Towards a Security Monitoring Plane for Named Data Networking: Application to Content Poisoning Attack

Hoang Long Mai†, Tan Nguyen*, Guillaume Doyen*, Rémi Cogranne*, Wissam Mallouli†, Edgardo Montes de Oca†, Olivier Festor‡

*ICD - UMR CNRS 6281, Troyes University of Technology, 10004 Troyes Cedex - France
†Montimage, 39 rue Bobillot, 75013 Paris - France
‡LORIA CNRS UMR 7503, TELECOM Nancy - University of Lorraine, 54506 Vandoeuvre-les-Nancy, France

Abstract—Named Data Networking (NDN) is the most mature proposal of the Information Centric Networking paradigm, a clean-slate approach for the Future Internet. Although NDN was designed to tackle security issues inherent to IP networks natively, newly introduced security attacks in its transitional phase threaten NDN’s practical deployment. Therefore, a security monitoring plane for NDN is indispensable before any potential deployment of this novel architecture in an operating context by any provider. We propose an approach for the monitoring and anomaly detection in NDN nodes leveraging Bayesian Network techniques. A list of monitored metrics is introduced as a quantitative measure to feature the behavior of an NDN node. By leveraging the hypothesis testing theory, a micro detector is developed to detect whenever the metric significantly changes from its normal behavior. A Bayesian network structure that correlates alarms from micro detectors is designed based on the expert knowledge of the NDN specification and the NFD implementation. The relevance and performance of our security monitoring approach are demonstrated by considering the Content Poisoning Attack (CPA), one of the most critical attacks in NDN, through numerous experiment data collected from a real NDN deployment.

Index Terms—Named Data Networking, Bayesian Network, security, anomaly detection, hypothesis testing

I. INTRODUCTION

Nowadays, Internet users are more interested in receiving content than knowing from where it comes. However, the current Internet was originally designed for host-to-host communications. Motivated by such need, the research community has proposed several network architectures for the Future Internet that shift from the current host-centric communication model to a content-centric one. These proposals are commonly referred to as Information Centric Networking (ICN) [1]. Among ICN proposals, Named Data Networking (NDN) [2] is currently the most mature solution.

Despite several advantages, NDN also introduces critical security issues in its transitional phase from IP network. Denial of service [3], content poisoning [4], cache privacy [5] are examples of such security flaws. If one wants NDN to be adopted and deployed by Internet Service Providers (ISP) in their operational infrastructures, a global security monitoring plane must be designed and implemented. This plane has to efficiently address existing security threats as a whole, while enabling the consideration of further threats that have not been revealed to date.

We propose such a monitoring plane composed of raw metrics provided by an NDN instrumentation as well as an anomaly detection engine which leverages the Bayesian Network (BN), a probabilistic graph-oriented approach that formalizes causal relationships between metrics while handling uncertainty using the probability theory. Firstly, a list of raw metrics that exhaustively feature the status of an NDN node from a data-plane perspective is established. Secondly, the structure of the Bayesian network is constructed on the basis of: (1) the proposed metrics list and, (2) the pipelines of packet processing in the NDN Forwarding Daemon (NFD [9]). Exploiting the Bayesian network, a micro detector engine that raises alarms whenever a given raw metric exceeds a given threshold is built, and we show how we integrate these micro detectors into our proposed Bayesian network to detect anomalies in NDN. Since the Content Poisoning Attack (CPA) is a current major threat in NDN where in-network caching is leveraged to spread poisonous data objects, we use this devastating attack to assess our security plane. A comprehensive study of this phenomenon has been performed [6] under real deployment conditions, thus assessing its reality as well as the different patterns that can be exploited by an attacker. We leverage the traffic data issued by these experiments to demonstrate (1) the performance of our micro detectors and (2) the capability of our Bayesian approach to detect CPA under different scenarios.

The paper is organized as follows. Section II presents related works on the NDN architecture, NDN security attacks, and the Bayesian Network. Section III presents our main contribution, including a list of metrics for the NDN monitoring plane, a micro detector raising the alarm whenever the metrics shift from normal behavior and a Bayesian Network that combines micro detectors’ alarms to infer anomalies. In section IV, by leveraging CPA data performed on a real NDN testbed, we provide numerical results that demonstrate our approach’s relevance and performance. Finally, section V summarizes the paper and presents our plans for future work.
II. RELATED WORK

A. Named Data Networking Background

As a clean-slate approach for the Future Internet, Named Data Networking (NDN) [2] is the most promising implementation among available Information Centric Networking (ICN) proposals [1] [7]. The key concept of this paradigm is to shift from the current host-centric communication model to a content-centric one. To that aim, it names each content object hierarchically instead of using IP addresses to identify hosts. Content names are accessible at the network layer and can be used to forward the content object. Besides, NDN utilizes a caching system to improve the delivery performance, as well as data signatures to ensure authenticity and data integrity. NDN defines two main types of packets for communications: (1) the Interest packet which represents a user’s request for a content name; and, (2) the Data packet which contains the actual data. Additionally, NDN uses the NACK packet [8] to notify errors between routers. This packet type has been implemented in the latest version of NFD. A router in NDN has four main components: (1) Faces which enables it to receive and forward packets; (2) a Pending Interest Table (PIT) which keeps track of forwarded Interest packets and holds reverse-path routing information for the Data packets; (3) a Content Store (CS) which caches valid Data packets that passed through the router; and finally (4) a Forwarding Information Base (FIB) which provides routing information for Interest packets.

B. Content Poisoning Attack

As an approach for the Future Internet, NDN was designed to natively tackle security issues inherent to IP network. Nevertheless, its communication model and newly introduced router components expose the network to other types of attacks [10] [11]. Among these attacks, the Content Poisoning Attack (CPA) is identified by the NDN community as one of the most significant attacks, beside Interest Flooding Attack (IFA).

In CPA, a legitimate Interest is responded to by a bad Data packets which can be inserted into the network by compromised routers or collaboration between malicious providers and consumers. Bad Data packets have valid content names, but their content is altered. Such an attack leverages NDN in-network caches to spread bad Data packets to as many users as possible. The attacker is likely to forge poisonous Data packets with popular content names to increase the attack’s impact. Proposed solutions to detect and mitigate CPA are restricted in number. They can be divided into two categories: (1) verification lessening based; and, (2) feedback (or exclusion) based. For targeted routers, a naive solution to mitigate CPA consists in checking every Data packet signature before forwarding it. However, in reality, this is impractical due to the expensive computation cost at line speed [12]. Therefore, the goal of the first category is to reduce such verification load on routers by changing the router’s verification routine [13] [14] [15] [16], or by changing the caching policy [17] to improve the resiliency to CPA. On the other hand, the solutions in the second category exploit the fact that a user can leverage the Exclude field to avoid unwanted bad Data packets by adapting the forwarding strategy [18] or prioritizing the contents that are least excluded [12]. It is worth noting that most of the previous works on this topic are based on simulated CPA. By contrast, our previous work in this area [6] provided a comprehensive study of real CPA scenarios as well as an empirical study relying on the attack implementation in a real NFD testbed.

C. Bayesian Network

The Bayesian Network (BN) [19] is a popular model in statistics to represent a set of correlated random variables. The structure of a BN is characterized by a directed acyclic graph consisting of nodes and directed edges. Each node represents a random variables $X_i$ and an edge from node $X_i$ to node $X_j$ represents a statistical conditional dependence between the corresponding random variables. Within the framework of BN, $X_i$ is called a parent of $X_j$ (i.e. $X_i \in pa(X_j)$) and $X_j$ is called a child of $X_i$. As in all Bayes statistical approaches, the relationship between variables is defined by the Conditional Probabilities Distributions (CPDs) $P[X_j|X_i]$ and the prior distribution on parent $X_i$. When a child is associated with several parents, its CDP depends on all its parents and eventually characterizes its distribution given the values of all of them. A Bayesian Network Classifier (BNC) is a BN used for classification, where one of its nodes takes values in a finite set of all possible classes or events that one needs to distinguish $C$ [20]. Given a BN structure and a set of observed data $\{x_1, \ldots, x_n\}$, BNC will return the class $\hat{c} \in C$ that has the maximum posterior estimation $\hat{c} = \max_{c \in C} P(c|x_1, \ldots, x_n)$.

Bayesian Networks have been used in a wide range of applications, especially ones related to diagnostics and predictive analytics such as medical diagnostic. In a monitoring plane with at least a dozen of metrics, it is hardly possible to represent all dependencies between variables and events. Thus, the first reason we choose BN is that it enables correlating most of the events with their impact on a small set of metrics. By repeating the process for all metrics and all variables, BN leverages all of those relations to classify the event under observation eventually. Second, BN allows designing anomaly detection at multiple levels, e.g. local detectors for aggregating local metrics, as well as a global detector for combining local detectors’ alarms. Thirdly, the observable metrics in computer networking in general are not fully predictable. BN can naturally handle the underlying random nature of observed metrics using the Bayes probabilistic approach.

III. A BAYESIAN NETWORK CLASSIFIER FOR ANOMALY DETECTION IN NDN

In order to consider the deployment of NDN in their infrastructure, ISPs will require NDN to operate safely. To that aim, any abnormal behavior potentially standing for a known or unknown attack should be addressed by a security monitoring plane that can both capture the behavior of selected metrics
and correlate them to identify potentially distributed attack patterns. To this end, this section presents the first elements towards the design and implementation of a monitoring plane for NDN. First, a comprehensive list of metrics to monitor in an NDN node is presented. Then, a micro detector is designed to raise alarms whenever a metric significantly shifts from its normal behavior. Finally, the results from all micro detectors are combined within a BN whose structure is based on a thorough expertise of the NDN specification and the NFD implementation.

A. Metric List

In this section, we propose a list of metrics to monitor in an NDN node. Such a list must be able to feature the node’s behavior and to distinguish between normal and abnormal traffic. To build an exhaustive list, all relevant components inside an NDN node are considered, including (1) the Faces; (2) the Content Store (CS); (3) the Pending Interest Table (PIT) and (4) the Forwarding Information Base (FIB). An NDN router receives and forwards packets (i.e. Interest, Data, NACK) through Faces. Obvious metrics for this component include In Interest, In Data, In NACK, Out Interest, Out Data, Out NACK, which are the numbers of incoming and outgoing packets in the sampling period. These metrics, which are similar to SNMP counters as defined in so-called Case Diagrams, allow determining various traffic’s characteristics such as the volume, the frequency, the correlation between requests (Interest) and contents (Data). Moreover, since the router can drop packets according to its strategy, it is also proposed to monitor the number of dropped packets (i.e. Drop Interest, Drop Data, Drop NACK). Because it is not expected from nodes to send and receive dropped packets, such metrics can help reveal an anomaly in NDN operations.

The CS is NDN router’s local cache. During its operation, cache misses and hits occur. Depending on the cache replacement policy, CS can decide to insert a new valid Data into its local store. Because a cache usually stores popular content to improve the delivery performance, changes in those metrics can reveal information related to the content popularity, e.g. when users prefer to watch a new trending video, or when a router is forced to cache unpopular content. For the CS, we monitor the number of occurrences of the miss (CS Miss), hit (CS Hit) and insert (CS Insert) events during a specified interval.

The PIT stands for a database where an NDN router tracks valid Interest it forwarded and to reverse-path forward Data packets. Obvious metrics for such a component rely in the number of entries created (PIT Create), deleted (PIT Delete) in a given time interval and the current number of entries (PIT Number). Moreover, since the NDN router aggregates Interest for the same content, created entries will surely be updated during the operation, leading to our choice to monitor PIT Update, the number of updates in the PIT per interval. Besides, an Interest has a lifetime period which stands for the time it keeps stored in the PIT before the arrival of a Data. If there is no Data or any notification, the matching PIT entry becomes unsatisfied and expires. Such a situation is probably related to abnormality and hence should be monitored (PIT Unsatisfied). Furthermore, the Interest lifetime can be tuned by NDN users, making it stay longer in the PIT. Deliberately increasing this lifetime could be an attempt to launch an attack (e.g. Interest flooding [21]). Such an attack type on the forwarding plane can be featured by the existing time of an entry (i.e. time elapsed from the entry creation up to its removal once satisfied). Beyond, such information can also be relevant to address network latency issues. In order to feature the existing time of entries in PIT, PIT Exist Time stands for the average of each value considered in the sampling period.

We deliberately decided not to cover the FIB in our metrics list. Different from other components involved in the metric list, the FIB belongs to the control plane, and its changes only occur due to static routing configurations or routing protocol announcements. Either way, its metrics are less likely to change as compared to those of other components and, thus, are less useful when one needs to feature the node’s behavior quickly. Moreover, we argue that the FIB malfunctions can be indirectly captured by other metrics. For instance, an increase in Drop Interest and Drop Data can mean that the FIB is not working correctly. Table I syntheses the proposed metric list and their description.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>In Interest</td>
<td>Periodic number of incoming Interest</td>
</tr>
<tr>
<td>In Data</td>
<td>Periodic number of incoming Data</td>
</tr>
<tr>
<td>In NACK</td>
<td>Periodic number of incoming NACK</td>
</tr>
<tr>
<td>Out Interest</td>
<td>Periodic number of outgoing Interest</td>
</tr>
<tr>
<td>Out Data</td>
<td>Periodic number of outgoing Data</td>
</tr>
<tr>
<td>Out NACK</td>
<td>Periodic number of outgoing NACK</td>
</tr>
<tr>
<td>Drop Interest</td>
<td>Periodic number of dropped Interest</td>
</tr>
<tr>
<td>Drop Data</td>
<td>Periodic number of dropped Data</td>
</tr>
<tr>
<td>Drop NACK</td>
<td>Periodic number of dropped NACK</td>
</tr>
<tr>
<td>CS Insert</td>
<td>Periodic number of insert in CS</td>
</tr>
<tr>
<td>CS Miss</td>
<td>Periodic number of Cache miss in CS</td>
</tr>
<tr>
<td>CS Hit</td>
<td>Periodic number of Cache hit in CS</td>
</tr>
<tr>
<td>PIT Create</td>
<td>Periodic number of PIT entries created</td>
</tr>
<tr>
<td>PIT Update</td>
<td>Periodic number of updates in PIT</td>
</tr>
<tr>
<td>PIT Delete</td>
<td>Periodic number of PIT entries deleted</td>
</tr>
<tr>
<td>PIT Unsatisfied</td>
<td>Periodic number of PIT entries unsatisfied</td>
</tr>
<tr>
<td>PIT Number</td>
<td>Current number of PIT entries</td>
</tr>
<tr>
<td>PIT Exist Time</td>
<td>Average of PIT entries existing time</td>
</tr>
</tbody>
</table>

Table I: List of metrics in an NDN node

B. Micro Detector

The metric list stands for a quantitative measure to estimate the status of an NDN node. Any aberration of these metrics could be a clue about an occurring anomaly. Therefore, in this subsection, a micro detector is presented to detect any significant change of a metric from its normal behavior. These micro detectors are built using statistical hypothesis testing theory and, more precisely, the Neyman-Pearson two-criteria approach because it allows achieving a prescribed Probability of False Alarms (PFAs).

Let us denote the \( x_i, i = \{1, \ldots, t\} \) a metric value observed at time \( i \). The foundations of hypothesis testing theory consist in modeling \( x_i \) with a statistical distribution \( P_{\theta_0} \) where \( \theta_0 \) is a distribution parameter. When some anomaly occurs, it
is expected that the distribution parameter will change to $\mathcal{P}_{\theta_1}$, with $\theta_1$ as the distribution parameter during abnormal periods. When receiving a new metric $x_t$, the problem of the micro detector is thus reduced to a choice between the following hypotheses:

$$H_0 : x_t \sim \mathcal{P}_{\theta_0} \quad \text{and} \quad H_1 : x_t \sim \mathcal{P}_{\theta_1}. \quad (1)$$

Many solutions exist to solve such a problem and among them, the Neyman-Pearson approach which aims at finding a test $\delta : \mathbb{R} \rightarrow \{ \mathcal{H}_0, \mathcal{H}_1 \}$ that satisfies:

$$\alpha_1(\delta) = \mathbb{P}_{\mathcal{H}_0} [\delta(x_1) = \mathcal{H}_1] < \alpha_0, \quad (2)$$

where $\mathbb{P}_{\mathcal{H}_i}[E]$ stands for the probability of event $E$ under hypothesis $\mathcal{H}_i$. In other words, Eq. (2) represents a constraint $\alpha_0$ on the probability of false alarm of the test $\delta$. Among all the tests satisfying this constraint, it is naturally wished to minimize the missed-detection probability, or equivalently to maximize the correct detection probability:

$$\beta(\delta; \theta_1) = \mathbb{P}_{\mathcal{H}_1} [\delta(x_1) = \mathcal{H}_1]. \quad (3)$$

Note that the correct detection probability, usually referred to as the power $\beta(\delta; \theta_1)$, depends on the distribution parameter $\theta_1$ when an anomaly occurs. When the distribution $\mathcal{D}$ belongs to the family of exponential distributions, it is possible to find an optimal test $\delta$ which maximizes the power function uniformly with respect to all values of $\theta_1 \neq \theta_0$.

Considering the necessity for simple micro detectors, we deliberately decided to model all the metrics using the normal (Gaussian) distribution which belongs to the family of exponential distributions. This distribution has already been used in our prior work on IFA detection in NDN and has shown its accuracy on real data [22], [23]. It is also assumed that an anomaly is expected to change the average values of metrics much more than their variance. Therefore, the problem considered at micro detector level is eventually defined by the following hypotheses:

$$x_1, \ldots, x_{t-n+1} \sim \begin{cases} 
\mathcal{N}(\mu_0; \sigma^2) & \text{under } \mathcal{H}_0, \\
\mathcal{N}(\mu_1; \sigma^2), \mu_1 < \mu_0 & \text{under } \mathcal{H}_1, \\
\mathcal{N}(\mu_2; \sigma^2), \mu_2 > \mu_0 & \text{under } \mathcal{H}_2, 
\end{cases} \quad (4)$$

where $n$ is the window size considered for the detection. The problem presented in (4) can be addressed easily by using a straightforward extension of Neyman-Pearson approach for multiple hypotheses referred to as “minimax constrained test”, see details in [24], [25], whose solution is simply presented here by the following test:

$$\delta(x_1, \ldots, x_{t-n+1}) \begin{cases} 
\mathcal{H}_0 & \text{if } \tau_1 \leq \sum_{t-n+1}^t x_t \leq \tau_2, \\
\mathcal{H}_1 & \text{if } \sum_{t-n+1}^t x_t < \tau_1, \\
\mathcal{H}_2 & \text{if } \sum_{t-n+1}^t x_t > \tau_2, 
\end{cases} \quad (5)$$

The thresholds $\tau_1$ and $\tau_2$ are established, to maintain the constraint (2) on false-alarm probability, as follows:

$$\tau_1 = \Phi^{-1}(\alpha_0/2) \sqrt{n} \sigma + n \mu_0 \quad (6)$$

$$\tau_2 = \Phi^{-1}(1-\alpha_0/2) \sqrt{n} \sigma + n \mu_0 \quad (7)$$

where $\Phi$ and $\Phi^{-1}$ respectively represents the standard normal cumulative distribution function and its inverse function. $\alpha_0$, as in Eq. (2), is the desired PFA of the micro detector. The Eqs. (6) and (7) show that the thresholds $\tau_1$ and $\tau_2$ are functions of $\alpha_0, n, \mu_0, \sigma^2$. While $\alpha_0$ and $n$ are chosen based on the requirements for the micro detector, $\mu_0, \sigma^2$ can be estimated from metric’s normal behavior. In short, the threshold $\tau$ can be computed in advance and guarantees the desired PFA, regardless of the metric’s behavior under attack. Moreover, using the decision threshold given in (6)–(7), the detection power of the micro detectors is given by:

$$\beta_1 = \Phi \left[ \Phi^{-1} \left( \frac{\alpha_0}{2} \right) \sqrt{n} \sigma + \sqrt{n} \frac{\mu_0 - \mu_1}{\sigma} \right] \quad (8)$$

$$\beta_2 = 1 - \Phi \left[ \Phi^{-1} \left( 1 - \frac{\alpha_0}{2} \right) \sqrt{n} \sigma + \sqrt{n} \frac{\mu_0 - \mu_2}{\sigma} \right] \quad (9)$$

As explained above, the only parameters that can be set by the user are $\alpha_0$ and $n$. While $\alpha_0$ represents the false-alarm probability, the number of samples used $n$ can be tuned to find a trade-off between quick and accurate detection. Although increasing $n$ may increase the detection power (8)–(9), it moves the thresholds (6)–(7) apart from $\mu_0$ and hence delay the detection. On the other hand, decreasing $n$ reduces the detection delay at the cost of lower detection power or higher probability of missed detection.
Incoming Interest

forwarding pipeline

Interest

Pipeline

Figure 2: Incoming Interest pipeline

C. Proposed Bayesian Network

Various attack types lead to different effects on a metric. This is why, alarms from a single micro detector cannot accurately detect and feature an occurring anomaly in an NDN node, thus making the combination of alarms from micro detectors, essential. To demonstrate the causal relationships between micro detectors, a Bayesian Network structure is proposed and depicted in Figure 1, whose node corresponds to the micro detector of a metric. The Anomaly node represents the anomalies that can occur in the NDN network. The directed edges in the BN are sketched based on NFD forwarding pipelines. A forwarding pipeline is a series of steps that operate on a packet or a PIT entry, triggered by a specific event [9]. We group NFD pipelines in four main categories that are triggered by external factors: (1) Incoming Interest; (2) Interest unsatisfied; (3) Incoming Data and (4) Incoming NACK. It is worth noting that the actual pipelines in [9] cover a lot of details in the NFD practical implementation. Due to space constraints, we deliberately simplify the pipelines to retain the most relevant information to the proposed metric list, as well as to keep the Bayesian network explanation straightforward and understandable.

1) Incoming Interest pipelines: Figure 2 illustrates the incoming Interest pipeline. When an Interest arrives, NFD first checks if it violates reserved prefix (e.g. /localhost prefix is reserved for internal communications between components) and drop it, meaning that In Interest impacts Drop Interest. Afterwards, if the Interest is duplicated with one that was already registered in the PIT, NFD sends a NACK message to notify the downstream. Otherwise, NFD will insert a new PIT entry or update the corresponding one that already exists by canceling the unsatisfied timer. Hence, Out NACK, PIT Create, and PIT Update are affected by In Interest. NFD then performs the CS lookup for a matching cached Data and enters the CS Miss or CS Hit pipelines accordingly, implying the influence of In Interest on CS Miss and CS Hit.

In case of a cache hit, the corresponding PIT entry will be removed after a while. NFD then verifies and refuses the cached Data packet if it violates reserved prefixes before sending it to downstream. As a result, CS Hit impacts PIT Delete, Drop Data and Out Data. If there is a cache miss, the corresponding PIT entry is updated once again by adding the incoming face of Interest and setting the unsatisfied

2) Interest unsatisfied pipeline: After forwarding an Interest, the NDN node waits for a Data or NACK packet from the upstream, but only for a while. Each entry in the PIT has an unsatisfy timer. When this timer expires, NFD considers that no upstream node can satisfy the Interest and remove the entry from the PIT. Hence, the PIT Delete is affected by PIT Unsatisfied. Because PIT Exist Time and PIT Number also change whenever a PIT entry is removed or created, they are also impacted by PIT Create and PIT Delete.

3) Incoming Data pipeline: Figure 3 depicts the incoming Data pipeline. When a Data arrives, NFD first checks and drops Data packets that violate any reserved prefix. NFD then verifies whether the Data matches any PIT entry. If no matching PIT entry is found, the Data is considered unsolicited. Depending on the policy of NFD, unsolicited Data can be dropped or inserted in the CS. If a corresponding PIT entry exists, the Data is also added in the CS. Note that even if the pipeline inserts the Data to the CS, whether it is stored and how long it stays in the CS is determined by CS admission and replacement policy [9]. Thus, In Data undoubtedly affects Drop Data and CS Insert.

When a Data is inserted into the CS, NFD will cancel the unsatisfied timer for each matching PIT entries, implying PIT updates. After a while, the corresponding PIT entry is deleted, and the Data packet is passed to the Outgoing Data pipeline, where NFD verifies and drops the Data if there is any prefix violation before it is forwarded to downstream. Thus, CS Insert impacts PIT Delete, Drop Data, Out Data as well as PIT Update.

4) Incoming NACK pipeline: When a NACK comes, NFD will look for a matching PIT entry. The NACK will be dropped if there is no relevant PIT entry. Otherwise, the corresponding PIT entries’ in-record will be erased, indicating PIT Update, before sending an outgoing NACK to downstream nodes. Therefore, In NACK influences Drop NACK, Out NACK and PIT Update.

IV. NUMERICAL RESULT

In this section, we present the topology and four scenarios we have considered to evaluate the proposed BNC. Next, we explain the mechanism used to extract metrics from NFD
logs, followed by setups implemented for our experiments. The micro detector’s model relevance is then evaluated. Afterwards, we address the learning efficiency of BNC with cross-validation. Finally, the performance of BNC is demonstrated regarding the impacts of the attack rate, the attack scenario and the BNC location, as well as how it can be improved by tuning the detector window.

A. Experiment Topology and Scenarios

As a follow-up of our previous work, we reuse the topology considered in [6] for our experiments, depicted in Figure 4. The topology consists of three routers: an edge router on the client side R1, a core router R2 and an edge router R3 that provides the access and caching system to the legitimate provider. Good (or legitimate) clients and bad clients (or attackers) connect to R1. On the client-side, we deploy two dedicated jNDN\textsuperscript{2}-based modules to emulate NDN traffic for the good client and the bad attacker. Good clients can avoid poisonous Data by excluding in their Interests ones that they previously received. Meanwhile, attackers assure that bad Data packets are always "fresh" and exist in the caching system by frequently sending Interest excluding legitimate Data packets. The good provider registers its prefix at R3 and responds to all the Interests with legitimate Data, while the bad provider registers its prefix maliciously at R2 and only replies with malicious Data. Due to the malicious registration, the bad provider’s route has a higher cost than that of the good provider. The average latencies are 100ms and 10ms for respectively the good provider and the bad provider.

To evaluate the performance of the proposed BNC, we reproduced the following scenarios:

1) Normal traffic: There is only legitimate traffic issued by a good client in this scenario. The number of Interest generated follows a Poisson distribution, and the requested content is selected according to a Zipf’s law. Those Interests are replied to by the Data from the good provider which connects through R3.

2) Double traffic: In this scenario, two good clients are connected through R1, and a good provider is connected through R3. Each client in this scenario behaves like in Normal traffic. We consider this scenario not only to the explore metrics’ behavior in case of abrupt changes in the network traffic but also to provide a learning dataset that challenges the BNC, by evaluating to what extent it can distinguish legitimate traffic changes against malicious ones.

3) CPA Best Route: In this scenario, router R2 uses the best route forwarding strategy which stands for the default setting in NFD. When an Interest packet arrives in R2, it is forwarded to the good provider. While waiting for Data, if R2 receives another Interest for the same content, the Interest will be sent to the bad provider as the second choice in the forwarding table. Thanks to the shorter delay, the bad provider will succeed in caching the bad Data packet in R2 before the good provider.

4) CPA Multicast: The second CPA scenario leverages the multicast strategy which is a second forwarding strategy of NFD, where router R2 forwards a received Interest packet to all of its next hops. Thanks to its shorter latency, the bad provider will reply to the Interest faster. Hence bad Data packets will be injected into R2’s cache.

More detail on the two CPA scenarios is given in [6]. We argue that the unsolicited scenario in [6] can be prevented by a patch of NFD, and hence it is not reproduced in this work.

B. Metric Extraction Mechanism

The NFD Management Protocol\textsuperscript{3} enables collecting data related to the status of an NDN node (e.g. In Interest, PIT Number). However, given the metric list that we need to collect, these statistics are not sufficient. To the best of our knowledge, currently, there is no mechanism to collect metrics that are unavailable in NFD Management Protocol, such as CS Hit, CS Miss, CS Insert, Drop Interest, Drop Data, Drop NACK. Therefore, we built a tool to collect those metrics. Since NFD is still under development, we avoid modifying its implementation and instead extract the necessary information from NFD log. For this purpose, we set up a monitoring probe in the NDN routers' systems to extract metrics we need. The monitoring probe we considered is the Montimage Monitoring Tool (MMT)\textsuperscript{4}, which is dedicated to the monitoring of network traffic, log files and application traces. Each log entry in NFD log corresponds to an event in NFD, and it indicates: (1) the timestamp; (2) the event name; (3) the face; and, (4) the corresponding Interest. Explicit metrics can be interpreted directly from these log entries, while implicit metrics can be deduced from the log. For instance, events such as OnIncomingInterest, OnContentStoreMiss, onOutgoingInterest provide information to directly update In Interest, CS Miss and

---

\textsuperscript{2}https://github.com/named-data/jndn

\textsuperscript{3}http://redmine.named-data.net/projects/nfd/wiki/Management/

\textsuperscript{4}http://www.montimage.com/products.html
for normal traffic. The objective of this setting is to help the micro detector understand its normal behavior so that the performance for various metrics can be demonstrated in the same figure. For most of the metrics (e.g. CS Hit, In Interest), the empirical and the theoretical PFA match closely, implying the ability to guarantee the prescribed PFA of the micro detector and the relevance of the model. Meanwhile, for a few metrics (e.g. PIT Number), our micro detector cannot ensure the performance for small prescribed PFA. As stated in the previous subsection, this phenomenon is due to the fact that these metrics are not well modeled by the normal distribution. However, as shown in the following section, a performance enhancement is possibly obtained by combining micro detectors, hence the modeling errors on their distribution is compensated by each others.

E. Learning Parameters of Proposed Bayesian Network

To evaluate the learning efficiency of BNC when the size of training set varies, we use the usual k-folds cross-validation method with $k = 5$. Consequently, the dataset is divided into 5 subsets. Each subset will, in turn, be used as testing data and the remaining four will be used as training data. The average misclassification rate (i.e. the total number of misclassified samples over the total number of samples) over training subsets is defined as $train\ error$, while the one obtained over testing subsets is called $cross-validation\ error$. Figure 5c shows the learning curves of the proposed BNC when the size of training dataset per scenario changes. When the size of
training set per scenario increases, the misclassification error starts decreasing. An optimal value is achieved around 280 training samples per scenario. After that, the misclassification error keeps increasing due to the well-known phenomenon of over-fitting. Therefore, the optimal value of 280 samples for the training set per scenario has been chosen, corresponding to about 23 minutes of collecting samples.

F. Bayesian Network Classifier Evaluation

The graphs in Figure 6a and 6b respectively trace the accuracy (i.e. the total number of samples classified correctly over the total number of samples) and the delay of the proposed BNC when the attacker rate changes. Regarding the attack rate’s impact, the figures indicate that when the attack rate is less than the Interest rate under normal traffic, BNC has a low accuracy, a high delay and those results have a large statistical spread. On the other hand, BNC achieves over 95% of accuracy with a delay of about one sample when the attack rate is higher than normal user rate.

With regards to the impact of attack scenario, one can note from Figure 6a and 6b that the accuracy and delay of BNC against CPA Multicast is better than in CPA Bestroute. The reason is that, in CPA Multicast, the Interests are more likely to be forwarded to the bad provider without much effort from the attacker, so the behavior of attack in any rates is visible.

Concerning to the location’s impact, in CPA Bestroute, BNC at R1 is more accurate than in R2 router because it receives Interest packets from clients and attackers first. As a result, its metrics will be more affected than those of R2. Meanwhile in CPA Multicast, since R2 is more likely to be poisoned due to the delay advantage of the bad provider, its metrics are impacted more obviously than that of R1. Therefore, BNC in R2 achieves a better accuracy than in R1.

Eventually, it is important to point out that the accuracy of BNC can be improved by increasing the detection window \( n \) of the micro detector. As shown in Figure 6c, for the worst case of CPA Bestroute with 1 Interest/s, the accuracy increased from 53% up to 93% by raising the \( n \) from 1 to 5. Nevertheless, it is worth noting that increasing the detection window will increase the detection delay of micro detectors (Eq. 8-9). Therefore, this trade-off should be considered carefully in the deployment.

V. CONCLUSION AND FUTURE WORK

In this paper, the first elements towards the design and implementation of a security monitoring plane for NDN have been proposed. To that aim, a careful selection of NFD metrics has been performed and for each of them, micro detectors, able to consider any abnormal variation, have been designed and evaluated both theoretically and empirically. By leveraging a Bayesian Network Classifier, the correlation between metrics have been featured, thus allowing the detection of any abnormal behavior in an NDN node. In order to validate our proposal, two attack scenarios of the Content Poisoning Attack have been considered in a real testbed, and they demonstrate the capability of our solution to accurately detect these attacks at different network locations and with various rates.

Our future research directions will focus on: (1) addressing different types of attacks to assess the genericity of our approach; (2) implementing the solution into MMT probes to contribute to the secure deployment of NFD; and (3) distributing the BNC to correlate different alerts issued by different nodes, thus enabling a potential trace-back mechanism.

ACKNOWLEDGMENT

This work is partially co-funded by (1) the French National Research Agency (ANR), DOCTOR project, \(<\text{ANR-14-CE28-0001}>\), started in 01/12/2014 and supported by the French Systematic cluster and (2) the CRCA and FEDER CyberSec Platform, \(<\text{201304601}>\).

REFERENCES


