PhD in Information Technology and Electrical Engineering
Università degli Studi di Napoli Federico II

PhD Student: Vittorio Orbinato

Cycle: XXXVI

Training and Research Activities Report

Academic year: 2021-22 - PhD Year: Second

Tutor: prof. Domenico Cotroneo

Co-Tutor: prof. Roberto Natella

Date: November 3rd, 2022

1. Information:

- **PhD student: Vittorio Orbinato** **PhD Cycle: XXXVI**
- **DR number: DR995144**
- **Date of birth: 20/11/1996**
- **Master Science degree: Computer Engineering** **University: Università degli Studi di Napoli Federico II**
- **Scholarship type: MUR PON**
- **Tutor: prof. Domenico Cotroneo**
- **Co-tutor: prof. Roberto Natella**

2. Study and training activities:

Activity	Type[1]	Hours	Credits	Dates	Organizer	Certificate[2]
Connecting the dots: Investigating an APT campaign using Splunk, Dr. Antonio Forzieri	**Seminar**	**2**	**0.4**	**26/11/2021**	**Proff. D. Cotroneo, S.P. Romano, R. Natella**	**Y**
Threat Hunting Essentials, Group-IB	**Seminar**	**2**	**0.4**	**03/12/2021**	**Proff. D. Cotroneo, S.P. Romano, R. Natella**	**Y**
Threat Hunting Use Cases, Group-IB	**Seminar**	**2**	**0.4**	**13/12/2021**	**Proff. D. Cotroneo, S.P. Romano, R. Natella**	**Y**
Virtualization Technologies and their Application	**Course**	**20**	**5.0**	**17/01/2022 - 04/03/2022**	**Prof. Luigi De Simone**	**Y**
Privacy-preserving Machine Learning, Dr. Vittorio Prodomo	**Seminar**	**2**	**0.4**	**14/10/2022**	**Proff. S.P. Romano, R. Natella**	**Y**

1) Courses, Seminar, Doctoral School, Research, Tutorship
2) Choose: Y or N

2.1. Study and training activities - credits earned

	Courses	Seminars	Research	Tutorship	Total
Bimonth 1		1.2	8.8	1.6	11.6
Bimonth 2			10.0		10.0
Bimonth 3	5.0		5.0		10.0
Bimonth 4			10.0		10.0
Bimonth 5			10.0		10.0
Bimonth 6		0.4	9.6		10.0
Total	5.0	1.6	53.4	1.6	61.6
Expected	30 - 70	10 - 30	80 - 140	0 – 4.8	

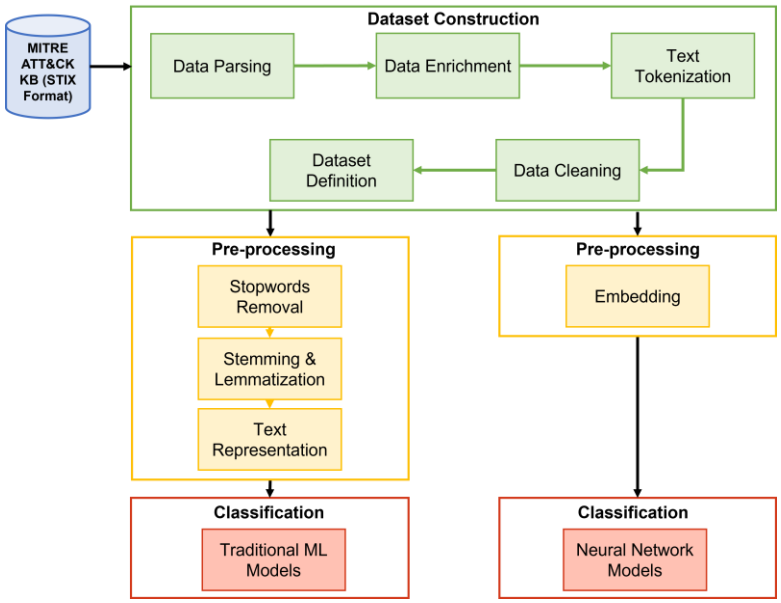
3. Research activity:

My second year in the ITEE PhD was focused on addressing the main issues in the automation of Adversary Emulation and the related activities.

3.1 Automatic mapping of unstructured Cyber Threat Intelligence

Adversary Emulation is a proactive approach that emulates attackers within an IT system for training and evaluation purposes. The goal of Adversary Emulation is to improve the resiliency of the infrastructure against adversary TTPs. Unfortunately, this approach comes with a high cost, in terms of specialized personnel (red teams) needed to emulate attackers, which limits its adoption. In recent years, adversary emulation tools have been emerging as a promising solution to make security exercises easier. These tools automate the execution of individual, low-level malicious actions, such as for credential stealing, lateral movement, data exfiltration, and more. Despite this support, they still need to be programmed to orchestrate multiple actions and to emulate the behaviors of a real attacker.
Adversary emulation exercises need to be guided by Cyber Threat Intelligence (CTI), i.e., information on the techniques and intents of attackers.
However, most CTI still comes in unstructured forms, such as incident reports written by security analysts, and documents leaked by insiders from attacker groups.
Converting CTI into a structured form is still a missing step towards supporting adversary emulation with automated tools. Therefore, the industry is currently investing efforts in standardizing CTI, such as the Structured Threat Information eXpression (STIX) representation format.
The emulation of an attack relies on accurate planning of adversarial activities, which is currently **performed by human operators**. To overcome this issue, we propose an approach to **automatically extract adversary TTPs from CTI**, **mapping them to a standard taxonomy** (e.g., MITRE ATT&CK) to feed the emulation tool.

Our experimental approach revolves around three phases: *Dataset Construction*, *Pre-processing*, and *Classification*.



The outcomes will be several classification models ready to be used to identify adversarial techniques. The goal is to fit CTI into a finite set of categories (i.e., ATT&CK techniques): this problem is addressed as a classification machine learning task. Therefore, new datasets were built to train and to evaluate machine learning algorithms and to perform several experiments on multiple classification techniques.

The *Dataset Construction* phase consists in the definition of a new dataset, composed by unstructured descriptions of adversarial techniques in natural language. The dataset was built using the **public knowledge base** of the MITRE ATT&CK framework, released using the STIX language. This phase consists in the definition of a new dataset, composed by unstructured descriptions of adversarial techniques in natural language. Each sample in the dataset describes one malicious technique, and has been annotated with a label, i.e., a technique from the taxonomy of the MITRE ATT&CK framework. All 188 techniques in the MITRE ATT&CK framework were covered. Each technique can appear in the dataset multiple times with different descriptions, as a technique may have been adopted in multiple attack campaigns, even if in different ways.

Once the dataset has been built, it needs to be *pre-processed*. The Pre-processing phase encompasses different steps depending on the type of the machine learning model. For traditional machine learning (ML) models, this phase starts with Stopwords Removal. The following step is Stemming and Lemmatization. The last step of this phase concerns the Text Representation: the information contained in a sentence is converted to a bag of words, with each sentence encoded as a one-hot vector. Using bags of words results in large feature vectors: this may cause loss of the word context and position inside the sentence. In a large corpus, it is common to have more frequent words, related to language patterns, that do not bring useful features to the classifier and less frequent yet informative words. To balance the representation, we can apply Term-Frequency Inverse Document-Frequency (TF–IDF) term weighting to transform count features into floating point values, preserving meaningful but rarer

information present in the entire corpus. On the contrary, for neural network models, the pre-processing phase consists of representing text with word embeddings: this type of representation encodes each word taking the context into account. In this way, it is possible to define the similarity between two words, by means of their distance.
The classification was performed using two sets of models:

- *Traditional ML models*: include well-studied classifiers used for NLP tasks since decades, including Naive Bayes, Logistic Regression, Support Vector Machines (SVM), and Multi-Layer Perceptron (MLP).
- *Deep neural networks*: we considered recent classifiers based on complex neural network architectures, including Recurrent Neural Networks (RNN), LSTM, Convolutional Neural Networks (CNN), and Transformers.

Among the deep learning-based classifiers, the **Transformer architecture** represents the most recent advancement in the field of NLP research. It introduces the self-attention mechanism, which weights the tokens of the input data according to their importance. Therefore, we also built a classifier using **SecBERT**, a Language Model (LM) pre-trained on cybersecurity terms.

3.2 Anti-detection in Adversary Emulation

A relevant problem in the adoption of Adversary Emulation tools is the lack of an important part of the attackers' techniques: *anti-detection*. In real scenarios, attackers pay attention to hide their traces, to avoid being detected by security teams and products (e.g., EDRs). Examples of anti-detection techniques include un-hooking probes used by EDRs to instrument and monitor DLL and API uses; disabling or hampering event tracers used by SIEM systems to collect events, such as by attacking the Event Tracing for Windows (ETW) subsystem; using malicious kernel modules to hide processes and files; encrypting malicious payloads (e.g., shellcodes). As a result, attackers and defenders are involved in a “cat-and-mouse” game, where more robust detection techniques are continuously developed to respond to new anti-detection techniques. However, such anti-detection techniques are not encompassed in adversary emulation tools. Emulating anti-detection techniques in adversary simulations can be a challenge, as the emulation tool must be customized for the specific EDR to be evaded, which requires significant skills and development efforts. Moreover, as EDRs evolve over time in response to anti-detection, adversary emulation tools need to keep up with this evolution to still be able to evade them. For these reasons, emulating anti-detection techniques in automated ways in adversary simulations is still impractical. We note that this is a significant limitation since the simulation is unable to reproduce sophisticated attack scenarios where the attacker leaves limited traces.

We performed an initial investigation to raise awareness on this problem, and to propose some directions for research in this area. We conducted a preliminary study on MITRE CALDERA, a well-known adversary emulation tool that reached a high level of maturity, by including most of the tactics and techniques of the MITRE ATT&CK matrix. We performed adversary simulations with CALDERA, using an experimental testbed equipped with well-known EDR solutions, and analyzed which parts of the simulations were detected by the EDRs. We found that a significant part of these simulations triggered alarms by the EDRs, thus making it questionable that the simulations are representative of a skilled attacker.

Following the results of this study, we developed a new threat emulation tool for virtualized systems. The tool was tested against several AVs/EDRs, to demonstrate its effectiveness in evading detection solutions, in comparison to state-of-the-art adversary emulation tools.

4. Research products:

Orbinato, V.; Barbaraci, M.; Natella, R.; Cotroneo, D.

"Automatic Mapping of Unstructured Cyber Threat Intelligence: An Experimental Study"

33rd International Symposium on Software Reliability Engineering, ISSRE22, 2022 (accepted)

5. Conferences and seminars attended

Conferences:

- 33rd International Symposium on Software Reliability Engineering (ISSRE 2022), Charlotte, North Carolina, 31/10-03/11/2022. I attended this conference as presenting author.

5. Periods abroad and/or in international research institutions

Abroad research period at the University of Coimbra, Portugal, under the supervision of Prof. Nuno Antunes and Prof. Marco Vieira. The research activities carried out in this period were focused on a field study on the vulnerabilities of the Xen hypervisor, to identify the relevant features of an intrusion, namely the *Abusive Functionality* and the *Erroneous State*. The ultimate goal of such study was the definition of intrusion models to assess the security of virtualized systems.

The abroad research period took place from May 18th, 2022, to December 19th, 2022.

7. Tutorship

Sistemi Operativi, SSD: ING-INF/05, Tutor: prof. Domenico Cotroneo, CdL Ingegneria Informatica

8. Plan for year three

For the next year, my plan is to bring all the current results of my research together, completing the pipeline of the automated execution of Adversary Emulation exercises. The idea is to be able to take any kind of unstructured CTI document, analyse it and generate an emulation plan ready to be executed on our custom threat emulator, developed during my second year. Moreover, in the last part of my abroad research period, I plan to propose a new approach for the definition of intrusion models for virtualized systems, along with some practical examples and Proof of Concepts (PoCs).