Research Statement

Phuong Cao *(pcao3@illinois.edu)*

My research belongs to the area of data-driven security, an emerging field that is still in its infancy and whose theoretical foundations are just being formed. The goals are to design dependable host/network security monitors and validate those techniques in large-scale production networks against real intrusions. To achieve this goal, I will leverage mathematical methods from the areas of probabilistic graphical models and statistics.

Background and Current Work

Network intrusions cause data breaches [1] and misuses of the target system [2, 3], costing billions of dollars each year. Any evidence these intrusions leave is often scattered across the distributed hosts. Even worse, operators may misconfigure security monitors, which attackers actively evade, leading to excessive false alerts and fatigue while missing real alerts of an ongoing intrusion. At the early stage of an intrusion, both human and automated tools can not reliably distinguish between attack attempts and legitimate activities. Despite a rich body of academic research and industrial detection tools, recent data breaches show that most if not all of these intrusions are only discovered at the final stage, when the damages are irreversible, e.g., leaking of Social Security Numbers, DNA sequence, or abusing the power of computing resources.

To date, operators deploy a dense array of security monitors across the system/network infrastructure and feed events generated by those monitors to a security information event management system for generating alerts. First, operators continuously update these monitors with the latest attack signatures and detection policies to generate events of suspicious activities. Despite the maintenance of these monitors, they do not guarantee a thorough observation of all intrusion-related activities. Second, these host/network monitors do not exchange events, and thus each monitor only has a partial view of the intrusion. Current intrusion detection systems provide little or no guarantees in terms of accuracy and how early they can detect an intrusion. Finally, operators only have limited training exercises against real attacks with little to no cross-institution collaboration, leaving them insufficient time to react. All these issues impede advances in intrusion detection.

In my earlier work, I lay a theoretical framework for preemptive intrusion detection that has been validated in a production network [4]. This framework builds on insights derived from real-world incidents to enhance the reliability of security monitors, design a new detection algorithm based on probabilistic graphical models, and deploy a security testbed for reproducing scenarios that closely mimic real intrusions in the production network. My work has benefited from my close interaction and collaboration with the security team at the National Center for Supercomputing Applications (NCSA) and their production network.

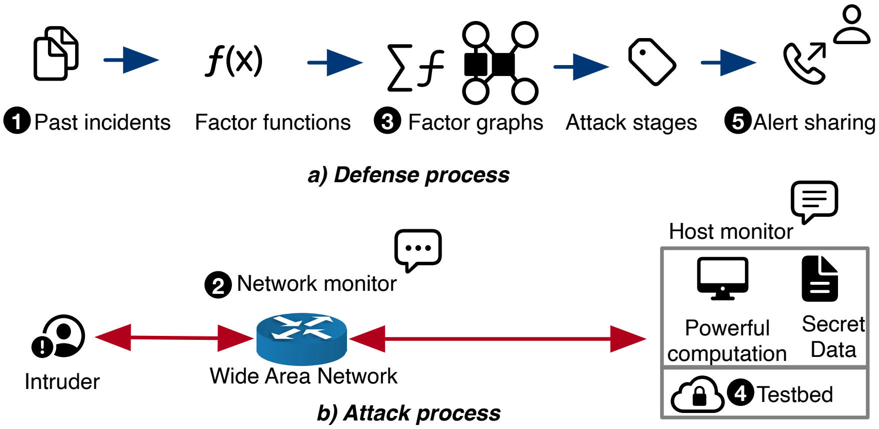

Figure 1: My Ph.D. proposal addresses research problems in the a) attack progress and the b) defense techniques.

Earlier PhD work

My Ph.D. proposal addresses research problems in i) how attacks progress and ii) the defense techniques employed by monitors. I describe my existing work and my proposal for each item in Fig.1 below.

1 Measurement Study of Security Incidents and Monitors [5]

Forensic reports of security incidents provide valuable insights regarding the cause, the progression, as well as how operators mitigate those incidents. Based on the analysis of data on past security incident reports at NCSA (Fig 1a), learning how attackers used unpublished toolkits to achieve their goals, and watching how operators respond. These attacks use sophisticated techniques to gain an initial access, ranging from using stolen legitimate credentials [6] to exploiting known remote code execution vulnerabilities. In some extreme cases, attackers have infiltrated deep into NCSA's computing infrastructure for an extended time and established a covert communication channel to receive malicious commands. The insights are three-fold. First, attackers can evade monitors, i.e., the monitor does not raise an alert on a slight modification of a known exploit code. Second, operators are overwhelmed by false alerts generated by those monitors. The ratio of false alerts vs. a real intrusion could go as high as a million to one. Third, these monitors only see a subset of events, but not the entire attack, because each monitor only focuses on a part of the system/network. Despite the fact that NCSA hosts expert developers, who constantly improve network security monitor (Bro/Zeek [7]) performance and its policies, these attacks often achieve their goals, e.g., stealing data, harvesting credentials, or sending spam emails, before the operators are notified. Thus, we need to improve the reliability and accuracy of these monitors.

2 Design of New Security Monitors [5]

To fix the weaknesses of the above monitors, I have built new monitors [8]: i) a new hypervisor-based (Hypertap [9, 10]) for host monitoring and a new honeypot (Caudit [5]) for network monitoring (Fig 1b). The hypervisor-based monitor is a kernel module that uses hardware architectural invariants, which cannot be modified by attackers and failures inside VMs. By running at the highest system privilege and using hardware architectural invariants [9], the Hypertap monitor records an attempt to escalate privilege and to hide malicious processes, typically found in root-kit — the most sophisticated type of malware. Therefore, Hypertap detects all real-world root-kit (tested in 2014) and avoids evasions that tamper with its monitoring events. The Caudit network monitor is a non-interactive honeypot that records Secure Shell (SSH) authentication attempts such as passwords or SSH key fingerprints. Caudit poses as a large-scale server farm of 65,536 machines (2^{16}) to attract attackers by being deployed on a /16 IP address space. Furthermore, Caudit mimics different SSH servers by randomizing its server fingerprints. Top attack attempts from Caudit are fed to a black hole router (BHR) deployed at NCSA. The BHR blocks these attempts at the network border (an average of 57 million attack attempts a day), thus significantly reduces the loads (up to 66×) on NCSA's internal security monitors.

3 Preemptive Intrusion Detection [11]

To preempt intrusions before data breaches and system misuses, I built PULSAR, a probabilistic graphical model (specifically Factor Graph — FG) based framework [12]. A FG is a compositional model that augments the prior knowledge with the temporal and the statistical relationships (in terms of factor functions) among observed events (Fig 1a). Using the constructed FGs at runtime, our approach performs real-time inference to find the current attack stage and to decide on an optimal response. Factor Graphs are suitable for modeling network intrusions. First, a FG's graphical model structure offers interpretability in terms of conditional dependencies among the observed events and attack stages. FGs encode these dependencies, e.g., the significance of events, using factor functions. Second, FGs have been proven in domains (e.g., healthcare) that have similar constraints like our attack preemption problem. While there are skeptics of using learning-based methods for IDS [13], we found that FGs work well for imbalanced data: our ground truth data (120 successful incidents in 10 years) is relatively small compared to hundred of thousands of unsuccessful attack attempts daily.

PULSAR infers the time evolution of an intrusion based on observed security events at runtime. The framework (i) learns the statistical significance of patterns of events from 120 past incidents; (ii) composes these patterns into FGs to capture the progression of the intrusion; and (iii) decides on preemptive actions. In our initial study, we uncovered six hidden intrusions — these were not detected by security analysts in post-incident forensic analysis. In testing with real intrusions unseen at NCSA, PULSAR stops 8 out of 10 replayed intrusions before system integrity violation, and all ten before data exfiltration. In a month-long production deployment of the framework, PULSAR took an average of one second to decide with a minimal (0.009%) false-positive rate.

4 Security Testbed [14, 15]

To validate the preemptive intrusion detection techniques above, we set up a security testbed embedded inside NCSA's infrastructure. The testbed's goals are to i) replay past intrusions in recorded incidents, ii) test intrusions unseen in NCSA production network, and iii) evaluate PULSAR's accuracy on attack attempts from the public Internet (Fig 1b). These goals are achieved as follows.

To simulate past incidents, I built a vulnerability reproduction tool (VRT) that creates old (vulnerable) Linux containers at any point in the past (2005–present). This allows the reproduction of an entire attack scenario that includes: i) an unpatched kernel/applications and their dependencies, ii) an attack container with exploit kits, and iii) system/network monitors. To test intrusions unseen in the production network, each vulnerable server is contained in a Linux container and further encapsulated in a QEMU virtual machine with limited capabilities. All containers used in our experiments ran in a network sandbox that implemented a Layer-3 private overlay network. Finally, I deployed the monitors and the detectors above in a physical machine located on the same rack of other legitimate machines in NCSA's production network. Using the testbed, any intrusion detection systems can be evaluated in realistic conditions.

Prelim Plan

My prelim plan is to build and to deploy an alert sharing network to detect coordinated attacks targeting distributed computing sites.

5 Alert Sharing Network

Recent coordinated attacks, exploiting critical vulnerabilities [16] or stolen credentials, have happened across multiple sites on a global scale [5]. For example, an attacker used a stolen credential at a site to access another site and attempted to steal research data [17]. While NCSA's security team has discovered and stopped this coordinated attack, very few institutions can afford such a nimble security team. This situation raises the need for an alert sharing network among sites to prevent such coordinated attacks.

To facilitate cross-site incident response and forensic analyses, I plan to extend our existing honeypot to support applications other than SSH. Our honeypot is being used in the bidirectional exchange of security-alert-related information with one site, the Singapore University of Technology and Design. The remaining sites are located in the U.S., e.g., the Pittsburgh Supercomputing Center, the Texas Advanced Computing Center, and Duke University. These sites, because of organizational policies, are only receiving unidirectional alerts from NCSA.

Goals. My goal is to investigate the trustworthiness and the performance guarantees of alert sharing techniques. The trustworthiness guarantees include the confidentiality, authenticity, and integrity of shared alerts against adversaries. These adversaries include man-in-the-middle attackers who may cause network congestion and ultimately impede real-time alert sharing and forensic investigations. The performance guarantees include statistical properties, say, guaranteeing less than 0.01% alerts loss on a threat sharing network, or deterministic, say, never more than a 1-second delay of alert sharing across sites on a threat sharing network. Furthermore, the sites can collaborate on learning the significance of each shared alerts, then incorporate those alerts in their decision-making process.

To achieve these goals, my approach is two-fold.

- First, I will deploy redundancy network routing techniques to send alert replicas across multiple network paths, thus maximizing the probability that the alert will be successfully delivered to receiving sites. This approach will address the problem of unreliable or late-arriving alerts because of network congestion.
- Second, I will implement distributed inference techniques on distributed FGs (DFG), a network of PGMs, to increase attack detection accuracy while still preserving the privacy of site-specific artifacts (e.g., internal IP addresses) in the shared alerts. This means each site has an instance of an FG to make its local decision. The significance of the DFG is that a site's local decision could be augmented by the information (alerts) and the remote belief (how critical is an alert) shared by other sites. This approach combines the local belief and the remote belief into a single PGM framework to make a better decision regarding an ongoing coordinated attack.

This alert sharing network will observe and characterize the latest attack activities across multiple network protocols such as VPN, TLS, SMTP, etc. Further, the observed attack attempts will be published into a data feed that other national and international sites (Fig 1a) can ingest.

References

[1] Seena Gressin. The equifax data breach: What to do. *US Federal Trade Commission, as viewed Oct*, 1, 2017.

[2] Rashid Tahir, Muhammad Huzaifa, Anupam Das, Mohammad Ahmad, Carl Gunter, Fareed Zaffar, Matthew Caesar, and Nikita Borisov. Mining on someone else's dime: Mitigating covert mining operations in clouds and enterprises. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pages 287–310. Springer, 2017.

[3] Cuong Pham, Phuong Cao, Zbigniew Kalbarczyk, and Ravishankar K Iyer. Toward a high availability cloud: Techniques and challenges. In *IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN 2012)*, pages 1–6. IEEE, 2012.

[4] Phuong Cao. *An experiment using factor graph for early attack detection*. PhD thesis, 2015.

[5] Phuong M Cao, Yuming Wu, Subho S Banerjee, Justin Azoff, Alex Withers, Zbigniew T Kalbarczyk, and Ravishankar K Iyer. {CAUDIT}: Continuous auditing of {SSH} servers to mitigate brute-force attacks. In *16th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 19)*, pages 667–682, 2019.

[6] Phuong Cao, Hongyang Li, Klara Nahrstedt, Zbigniew Kalbarczyk, Ravishankar Iyer, and Adam J Slagell. Personalized password guessing: a new security threat. In *Proceedings of the 2014 Symposium and Bootcamp on the Science of Security*, page 22. ACM, 2014.

[7] Vern Paxson. Bro: a system for detecting network intruders in real-time. *Computer networks*, 31(23-24):2435–2463, 1999.

[8] Shuo Chen, Matt McCutchen, Phuong Cao, Shaz Qadeer, and Ravishankar K Iyer. Svauth–a single-sign-on integration solution with runtime verification. In *International Conference on Runtime Verification*, pages 349–358. Springer, 2017.

[9] Cuong Pham, Zachary Estrada, Phuong Cao, Zbigniew Kalbarczyk, and Ravishankar K Iyer. Reliability and security monitoring of virtual machines using hardware architectural invariants. In *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pages 13–24. IEEE, 2014.

[10] Cuong Pham, Zachary J Estrada, Phuong Cao, Zbigniew Kalbarczyk, and Ravishankar K Iyer. Building reliable and secure virtual machines using architectural invariants. *IEEE Security & Privacy*, 12(5):82–85, 2014.

[11] Phuong Cao, Eric Badger, Zbigniew Kalbarczyk, Ravishankar Iyer, and Adam Slagell. Preemptive intrusion detection: Theoretical framework and real-world measurements. In *Proceedings of the 2015 Symposium and Bootcamp on the Science of Security*, page 5. ACM, 2015.

[12] Phuong Cao. On preempting advanced persistent threats using probabilistic graphical models. *ArXiv*, abs/1903.08826, 2019.

[13] Robin Sommer and Vern Paxson. Outside the closed world: On using machine learning for network intrusion detection. In *2010 IEEE symposium on security and privacy*, pages 305–316. IEEE, 2010.

[14] Phuong Cao, Eric C Badger, Zbigniew T Kalbarczyk, Ravishankar K Iyer, Alexander Withers, and Adam J Slagell. Towards an unified security testbed and security analytics framework. In *Proceedings of the 2015 Symposium and Bootcamp on the Science of Security*, page 24. ACM, 2015.

[15] Phuong Cao, Eric C Badger, Zbigniew T Kalbarczyk, and Ravishankar K Iyer. A framework for generation, replay, and analysis of real-world attack variants. In *Proceedings of the Symposium and Bootcamp on the Science of Security*, pages 28–37. ACM, 2016.

[16] Zakir Durumeric, Frank Li, James Kasten, Johanna Amann, Jethro Beekman, Mathias Payer, Nicolas Weaver, David Adrian, Vern Paxson, Michael Bailey, et al. The matter of heartbleed. In *Proceedings of the 2014 conference on internet measurement conference*, pages 475–488. ACM, 2014.

[17] Keywhan Chung, Phuong Cao, Yuming Wu, Zbigniew T Kalbarczyk, Ravishankar K Iyer, and Alexander Withers. Traction: an infrastructure for trusted alert sharing and collaborative mitigation. In *Proceedings of the 6th Annual Symposium on Hot Topics in the Science of Security*, page 27. ACM, 2019.