “Red”
5. A checkbox labeled as “White”
6. A checkbox labeled as “Blue”
7. A label labeled as “Event type”
8. A checkbox labeled as “Red”
9. A checkbox labeled as “White”
10. A checkbox labeled as “Blue”
11. A text field labeled as “Start timestamp”
12. A text field labeled as “End timestamp”
13. A button labeled as “Apply Filter”.

Concrete Class: Log Entry Configuration UI

Superclass: None

Subclasses: None

Description: Log entry configuration allows the user to configure log entries.

Responsibilities:

1. Track logs
2. Update log tables
3. Track vectors

Collaborations:

Attributes:

1. The log entry configuration shall have the following components:
   a. Label labeled as “Log entry configuration”
   b. A Log entry table.
2. The log entry table in the log entry configuration shall include the following components:
   a. A column of checkboxes
   b. A column of text fields with the column header labeled as “List number”
   c. An upward/downward arrow within the column header labeled as “List number”
   d. A column of text fields with the column header labeled as “Log entry timestamp”
   e. An upward/downward arrow within the column header labeled as “Log entry timestamp”
   f. A column of text fields with the column header labeled as “Log entry event (including Log entry content, host, source, sourcetype)”
   g. An upward/downward arrow within the column header labeled as “Log entry event”
   h. A column of dropdown boxes with the column header labeled as “Vector”.

Concrete Class: Export Configuration UI

Superclass: None

Subclasses: None

Description: Allows the user to export configuration.

Responsibilities:

1. Track the current state.
2. Know the path of current directory
3. Know all configurations

Collaborations:

Attributes:

1. The export configuration shall have the following components:
   a. A dropdown box labeled as “Export format”
   b. A button labeled as “Export”.

Concrete Class: Change Configuration

Superclass: None

Subclasses: None

Description: Allows the user to change the configuration by allowing commits to be done and also allow the undo of other action to the event configuration

Responsibilities:

1. Allows users to commit changes.
2. Keep track of changes.
3. Undo changes.

Collaborations:

Attributes:

1. A text area labeled as”change list”.
2. A button labeled as”commit”
3. A button labeled as “undo”

Concrete Class: Vector DB Configuration UI

Superclass: None

Subclasses: None

Description: Allows the user to configure the DataBase.

Responsibilities:

1. Display connection status
2. Pull vector DB tables

Collaborations:

3. Push vector DB tables
4. Approval / Sync

Attributes:

1. The vector DB configuration for analyst shall have the following components:
2. A label labeled as “Connection status to lead:”
3. A label labeled with the connection status
4. A label labeled as “Pulled vector DB table (Analyst)”
5. A pulled vector DB table
6. A button labeled as “Pull”
7. A label labeled as “Pushed vector DB table (Analyst)”
8. A pushed vector DB table
9. A button labeled as “Push”.

Concrete Class: Icon Configuration UI

Superclass: None

Subclasses: None

Description: Allows the user to give visual representation of items in graph views

Responsibilities:

1. Track Icons
2. Tracks logs
3. Tracks vectors
4. Tracks Nodes
5. Tracks relationships

Collaborations:

Attributes:

1. The icon configuration shall have the following components:
   a. A label labeled as “Icon configuration”
   b. An icon table
   c. A button labeled as “Add Icon”
   d. A button labeled as “Delete Icon”
   e. A button labeled as “Edit Icon”.
   f. The icon table in the icon configuration shall include the following components:
   g. A column of checkboxes
   h. A column of text fields with the column header labeled as “Icon name”
   i. An upward/downward arrow within the column header labeled as “Icon name”
   j. A column of text fields with the column header labeled as “Icon source”
   k. An upward/downward arrow within the column header labeled as “Icon source”
   l. A column of images with the column header labeled as “Image preview”
   m. An upward/downward arrow within the column header labeled as “Image

preview”

Concrete Class: Graph Builder Configuration UI

Superclass: None

Subclasses: None

Description: This class enables the user to configure the graph builder.

Responsibilities:

Collaborations:

Attributes:

1. The graph builder configuration shall have the following components:
    a. A dropdown box labeled as “Vector”
    b. A label labeled as “Description:”
    c. A label labeled with the description of the selected vector
    d. A button labeled as “Add node”
    e. A button labeled as “Add relationship”
    f. A button labeled as “Delete node”
    g. A button labeled as “Delete relationship
    h. A button labeled as “Edit node”
    i. A button labeled as “Edit relationship.

Concrete Class: Nodes Configuration in Tabular Format UI

Superclass:

Subclasses: None

Description: Allows users to edit the nodes in tabular UI.

Responsibilities:
1. Track node config
2. Track nodes
3. Track event types
4. Track logs

Collaborations:

5. Track Icon types
6. Track times

Attributes:

1. A column of checkboxes
2. A row of toggles with the row header labeled as “Node property visibility”
3. A column of labels with the column header labeled as “Node ID”
4. An upward/downward arrow within the column header labeled as “Node ID”
5. A column of text fields with the column header labeled as “Node name”
6. An upward/downward arrow within the column header labeled as “Node name”
7. A column of text fields with the column header labeled as “Node timestamp”
8. An upward/downward arrow within the column header labeled as “Node timestamp”
9. A column of text fields with the column header labeled as “Node description”
10. An upward/downward arrow within the column header labeled as “Node description”
11. A column of text fields with the column header labeled as “Log entry reference”
12. An upward/downward arrow within the column header labeled as “Log entry reference”
13. A column of dropdown boxes with the column header labeled as “Log creator”
14. An upward/downward arrow within the column header labeled as “Log creator”
15. A column of dropdown boxes with the column header labeled as “Event type”
16. An upward/downward arrow within the column header labeled as “Event type”
17. A column of dropdown boxes with the column header labeled as “Icon type”
18. An upward/downward arrow within the column header labeled as “Icon type”
19. A column of text fields with the column header labeled as “Source”
20. An upward/downward arrow within the column header labeled as “Source”
21. A column of toggles with the column header labeled as “Node visibility”
22. An upward/downward arrow within the column header labeled as “Node visibility”

Concrete Class: Nodes Configuration in Graphical Format UI

Superclass:

Subclasses: None

Description: Allows the user to configure nodes.

Responsibilities:

1. Track timelines
2. Track node properties
3. Track Relationships
4. Update graphs in tabular

Collaborations:

Attributes:
The nodes configuration in graphical format shall have the following components:

1. A label labeled as “Nodes configuration in graphical format”
2. A dropdown box labeled as “Timeline orientation”
3. A dropdown box labeled as “Interval units”
4. A text field labeled as “Interval”
5. A timeline
6. A set of nodes with node properties
7. A set of relationships with their associated label
8. A button labeled as “Zoom in”
9. A button labeled as “Zoom out”

Concrete Class: Relationship Configuration

Superclass: None

Subclasses: None

Description: Allows the user to correlate relationships.

Responsibilities:

1. Track all nodes
2. Track graphs
3. Manage Relationships
4. Tack Relationship tables

Collaborations:

Attributes:

1. The relationship configuration shall have the following components:
2. A label labeled as “relationship configuration”
3. A relationship table.
4. The relationship table in the relationship configuration shall include the following components:
5. A column of checkboxes
6. A column of labels with the column header labeled as “Relationship ID”
7. An upward/downward arrow within the column header labeled as “Relationship ID”
8. A column of text fields with the column header labeled as “Parent”
9. An upward/downward arrow within the column header labeled as “Parent”
10. A column of images with the column header labeled as “Child”
11. An upward/downward arrow within the column header labeled as “Child”
12. A column of images with the column header labeled as “Label”
13. An upward/downward arrow within the column header labeled as “Label”.