Content Poisoning in Named Data Networking: Comprehensive Characterization of real Deployment

Tan Nguyen, Xavier Marchal, Guillaume Doyen, Thibault Cholez, Rémi Cogranne

Content Poisoning in Named Data Networking: Comprehensive Characterization of real Deployment

Tan Nguyen*, Xavier Marchal†, Guillaume Doyen*, Thibault Cholez† and Rémi Cogranne*
*ICD - STMR - UMR 6281 CNRS
Troyes University of Technology, Troyes, France
Email: {ngoc_tan.nguyen, guillaume.doyen, remi.cogranne}@utt.fr
†LORIA, UMR 7503 (University of Lorraine, CNRS, INRIA), Nancy, France
Email: {xavier.marchal, thibault.cholez}@loria.fr

***Abstract*—Information Centric Networking (ICN) is seen as a promising solution to re-conciliate the Internet usage with its core architecture. However, to be considered as a realistic alternative to IP, ICN must evolve from a pure academic proposition deployed in test environments to an operational solution in which security is assessed from the protocol design to its running implementation. Among ICN solutions, Named Data Networking (NDN), together with its reference implementation NDN Forwarding Daemon (NFD), acts as the most mature proposal but its vulnerability against the Content Poisoning Attack (CPA) is considered as a critical threat that can jeopardize this architecture. So far, existing works in that area have fallen into the pit of coupling a biased and partial phenomenon analysis with a proposed solution, hence lacking a comprehensive understanding of the attack's feasibility and impact in a real network. In this paper, we demonstrate through an experimental measurement campaign that CPA can easily and widely affect NDN. Our contribution is threefold: (1) we propose three realistic attack scenarios relying on both protocol design and implementation weaknesses; (2) we present their implementation and evaluation in a testbed based on the latest NFD version; and (3) we analyze their impact on the different ICN nodes (clients, access and core routers, content provider) composing a realistic topology.**

I. Introduction

The Internet pursues its fast-paced evolution with more and more users and bandwidth-consuming services, putting a high pressure on the underlying infrastructure. To address this challenge, several disruptive networking technologies have emerged. Among the most promising ones, Information-Centric Networking (ICN) architectures, and particularly Named-Data Networking (NDN), are based on the observation that the main usage of today's Internet is related to content diffusion. They propose a paradigm shift to optimize data delivery by moving from host-centric diffusion mechanisms to content-centric ones. More precisely, content names are addressable at the network level and content can be delivered in a multicast manner and from any nodes, thanks to stateful NDN routers including a caching capability to optimize the delivery of popular data.

After a few years of research, the ICN paradigm, and especially the NDN solution, is now mature enough to move to an implementation and deployment stage, thus enabling telco-operators to consider it as a viable alternative to the legacy IP stack. However, the security of ICN protocols, as well as their implementation must first be assessed to make them safe alternatives that could be easily adopted by potential stakeholders. In that effort, we focus on the NDN proposal and its NDN Forwarding Daemon (NFD), which stands for the most acknowledged ICN solution in the research community. While most research efforts have been focused on caching performances or on the specific Interest Flooding Attack (IFA), less attention has been drawn to cache-related issues. More precisely, the Content Poisoning Attack (CPA) is identified by the NDN board as the next most important threat related to NDN right after IFA[1], while not well investigated so far. However, one of the most important issues of CPA relies on the lack of a detailed implementation, as well as a dedicated and comprehensive study of the phenomenon in "real life scenario". This prevents researchers from acquiring insights into feasibility and impact of CPA and, of course, from designing solutions that could improve the protocol resilience against CPA.

In order to overcome these deficiencies, we propose a detailed description of CPA in realistic deployments and conduct a comprehensive characterization of this attack. We present the results of a complete measurement campaign achieved through a rigorous methodology where (1) we define three realistic attack scenarios leading to CPA despite recent protection mechanisms; (2) we implement them on a testbed and measure the effect of attack parameters through many experiments; (3) we evaluate the impact of the attack on the main network actors (clients, access and core routers, content provider); and (4) we propose a first selection of metrics to characterize these attacks.

The rest of the paper is organized as follows. Section II presents the related work on NDN and on the CPA. Section III describes the attack scenarios that are deeply investigated in Section IV through extensive experiments to evaluate and characterize their impact on the network. Finally, Section V gives our conclusion on the current threat of CPA.

II. Related works

A. Named Data Networking background

Named Data Networking [1] is one of the most accomplished proposals for Information Centric Networking [2] [3],

[1] see http://named-data.net/project/faq/

a recent research effort for a clean-slate network for the Future Internet, and consequently the most promising candidate for a potential deployment. The key concept of NDN relies on a network based on named content objects, organized into a hierarchical scheme, instead of legacy IP addresses. Communications in NDN are performed by two types of packets: (1) *Interest* and (2) *Data*. A user sends an *Interest* packet to express his request for a content, then receives a corresponding *Data* packet in return, together with a signature to ensure its integrity and authenticity. Users can even avoid unwanted versions of a *Data* by specifying their suffix component in the *Exclude* field of an *Interest*. In NDN, a router has many faces - a generalization of interfaces in IP networks - and it owns three internal components. First is the *Forwarding Information Base* (FIB) which contains routing information for *Interest* packets. Secondly, the *Pending Interest Table* (PIT) contains entries for each forwarded *Interest*, and uses them as reverse-path routing information for *Data* delivery. Finally, the *Content Store* (CS) is an essential local cache which stores recently requested content to reduce congestion and improve performances.

B. Cache-related attacks

Caching is an important feature in ICN. However, it also exposes the network to other threats [4] [5] [6] [7]. Attackers can exploit caches to obtain unauthorized information or to sabotage the system. Beside the CPA, which will be addressed in this paper, other caching-related attacks can be classified into two other types: time analysis and cache pollution. In most related papers [7] [8] [9], the time analysis is described as an attack exploiting the difference in delivery delay of *Data* from the original provider and cached copies, to learn about users' recent requests, thus violating their privacy.

In cache pollution, an attacker forces caches to store content irregularly in order to ruin the cache's performance for legitimate users [10] [11] [12]. This attack has been widely studied in IP, especially for web-caching. The main goal of this attack is to degrade caching performance, by sending more requests for unpopular content. Popular contents will be found less frequently in caches, hence more requests will be forwarded upstream, increasing network traffic.

C. Content Poisoning attack

In CPA, legitimate *Interests* are still responded, but by malicious *Data*[2] which are possibly inserted by (1) compromised routers or (2) collaboration between bad providers and clients. Those *Data* still have valid content names but the content was altered. Such attack leverages NDN in-network caches to spread bad *Data* to as many users as possible. The attacker is likely to forge poisonous *Data* with popular content names to increase the attack's effect scale. In [13], the authors indicated two types of poisonous *Data* in CPA: (1) *Corrupted* and (2) *Fake Data*. The content in both cases is modified. In the first type, the bad provider does not own the valid signing information to correctly sign the modified content. This type of *Data* is more likely to be created but can be detected by verifying its signature which will fail. In the second case, the bad provider has the valid information to correctly sign the packet, leading to a successful signature verification. Therefore, it is harder to create but impossible to detect by end-systems.

[2]In the following, we indifferently use the terms *bad*, *compromised*, *malicious*, *poisonous* to mention entities or packets under attacker control; and the terms *legitimate*, *good* to mention legitimate entities and packets.

D. Content Poisoning detection and mitigation

To date, proposed solutions to detect and mitigate CPA are restricted in number and can be divided into two categories: (1) verification lessening and (2) feedback (or exclusion) based. Routers can mitigate CPA by verifying *Data* before forwarding. However, in reality, it is impractical for a router to do so due to the expensive computation cost at line speed [14] and its inapplicability to the second CPA type. Therefore, the main goal of the first approach is to reduce such verification load on routers, by changing router's verification routine [15] [16] [12] [17], or caching policy [18] while maintaining its resilience against CPA. A typical solution for this category is the proposal of Kim et al. [16]. In this approach, a router accepts to cache all the *Data* it forwards, but only verifies them when there is a cache-hit. Successfully verified *Data* are forwarded without further verification. This significantly reduces the routers' load and still maintains verification for popular content. However, this solution only solves the problem locally. Bad clients can still re-issue bad *Data* to insert them in a cache, or increase the load on the router by sending *Interests* to create cache-hit for unpopular content.

On the other hand, the solutions in the second category [19] [14] exploit the fact that a user can leverage the *Exclude* field to avoid unwanted bad *Data*. *Content ranking* [14] is a typical solution in this category in which a router ranks its cached copies based on three features of users' exclusions: (1) number of exclusions; (2) time distribution and (3) number of exclusion's incoming faces. Cached copies with higher rank are more likely to be returned to users. However, one of the drawbacks is that it relies on exclusions issued by clients, and thus it is more likely to be compromised. In addition, exclusion is a part of content exploration in NDN. Hence, good *Data* is possibly marked as bad *Data* when users exclude it to reach different content.

Besides their own weaknesses, previous works on CPA share some common drawbacks. First is the inconsistency in CPA's impact evaluation. Since their authors usually couple such an evaluation with their proposed solutions, the understanding of this phenomenon is partial and biased towards emphasizing the proposed solutions. Therefore, the results are neither re-usable nor comparable. Secondly, simulation scenarios rely on a one-shot attack in which clients often stop when receiving good *Data*, while CPA is more likely going to operate as a flow, hence leaving blind spots about the phenomenon. Thirdly, most of the previous works overestimate the CPA with unrealistic behaviors. For example, they do not enable the use of *Exclude* field to avoid bad *Data*, or consider pre-polluted caches with an impractically high

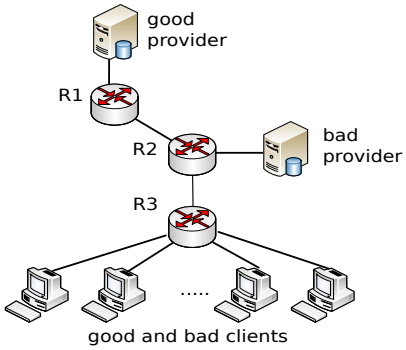


Figure 1. Use-case topology for Content Poisoning Attack

percentage of bad *Data*. Although there are explanations on the attack scenario and how bad *Data* is inserted in caches, they are insufficient to explain why CPA can achieve such high percentage of bad *Data* in caches.

These common drawbacks elevate the necessity for a devoted and in-depth investigation for CPA before developing any detection or mitigation for this attack. To that aim, we propose (1) to move from a simulation environment to a real testbed deploying the NFD implementation of NDN; (2) by a detailed protocol, implementation understandings and analysis, to discover novel attack scenarios and to formalize existing ones and (3) to evaluate exhaustively the attack's impacts on all of the involved components (clients, routers and content providers), which will be presented in the upcoming sections.

III. Attacks scenarios

In this section, we describe the topology we implemented to study the CPA's impact, as well as three attack scenarios that can be exploited to insert bad *Data*. These are: (1) *unregistered* remote provider; (2) *multicast* forwarding and (3) *bestroute* forwarding. We do not consider the case of a compromised router which can easily respond to an *Interest* with a bad *Data*, since taking the control of a network element is hardly feasible in a real operating context while leveraging end-hosts, on both the provider and consumer sides to perform the attack, is easier. Consequently, our three scenarios consider cases where CPA is carried out by a single unregistered remote provider or by a collaboration between bad providers and clients coordinated by a single attacker.

A. Topology and entities' behavior

The topology implemented to study CPA is illustrated in Fig. 1. We argue that this topology, together with the behaviors of all implied components, is comprehensive enough to achieve the main purpose of proving the feasibility and featuring CPA impacts because of the following features.

First, it exhibits the general role of the different nodes/-functions involved in this attack in NDN and reflects a typical network operator structure: a core router R2, with a large cache, where providers are located; an intermediate router R1 on the R2's route toward the good provider, which helps reducing the delay of its *Data*; and an access router R3, with a smaller cache, where both good and bad clients are connected. That way, this topology enables us to highlight the effect of CPA for all involved components.

Secondly, the user behavior is as close as possible from reality. We assume that clients always issue *Interest* packets for fresh and latest *Data* of the content (thanks to *MustBeFresh* and *ChildSelector* bits of *Interest* packets). Legitimate ones send requests over the whole content popularity and only accept good *Data* after verifying. When receiving bad *Data*, they re-issue another *Interest* with the *Exclude* field enabled to avoid the previously received bad *Data*. Routers would not verify *Data* due to the impracticality of verification at line speed [14]. Bad clients also request for the whole content popularity, but for the contents they target, they act in an opposite way by excluding good *Data*, thus favoring the dissemination of bad *Data*.

Thirdly, our topology only counts one good provider, who replies to *Interests* for the whole content popularity. This case stands for one of the worst cases where legitimate content availability is limited to one route. In addition, the legitimate provider is located farther from R2 than the malicious one, leading to a longer delay in the content delivery. However, the cache on intermediate router R1 helps in saving such delay, once *Data* is stored there. Besides, due to its valid registration, we consider that the path toward the legitimate provider always has a lower cost than that toward the bad provider. In addition, the good provider will automatically update its *Data* when the freshness period[3] expires.

Moreover, it is necessary to mention that, in a network aiming to optimize the delivery efficiency and support content availability, traditional secure routing (e.g. NLSR [20]) is too restrictive and consequently not fully acknowledged as a realistic deployment case of NDN [19]. Besides, NDN board stated that "The namespace management is not part of the NDN architecture". Therefore it is not easy to answer the question precisely on prefix registration security, hence NDN is still exposed to malicious registration. Hence, we argue that content providers can openly publish their contents under some registered prefixes. Such registration might be simple for a legitimate provider, but is not straight for an attacker to get many providers registered. Hence our topology only counts one bad provider which is located near the core router R2. This bad provider only replies to *Interests* for the contents it chooses to poison. To challenge the legitimate provider, we consider that the attacker always selects the most popular contents to poison, given a popularity distribution. Such knowledge can easily be gained, e.g. in the case of web traffic through publicly available information about website popularity ranking.

Besides, since bad *Data* is useless after being excluded, the bad provider must update its *Data* whenever it receives an *Interest* excluding current bad *Data*, to maintain the attack persistence. To increase the number of victims who may receive bad *Data* by issuing naive requests (i.e. without exclusion for bad *Data*), the attacker can set the bad *Data*'s *FreshnessPeriod* to a high value.

[3]see http://named-data.net/doc/NDN-TLV/current/data.html

B. Unregistered remote provider scenario

Unregistered remote provider scenario proposes to exploit a weakness we discovered in NDN implementation which exhibits an unspecified behavior [21]. For each incoming *Interest*, the NDN forwarding component keeps track of the requested content name and the faces to which the *Interest* is forwarded in a corresponding PIT entry. However, these out-records seem to be only used for NACK (negative acknowledgment) packets processing. When a *Data* is received, the forwarder only performs a PIT match checking. This means that *Data* coming from any faces can satisfy a pending *Interest*, even from a face to which this *Interest* has not been forwarded. Exploiting this flaw, the attacker can deploy an unregistered remote provider by taking control of any client connected to one of the nodes (R2 in our topology) on the path between clients and a good provider, and makes it send bad *Data* so that those *Data* can match pending *Interests* on routers toward the clients.

One should note that the bad provider is blind to the consumers' *Interest*, but can still get information on the recently requested contents, thanks to a time analysis attack [5]. When an *Interest* arrives at R2, a race condition begins between the good and the bad providers. Only the first matching *Data* packet received by R2 will be accepted and will consume the PIT entry, while all the latter are dropped, as long as a new PIT entry is not recreated. As a consequence, a malicious *Data* packet has a higher chance to match an *Interest* for a targeted content if it arrives at R2 during the time window $[t_{receive}; t_{receive} + t_{gpDelay}]$; where $t_{receive}$ is the time when R2 receives the *Interest* and $t_{gpDelay}$ is the delay of corresponding *Data* from the good provider. Since estimating this time window is hardly feasible for the bad provider, it must send poisonous *Data* for targeted contents regularly at a sufficient rate to increase the success chance of the attack.

C. Multicast forwarding scenario

Multicast is one of the possible forwarding strategies integrated into the current NDN implementation [22]. When a router using this strategy receives an *Interest*, it forwards it to all faces registered in the corresponding FIB entry. While the CPA is carried out by the sole effort from the bad provider in the *unregistered remote provider* scenario, the *multicast* scenario requires collaborating clients in order to pull and insert poisonous *Data* in caches. Especially, bad clients regularly send *Interests* only for the targeted content name, but exclude the current copies of *Data* in order to bypass caches. This forces R3 to forward the requests toward R2, which will, in turn, forward them to both the legitimate and bad providers, according to the *multicast* forwarding strategy. Consequently, a *Data* packet is returned by both providers. However, due to the shorter delay, the bad *Data* arrives at R2 first, consumes the corresponding PIT entry and stores it in the cache of R2 and R3. Meanwhile the legitimate *Data* is dropped due to its late arrival that prevents it from matching any PIT entry on R2.

D. Best route forwarding scenario

Best route is the default forwarding strategy used by the current NDN forwarder [22]. A router running this strategy forwards the incoming *Interest* to the face with the lowest cost in the corresponding FIB entry. If there are two faces with the same lowest cost, the router will use the first one registered. After an *Interest* is forwarded, a similar *Interest* with the same content name, selectors but different nonce (a random value to avoid *Interest* looping) would be suppressed if it's received during a retransmission suppression interval. An *Interest* received after this interval is considered as a retransmission, and will be forwarded to the next lowest-cost face that has not been previously used, hence opening the door for the bad provider to act. When all registered faces in the FIB entry have been used, it is forwarded again to the first-used face. Thanks to the collaboration with malicious clients, the attacker can generate additional similar *Interests*, forcing router R2 to use the other route to the bad provider, hence pulling the bad *Data* to cache in R2 and R3.

IV. Attack evaluation

In this section, we present the experimentation setup implemented on the basis of the topology presented above and the attack scenarios identified. Then, we evaluate the attack's impact from the client's side as the main target, and then on the provider's and router's sides since they also suffer from collateral damages. It is necessary to mention that in our figures, red, green and blue colors stand for *bestroute*, *multicast* and *unregistered remote provider* scenario respectively. Finally, by leveraging a Principal Component Analysis, we reveal synthetic attack patterns for all scenarios.

A. Experimental setup

To proceed with our experiments, we deployed the topology described in Fig. 1 using NFD (0.4.1) on Docker containers, one container per component. We configured an artificial latency on both the good and bad providers in order to emulate a more realistic network distance between a server and its users in the real Internet. In our experiments, all clients and providers are remotely connected to NFD nodes so that content caches are only present on the NDN routers. The constant values shared by all our experiments are listed in Table I and most of the values are motivated in [23]. Each experiment lasts 600 seconds, with first 300 seconds spent without attack in order to compare the statistics before and after the attack. Finally, each experiment is run 5 times and all the curve points depicted subsequently stand for the average of the 5 results bounded with a 95% confidence interval.

1) Attack parameters: we consider the attack rate as the main parameter that impacts the attack success. For *unregistered remote provider* scenario, this parameter stands for the number of bad *Data* sent per second by the *unregistered* remote provider, varies in range $[10, 1000]$ following a logarithmic scale and is set to 50 Data/s as a default value. We do not run experiments for the range from 1 to 10 (i.e. less than good Interest rate) since the CPA is barely successful

Constant	Value
Good provider content	10000 contents
Good provider content freshness	90 sec
Good provider link latency	100ms
Bad provider content freshness	120 sec
Bad provider link latency	10ms
Big cache size	1000 contents
Small cache size	500 contents
Users *Interest* rate	10 content/sec
Zipf distribution factor	1.5
Max exclude components	700

Table I
EXPERIMENTAL CONSTANTS

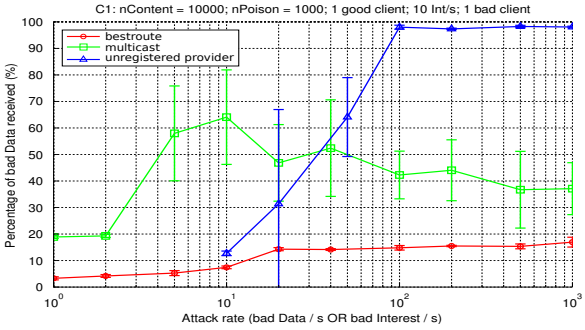


Figure 2. Attack rate effect on the legitimate client

with such attack rate. For *multicast* and *bestroute* scenarios, it stands for the number of bad *Interests* injected per second, varies in a range $[1, 1000]$ following a logarithmic scale and is set to 10 Interests/s, i.e. equals the good *Interest* rate. As a second parameter, we also considered the fraction of the most popular content which is poisoned by the attacker. The value varies in a range $[0.01, 10]$ percentage of content, grows in a logarithmic way also and is set to 1% as a default value.

2) Client behavior: The client behavior is implemented as described in section III-A However, the client cannot exclude NDN names indefinitely, due to packet size limitation of the NDN protocol. Therefore, it will consider a content unreachable when the *Exclude* size reaches a defined value and will ask this content later with a new empty *Exclude* field. After the client receives a legitimate *Data*, it still maintains excluded names in a database to reuse them in the future when asking for that content again. We decide to use this way to proceed over the naive way (no memory about the previous trials) because the client doesn't know whether previously received bad *Data* still exist in the network or not.

B. Impacts on legitimate client

Fig. 2 and Fig. 3 reveal the effect of CPA on legitimate clients. They depict the percentage of bad *Data* a good client receives when it requests for content according to the attack rate (Fig. 2), and according to the number of targeted contents (Fig. 3).

Fig. 2 shows that in the *best route* scenario, the legitimate client suffers the least from CPA. The damage just slightly increases when the attack rate is higher than that of legitimate traffic and seems unchanged for higher attack rates. Since the legitimate provider is located on the lower-cost route, the requests are always prior forwarded to this provider, and R2

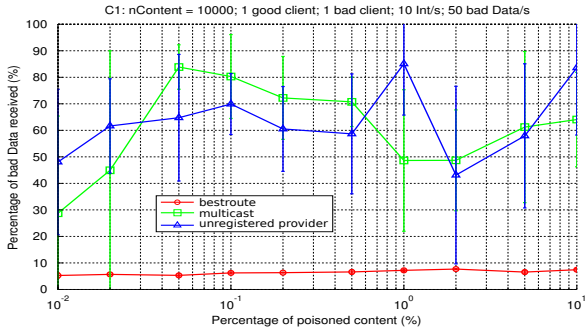


Figure 3. Number of poisoned contents effect on the legitimate client

only uses the other route when there is a re-transmission during the delay of the good provider, giving the bad provider low chance to insert bad *Data*. The *multicast* and *unregistered remote provider* scenarios poison the legitimate client more effectively. Especially for the *unregistered* one with a high attack rate, nearly 100% of *Data* received is poisonous. Under high attack rate, incoming *Interests* in R2 are mostly matched by fresh and new bad *Data*, despite the effort of exclusion. On the other hand, the effect of *multicast* tends to decrease when the attack rate increases. In the *multicast* scenario, when the attacker sends too many *Interests*, more good *Data* will be pulled to R1's cache, giving good *Data* a better delay than *Data* come from bad provider. Hence, when an *Interest* on R2 is forwarded to both routes, good *Data* now have a higher chance to arrive at R2 sooner. Fig. 3 shows that even when the attacker changes the number of content he targets, the *best route* scenario still maintains its protection against CPA. Furthermore, for *multicast* and *unregistered remote provider* scenarios also, the number of target contents does not have a clear impact on the legitimate client. This implies that if an attacker wants to improve the damage on legitimate clients, he should not put much effort on expanding the number of target contents, but rather focus on a few highly popular ones.

C. Impact on the content provider

Fig. 4 and Fig. 5 depict the side effect of the CPA on the legitimate provider. More specifically, they show the mean number of additional *Interests* that the provider must handle, as compared to the phase without attack, according to the attack rate in Fig. 4 and according to the number of targeted content in Fig. 5. The unit of the attack rate differs between attack scenarios, as explained in section IV-A1.

Fig. 4 shows that the *unregistered remote provider* scenario does not increase the legitimate provider's load whatever the attack rate. This can be explained by the nature of this attack which is not driven by *Interest*, contrary to the other two. Although an unregistered remote provider may consume a PIT entry on R2 with its bad Data, the legitimate Interest that created this PIT entry has already been forwarded to the legitimate provider beforehand, even if the returning Data will be dropped. This means that the provider has less hint to detect CPA in this scenario. On the opposite, the effect on the provider's load is highly related to the attack rate in the case of the two other scenarios. Since most *Interests* issued

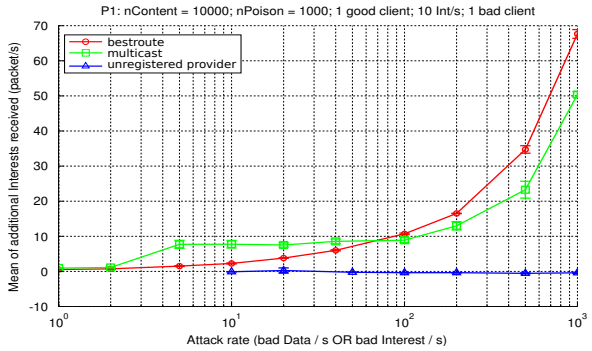


Figure 4. Attack rate effect on the legitimate provider

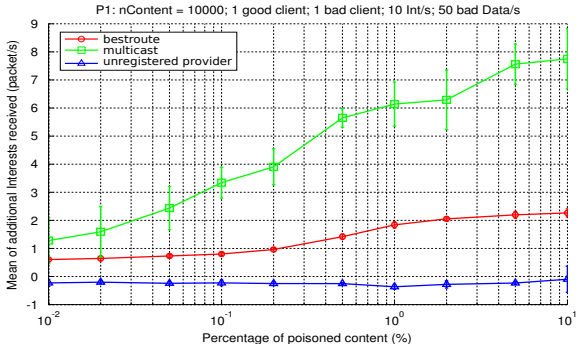


Figure 5. Number of poisoned contents effect on the legitimate provider

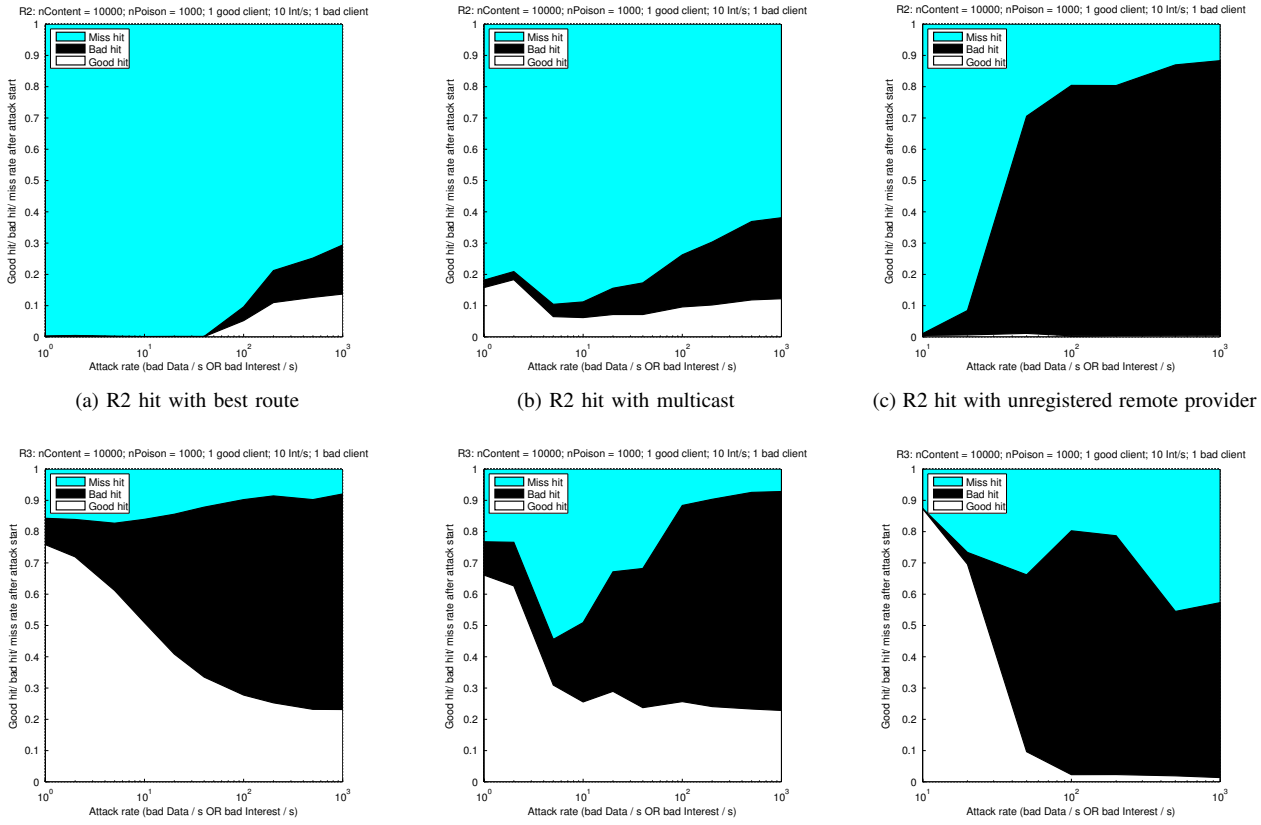
by bad clients will be forwarded to the legitimate provider, he must handle more requests when the attack rate increases under the *best route* or *multicast* scenarios which both exhibit this similar trend.

Fig. 5 shows that the *unregistered remote povider* scenario has no impact on the legitimate provider whatever the range of attacked content. As explained previously, this is due to the fact that this attack does not rely on *Interests* to propagate pollution but directly issues bad *Data*. For *best route* and *multicast* scenarios, Fig. 5 shows a nearly linear growth of the amount of additional *Interests* received by the legitimate provider with the range of targeted content. Indeed, when a bad user targets a wider range of content, each emitted *Interests* has fewer chances to be gathered on a given PIT entry as the range of names is wider, and consequently more *Interests* are forwarded to the provider.

D. Impact on routers' cache

This section deals with the impact of the CPA on the most important routers of our topology: core router R2 and access router R3, as illustrated in Fig. 1. Measuring the effect on caches is of prime importance considering the central role of caches to avoid network congestion in the NDN architecture, and consequently the high amount of resources dedicated to caching. It is even more critical if we consider that attackers can exploit network caches to maintain and amplify the pollution at a reduced cost. The two rows in Fig. 6 respectively illustrate measurements for R2 and R3. They show the average proportion of good hits (the real *Data* is retrieved), bad hits (a corrupted *Data* is retrieved) and missing hits (no *Data* is retrieved) in routers' cache according to the attack rate for each of the three attack scenarios that have a dedicated sub-figure. Fig. 7 and Fig. 8 then try to catch the attack efficiency in corrupting the cache by measuring the effect of the attack rate on the percentage of cache insertions causing a bad hit on router R2 and R3, respectively. We first consider core router R2. We can already notice that even without an attack, R2's cache is not useful when solicited as we can observe a very large proportion of miss hits at the lowest attack rates. Fig. 6a shows that the *best route* attack does not affect the R2 cache at low rates and only has a limited impact with only 10% of bad hits with the highest attack rate. This can be explained by the fact that the client will retrieve the majority of polluted *Data* from R3 and consequently, the majority of *Interest* forwarded to R2 by R3 (i.e. after a cache miss on R3) already exclude the names of most of the bad *Data* preventing bad hits in R2 cache. In the case of the *multicast* attack (Fig. 6b), the effect is globally higher with a proportion of bad hits increasing from 2% to 25% with the attack rate. The *multicast* scheme offers better opportunities for the bad provider to answer back with polluted *Data* which can explain this result. Finally, the *unregistered remote provider* attack exhibits a totally different behavior on R2 in Fig. 6c. The ratio of bad hits increases rapidly with the attack rate, achieving 80% of bad hits for an attack rate of 100 bad *Data* per second. Then, increasing the attack rate as less impact as the bad hit ratio only increases by 5% with 10 times more aggressive attack. This attack, if very effective in propagating pollution through the cache as the flow of newly generated bad *Data*, easily enters the cache by consuming legitimate *Interests*. The attacker being always one step ahead of the client excluding names also explains the lower proportion of miss hits.

Looking at the general aspect of the curves in the second row of Fig. 6, we can already see that the cache of the access router R3 shows a totally different behavior from the core router R2 when exposed to the same attacks. First of all, R3 is more prone to cache *Data* as we can observe a very high proportion of good hits at lowest attack rates. Then, all attacks show a similar trend with the rate of good hits decreasing to the benefit of bad hits when the attack rate increases. This effect is however more progressive for the *best route* (Fig. 6d) than for *multicast* (Fig. 6e) which only increases the proportion of miss hits at lowest attack rates, before increasing the bad hits when the attack rate reaches 500 *Interests* per second. In both cases, at the highest attack rates, the proportion of good hits is small ($\simeq$20%) while the proportion of bad hits is very high ($\simeq$70%), and the miss hits ratio is under (10%), making both attacks very successful on R3. The aforementioned trend is made quicker by the *unregistered remote provider* scheme, but the maximum attack rate ends with a different ratio from the other two. There are no more good hits in Fig. 6f but the cache hits are balanced between 60% of bad hits and 40% of miss hits. However, lower attack rates can achieve better results for this specific attack with a proportion of bad hits going up to 80% for 100 bad *Data* per second.



(a) R2 hit with best route (b) R2 hit with multicast (c) R2 hit with unregistered remote provider

(d) R3 hit with best route (e) R3 hit with multicast (f) R3 hit with unregistered remote provider

Figure 6. Attack rate effect on caches of core router R2 (a)(b)(c) and access router R3 (d)(e)(f) for each attack scenario

To conclude on the attack level, we can state that the *unregistered remote provider* scheme does a better job in polluting the routers in the path toward the client, but its efficiency decreases when it traverses routers. An alternative scenario with the unregistered remote provider connected to the access router, reproducing Fig. 6c closer to the clients, would make it even more difficult for them to obtain good Data. We can also notice that the larger amount of miss hits on R3 compared to the other scenarios have good chances to end up as bad hits on R2. Concerning the *best route* and *multicast* scenarios, they have a less significant impact on the core router R2 but still have a high impact on the good hit depletion on access router R3. Fig. 7 and 8 are perfectly inline with the previous results. For instance, we can notice a peek in attack efficiency at the rate of 50 bad *Data* per second for the *unregistered remote provider* scenario on R2, which matches the start of the flat behavior on Fig. 6c. Similarly, the maximum efficiency of the *unregistered remote provider* attack at 100 bad *Data* per second matches the attack rate corresponding to the highest ratio of bad hit in Fig. 6f. The *best route* and *multicast* attacks do not show such local maximum of their efficiency. The *best route* attack has almost a constant efficiency per *Interest* regarding the attack rate, while the *multicast* attack is even more efficient per *Interest* when the attack is more aggressive. Overall, we can conclude that the

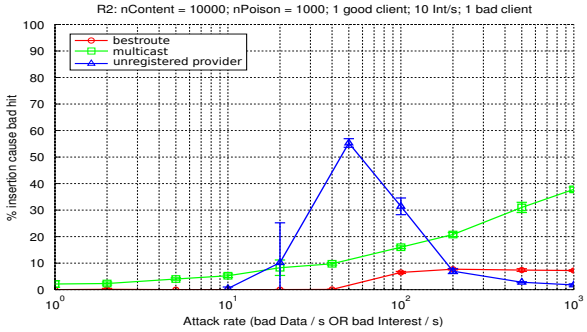


Figure 7. Attack rate effect on router 2

unregistered remote provider attack is the most efficient per packet sent to pollute routers and should quickly get fixed by the NDN community.

E. Attack footprint for the different scenarios

Finally, as a follow-up in the analysis of multiple-variable data, we performed a Principal Component Analysis (PCA) on our overall dataset to reveal correlations of all metrics and parameters. The values of the first two components, that account for 80.5% of the total variance of data, are provided as rows of the Table II. This table shows that the first component, accounting for 56.5% of measurements variance, is featured by a high impact on bad hit ratio, resources wasted

Table II
VALUES OF THE TWO FIRSTS PRINCIPAL COMPONENTS WITH THE LABEL OF ASSOCIATED METRICS.

Provider's side	Core router's side				Access router's side.				Client's side	
# additional *Interests*	% good hit	% bad hit	% miss hit	Resources waste	% good hit	% bad hit	% miss hit	Resources waste	% bad *Data* received	# bad *Data*
-0.1618	-0.0778	0.3913	-0.3891	0.3036	-0.3554	0.2976	0.1227	0.4018	0.3243	0.2731
0.4252	0.3178	-0.144	0.1085	-0.1963	-0.24	-0.2521	0.5727	-0.0401	0.3626	0.2549

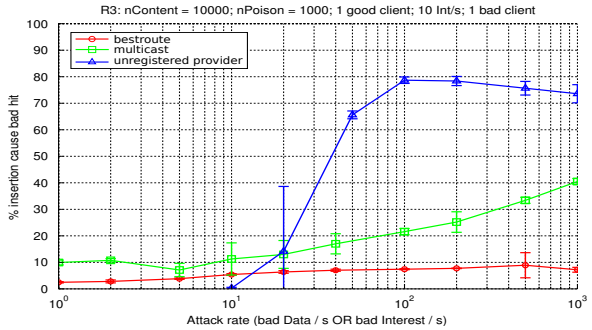


Figure 8. Attack rate effect on router 3

for bad hits of both routers, together with a high number and rate of bad *Data* to the good client. As such, this first component represents the main expected impact of the CPA with the injection of bad *Data* in routers' cache. Meanwhile, the second principal component, accounting for 24% of total data variance, shows a similar impact on the number of bad *Data* to the client, but a much higher impact on the miss ratio of the access router, a lesser extent on the core router and on the additional traffic to the provider. This exhibits the side effect of the CPA that prevents the routers from caching good *Data*, hence creating a higher rate of miss hit and traffic to the legitimate provider. Fig. 9 now presents the projection of individual measurements on these first two components as well as the mean projection for each scenario. The figure clearly shows that the components distinguish the *unregistered remote provider* scenario from the *multicast* and *best route* scenarios that exhibit the same operating mode. In the figure, the cyan circle points out the projections of the experiments with least attack impact (lowest attack rate). Similarly, the dashed arrow indicate the direction toward which the results move when the attack strength increases. The figure clearly shows that the *unregistered remote provider* scenario has a specific footprint mainly captured by the first principal component. As expected, for this case the bad *Data* creates a high rate of bad hit. It also shows that the *best route* and *multicast* scenarios have similar impacts when the attack rate increases, mostly featured by the second principal component. Indeed, those scenarios create a higher rate of miss hit as the legitimate clients try to avoid the bad *Data* from caches while, on the contrary, the bad client tries to prevent the caching of good *Data*. This also explains the higher number of requests forwarded to the provider. Finally, one can remark that the *best route* and *multicast* attack scenarios are also, to a lesser extent, characterized by the first principal component. However, the *best route* scenario exhibits a much smaller impact.

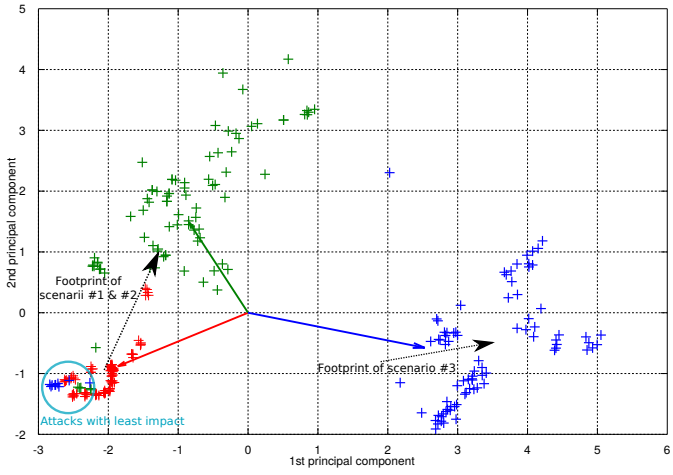


Figure 9. Projections of the measurements on the two first principal components. The solid line arrow represents the mean projection of each scenario and the '+' represent the projections of individual experiments. The black dashed arrows denote the direction when the attack rate increases.

V. CONCLUSION

By proposing to study the real behavior of NDN network entities under three attack scenarios we specified and implemented, we were able to highlight critical weaknesses in both the NDN protocol design and the NFD implementation which can be exploited to perform successful CPA. Moreover, through numerous experiment results, we have proved to what extent CPA is feasible in reality. On the one hand, the attack we discovered relying on the *unregistered remote provider* scenario has a low effect on the provider, but the highest effect on the good client as well as on the core and access routers, especially with a high attack rate. It might be the easiest one to implement, but also to fix with a dedicated patch in a future release of NFD. In the meantime, it constitutes a real threat for NDN. On the other hand, the *best route* and *multicast* scenarios are more difficult to perform but they are harder to circumvent since they rely on a standard use of the the NDN protocol. They exhibit a lower impact on the client's side and a higher one on the provider's side. The impact on the provider and on the access router are improved with higher attack rates but seems invariant on the client's side. Finally, whatever the attack rate, clients are well-protected from CPA in the *best route* scenario.

In future work, on the basis of this experimental assessment, we will propose a mathematical model for CPA. From such an attack model, we plan to design a detector based on *statistical hypothesis testing theory*, as well as a mitigation strategy that can handle large scale topologies. Contributing to the design and implementation of robust and safe solutions to attacks is essential for the future deployment of NDN.

ACKNOWLEDGMENTS

This work is partially co-funded by (1) the French National Research Agency (ANR), DOCTOR project, <ANR-14-CE28-0001>, started in 01/12/2014 and supported by the French Systematic cluster and (2) the CRCA and FEDER CyberSec Platform, <D201304601>.

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] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "A survey of information-centric networking research," *Communications Surveys & Tutorials, IEEE*, vol. 16, no. 2, pp. 1024–1049, 2014.

[3] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *Communications Magazine, IEEE*, vol. 50, no. 7, pp. 26–36, 2012.

[4] A. Chaabane, E. De Cristofaro, M. A. Kaafar, and E. Uzun, "Privacy in content-oriented networking: Threats and countermeasures," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 3, pp. 25–33, 2013.

[5] T. Lauinger, "Security & scalability of content-centric networking," Ph.D. dissertation, TU Darmstadt, 2010.

[6] S. Arianfar, T. Koponen, B. Raghavan, and S. Shenker, "On preserving privacy in information-centric networks," in *Proc of SIGCOMM Workshop on ICN*, 2011.

[7] G. Acs, M. Conti, P. Gasti, C. Ghali, and G. Tsudik, "Cache privacy in named-data networking," in *Distributed Computing Systems (ICDCS), 2013 IEEE 33rd International Conference on*. IEEE, 2013, pp. 41–51.

[8] N. Ntuli and S. Han, "Detecting router cache snooping in named data networking," in *ICT Convergence (ICTC), 2012 International Conference on*. IEEE, 2012, pp. 714–718.

[9] A. Mohaisen, X. Zhang, M. Schuchard, H. Xie, and Y. Kim, "Protecting access privacy of cached contents in information centric networks," in *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*. ACM, 2013, pp. 173–178.

[10] M. Xie, I. Widjaja, and H. Wang, "Enhancing cache robustness for content-centric networking," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 2426–2434.

[11] M. Conti, P. Gasti, and M. Teoli, "A lightweight mechanism for detection of cache pollution attacks in named data networking," *Computer Networks*, vol. 57, no. 16, pp. 3178–3191, 2013.

[12] I. Ribeiro, A. Rocha, C. Albuquerque, and F. Guimaraes, "On the possibility of mitigating content pollution in content-centric networking," in *Local Computer Networks (LCN), 2014 IEEE 39th Conference on*. IEEE, 2014, pp. 498–501.

[13] P. Gasti, G. Tsudik, E. Uzun, and L. Zhang, "Dos and ddos in named data networking," in *Computer Communications and Networks (ICCCN), 2013 22nd International Conference on*. IEEE, 2013, pp. 1–7.

[14] C. Ghali, G. Tsudik, and E. Uzun, "Needle in a haystack: Mitigating content poisoning in named-data networking," in *Proceedings of NDSS Workshop on Security of Emerging Networking Technologies (SENT)*, 2014.

[15] G. Bianchi, A. Detti, A. Caponi, and N. Blefari Melazzi, "Check before storing: What is the performance price of content integrity verification in lru caching?" *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 3, pp. 59–67, 2013.

[16] D. Kim, S. Nam, J. Bi, and I. Yeom, "Efficient content verification in named data networking," in *Proceedings of the 2nd International Conference on Information-Centric Networking*. ACM, 2015, pp. 109–116.

[17] C. Ghali, G. Tsudik, and E. Uzun, "Network-layer trust in named-data networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 12–19, 2014.

[18] A. Karami and M. Guerrero-Zapata, "An anfis-based cache replacement method for mitigating cache pollution attacks in named data networking," *Computer Networks*, vol. 80, pp. 51–65, 2015.

[19] S. DiBenedetto and C. Papadopoulos, "Mitigating poisoned content with forwarding strategy," in *The third Workshop on Name-Oriented Mobility (NOM)*. IEEE, 2016.

[20] A. Hoque, S. O. Amin, A. Alyyan, B. Zhang, L. Zhang, and L. Wang, "Nlsr: named-data link state routing protocol," in *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking*. ACM, 2013, pp. 15–20.

[21] X. Marchal, T. Cholez, and O. Festor, "Pit matching from unregistered remote faces: A critical NDN vulnerability," in *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, ser. ACM-ICN '16. ACM, 2016, pp. 211–212. [Online]. Available: http://doi.acm.org/10.1145/2984356.2985224

[22] A. Afanasyev, J. Shi, B. Zhang, L. Zhang, I. Moiseenko, Y. Yu, W. Shang, Y. Huang, J. P. Abraham, S. DiBenedetto *et al.*, "Nfd developer's guide," Technical Report NDN-0021, NDN, Tech. Rep., 2014.

[23] D. Rossi and G. Rossini, "Caching performance of content centric networks under multi-path routing (and more)," *Relatório técnico, Telecom ParisTech*, 2011.