Final Project Report

Cyber Threat Mission Builder	
Principal Investigator / Email Address	N/A
Project Team Lead	University of West Florida
Project Designation	20-11-10
MxD Contract Number	2021-08
Project Participants	University of West Florida Siemens
MxD Funding Value	N/A
Project Team Cost Share	N/A
Award Date	4/30/2021
Completion Date	4/30/2022

This project was completed under the Technology Investment Agreement W15QKN-19-3-0003, between Army Contracting Command – New Jersey and MxD. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Department of the Army.

TABLE OF CONTENTS

I. EXECUTIVE SUMMARY

The cyber threat mission builder seeks to assist industry members in the design of realistic cyber threat missions that their individual organizations might encounter. By designing realistic threat missions, organizations can develop a better understanding of the attack surface they expose to adversaries as well as the types of adversaries that might target them, the types of attacks, and the motivation behind the attacks. The cyber threat mission builder assists these individuals by walking them through the definition of their target environment and the stages of an adversary mission, along with specific tactics that adversaries use to accomplish the goals of those stages. This leads to a better understanding of threats, and a better ability to assess defenses. The level of detail captured in the cyber threat mission models makes it possible to produce mission scripts that can be used for training exercises such as incident response and penetration testing.

This project used Agile software development methodologies to produce a containerized solution suitable for cloud deployment. The system architecture uses a microservices back-end container that implements a REST API via the Django web application framework. The system also provides a GUI front-end that leverages Vue.js as a JavaScript framework to define the user experience. The use of Docker containers allows for a simple one-command deployment of the system, at which point the user can interact with the GUI using a web browser. Mission models are stored in a SQL database supported by the back-end container. The database is pre-populated with threat intelligence data from the MITRE ATT&CK enterprise framework. The techniques from MITRE ATT&CK are recommended to the system user at the appropriate stage in mission model definition. The system supports automated unit testing and regression testing via the Django test framework.

The result of the initial year of effort is a system that can be deployed using Docker's docker-compose utility to create and start containers for the required services. Users can interact with the target environment builder to create and update target environments that consist of resources and services that represent possible attack targets for an adversary. With a defined target environment, the user can then move on to the creation of a mission model. Using a drag-and-drop interface, the user can select techniques from the MITRE ATT&CK framework that an adversary might realistically employ to complete a cyber-attack. The completed mission is stored in a SQL database for future reference and modification.

Future recommendations include incorporating additional threat intelligence sources that are queried using programming APIs. The queries would be based on the resources and services defined in the target environment and would be used to improve recommendations of adversary techniques based on their relevance to the environment, mission type, adversary type, and mission stage. Additionally, the manual definition of target environments can be time consuming and error prone for sizeable organizations. For that reason, the ability to import data from network mapping tools would be a valuable feature to enhance adoption of the tool. The current system implements the Django admin feature for user account creation and user authentication but is not currently accessible as the system is still running in development mode. Additional containers would be useful to provide proper production deployment and allow for these features to be exposed. An option to export a mission model in a machinable format (such as JavaScript Object Notation) would also be beneficial for sharing of mission models and the creation of mission scripts for table top or simulated mission execution.

II. PROJECT DELIVERABLES

The following list includes all deliverables created through this project. These deliverables will be referenced throughout this report and should be accessible on the MxD membership portal in accordance with the rights defined by the Membership Agreement.

Table 1: MxD 20-11-10 Project Deliverables

#	DELIVERABLE NAME	DESCRIPTION	FORMAT OF DELIVERY
1	Final Technical Report	Report will include a comprehensive, cumulative, and substantive summary of all technical advancements and significant accomplishments achieved during the project.	PDF Document
2	Transition Plan	Written plan for successful transition of project outcomes after period of performance including distribution and follow-on efforts for phase(s) 2 & 3.	PDF Document
3	Final Technical Presentation/Demonstration	Presentation will include a comprehensive, cumulative, and substantive summary of all technical advancement and significant accomplishments	PDF Document and Video Demonstration
4	Educational Impact	Documentation on course/lab module presented to students demonstrating the use of technology to increase awareness in topic area.	PDF Document
5	GUI-based Mission Builder	Full source code for the mission builder frontend and database interface	Compressed Archive
6	VM Instances	Virtual machines providing installed and configured Mission Builder, Database, and test environment target systems	Open Virtual Appliance (OVA)
7	Collection of firmware	Any genuine and modified firmware updates developed to simulate low-level device attacks	Compressed Archive
8	Database for TTPs and CVEs	Scripts required to create a new instance of the backend database	Compressed Archive
9	Scripts	Scripts developed to automate events within the mission target environment	Compressed Archive
10	Key Performance Indicators	Report addressing final KPI values	PDF Document
11	Monthly Technical and Financial Reports	Reports addressing progress	PDF Document
12	Project Documentation	Users Guide and Administrators Guide	PDF Document
13	Project Assessment and Evaluation Plan	Test Plan and Test Results	PDF Document
14	Quarterly Technical Reviews	Reports addressing progress	PDF Document

III. PROJECT REVIEW

Use Cases & Problem Statement

The manufacturing industry faces different potential cyber-attacks beyond what traditional information technology encounters: supply chain tampering, part program manipulation, firmware corruption, device miscalibration. Existing mass-market tools are generic in nature, require in-depth cybersecurity awareness, and focus primarily on traditional IT infrastructure. These tools offer little support for the additional digital devices present in a manufacturing environment. This project seeks to bridge that gap by incorporating these non-traditional devices into a solution for modeling realistic cyber threat missions that might target the manufacturing sector.

The purpose of this project is to enable manufacturing industry partners to leverage real-world threat intelligence data in the creation of cyber threat scenarios that can be used to test and evaluate their environments. These scenarios take the form of cyber threat mission models that are sufficiently detailed to enable the simulation of a cyber-attack.

MxD Membership Use Case: As an Information Security Officer, I want assess my infrastructure so that I can defend against current and emerging threats

- Is my organization vulnerable to recently reported attacks?
- What changes can I make to better protect my organization?

Commercialized Solution Use Case: As an Information Security Officer, I want to model my infrastructure and build realistic cyber threat scenarios so that I can simulate a cyber-attack for assessment and training purposes

- Perform “what if” and “zero-day” simulations
- Evaluate security products and alternate configurations

Scope & Objectives

Provide the manufacturing industry with a tool for building realistic cyber threat missions which can be used for vulnerability assessments, penetration testing, security product evaluation, and security training.

Create a web-based GUI that guides the user through the construction of a cyber threat mission model, using each successive choice to provide better recommendations about how the mission might unfold. The mission model can then be used to produce a mission script that can simulate mission execution in a simulated environment.

Planned Benefits

Existing solutions use generic tools (such as Kali Linux and Metasploit) to perform vulnerability assessment and penetration testing, but the actions taken lack motivation and context with respect to the target environment. The goals of the adversary and the type of target need to be considered.

IV. TECHNICAL APPROACH

UWF's approach is to develop a mission builder based upon real-world threat intelligence feeds, CVEs, and intended adversary effects. The system is designed around two high-level concepts, a target environment and a mission builder. The target environment represents the resources available within an organization that is the target of a cyber-attack. As such, it defines the attack surface. The target environment acts as a catalog of resources available within the GUI that map to the "digital twin" Manufacturing Industrial Base (MIB) network. A Graphical User Interface (GUI) is provided for creating new target environments, adding resources to the environment, and adding services to those resources. The mission builder will also have a GUI where the adversary tactics from the MITRE ATT&CK framework are presented in a graphical manner. Based upon the defined resources, a database of adversary TTPs will populate options for each mission stage. The person building a threat mission model would then choose which TTPs will be used to accomplish which cyber threat mission stages.

The selected TTPs will become "events" within the threat mission model. Multiple "events" can be associate with a single cyber kill chain stage. Some stages may not have any events while some stages may have many. Events will be sequenced, indicating that one event will occur prior to another.

The anticipated workflow involves two main steps, the definition of the target environment (that will be attacked) and assigning events to mission steps/stages.

Threat Intelligence data will come from publicly available Open Source Intelligence (OSINT) sites. These include Open Threat Exchange (OTX) and VirusTotal which both offer Application Programming Interfaces (APIs) for automated queries. These threat feeds include IoCs associated with the attacks. Common Vulnerabilities and Exposures (CVEs) are freely distributed as a download in a variety of formats to include Comma Separated Values (CSVs) and Extensible Markup Language (XML). The CVE data identifies specific makes, models and versions of vulnerable equipment. We will also leverage the community-developed list of Common Weaknesses Enumerations (CWEs) that is also freely available in automation friendly formats.

The initial set of TTPs will come from the MITRE ATT&CK enterprise framework which offers programmatic access using Structured Threat Information Expression (STIX) and Trusted Automated Exchange of Intelligence Information (TAXII).

This collection of threats, IoCs, vulnerable systems, and TTPs will be processed and normalized through the backend of the Resource Builder GUI and stored in a SQL database (SQLite). The Resource Builder GUI is primarily intended as an interface to facilitate the data entry and annotation of resources.

The system GUIs were developed using the Vue.js JavaScript web application framework leveraging Django as a RESTful API providing a microservices backend.

Complete Project Start Activities

This initial task involved setting up project activities within the project management software used at UWF (Redmine). The project used Agile project management principles to schedule tasks using a ticket management system with priorities and anticipated due dates. Also, during this period, UWF advertised and filled two student research positions for one front-end developer and one back-end developer. UWF also initiated a request for software licenses from Siemens, our industry partner and cost-share provider. While these initial activities were

completed early on, turn-over in student research positions and changes in software licensing made this task something of an on-going effort.

Design phase, Preliminary Key Performance Indicators

The design phase for this task focused on the initial high-level design of the Cyber Threat Mission Builder. It involved creation of Use Cases and storyboards for workflow of the GUI. At this stage UWF also proposed an initial set of KPIs to use for project assessment purposes. This task finished with the documentation of requirements, use cases, GUI designs, concept diagrams, and initial KPIs in the Redmine project management system. The Redmine Wiki contains this information and is included in the Deliverables/Project Documentation/Redmine_Wiki folder. The Wiki can be viewed by opening the file Wiki.html in a web browser.

System Implementation phase; database design

The system implementation phase involved the use of Agile methodologists to create an initial functioning prototype that runs but with little functionality in place. This was supported by using a system design that separated the front-end web GUI implementation from the back-end microservice powered REST API. Vue.js was used to create a project for developing the GUI while Django was used to provide the REST framework. During this task, the design documents from the prior task were reviewed and initial data models were created and Entity Relationship Diagrams (ERD) produced to capture the database design.

Iterative development; unit and integration testing

The development strategy for the Cyber Threat Mission Builder involved incremental development and agile methodologies. As such, individual development tasks were identified and added to the work backlog in Redmine. Each week, developers on the project would claim tasks from the backlog based on their relevance and priority to the current sprint. Each task was given its own branch in GitHub and the developers would implement the described capability for the system, bench test it, and eventually submit a pull request (PR) after committing changes to GitHub. Another team member would pull their branch, build it, and test it to verify that it worked and did not break previous features. This was an ongoing task that ran to the completion of year 1, with the goal of implementing as many features as possible within the allotted time.

GUI based Mission Builder and Resource Builder

This task is predominantly an end goal driven target, where feedback from reviews drove choices in GUI design and User Experience (UX) choices. As the project evolved through iterative development, new features were implemented and tested, leading to tweaks to the UX and refactoring of the design. This in turn led to tweaks to the data model and database design to support those changes. Like iterative development, this is an ongoing task that completes only when the project ends.

Automation scripts and VM instances

Throughout the project, new automation scripts were developed for querying various sources of threat intelligence data, for pre-populating new database instances with such data (specifically the MITRE ATT&CK enterprise framework TTPs), and for setting up new virtual machines and Docker containers.

Simulation of firmware attacks

This task involved searching threat intelligence reports for instances of attacks against firmware that runs manufacturing and industrial control systems. Vulnerable firmware was identified and attempts made to obtain copies of those specific software releases. These were obtained to facilitate future efforts of simulating firmware attacks against manufacturing systems via the use of digital twins, physical devices, or virtual machines.

Educational modules; Training scenarios

The early prototype of the system was used to run a workshop for training educators on the creation of scenario-based learning for cybersecurity students. This “train the trainer” approach has a multiplicative effect on dissemination of the scenario design skills and use of the Cyber Threat Mission Builder approach to scenario development. During these workshops, participants generated realistic, threat intelligence based scenarios against hypothetical targets in various industry sectors.

Final Key Performance Indicators

Upon completion of the year 1 period of performance, UWF updated the values of the Key Performance Indicators initially identified with their final values for comparison against the project target values. This resulted in a final version of the KPI document for year 1.

Educational Impact Report

An educational impact report was prepared at the end of the project that captured not only the educational modules and training scenarios activities performed during year 1, but also included a copy of a conference paper that was accepted and presented at a conference that year (as a result of the work performed).

V. RESULTS

This section consists of the following subsections; system overview, system requirements, system architecture, features & attributes, target users & modes of operation, software development/design documentation.

Table 2: MxD 20-11-10 Technology Deliverables

#	DELIVERABLE NAME	DESCRIPTION	FORMAT OF DELIVERY
5	GUI-based Mission Builder	Full source code for the mission builder frontend and database interface	Compressed Archive
6	VM Instances	Virtual machines providing installed and configured Mission Builder, Database, and test environment target systems	Open Virtual Appliance (OVA)
7	Collection of firmware	Any genuine and modified firmware updates developed to simulate low-level device attacks	Compressed Archive
8	Database for TTPs and CVEs	Scripts required to create a new instance of the backend database	Compressed Archive
9	Scripts	Scripts developed to automate events within the mission target environment	Compressed Archive

System Overview

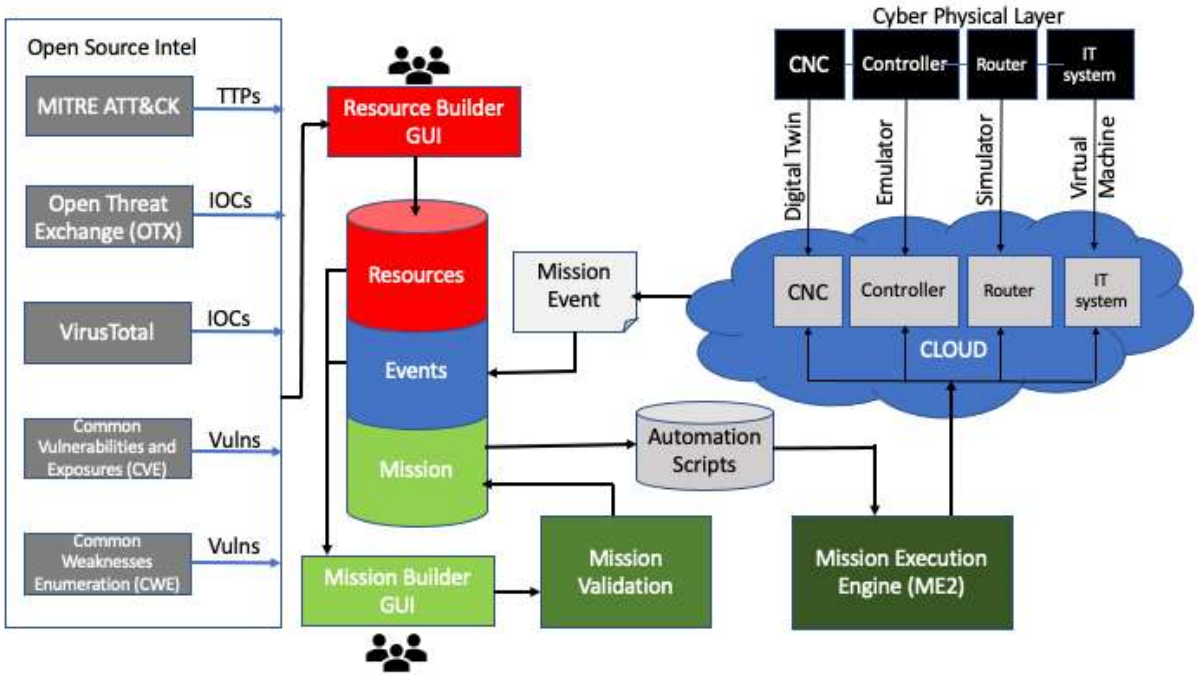
The Cyber Threat Mission Builder is a GUI-based Mission Builder that is accessible via a web browser. The Cyber Threat Mission Builder has features that allow for the definition of a target environment, which represents the attack surface for a cyber adversary. The system also features a Mission editor which allows the user to select an existing target environment and plan a cyber threat mission using techniques from the MITRE ATT&CK enterprise framework. In the creation of a mission model, the user is guided through all of the stages of the cyber threat mission and is presented with relevant techniques an adversary might employ to achieve the goal of that mission stage. Upon completion, the user will have a complete model of a targeted cyber-attack that leverages realistic techniques employed by adversaries. Proof-of-concept scripts were also developed that can query threat intelligence sources for information relevant to the resources and services in the target environment.

System Requirements

System requirements were driven by the need to be able to capture the resources and services present within an environment which may be the target of a cyber-attack. These environments need to support description of both traditional information technology assets as well as special purpose devices unique to manufacturing industries (either as physical, virtual, or digital twin representations). The system requirements were further influenced by the need to leverage the details of the environment against threat intelligence data and modern threat modeling frameworks (such as the MITRE ATT&CK frameworks). These requirements were subsequently documented on the Redmine Wiki and are accessible via the Wiki content located in the Project Documentation of the final deliverable.

System Architecture

Figure 1: MxD 20-11-10 Project Architecture



From a user perspective, the system offers two specific GUI builders, one for building target environments (labeled Resource Builder GUI in the figure but subsequently renamed), and one for building Missions. The target environment builder is used to capture details of the organizations cyber resources, including physical devices, virtual machines, digital twins (and other emulators), and simulators. The mission builder defines the events of a cyber threat mission, using TTPs obtained from MITRE ATT&CK.

The software architecture of the system uses a front-end that hosts a Vue.js-based web GUI application. The system has a back-end that provides persistent storage in a SQL database (SQLite at present) that is accessed using the REST framework provided by Django as a microservice. The front-end communicates with the back-end using REST endpoints (URLs) for performing various CRUD (Create, Read, Update, Delete) operations. Currently Django is hosting its services via its development server on port 8000 and the web GUI is accessible on port 8080.

Features & Attributes

The following features are currently available to the user of the Cyber Threat Mission Builder

- Deployment of a new system instance using Docker (docker-compose up)
- Create, Edit, Delete operations for target environments
- Create, Edit, Delete operations for resources assigned to a target environment
- Create, Edit, Delete operations for services running on resources
- Create, Edit, Delete operations for Missions
- Drag-and-drop assignment of TTPs to Mission stages
- Storage of mission events assigned to a Mission

- Pre-population of the database with MITRE ATT&CK TTPs

Target Users & Modes of Operation

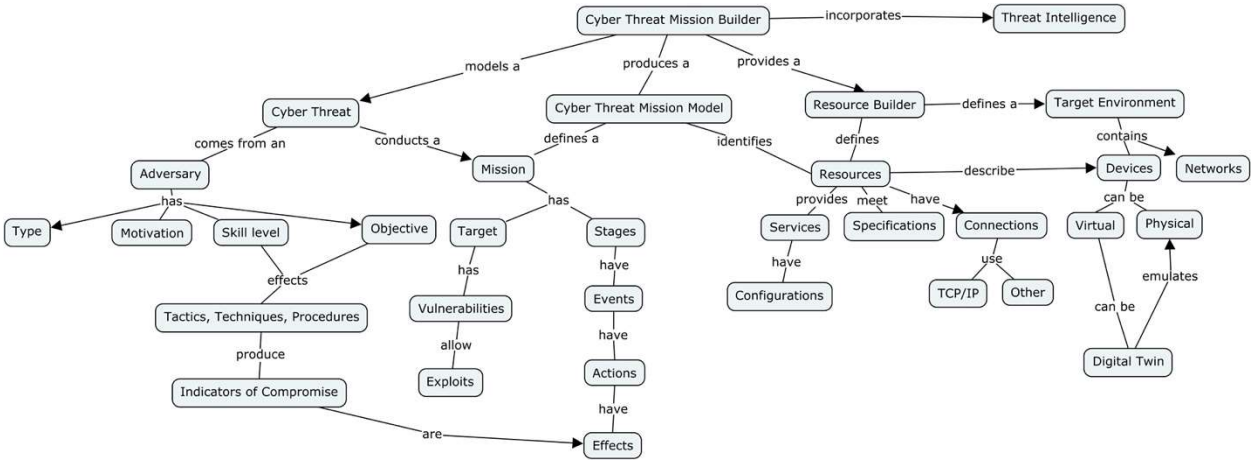
In its current state, the system does implement Django's user authentication feature but has not yet exposed the administrative console. As such, there is a default user account provided that is automatically used for all requested transactions. Future development should seek to properly deploy the Django back-end and Vue.js front-end (perhaps via Nginx), and expose the admin console for user management and authentication.

The expected users of this system range from cyber security trainers/educators, cyber competition designers, penetration testers, vulnerability assessors, cyber security officers, and risk managers. There are no different modes of operation at present. Everyone encounters the same user experience.

Software Development/Design Documentation

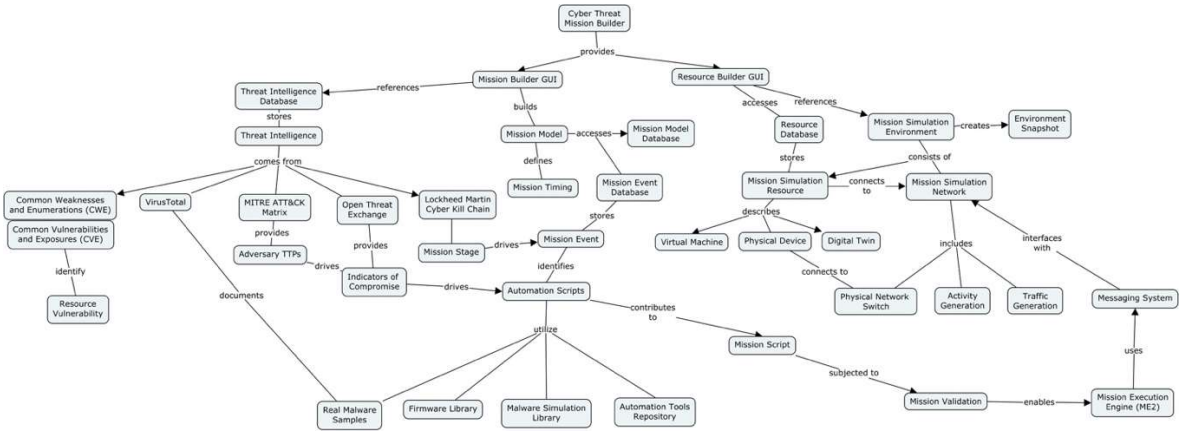
The concept map below seeks to explain what a Cyber Threat Mission builder is, what it does, and how it relates to various other concepts.

Figure 2: MxD 20-11-10 Concept Map – What is a Cyber Threat Mission Builder



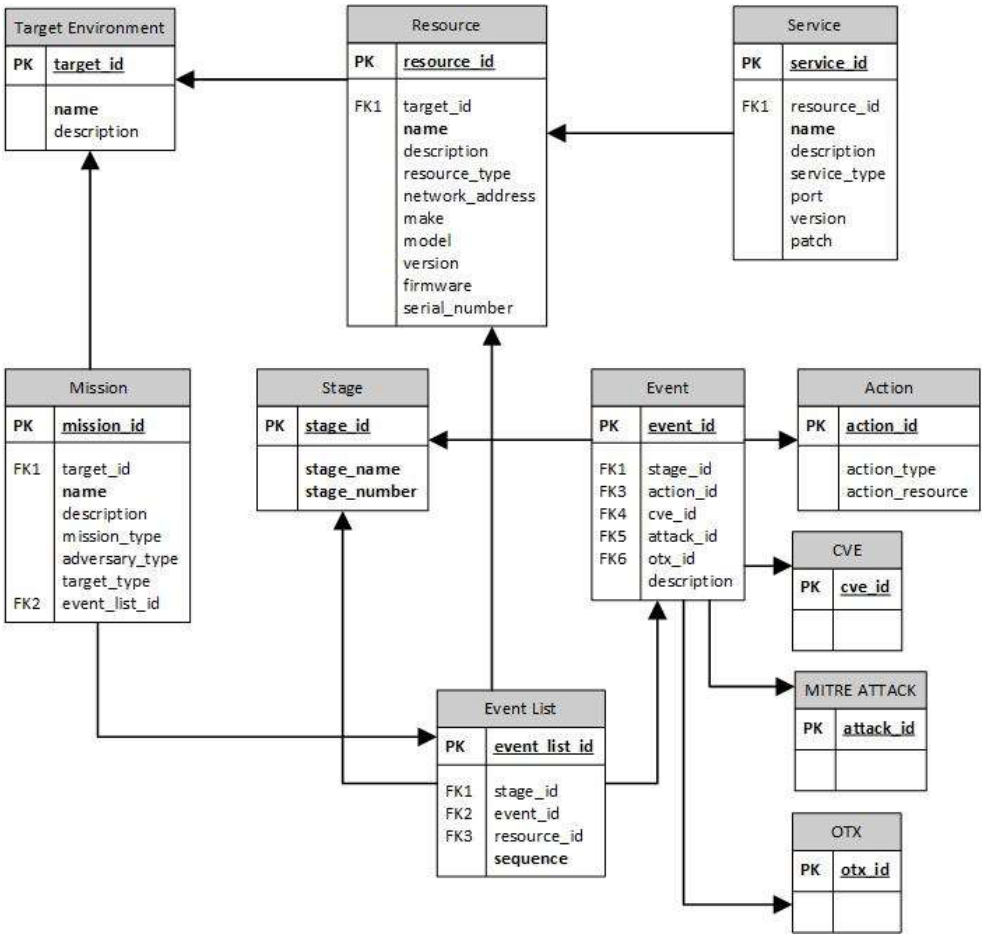
The following concept map seeks to explain the components of the Cyber Threat Mission Builder and the interactions and composition of those components.

Figure 3: MxD 20-11-10 Project Architecture – Components of a Cyber Threat Mission Builder



The Entity Relationship Diagram (ERD) depicted below seeks to capture the relationships of the various data tables in the SQL database model, including table names, primary and foreign keys, and table attributes.

Figure 4: MxD 20-11-10 Entity Relationship Diagram



VI. DISCUSSION & ANALYSIS

This section seeks to address industry impact of the project and the key performance indicators & metrics for the system. It continues on with an explanation of what is needed to access the technology developed, applications to workforce development, and lessons learned.

Industry Impact

The current state of the system allows for the end user to define an environment that might fall subject to a cyber-attack and to create a cyber-attack mission model that defines a sequence of events that could occur in the target environment on behalf of an adversary. The industry impact of this system is that it allows for the design of realistic cyber threat missions that could impact an organization based on its attack surface and the realistic techniques employed by cyber adversaries. Subsequent development efforts on the system would enable the export of the cyber threat mission models as a machine processable cyber threat mission script that can be automated and manually executed.

Key Performance Indicators & Metrics

Number of CVEs available

This value is a whole number that represents the total count of Common Vulnerabilities and Exploits that are available for use by the Cyber Threat Mission Builder. The success metric for this KPI is to make available the full database of known CVEs that were publicly disclosed at the time the project was proposed. Therefore, the success metric is to have 5,943 or more CVEs available to the system. Baseline is 0.

Number of CWEs available

This value is a whole number that represents the total count of Common Weakness Enumerations that are available for use by the Cyber Threat Mission Builder. The success metric for this KPI is to make available the full database of known CWEs that were publicly disclosed at the time the project was proposed. Therefore, the success metric is to have 922 or more CWEs available to the system (CWE list version 4.5). Baseline is 0.

Number of ATT&CK TTPs

This value is a whole number that represents the total count of MITRE ATT&CK Tactics, Techniques and Procedures that are available for use by the Cyber Threat Mission Builder. The success metric for this KPI is to make available the full database of TTPs that were listed in the MITRE ATT&CK framework at the time the project was proposed. Therefore, the success metric is to have 281 or more TTPs available to the system. Baseline is 0.

Number of IoCs

This value is a whole number that represents the total count of different Indicators of Compromise that can be included in Cyber Threat Mission models produced by the Cyber Threat Mission Builder. The primary source for IoCs will be the MITRE ATT&CK framework. A secondary source will by the Open Threat eXchange. The success metric for this KPI is to have a minimum of 50 different IoCs to choose from when building a Cyber Threat Mission Model. Baseline is 0.

Number of CAPECs referenced

This value is a whole number that represents the count of different attack patterns from the Common Attack Pattern Enumeration and Classification (CAPEC) list that can be referenced in

a Cyber Threat Mission Model. The success metric for this KPI is to have a minimum of 50 different CAPEC entries to choose from when building a Cyber Threat Mission Model. Baseline is 0.

Number of automated events

This value is a whole number that represents the count of events that have automation that can produce their associated Indicator(s) of Compromise (IoCs). The success metric for this KPI is to have a minimum of 50 different automated events that can be included in Cyber Threat Mission models. Baseline is 0.

Number of missions that can be generated

This value is a whole number that represents the count of the total number of distinct Cyber Threat Mission models the Cyber Threat Mission Builder can create. The success metric for this KPI is to have a minimum of 50 distinct Cyber Threat Missions be possible via different configurations of mission events. Baseline is 0.

Number of firmware images

This value is a whole number that represents the count of firmware images that have been amassed for use in Cyber Threat Mission models. The success metric for this KPI is to have a minimum of 5 firmware images available to the Cyber Threat Mission Builder. Baseline is 0.

Number of automation scripts developed and tested

This value is a whole number that represents the count of automation scripts that have been produced to support the simulation of activities taking place on resources within the target environment associated with the Cyber Threat Mission model. The success metric for this KPI is to have a minimum of 5 automation scripts. Baseline is 0.

Quality of GUI-based Mission Builder

This KPI is a qualitative assessment of the GUI for the Cyber Threat Mission Builder that is based upon the results of conducting unit, integration, and regression tests. The value will be a percentile score (0 to 100) that reflects the number of passing tests compared to the total number of tests. Passing criteria for this metric will be a percentile score of 90 or higher.

Number of unique VM systems

This value is a whole number that represents the count of unique virtual machines developed and used for testing mission models produced by the Cyber Threat Mission Builder. The success metric for this KPI is to have 20 or more unique VMs available within the target environment that can be used as resources in the mission model.

Number of research papers published

This value is a whole number that represents the count of published research papers associated with the work performed on the Cyber Threat Mission Builder. The success metric for this KPI is to have at least one research paper submitted to a conference, journal, or as a book chapter prior to the end of the period-of-performance. Baseline is 0.

Number of educational models and training scenarios developed

This value is a whole number that represents the count of educational models and training scenarios that were developed in conjunction with the Cyber Threat Mission Builder. The

success metric for this KPI is to have a combined total of 5 or more educational models and training scenarios.

Table 3: KPI's and Metrics

METRIC	BASELINE	GOAL	RESULTS	VALIDATION METHOD
Number of CVEs available	0	5943	5943	Counting / Value Comparison
Number of CWEs available	0	922	922	Counting / Value Comparison
Number of ATT&CK TTPs	0	281	281	Counting / Value Comparison
Number of IoCs	0	50	>50	Counting / Value Comparison
Number of CAPECs referenced	0	50	0	Counting / Value Comparison
Number of automated events	0	50	0	Counting / Value Comparison
Number of missions that can be generated	0	50	>50	Counting / Value Comparison
Number of firmware images	0	5	3	Counting / Value Comparison
Number of automation scripts developed and tested	0	5	11	Counting / Value Comparison
Quality of GUI-based Mission Builder	0%	90% passing	100% passing	Ratio of passing unit tests to total unit tests
Number of unique VM systems	0	20	4	Counting / Value Comparison
Number of research papers published	0	1	1	Counting / Value Comparison
Number of educational models and training scenarios developed	0	5	6	Counting / Value Comparison

Accessing the Technology

a. What Background Intellectual Property (software, designs, data, etc.) is needed to use or modify the technology outcomes?
    a. Future software development can be performed using a text editor, Docker, and a web browser (or) using a Linux machine (virtual or physical) with the required development libraries in place. An Open Virtualization Archive (OVA) file of a development virtual machine is provided in the deliverables.
    b. Future changes to design documents may require use of Microsoft Visio Professional or Microsoft PowerPoint. Modifications to Concept Map files (cmap) will require the CMAP Tools product available from the Institute for Human and Machine Cognition (IHMC) - https://www.ihmc.us/cmaptools/
b. What are the technical and systems requirements for using the technology outcomes? (e.g. What equipment is required for use? What software or hardware specifications are required? What enterprise systems are needed? What data is needed- types of data sources, format of data, historical data, etc.?)
    a. Technical system requires at present are a host computer capable of running Docker and that has Internet access (to obtain the base Docker image). The project has been tested with Docker version 20.10.13 build a224086 and Docker Desktop version 4.6.1 (76265).
c. What expertise is needed to implement the technology?
    a. Python, Virtualenv, Django, REST, SQL, JavaScript, Vue.js, npm, Docker

d. If the technology will be distributed through channels other the MxD Member Portal (e.g. through a local MEP or commercialization effort), please explain how MxD members can access the technology.
    a. No additional distribution channels are planned at this stage.

Workforce Development

A faculty development workshop on scenario-building utilizing the cyber threat mission builder was conducted in January 2022. It involved six faculty participants from six other academic institutions. Participants engaged in a tabletop exercise to design and implement cybersecurity scenarios delineated by the stages of the cyber kill chain. For each stage, a tactic with an associated technique was selected from the MITRE ATT&CK matrix. Each technique was further elaborated with a corresponding procedure supported by Indicators of Compromise (IoC). The principal takeaway from the workshop is the acquired knowledge on developing realistic cybersecurity scenarios to enhance the learning process through critical thinking and hands-on exercises. Two other workshops are scheduled in June and August 2022.

Lessons Learned

a. What were the major technical lessons learned throughout the period of performance?
    a. Technical approach using Django and Vue.js was effective
    b. Microservices and REST framework lend the solution to containerization and scaling in the cloud
    c. Vue.js offers powerful out-of-the-box GUI elements
    d. Github, Redmine, and PRs are vital to progress
    e. Back-end code to query all threat intelligence APIs is functional as proof-of-concept
b. Describe the problems (technical and programmatic) that were encountered and how they were solved?
    a. Access to cloud services for prototyping is highly limited in an academic environment
    b. Simulation environment was one VMware-powered rack mounted server on premises (on loan, now returned)
    c. No centralized database server for the back-end hindered later stage development progress
c. How did the outcomes deviate from what was proposed in the Proposal/Statement of Work? If different, explain.
    a. The events recommended to the user of the cyber threat mission builder are currently limited to filtering the MITRE ATT&CK techniques down to those relevant to the mission stage. Subsequent work would also query CVE databases using information from the target environment to provide better recommendations. Incorporating the type of adversary mission into the recommender system would also assist users in designing more realistic missions without the user having to have that level of knowledge about tactics
d. Were the project goals attained? If not, what changes would help meet the goals in the future?
    a. While the system does integrate threat intelligence data by way of the MITRE ATT&CK framework, incorporating additional sources of data were originally in the plans. A limiting factor in development was turnover in student research assistants and the entry-level experience of the student population.

e. What risks were realized throughout this project? What recommendations would you give to those doing follow-on research to mitigate those risks?
    a. Research assistant turnover. No solution. Students will come and go.
    b. Access to resources. Many dev resources, such as cloud access and network accessible database servers, require subscriptions that are no sustainable or available to academics. Ideally seek these resources via a cost-share from an industry partner.
f. What else would the team have done differently to improve the results?
    a. Integration of Docker into the development workflow earlier on would have simplified deployment to new student research assistants and minimized the need for full virtual machines
    b. Rigid adherence to the Github workflow, maintaining a main and dev branch with individual branches for all assigned tasks prevents issues with merges. Students had a tendency to create a branch for a task and never ever merge it which led to code drift. PRs are vital.

VII. CONCLUSIONS & FUTURE WORK

This research demonstrated the technological feasibility and validity of the idea for a threat intelligence driven cyber threat mission builder. The results of the research are an initial version of a system that prompts the users to consider the resources in their environment that might be targeted in a cyber-attack, the types of adversaries likely to perform an attack and the motivation of the adversary, and subsequently to use that context to select techniques that might be employed by an adversary to build a model of a realistic cyber threat mission. The very process of building a model can lead to insights about an organizations threat exposure.

Proposed follow on work for a second year focuses on integrating more threat intelligence data in the recommendation of adversary techniques, automated data entry for target environments that are based upon real-world organizations (such as data provided by network mapping tools such as NMAP), and enhanced use of containerization to facilitate cloud deployment of the system.

Next Steps & Challenges

The system needs to be modified before deployment into a production environment. Specifically, the current state of the system uses development servers that are not suitable for real-world deployment as they lack security controls and do not operate at scale. The recommended approach would be to incorporate an Nginx server to host the Django and Vue.js (npm) servers properly.

The strategy for distributing the project outcomes is to use the MxD Membership Portal.

Transition Plan

The table below provides a catalog of all of the project deliverables and their respective transition routes. Deliverables can transition through deployment at an industry member's site, as an educational reference or through a commercialization effort. Each of these transition routes are detailed below.

Table 4: Deliverable Deployment Summary

#	DELIVERABLE FILE NAME	TECHNOLOGY INTEGRATION	EDUCATION	COMMERCIALIZE
1	GUI-based Mission Builder	*X*		
2	VM Instances	*X*		
3	Collection of firmware	*X*		
4	Database for TTPs and CVEs	*X*		
5	Scripts	*X*		

GUI-based Mission Builder

The GUI-based Mission Builder is available as a compressed archive containing all required project files needed to build Docker containers hosting the Cyber Threat Mission Builder. This archive is available in the Deliverables folder and will likely be made available via the MxD Member Portal.

VM Instances

There is an Open Virtualization Archive (OVA) in the Deliverables folder for the project. That OVA can be instantiated on most popular virtualization products (VMware Workstation, VMware VCenter, Oracle VirtualBox). This machine is an instance of an Ubuntu 20.04 Workstation pre-

configured for development of Django and Vue.js projects. This archive is available in the Deliverables folder and will likely be made available via the MxD Member Portal.

Collection of firmware

The firmware samples collected to support simulation of firmware attacks within a target environment are located in the Deliverables folder of the project. This archive is available in the Deliverables folder and will likely be made available via the MxD Member Portal.

Database for TTPs and CVEs

The database design documents, scripts for prepopulating database tables, and an example SQLite database instance are in the Deliverables folder for this project. This archive is available in the Deliverables folder and will likely be made available via the MxD Member Portal.

Scripts

The automation scripts for this project are available in the Deliverables folder for this project. They are also included in the GUI-based Mission Builder archive. This archive is available in the Deliverables folder and will likely be made available via the MxD Member Portal.

VIII. APPENDICES

Appendix A: Definitions

N/A

Appendix B: Demos

Demonstrations are included in the Deliverables folder within Final Technical Presentation/Demonstration and Quarterly Technical Reviews

Appendix C: Validation & Testing

The project test plan is included in the Deliverables folder within Project Assessment and Evaluation Plan

Appendix D: User Resources

The User and Administrator Guide is included in the Deliverables within Project Documentation