Reducing impacts of incoming attacks using live redirection and cloning of networks

Andreas Straube Elbæk
Aalborg University
Aelbek17@student.aau.dk

Diogo Remião
Aalborg University
Dremia20@student.aau.dk

Marcus Bisgaard Jensen
Aalborg University
Marcje17@student.aau.dk

Robert Nedergaard Nielsen
Aalborg University
Rnni17@student.aau.dk

***Abstract*—**

I. Introduction

In a world of ever growing digitalisation, data is the new gold. If in the past, a bank's worst fear would be to have its vault cracked open, now it would be to have its data servers breached.

As it happens to be the case, today's organisations worst problem is, without doubt, cyber security, with Accenture reporting that 68% of business leaders acknowledge the increasing risks in cybersecurity[1].

It is known how serious a critical breach in a company's servers can be, with Equifax settling for a 575 million dollars fine for losing personal information of almost 150 million people[2]. Even countries' institutions are subjected to attacks, with Herjavec claiming that 93% of healthcare organisations have had a data breach[3]. With hackers constantly searching for possible security faults in systems, all organisations are forced into a continuous investment in cyber security, with Gartner predicting the cybersecurity market to reach 170.4 billion dollars in 2022[4].

Commonly the attacker has the advantage due to cost and information asymmetry between an attacker and a defender with reasons listed here[5].

1) Only a single vulnerability in a system is needed for an attacker to gain access to the system. This could be via an exploit that use as little as 20 lines of code while defenders have to ensure not a single vulnerability is present over millions of lines of code. This creates a cost asymmetry.
2) Defensive measures for exploits and vulnerabilities are based on prior knowledge while attackers have access to 0-day-exploits, here defenders will always be in a position of catching up. This creates an information asymmetry.
3) The static nature of networks employed in work environments gives attackers time to analyse the network. The defenders do not have the opportunity to analyse the attacker before they strike.

In order to even the field defenders can employ strategies like Moving Target Defence (MTD). This can help mitigate the information asymmetry enabled by a static network. This however is still a defensive measure and does not give the defenders an explicit advantage. To achieve this defenders can take an offensive position. This have been achieved by utilising research honeypots or honeynets to lure in attackers and then analyse their tools and methods[6]. When operating a cooperate network the attention of attackers is usually unwanted, so this method is not always desired. This paper propose taking an offensive approach by redirecting inbound malicious traffic to a shadow network when detected as seen in Figure 1. A shadow network in this context is a clone of an original network. Ideally, here no harm can be done. What this achieves is:

- Wasting the attackers time.
- Give the defenders the opportunity to see what methods the attacker employ.
- Put the defenders in a position where they can start profiling the attacker.

This way, defenders will not have to shutdown vital services in order to protect data, minimising negative effects of the attack.

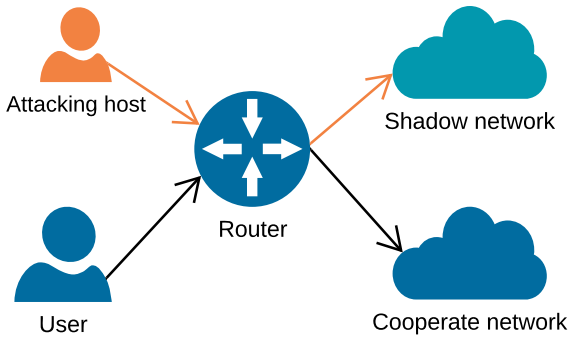

Figure 1. Redirecting traffic from an attacking host to a shadow network

This strategy is based extensively in a humanistic approach to a cyber attack. Until now, little to no attention would be given to whether it was a machine-based attack or a human-based attack. The defence mechanism would be very similar for both situations[7]. By understanding the context of an attack, defenders can adapt their strategy accordingly. For example, defenders need to take into account that human attackers will be monitoring various parameters regarding the status of the attack. By redirecting an attacker, latency will be added to the network traffic. A discrepancy in latency can

potentially alert an attack that their presence is known and thus the defence will be compromised.

To put what is proposed in the context of an incident response NIST 800-61 from National Institute of Standards and Technology (NIST) is considered[8]. NIST 800-61 is an incident response guide with the following steps:

- Organizing a Computer Incident Response Capability
- Handling an Incident
  - Identify
  - Contain
  - Eradicate
  - Recover
- Coordination and Information Sharing

Redirecting traffic to a shadow network can be utilised as part of the containment step. In turn this will also yield more information for *Coordination and Information Sharing* activities, especially regarding what indicators to watch for in the future to detect similar incidents. Considering the steps when handling an incident there is a distinction between identification, containment, eradication and recovery. Using this distinction a method for containment is what is proposed, this does not include the other steps of incident response, and relies on identification of an attacking host to have already taken place.

As for the technical side of it, the defence strategy is based on a number of different technologies to be described in the paper. To bring some context to the matter consider this. A target of a cyber attack is characterised by some unique indicators, i.e., IP address, that identify the target for both the attackers and the defenders. A MTD mechanism normally employed acts as a shuffling algorithm, constantly changing these identifiers and therefore the structure of the network, making it more difficult for the attackers to map the network and employ a successful attack plan [9]. This paper, presents a different approach. Instead of shuffling these identifiers, the idea is to clone the host under attack and hosts in the same network and redirect all traffic to the cloned network. Ideally, the attackers will not know since as far as they are concerned, the identifiers are the same and therefore the target has not changed.

This strategy can see possible applications in enterprise network systems, i.e, corporations and institutions, were important data is stored and needs to be safeguarded.

The paper will start by examining related work in Section II looking more into the MTD and shadow network concepts. After that in Section III the concept seen on Figure 1 will be elaborated and the needed components described, how these will be implemented in the system is then described in Section IV and in Section V the system performance is tested and evaluated.

II. Related work

As mentioned in the introduction the goal is to implement an active approach to incident response. That is to create a system that can clone a whole network and redirect malicious traffic to the cloned network. This section will examine related work, starting with examining different versions of MTD and then moving on to shadow and honey networks.

MTD a proactive defence mechanism based on changing aspects of the network, such as the IP address of hosts. Static defensive strategies are subjected to more pressure, as the attackers can usually stick to an attack plan from beginning, because the network they are attacking always has the same characteristics. Identifying an active IP address of a target host is the first step towards an attack, exposing the hosts to possible breaches. The purpose of MTD is to create an ever-changing attack surface. By changing different aspects of the network, for example, shuffling the virtual IP addresses of the hosts, the attackers are forced to continuously re-evaluate their attack strategy. And hopefully, by the time the attackers notices, the network will have changed enough, preventing the exploit from working in the new configuration[10].

One way of implementing MTD is OpenFlow Random Host Mutation (OF-RHM) which frequently mutates end-hosts IP addresses in a randomised way. This of course causes disruption in the attackers plan, as all assumptions related to a static network become false. By continuously re-assigning virtual IP addresses, attackers have much more difficulty mapping the network. This is implemented as a Sofware Defined Network (SDN), which facilitates the deployment of the algorithm and provides an efficient IP mutation scheme[9]. However, this method differs from what is proposed in this article in the sense that it relies on randomisation to confuse the attacker, while this proposal does not rely on randomisation, as the main objective is to trick the attacker into thinking that nothing changed in the network.

Another way is Moving Target IPv6 Defence (MT6D) this consists of hiding and rotating IPv6 addresses of communicating hosts, without disrupting ongoing traffic. For this, MT6D makes use of tunnelled packets, although it requires significant overhead to make this tunnelling possible. This method is much harder to implement as it does not rely on SDN to be deployed, a concept crucial in this proposal, to shorten deployment time[11].

Shadow networks or decoy networks are essentially a collection of devices on a network, intended to confuse, distract or isolate a possible attacker. Shadow and decoy is sometimes used synonymously but invites two different utilisations. A shadow network is often used in combination with MTD for example if the controller in the MTD system detects that the network is being scanned, it can redirect the scan to the shadow network. This shadow network can ideally be a simulation of all the devices within the normal network, leading the attacker to believe that they arrived in a the target network[10].

A decoy network is slightly different, consisting of a collection of fake devices meant for distracting the attacker. These devises are not necessarily a copy of any existing devices, but only serves the purpose of confusing the attackers. The devices can also be used for identifying attacks as these decoys should not receive any traffic from legitimate hosts. If they start receiving traffic it is an indicator that something is wrong

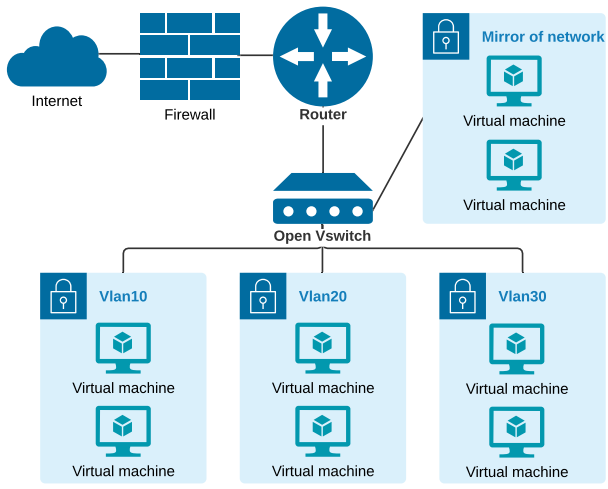

Figure 2. Illustration of the concept.

within the network.

The way that a honeynet differs from both a shadow network and decoy network is its purpose. With shadow and decoy network the overall goal is to identify or contain an attack. Where as the purpose of a honeynet is to lure an attacker in and then record their every move[12]. This data is used to gain insights that can help protect against similar attacks.

Redirecting traffic to a shadow network can waste an attackers time much like MTD by making network probing less rewarding. And much like honeynets this allows a good actor to analyse an attacker. This concept and design will be described more in the following section.

III. Design

In this section, the design of the system will be described . First, a scenario need to be set, from which a concept will be created. This concept will serve as the foundation for the actual system design and implementation.

Lets consider a scenario were a server is under attack. The attackers have gained access to the server, and are now searching and mapping vulnerabilities in the underlying network. This causes unusual activity spikes in the network, signalling the defenders that an attack is happening. As previously mentioned, detection is not the focus of this paper, therefore from now on we assume that the attack was detected. Nonetheless, a few approaches could be:

- Intrusion detection system (IDS).
- Decoys in the network receiving unexpected traffic.
- Manual monitoring of the network.

The defenders then have to make a decision on how to handle the attack. A straightforward approach is to block all traffic to and from the infected server, this causes minimal data loss at the cost of severe service disruption. However, in this scenario the defenders will employ an active approach to the defence, choosing to redirect all incoming malicious traffic to a decoy network where the behaviour of the attacker can be monitored. This concept is described more in Section III-A

A. Concept

Figure 2 shows the setup of the system, which consists of a router connected to the internet through a firewall and/or an Intrusion detection system (IDS). The idea is then to mirror the network and redirect incoming traffic to the mirrored network if an attack is detected in the IDS.

The main advantage of mirroring the network and the hosts is that it provides a controlled environment to contain the attack. This of course requires sensitive information to be deleted from the cloned hosts as the purpose is to avoid any loss of important data. Another advantage of cloning the network is that the hosts can behave as a honeynet, monitoring the behaviour of the intruder in order to identify possible comparable attacks. Despite this, note that it differs from a ordinary honeynet, as it will not be running all the time, but only deployed on demand.

The following sections will examine how the router, switch and hosts seen on Figure 2 will be implemented. For practicality reasons, the before mentioned elements are going to be virtualised, this will also provide the flexibility to have network configurations with different hosts like servers, workstations and printers. The first section will start by examining how the hosts are virtualised, then the router and ending with the switch.

B. Virtual hosts

As mentioned before the hosts in the network will be virtualised, this is done using VirtualBox (VBox). The reason for choosing VBox is its ability to control it programmatically, this is done using our own library[13]. This library is developed using VBox's command-line interface VboxManage.

Another benefit of having this programmatic control is the ability to automate the process of setting up and cloning the hosts in the system. This setup process is done based on a configuration file, the format of the file is chosen to be YAML, the reason for this choice is the readability of YAML. This way of configuring the networks gives the flexibility to easily change the network, by simply changing this file.

C. Routing

For the setup to be able to emulate different network topologies some kind of routing is needed. An example of a network that could be created is a corporate network. A corporate network is often divided into multiple subnets, which is called network separation. In this case the router should be able to forward traffic both between the local networks, but also from the local networks to the internet[14].

This system supports multiple ways of doing this, one way is by spinning up a virtual version of PfSense. However, in this setup a simple Ubuntu VM is chosen. The routing is then done using iptables, by manipulating the Prerouting, Postrouting and Forwarding tables.

- **Prerouting:** Is defined as the alternation of incoming packets when they are received. It is used to route the packets to the correct destination within the network.

- **Forwarding:** Is the routing of internal packets from a source to an internal destination.
- **Postrouting:** Is defined as the alternation of outgoing packets just before they are sent.

D. Switch

In order to emulate different network topologies, it is also necessary to be able to have one or more switches in the network. This multiplicity is made easier by using virtual switches instead of physical switches, as no considerations concerning varying hardware or hardware availability are necessary. Another issue is the possibility of using physical switches with virtual machines, it would demand multiple interfaces on the host, multiple hosts or make the VMs on the host share the same interface divided into Virtual Local Area Network (VLAN). For this reason it is chosen to utilise a virtual switch for the system.

This virtual switch will be deployed using Open Virtual Switch (OVS). This is chosen as it can be working at both layer 2 and layer 3 and allow for easy programmatic control, which will make it easier to manipulate the network topology on the go. With all elements of the system described the next section will examine how they will be combined into a system that can emulate a real network and change it's topology when an attack is detected.

IV. Implementation

This section will introduce the methods used to implement the different parts of the program.

A. Config file

As mentioned in design, the library created is able to read from a YAML file and deploy the Virtual Machines accordingly to what is specified in the file. This implementation is very useful, as it allows for easy configuration by the user. This is due to YAML easy-to-read syntax, which was one of the main reasons why it was chosen for the config file. The structure of the file is as it follow:

```
name: NAP
dns: 1.1.1.1
nets:
  - subnet: 172.0.1.0
    min: 172.0.1.6
    max: 172.0.1.254
    vlan: 10
    path: ./vms/net10
    hosts:
      - id: Host1
        ram: 256
        cpus: 1
  - subnet: 172.0.2.0
    min: 172.0.2.6
    max: 172.0.2.254
    vlan: 20
    path: ./vms/net20
    hosts:
      - id: Host1
        ram: 256
        cpus: 1
```

B. Network

The initial setup of the network is as described in Figure 2. There will be three VLANs, each of them with two VMs connected to them. The reason for having multiple VLANs is to simulate an enterprise network, where multiple hosts are present. If one of them is under attack, the system needs to be able to clone the topology of the correct VLAN and redirect its traffic. More over, the presence of multiple VMs in the VLAN demonstrates the capability of the cloning method to clone multiple devices.

C. Automation

In order to implement a working prototype of the product it is necessary to automate some of the processes. There are several different aspects that needs automatisation.

1) Routing:

2) Manipulation of hosts:

V. Testing

In order to test this implementation, there is the need to first set which are the parameters that are considered critical. The success of this design will depend of course on getting good results regarding these.

A. Deployment time

Given that we are working with a virtualised network, deployment time should be much faster than with physical hardware from the start. However, since a library was developed to give programmatic control over the setup, initial deployment time should be even faster, given that it is reading from a configuration file and therefore, should not require human intervention. While this is not necessarily a critical aspect to the project, it is a always good to have a fast deployment time, if not only for testing purposes.

B. Cloning time

The initial premise when considering cloning time is that the attack was detected and the compromised hosts are know. From that point on, the cloning process begins. This the first part of the defence mechanism. The system needs to be able to clone the attack hosts as fast as possible, effectively mirroring the hosts in a shadow network to were traffic will be redirected. Therefore, cloning time is critical and testing should be done taking into consideration scalability and how is time affected by the size of the network.

C. Traffic redirection

After the shadow network is up and running, the routing mechanism is deployed. The job of the re-routing mechanism

is to redirect all traffic related to the attacked host to the new shadow network. An important aspect needs to be taken into consideration here. The attackers are expected to be monitoring the state of the attack, and there is a lot of information that they have access to. Two parameters are critical however.

The first is packet loss. Given that there are no perfect transmission mediums, it is normal to have some small percentage of packet loss. However, if this percentage increases dramatically is a small amount of space, attackers will know that something has happened. Therefore, it is crucial that, when making the initial transition redirecting the packets, this is done as fast as possible and with the least amount of packets lost in the process.

THe second parameters is latency. If there are no substantial changes to the connection medium or to the network, successive pings to a host should have approximately the same latency. Then again, spikes in latency show that there has been a change to the network and attackers will seek to updated their attack plan and could find out about the shadow network, countering the defence strategy. It is therefore crucial that the routing done at the switch level is as fast as possible.

VI. Conclusion

This paper presented a detailed description on how automated traffic redirection can be used to isolate malicious traffic in a network replica. The results prove that the redirection of specific network packets can be handled with little latency, to preserve the attackers nescience. It was showed that it is possible to live clone virtual machines and redirect malicious traffic to the clone where the attack can be observed and analysed. This live redirection ensures that a company will be able to keep operating and accessing their network devices. The paper has shown how to design and implement a virtual solution for the aforementioned traffic redirection.

Acknowledgments

References

[1] AccentureSecurity, *The cost of cybercrime*, https://www.accenture.com/_acnmedia/PDF-96/Accenture-2019-Cost-of-Cybercrime-Study-Final.pdf, [Visited 30-09-2020], 2019.

[2] Equifax, *Equifax announces cybersecurity incident involving consumer information*, https://investor.equifax.com/news-and-events/press-releases/2017/09-07-2017-213000628, [Visited 30-09-2020], 2017.

[3] H. group, *The 2020 healthcare cybersecurity report*, https://www.herjavecgroup.com/wp-content/uploads/2019/12/Healthcare-Cybersecurity-Report-2020.pdf, [Visited 30-09-2020], 2020.

[4] Gartner, *Forecast analysis: Information security, worldwide, 2q18 update*, https://www.gartner.com/en/documents/3889055, [Visited 30-09-2020], 2018.

[5] S. Jajodia, A. K. Ghosh, V. Swarup, and C. Wang, *Moving target defense creating asymmetric uncertainty for cyber threats*, https://www.semanticscholar.org/paper/Moving-Target-Defense-Creating-Asymmetric-for-Cyber-Jajodia-Ghosh/a970352e2a6c3c998c4e483e2d78c4b3643c7809, [Visited 19-11-2020], 2020.

[6] Imperva, *Honeypot*, https://www.imperva.com/learn/application-security/honeypot-honeynet/, [Visited 23-11-2020], 2020.

[7] V. F. Mancuso, A. J. Strang, G. J. Funke, and V. S. Finomore, "Human factors of cyber attacks: A framework for human-centered research," *Sage journals*, September 2014.

[8] E. C. Thompson, *Cybersecurity incident response how to contain, eradicate, and recover from incidents*, https://link.springer.com/chapter/10.1007/978-1-4842-3870-7_3, [Visited 19-11-2020], 2020.

[9] J. H. Jafarian, E. Al-Shaer, and Q. Duan, "Openflow random host mutation: Transparent moving target defense using software defined networking," 2012.

[10] D. W. Li Wang, *Moving target defense against network reconnaissance with software defined networking*, https://link.springer.com/chapter/10.1007/978-3-319-45871-7_13, [Visited 5-10-2020], 2020.

[11] M. Dunlop, S. Groat, R. Marchany, and J. Tront, "Implementing an ipv6 moving target defense on a live network," June 2012.

[12] Kaspersky, *What is a honeypot?* https://www.kaspersky.com/resource-center/threats/what-is-a-honeypot, [Visited 17-11-2020], 2020.

[13] G. 720, *P7-nap/vbox*, https://github.com/p7-nap/vbox, [Visited 18-11-2020], 2020.

[14] P. Networks, *What is network segmentation?* https://www.paloaltonetworks.com/cyberpedia/what-is-network-segmentation, [Visited 01-10-2020], 2020.