On the Readiness of NDN for a Secure Deployment: the case of Pending Interest Table

Hoang Long Mai[1], Ngoc Tan Nguyen[1], Guillaume Doyen[1], Alain Ploix[1] and Remi Cogranne[2]

[1]HETIC/ERA – [2]ROSAS/LM2S
Institut Charles Delaunay – UMR CNRS 6281
Troyes University of Technology, 10004 Troyes Cedex - France

Abstract. Named Data Networking (NDN) is one the proposals for the Future Internet design relying on the Information Centric Networking paradigm and probably the most promising. To enable a large-scale deployment by Internet Service Providers, however, a well-established security is fundamental. While numerous prior works study the security of NDN, a large amount of those works have been conducted using simulation frameworks which prevent the consideration of potential threats and flaws in a real deployment context. Toward this effort, this paper studies the practical vulnerabilities exposed by NFD, the current implementation of NDN, and especially its Pending Interest Table. An attack scenario, based on the Interest Flooding Attack, is implemented on NFD routers deployed in a Network Function Virtualization environment. We show that the current implementation, though designed to be flexible, has some flaws that can ease the mounting of attacks in a real NDN network. Several recommendations are proposed for the security of future implementations.

1 Introduction

The foundations of current Internet architecture have not been deeply changed since the first proposal back in the 70s. In contrast, the use of the Internet has dramatically changed over the last decades with, to cite a few aspects, a tremendous growth of traffic, connected devices and needs for mobility and security. Such a profound change in the usage of the Internet challenges the current IP network. Therefore, there are currently many efforts to design the Future Internet and the Information Centric Networking (ICN) paradigm is a clean-slate approach that essentially proposes to move from the current host-based IP network to a content-based network.

To date, Named Data Networking (NDN) [1] is one of the most accomplished and studied ICN proposals and, hence one of the first candidates for a deployment in an operated context by Internet Service Providers (ISP). This evolution of the paradigm can be assessed by different recent efforts which all focus on providing solutions for a reliable deployment. Among the academic contributions, the NDN Testbed [1] and the CONET solution [2] deployed over the OFELIA testbed provide tools and an experimental feedback on early deployments, Ren et al. [3] address the hosting features which could facilitate ICN deployments by, for instance, leveraging Network Function Virtualization (NFV), while A. Afanasyev [4] proposes NDNS, a resolution framework which solves operational issues such as key registrations and retrievals. Jointly, from the standardization perspective, some Internet-Drafts are under publication by the Internet Engineering Task Force (IETF) such as [5, 6] to respectively identify ICN management considerations or propose NDN Message Format.

Moving from a lab restricted solution to a fully deployable solution, the NDN security must also be assessed and bound with operational constraints. If a first step of early security flaws identification, detection and mitigation has already been achieved in prior works (e.g. denial of service, cache pollution and poisoning, route poisoning, ...), their assessments only rely on simulated environments which provide a partial NDN behavior. As such, they are insufficient to address all threats which can occur in real implementations and a real context.

Taking part in this effort, this paper proposes to deploy a real NDN testbed based on the last version of the Network Forwarding Daemon (NFD) in a NFV context. We implement an Interest Flooding Attack (IFA) in order to observe the behavior of the NFD process and especially the Pending Interest table (PIT). The main goals of such experiments are (1) to exhibit unexpected behaviors when the network is stressed by the IFA, (2) to compare the empirical limits of the current implementation with the theoretical and simulated ones, and (3) to eventually provide a set of recommendations regarding the NFD security.

The paper is organized as follows. Section 2 presents in more details the related works focusing on NDN security issues and especially on the IFA. Then, Section 3 details the experimental setup that is implemented in the current study. Section 4 presents the numerical results and shows how an IFA can actually be implemented with success thanks to vulnerabilities of NFD. Finally, Section 5 concludes the paper and proposes some recommendations for the future development of NFD.

2 Related work

In this section we briefly introduce the NDN data-plane architecture and router components. Then, we review the set of literature which focuses on potential attacks related to the router overloading and especially the Pending Interest Table. Finally, we introduce the set of design and architectural solutions which have been proposed previously to protect NDN from these attacks.

2.1 Named Data Networking

Among all the proposals which aim at bringing a novel Internet protocol design, NDN [1] is considered as one of the most promising Future Internet solution. The novelty of NDN relies in the key concept of naming content objects instead of naming hosts with IP addresses. NDN uses a hierarchical naming scheme for content objects, like a Uniform Resource Identifier (URI). Communication in NDN is achieved by two types of packets: (1) *Interest* and (2) *Data*. A user issues his/her demand for some content by sending an *Interest* packet. In return, a *Data* packet containing the requested content is sent back to the user. In NDN, a router exhibits many faces which stand for a generalization of interfaces in IP networks and it owns three main components which are combined to build the forwarding process. Firstly, the *Content Store* (CS) is a local cache that improves content delivery by storing recently requested content. Secondly, the *Forwarding Information Base* (FIB) contains routing information for *Interest* packets. Finally, the *Pending Interest Table* (PIT) contains routing information for *Data* packets. More precisely, for each forwarded *Interest*, its incoming faces are saved in a PIT entry, so that the corresponding *Data* can be sent back to the user. For each received *Data*, the corresponding PIT entry is then removed. Consequently, NDN defines a full-state data-plane which, although enabling an efficient routing of *Interest* and *Data* packets, also brings novel threats related to the state maintenance in each routers. Especially, the PIT overload phenomenon, which can occur due to a malicious traffic activity or a legitimate network overload, has been extensively studied, such as in [7] which, with the help of a custom simulator, provides guidelines for the design and implementation of this component.

2.2 Detection and mitigation of PIT overload attacks

A large set of studies has proposed various solutions against the IFA [8], a variation of the Denial of Service (DoS) attack in NDN, which consists in overloading the PIT by sending a large amount of malicious *Interest* packets for non-existent content. Such *Interests* cannot be resolved by any *Data* packet. As such, the corresponding PIT entry cannot be removed by a *Data* packet, but only by the entry lifetime expiration. When the PIT is overloaded, new *Interest* packets cannot be handled and, thus, are dropped. This attack induces serious consequences on network for two reasons. Firstly, it can cause large scale damage by targeting the network infrastructure. Secondly, *Interest*s for non-existing content can be easily generated by any user.

In [9], Dai et al. present their *Interest* trace back mitigation strategy. Whenever the PIT's size exceeds a threshold, a spoofed *Data* packet is created by the router to resolve a long-unsatisfied *Interest*. These *Data* are eventually forwarded back to the source of attack by tracing PIT entries. At the same time, routers also limit the incoming packet rate of faces to which they send fake *Data*. Profiting from statistics to identify harmful faces, the Poseidon approach, proposed by Compagno et al. in [10], maintains two measures: (1) the satisfaction ratio and (2) the PIT space used by *Interests* from the concerned face. Once an alarm occurs, a router issues an alert message to its neighbor on the malicious face. When the latter receives an alert, it also triggers the same counter-measure, but with a lower threshold, in order to better identify the compromised face. In [11], Afanasyev et al. proposed the satisfaction-based push-back counter-measure. The idea of this proposal is similar to the Poseidon proposal: routers exchange announcements with neighbors and adjust their reactions based on these messages. Although this solution monitors the satisfaction ratio, it does not have a separate detection phase. The ratio is actually used to periodically calculate the *Interest* limit exchanged in announcements between routers. Leveraging the statistical hypothesis testing theory, in our previous work [12, 13], we have also proposed a low-computation-cost detector against IFA which enables the theoretical performance assessment of the detector: assessment of the confidence given in results, setup of the detection threshold at the design stage and independence from the attack behavior.

If all these solutions provide different strategies for the detection and mitigation of IFA, they are all based on a theoretical behavior of routers and an evaluation performed in a simulation environment (e.g. ndnSim). As such, they only consider a restricted behavior of NDN components and attack patterns while omitting practical solutions that could mitigate this attack as well as the set of related threats which occur in case of a real deployment and lead to a different behavior of a NDN topology.

2.3 Design and implementation considerations

In this effort, the authors of [14] introduce the *Interest NACK* packet to extend the forwarding mechanism of NDN. When a NDN router can neither satisfy nor forward an *Interest*, an *Interest NACK* is generated and sent to the downstream router. In other words, an *Interest NACK* is a packet which carries an error code to notify and prevent the downstream router from sending further *Interest*s with the same content name. There are currently three types of error codes defined as follows:

- *Duplicate NACK*: the router is still waiting for *Data* for an identical *Interest* packet;
- *Congestion NACK*: the *Interest* packet cannot be forwarded due to a congestion occurring on the outgoing link;
- *No data NACK*: the router cannot receive any *Data* to satisfy the *Interest* packet for some reason (e.g. no path available in FIB or PIT entry times out).

As such, *Interest NACK* has two benefits on a NDN topology: (1) the routers release the PIT entries much faster than waiting for the lifetime expiration, thus bringing a natural mitigation of the IFA and (2) it also helps the downstream routers to determine the cause of *NACK* in order to decide the further forwarding strategies.

To conclude this literature review, we state that in order to make NDN a secure and performant data-plane solution deployable by ISP, the efforts must now target the implementation of components which reveal novel weaknesses that cannot have been handled at the design stage nor through simulation environments. In this effort we propose in this paper to feature the PIT overload phenomenon from a practical perspective to assess its feasibility, understand its consequences on operated routers, identify potentially unrevealed aspects of the phenomenon and eventually provide a set of guidelines for the implementation roadmap.

3 Experimental framework

In order to evaluate the readiness of NDN for a realistic deployment use-case, we leveraged system and networking virtualization, thus fitting with a NFV [15] scenario. Such a deployment hypothesis is currently considered as an opportunity to accelerate and facilitate the deployment of novel networking functions, or even full data-planes, while preserving legacy ones without increasing Capital Expenditure (CAPEX) and it is the most credible hypothesis for a NDN deployment [3]. In this context, we present in this section different scenarios we have considered to implement the IFA on a real NDN infrastructure, the subsequent test architecture we have deployed and the set of tools we have implemented.

3.1 Scenarios

We consider a set of attack scenarios which goes beyond the basic generation of a large amout of *Interest* packets in a short time to overload the PIT as described in current literature. By contrast, this set considers realistic flaws in NDN and brings IFA from a pure simulated attack to reality.

Congestion on the link between provider and router This first scenario deals with the straightest way to implement an IFA. For the attacker, it consists in sending a large amount of *Interests* in a short time to, rather than try to fill the PIT of upstream routers, make the link between routers and a provider congested. When the link between the provider and the last upstream router is congested, the provider cannot send *NACK* packets to notify the router anymore. Therefore, at the time of congestion, the router is under attack without the presence of *NACK*. One drawback of this scenario resides in the congestion point which can happen on any link separating the attacker from the provider (e.g. on any intermediate router), thus making it strongly dependent from the capacity of each link in the end-to-end path. Even more, if the *Congestion NACK* mechanism has still not been implemented to date, its acknowledgement by the NDN community for an integration in future implementations make this scenario almost obsolete and as such, we do not consider it in the following of our study.

Exploit the *No Data NACK* to accumulate PIT entries This scenario exploits the vulnerability design of the *No Data NACK*, which allows the PIT to keep an *Interest* until its lifetime expires even if it received an *NACK Interest* [14]. This scenario is simple to deploy. For example, we consider an intermediate router $R1$ owning an entry in FIB to forward all *Interests* of a given prefix to a router $R2$. But in turn, this router does not own a route in FIB to forward the *Interest*. It will thus generate a *No Data NACK* packet to and send it to the downstream router $R1$. However, when $R1$ receives the *No Data NACK* packet, it will not remove the PIT entry of the dedicated *Interest* because its face bound to $R2$ is still available. The PIT entry will be removed only when the router has no available face in FIB to send the *Interest*. This design leads to a potential vulnerability in the accumulation of PIT entries which can be exploited to perform an IFA.

Stretch the data providing delay by a malicious provider As mentioned above, the *NACK* mechanism makes the forwarding decisions on downstream routers totally dependent from the upstream router. Therefore, when a top upstream node (e.g. a data provider) does not send the *NACK* downstream, the whole topology cannot detect the attack. To illustrate this case, we propose in this third IFA scenario to create a malicious data provider exhibiting an abnormal long response delay to *Interests* which however is lower than the lifetime of *Interests*. As a consequence, the downstream router will not receive any *NACK* packet while its PIT will accumulate entries. This scenario, while requiring the cooperation of both the consumer and provider sides, remains easy to implement especially given the opportunity of the current Internet to let end-users operate their own virtualized content servers.

3.2 Testbed architecture

Figure 1 illustrates the overall architecture of the experimental framework we have deployed for our study. In order to emulate the presence of several physical servers hosting NDN network functions, we have deployed it in OpenStack[1]. The latter provides an Infrastructure as a Service (IaaS) enabling the testbed[2] scalability for future larger experiments. Each virtual machine, emulating an ISP server, follows the *large* template configuration of OpenStack and hosts Linux Ubuntu as an operating system.

On that basis we have implemented a set of tools which enables the implementation of NDN in an NFV context. As such, each emulated server hosts Docker[3] as a container-based virtualization framework of network functions and OpenVSwitch[4] as the infrastructure-layer networking component. In order to enable the communication between containers in a realistic NFV use-case, we have leveraged VXLan as a transport framework for all data from one container to another. The network function we consider here is NFD (NDN Forwarding Daemon), configured as an overlay which encapsulates all *Interest* and *Data*

[1] "`https://www.openstack.org/`

[2] One can note that OpenStack is used as a seamless system and networking virtualization framework which does not take part of the subsequent architecture

[3] `https://www.docker.com/`

[4] `http://openvswitch.org/`

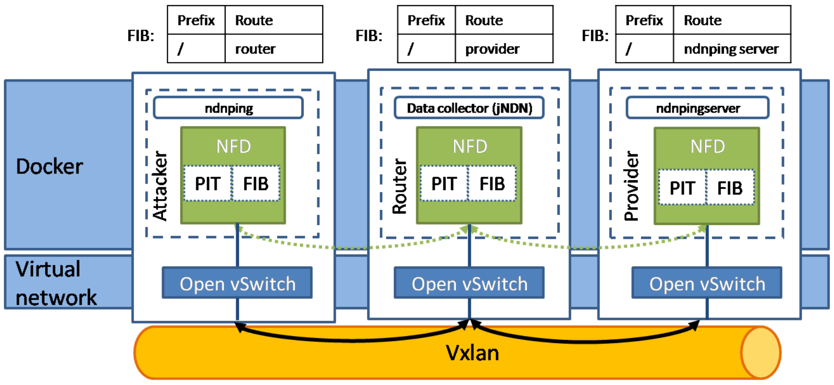


Fig. 1: Architecture of our experimental framework

packets in IP/UDP channels. In order to collect all data related to our experiments, we have developed a dedicated module based on the jNDN[5] client which collects every second all status information (e.g. number entries in the PIT, total number of In and Out packets) as well as all face statistics (e.g. number of In *Interest* packets, number of Out *Data* packets) of the instrumented router through the NFD Management Protocol[6]. All this information is registered in a JSON format for further processing. Finally, as a traffic producer and consumer, we use *ndnping*, a tool originally designed to test the connection between two NDN routers. We have modified its source code to generate dedicated packets as an attacker with a similar intention could easily do. Especially, we generate *Interests* with the following prefix: */test/ping/123456789*, where *test* is a static prefix configured to identify an experiment, *ping* identifies the packets belonging to *ndnping* and finally *123456789* is a unique identification number of *Interest*, randomly generated at start of the tool and then increased by one for each packet generation. This value is also used in the *Nonce* field of NDN packets to ensure that there are no duplicate *Interests*.

The set of tools we have considered is summarized in Table 1. One can note that we used the latest NFD version. The latter allows NFD to support the *Interest NACK* packet and creates dedicated incoming and outgoing *NACK* pipelines. It especially implements *Duplicate NACK* in *Interest* a loop pipeline and *No Data NACK* in the forwarding strategy. However at the date of the experiments, the *Congestion NACK* was still not supported.

4 Results

We have implemented the second scenario presented above to reproduce the IFA because it stands for the straightest case for an attacker while actually exploit-

[5] `https://github.com/named-data/jndn`
[6] `http://redmine.named-data.net/projects/nfd/wiki/Management/`

Component	Software	Version
Forwarding engine	NFD	0.4.0
Content provider and attacker	ndnping	N.A.
Measurement client	jNDN	0.9
Virtualization layer	Docker	1.9.1
Network virtualization	Open vSwitch	2.0.2
Operating systems	Ubuntu	14.04 LTS

Table 1: Experimental tools implemented for our study

ing vulnerabilities in the current NDN design and its NFD implementation. In order to capture its operating mode, we have first identified the set of factors which impact the attack success. In a second time, we have quantified them with several iterations, all varying the factor values. In order to get reliable results, for each value of a factor, the experiments were performed five times and the following results provide the average of each measured factor bounded with a 95% confidence interval. The set of factors we consider in our experiments are:

- The memory size allocated to the NFD process;
- The attack rate given by the number of *Interests* per second generated by the attacker;
- The lifetime of *Interests*;
- The prefix size given by its number of characters;
- The number of levels which compose the prefix.

4.1 The PIT overload phenomenon

The first result we observed as an immediate consequence of a PIT overload is the system crash of the NFD process. This result is unexpected since all previous works, based on simulations, feature the PIT overload process by a packet drop which does not impact the forwarding process proper execution. To illustrate this phenomenon, we provide in Figure 2 an extract of the NFD log related to a PIT entry insertion. The log clearly shows that, when overloaded, the NFD process simply stops. We conclude that currently, no protection scheme prevents NFD from a memory overload, whose consequences are highly damageable in case of a real deployment by an ISP. Consequently, in all subsequent experiments, we consider the router crash, denoted as the collapse point, as the main output of our experiments and we address the different factors which lead to such a phenomenon.

4.2 Factors impacting the PIT overload

Memory allocation As mentioned in [7], the memory capacity allocated to the NFD process has an important impact on the PIT capacity. A router with a larger memory has undoubtedly a larger capacity of PIT entries. As such, in order

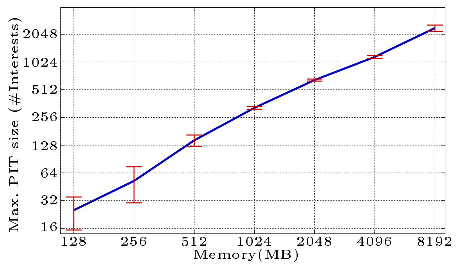
```
1454422643.526260 TRACE: [LinkService] [id=264,local=tcp4://173.16.1.33:6363,remote=tcp4
://173.16.1.11:40521] receiveInterest
1454422643.551709 DEBUG: [Forwarder] onIncomingInterest face=264 interest=/ping
/1971374167
1454422643.581967 TRACE: [NameTree] lookup /ping/1971374167
1454422643.602334 TRACE: [NameTree] insert /
1454422643.623407 TRACE: [NameTree] Name / hash value = 0 location = 0
1454422643.647951 TRACE: [NameTree] insert /ping
1454422643.676848 TRACE: [NameTree] Name /ping hash value = 465469377468200915 location =
 30675
1454422643.699100 TRACE: [NameTree] insert /ping/1971374167
1454422643.726028 TRACE: [NameTree] Name /ping/1971374167 hash value =
12459082988191734691 location = 56227
1454422643.746207 TRACE: [NameTree] Did not find /ping/1971374167, need to insert it to
the table
```

Fig. 2: Log trace of NFD for a PIT entry insertion when the router crashes

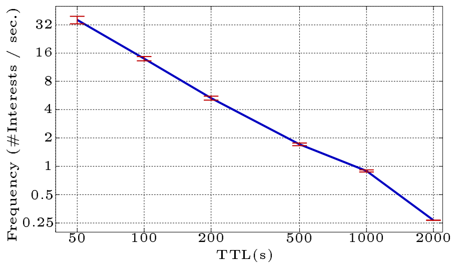
to clearly evaluate the impact of the allocated memory on the collapse point, we have configured Docker with different amounts of memory for the NFD container. In order to exacerbate the overload phenomenon, we have considered extra-long prefixes containing 256 levels and 522 characters in total. The result, depicted in Figure 4.a, although predictable, indicates that the the PIT collapse point is proportional to the amount of allocated memory. However, the internal structure of the PIT, called the *nameTree*, is actually implemented in a structure shared with other NFD components relying on both a tree and a hashtable for fast-lookup[7]. This data structure leads to the average use of 0,27 entries per MByte of allocated memory to the NFD process, which also indicates that extra-long prefixes are very costly to store.

Attack pattern Secondly, we have studied the impact of lifetime and frequency of *Interest* packets on the PIT collapse point which together form an attack pattern. The lifetime of *Interests* is an important aspect, because the longer *Interests* stay in the PIT, the faster the PIT collapse point is reached. Furthermore, the default lifetime of *Interest* in NDN is 4 seconds, but content consumers can arbitrarily fix the lifetime of their *Interests*. As a consequence, a malicious user can intentionally flood NFD with large *Interest* lifetime values in order to multiply the impact of the IFA and currently there is no protection in NFD which could prevent this phenomenon. In order to understand the relation which binds the lifetime with the packet frequency to reach the PIT collapse point, we have modified the source code of *ndnping* to generate *Interests* with a prefix containing 32 levels and 72 characters and allocated 128MB of memory to the Docker container hosting NFD. The results of these experiments are depicted in Figure 3.b. The smallest lifetime we have measured to successfully overload the PIT is 50 seconds. However, our scenario only considers a basic implementation of IFA with a single attacker. We can imagine that in the reality of such an attack, a system of botnets would generate a larger frequency of *Interests* without any congestion issue. Therefore, the 50-second value is not the definitive smallest lifetime to

[7] See section 3.6 of the NFD developer guide, available at `http://named-data.net/wp-content/uploads/2014/07/NFD-developer-guide.pdf`



(a)



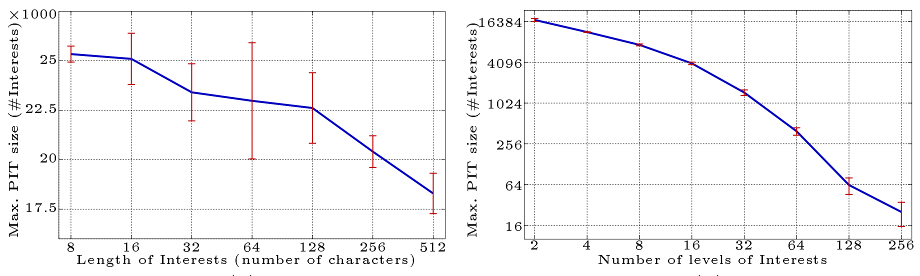
(b)

Fig. 3: Collapse point of the forwarding daemon according to **(a)** the NFD container memory (log-log scale) and **(b)** *Interest* frequency and lifetime (log-log scale)

perform an IFA and an attacker can perform the attack with a smaller *Interest* lifetime by leveraging more attack sources. Furthermore, these results also assess the potential vulnerability of NDN to the accumulation of PIT entries in case of *No Data NACK*. It especially shows that an attacker can perform flooding attacks with a small packet frequency by simply extending the *Interest* lifetime. A mechanism dedicated to the PIT cleaning before the *Interest* lifetime expires or the limit of the *Interest* lifetime is required to prevent this phenomenon.

Length of *Interests* We have then studied the impact of the size of *Interests* on the PIT collapse point since the implementation of the PIT in NFD is designed as a data-structure hosted directly in the NFD process memory. Hence, the more complexity of *Interests*, the more space in memory is needed to host them. As a first feature of *Interests*, we have considered the length of *Interests*, which is indicated in [7] as an important factor. To that aim, we have generated in this case different prefixes with a fixed number of two levels but with a variable length given by the number of characters which form the prefix. In a similar way with previous experiments, we have considered a container with 128MB to host NFD. The results, presented in Figure 4.a show that the length of *Interest* has an impact on the number of entries required to reach the PIT collapse point, but this impact is reasonable. Indeed, growing of an 8 factor the length of *Interests* only decreases the PIT collapse point of 28%.

Number of levels of *Interests* As a last impact factor on the PIT collapse point, we have considered the number of levels which form a prefix in an *Interest*. The naming convention of content in NDN follows a hierarchical scheme, which is similar to URI and at this time, there is no limit to the number of levels. In order to measure this impact, we have created prefixes with various levels, but with a constant length of 522 characters. The result shown in Figure 3.b reveals the important impact of this factor on the PIT collapse point which, to the best of our knowledge, has not been identified to date. Specifically, we observe that the number levels of *Interests*, growing from 2 to 256, drastically reduces the PIT collapse point of almost three orders of magnitude, thus providing an easy flaw to exploit for any attacker to successfully perform an IFA with small means.



(a) (b)

Fig. 4: Collapse point of the forwarding daemon according to **(a)** the packet size of *Interest*s (semi-log scale) and **(b)** the number levels of *Interest*s (log-log scale)

5 Conclusion and future work

In this paper, we have studied the *Interest* Flooding Attack from a practical perspective, by implementing it on the last version of the NDN Forwarding Daemon deployed in a NFV context. The goal of this study relies in the security assessment of NDN since it is now considered as a promising solution for a real deployment by ISP. We have especially implemented a scenario which, despite the integration of *NACK Interests* in the NDN protocol specification demonstrates the feasibility of this attack. From this scenario, we have first shown that the practical consequence of a PIT overload is the crash of the forwarding daemon which is highly damageable in an operated context. Then, we have identified the set of factors which impact the attack success. These are: the *Interest* packet frequency, the lifetime of *Interest*, the size of prefixes in *Interests* and especially the number of levels it counts.

From this results, we conclude that basic security enforcement mechanisms are missing and have to be integrated in NFD to enable this implementation to be actually deployed in any production environment. Especially, as recommendations, we state that some basic upper limits must be implemented in NFD on especially (1) the lifetime of *Interest* which is currently freely fixed by the user, (2) the number of levels in the prefix of *Interests* which leads to an exponential memory exhaustion and (3) the amount of memory allocated to the sole PIT data structure in order to prevent the daemon crash but rather drop exceeding *Interests*.

As a future work, we plan to extend this study to other NDN protocol fields to evaluate their potential vulnerabilities. We also plan to extend our testbed to implement more advanced attack scenarios such as distributed ones. Finally, from a long term perspective, we plan to couple our testbed with an existing content delivery service to estimate to what extend existing services carried by a NDN protocol stack could suffer from this kind of attack.

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 et al. "Named Data Networking". In: *ACM SIGCOMM Computer Communication Review* 44.3 (2014), pp. 66–73.

[2] S. Salsano et al. "Information centric networking over SDN and OpenFlow: Architectural aspects and experiments on the OFELIA testbed". In: *Computer Networks* 57.16 (2013), pp. 3207–3221.

[3] J. Ren et al. "On the deployment of information-centric network: Programmability and virtualization". In: *Computing, Networking and Communications (ICNC), 2015 International Conference on*. IEEE. 2015, pp. 690–694.

[4] A. Afanasyev. "Addressing operational challenges in Named Data Networking through NDNS distributed database". PhD thesis. UCLA, 2013.

[5] Ivan Vidal et al. *ICN Management Considerations*. Tech. rep. Version 6th. IETF, Internet-draft, 2014.

[6] M. Stapp. *NDN Message Format Proposal*. Tech. rep. Version 1st. IETF, Internet-draft, 2015.

[7] M. Virgilio, G. Marchetto, and R. Sisto. "PIT overload analysis in content centric networks". In: *Proc. of 3rd ACM SIGCOMM workshop on Information-centric networking*. ACM. 2013, pp. 67–72.

[8] P. Gasti et al. "DoS and DDoS in Named Data Networking". In: *Computer Communications and Networks (ICCCN), Intl' Conference on*. IEEE. 2013, pp. 1–7.

[9] H. Dai et al. "Mitigate DDoD attacks in NDN by Interest traceback". In: *Proc. of IEEE INFOCOM NOMEN Workshop*. 2013.

[10] A. Compagno et al. "Poseidon: Mitigating Interest flooding DDoS attacks in Named Data Networking". In: *Local Computer Networks (LCN), Intl' Conference on*. IEEE. 2013, pp. 630–638.

[11] A. Afanasyev et al. "Interest flooding attack and countermeasures in Named Data Networking". In: *IFIP Networking Conference*. IEEE. 2013, pp. 1–9.

[12] Nguyen T., R. Cogranne, and G. Doyen. "An optimal statistical test for robust detection against Interest flooding attacks in CCN". In: *Integrated Network Management (IM), IFIP/IEEE Intl' Symposium on*. 2015, pp. 252–260.

[13] T.N. Nguyen et al. "Detection of Interest flooding attacks in Named Data Networking using hypothesis testing". In: *Information Forensics and Security (WIFS), IEEE Intl' Workshop on*. 2015, pp. 1–6.

[14] Cheng Yi et al. "A case for stateful forwarding plane". In: *Computer Communications* 36.7 (2013), pp. 779–791.

[15] R. Jain and S. Paul. "Network virtualization and software defined networking for cloud computing: a survey". In: *Communications Magazine* 51.11 (2013), pp. 24–31.