PIT matching from unregistered remote Faces: a critical NDN vulnerability

Xavier Marchal, Thibault Cholez, Olivier Festor

PIT matching from unregistered remote Faces: a critical NDN vulnerability

Xavier MARCHAL, Thibault CHOLEZ, Olivier FESTOR
LORIA, UMR 7503 (University of Lorraine, CNRS, INRIA)
Vandoeuvre-les-Nancy, F-54506, France
{xavier.marchal, thibault.cholez, olivier.festor}@loria.fr

ABSTRACT

In this work, we describe two attack scenarios exploiting a NDN vulnerability based on the fact that malicious nodes can send unexpected Data that can consume legitimate PIT entries, thus badly affecting NDN communications. We also propose two ways to prevent it. Both attacks and remediation strategies will be demonstrated at the conference.

1. INTRODUCTION

Named-Data Networking (NDN) is one of the most advanced ICN architecture and already features a set of running software with NDN Forwarding Demon (NFD) at the centre. Shortly, new testbed involving real users and applications will complement simulations of NDN. However, the security of NDN/NFD is still in the early stages and not ready to such exposition. In this demonstration, we will show a new vulnerability of NDN that is easy to exploit and can lead to very serious attacks, badly affecting the network.

This vulnerability is due to an absence of control at the precise moment when NFD receives an incoming Data. In fact, NFD only checks two points: if the Data belongs to the localhost scope, or if it matches an existing PIT entry, but not if the Data comes from a valid Face. This is a critical shortage because it allows malicious users to directly send Data to perform attacks like DoS and cache poisoning without having to register a prefix in the router's FIB beforehand to receive legitimate Interests. After these checks, NFD continues to process the Data packet by caching it and so on (Listing 1). It can be explained because the NDN protocol makes the hypothesis that a node cannot send a Data packet without having previously received the corresponding Interest (receiver driven communication). However, NFD should consider malicious nodes that decide to not follow the standard way to proceed with NDN communications and send Data unexpectedly.

In the rest of this short paper, we describe two attack scenarios exploiting this vulnerability and two possible solutions to mitigate it.

Listing 1: vulnerability location

```
void
Forwarder::onIncomingData(Face& inFace,
const Data& data) {
    {...} //some stuff and scope check
    //PIT match check
    pit::DataMatchResult pitMatches = m_pit.
        findAllDataMatches(data);
    if(pitMatches.begin()==pitMatches.end()){
            //goto Data unsolicited pipeline
            this->onDataUnsolicited(inFace,data
                );
            return;
    }
    // CS insert
    m_cs.insert(data);
    {...} //and so on
}
```

2. ATTACK SCENARIOS

The scenarios exploiting the vulnerability have low requirements that concern the network topology. In fact, the attacker only needs to be directly connected to a network node (i.e. executing NFD) on the path between the client and the provider, more likely on the same access router.

Our demonstrations are based on the simple topology illustrated in Figures 1 and 2, where we have one consumer C connected to a NDN node named R1. R1 has a route to another NDN node R2 which, itself, has a route to a producer S. In this way, C can send Interests which will be forwarded toward S. The attacker named A is connected like a consumer to a router (i.e. through a Face with no registered route for S content) on the path between C and S, on R1 in our example.

2.1 Timing Attack

In the first scenario (Figure 1), A, despite being a simple consumer, does not send Interest packets but Data packets and try to match the names of Interest packets sent by C. A is blind to the emitted Interests but can be helped with cache probing or can target a specific content. Then, a race condition begins between A and S at the router R1, and only the first packet to arrive will be accepted, all the others will be dropped as long as a new PIT entry is not recreated by C. In order to increase its success rate, A can flood the router with Data packets at a rate that depends on the network latency, because the Data packet must be received at the right time between the moment when the router forwards the Interest, and when the producer Data reaches back the router. To make things worse, we also noticed that NFD

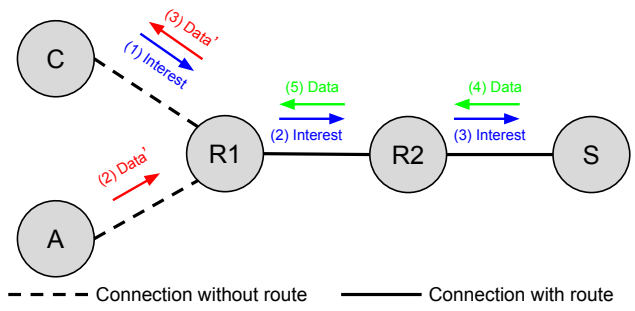

Figure 1: PIT DoS by Timing attack

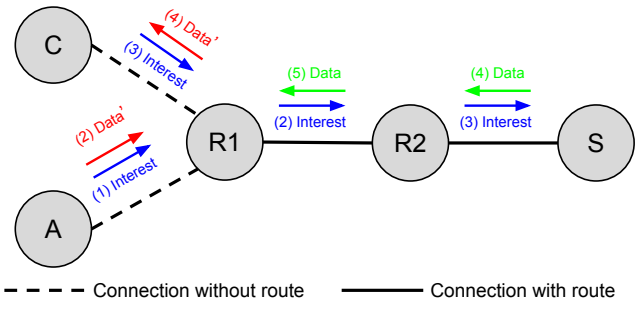

Figure 2: Cache poisoning by self-answering

keeps a PIT entry alive for a few milliseconds each time it is matched. Consequently, if the attacker sends Data packets for a specific PIT entry fast enough, he can keep this entry alive in the PIT forever and do multiple cache insertions of bad Data thanks to this entry.

2.2 Self-answering Attack

In the second scenario (Figure 2), A sends an Interest packet to the router immediately followed by a Data packet that will match the preceding Interest. This behaviour is a very strong vector of cache poisoning because it makes it possible to insert Data packets in cache without the help of any other actor and without any chance to fail. Furthermore, due to a short delay preceding the deletion of PIT entries, the attacker is not limited to prefixes registered in NFD's FIB but can insert Data with any name. By constantly adding Data in the router's cache, an attacker can completely block the access to a specific resource at the level of the attacked router. Indeed, users' Interests will always be answered by malicious Data packets placed in cache. Moreover, with the perpetual renewal of these malicious Data packets, the exclude field used in Interests to evade poisoning becomes inefficient.

Figure 3 shows the effect of these two scenarios on the proportion of legitimate contents received by a client exposed to different attack rates. We set the client Interest rate to 4 Interest/s, the freshness of the illegitimate Data to 0 ms for Timing attack and 2s for self-answering. We used 100 contents following a Zipf distribution for their popularity. The network latency is 100 ms between R1 and S. We can clearly see that the performance of the network decreases under attack.

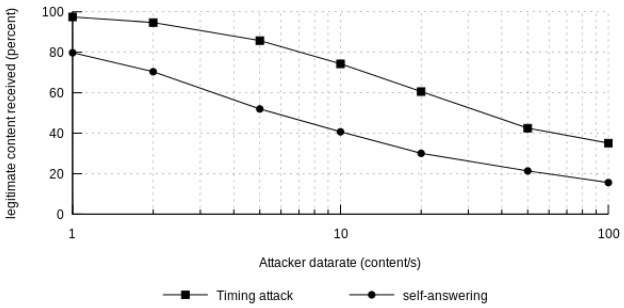

Figure 3: Effect of timing and self-answering attacks on Data received by the client

3. REMEDIATION STRATEGIES

3.1 Comparing the Data incoming Face with Interest outgoing Face(s)

To prevent attackers to send Data from unestablished routes, NDN nodes must be able to remember each Face to which is forwarded an Interest packet. Then, only Data packets coming from the same Face(s) should be allowed to consume the PIT entry. Any Data packet coming from another Face is unexpected and should be considered malicious. NFD already keeps track of Interests' outgoing Faces, but only to process Interest NAcks. The simplest way to fix this vulnerability is thus to extend the additional checks performed for Interest NAcks to Data packets. This verification can only occur at the PIT level because Out-records are stored in PIT entries.

3.2 Comparing the Data incoming Face with FIB entries

The other solution consists in comparing the incoming Face of the Data with FIB entries. In fact, only Data packets coming from a Face of the NDN node with a valid route to the name carried by the Data should be accepted. This strategy is more suited for multicast forwarding than the previous one since remembering the outgoing Faces does not provide any added value to the information already available in the FIB. Moreover, not relying on additional information makes this solution stateless. This method also has the advantage to be applicable prior to any PIT lookup.

4. CONCLUSION

We showed that an attacker can easily disrupt NDN communications by sending unexpected Data despite the absence of legitimate route for the attacked contents, and we described two attacks: the timing attack and the self-answering attack exploiting this vulnerability. While critical, this issue could be easily corrected and we proposed two possible ways to do it by performing additional verifications on the incoming Data based on information available either at the PIT or at the FIB level.

Acknowledgement
This work is partially funded by the French National Research Agency (ANR), DOCTOR project, under grant <ANR-14-CE28-0001>.