

# Senior Project Outline – Bull’s Eye

Table of Contents

[Senior Project Outline – Bull’s Eye 1](#_Toc414007606)

[Introduction 1](#_Toc414007607)

[1. Express 2013 for Windows (Links to an external site.) 1](#_Toc414007608)

[STIX and TAXII 2](#_Toc414007609)

[Senior Project Definitions 2](#_Toc414007610)

[STIX Core 2](#_Toc414007611)

[Student Requirements: 2](#_Toc414007612)

[TAXII Protocol 3](#_Toc414007613)

[Student Requirements: 3](#_Toc414007614)

[BullsEye Database 3](#_Toc414007615)

[Student Requirements: 3](#_Toc414007616)

[BullsEye Cloud Database 4](#_Toc414007617)

[Student Requirements: 4](#_Toc414007618)

[BullsEye Information Sharing Services 4](#_Toc414007619)

[Student Requirements: 5](#_Toc414007620)

[BullsEye UI - Forms.NET 5](#_Toc414007621)

[Student Requirements: 6](#_Toc414007622)

[BullsEye UI – ASP.NET 6](#_Toc414007623)

[Student Requirements: 7](#_Toc414007624)

[BullsEye UI – XAML.NET 7](#_Toc414007625)

[Student Requirements: 7](#_Toc414007626)

[BullsEye UI – 3D Enterprise 8](#_Toc414007627)

[Student Requirements: 8](#_Toc414007628)

[BullsEye UI – 3D Consumer 9](#_Toc414007629)

[Student Requirements: 9](#_Toc414007630)

[BullsEye Asset Manager 10](#_Toc414007631)

[Student Requirements: 10](#_Toc414007632)

[BullsEye Ticket Manager 10](#_Toc414007633)

[Student Requirements: 10](#_Toc414007634)

[BullsEye Windows Mobile 11](#_Toc414007635)

[Student Requirements: 11](#_Toc414007636)

# Introduction

The goal of this project is to provide a light-weight and cost effect Cyber Security monitoring and threat information sharing system. The system is based off of STIX/TAXII protocols to enumerate vulnerabilities which can then feed into an Asset Management system for vulnerability detection. The following sections outline the different phases of the project. Each phase identifies the project description and resources (students) required to fulfill the task. All components are written in C# using the .NET framework and infrastructure and based off the initiatives located at <https://github.com/STIXProject>. Student resources identified as “novice” can use the following material to learn C# and get “up to speed” on programing in that language.

C# resources:

### [Express 2013 for Windows (Links to an external site.)](http://www.visualstudio.com/en-us/products/visual-studio-express-vs.aspx)

1. [Introduction to C# .NET  (Links to an external site.)](https://www.youtube.com/playlist?list=PL1C97E0468DDBB929)
2. NetworkProgramming.png **C# 2012 for Programmers,** 5th Edition; Paul and Harvey Deitel ISBN 978-0-13-344057-7

# STIX and TAXII

STIX and TAXII are becoming more dominate players in the threat information arena. In the STIX [news letter](http://stix.mitre.org/news/), it was stated that "The field of cybersecurity is already rife with acronyms. But listen for a couple of key new terms at today's White House summit on cybersecurity that are essential for explaining how the government will expand information sharing with the private sector… The two words are [STIX](http://stix.mitre.org/) and [TAXII](http://taxii.mitre.org/), a programming language and data delivery method that are meant to bring these parties together in the virtual word. They offer a potential two-way street to the [information sharing and collaboration](http://www.whitehouse.gov/issues/foreign-policy/cybersecurity/summit#section-collaboration) that government officials, retailers and Wall Street want more of to fight cybercrimes." Students new to this technology should review the introduction slides “STIX Module 1 - Session 1 - STIX Introduction” and “TAXII Module 1 - Session 1 - Overview” to get a better ideas of the intent and capabilities of STIX and TAXII.

# Senior Project Definitions

The following is a list of projects that are under consideration. If you have interest in any of these project, contact Bill Gauvin at [wgauvin@usf.edu](mailto:wgauvin@usf.edu). These projects are just an outline, if you have a more specific interest in Cyber Threat Intelligence and a new proposal, contact me as well.

## STIX Core

Core component to provide STIX parsing of XML formatted messages. Used by other components to process messages and provide a data-feed to other users (UI, Databases and asset enumeration). The current Python implementation provided by the Department of Homeland Security (DHS) will be used as the model. The unit test provided by the Python implementation will be used to measure the success of this phase of the project. This initiative will take more than one person to complete the work and may span several semesters of work.

### Student Requirements:

Number of Students: 1 to 2, as student resources become available.

Level: Novice C# programmer. The student should have experience in C++ and Java, as well as familiarity with an Integrated Development Environment. Students should review the Department of Homeland Security (DHS) presentation “STIX Module 1 - Session 1 - STIX Introduction” for further details on the language format.

Goal: Learn how to program in a development environment which uses an Agile process to define the requirements, architecture, design, development and testing for key components of the core system. GitHub will be used as the source repository for all projects of BullsEye. The specific goal for the student is to learn C# and the technology specific to Cyber Threat Information Sharing. The student will become familiar with basic C# programming, creating libraries and parsing XML formatted messages.

## TAXII Protocol

Core communication protocol used to provide publishing/subscriber feeds to BullsEye components from NIST, DHS and other government and enterprise feeds. The current Python reference implementation will be used as the guidelines for development. The measure of success will be the ability to register for security information feeds from the public TAXII server site and obtain TAXII messages which will then be fed into the STIX core for message processing.

### Student Requirements:

Number of Students: 1 to 2, as student resources become available.

Level: Novice C# programmer. The student should have experience in C++, Java, network protocols and programming, as well as familiarity with an Integrated Development Environment. Students should review the Department of Homeland Security (DHS) presentation “TAXII Module 1 - Session 1 - Overview” for further details on the protocol format and operational characteristics.

Goal: Learn how to program in a development environment which uses an Agile process to define the requirements, architecture, design, development and testing for key components of the core system. GitHub will be used as the source repository for all projects of BullsEye. The specific goal for the student is to learn C# and the technology specific to Cyber Threat Information Sharing. The student will become familiar with basic C# programming, creating libraries and implementing a tunneled network protocol (TAXII) using HTTP/HTTPs.

## BullsEye Database

Localized database used to store vulnerability information. This component provides the means to enumerate vulnerabilities and filter on specific assets (located in an Asset Management Database). The student is responsible for command line tools to input data, query the database and export data in required formats (for example CSV) as they are defined.

### Student Requirements:

Number of Students: 1

Level: Novice C# programmer. The student should have experience in C++, Java, and database programming, as well as familiarity with an Integrated Development Environment. Students should review the Department of Homeland Security (DHS) presentation “STIX Module 1 - Session 1 - STIX Introduction” for further details on the language format.

Goal: Learn how to program in a development environment which uses an Agile process to define the requirements, architecture, design, development and testing for key components of the core system. GitHub will be used as the source repository for all projects of BullsEye. The specific goal for the student is to learn C# and the technology specific to Cyber Threat Information Sharing. The student will become familiar with basic C# programming, creating command line tools, defining the schema used to contain threat information and implementing a threat information database.

Dependencies: This project is dependent on the STIX core project. The student may need to implement “stubs” used in STIX XML parsing or translation methods in order to implement their portion of the project.

## BullsEye Cloud Database

Cloud database used to store vulnerability information. This component provides the means to enumerate vulnerabilities and filter on specific assets (located in an Asset Management Database). This component is used by both distributed and mobile applications for vulnerability assessment. The project uses Azure as the cloud service mechanism to provide high availability and scale to the BullsEye solution set.

### Student Requirements:

Number of Students: 1 to 2, as student resources become available. This project may be split up into Threat information and asset information by the students if desired.

Level: Novice C# programmer. The student should have experience in C++, Java, databases, network protocols and programming and be familiar with an Integrated Development Environment. Students should review the Department of Homeland Security (DHS) presentation “STIX Module 1 - Session 1 - STIX Introduction” for further details on the language format.

Goal: Learn how to program in a development environment which uses an Agile process to define the requirements, architecture, design, development and testing for key components of the core system. GitHub will be used as the source repository for all projects of BullsEye. The specific goal for the student is to learn C# and the technology specific to Cyber Threat Information Sharing. The student will become familiar with basic C# programming, Windows Azure for Cloud Computing, creating both a UI test front-end and command line tools, defining the schema used to contain threat and asset information; and the implementation of this information within the cloud database.

Dependencies: This project is dependent on the STIX core project. The student may need to implement “stubs” used in STIX XML parsing or translation methods in order to implement their portion of the project.

## BullsEye Information Sharing Services

Localized service used to both retrieve and aggregate security information and propagate this information to other servers and devices, to include a mobile component. The information sharing service will collect information from a number of “feeds” and aggregate the information into a meaningful collection. The information may come from DHS STIX/TAXII feeds as well as the National Vulnerability Database (NVD).

### Student Requirements:

Number of Students: 1 to 2, as student resources become available.

Level: Novice C# programmer. The student should have experience in C++, Java, databases, network protocols and programming and be familiar with an Integrated Development Environment.

Goal: Learn how to program in a development environment which uses an Agile process to define the requirements, architecture, design, development and testing for key components of the core system. GitHub will be used as the source repository for all projects of BullsEye. The specific goal for the student is to learn C# and the technology specific to Cyber Threat Information Sharing. The student will become familiar with basic C# programming, creating libraries and using TAXII and other network protocols to pull from various information feeds that will then be aggregated using student defined processes.

## BullsEye UI - Forms.NET

Native Windows Forms (C# .NET) application used to navigate STIX messages and enumerate objects using a graphical representation, much like the current stix-viz located at <https://github.com/STIXProject/stix-viz> and displayed in the diagram below:

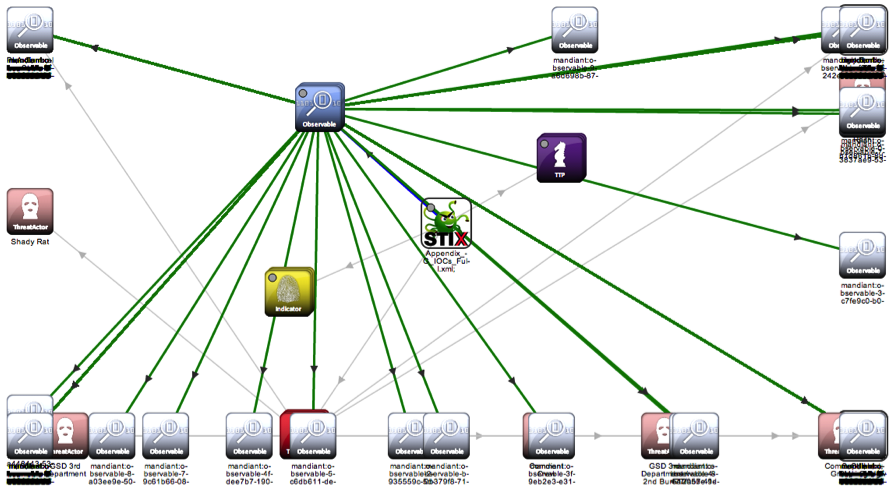


Figure 1 – STIX Graphical User Interface

The native UI will allow analyst to navigate through/and show the relationships between common properties of an attack to include campaigns, TTPs, Threat Actors, Targets, Indicators and Incidents, Observables and Course of Actions. While old (STIX v1.1 has changed slightly), some of these relationships are displayed below.

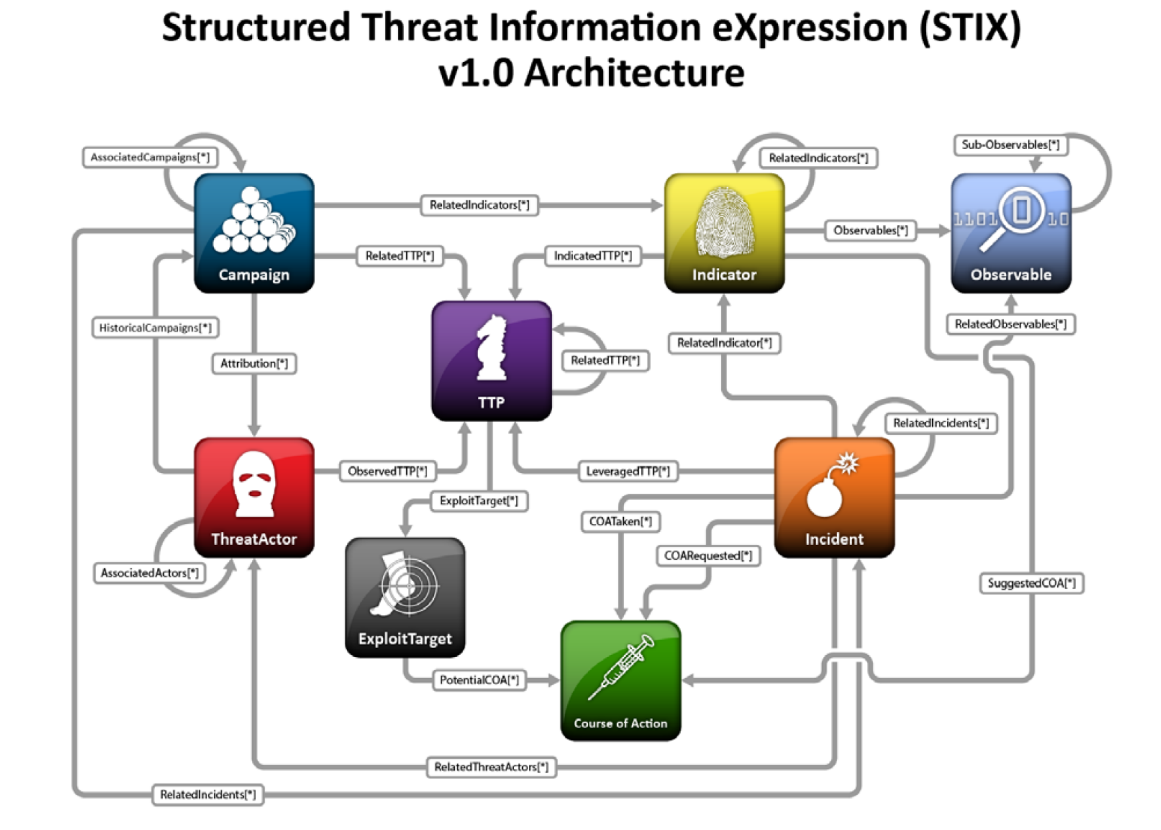


Figure 2 – STIX Object Relationship

### Student Requirements:

Number of Students: 1

Level: Novice C# programmer. The student should have experience in C++, Java, UI development and programming and be familiar with an Integrated Development Environment. Students should review the Department of Homeland Security (DHS) presentation “STIX Module 1 - Session 1 - STIX Introduction” for further details on the language format.

Goal: Learn how to program in a development environment which uses an Agile process to define the requirements, architecture, design, development and testing for key components of the core system. GitHub will be used as the source repository for all projects of BullsEye. The specific goal for the student is to learn C# technology as well as User Interface application development used to present Cyber Threat Information using presentation techniques that make it easy to understand and share that information. The student will become familiar with basic C# programming, creating a native Windows Forms application in C# .NET.

Dependencies: This project is dependent on the STIX core project. The student may need to implement “stubs” used in STIX XML parsing or translation methods in order to implement their portion of the project.

## BullsEye UI – ASP.NET

Windows Web application (C# ASP .NET) used to navigate STIX messages and enumerate objects using a graphical representation, much like the current stix-viz located at <https://github.com/STIXProject/stix-viz>, refer to figure 1. The ASP UI will be hosted on an IIS server (IIS express for development) and allow analyst to navigate through/and show the relationships between common properties of an attack to include campaigns, TTPs, Threat Actors, Targets, Indicators and Incidents, Observables and Course of Actions. Refer to Figure 2 to understand the relationships between the STIX objects.

### Student Requirements:

Number of Students: 1

Level: Novice C# programmer. The student should have experience in C++, Java, UI/Web application development and programming; and be familiar with an Integrated Development Environment. Students should review the Department of Homeland Security (DHS) presentation “STIX Module 1 - Session 1 - STIX Introduction” for further details on the language format.

Goal: Learn how to program in a development environment which uses an Agile process to define the requirements, architecture, design, development and testing for key components of the core system. GitHub will be used as the source repository for all projects of BullsEye. The specific goal for the student is to learn C# APS .NET technology as well as User Interface application development used to present Cyber Threat Information using presentation techniques that make it easy to understand and share that information. The student will become familiar with basic C# programming, creating an ASP .NET Web Forms application in C#.

Dependencies: This project is dependent on the STIX core project. The student may need to implement “stubs” used in STIX XML parsing or translation methods in order to implement their portion of the project.

## BullsEye UI – XAML.NET

Windows 8.0 application (C# XAML .NET) used to navigate STIX messages and enumerate objects using a graphical representation, much like the current stix-viz located at <https://github.com/STIXProject/stix-viz>, refer to figure 1.The XAML UI will be hosted on Windows 8.0, which provisions for desktops, laptops and touch-pad devices. The application allows analyst to navigate through/and show the relationships between common properties of an attack to include campaigns, TTPs, Threat Actors, Targets, Indicators and Incidents, Observables and Course of Actions. Refer to Figure 2 to understand the relationships between the STIX objects.

### Student Requirements:

Number of Students: 1

Level: Novice C# programmer. The student should have experience in C++, Java, UI/Web application development and programming; and be familiar with an Integrated Development Environment. Students should review the Department of Homeland Security (DHS) presentation “STIX Module 1 - Session 1 - STIX Introduction” for further details on the language format.

Goal: Learn how to program in a development environment which uses an Agile process to define the requirements, architecture, design, development and testing for key components of the core system. GitHub will be used as the source repository for all projects of BullsEye. The specific goal for the student is to learn C# XAML .NET technology as well as User Interface application development used to present Cyber Threat Information using presentation techniques that make it easy to understand and share that information. The student will become familiar with basic C# programming, creating an XAML .NET for the latest Windows 8.0 platforms.

Dependencies: This project is dependent on the STIX core project. The student may need to implement “stubs” used in STIX XML parsing or translation methods in order to implement their portion of the project.

## BullsEye UI – 3D Enterprise

Windows 3D application (C# .NET and Unity SDK) used to navigate STIX messages and enumerate objects using a graphical representation, presenting the information current being displayed in stix-viz located at <https://github.com/STIXProject/stix-viz>, refer to figure 1. The UI can be hosted on multiple platforms that is supported by Unity, which provisions for servers, desktops, laptops, touch-pad devices and phones. The application allows analyst to navigate through/and show the relationships between common properties of an attack to include campaigns, TTPs, Threat Actors, Targets, Indicators and Incidents, Observables and Course of Actions using 3D technology. The enterprise version will provision for a “worldly” view of a corporations resources and visually show the current state of systems under attack, consolidated into a HUD with drill-down/walk through capabilities. As users navigate through the system, they can create new “ticket” request to resolve problems using the BullsEye Ticket Management Service. Refer to Figure 2 to understand the relationships between the STIX objects.

### Student Requirements:

Number of Students: 1 student per design, if multiple students come up with different designs, each will have a separate project. The goal is to see what new 3D techniques work and which ones do not. The BullsEye team (all those involved in BullsEye projects) will help evaluate the different designs.

Level: Experienced C# .NET/Unity programmer. The student should have experience in C++, Java, UI/Web application development, Unity and programming; and be familiar with an Integrated Development Environment. Students should review the Department of Homeland Security (DHS) presentation “STIX Module 1 - Session 1 - STIX Introduction” for further details on the language format.

Goal: Learn how to program in a development environment which uses an Agile process to define the requirements, architecture, design, development and testing for key components of the core system. GitHub will be used as the source repository for all projects of BullsEye. The specific goal for the student is to learn C# .NET/Unity technology as well as 3D User Interface application development used to present Cyber Threat Information using presentation techniques that make it easy to understand and share that information. The student will become familiar with basic C# programming, creating a 3D .NET application while introducing new design concepts for information sharing.

Dependencies: This project is dependent on the STIX core project. The student may need to implement “stubs” used in STIX XML parsing or translation methods in order to implement their portion of the project.

## BullsEye UI – 3D Consumer

Windows 3D application (C# .NET and Unity SDK) used to navigate STIX messages and enumerate objects using a graphical representation, presenting the information current being displayed in stix-viz located at <https://github.com/STIXProject/stix-viz>, refer to figure 1. The UI can be hosted on multiple platforms that is supported by Unity, which provisions for servers, desktops, laptops, touch-pad devices and phones. The application allows analyst to navigate through/and show the relationships between common properties of an attack to include campaigns, TTPs, Threat Actors, Targets, Indicators and Incidents, Observables and Course of Actions using 3D technology. The consumer version will provision for “non-experts” and will be deployed in “mom and pop shops”, for example, a lawyer’s office or small business. The users will be able to view their resources and visually see the current state of systems under attack, consolidated into a HUD with drill-down/walk through capabilities that will work them through the correct course of action (COA) to take to correct the problem. It is a goal for this to become an automated process at some time. Refer to Figure 2 to understand the relationships between the STIX objects.

### Student Requirements:

Number of Students: 1 student per design, if multiple students come up with different designs, each will have a separate project. The goal is to see what new 3D techniques work and which ones do not. The BullsEye team (all those involved in BullsEye projects) will help evaluate the different designs.

Level: Experienced C# .NET/Unity programmer. The student should have experience in C++, Java, UI/Web application development, Unity and programming; and be familiar with an Integrated Development Environment. Students should review the Department of Homeland Security (DHS) presentation “STIX Module 1 - Session 1 - STIX Introduction” for further details on the language format.

Goal: Learn how to program in a development environment which uses an Agile process to define the requirements, architecture, design, development and testing for key components of the core system. GitHub will be used as the source repository for all projects of BullsEye. The specific goal for the student is to learn C# .NET/Unity technology as well as 3D User Interface application development used to present Cyber Threat Information using presentation techniques that make it easy to understand and share that information. The student will become familiar with basic C# programming, creating a 3D .NET application while introducing new design concepts for information sharing.

Dependencies: This project is dependent on the STIX core project. The student may need to implement “stubs” used in STIX XML parsing or translation methods in order to implement their portion of the project.

## BullsEye Asset Manager

Used to gather asset management information from monitored systems and store them in an asset management database. The database can be scanned, given current STIX vulnerability information and detect vulnerabilities within a protected environment. The project consist of an Asset Management Service (Windows Service) and agents which reside on monitored clients and relay asset information back to the AMS.

### Student Requirements:

Number of Students: 1 to 2, as student resources become available.

Level: Novice C# programmer. The student should have experience in C++, Java, network protocols and programming; and be familiar with an Integrated Development Environment. Students should review the Department of Homeland Security (DHS) presentation “STIX Module 1 - Session 1 - STIX Introduction” for further details on the language format. In addition, students interested in this area should also review the National Institute of Standards and Technology (NIST) formats for Security Content Automation Protocol (SCAP)/Asset Reporting Format (ARF) located [here](http://scap.nist.gov/specifications/arf/).

Goal: Learn how to program in a development environment which uses an Agile process to define the requirements, architecture, design, development and testing for key components of the core system. GitHub will be used as the source repository for all projects of BullsEye. The specific goal for the student is to learn C# and the technology specifically to collect asset information. The student will learn the APIs provided by Microsoft to enumerate the system database as well as scan for files, directories and registry key to uncover the presence of applications, devices and other assets, which will be recorded and retained by the Asset Management System. The student will become familiar with basic C# programming, creating libraries and implementing network communications protocols used in the sharing of asset information. They will also become familiar with standards used in asset reporting.

## BullsEye Ticket Manager

Used to manage the workflow from potential exploits recognized within the asset management information system and various UIs provisioned for BullsEye. The work flow is identified, prioritized and then tracked for completion. The tacking mechanism used the Asset Manager agents to report on the “state” of the applications defined in the work flow. For example, if there is an exploit on a version of an Adobe Reader and the Asset Manager determines that specific system under management are running that version, then a ticket is created for each system and the Ticket Manager will continue to request enumerations for those systems. Systems that are updated will automatically be determined and market as complete, other systems will start to have their ticket priority raised (and notifications/alarms sent) to indicate failure of compliance.

### Student Requirements:

Number of Students: 1 to 2, as student resources become available.

Level: Novice C# programmer. The student should have experience in C++, Java, network protocols and programming; and be familiar with an Integrated Development Environment.

Goal: Learn how to program in a development environment which uses an Agile process to define the requirements, architecture, design, development and testing for key components of the core system. GitHub will be used as the source repository for all projects of BullsEye. The specific goal for the student is to learn C# and the technology specific to Cyber Threat Information Sharing and work flow processing. The student will become familiar with basic C# programming, creating an application and using network protocols to track the progress of updates.

Dependencies: This project is dependent on the Asset Manager Service project. The student may need to implement “stubs” used to create asset events that would be used to simulate ticketing actions, as well as agent “stubs” used to return asset information on specific work flow monitoring processes.

## BullsEye Windows Mobile

Mobile application used to receive information from a STIX/TAXII aware notification system. Used to provide timely information on new exploits made known from the automated information sharing and analysis defined by the BullsEye Asset Manager, Automated Information Sharing server and the Tick Management system.

### Student Requirements:

Number of Students: 1 to 2, as student resources become available.

Level: Experienced mobile application programmer. The student should have experience in C++, Java, network protocols; and programming; and be familiar with an Integrated Development Environment.

Goal: Learn how to program in a development environment which uses an Agile process to define the requirements, architecture, design, development and testing for key components of the core system. GitHub will be used as the source repository for all projects of BullsEye. The specific goal for the student is to learn how to develop mobile applications in C# and the technology specific to Cyber Threat Information Sharing. The student will become familiar with Windows 8.1 Mobile application development using Hyper-V, as well as network communication of both RESTful and Cloud Services.

Dependencies: This project is dependent on a Cloud and RESTful service. The student may need to implement “stubs” used generate a Cloud Data source and a dummy RESTful service which provides “static” information for mobile queries.