Simulated Cyber Analyst for Network Vulnerability Assessment

Ning Wang, Abhilash Karpurapu
University of Southern California
Los Angeles, CA
{nwang, akarpurapu}@ict.usc.edu

Eric W. Holder
US Army Research Laboratory
Ft. Huachuca, AZ
eric.w.holder4.civ@army.mil

ABSTRACT

Artificial intelligence (AI) is playing an increasingly important role in the Combatant Command Cyber Protection Team's (CCMD CPT) planning process. With petabytes of past cyber incident data available, AI can be a useful tool to understand the complex relationships within system components, vulnerabilities, threats, and implications on future missions. Because such AI often works alongside human cyber operators to support mission commanders in decision-making, the understanding of the AI's decisions and the rationale behind such decisions can be key to the success of this human-AI team. An analyst or operator often needs to explain analysis by the AI to a commander when recommending courses of action. It is critical to make core decision factors, assumptions, uncertainties, and the variables that drove the analysis accessible to the human.

In this project, we designed a simulated cyber analyst to advise mission planners on the target network systems in terms of what has happened (incidents, vulnerabilities, threat presence), likely follow-on adversary activities, and where to monitor, harden, or counteract those activities. We synthesized a dataset that includes incident reports on past attacks on military networks. AI techniques of varying explainability were applied to analyze the dataset to determine vulnerabilities of a set of simulated target networks. We then developed automatically generated explanations for the AI techniques to explain the policies and how such policies are learned. Such explanations were fed into the simulated cyber analyst to justify its vulnerability analysis and recommendations on a course of action. The cyber analysis is placed in an experimental testbed with simulated target networks to study how such explanations impact human-automation team performance. In this paper, we will discuss our research into simulating cyber incident data, the AI techniques applied, and the automatically generated transparency communication for the simulated cyber analyst testbed.

ABOUT THE AUTHORS

Ning Wang, Ph.D. is a research associate professor of computer science of University of Southern California. Dr. Wang's research focuses on explainable AI for human-machine teaming, AI for education, and AI education for youth and the general public.

Abhilash Karpurapu, M.S. is currently a software development engineer at Amazon. Mr. Karpurapu was a student AI researcher at the Institute for Creative Technologies at the University of Southern California (USC). He received his Master of Science degree in Computer Science from USC.

Eric W. Holder, Ph.D. is a Research Psychologist of the Human Research & Engineering Directorate of the US Army Research Laboratory in Ft. Huachuca. Dr. Holder's research focuses on conducting both Human Systems Integration testing on tools and products, as well as applied research to define requirements and the conditions and context of use for future systems.

Simulated Cyber Analyst for Network Vulnerability Assessment

Ning Wang, Abhilash Karpurapu
University of Southern California
Los Angeles, CA
{nwang, akarpurapu}@ict.usc.edu

Eric W. Holder
US Army Research Laboratory
Ft. Huachuca, AZ
eric.w.holder4.civ@army.mil

ARTIFICIAL INTELLIGENCE IN NETWORK VULNERABILITY ASSESSMENT

As the DoD pushes towards the unified platform to conduct joint cyber operations, there is a growing need to ensure that cyber operators from across the services can work together to successfully execute missions. This will require effective communication and collaboration predicated upon a shared understanding of the cyber environment. This requirement is further complicated by the fact that the future operating environment will likely include input and output from AI-driven intelligent agents, who will play a key role in providing information and visualization support to the combatant command (CCMD) cyber protection teams (CPTs), mission owners, and network owners for the identification of cyber key terrain (KT-C) and mission relevant terrain-cyber (MRT-C) and the planning and tasks related to protecting and utilizing this key terrain (HQ USINDOPACOM 2019; Raymond et al. 2014).

One of the reasons to include AI in this workflow is its ability to analyze large amounts of data (O'Leary 2013). In fact, there exist massive amounts of data in cyber terrain stockpiled that may contain valuable insights, but CPTs lack the manpower and pressing priority to force analysis (Holder & Wang, 2021). One of these data sources was cyber incident data (e.g., via the Joint Incident Management System (JIMS) and other systems), some of which have been investigated fully and others not. The incident reports are available in large databases but would need to be related to system components, threats, and vulnerabilities to be useful, along with the requirement to understand the analysis and its implications and assumptions in order to create a plan and brief it. There are also databases of known exploits and vulnerabilities, such as the common vulnerabilities and exposures (CVE) database (MITRE, 2022b), Nessus, or the Air Force's Genesis tool. Threat models that include attack patterns and tactics, techniques and procedures (TTPs), and the signatures left behind, such as those based on the MITRE ATT&CK model (MITRE, 2022a), are also available. AI can play a role in the CCMD CPT planning process as part of the toolkit to support the planning process by mining the cyber incidents data and combining them with other relevant data sources, such as known exploits and vulnerabilities and cyber-attack models.

At the CCMD level, there will be a large number of networks within their area of operations that support CCMD missions to various degrees. When assigned a mission, the cyber protection team (CPT) planner first identifies the networks and systems involved in that mission, the KT-C or MRT-C. These networks, systems, and components represent the base of the workflow for a junior cyber analyst agent. Thus, the output requirement for the cyber terrain AI can be drawn from the typical workflow between junior and senior cyber analysis and how those outputs can be used (Holder & Wang, 2021). Interviews with instructors at the Intelligence Center of Excellence have highlighted the need for explainable outcomes and the concept of keeping the human as the senior analyst and using AI as the junior analyst agent to do the legwork and produce usable outputs to the senior analyst. The human senior analyst then uses the results to iteratively ask clarifying questions of the junior analyst, and to brief further up and down the chain of command. The senior analyst is going to have to explain and field questions on the results and suggestions he or she is making to the commander and will need to be able to provide those justifications and supporting details. This means that this applied use case of AI needs to be combined with explainability or combined with explainable AI (XAI) methods to explain the results and assumptions of the AI analysis and get the planner up to speed on the target system in terms of what has happened (incidents, vulnerabilities, threat presence), likely follow-on adversary activities and where to monitor, harden, or counteract those activities.

An XAI-driven junior cyber analyst agent could be used to allow the human operator to identify the systems and components of interest and to present relevant data concerning exploits and vulnerabilities and the incident reports from the target system and similar systems/components to look for signatures that identify past attacks, likely threat actors present, and next steps in attacks (adversary courses of action), along with the logic of these recommendations and additional recommendations on required actions (hardening, sensors or monitoring). AI should be able to "see" much more in terms of patterns of connections between large databases (e.g., incidents-JIMS, threat-ATT&CK, and vulnerabilities-CVE) than a human operator and do this much faster. This will allow the senior analyst to plan what the CPT mission team should do next to either protect the mission, gather more information (sensors, from network owner, etc.), or facilitate discussions with the mission owner and network owners.

The goal of the project is to study how transparency communication provided by AI can impact human-AI teaming in cyber operations. To achieve this goal, we created a dataset that simulates past attacks on military networks. We then applied AI techniques to analyze the dataset to determine vulnerabilities of a set of simulated target networks. Next, we designed explanation algorithms based on the AI technique to explain the policies and how such policies are learned. Finally, a testbed is designed for experimentation of how such explanations impact human-automation team performance. This paper will explain the development of this use case and the information requirements for an AI-driven junior cyber analyst.

EXPLAINABLE AI

Artificial intelligence (AI) is at the core of the future of Army technology. The US Department of Defense (DoD) is investing $2 billion to create human-like AI to be the Soldiers' "partners in problem-solving" (Walker 2018). Cyber security analysis tools will be powered by state-of-the-art AI. Such AI will work alongside the cyber operators on and off the battlefield. For many future use cases, including cyber operations, the understanding of the decisions of the AI and the rationale behind such decisions can be key to the success of the man-machine team. However, the complexity and the "black-box" nature of many AI algorithms create a barrier for establishing such understanding within their human counterparts. Without such understanding, a human interacting with such an AI system is likely to fall into the pitfall of misuse or disuse of the automation (Parasuraman and Riley 1997). For an AI to play an effective role in many human-machine teams, it must make its critical decisions understood by its human counterparts.

Early work in explainable AI (XAI) focused on generating explanations of expert decisions within rule-based and logic-based AI systems, not addressing the quantitative nature of much of the AI used today (Swartout and Moore 1993; Johnson 1994; van Lent et al. 2004; Core et al. 2006). More recent work on agent-based XAI used Markov decision processes (MDPs), the completely observable subclass of partially observable MDPs (POMDPs) (Elizalde et al. 2008; Dodson et al. 2011; Khan et al. 2011) and later for POMDPS (Pynadath et al. 2016). More recently, as machine learning (ML) systems become more prevalent in our everyday life, there has been a surge of research into making their decision-making more transparent (Ribeiro et al. 2016; Hendricks et al. 2016; Guo et al. 2018). Some of these efforts have taken the approach of incorporating human-interpretable models, such as AND-OR trees (Si and Zhu 2013), Bayesian networks (Shih et al. 2018), or decision-trees (Pynadath, et al., 2018, 2022) into the ML process. To address the needs to make AI more transparent, DARPA created an explainable AI program in 2016. In a review of the program mid-way through its 4-year course, the program director, Dave Gunning, has pointed out that state-of-the-art XAI research has enjoyed success in explaining the decisions behind, for example, recommendation systems for image recognition (Hendricks et al. 2016; Chang et al. 2018) and AI playing video games (Koul et al. 2018). However, generating explanations for automation with ML will remain a significant challenge for years to come, because of the fundamentally higher level of complexity of the AI decisions for automation relative to those for simpler recommendation systems (Gunning 2019). An AI-driven cyber analyst agent, as a decision-support automation, still presents a challenge to making its decision process transparent, given the state-of-the-art XAI research.

JUNIOR CYBER ANALYST

As networks of hosts continue to grow in both size and criticality to operations, evaluating their vulnerability to attacks become increasingly more important to automate. The human-AI interaction in the cyber application described here centers around the vulnerability and threat analysis based on the MITRE ATT&CK model (MITRE, 2022a). Specifically, the research utilizes AI methods to analyze the past cyber incident reports for the target network and similar networks based on the publicly known adversary tactics and techniques described in the MITRE ATT&CK repository and known vulnerabilities. The ATT&CK model lays out the steps of cyber-attacks into a kill chain (reconnaissance, resource development, initial access, execution, persistence, privilege escalation, defense evasion, credential access, discovery, lateral movement, collection, command and control, exfiltration, impact) with known techniques that are used at each step of the chain to accomplish that goal. For threats that are known, primarily advanced persistent threats (APTs), the model provides patterns of the techniques used by each specific threat at each step based on their history of attacks. A tactic is a behavior that supports a strategic goal; a technique is a possible method of executing a tactic. Each technique is performed through various procedures. A sequence of techniques from different tactics used for an attack is called a TTP (tactics, techniques, procedures) chain. The combination of MITRE ATT&CK techniques in a TTP chain represents various attack scenarios that can be composed in an attack graph (Jha et al. 2002).

The combination of MITRE ATT&CK techniques in a TTP chain represented in an attack graph captures the life cycle of an attack. Lockheed Martin first described the cyber-attack life cycle as the cyber kill chain that composes of seven stages: reconnaissance, weaponize, deliver, exploit, control, execute, and maintain (Hutchins et al. 2011). The tactics in ATT&CK follow this life cycle as well. An attack sequence would involve at least one technique per tactic, and a completed (post-exploit) attack sequence would be built by moving from left (i.e., initial access) of the ATT&CK matrix to right (i.e., command and control). It is possible for multiple techniques to be used for one tactic. For example, a well-known attack group APT28 might try both a spear phishing attachment and spear phishing link techniques as initial access tactics. It is not necessary for an attacker to use all post-exploit tactics. Rather, the attacker will likely use the minimum number of tactics to achieve the objective, as it is more efficient and provides less chance of discovery. For example, APT28 may perform initial access to the credentials of an administrative assistant using a spear phishing link technique delivered through an email. Once they have the admin's credentials, APT28 can look for documents through file and directory discovery in the discovery stage. If the data APT28 is after is in a folder to which the admin also has access, then there is no need to go through the privilege escalation phase. In the end, APT28 could use various techniques in the collection phase, such as data from the local system, to download files to their own machine.

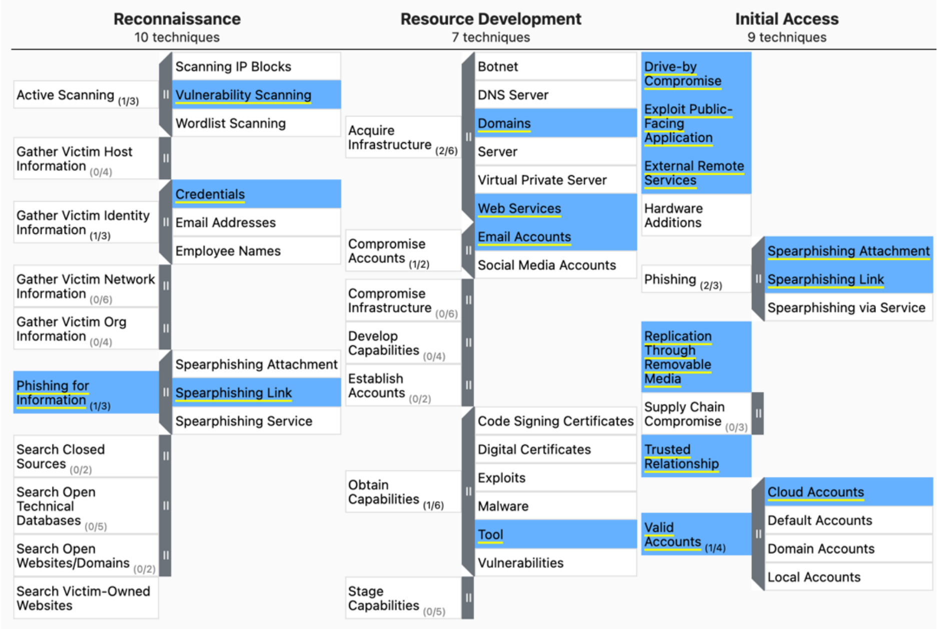
Attack graphs can serve as a basis for detection, defense, and forensic analysis. All systems will exhibit at least some vulnerabilities, and many of these are known and/or discovered and reported. When evaluating the security of a network, it is not enough to consider the presence or absence of isolated vulnerabilities. A large network builds upon multiple platforms and diverse software packages and supports several modes of connectivity. The assessment of the vulnerability of a network of hosts should consider the effects of interactions of local vulnerabilities and include global vulnerabilities introduced by interconnections. Scanning tools can determine the vulnerabilities of individual hosts. An attack graph can express the local vulnerability information along with other information about the network, such as connectivity between hosts. Each path in an attack graph is a series of exploits that lead to an undesirable state (e.g., a state where an intruder has obtained administrative access to a critical host). An attack graph is a succinct representation of all paths through a system that end in a state where an intruder has successfully achieved his goal. The ATT&CK dataset can be used to construct attack graphs that are associated with different known threat groups to represent their standard TTPs. The idea was to analyze the patterns presented in a TTP chain to identify likely adversaries, their courses of action (COAs) on a network or system of interest, and the vulnerabilities they exploit. This also would help support incident response actions (what to harden to prevent further threat actions) and attribution of attacks and incidents.

Simulated Dataset

The cyber incident report data can represent different scenarios in the attack graphs built using the techniques from the ATT&CK model. While the US military has a stockpile of reports and data on cyber incidents, we choose to not use the real incident reports dataset for the research of this project, due to the security restrictions and the sensitive nature of such data. Alternatively, we created a simulated dataset based on the structure of the cyber incident report used in the US military. Each incident report describes the technical specification of the target system, techniques used by the adversary, detection methods, impact on the target system and the mission, and other technical and nontechnical information related to the incident. For example, some of the non-technical information includes military categorization of the attack, date and time of incident and reporting, and impact on the operational unit and the mission. The technical details of the incident are simulated using the techniques from different tactics from the ATT&CK repository. The ATT&CK repository organizes the techniques both by tactics, each corresponding to a stage of the attack, and by attack groups. It describes the known attack groups, techniques used by those groups, common software that implements those techniques, and high-level discussions of the vulnerabilities explored, impact of the attack techniques, and remedies. For example, the "technique, tool, or exploit used" by the adversary can map onto the techniques, such as power shell and network sniffing, described in the MITRE ATT&CK repository. The "root causes", "method of detection", and "mitigation strategies" can be simulated using the "software", "detection", and "mitigations" subfield in the description of a technique in the ATT&CK matrix. To delve into details of the vulnerabilities, we connected the vulnerability descriptions in the ATT&CK database with the CVE database (MITRE, 2022b), which includes technical details of the vulnerabilities that are impacted by an attack, by different attack groups using different techniques. We have compiled all the data containing cyber-attack Techniques and Sub-techniques from different Tactics used by all the groups described in the ATT&CK set. Additional technical details describing the Techniques and Sub-techniques are fetched from the CVE. In the simulated cyber incident dataset, each incident is entered as a row of in the database, which includes timestamps, techniques, sub-techniques, software tools used in the attack, the vulnerabilities exploited, and impact on the mission. For an example, a row in the dataset would appear similar to the following:

Reporting Incident Number: AXHER98234923
OS: MacOS
Date Reported: 09/01/2020
Status: OPEN
Target IP: 172.20.1.0
Intruder: Blue Mockingbird
Technique, Tool or Exploit used-
Reconnaissance: Null
Resource Development: Null
Initial Access: Exploit public facing application
Execution: Command and scripting Interpreter
Persistence: Hijack execution flow
Privilege Escalation: Access token manipulation
Defense Evasion: Hijack execution flow
Credential Access: OS Credential dumping
Discovery: System Information Discovery
Lateral Movement: Remote Services
Collection: Null
Command and Control: Proxy
Exfiltration: Null
Impact: Resource Hijacking
Technical Impact:
attackVector': 'NETWORK', 'attackComplexity': 'LOW', 'privilegesRequired': 'NONE', 'userInteraction': 'REQUIRED', 'scope': 'CHANGED', 'confidentialityImpact': 'LOW', 'integrityImpact': 'LOW', 'availabilityImpact': 'NONE', 'baseScore': 6.1, 'baseSeverity': 'MEDIUM', 'exploitabilityScore': 2.8, 'impactScore': 2.7,
Vulnerability Information: 'The google-analyticator plugin before 5.2.1 for WordPress has insufficient HTML sanitization for Google Analytics API text.'
Mission Impact: 12 Staff hours required to address the vulnerabilities on this node

In a real cyber-attack scenario, unless a group publicly claims responsibility for the attacks, it is often uncertain which group is behind the attack. To simulate such uncertainty, we added "noise" to the simulated data. Each group (e.g., a group described in the "Suspected Group" field) uses certain kinds of techniques for different tactics (described in MITRE ATT&CK) to orchestrate an attack. The absence and presence of certain techniques for the series of tactics used can thus be considered the "signature" of a specific attack group. For example, a well-known attack group APT28 often uses "phishing for information", "vulnerability scanning", "credentials" and "spearfishing link" techniques for reconnaissance, but not "active scanning", "searching for closed resources" etc. And the group uses "domains", "web services", and "email accounts" for "resource development" (Figure 1). We added noise to our simulated dataset to make identification of such a "signature" ambiguous. For example, adding random techniques to the certain tactics the group that are not observed before in the attack history of a certain attack group, or removing some techniques from certain tactics that are usually observed in the attack history of a certain attack group. The intuition behind this approach is that an attack group might use different-than-usual techniques in future attacks. By modifying its attacking signature, we will only be able to identify an attack group with some level of uncertainty. So, some of the techniques from the example above may be randomly removed, or some of the techniques from the above example can be replaced with new randomly selected ones, to add noise to the simulated data. However, other information, such as the vulnerabilities, mission impact, etc., remain unchanged even though noise is added. The simulated dataset contains the attack's features of all groups found on MITRE ATT&CK repository, and the vulnerabilities associated with each attack, which helps in gauging the impact of an attack based on the metrics from the CVE database (MITRE, 2022b).



Reconnaissance
10 techniques

- Active Scanning (1/3)
  - Scanning IP Blocks
  - Vulnerability Scanning
  - Wordlist Scanning
- Gather Victim Host Information (0/4)
- Gather Victim Identity Information (1/3)
  - Credentials
  - Email Addresses
  - Employee Names
- Gather Victim Network Information (0/6)
- Gather Victim Org Information (0/4)
- Phishing for Information (1/3)
  - Spearphishing Attachment
  - Spearphishing Link
  - Spearphishing Service
- Search Closed Sources (0/2)
- Search Open Technical Databases (0/5)
- Search Open Websites/Domains (0/2)
- Search Victim-Owned Websites

Resource Development
7 techniques

- Acquire Infrastructure (2/6)
  - Botnet
  - DNS Server
  - Domains
  - Server
  - Virtual Private Server
  - Web Services
- Compromise Accounts (1/2)
  - Email Accounts
  - Social Media Accounts
- Compromise Infrastructure (0/6)
- Develop Capabilities (0/4)
- Establish Accounts (0/2)
- Obtain Capabilities (1/6)
  - Code Signing Certificates
  - Digital Certificates
  - Exploits
  - Malware
  - Tool
  - Vulnerabilities
- Stage Capabilities (0/5)

Initial Access
9 techniques

- Drive-by Compromise
- Exploit Public-Facing Application
- External Remote Services
- Hardware Additions
- Phishing (2/3)
  - Spearphishing Attachment
  - Spearphishing Link
  - Spearphishing via Service
- Replication Through Removable Media
- Supply Chain Compromise (0/3)
- Trusted Relationship
- Valid Accounts (1/4)
  - Cloud Accounts
  - Default Accounts
  - Domain Accounts
  - Local Accounts

Figure 1. Enterprise techniques used by APT28 (ATT&CK group G007 v4.0) for the Reconnaissance, Resource Development, and Initial Access tactics, retrieved from mitre.org in 2022.

Simulated Target Network

We simulated three networks to be employed in upcoming missions to study how to apply AI techniques for network vulnerability analysis, and how communication supported by explainable AI can impact human perception of the vulnerabilities. These three networks are implemented in the online junior cyber analyst testbed. Three networks are simulated with different network topologies containing different numbers of nodes in each. In each simulated network, every node in the network has a workstation. A workstation can have many nodes and each node will have its own Operating System (OS) so that it is connected to a special type of workstation that is called a local network. Screenshots of one of the simulated target networks is shown in Figure 2. Each workstation has its corresponding server. Each node in this network has a history of 300-350 attacks reported in the simulated cyber-incidents database, discussed above. Of these cyber incidents, 70-80% of the past attacks are considered to be addressed and the remaining 20% or so of the recent ones are yet to be addressed. The vulnerability information displayed for the highly vulnerable nodes in Figure 2 is calculated based on the recent attacks that are left unaddressed. Each Attack will contain the information about the type of techniques used from each tactic (or stage) to orchestrate the attack on the node.

	Network 1	**Network 2**	**Network 3**
Number of Nodes	11	10	11
Total Highly Vulnerable Nodes	4	5	7
Number of Servers	5	4	5
Number of nodes with the highest number of Attacks	2	2	1

Table 1: Information about the three simulated networks used in the junior cyber analyst testbed.

Attack Group Analysis

With the simulated dataset on cyber-incidents and three simulated target networks, we set out to design an AI-driven junior cyber analyst. The simulated dataset contains cyber-incidents on networks that are similar to the target network. Thus, the goal of junior cyber analyst is to understand for the target network (1) who might be behind the past attacks and (2) what might happen next if the vulnerabilities are left unaddressed, based on what it learns about the similar networks from the cyber-incident dataset. The identity of the adversary in the past attack can be inferred through policies derived from supervised learning on the simulated dataset. Using the tactics as features, techniques as feature values, and the adversaries as labels, identifying the adversary becomes a classification task for machine learning. To predict the attack group from the observed techniques on the node in the network, we use the simulated dataset as described in the previous section. To create the training data, we analyzed the observed techniques for each attack orchestrated by an attack group from the cyber incident data and created vectors using one-shot encoding described as follows:

a. Based on the total number of techniques in the MITRE ATT&CK repository, we assigned a unique ID to each of the techniques. For example, there are a total of 191 techniques. So, we assigned an index to each technique ranging from 0 to 190 for all the 191 techniques and stored these index values with the name of the technique as keys in a hash table. In a similar fashion, we assigned indexes to the attack groups so that for all the attack groups present in the dataset, there is a unique ID associated with it. These IDs are stored in another hash table with the name of the group name as the key.
b. For each attack observed, we created a vector of 0's of size 191. Then, based on the techniques observed in each attack in the cyber-incident dataset, we switched the value of 0 to 1. Thus, for each attack in the cyber-incident dataset, there is a unique vector of 0s and 1s for each attack. In the similar fashion, we created signatures for each attack group, using the vectors of 0s and 1s, based on the techniques they commonly use, as described in the MITRE ATT&CK model.
c. We did this for all the attacks present in the dataset and created the dataset with Y labels as indexes of the attack groups, and X values containing the vectors for each attack.

To add noise to the dataset to synthesize the ambiguity in determining the attack groups in real-world attacks, we generated a random index between 0 and 190 for the vector of size 191 and swapped the value contained in the index from 0 to 1 or from 1 to 0. This is to simulate the use of techniques different from the exact ones used by the labeled attack groups in the dataset, i.e., the attackers might use new attack techniques to orchestrate future attacks or might not always use the old attack techniques that they were using before. We add noise to 20% of the vector size. In other words, we changed values to the index of 38 random techniques within the entire vector from 0's to 1's or from 1's to 0's.

We then split the data into 80% as training and 20% testing, and applied several machine learning algorithms to the data to learn a model/policy to predict the attack groups behind the attacks. Because we aim to generate explanations on how attack groups are determined, we focused on machine learning methods that are more humanly interpretable. The performances of the different algorithms are shown in Table 2:

Model	**Accuracy**
K Nearest Neighbors	0.9320910938959398
Decision Tree	0.8809406076589198
Naive Bayes	0.9845063458051755
SVM	0.9707158947310587
Random Forests	0.48338003406406244

Table 2: Performances of five machine learning methods used to learn a model to predict attack groups using the simulated cyber-incident dataset.

As the results show, the K Nearest Neighbor and Decision Tree algorithms out-performed the others on our dataset. We then experimented with the generation of explanations for the policies learned through both algorithms. For decision-trees, the nodes in the tree represent the series of ATT&CK techniques used to identify the group. And the leaves at the bottom of the tree represent the attack groups. We then generated the explanation that describes the path from the root of the tree to the leaf of the tree. For example, "The series of techniques shown in the decision tree path that lead to the identification of the attacker are: Remote Services, Network Sniffing, Credentials from Password Stores, Software Deployment Tools, Lateral Tool Transfer, Use Alternative Authentication Material, Exploitation of Remote Services, Replication Through Removable Media, Virtualization/Sandbox Evasions, OS Credential Dumping, Fallback Channels." Such explanations, while accurate, are difficult for novices without deep knowledge of cyber operations to parse and understand. Thus, we chose to generate the explanations based on the Nearest Neighbor by providing the nearest cluster centers, which represent different attack groups, and the distance to the center as a confidence score. An example of the explanation is shown in Figure 3. This would allow the human operator to evaluate, and possible determine which of multiple adversaries is most likely responsible.

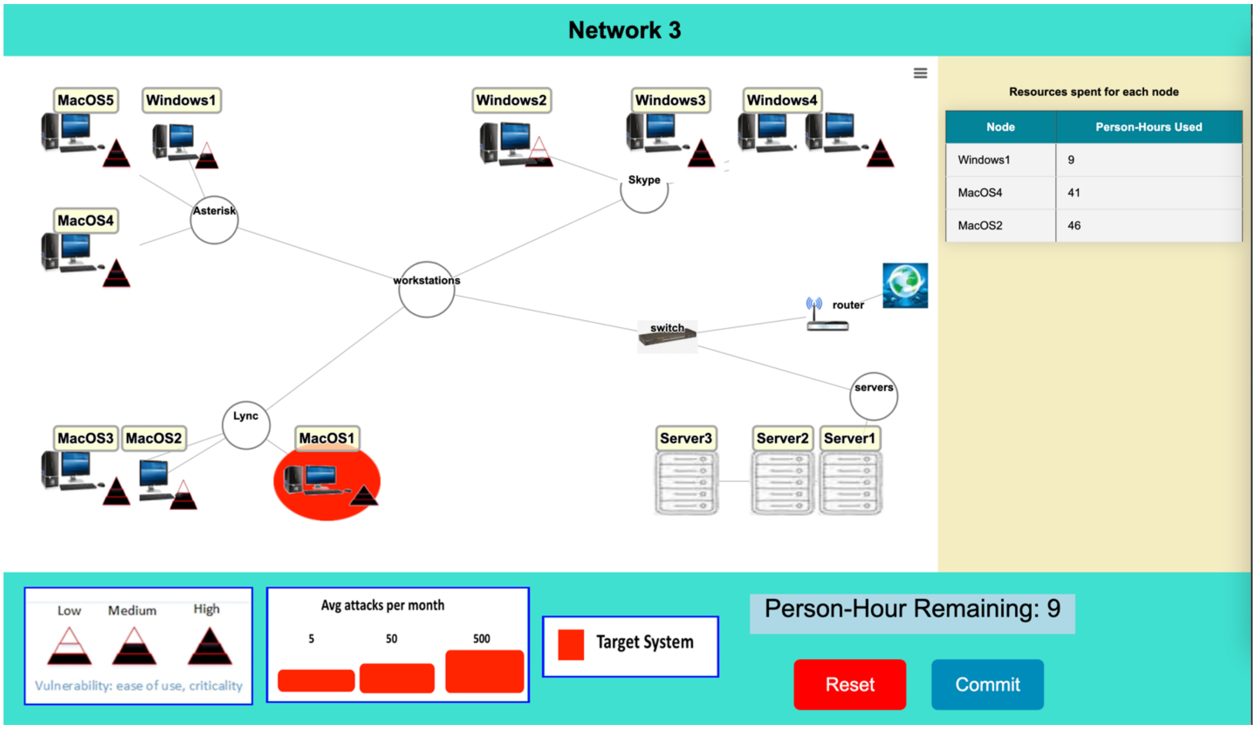
Predictions of Next Steps

Upon identifying the adversary, we can thus infer the possible next steps in the event of a future attack, if the vulnerabilities are left unaddressed. One method we experimented with, in order to understand what might happen next to the simulated target network, is to analyze what commonly happens together in an attack based on the behavior of the adversaries described in the MITRE ATT&CK repository. The analysis of commonly associated steps, such as techniques often used in an attack, can be achieved using unsupervised machine learning on incident reports, such as hierarchical clustering (Al-Shaer et al. 2020). In our work, the model used for prediction of the next technique of an attack was trigram probabilities. We build trigrams by first looking at the sequence of techniques used in any given attack in the cyber-incident dataset. And with a rolling window size of three, we build a trigram with the series of three techniques within that window. We then calculate the probability of a specific trigram (or a sequence of three techniques in an attack) by the total number of such trigrams observed in our dataset by a specific attack group, divided by the total number of attacks of the attack group orchestrated in the simulated cyber-incident dataset. These trigram probabilities for various combinations of three techniques of all attacks for all groups are then stored in a hash table. The keys of the hash table are tuples that are made up of the attack group's name, the trigram, and the trigram probability as described above. Thus, to predict possible next steps in an attack, we can first look at the two most recent techniques used in the attack on a node in the target network and then create a list of all the possible 3rd techniques that can happen in the next stage of an attack by searching through the trigrams, i.e., combinations of techniques of <*observed technique 1*, *observed technique 2*, *possible technique 3*>. We then look up the probabilities of all such tuples of techniques by extracting their values from the hash table that we created earlier. We then choose the tuple with the highest probability and predict the 3rd technique in the tuple as the next steps that can happen in future attacks. An example of explanations on the likely next steps are shown in Figure 3.

JUNIOR CYBER ANALYST TESTBED

To test the impact of the explanations on the human performance when working with the junior analyst, we put together the junior cyber analyst testbed. The testbed presents the user with three target networks for vulnerability analysis. The user can inspect the vulnerability analysis and the recommended remediation strategy to decide how to allocate limited resources (as person-hours) to address the vulnerabilities. The recourses provided are not enough to address all the vulnerabilities. This is to create the need for the users to examine the network and read through the explanations about the vulnerability diagnoses and impact estimates in order to make sound decisions on how to allocate the resources. The soundness of such decisions will be "tested". After the user finishes allocating the resources, a simulated outcome of the impact of the network on mission performance is displayed, before moving onto analyzing the next network. For example, some of the workstations could be compromised and the impact could be extended to all the workstations in the local network. Each target network presents a separate task for the user. An example network is shown in Figure 2. The interface displays at-a-glance the network nodes with possible vulnerabilities, the severity of the vulnerabilities, resources allocated (and the nodes they are allocated to), and resources remaining.

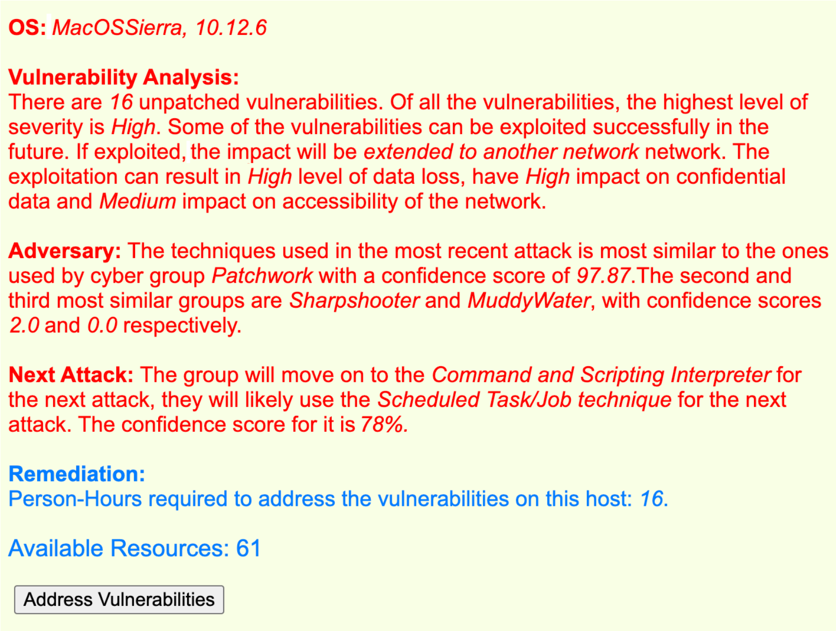
The user can click on a network node to inspect a network node with vulnerabilities. After the mouse click, a window that contains the details of the vulnerability analysis will pop up, as shown in Figure 2. The analysis includes the number of vulnerabilities, the severity of vulnerabilities, the estimated impact, the possible adversaries behind past attacks, the likely follow-up actions by them, and the resources needed to address the vulnerabilities. The resources, i.e., the person-hours, required to address the vulnerability is proportional to the number of unpatched vulnerabilities and various indices of the severity of the vulnerability, such as exploitability, impact if unaddressed, etc. If the user decides to address the vulnerabilities, the person-hours required will be deducted from the available person-hours.



Resources spent for each node

Node	Person-Hours Used
Windows1	9
MacOS4	41
MacOS2	46

Figure 2: Screenshot of a simulated network. In this network, there are 3 workstations (Lync, Asterisk and Skype). All the workstations are connected to the internet using a router. Vulnerability is represented by the triangles at every node, and the nodes that are attacked many times are displayed with a red background.



OS: *MacOSSierra, 10.12.6*

Vulnerability Analysis:
There are *16* unpatched vulnerabilities. Of all the vulnerabilities, the highest level of severity is *High*. Some of the vulnerabilities can be exploited successfully in the future. If exploited, the impact will be *extended to another network* network. The exploitation can result in *High* level of data loss, have *High* impact on confidential data and *Medium* impact on accessibility of the network.

Adversary: The techniques used in the most recent attack is most similar to the ones used by cyber group *Patchwork* with a confidence score of *97.87*.The second and third most similar groups are *Sharpshooter* and *MuddyWater*, with confidence scores *2.0* and *0.0* respectively.

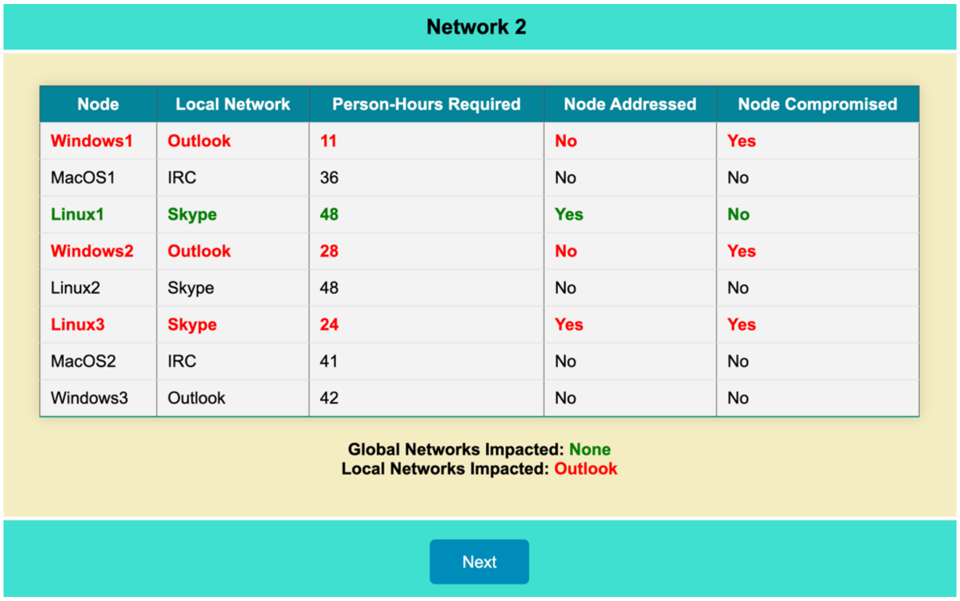
Next Attack: The group will move on to the *Command and Scripting Interpreter* for the next attack, they will likely use the *Scheduled Task/Job technique* for the next attack. The confidence score for it is *78%.*

Remediation:
Person-Hours required to address the vulnerabilities on this host: *16.*

Available Resources: 61

Figure 3: The pop-up window after a network node is selected. Details of the vulnerability analysis are shown here.

After the user completes the resource allocation and clicks the "Commit" button in Figure 2, they will be shown the outcome of how the network and the mission it is deployed to are impacted. Figure 4 shows all the nodes present in the network, the number of resources allotted for each node, and if a specific node is compromised or not. The nodes that are addressed and not attacked are colored in green and the nodes that are attacked are colored in red. At the bottom, we can see how the allocation of such resources may contribute towards the collapse of a network either globally or locally. The local network is impacted if more than two nodes in the same network are impacted. And the global network is considered to be impacted if more than half of the local networks are impacted. The probability that a node is attacked is based on statistics from real-life network intrusions (e.g., frequency of cyber-attacks), type of attacks, the scale of impact of the vulnerability (e.g., local or global network), etc.



Network 2

Node	Local Network	Person-Hours Required	Node Addressed	Node Compromised
Windows1	Outlook	11	No	Yes
MacOS1	IRC	36	No	No
Linux1	Skype	48	Yes	No
Windows2	Outlook	28	No	Yes
Linux2	Skype	48	No	No
Linux3	Skype	24	Yes	Yes
MacOS2	IRC	41	No	No
Windows3	Outlook	42	No	No

Global Networks Impacted: None
Local Networks Impacted: Outlook

Figure 4: The screenshot shows the nodes post-attack

CONCLUSIONS AND FUTURE WORK

In this paper, we discussed the design of a junior cyber analyst simulation testbed for the study of human-AI teaming, where a human participant plays the role of a senior analyst to perform network vulnerability analysis and implement mitigation strategies. We created a simulated dataset of past cyber-incidents on military networks using techniques and adversaries described in the MITRE ATT&CK repository. We then experimented with various machine-learning methods to build models/policies to predict adversaries behind the attacks, which then leads to the prediction of next steps in the attack. One of the immediate next steps is to conduct human-subject experiments with the simulation testbed to study how such explanations on AI-supported vulnerability analysis can facilitate a cyber operator's decision-making, which can further inform the design of the explanations and the choice of applicable AI techniques.

ACKNOWLEDGEMENTS

This project is supported by the Cyber Security and Applied Research Program from the Office of the Assistant Secretary of Defense Research and Engineering. Statements and opinions expressed do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

REFERENCES

Al-Shaer, R., Spring, J., Christou, E. (2020). Learning the associations of MITRE ATT&CK adversarial techniques.
Bishop, C.M. (2006). Pattern recognition and machine learning. Springer, Berlin

Chang, C. H., Creager, E., Goldenberg, A., & Duvenaud, D. (2018). Explaining image classifiers by adaptive dropout and generative in-filling. arXiv preprint arXiv:1807.08024

Core, M., Traum, D., Lane, H.C., Swartout, W., Gratch, J., van Lent, M., Marsella, S. (2006) Teaching negotiation skills through practice and reflection with virtual humans. Simulation 82(11):685–701

Dodson, T., Mattei, N., Goldsmith, J. (2011) A natural language argumentation interface for explanation generation in Markov decision processes. International conference on algorithmic decision theory. Springer, Berlin, pp 42–55

Elizalde, F., Sucar, L. E., Luque,M., Diez, J., & Reyes, A. (2008). Policy explanation in factored Markov decision processes. In proceeding of the 4th European workshop on probabilistic graphical models (pp. 97-104)

Gunning, Dave. (2019). Keynote presented at the ACM IUI 2019.

Guo,W., Mu, D., Xu, J., Su, P., Wang, G., & Xing, X. (2018). Lemna: explaining deep learning based security applications. In proceedings of the ACM conference on computer and communications security (pp. 364-379).

Hendricks, L.A., Akata, Z., Rohrbach, M., Donahue, J., Schiele, B., Darrell, T. (2016) Generating visual explanations. In European conference on computer vision. Springer, Cham, pp 3–19

HQUSINDOPACOM(2019).Mission relevant terrain—cyber (MRT-C) campaign update. Briefing

Hutchins, E.M., Cloppert, M.J., Amin, R.M. (2011) Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. Retrieved December 15, 2020

Jha, S., Sheyner, O., Wing, J. (2002) Two formal analysis of attack graphs. Retrieved Dec. 15, 2020 from http://www.cs.cmu.edu/wing/publications/Jha-Wing02.pdf

Johnson, W. L. (1994). Agents that learn to explain themselves. In Proc. of the 12th National Conference on artificial intelligence.

Khan O, Poupart P, Black J, Sucar LE, Morales EF, Hoey J (2011) Automatically generated explanations for Markov decision processes. In: Decision Theory Models for Applications in Artificial Intelligence: Concepts and Solutions, pp 144–163

Koul, A., Greydanus, S., & Fern, A. (2018). Learning finite state representations of recurrent policy networks.

Mitre Corporation. (2022a). Adversarial tactics, techniques & common knowledge. Retrieved from Mitre.org: https://attack.mitre.org/

Mitre Corporation. (2022b). Common Vulnerabilities and Exposures (CVE). Retrieved from https://www.cve.org/.

O'Leary DE (2013) Artificial intelligence and big data. IEEE Intell Syst 28(2):96–99

Parasuraman R, Riley V (1997) Humans and automation: use, misuse, disuse, abuse. Hum Factors 39(2):230–253

Pynadath, D. V., Rosoff, H., John, R.S. (2016) Semi-automated construction of decision-theoreticmodels of human behavior. In Proceedings of the international conference on autonomous agents & multiagent systems, pp 891–899

Raymond D, Conti G, Cross T, Nowatkowski M (2014) Key terrain in cyberspace: seeking the high ground. In: Proceedings of the 6th international conference on cyber conflict.

Estonia, Tallin Ribeiro,M. T., Singh, S.,&Guestrin, C. (2016).Why should I trust you?: explaining the predictions of any classifier. In proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1135-1144). ACM

Shih, A., Choi, A., & Darwiche, A. (2018). A symbolic approach to explaining Bayesian network classifiers. arXiv preprint arXiv: 1805.03364

Si, Z., Zhu, S.C. (2013) Learning and-or templates for object recognition and detection. IEEE Trans Pattern Anal Mach Intell 35(9):2189–220

Swartout, W.R., Moore, J.D. (1993) Explanation in second generation expert systems. In: In second generation expert systems. Springer, Berlin, pp 543–585

Walker, S. (2018). Closing remarks presented at DARPA D60 symposium

2016 US Cyber Command (2020). Cyber warfare publication 3–33.4: cyber protection team (cpt) organization, functions and employment (28 January 2020)

van Lent, M., Fisher,W. & Mancuso, M. (2004). An explainable artificial intelligence system for small-unit tactical behavior. Proc. of the 16th conference on innovative applications of artificial intelligence (IAAI)

Holder, E., & Wang, N. (2021). Explainable artificial intelligence (XAI) interactively working with humans as a junior cyber analyst. *Human-Intelligent Systems Integration*, *3*(2), 139-153.

Pynadath, D.V., Wang, N., & Barnes, M.J. (2018). Transparency Communication for Reinforcement Learning in Human Robot Interactions. In Proceedings of the Workshop on Explainable Artificial Intelligence (XAI) of the 27th International Joint Conference on Artificial Intelligence.

Pynadath, D.V., Gurney, N., and Wang, N. (2022) Explainable Reinforcement Learning in Human-Robot Teams: The Impact of Decision-Tree Explanations on Transparency. To appear in the International Symposium on Robot and Human Interactive Communication (RO-MAN).