Honeypot Testbed for Network Defence Strategy Evaluation

Jana Medková*†, Martin Husák*†, Martin Vizváry*, Pavel Čeleda*
*Institute of Computer Science, †Faculty of Informatics
Masaryk University, Brno, Czech Republic
{medkova, husakm, vizvary, celeda}@ics.muni.cz

***Abstract*—In this paper, we describe a network defence strategy testbed, which could be utilized for testing the strategy decision logic against simulated attacks or real attackers. The testbed relies on a network of honeypots and the high level of logging and monitoring the honeypots provide. Its main advantage is that only the decision logic implementation is needed in order to test the strategy. The testbed also evaluates the tested network defence strategy. We demonstrate an example of network defence strategy implementation, the test setup, progress, and results. The source code of the testbed is available on GitHub.**

I. Introduction

The evaluation of network defence strategy is extremely demanding. Just the setup and evaluation of a strategy in a simulated environment takes a lot of effort. Evaluation on a test network is even more difficult. Even though we usually need to test only the strategy decision logic, all the components must be deployed. This includes automated attacks, detection methods, and tools to carry out the response selected by the tested strategy.

In this paper, we present our testbed for evaluation of network defence strategies. It acquits us of setting up the detection and response execution tools, only the strategy decision logic must be implemented. We utilize a honeynet deployed in KYPO, a cloud-based testbed for simulating cyber attacks [1]. The utilization of honeypots gives us an advantage of detailed logging and monitoring since their original purpose is attack analysis. This allows us to effortlessly and reliably detect ongoing attacks and estimate their impact on services.

The tested strategy could be evaluated either in a simple scenario against simulated attacks or against real attackers. For evaluation with simulated attack, the testbed contains a component capable of carrying out the attacks. However, the honeynet is also attacked on a regular basis by a wide variety of attackers, which allows us to analyse its efficiency in a more realistic scenario. The basic setup can be extended by using more honeypots or honeypots with different services.

The testbed described in this paper was already utilized in an experiment [2]. In the experiment, the results of evaluation of a strategy in a simulated environment were compared to results of an evaluation in a real network against real attackers.

II. Testbed Description

Our testbed for network defence strategy evaluation is deployed in a sandbox in KYPO, a cloud-based framework for simulating network scenarios. KYPO allows us to easily setup a virtual network through a web portal.

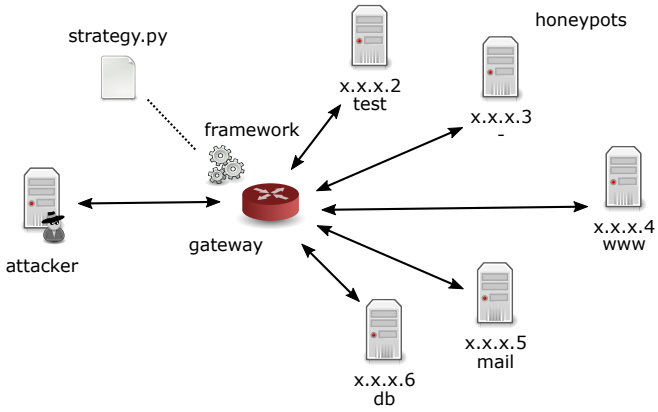

Fig. 1: Testbed Topology

We limit the types of attacks against the tested strategies to SSH brute-force attacks only. We use five high-interaction honeypots placed in the same subnet and assigned consequent IP addresses (Figure 1). Some of the honeypots are assigned a domain name to lure potential attackers. The domain names (www, mail, db) suggest a role of each honeypot so that the attacker can select his target according to his objective. Each honeypot is running SSH server with password authentication and responds to authentication attempts. All attempts on authentication are logged into a central database. The entries in the database contain the information about the attacker's IP address, honeypot's IP address, a timestamp of the attempt and the combination of username and password.

The network traffic of the honeynet flows through a gateway that has the capability to manipulate the traffic. The honeynet gateway can add or remove a firewall rule to block the attacker from accessing a specific honeypot. Restricting attacker's access to one of the honeypots does not influence his access to other targets.

The testbed provides both detection and response functionality and takes care of monitoring the response actions (Figure 2). The records in the central database of authentication attempts are used for ongoing attack detection. The response selected by the defence strategy is implemented at the gateway, where firewall rules are added to block/allow traffic from the attacker to the target.

One of the main advantages of the testbed is that the

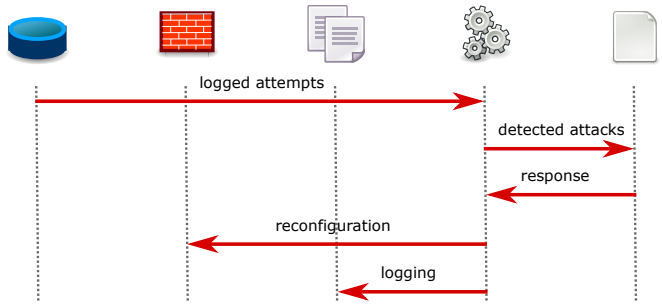

Fig. 2: Strategy Execution. Every minute, new records in the database are retrieved and the tested strategy decision logic is invoked to find an optimal response. Based on the response, the firewall is configured to block/allow the traffic from the attacker to the honeypot. Each reconfiguration is logged along with timestamp, attacker's IP address and targeted honeypot's IP address for later use in evaluation.

parameters needed to run the strategy can be estimated very easily. We had identical honeynet running for several years, and therefore we have a large set of historical data available. Therefore we are able to estimate parameters such as the probability of successful attempt of the value attackers assign each honeypot. The historical data also provide a good baseline, since we can evaluate the efficiency of no defence at all.

We evaluate the efficiency of the tested strategy per each attack. From the logs, we determine the total amount of firewall reconfigurations, the total amount of time, for which the service was unavailable and whether the attack was successful or not. Based on these counts and the cost parameters (damage suffered in case of a successful attack, the cost of reconfiguring a firewall and a penalty for a unit of time of service being unavailable) we compute the total score per attack. Computing the score per each attack enables more detailed analysis of strategy efficiency, such as studying the variance of the results, the correlation between the score and various characteristics of the attack, etc.

The testbed can be further extended by including false positive and false negative detections, detection of the attack intensity, etc.

III. Network Strategy Deployment Demonstration

In the demonstration of the testbed, we are going to present the whole testbed with a sample implementation of a network defence strategy. We are going to describe the setup and requirements of the testbed as well at its advantages and limitations.

The sample defence strategy is based on cost analysis, a commonly used concept in network defence. The cost sensitive strategy considers the positive and negative effects of each response in terms of unnecessary reconfigurations, the risk of successful attack and the unavailability of the service. The strategy was one of the evaluated strategies in our experiment described in the full paper. The strategy was deployed in a testbed for 15 days facing real attackers. During that interval, we observed 437 attacks on the honeypots, which provided us with a statistically sufficient data sample for further analysis.

The testbed and the strategy are implemented in python. The strategy must be implemented as a derived class that overrides a method `defend(attacks)`, which takes as an input the list of attacks observed since the last time the strategy was run. The base class provides methods `block(srcip, dstip)` and `unblock(srcip, dstip)`, which can be called to reconfigure firewall, as well as method `get_firewall_state(srcip)` which returns a description of current firewall configuration for given source IP address and method `get_configuration()` that returns the strategy input parameters. The input parameters must be located in a JSON configuration file. For each honeypot, it must specify the value of damage in case of a successful attack, an availability penalty per unit of time and the cost of reconfiguration. Those same values will also be used during evaluation. All reconfigurations of firewall rules are logged in JSON to a text file. The whole source code and documentation is available in the GitHub repository[1].

IV. Conclusions

We have presented a testbed for evaluating network defence strategies. The testbed relies on a honeynet which is defended by the strategy against simulated attacks or real attackers and is collecting data for evaluation of the appropriateness of the response. The setup provides a tested strategy the ability to detect attacks as well as to execute the optimal response. The simple setup, complete information and control over the testbed, detailed monitoring and abundance of historical data provides an excellent way to estimate input parameters needed by the network defence strategy. These parameters would be very hard to estimate in newly setup or unmonitored test network. The performance of the tested strategy can be evaluated using the logs collected during the deployment and the quantified objectives of the defence strategy. For example of utilization of the proposed testbed, we recommend reading the full paper [2].

Acknowledgement

This research was supported by the Security Research Programme of the Czech Republic 2015 - 2020 (BV III / 1 VS) granted by the Ministry of the Interior of the Czech Republic under No. VI20162019014 Simulation, detection, and mitigation of cyber threats endangering critical infrastructure.

References

[1] D. Kouřil, T. Rebok, T. Jirsík, J. Čegan, M. Drašar, M. Vizváry, and J. Vykopal, "Cloud-based testbed for simulation of cyber attacks," in *Network Operations and Management Symposium (NOMS), 2014 IEEE*, May 2014, pp. 1–6.

[2] J. Medková, M. Husák, and M. Drašar, "Network defence strategy evaluation: Simulation vs. live network," in *Proceedings of the 2017 IFIP/IEEE International Symposium on Integrated Network Management, IM 2017*, 2017.

[1] https://github.com/csirt-mu/DefenceStrategyTestbed