POSTER: Dragging Attackers to Honeypots for Effective Analysis of Cyber Threats

Martin Husák and Jan Vykopal
Institute of Computer Science
Masaryk University, Brno, Czech Republic
husakm@ics.muni.cz, vykopal@ics.muni.cz

***Abstract*—With the rising number of cyber threats in communication networks, there is a demand for attack analysis and the identification of new threats. Honeypots, tools for attack analysis and zero-day exploit discovery, are passive in waiting for an attacker. This paper proposes a novel approach to the effective utilization of honeypots based on cooperation between honeypots and the network in which they are deployed. We propose a framework for recognition of attacks in their early phase and dragging the network traffic to a honeypot before the attack causes any harm. We use flow-based network monitoring to detect initial phases of the attacks and propose prediction of the later phases of the attack. Malicious network traffic will be redirected to a honeypot for further analysis using a concept of a network funnel.**

I. Introduction

The number of cyber threats is rising with new exploits and attacks being discovered on a daily basis. To set up proper countermeasures we need to identify and analyze the threats. Honeypots, whose sole task is to be probed or compromised, are well-known tools for attack analysis [1]. The shortcomings of honeypots include passive waiting for an attack to appear and a lack of production-level monitoring [2].

In this paper we discuss contemporary challenges for server-side, general purpose honeypots [1] and state current research questions. We propose tighter cooperation between honeypots and the network in which they are deployed. Network monitoring enables us to further analyze attacks against honeypots and identify events preceding the actual attack. The monitoring of the entire network, not only honeypots, helps to understand how the attacker finds a victim. We can learn about the events preceding the attack to predict it or to improve attractiveness of the honeypot to an attacker.

We are seeking a solution for detection and prevention of cyber attacks. We split an attack in the initial phase and intrusion phase. Initial phase consists of victim discovery and attack preparation. Intrusion phase is the attack itself. We try to predict the intrusion phase based on the detection of the initial phase. We should be able to detect the malicious traffic before it makes any harm and redirect it to honeypot. The redirection of suspicious network traffic prevents the attacker from accessing the production network and supplies the honeypot with malicious network traffic for further analysis. The attacker may interact with honeypot without limitations while the production network is safe and we can analyze the attack.

II. Research Questions

To enhance the utilization of honeypots and increase the chances of discovering new cyber threats, we need to identify weak spots in honeypot deployment and understand the attack procedure. We have identified three research questions that we present with a brief commentary on each. In the following section, we discuss research approach and a proposed solution to them.

(i) *How does an attacker search for targets?* We need to map the attack vector, the ways an attacker learns about new targets and picks the target to attack. Knowing the enemy's next move is the first step in taking appropriate countermeasures.

(ii) *How can we identify the attacker early and predict the attack?* Once the attack vectors are mapped, we can detect events preceding the attack and predict the intrusion phase of the attack. Formal description and modeling of the attack may significantly improve the detection and prediction capabilities.

(iii) *How can we prevent the attack and still be able to analyze it?* Common security measures, e.g., blocking, prevents the attacker from accessing the network but also prevents an analysis of the attack. The redirection of the malicious traffic to the honeypot for further analysis is a suitable solution.

III. Proposed Approach

Our proposed approach relies on advanced network monitoring and network management. Network monitoring is at first utilized in the mapping of the attack vectors and later in the early identification of malicious network traffic. The desired result of our work is a network setup that detects the attacker during network reconnaissance and prevents the attacker from accessing the production network by redirecting the malicious network traffic to the honeypot.

A. Mapping the Attack Vector

To map the attack vector, we need to analyze an attack and identify the preceding activity in the network that is linked to the attack. We do not need to rely only on honeypots, any observed intrusion is suitable for analysis. The formal model of the attack vector will be used to describe observed events. The desired form of attack modeling is a causal network [3] that shows which observable events lead to an intrusion. Known example is a scanning phase of brute-force attacks that can be detected separately [4], [5].

We propose using network monitoring based on network flows, namely NetFlow [6] and IPFIX [7]. A common attack preparation observable via flow monitoring involves ping sweep and horizontal and vertical network scanning. An interesting network reconnaissance technique involves DNS queries. The attacker asks for reverse DNS records of IP addresses to spot potential targets [8]. The attacker can easily avoid honeypots because they are not always deployed in the address space with assigned domain names. We consider a DNS query for an IP address of a honeypot as suspicious as a connection attempt to a honeypot itself. While pure flow monitoring is unsuitable and DPI is resource demanding, we propose using extended flow monitoring. IPFIX will be used for the monitoring of suspicious DNS queries.

Mapping of the attract vector is closely tied to honeypot advertising. The problem is how to attract the attackers to access a honeypot. Common techniques of advertising honeypot involve assigning domain names, running various services or providing eye-catching content [2]. We have to consider successful attempts at attracting attackers to a honeypot as a source of data for attack vector mapping.

B. *Early Detection*

We plan to automatically match network traffic against the formal models of the proposed attack vectors. Causal networks will be used to formally describe the procedure of the attacks. The progress of the potential attacker will be matched against a causal network to recognize and predict the attack [3], [9]. We should be able to predict the attack and redirect network traffic from the attacker to the honeypot before the intrusion phase of the attack. The attack will target a honeypot while the production network will be safe.

The attack recognition and prediction mechanisms will process the data provided by network monitoring and network intrusion detection tools. We can effectively use the existing network anomaly or intrusion detection tools to provide necessary data. Network anomaly and initial attack phase detection tools will suggest potential attackers.

To emphasize the difference between our proposal and existing solutions such as Shadow Honeypots [10], we use honeypots as both an intrusion detection system and intrusion analysis tool. In our previous work we presented the monitoring of honeypots using network flows [11]. We use a key assumption of network-based monitoring of honeypots; in that any network traffic incoming to honeypots is by nature suspicious [1]. The access to a honeypot will be one of the reasons to redirect all the traffic from the attacker. Once the attacker accesses the honeypot, any other connection attempt will end up at it.

Although honeypots are generally considered free of false positives, we have to be cautious in network monitoring. Spoofed network traffic needs to be avoided as we have observed during distributed reflected denial of service attack, where honeypots were abused as reflectors of spoofed flooding traffic [12]. From the point of view of honeypots it was evaluated as TCP SYN scanning from the IP address that was actually a victim of a DRDoS attack. On the other hand, the traffic was part of an attack so it does not contradict the key assumption of honeypot monitoring.

C. *Network Funnel*

To increase the amount of network traffic coming to honeypots, we propose a concept of a network funnel. This concept deals with both dragging the attackers to honeypots and preventing them from accessing the protected network. Our goal is to redirect the malicious network traffic to honeypot transparently so that the attacker is not aware of it. Redirection prevents the attacker from interacting with hosts in the production network while the honeypot responds to any traffic from the attacker. The network funnel is inspired by the redirection of network traffic destined to unassigned IP addresses, i.e., darkspace. The main difference is that we redirect the traffic originally destined to legitimate hosts, not the unassigned address space.

Unlike projects such as Shadow Honeypots [10], we do not need to create new honeypots as a shadow copy of a production network. The goal is to drag more malicious traffic to existing honeypots to fully utilize the existing infrastructure and resources. We focus on honeypots that already face the public Internet to cover their current network traffic as well as redirected traffic. The other difference from existing solutions is our aim at protecting large network infrastructures. We do not have capabilities to provide a shadow copy of the whole network.

The traffic manipulation can be achieved by enhancing the capabilities of Software Defined Networking (SDN). Projects like AVANT-GUARD [13] are providing a layer upon SDN that is supposed to add network features usable in security enforcement, e.g., the redirection of malicious traffic. However, we are looking for a solution that does not depend on SDN or any other advanced framework. We prefer an undemanding and portable solution based on GRE tunnels [14] between edge routers and honeypots.

IV. Conclusion

In summary, we presented research questions and outlined an approach to solving problems regarding honeypots in today's networks. We proposed the utilization of network monitoring to extend data sources for attack analysis and to identify malicious network traffic early. We proposed a concept of a network funnel in which the malicious traffic is redirected to the honeypots. The redirection is triggered by a cyber attack prediction and prevents the attacker from accessing the production network while the honeypots have a chance to analyze the intrusion.

Our future work includes mapping the attack vector, formalizing the progress of intrusion and identifying patterns in network traffic that precede the intrusion. Once we detect an event commonly preceding intrusion, we can prevent it by redirecting the malicious traffic to a honeypot for further analysis.

Acknowledgment

This paper is supported by the Czech Ministry of Interior under Identification code VF2013201531.

References

[1] N. Provos and T. Holz, *Virtual Honeypots: From Botnet Tracking to Intrusion Detection*, 1st ed. Addison-Wesley Professional, 2007.

[2] European Network and Information Security Agency (ENISA), "Proactive Detection of Security Incidents II - Honeypots," http://www.enisa.europa.eu/activities/cert/support/proactive-detection/proactive-detection-of-security-incidents-II-honeypots, 2012.

[3] X. Qin and W. Lee, "Attack plan recognition and prediction using causal networks," in *Computer Security Applications Conference, 2004. 20th Annual*, Dec 2004, pp. 370–379.

[4] L. Hellemons, L. Hendriks, R. Hofstede, A. Sperotto, R. Sadre, and A. Pras, "SSHCure: A Flow-Based SSH Intrusion Detection System," in *Dependable Networks and Services*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, vol. 7279, pp. 86–97.

[5] J. Vykopal, "A Flow-Level Taxonomy and Prevalence of Brute Force Attacks," in *Advances in Computing and Communications*, ser. Communications in Computer and Information Science. Springer Berlin Heidelberg, 2011, vol. 191, pp. 666–675.

[6] B. Claise, "Cisco Systems NetFlow Services Export Version 9," RFC 3954 (Informational), October 2004.

[7] B. Claise, B. Trammell, and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information," RFC 7011 (INTERNET STANDARD), Internet Engineering Task Force, Sep. 2013. [Online]. Available: http://www.ietf.org/rfc/rfc7011.txt

[8] J. Oberheide, M. Karir, and Z. M. Mao, "Characterizing Dark DNS Behavior," in *Proceedings of the 4th international conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, ser. DIMVA '07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 140–156.

[9] X. Wei-wei and W. Hai-feng, "Prediction model of network security situation based on regression analysis," in *Wireless Communications, Networking and Information Security (WCNIS), 2010 IEEE International Conference on*, June 2010, pp. 616–619.

[10] K. G. Anagnostakis, S. Sidiroglou, P. Akritidis, M. Polychronakis, A. D. Keromytis, and E. P. Markatos, "Shadow honeypots," *International Journal of Computer and Network Security*, vol. 2, no. 9, 2010.

[11] M. Husák and M. Drašar, "Flow-based Monitoring of Honeypots," in *Security and Protection of Information 2013*. Brno: Univerzita obrany, 2013, pp. 63–70.

[12] M. Husák and M. Vizváry, "POSTER: Reflected Attacks Abusing Honeypots," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, ser. CCS '13. New York, NY, USA: ACM, 2013, pp. 1449–1452.

[13] S. Shin, V. Yegneswaran, P. Porras, and G. Gu, "AVANT-GUARD: Scalable and Vigilant Switch Flow Management in Software-defined Networks," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, ser. CCS '13. New York, NY, USA: ACM, 2013, pp. 413–424.

[14] S. Hanks, T. Li, D. Farinacci, and P. Traina, "Generic Routing Encapsulation over IPv4 networks," RFC 1702 (Informational), Internet Engineering Task Force, Oct. 1994. [Online]. Available: http://www.ietf.org/rfc/rfc1702.txt