# Implementation of Content Poisoning Attack Detection and Reaction in Virtualized NDN Networks

Hoang Long Mai*, Messaoud Aouadj†, Guillaume Doyen†, Daishi Kondo‡,
Xavier Marchal‡, Thibault Cholez‡, Edgardo Montes de Oca*, Wissam Mallouli*
*Montimage, 39 rue Bobillot, 75013 Paris - France
†ICD - UMR CNRS 6281, Troyes University of Technology, 10004 Troyes Cedex - France
‡LORIA CNRS UMR 7503, TELECOM Nancy - University of Lorraine, 54506 Vandoeuvre-les-Nancy, France

***Abstract*—The orchestration of counter-measures in the context of security incidents remains a challenging task for network operators. The main objective of this demonstration is to present how this orchestration is possible in the context of a virtualized NDN network. Based on an adaptation of the TOSCA topology and orchestration model, it is possible to trigger these counter-measures after the detection of NDN specific attacks. We show how the Montimage Monitoring Tool (MMT) has been adapted to detect typical Content Poisoning Attack (CPA), and how the orchestrator can trigger reactions to mitigate their impact on the network.**

## I. Introduction

Named Data Networking (NDN) [1] is a networking concept, moving away from a host-based communication network toward a content-based one that is better suited for the massive diffusion of content in todays major Internet use-cases. As a new approach, NDN introduced router components expose the network to new attack types [2]. Among them, Content Poisoning Attack (CPA)[3], [4] is identified by the NDN community as one of the most significant attacks. In a CPA, with the cooperation of a malicious client, a poisonous provider can insert Bad Data into the cache of intermediate routers so that a request from legitimate users receives this altered content.

For a secure deployment of NDN in an ISP's infrastructure, it is essential to design a security monitoring plane that can detect security incidents and automatically trigger a reaction strategy to ensure the network's robustness and continuity.

The different paper sections determine the different steps of the demonstration. In Section II, we present our testbed technical environment. In Section III, we describe in detail the demonstration scenario. Finally, in Section IV we conclude this article.

## II. Testbed technical environment

The demonstration is performed by connecting using ssh to a remote testbed deployed in the premises of University of Troyes (UTT). The UTT testbed is composed of several servers that allow, thanks to the use of virtualization technologies, to easily reproduce realistic networks. This facilitates the development, test and validation of innovative algorithms and network paradigms.

The deployed nodes (NDN routers, NDN firewalls and NDN signature verification modules) are hosted in virtual machines (VMs) that are created and managed using Docker containers in OpenStack platform.

### *A. Virtualized Data plane*

The Virtualized Data plane includes different VNFs (Virtualised Network Functions) deployed over a NFV infrastructure. More specifically, Doctor's VNFs are containerized NDN applications deployed over a cluster of Docker nodes that are managed as a single virtual infrastructure thanks to Docker Swarm.

The service function chaining (SFC) between VNFs is ensured through the configuration of NFD[1] modules (NDN Forwarding Daemon), a network forwarder that implements and evolves together with the NDN protocol.

### *B. Control plane*

*1) Firewall:* Currently, NFD does not support the appending of filtering rules to perform Interest packet filtering (e.g., spam filter). Indeed, by modifying the NFD, it can implement the filtering function, but this may cause performance degradation of the NFD and the function is not mandatory for all of routers. Thus, an NDN firewall, written in C++, has been implemented in the DOCTOR project[2] that can be easily managed by an orchestrator to add or remove filtering rules.

*2) MMT:* MMT, written in C, is a network monitoring solution that relies on Deep Packet Inspection (DPI) technology. It has been adapted to NDN technology to gather 18 different metrics useful for detecting potential attacks [5]. It uses Bayesian Networks (BNs) to correlate these metrics and identify anomalies. The changes in the values of these 18 parameters are learned from different legitimate and attack scenarios allowing the detection of CPA attacks with a 95% true positive rate.

[1] https://redmine.named-data.net/projects/nfd/wiki/Management

[2] http://www.doctor-project.org/

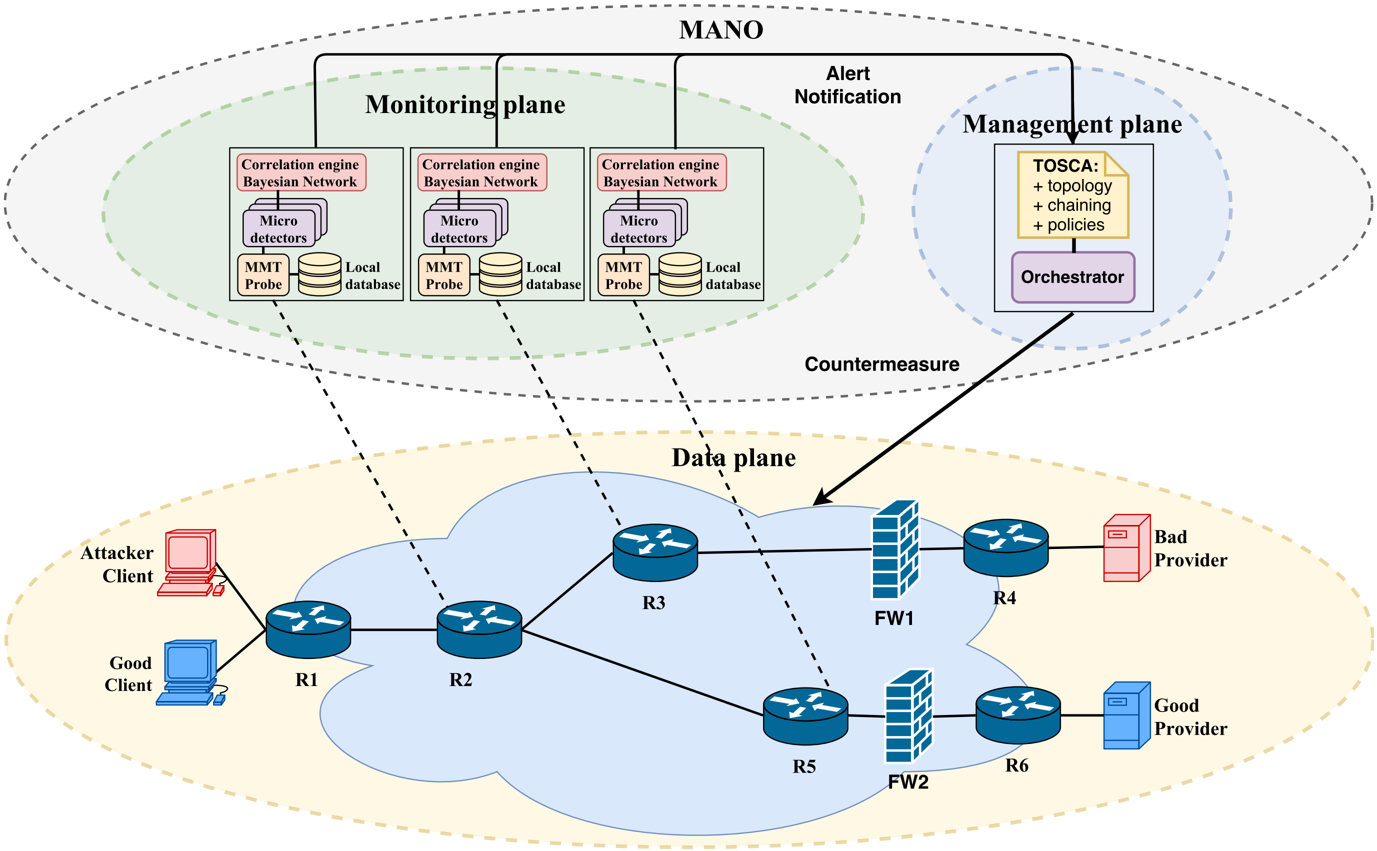

Fig. 1. NDN Network Architecture.

*3) Verification signatures module:* Ideally, to ensure security in an NDN stack, Data packets are signed once when published. In contrast, a Data packet may be fetched and verified many times. One can use verification-efficient signature algorithms, such as RSA, to reduce the overhead of verification. If one also wants to reduce the overhead of signature generation, symmetric key authentication can be used. Signature verification may involve multiple rounds of certificate fetching and verification. Caching validated certificates can help improve the performance. These validated certificates can be reused before expiration or revocation.

### *C. Management & orchestration plane*

The Doctor NDN orchestrator relies on a content-oriented TOSCA[3] profile to capture deployment and operational behavior requirements of each NDN network service. TOSCA is an OASIS standard language originally defined to describe cloud workloads as a topology template and Doctor's TOSCA profile extends OASIS standard simple Profile for NFV so as to take into account NDN specificities.

The orchestrator has been implemented from scratch to be able to interpret and instantiate the different content-oriented TOSCA templates allowing: (1) the deployment of NDN VNFs; (2) content classification and service chaining; and, (3) the definition of performance and security policies.

## III. Demonstration scenario

### *A. Initial deployment of the network topology*

In this demo we demonstrate the usage of TOSCA to specify the network topology described in Fig.1 and define the chaining between NDN nodes and firewalls. The deployment of the architecture is done in a matter of seconds (around 30 seconds) including the spawning of different nodes and the configuring of the NFD Daemon.

### *B. Running CPA attacks*

We emulate CPAs as illustrated in Figure 1, where routers use the default forwarding strategy in NFD: best route. A Good Client is connecting to R1 and requests Data from a Good Provider, its Interest is forwarded to the Good Provider. At the same time, an Attacker Client also connects to R1 and frequently sends Interests to force router R2 to cache the malicious content from the Bad Provider. Therefore, while waiting for Data, the R2 receives the second Interest request the same Data from the Attacker Client, and the Interest will be sent to the Bad Provider as the second choice in the forwarding table. Finally, thanks to the shorter delay, the Bad Provider will succeed in caching the bad Data packet in R2 before the Good Provider.

[3] https://www.oasis-open.org/committees/tosca

### C. Detection of CPA attack using MMT

*1) Metrics computation:* NDN introduces the NFD Management Protocol[4], which allows retrieving the status of an NDN node, e.g., In Interest and PIT Number. However, other metrics, such as CS Hit, CS Miss, PIT Exist Time, etc., cannot be retrieved from the NFD Management Protocol. For this purpose, the MMT Probe[5], an industrial network monitoring probe written in C, has been extended and installed in the NDN routers containers to analyze traffic data flows and NFD logs for extracting 18 different metrics needed.

*2) Micro-detector:* The different metrics will continuously change and correspond to various ranges of values. As a consequence, directly combining them into a detector would increase the complexity and computation cost. Therefore, the raw metrics collected by the MMT Probe will be communicated to the micro detectors, which is written also in C to analyze the behavior of each metric, and raise an alarm whenever it deviates notably from its normal behavior. The general idea is to statistically and dynamically model each metric in the case of no-attack traffic. Based on such reference models, it is possible to detect, with a prescribed significance level, when a metric does not follow normal behavior. To that aim, we collect the trace of a no-attack traffic in one week to get the distribution and the threshold of each metric.

*3) Correlation engine of Security Events:* It is, however, important to note that the individual detection of each of micro-detectors cannot reliably identify an CPA, but the correlation of these events provides a strong indication that it is an attack. To that aim, a Bayesian Network (BN) structure, a well-known statistical model to infer the occurrence of events from correlated metrics is proposed to demonstrate the causal relationships between micro detectors. The structure and relations in our BN [5] are outlined based on the NFD Developer Guide [6]. While the parameters defining the relation between the nodes in the BN are learned from different attacks and no-attack scenarios, where we emulated the no-attack traffic and the traffic containing CPA & IFA attacks. We implement the BN structure using pgmpy[6] a python library for Bayesian Network Models, the input for this BN is the results from 18 micro-detectors and the output is an alert sent to the network orchestrator if the router is under attack.

### D. Orchestration of countermeasure

We also rely on the TOSCA modeling language to specify different reaction policies to be triggered when a security incident is detected (event CPA alert). The action to be performed is to be defined and applied by the orchestrator (e.g., update router mode to verify signature of transmitted data through router R3 in figure 1).

```
policies:
 - CPA_countermeasure:
     type: tosca.policies.nfv.doctor.security.signature
    _verification
     targets: [router_3, router_5]
     triggers:
      peeringPoint1_verification:
       event_type: tosca.nfv.doctor.security.alert.cpa
       condition:
        constraint: triggred_by router_3
       action:
        action_type: update_router_mode
        mode: signing
        target_router: router_3
```

The demonstrated scenario in this paper is related to the CPA attack. In this case, we activate the verification of signatures of the NDN data that will detect the list of poisoned content (having bad signatures). As a second step, the orchestrator will update the NDN firewall to block this malicious content.

## IV. Conclusion

In this demonstration, the elements needed for the design and implementation of a security-based orchestration for virtualized NDN networks have been proposed. By extending the TOSCA model, we show how the orchestrator can not only deploy a NDN network but also mitigate typical NDN attacks thanks to the MMT. To validate our proposal, one of the most critical attacks in NDN: CPA has been considered in a real tested. The results obtained demonstrate the capability of the proposed monitoring plane which is able to accurately detect these attacks and the effectiveness of the orchestrator to mitigate them.

## Acknowledgment

This work is partially co-funded by the French National Research Agency (ANR), DOCTOR project (<ANR-14- CE28-0001>), the French Systematic cluster, the celticPlus project SENDATE (<C2015/3-1>), and the CRCA and FEDER CyberSec Platform (<201304601>).

## References

[1] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang *et al.*, "Named data networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.

[2] M. Amadeo, C. Campolo, and A. Molinaro, "Information-centric networking for connected vehicles: a survey and future perspectives," *IEEE Communications Magazine*, vol. 54, no. 2, pp. 98–104, 2016.

[3] T. Nguyen, X. Marchal, G. Doyen, T. Cholez, and R. Cogranne, "Content poisoning in named data networking: Comprehensive characterization of real deployment," in *Integrated Network and Service Management (IM), 2017 IFIP/IEEE Symposium on.* IEEE, 2017, pp. 72–80.

[4] Z. Xu, B. Chen, N. Wang, Y. Zhang, and Z. Li, "Elda: Towards efficient and lightweight detection of cache pollution attacks in ndn," in *Local Computer Networks (LCN), 2015 IEEE 40th Conference on.* IEEE, 2015, pp. 82–90.

[5] H. L. Mai, N. T. Nguyen, G. Doyen, R. Cogranne, M. Wissam, E. Montes de Oca, and O. Festor, "Towards a security monitoring plane for named data networking: Application to content poisoning attack," in *Network Operations and Management Symposium (NOMS), 2018 IEEE.* IEEE, 2018.

[6] A. Afanasyev, J. Shi, B. Zhang, L. Zhang, I. Moiseenko, Y. Yu, W. Shang, Y. Huang, J. P. Abraham, S. DiBenedetto, and others, "NFD developers guide," Technical Report NDN-0021, Oct. 2016. [Online]. Available: https://named-data.net/wp-content/uploads/2016/10/ndn-0021-7-nfd-developer-guide.pdf

[4] https://redmine.named-data.net/projects/nfd/wiki/Management

[5] http://www.montimage.com/products.html

[6] https://github.com/pgmpy/pgmpy