An Adaptive Moving Target Defense Approach for Software-Defined Networking Protection

Emídio Neto (UFRN), Rodrigo S. S. Nunes (LaTARC/IFRN)
Cristian H. M. Souza (LaTARC/IFRN), Felipe S. Dantas Silva (LaTARC/IFRN),
Túlio Pascoal (University of Luxembourg), Augusto Neto (UFRN)

IEEE/IFIP Network Operations and Management Symposium (NOMS/2023)

Agenda

- Motivation
- MADS
- Experimental Evaluation
- Conclusions and future work

Motivation

Motivation

- *Softwarization:*
  - Enabler for the deployment of defense mechanisms for next generation SDN
  - Several previous works demonstrate ways to compromise SDN through targeted attacks
    - DoS/DDoS
    - Slow TCAM
    - **Scanning (e.g., Inference of network rules timeouts, services)**

Motivation

- *Moving Target Defense (MTD)*:
  - Aims to dynamically change parameters and/or characteristics of systems;
  - Reduce windows of opportunity that attackers can detect to attack computer systems and/or networks;
  - Currently it is widely used to prevent scanning attacks.;
- MTD + *Softwarization*:
  - MTD requires high programmability capabilities;
  - Dynamic changes of network characteristics, options, setup;
  - Currently, this can be implemented both in the data plane (e.g., through P4 capabilities) and in the **control plane**.

Motivation

- The literature shows a few related works that focused on protecting software networks against scanning attacks with MTD:
    - The solutions are mainly based on adding latency to malicious packets to make it difficult to identify and deduce network information;
- Despite being effective, these solutions directly impact the network performance (e.g., QoS)

Motivation

- *MTD Adaptive Delay System* (MADS):
  - An adaptive solution for software-defined network protection based on the MTD approach
- Unlike related solutions, MADS triggers the MTD mechanism adaptively:
  - MADS applies latency based on target network behavior
    - Only in situations where the network is actually impacted by a scanning attack
  - Maintains the **same level of network protection** with l**ess degradation of the network.**

Motivation

- In this way, it prevents the network and legitimate packets from being continuously impacted by the MTD.
- MADS relies on scanning attack modeling to determine thresholds:
- Thresholds are used to identify the presence and impact of an attack on the network at a given time. Based on this, MADS performs the activation of its MTD actions.

MADS

MADS - Overview

- MADS applies latency settings to the network adaptively, rather than continuously (as exhaustively performed by related works)
- Minimizes the negative impact that a MTD technique based on adding latency can impose in terms of network QoS
- Supports adaptive latency configuration as a functional block embedded in the network control plane
- Adaptive capabilities are supported by network state monitoring

MADS - Activation

- MADS monitors the status of flows every **Tmr** (seconds). For example, **Tmr = 10s** [Zarek et al.]
- Defines thresholds of bytes transferred to the switch interfaces through a previous observability period defined by **SR**;
- **Trd** is **SR** multiplied by the value of **Tmr**:
  - For example, **Trd** = **SR** * **Tmr** -> 12 * 10 = 120s
  - **Trd** is used as a trigger to deactivate the mechanism (acts as a hard timeout for MADS)
- If the current throughput rate of suspect packets exceeds the defined values (**SR**) between each **Tmr** monitoring interval, MADS enables the MTD

MADS - Latency definition

- Methodology based on the work of [Ma et. al 2014]:
  - Uses ICMP packets to define values for FirstPacket (t1) and LastPacket (t2);
  - Builds a list Dt from subtracting the values of t2 by t1;

$$Dt = \{dt(i) | dt(i) = t_1(i) - t_2(i), i > 0\} \quad (1)$$

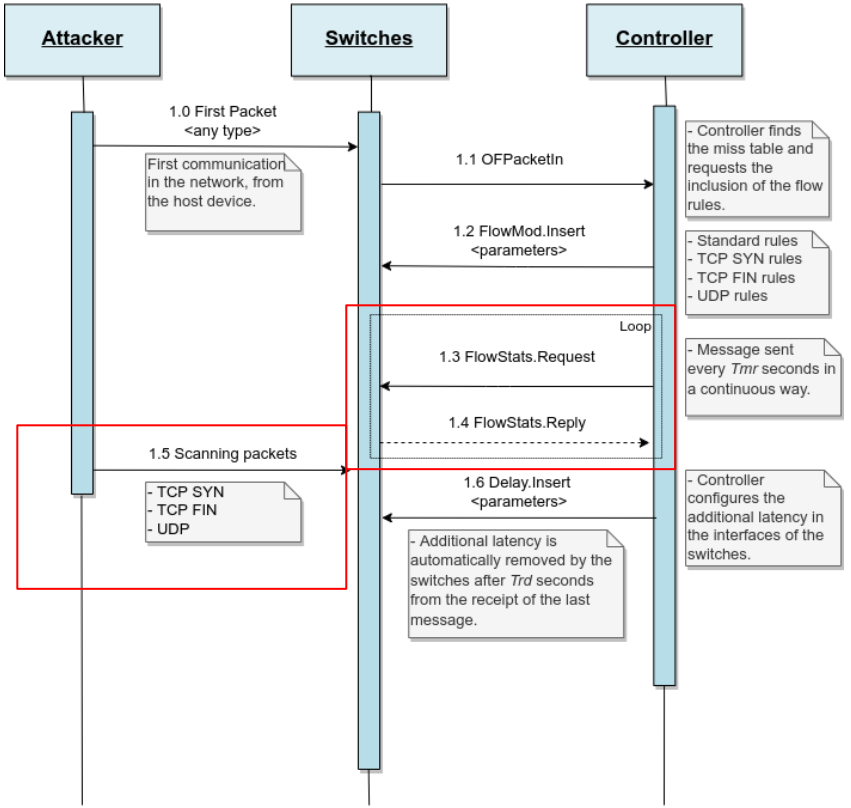
MADS - Latency definition

- Methodology based on the work of [Ma et al. 2014:
  - Randomly selects a value, which must be between the minimum and maximum limits present in the Dt list;
  - Random value is added to t2 to obtain the delay value to be added on the switch interface where the scanning originated

$$T2 = \{t_2'(i) | t_2'(i) = t_2(i) + Random[Min(Dt), Max(Dt)], i > 0\} \quad (2)$$

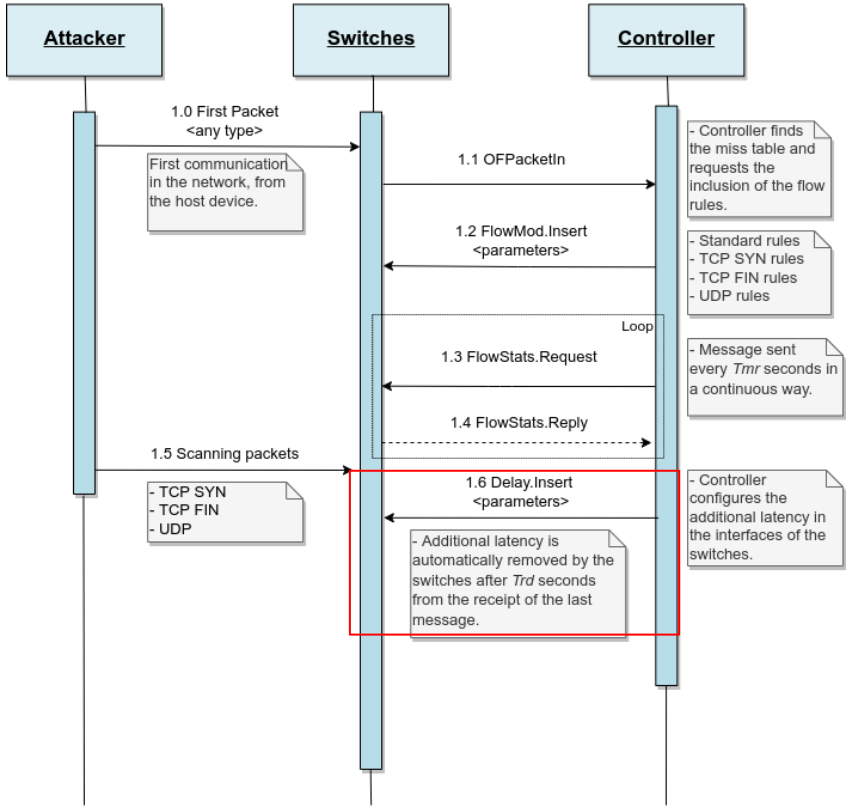
MADS - Workflow

- MADS performs state monitoring of flow rules at intervals defined by **Tmr** (Step 1.3 and 1.4)
- Scanning (Step 1.5), causes the byte count of some rules to increase in a way that does not match the value defined for **SR**.



MADS - Workflow

- MADS triggers the MTD actions by sending a Delay.Insert message to the device (Step 1.6)
- The latency is automatically removed by the data plane after they no longer receive Delay.Insert messages from MADS for **Trd** seconds



Experimental Evaluation

Experimental Evaluation

- Evaluate the impact generated by MTD defenses on network performance;
- We compared [Ma et al. 2014], [Hou et al. 2020] and MADS
- We collected metrics like:
  - RTT
  - Throughput
  - Bad TCP

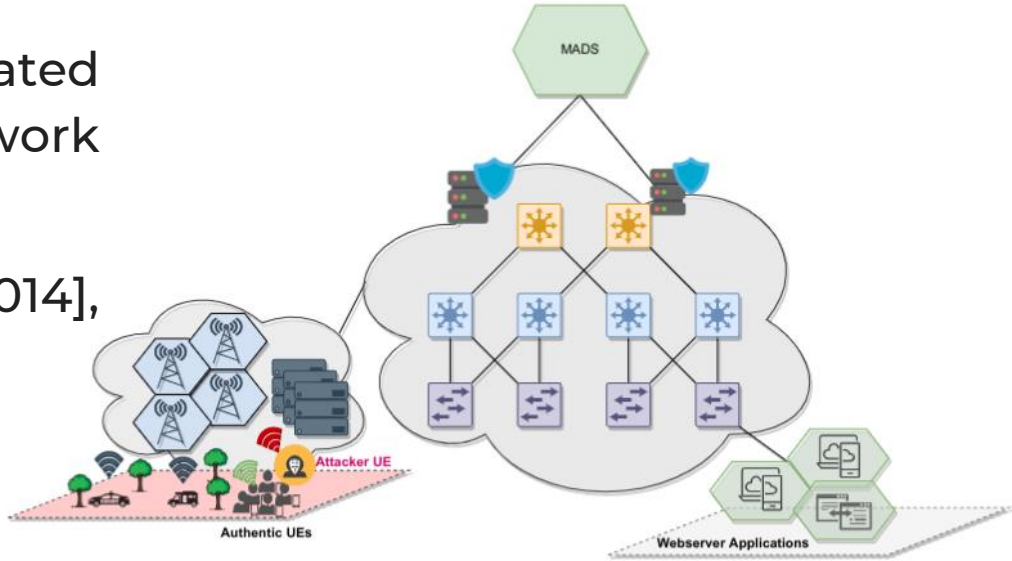


Fig. 3. Topology set-up.

Experimental Evaluation

- Each experiment lasted 78 minutes with the following approach:
  - *Clients generates HTTP traffic to web applications at 1 second intervals*;
  - *Attacker* performs a scanning attack, sending requests to the network in search of open TCP ports with a duration of 13 minutes
- We consider **Tmr** = 10s and **Trd** = 2minutes (120s);

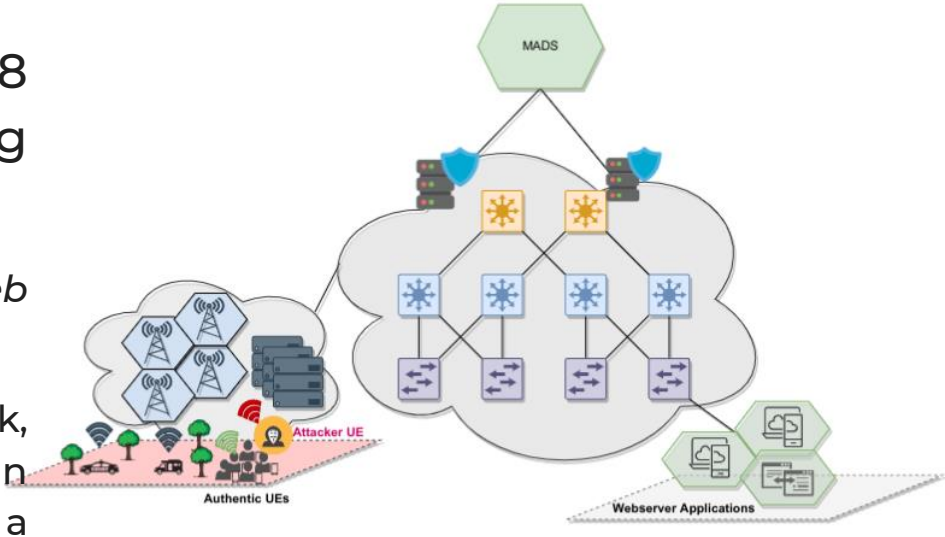


Fig. 3. Topology set-up.

Experimental Evaluation

TABLE I
OVERALL COMPARISON OF THE RTT, THROUGHPUT AND BAD TCP COLLECTED DURING THE EXPERIMENTS.

	Ma2014		Hou2020		MADS	
Metric	**Avg. (ms)**	**Std. (KB/s)**	**Avg.**	**Std.**	**Avg.**	**Std.**
RTT	20.72	0.61	0.12	0.06	0.11	0.08
Throughput	47.05	7.64	1215.19	266.48	1274.44	306.67
Bad TCP	17.15	4.14	–	–	–	–

Experimental Evaluation

- MADS and [Hou2020] presented a similar RTT, remaining below 0.2ms most of the time and reaching a latency 99.4% lower than the proposal of [Ma2014];

- MADS reached a Throughput 4.87% higher when compared to [Hou2020]

Resultados

- For all parameters, the proposals [Hou2020] and MADS had very similar results (4.87%);
- The additional latency is only introduced for traffic generated by the attacker;
- [Hou2020] can represent an issue for the network due to the packet processing overhead in the controller.

Conclusions and future work

Conclusions and future work

- It is evident that MTD techniques are being used more and more, mainly to combat DoS attacks, scanning;
- We consider MTD for protection against scanning attacks in software-defined networks;
- MADS is able to maintain the efficiency of the MTD strategy to mitigate scanning attacks;
- In addition, the effects of QoS degradation observed in the MADS operation are very soft when we compare to the state of the art;

Conclusions and future work

- Adoption of new parameters (number of hops and topology size) for the MTD;
- Real-time AI to support the decision-making process (value of delays to be configured) according to the behavior and security level of the network

An Adaptive Moving Target Defense Approach for Software-Defined Networking Protection

Emídio Neto (UFRN) - zshemidio@gmail.com

Rodrigo S. S. Nunes (LaTARC/IFRN), Cristian H. M. Souza (LaTARC/IFRN),

Felipe S. Dantas Silva (LaTARC/IFRN), Túlio Pascoal (University of Luxembourg), Augusto Neto (UFRN)

IEEE/IFIP Network Operations and Management Symposium (NOMS/2023)