Characterizing Attacker Behavior in a Cybersecurity Penetration Testing Competition

Nuthan Munaiah*, Akond Rahman†, Justin Pelletier*, Laurie Williams†, and Andrew Meneely*
*Rochester Institute of Technology, Rochester, NY 14623
Email: nm6061@rit.edu, jxpics@rit.edu, axmvse@rit.edu
†North Carolina State University, Raleigh, NC 27695
Email: aarahman@ncsu.edu, williams@csc.ncsu.edu

***Abstract*—*Background:* Inculcating an attacker mindset (i.e. learning to think like an attacker) is an essential skill for engineers and administrators to improve the overall security of software. Describing the approach that adversaries use to discover and exploit vulnerabilities to infiltrate software systems can help inform such an attacker mindset. *Aims:* Our goal is to assist developers and administrators in inculcating an attacker mindset by proposing an approach to codify attacker behavior in cybersecurity penetration testing competition. *Method:* We use an existing multimodal dataset of events captured during the 2018 National Collegiate Penetration Testing Competition (CPTC'18) to characterize the approach a team of attackers used to discover and exploit vulnerabilities. *Results:* We collected 44 events to characterize the approach that one of the participating teams took to discover and exploit seven vulnerabilities. We used the MITRE ATT&CK™ framework to codify the approach in terms of tactics and techniques. *Conclusions:* We show that characterizing attackers' campaign as a chronological sequence of MITRE ATT&CK™ tactics and techniques is feasible. We hope that such a characterization can inform the attacker mindset of engineers and administrators in their pursuit of engineering secure software systems.**

***Index Terms*—security, vulnerability, attack**

I. Introduction

The importance of security in the engineering and administration of software systems is unquestionable. The last decade has been witness to numerous security vulnerabilities in software exploited to catastrophic effects. While some of these vulnerabilities (like Heartbleed in OpenSSL, Shellshock in Bash, and a vulnerability in Apache Struts that led to the recent Equifax data breach) have garnered widespread infamy, there are several others that developers regularly discover and address. Despite best efforts, the potential for yet undiscovered vulnerabilities in software presents a grim reality [1].

Adversaries, be they ethical white hat hackers or malicious black hat hackers, are behind every vulnerability exploit. The ability to understand the way in which these adversaries discover and exploit vulnerabilities to infiltrate systems can help bring the attacker mindset out of the wild and into the software development process.

The foundational element in inculcating the attacker mindset is to think like an attacker when implementing code or configuration changes to software systems. As the complexity of the software system increases, speculating the different ways in which an attacker could exploit a vulnerability becomes a daunting task. The key, therefore, is to leverage empirical information to characterize the typical ways in which attackers tend to discover and exploit vulnerabilities. The limitation, however, is the scarcity of such empirical information.

978-1-7281-2968-6/19/$31.00 ©2019 IEEE

In the past, researchers [2], [3] have attempted to infer attackers' modus operandi from packet capture or intrusion alert datasets. These datasets, being at the network level, do not render themselves useful for analyses using a post-compromise attacker-behavior-based investigation model such as the MITRE ATT&CK™ [4]. However, a recently released multimodal dataset [5] of events captured during a cybersecurity penetration testing competition provides an opportunity to empirically characterize the approach that participating teams used to discover and exploit vulnerabilities in a controlled environment.

Our goal is to assist developers and administrators in inculcating an attacker mindset by proposing an approach to codify attacker behavior in cybersecurity penetration testing competition. We address the following research question to accomplish this goal:

RQ What MITRE ATT&CK™ tactics and techniques can be inferred from the competition data set?

We analyzed a multimodal dataset containing over 500 million events from six teams of attackers captured and curated by Munaiah *et al.* [5] during the 2018 National Collegiate Penetration Testing Competition (CPTC'18). The primary contribution of our work is an approach to describe attacker behavior using empirical data that captures the vulnerability discovery and exploit from both the victims' and attackers' perspectives.

II. Dataset

The dataset used in addressing the research question was collected and curated by Munaiah *et al.* [5] during the 2018 National Collegiate Penetration Testing Competition (CPTC'18). An interesting aspect of the dataset is that it contains an account of the entire competition from both the victim's and the attacker's perspective. The paper [5] accompanying the dataset has more information about the competition, in general, and the dataset, in particular. However, we provide a brief overview of details that are relevant to this study in the remainder of this section.

CPTC'18 had nine teams competing to discover and exploit vulnerabilities in an enterprise cyberinfrastructure advertised as belonging to a fictitious ride sharing organization (WHEELZ). Each team was provided an isolated and identical copy of the competition infrastructure. The infrastructure was composed of four subnetworks (`vdi`, `corp`, `prod`, and `auto`) with some level of interconnection between them. The `vdi` subnetwork contained hosts that the attackers had full access to to carry out their attack campaign. The range of addresses in `corp`, `prod`, and `auto` was the only other piece of information about the infrastructure provided to the participants.

The dataset is a collection of events captured and indexed by Splunk, a log aggregation platform, from all hosts in the competition infrastructure. Each host in the competition infrastructure was instrumented with a Splunk agent which periodically captured information from a plethora of sources on the host. In addition to the raw data, all events in the dataset are associated with essential metadata, such as the timestamp at which the event was indexed; the name of the host where the event originated; and source type identifying the source of the event on the host.

The event source type is critical to the analysis of the dataset because it indicates the kind of data an event contains, and can be used as a proxy for granularity of the event data. For instance, an event with `mongodb` as the source type contains information about log generated by MongoDB instance on a host, whereas, an event with `stream:tcp` as the source type contains information about TCP packets sent to/from the host. When analyzing the dataset, a `mongodb` event, being at the application level, is at a higher level of granularity than `stream:tcp` event, which is at a network level.

III. VULNERABILITIES

The dataset contains over 500 million events associated with over 113 source types. The metadata associated with the events (source type, in particular) affords us the ability to filter events with relative ease. However, any approach to analyze the events with no hint of attributes of relevance would be akin to searching for a needle in a haystack. The key, therefore, is to leverage the knowledge of the known vulnerabilities in the competition infrastructure that were attributed to have been discovered and/or exploited by an attacker (or, more accurately, a team of attackers).

The competition infrastructure was engineered to have 74 known vulnerabilities with varying levels of severity and difficulty. In aggregate, 29 of the 74 vulnerabilities were discovered and/or exploited by participating teams.

In this preliminary work, we use vulnerabilities attributed to Team 1 to present our approach to characterize attacker behavior. Team 1 was attributed to have discovered and/or exploited the following seven vulnerabilities: (1) Unauthenticated remote administrator access to MongoDB, (2) blank domain administrator password, (3) no SSL on sensitive web application, (4) cleartext credentials in chat history, (5) API documentation leaked on HeckForums, (6) backup file on workstation, and (7) unauthenticated access to sensitive API.

Each of the aforementioned vulnerabilities were isolated to a specific host. For instance, the unauthenticated access to sensitive API vulnerability was in an application that was deployed on `onramp-00` host in the `corp` network. The prior knowledge of vulnerabilities that a team discovered and/or exploited alleviates the seemingly intractable task of gathering evidence from the dataset to construct a timeline of events that characterize the attackers' modus operandi.

We use the unauthenticated remote administrator access to MongoDB as an example at specific points (like in describing the methodology in Section V) in the remainder of this paper. We refer to this vulnerability as *MongoDB vulnerability* for brevity. The MongoDB vulnerability was caused by incorrect configuration of the MongoDB server instance on the `corp\talk-00` which permitted remote access with no password. An attacker could exploit the vulnerability by running `mongo --port 27017 10.0.0.20`, where, `10.0.0.20` is the IP address of `corp\talk-00`.

In the remainder of this paper, we use the term *attack* to refer to the act of an attacker discovering and/or exploiting a vulnerability.

IV. LIFECYCLE OF AN ATTACK

While a timeline of events that characterizes the modus operandi of an attacker is interesting, inferring the tactics and techniques that the attacker may have used in their campaign is essential to developing a generalizable understanding of the behavior of a typical attacker. One way to achieve such a generalizable understanding is to describe the timeline of events using a framework that is meant to characterize the lifecycle of an attack. Mandiant's Attack Lifecycle Model [6], Lockheed Martin's Cyber Kill Chain® [7], and the MITRE Adversarial Tactics Techniques and Common Knowledge (ATT&CK™) framework [4] are the most commonly used approaches to characterize the lifecycle of an attack.

In our work, we use ATT&CK to translate the timeline of events in attackers' modus operandi to tactics and techniques. We chose ATT&CK because of its emphasis on *post-compromise* modeling of the threat in an environment based on *behavioral aspects* of the attacker [4].

A. MITRE ATT&CK™ Framework

The MITRE ATT&CK™ framework is a knowledge base of tactics and techniques that may be used to model the behavior of an attacker [8]. The framework is updated every quarter and its contents are curated by MITRE with contributions from the broader cybersecurity community.

As the name indicates, the framework is a collection of tactics and techniques that an attacker can use to discover and/or exploit a vulnerability. The tactics and techniques are informed by intelligence from incident reporting on real cyber attacks that are publicly available. As of this writing, ATT&CK is organized into two technology domains: Enterprise, representing a traditional enterprise network, and Mobile, representing mobile communication devices. In our work, we restrict ourselves to the Enterprise technology domain since the competition infrastructure resembles that of a traditional enterprise network. At the time we used the framework in our work, ATT&CK had 11 tactics and 223 techniques.

In the context of an attack, ATT&CK technique answers the question "*How* did the attacker discover and/or exploit a vulnerability?" and the corresponding ATT&CK tactic answers the question "*Why* did the attacker discover and/or exploit a vulnerability?" In other words, an ATT&CK technique is the *approach* an attacker used whereas the corresponding ATT&CK tactic is the *objective* behind the attack. For instance, an attacker can use a port scanning utility like `nmap` (the approach) to identify all the open ports on a machine (the objective). In terms of ATT&CK, we can describe this attack as the attacker using the *network service scanning* technique and the *discovery* tactic.

V. Methodology

In this section, we describe our methodology to (1) gather evidence of an attack and (2) characterize attacker behavior by identifying ATT&CK tactic(s) and technique(s) that capture the objective and approach of the attacker in the attack, respectively.

A. Evidence Gathering

In this phase, we gather evidence to show, *beyond reasonable doubt*, a team having discovered and/or exploited a vulnerability. The primary source of evidence is the events dataset [5]; however, in cases where we did not have enough evidence from the events dataset, we used the reports that the teams submitted to fill the gaps. These reports, submitted by the teams at the end of the competition, summarize the attacks that a team successfully accomplished. The evidence, when chronologically ordered, describes the process a team used to discover and/or exploit a vulnerability.

The approach, being manual, required us to sift through thousands of events from the dataset. Fortunately, knowledge of a particular known vulnerability helped considerably narrow down the number of events to analyze.

While some vulnerabilities may be independently discovered, others are dependent on specific vulnerabilities having been previously discovered and exploited. For instance, the cleartext credentials in chat history vulnerability that Team 1 discovered is dependent on the MongoDB vulnerability. In gathering evidence, we also captured such dependency relationships between vulnerabilities.

In the case of some vulnerabilities, the discovery is achieved through exploit. As a result, the evidence gathered for such vulnerabilities show both discovery and exploit simultaneously. For instance, an attacker can discover the MongoDB vulnerability by actually connecting to the MongoDB instance from a remote host without providing any credentials (i.e. by exploiting the vulnerability).

The approach to gather evidence and collate it into a timeline is summarized below. We used Splunk Web Interface to query and/or filter events when gathering evidence.

Victim's Perspective

At a high-level, the approach to gather evidence of an attack from the victim's perspective has five steps: (1) identify the victim; (2) filter events by victim; (3) enumerate source types; (4) evaluate events associated with each source type based on its hypothesized relevance; and (5) identify first instance of attack.

In our work, since the environment was controlled, the host targeted in an attack was known; we refer to this host as the *victim*. We used the identity of the victim to only consider events associated with the victim. We then enumerated the distinct source types that the events were associated with. We familiarized ourselves with the meaning of the source types enumerated and then used the specifics of the vulnerability to hypothesize source types likely to contain evidence of the attack. The guiding principle we used was granularity: events at application level were preferred to those at the network level. If a particular source type yielded no useful information, we moved to source types at a lower level of granularity until we had enough evidence to describe the attack. We were only interested in the first instance of the attack because subsequent attacks would be unlikely to include the discovery aspect of the attack. The events associated with the first instance of the attack provide two key pieces of information: (1) the timestamp of the attack and (2) the identity of the attacker.

In case of the MongoDB vulnerability, we know `corp\talk-00` was the victim. The dataset has 888,220 events associated with `corp\talk-00` across 45 distinct source types. Fortunately, `corp\talk-00` was instrumented with Suricata Intrusion Detection System (IDS), which was successful in identifying the attack as evidenced by a `suricata:alert` event. The `suricata:alert` event identified the first instance of the attack as originating from `vdi\t1-vdi-kali06` (the attacker) at `11/3/2018 11:15:41 AM EDT` (the timestamp). If there was no IDS monitoring the host, we would have started our analysis with `mongodb` events which represent entries from the `/var/log/mongodb/mongodb.log` file. If `mongodb` events yielded no useful information, we would have moved on to `netstat` or `stream:tcp` events.

Attacker's Perspective

At a high-level, the approach to gather evidence of an attack from the attacker's perspective is very similar to that which was used to gather evidence of the attack from the victim's perspective. However, the knowledge of the timestamp of the attack considerably reduces the number of events we have to sift through by providing an instant in time from which to work backwards to identify the attacker's actions. We were interested in identifying the attacker's action that directly led to the attack as we observed on the victim. Therefore, the event we were interested in would likely be at or near the timestamp of an event we identified on the victim. We use the term "near" here because some events are captured periodically. For instance, events associated with the `ps` source type are captured every 30 seconds, so a process invoked to exploit the vulnerability may not show up at the precise timestamp identified on the victim. The relevance of granularity is much more here than it was for the victim because describing the attacker's approach at the application level of granularity is much more insightful than at the network level.

In the case of the MongoDB vulnerability, working backwards from `11/3/2018 11:15:41 AM EDT`, we found a `ps` event at `11/3/2018 11:13:06 AM EDT` that showed the attacker had been running `mongo --port 27017 10.0.0.20` command for 27 seconds which was likely the session that triggered the alert on the victim.

Timeline of Events

In addition to the event(s) showing the attack from both the victim's and attacker's perspective, we need events to describe the discovery aspect if we were to characterize the holistic behavior of the attacker. We worked backward from the timestamp of the event showing the attacker successfully attacking the victim to identify events that can describe the approach the attacker used to discover the vulnerability. The approach to identify these discovery events were guided by us questioning every element in all events we had gathered so far. The search for events to characterize the discovery aspect of a vulnerability was restricted to the host used by the attacker.

In case of the MongoDB vulnerability, we knew that the attacker successfully attacked the victim as evidenced by the attacker running `mongo --port 27017 10.0.0.20`. However, to characterize the discovery aspect of the attack, we must gather evidence of the attacker identifying that (1) `10.0.0.20` was an address assigned to a host in the `corp` network and (2) port `27017` was open on the host. Analyzing the dataset to address these two questions, we found events showing that the attacker used `onetwopunch` [9] which internally invoked `unicornscan` [10] and `nmap` [11] in sequence to discover that port `27017` (the default MongoDB port) was open on a host accessible at `10.0.0.20` (the IP address assigned to the victim).

B. Behavior Characterization

The evidence gathered provides a timeline to describe the way in which an attacker discovered and then exploited a vulnerability. While the timeline of events itself is quite useful, its granularity makes inferring generalizable attacker behavior rather difficult. To overcome this difficulty, we mapped each event in the attackers' timeline to tactics and techniques in the ATT&CK framework. The mapping provides a way to describe the attack at a higher level of granularity and, more importantly, from the perspective of an attacker.

The approach we used to map the events to ATT&CK tactics and techniques is summarized in this section. Since the number of ATT&CK techniques is much larger than the number of ATT&CK tactics, we first identified the ATT&CK tactic to reduce the number of ATT&CK techniques we had to evaluate.

The approach to map events to ATT&CK tactics and techniques, being manual, is subjective. We mitigated the subjectivity by having at least two authors perform the mapping of events to ATT&CK tactics and techniques and used Cohen's κ, an inter-rater reliability measure, to quantify and reason about the level of agreement.

ATT&CK Tactic

We mapped an event to ATT&CK tactic by inferring the objective of the attacker from the evidence associated with the event. For instance, if we have evidence that the attacker ran `nmap 10.0.0.20`, we can infer, based on the utility of `nmap` [11], that the attacker was attempting to enumerate the open ports on `10.0.0.20`. We used the inferred objective to identify the ATT&CK tactic, which, in the case of `nmap 10.0.0.20` event would be *discovery*.

ATT&CK Technique

We mapped an event to an appropriate ATT&CK technique by using the description of the various techniques associated with the ATT&CK tactic previously identified and finding the one that closely matched that which the attacker used. For instance, to *discover* information about a host, the attacker used `nmap` which is a port scanning utility. The use of `nmap` is best described by the *network service scanning* ATT&CK technique which mentions the use of "port scans" as an approach for "listing of services running on remote hosts".[1]

VI. Results

Question: *What MITRE ATT&CK™ tactics and techniques can be inferred from the competition data set?*

The motivation for this research question is to assess if the events dataset has the information needed to characterize an attack from the perspective of the attacker.

We applied the methodology described in Section V to all the vulnerabilities discovered and/or exploited by Team 1. We found 47 events to describe Team 1 discovering and exploiting the seven vulnerabilities that were attributed to them. 44 of the 47 events were sourced from the events dataset, whereas, the remaining 3 events were sourced from the report that Team 1 submitted describing their campaign. 3 of the 47 events were captured merely to add context. For instance, the attackers used unauthenticated access to sensitive API to escalate the privileges of a compromised user account by adding the account to `Administrators`, `Domain Administrators`, and `Remote Desktop Users` groups. We captured an `WinEventLog` event merely to add context to the analysis by showing that the compromised user account was not part of these groups prior to the attack.

When gathering evidence of attacks from the dataset, we noted dependencies between vulnerabilities. For instance, a Team 1 attacker had to exploit the MongoDB vulnerability and then the cleartext credentials in chat history vulnerability to gain access to valid employee credentials (one of two valid employee credentials that Team 1 uncovered). We clustered these dependencies into scenarios with each scenario having a tangible end goal for the attacker. In each scenario, the attacker has to successfully exploit one or more vulnerabilities and/or other scenarios in sequence to achieve the goal.

We identified five scenarios into which the seven vulnerabilities of Team 1 could be clustered into. These scenarios, along

[1] https://attack.mitre.org/techniques/T1046/

with their constituent vulnerabilities and/or other scenarios, are as follows:

S1. Access to Valid Employee Credentials I: Unauthenticated remote administrator access to MongoDB and cleartext credentials in chat history.
S2. Unfettered Access to Active Directory Infrastructure: Blank domain administrator password or unauthenticated access to sensitive API and scenario #1.
S3. Access to Valid Employee Credentials II: Scenario #1, Scenario #2, and backup file on workstation.
S4. Potential for Eavesdropping: No SSL on sensitive web application.
S5. Access to Sensitive Internal Documentation: API documentation leaked on HeckForums.

Two authors independently mapped the 44 events (47 minus 3 events adding context) to ATT&CK tactics and techniques. The inter-rater reliability, measured by Cohen's κ, between the two authors was 0.4890 (p-value < 0.05) for ATT&CK tactics and 0.4370 (p-value < 0.05) for ATT&CK techniques. The inter-rater reliability score is regarded as *moderate agreement* by Landis and Koch [12].

In terms of the ATT&CK tactic, the two authors disagreed on the mapping of 16 of the 44 (36%) events, 4 (25%) of which were resolved by combining the ATT&CK tactic mapped by both authors. In terms of the ATT&CK technique, the two authors disagreed on the mapping of 19 of the 44 (43%) events, 8 (42%) of which were resolved by combining the ATT&CK technique mapped by both authors.

A common theme that emerged during the discussion of the disagreements was that mapping an event in isolation (i.e. disregarding the chronology of the events) often led to an incorrect mapping. For instance, a `linux_secure` event showed that a compromised user account was used to authenticate to a host in the network. In isolation, this event could be mapped as the attacker attempting initial access (tactic) using valid accounts (technique). However, if we use the knowledge that the attackers obtained the credentials to the compromised user account by exploiting another vulnerability, then we would map the event as the attacker attempting to move laterally (tactic) using valid accounts (technique). The role of subjectivity is quite pronounced and, therefore, there is a need to have at least two authors independently map the events to ATT&CK tactic and technique. In the future, we intend on having the disagreements be resolved by an expert well versed in penetration testing to further increase the validity of the mapping.

Once the disagreements were resolved, we chronologically ordered the ATT&CK tactics and techniques to provide an overview of the entire campaign of attack. Shown in Figure 1 is a flow diagram summarizing the approach, in terms of ATT&CK tactics and techniques, that Team 1 took to discover and exploit the vulnerabilities attributed to them. Each sequence of ATT&CK tactics and techniques in the diagram terminates at a tangible end goal for the attacker indicating the accomplishment of a scenario.

As seen in Figure 1, some vulnerabilities are trivial to attack, requiring a single step (often discovery using network service scanning) to discover. The characterization can not only inform the attacker mindset of engineers and administrators but also be used to reason about the extent of discoverability of vulnerabilities.

VII. Limitations

In this section, we highlight possible limitations of our work and the steps taken to alleviate some of them.

Internal Validity

There are two threats to internal validity in our work: completeness of the events dataset and methodological subjectivity.

The events dataset, though insightful, is not complete. There are certain actions of the attacker that is not represented by any event in the dataset. For instance, we know, from the report submitted, that Team 1 dumped a particular database after exploiting the MongoDB vulnerability. However, there is no event in the dataset to support this assertion. We overcame this limitation by using the information in the report that a team submitted to fill the gaps in the events dataset.

The approach we used to gather evidence from the events dataset to describe the timeline of an attack, being inherently manual, may have been affected by subjectivity. We could have had multiple authors independently gather evidence of an attack and compare the gathered evidence to assess the level of agreement (using an inter-rater reliability measure such as Cohen's κ). However, since the purpose of gathering evidence of an attack is to implicate an attacker beyond reasonable doubt, the subjectivity is unlikely to be a limitation. In other words, the evidence gathered must be sufficient enough to describe the approach an attacker used to discover and exploit a vulnerability but its exhaustiveness is not an essential property. However, as we expand our analysis to include vulnerabilities discovered and exploited by the remaining teams, we intend to have an actual penetration tester gather evidence of attacks to assess if the subjectivity in gathering evidence has a non-negligible impact on the resultant timeline of events.

External Validity

The attack narratives derived from the events dataset represent the modus operandi of students playing the role of attackers. We cannot claim that these narratives represent the approach of a real adversary because of two key distinctions between students and adversaries: (1) students were provided initial access to the infrastructure that an adversary may not have and, more importantly, (2) students were not constrained by concerns of any legal ramifications from their actions. The role of students having been provided initial access is not entirely relevant as Casey and Willis [13] highlight the present reality of employees and contractors (i.e. insiders) carrying out the most damaging attacks on businesses.

VIII. Related Work

The ontological coding aspect of our work is inspired by the concept of *attack narratives* proposed by Mireles *et al.* [3]. Mireles *et al.* [3] described attacks as narratives by

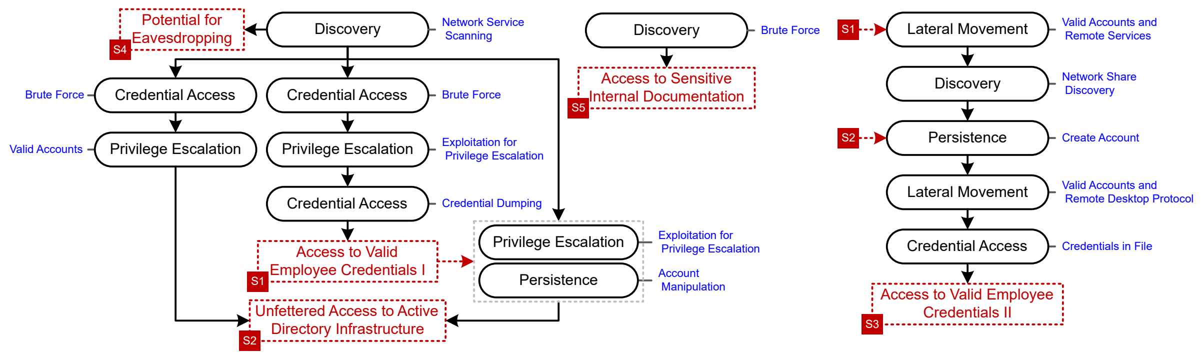

Fig. 1. Chronological flow diagram summarizing the approach that Team 1 took to discover and exploit vulnerabilities attributed to them. The chronology is depicted in terms of ATT&CK tactics (in black) and corresponding techniques (in blue) with each sequence terminating at a scenario (in red).

constructing flow models of attacks from network traffic. In this study, we examine events from a broad variety of sources on the attacker and victim hosts. With this richer data set, we gain a more detailed timeline of events. For instance, the dataset not only has information to characterize if an attacker used port scanning but also the specific port scanning utility (`nmap` or `unicornscan`) that the attacker used with all of the command line flags they specified. The dataset eliminates the need to *infer* attacker behavior from network traffic by actually *capturing* the attackers' actions.

IX. Summary

Our goal is to assist developers and administrators in inculcating an attacker mindset by proposing an approach to codify attacker behavior in cybersecurity penetration testing competition. We are in the process of analyzing a dataset of over 500 million events from six teams of attackers curated by Munaiah *et al.* [5] during the 2018 National Collegiate Penetration Testing Competition. We created detailed event timelines using the MITRE ATT&CK™ framework to characterize and describe attacker behavior in a standard way. In our preliminary analysis, we found 44 events to characterize the modus operandi of one team of attackers that discovered and exploited seven vulnerabilities.

While we demonstrate the feasibility of using empirical data to characterize attacker behavior, the analysis of vulnerabilities discovered and exploited by a *single team* is unlikely to yield generalizable insights about the behavior of a typical attacker. In our ongoing work, we are applying the methodology presented in this paper to characterize the behavior of the remaining teams of attackers. We envision the comparison of the modus operandi of multiple teams to reveal similarities in approach which can provide some generalizable insights on the typical ways in which weaknesses in software systems are discovered and exploited. We hope that such knowledge can inform the attacker mindset of engineers and administrators in their pursuit of engineering secure software systems.

Acknowledgments

Our work is supported in part by the DARPA Small Business Innovation Research program, the U.S. National Science Foundation CyberCorps® Scholarship for Service program (1433736), and the U.S. National Security Agency Science of Security Lablet at North Carolina State University (H98230-17-D-0080/2018-0438-02).

References

[1] J. M. Germain, "Open Source Is Everywhere and So Are Vulnerabilities, Says Black Duck Report," Accessed: June 10, 2019. [Online]. Available: https://www.linuxinsider.com/story/85338.html

[2] P. Ning and D. Xu, "Learning Attack Strategies from Intrusion Alerts," in *Proceedings of the 10th ACM Conference on Computer and Communications Security*, ser. CCS '03. New York, NY, USA: ACM, 2003, pp. 200–209. [Online]. Available: http://doi.acm.org/10.1145/948109.948137

[3] J. D. Mireles, J. Cho, and S. Xu, "Extracting Attack Narratives from Traffic Datasets," in *2016 International Conference on Cyber Conflict (CyCon U.S.)*, Oct 2016, pp. 1–6.

[4] B. E. Strom, J. A. Battaglia, M. S. Kemmerer, W. Kupersanin, D. P. Miller, C. Wampler, S. M. Whitley, and R. D. Wolf, "Finding Cyber Threats with ATT&CK™-Based Analytics," The MITRE Corporation, Tech. Rep., 2017. [Online]. Available: https://www.mitre.org/sites/default/files/publications/16-3713-finding-cyber-threats with att&ck-based-analytics.pdf

[5] N. Munaiah, J. Pelletier, S.-H. Su, S. J. Yang, and A. Meneely, "A Cybersecurity Dataset Derived from the National Collegiate Penetration Testing Competition," in *HICSS Symposium on Cybersecurity Big Data Analytics*, Nov 2019.

[6] Mandiant Consulting Services, "APT1: Exposing One of China's Cyber Espionage Units," FireEye, Inc., Tech. Rep., 2013. [Online]. Available: https://www.fireeye.com/content/dam/fireeye-www/services/pdfs/mandiant-apt1-report.pdf

[7] Lockheed Martin, "Cyber Kill Chain® — Lockheed Martin," Accessed: June 3, 2019. [Online]. Available: https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

[8] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, "MITRE ATT&CK™: Design and Philosophy," The MITRE Corporation, Tech. Rep., 2018. [Online]. Available: https://www.mitre.org/sites/default/files/publications/pr-18-0944-11-mitre-attack-design-and-philosophy.pdf

[9] H. Rodriguez, "`onetwopunch`," Accessed: June 5, 2019. [Online]. Available: https://github.com/superkojiman/onetwopunch

[10] J. Louis, "`unicornscan`," Accessed: June 5, 2019. [Online]. Available: https://linux.die.net/man/1/unicornscan

[11] G. Lyon, "`nmap`," Accessed: June 5, 2019. [Online]. Available: https://linux.die.net/man/1/nmap

[12] J. R. Landis and G. G. Koch, "The Measurement of Observer Agreement for Categorical Data," *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977. [Online]. Available: http://www.jstor.org/stable/2529310

[13] T. Casey and B. Willis, "Wargames: Serious Play that Tests Enterprise Security Assumptions," Intel Corporation, Tech. Rep., 2017. [Online]. Available: https://www.sbs.ox.ac.uk/cybersecurity-capacity/system/files/Intel - Wargames- Serious Play that Tests Enterprise Security Assumptions.pdf