Network Modeling for Security Analytics

Bailey Smith
The University of Tulsa
bailey-smith@utulsa.edu

Whitney Caruthers
The University of Tulsa
wbc782@utulsa.edu

Dalton Stewart
The University of Tulsa
dalton-stewart@utulsa.edu

Dr. Peter Hawrylak
The University of Tulsa
peter-hawrylak@utulsa.edu

Dr. John Hale
The University of Tulsa
john-hale@utulsa.edu

ABSTRACT

Comprehensive network modeling remains a challenge for the security analyst. Complete coverage and depth of detail in network models is difficult to achieve for large and complex networks, especially when significant effort in manual elaboration is required. This paper describes an automated approach to network model acquisition using coordinated bump-in-the-wire devices. The system described here relieves a substantial burden from the modeler while offering improved visibility over competing solutions.

CCS Concepts

•Security and privacy → Vulnerability scanners; Intrusion detection systems; Network security;

Keywords

Network Model, Active Monitoring

1. INTRODUCTION

Capturing the elements in a network, their properties and relationships, is a vital aspect of network management and security. The information held in a network model can be used for many purposes, from performance analysis to security modeling to risk assessment. However, the utility of such models depends entirely on their completeness, accuracy and level of detail. Hand-generation of network models is only plausible for small systems. Modeling of large networks requires a significant degree of automation. Manageability of an automated solution is also a key to practical network modeling for large networks.

In this paper a system, PINDAQ (PI Network Data Acquisition) for automated network modeling that is simple to deploy and offers increased visibility using bump-in-the-wire devices coordinating active monitoring processes is presented. This approach allows administrators and analysts

to observe all connected systems and their state, along with running applications and open ports. As such, these capabilities are ideally suited for an automated modeling solution designed to support vulnerability analysis.

The remainder of this paper is organized as follows. In Section 2 presents related work. Section 3 presents network model representation and the design of the passive network modeling device, PINDAQ. Section 4 presents applications that use the information from the passive network modeling device, and Section 5 presents conclusions and future work.

2. BACKGROUND

This section discusses network modeling for security analytics. It presents, as a foundation for network modeling, related work in passive and active monitoring.

2.1 Network Models for Security Analytics

Security analytics necessitate system representations modeling security-relevant properties. The focus of [13] is on analyzing the vulnerabilities in a network. Their first step is creating a configuration file holding all the information about their network and each device in the network. In [14] the authors use model checking to drive network security analysis, relying on a network modeling scheme to capture network state information. Similarly, [5] proposes a formal network model for attack graph generation. Common among these efforts is the need to express connection and trust relationships between network elements, as well as relevant configuration and service information for hosts and devices. The PINDAQ system presented in this work helps to automate this process and provides features to simplify periodic updates of a network model.

2.2 Passive Monitoring

Passive monitoring is the capture and analysis of traffic on a network absent packet insertion or modification [2]. The utility of monitoring for network security analysis is well-established [7, 8, 10, 16]. It can be used to infer network topology, reveal active services and fingerprint host operating systems. Packet capture and protocol analysis tools such as Ettercap [12] and Wireshark [3] offer low level passive monitoring capabilities. SNMP [15] supports network device management and analysis through passive observation.

Passive monitoring is a minimally invasive technique that limits the potential for adverse impact on network perfor-

Permission to make digital or hard copies of all or part 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 components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CISRC '16, April 05-07, 2016, Oak Ridge, TN, USA

© 2016 ACM. ISBN 978-1-4503-3752-6/16/04...\$15.00

DOI: <http://dx.doi.org/10.1145/2897795.2897817>

mance. However, passive monitoring solutions often struggle to recognize applications that use non-standard port numbers, resulting in the incorrect classification of traffic. Moreover, it may offer limited visibility of a network.

Louthan, *et. al.*, [8] describe a tool, Stream-Aware Network-protocol Detector (SAND), to identify traffic without relying on the port number. SAND is a passive monitoring architecture designed to develop detailed observations of the network's traffic. Moore and Papagiannaki [11] discuss the difficulties faced by passive network traffic identification based on port numbers in more detail. They motivate this kind of identification by underscoring the importance of knowing applications or network resources that are being used incorrectly or maliciously. PINDAQ also employs passive monitoring and works to address some of these difficulties.

2.3 Active Monitoring

Active monitoring engages network discovery by sending packets on a computer network to elicit responses for analysis. In essence, it creates network traffic whereas passive monitoring analyzes the traffic that already exists on the network. One advantage of active monitoring over passive monitoring is that the former confers more control over the monitoring process and can be directed to investigate specific aspects of network behavior and function.

NMAP [4] is an active monitoring tool that scans and probes networks to generate information related to vulnerabilities in the network. NMAP sends packets to identify active ports on a system, and generates operating system information by analyzing the TCP/IP stack. As a foundational technology for network monitoring, NMAP has been applied to and extended for use in a variety of network security toolchains. For example, [9] adopts NMAP to coordinate active monitoring on a large scale with operating system detection and host fingerprinting features. We will be using NMAP as a tool to develop our network models. The information provided by NMAP will be an important piece of the overall network model as it will allow us to determine what operating systems are running on computers connected to the network and other information associated with the network components.

3. NETWORK MODELS

Network models constitute a useful representation of a network, potentially comprised of both high and low level data. At the highest level, a model typically stores the connections between machines, the trust relationships between them, as well as the presence of intrusion detection systems, firewalls, and other objects that can affect the exploitation of a network. At a lower level, a model may contain local information, such as applications that could be exploited, open ports, and user accounts. A comprehensive model must show all relationships between machines, every level of trust that can be exploited by an attacker, and all qualities, such as operating systems, programs, and version numbers, of the individual machines in the network that can be exploited. Naturally, concerns of practicality in modeling dominate, and intelligent selection of what to admit and what to elide can make all the difference in model utility.

3.1 Representation

The information gained by developing the network model is stored in a database. This includes the applications run-

ning on each machine, trust relationships, and open ports. The textual representation of the network model includes categories for assets, facts, and topologies. Assets are all the electronic systems in the network, for example, computers, routers, and switches. The facts are the qualities of each of the systems, including programs running on the system and their version numbers. The topology represents the disjointed sets of information obtained by separate PINDAQs, and each topology shows each asset's relation with other assets within that set of systems.

Figure 1 (left) contains an example network with the textual representation its network model shown in **Figure 1** (right). Each electronic system in the network model is contained in the textual representation as assets with specific information included as qualities. For example, Workstation1 is an asset running Windows 7, Adobe Reader Version 1.2, and Internet Explorer version 11. The connections between systems are shown under topology. For example, Switch2 connects to the router, Workstation3, and Workstation4.

3.2 Acquisition

There are several ways of generating this information, including manual creation of the entire model. Of course, as this would be very time consuming, error prone, and most likely incomplete, it makes far more sense to use automated tools. This can include tools running on a network, watching traffic to identify machines and traffic patterns, as well as local tools that can report on running services and local users. In all likelihood, the final product will be a comprehensive toolkit containing aspects of both local and network monitoring.

PINDAQ is built on the the Raspberry PI 2 Model B platform [1]. Network mapping and monitoring tools to develop the network model will be installed on the Raspberry PI. The goal is not to be stealthy which expands the number of tools can be used to collect information for the network model generation. Instead, PINDAQ is designed to be a tool that network owners can use to monitor and map their networks. If left connected for long enough and deployed appropriately, the PINDAQs will collect all the information about the network. The information collected is concentrated into a database and then used to generate the model of the network.

PINDAQ works by using common tools to create an in-depth topology for the network. Using NMAP, each PINDAQ can scan a network to build a simple topology to serve as a framework for future use. After that, the OpenVAS vulnerability scanner is employed to find vulnerabilities within the network. By using multiple PINDAQs, all running the same software suite, they will be able to map the network from multiple perspectives. After each PINDAQ has completed its scans, the MicroSD cards are removed from the nodes to add the data to the network database. If needed, new microSD cards containing new updates to the software suite can be added to keep the information as up-to-date.

3.3 Synthesis and Integration

The use of multiple PINDAQs can increase the speed and accuracy of model acquisition. However, using multiple PINDAQs poses a few problems, such as a PINDAQ obtaining information duplicating information of one of the other PINDAQs, insufficient information, or erroneous information. Ultimately, collaboration between PINDAQs is necessary to

Figure 1: Example Network Model ©Bailey Smith

avoid and remove such errors in order to allow for the generation of a complete and accurate network model.

Duplicate information is created if two PINDAQs reside on the same disjointed set of network. They generate the same set of information, but it would seem as if it was two different parts of the network because the information came from separate PINDAQs. This would create the same set of information for each PINDAQ. Another duplication of information would occur if there were two PINDAQs on different parts of the network, but they could each see one area of the network in between them. Then only part of the information generated would be the same. It would seem as if there were two disjointed parts of the network that had some of the same information. If either of the ways to generate duplicate information does not have a way to filter out duplications, then an incorrect network model will result.

Insufficient information can be caused by not having enough PINDAQs connected to the network, or not having them in the correct places. For example, **Figure 1** (left) shows a network containing 2 PINDAQs, PINDAQ1 and PINDAQ2. Without PINDAQ2, information about Switch2, Workstation3, and Workstation4 would be lost. In this case, PINDAQ1 would report a complete scan, yet yield an incomplete network map because it cannot see past Switch2. This example shows how not having enough PINDAQs can lead to insufficient information, but it can be fixed by adding another PINDAQ, or by moving the one remaining PINDAQ to a different part of the network. One solution is to connect PINDAQ1 to the Router, and other is to add PINDAQ2 back to Switch2. Either option will allow the mapping of the complete network. It is important to make sure there is a PINDAQ connected to each part of the network, to obtain a complete picture of the network.

Erroneous information would result when two PINDAQs scanned the same asset on the network and received different information. For example, one decides the operating system is Ubuntu Server and the other says it is a Linux based operating system. Although this information is not in direct conflict, they still differ. It would be important to determine whether the PINDAQs were actually scanning the same computer (e.g., if the computer were running multiple virtual machines), and, if they were, which information is the most accurate. This is important to allow for the identification and removal of items that are scanned by multiple PINDAQs.

Therefore, the PINDAQs will establish connections between themselves and merge their individual databases, so that information gathered can be appropriately analyzed, sorted, and, if required, discarded. This solution will help eliminate local redundancy and inaccuracies as well as increase the efficiency in creating an accurate network model.

4. APPLICATIONS

This section presents applications of the PINDAQ system to secure networks.

4.1 Attack Graph Generation

One use for the generated network models is attack graph generation. The generation of attack graphs requires a network model as the initial system state to begin the generation. Each exploit in a database is then checked to see if it applies to the initial state. If the conditions for the exploit to apply, preconditions, match the network model then the exploit is applied and generates a new state with the conditions that result from the exploit, postconditions, applied to the initial state. Each new state created then contin-

ues the process, generating all the resulting states from the recently generated state. In this way, the generation of network models is extremely useful as the initial network model needed to construct the attack graph can be generated easily using PINDAQ. Attack graph generation relies upon the initial network model being accurate and complete which is difficult to accomplish if the network model was generated by hand, especially for large networks. The network model created by PINDAQ is accurate and easier to generate than manually creating the network model. This will increase the usefulness and applicability of attack graphs which is currently limited because generation of graphs over large networks is commonly impractical.

When a network changes, previously, a completely new network model would have to be generated. Now, the PINDAQ, if still connected to the system, would collect the information and place it in the database with the other network information. Then the network model could be updated incrementally and automatically.

4.2 IDSs and IPSs

An Intrusion Detection System (IDS) monitors network activity to determine if malicious or abnormal activity is occurring [6]. The integration of a solution like PINDAQ into an IDS potentially improves the accuracy of an IDS. Supplied with a PINDAQ generated network model, an IDS could recognize and assess changes on the system, notifying an administrator if they were potentially malicious. This has the greatest prospective benefit in identifying and mitigating multi-stage attacks.

Like an IDS, an Intrusion Prevention System (IPS) aims to protect a network by monitoring network and user activity [17]. An IPS works to prevent malicious activity, whereas an IDS detects whether malicious activity is occurring. By integrating PINDAQ generated network models the IPS would be able to recognize if the actions occurring could be malicious. The IPS would know the current network model and recognize when the elements of the network change. If the changes are malicious, or seem malicious, the IPS would then take steps to prevent them from continuing.

5. CONCLUSIONS AND FUTURE WORK

Automatic generation of network models is very useful in network management and security. The PINDAQ system provides an automated and distributed method to collect data about a network in order to generate a model of it. This information can feed several types of network security tools. We plan to complete our implementation of PINDAQ on the Raspberry PI platform and use it to support network modeling for Attack Graph generation. The network models provided by PINDAQ will also be useful in Intrusion Detection Systems, allowing for the IDS to more accurately detect attack on networks as a whole, and to maintain a current picture of the network. PINDAQ will also assist Intrusion Prevention Systems in detecting malicious activity.

6. ACKNOWLEDGEMENT

This material is based on work supported by the National Science Foundation under Grant No. 1524940. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

7. REFERENCES

- [1] Raspberry pi. <https://www.adafruit.com/raspberrypi>.
- [2] G. Bartlett, J. Heidemann, and C. Papadopoulos. Understanding passive and active service discovery. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 57–70. ACM, 2007.
- [3] G. Combs et al. Wireshark-network protocol analyzer. *Version 0.99*, 5, 2008.
- [4] T. Corcoran. An introduction to Nmap. Technical report, 2001.
- [5] J. Dawkins and J. Hale. A systematic approach to multi-stage network attack analysis. In *Information Assurance Workshop, 2004. Proceedings. Second IEEE International*, pages 48–56. IEEE, 2004.
- [6] D. E. Denning. An intrusion-detection model. *Software Engineering, IEEE Transactions on*, (2):222–232, 1987.
- [7] L. P. Gaspar, R. N. Sanchez, D. W. Antunes, and E. Meneghetti. A SNMP-based platform for distributed stateful intrusion detection in enterprise networks. *Selected Areas in Communications, IEEE Journal on*, 23(10):1973–1982, 2005.
- [8] G. Louthan, C. McMillan, C. Johnson, and J. Hale. Toward robust and extensible automatic protocol identification. In *International Conference on Internet Computing*, pages 104–108, 2009.
- [9] G. W. Manes, D. Schulte, S. Guenther, and S. Shenoi. Netglean: A methodology for distributed network security scanning. *Journal of Network and Systems Management*, 13(3):329–344.
- [10] D. Montigny-Leboeuf, F. Massicotte, et al. Passive network discovery for real time situation awareness. Technical report, DTIC Document, 2004.
- [11] A. W. Moore and K. Papagiannaki. Toward the accurate identification of network applications. In *Proceedings of the 6th International Conference on Passive and Active Network Measurement, PAM’05*, pages 41–54, Berlin, Heidelberg, 2005. Springer-Verlag.
- [12] A. Ornaghi and M. Valleri. Ettercap, 2005.
- [13] C. Phillips and L. P. Swiler. A graph-based system for network-vulnerability analysis. In *Proceedings of the 1998 Workshop on New Security Paradigms, NSPW ’98*, pages 71–79, New York, NY, USA, 1998. ACM.
- [14] R. W. Ritchey and P. Ammann. Using model checking to analyze network vulnerabilities. In *Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on*, pages 156–165. IEEE, 2000.
- [15] W. Stallings. *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*. Addison-Wesley Longman, 1998.
- [16] Y. Thomas, H. Debar, and B. Morin. Improving security management through passive network observation. In *Proceedings of Availability, Reliability and Security, 2006*, pages 382–389. IEEE, 2006.
- [17] X. Zhang, C. Li, and W. Zheng. Intrusion prevention system design. In *Proceedings. The Fourth International Conference on Computer and Information Technology*, pages 386–390. IEEE, 2004.