ThreatZoom: Neural Network for Automated Vulnerability Mitigation

Ehsan Aghaei
University of North Carolina-Charlotte
Charlotte, North Carolina, USA
eaghaei@uncc.edu

Ehab Al-shaer
University of North Carolina-Charlotte
Charlotte, North Carolina, USA
ealshaer@uncc.edu

ABSTRACT

Increasing the variety and quantity of cyber threats becoming the evident that traditional human-in-loop approaches are no longer sufficient to keep systems safe. To address this momentous moot point, forward-thinking pioneers propose new cyber security strategy using automation to build a more efficient and cheaper defense. Associating large number of unpatchable CVEs (vulnerability descriptions) generated everyday to appropriate CWE (weakness) and CAPEC (attack pattern) can be used to automatically infer the expected impact and corresponding mitigation course of actions for that new CVE. Routinely, adversary exploits a vulnerability to trigger a cyber attack where this vulnerability results from a product or system weakness. Hence, finding a common system weakness associated with a vulnerability within a particular product can help to identifying the software, system, or architecture flaw and the potential attack impacts. This identification leads to prevent, detect, and mitigate those flaws. On the other hand, after recognizing the cause and the effect of a vulnerability, discovering the procedural-oriented description of the attack to create behavioral observables for detection and mitigation is necessary that can be derived from CAPEC and ATTCK. Mapping the CWE to CAPEC and ATTCK which provides pre-TTP and post-TTP respectively where TTP stands for Tactics, Techniques, and Procedures. Having all CWE, CAPEC, and ATTCK in one hand enables us to find corresponding mitigation for each one. On the other hand, extracting threat actions provided by each of these concepts leads to find another type of mitigation coming from Critical Security Controls (CSC).

In this proposal, the target is to do mapping all the way from CVE to CAPEC and ATTCk automatically using machine learning, deep learning, and natural language processing and find the appropriate mitigation for each one and then find a proper patch as course of action defense. So far, we have introduced a neural network model which successfully classifies CVE to CWE automatically and as working on a deep learning model to classify CWEs to CAPEC.

KEYWORDS

Cyber Security, CVE, CWE, CAPEC

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
HotSoS, April 1–3, 2019, Nashville, TN, USA
© 2019 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-7147-6/19/04…$15.00
https://doi.org/10.1145/3314058.3318167

1 INTRODUCTION

Now a day, adversaries exploit new vulnerabilities, and perform new attacks that make cybersecurity getting to be tougher consistently. Although, regulatory organizations change and amend standards, threat actors are always one step ahead and innovate constantly. Hence, despite the availability of various security products, there are lots of new threats appear everyday. This renders an exponential growth in data from security sensors to protect systems. This growth inbreeds high expense and excessive shortage of cybersecurity experts who can analyze, understand, and respond to the incident gained from data. Meantime, a human-dependent task often is not flawless and more complicated tasks result less accurate outcomes. Hence, a major challenge is to discover an automated solution employing machine learning to process the information extracted from data without relying on human-being to have cheaper and more precise defensive system. Routinely, an adversary exploits a vulnerability to trigger a cyber attack where this vulnerability results from a product or system weakness. Thus, finding a common system weakness associated with a vulnerability within a particular product may help to identifying the software, system, or architecture flaw and the potential attack impacts. This identification leads to prevent, detect, and mitigate those flaws.

The Common Vulnerabilities and Exposures (CVE) reports represent the standard means for sharing cybersecurity vulnerabilities . The Common Weakness Enumeration (CWE) is a hierarchical category system for software weaknesses for the purpose of understanding flaws in software, their potential impact and identifying means to detect, fix, and prevent those flaws. A great effort has been made to manually classify some CVEs to one or more CWE classes in order to provide insight into the understanding of the potential threats and identifying mitigation techniques. However, considering the large number of produced CVEs, and the labor and time-consuming of the manual classification, few thousands CVEs out of more than hundred thousand existing CVEs have been classified to CWEs so far. This will tremendously limit the effectiveness of threat mitigation, particularly when a patching is unavailable, and undermine the usability of CVE sharing.

CWEs are non-disjoint classes and they are organized in a hierarchical (tree) structure to reflect this relationship. In addition, CVE enables a number of different weakness and countermeasures. Thus, a CVE instance can be classified into one or more interdependent CWE classes that belong to the same path in the CWE tree structure, or dependent CWE classes that belong to different paths in the CWE tree structure. In addition, many of the CWE classes includes a small number of CVE instances. Moreover, there is less than 5% of the CVE that were classified manually by MITRE experts, and there are about 50% of the created CWE classes were not used in CVE

classification. These properties of the existing CVE-CWE dataset impose serious challenges in predicting the CWE relationship of the unclassified CVEs. As of now, there is no system to automatically perform this classification which is a compelling reason for manual classification. This dependency on human expert is dramatically expensive, time consuming, inaccurate, and keeps this as a life-time problem. Manual classification of each upcoming CVE requires experts to sit, learn, and remember every single CWE for a comparative study which is not feasible in the real-world.

The initial step to find a proper patch (defense) for a vulnerability is to discover the corresponding threats and then the mitigation. In general, a CWE shows the cause and consequence of an attack that is part of the threat definition. After recognizing the cause and effect of a vulnerability, it is important to discover the procedural-oriented description of the attack to create behavioral observables for detection and mitigation. Hence, another concept called CAPEC (Common Attack Pattern Enumeration and Classification) introduced by MITRE. CAPEC particularly helps to understand **how** and **where** a system is likely to be attacked. In other words, the action and the cyber object and consequently the resulting attack can be extracted from CAPEC, which are not provided by a vulnerability.

A vulnerability uses a weakness (cause) to create an attack (consequence) using an attack pattern (CAPEC). Mapping a vulnerability to CWE leads to understanding software, system, or architecture flaws that allows this exploit. This mapping helps to Understand the potential impact and identifying means to prevent, detect, and mitigate the indicated flaws. Meanwhile, mapping CWEs to CAPEC provides the most common methods that attackers use to exploit vulnerabilities resulting from CWEs. Statistics indicate 70% of the CWEs and 51% of the CAPECs provide course of action (CoA) mitigation while more than 95% of the CVEs are not classified to CWEs and 64% of CWEs are not mapped to CAPECs. If the CWE is mapped to CAPEC then it is possible to map CWE to MITRE ATTACK (ATT&CK) following the similar approach since both CAPEC and ATT&CK provides almost the same concepts. In general, CAPEC discusses pre-attack techniques and ATT&CK provides post-attack techniques.

Along with CoA mitigation found by CWE, CAPEC, and ATT&CK, the threat actions can be extracted from those. Threat action defines what steps the attacker follows to cause or contribute to a specific malicious activity. Every individual cyber attack has at least one threat actions but in most cases, it comprises multiple actions. These threat actions may lead to finding another type of mitigation called CIS critical security control (CSC) mitigation. CIS CSC recommends a set of actions for mitigation that provide specific and actionable ways to stop many pervasive attacks. By having all these concepts in one place, we already know the attacker behavior and we can predict an attack and find a proper solution or a patch for that as CoA defense. The CoA defenses for attacks include network hardening, service hardening, and detection. Figure 1 shows all the steps from taking a vulnerability, finding threats and threat actions, discovering mitigation, and finally finding CoA defense.

So far we have proposed a novel algorithm that automatically estimates the CWE class corresponding to a CVE instance. The algorithm takes the text description for a CVE instance as input and its output is the list of CWE classes along the path that connects the root of CWE hierarchical structure to the fine-grain CWE class

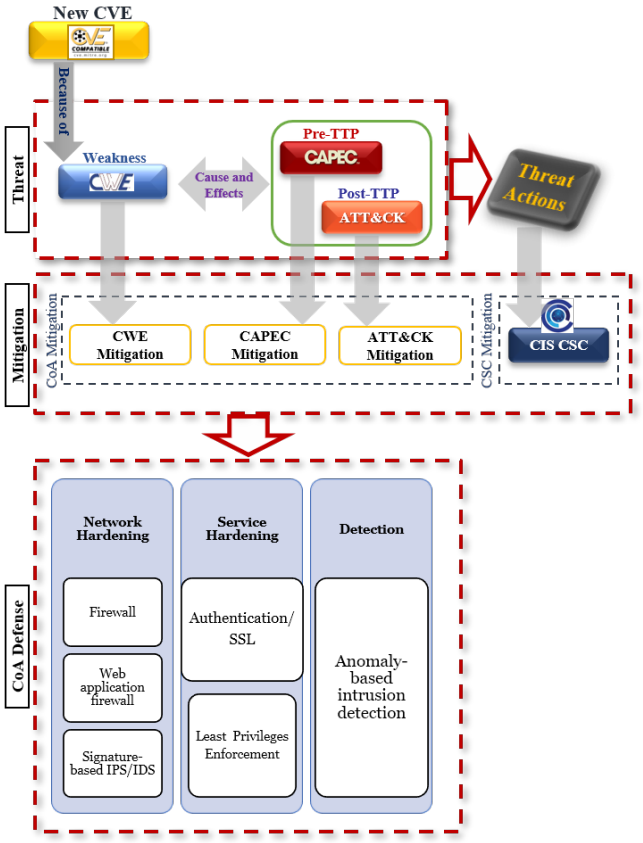


Figure 1: Flow of reaching CoA mitigation from a vulnerability

corresponding to the input instance. The classification procedure consists of three steps: (1) pre-processing, (2) feature extraction, and (3) hierarchical decision-making process. The algorithm extracts the text from the name and description of existing CWE classes and their associated CVE instances. This text is processed to extract features representing the CWE classes. Each feature is assigned a score that reflects its importance in the feature space. A CVE instance with no CWE class associated is analyzed using this feature space and a list of candidate CWE classes is generated. The analysis is performed within a neural network framework. A hierarchical neural network is used to estimate the class associated with that instance at each level. The weights of the neural network are assigned the values of the feature scores. This framework serves as a profound method to adjust these scores to reduce the classification error. This work has four key contributions: (1) the design of the first algorithm that automatically classifies CVEs to CWE classes, (2) a novel hierarchical decision approach that is able to trace CWE classes to the finest grain in the CWE hierarchical structure, (3) a new approach to compute the score that reflects the importance of a feature in describing the vulnerabilities of CVE instances at each level in the CWE hierarchical structure, and (4) an adaptive learning technique that incorporates the computed feature scores in a neural network framework to allow an automatic adjustment

for feature scores to reduce classification errors. The algorithm is tested using CVE instances provided by MITRE. The test results show promise to heighten threat understanding of CVE instances and may allow experts to take the proper actions to address the threats and maintain the cybersecurity of the networks.

Leveraging deep learning, we are mapping CWEs to CAPEC and ATT/CK and find the corresponding mitigation. Meantime, each of mentioned concepts provides different kinds of threat actions which are being extracted using synthetic analysis of their description to feed another deep learning model to find the proper and accurate path automatically.

REFERENCES