UCLA CS217A 18W Project Report

NDN DoS Resilience

Vishrant Vasavada
UCLA
vasavada@cs.ucla.edu

Siva Kesava Reddy K
UCLA
sivakesava@cs.ucla.edu

ABSTRACT

Named Data Network (NDN) is a newly proposed Internet Architecture, also nicknamed as *the future Internet Architecture*. Instead of using IP addresses to push the data to specific locations, it retrieves data by name. Security and privacy should be treated as fundamental requirements for this newly evolved architecture to avoid past pitfalls with TCP/IP architecture. It is thus necessary to check resilience of NDN towards Denial of Service (DoS) attacks that plague today's Internet.

In this project, we performed basic analysis of how NDN's design could provide in-built security and detection mechanism towards DoS attacks. Through our results, we show that NDN Router's Content Store and Interest Aggregation Technique can provide some resilience towards DoS traffic in NDN while Interest Satisfaction Ratio and Traffic Composition Knowledge can help detect ongoing DoS attack.

1 INTRODUCTION

TCP/IP architecture is the largest deployed technology today. It has successfully achieved the goal to carry data from one point to another. However, it isn't bulletproof. When the TCP/IP protocols were designed, engineers didn't really think about security. Over the years, this world has seen increasing number of bad guys on the Internet exploiting its poor design. Denial of Service (DoS) attacks are the biggest threat to the Internet today. It is a rapidly growing problem and new types of DoS attack keep showing up every few months [5][6]

Security researchers have proposed many solutions to defend against such attacks. However, they address some problems but not all. The problem, in fact, is design of the basic network communication model which invites such attacks. Anyone can send packets to any IP address and port. The source IP address can also be spoofed easily. These characteristics allow attackers to carry out DoS at minimum cost.

Named Data Networking (NDN)[2] is an ongoing research project aimed towards designing the future Internet architecture using Information Centric (ICN) approach to networking. In contrast to TCP/IP architecture, NDN's key goal is "security by design". In this project, we tried to answer the following two questions through our simulations:

- Can NDN's design stop DoS attack from the very beginning?
- How does NDN's design help in fast detection of DoS attack?

Data (content) is the first-class citizen in NDN unlike IP addresses in TCP/IP. It adheres to *pull-model* where it fetches data from the network by sending out *interest packets* rather than traditional TCP/IP *push-model* where data is pushed into the network based on the destination IP address. This already eliminates attacks such as address spoofing. However, attackers can still overwhelm routers and producer of content by flooding them with interest packets.

In this project, we simulated such flooding attacks using *ndnSIM*[4], an open-source NS3 based simulator designed to examine and evaluate various components of NDN architecture. As per our knowledge, our effort is the first step towards DoS Simulation in NDN that will allow assessment and possible mitigation of DoS attacks in NDN. The rest of the report is organized as follows: We provide overview of NDN architecture and ndnSIM in 2. We also discuss our additions into ndnSIM in the same section. We describe our topology in 3 and discuss different scenarios in 4. We finally summarize related work in 5 and discuss future work and conclude in 6.

2 BACKGROUND

In this section, we describe NDN Architecture and ndnSIM. We also describe the additions we had to make to ndnSIM to achieve our goals.

2.1 NDN

Named Data Networking (NDN) is ongoing research project towards designing the future Internet architecture using Information Centric (ICN) approach to networking. NDN adheres to using *pull-model* where it fetches the data from the network using *interest packets* rather than traditional TCP/IP *push-model* where data is pushed into the network based on the destination IP address. If there is some entity that can serve a given interest, it returns the corresponding *content packet*. NDN Router, known as, *NDN Forwarding Daemon* consists of three basic components:

- Content Store: Used for content caching and retrieving
- Forwarding Information Base (FIB): Used for forwarding packets. It contains a table of name prefixes and forwarding interfaces
- Pending Interest Table (PIT): Used for storing currently unsatisfied interests and incoming interfaces

When NFD gets an interest packet, it first checks whether the corresponding data is already in its content store. If it is, the data packet is sent back to the corresponding incoming interface. Otherwise, the interest name is checked against entries in the PIT. If there is another pending interest with the same name, it just adds the incoming interest of this interest to the existing PIT entry. If the name doesn't exist in the PIT, the interest is added to it and is forwarded based on entries in FIB and chosen strategy for forwarding.

When NFD gets a data packet, its name is used to lookup the PIT for corresponding interest packet entry. Once the matching PIT entry is found, NFD sends the data to all the interfaces from which

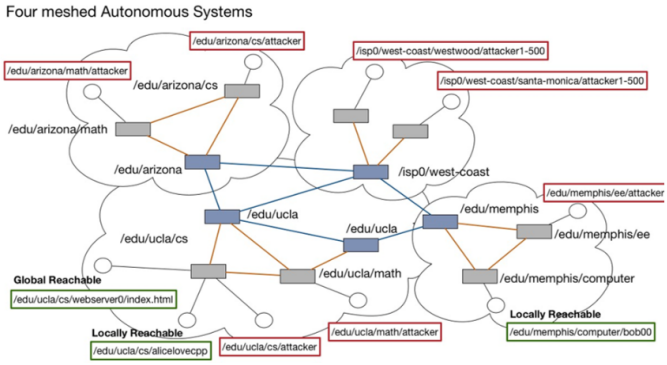

Figure 1: Topology

Figure 2: Plot showing rate of incoming interests for UCLA CS NFD (left) and alicelovecpp (right)

interest was received. It also caches the data in its Content Store and then removes the entry from PIT. In case the corresponding interest packet isn't found in PIT (e.g. it has expired), the data packet is simply discarded.

2.2 ndnSIM

ndnSIM is an open-source NS3 based simulator designed to evaluate and examine different components of NDN Architecture. It allows one to create network topology and simulate traffic. There are also a couple of in-built tracers through which one can obtain metrics such as the number of incoming interests, the number of satisfied interests, the number of dropped packets and other useful statistics. In ndnSIM, one can specify which nodes would serve as consumers and which nodes as producers. Producers can declare the prefix and can serve interests names containing this prefix. However, ndnSIM automatically appends a sequence number to interest packets sent out by in-built consumer application. For example, if producer's prefix is "/prefix" and if we set consumer's interest packet prefix as "/prefix/xyz", the interest packets generated will be "/prefix/xyz/1", "/prefix/xyz/2", etc. All these will be accepted by the producer (as valid interests) because the name contains "/prefix". Hence, it is impossible to send out fake interests to the producer. One of our scenarios involve DoS attack with fake interests and so achieving this was necessary. We wanted a way to treat "/prefix/1", "/prefix/2", etc. as valid interests but "/prefix/xyz", "/prefix/abc", etc as fake interests i.e. producer should only accept interest (as valid interest) if it ends with an automatically appended sequence number. Otherwise, if interest name ends with arbitrary random string not ending with a digit, it should be considered as fake interest by the producer which will then send back a NACK. To achieve this, we created our own consumer and producer application. Consumer application has an option where one could specify whether to auto-append sequence numbers, otherwise, it allows one to create fake interests with random strings.

3 TOPOLOGY

The network topology we use for our experimentation and evaluation is shown in figure 1. The nodes marked in the red box are all attackers (hence, consumers) while the ones marked in green box are producers. Note that in the real world situation, these consumers can also act as producers. However, for simplicity in our simulation, we'll assume that these are just consumers that send out interest packets towards the producers marked in the green box. The text inside box represents the prefix. Producers serve interest names containing their corresponding prefixes if they are valid. The blue boxes in topology are border NFDs while the grey boxes are local NFDs. Each NFD also has a prefix which helps other NFDs to calculate routes appropriately and populate their FIBs.

Another thing to notice from the topology diagram is that producer are marked as *globally reachable* and *locally reachable*. Prefixes for *locally reachable* producers are only known to consumers in their local network while that for *globally reachable* producers are known to all consumers. This is analogous to *www.google.com* which is globally reachable since anyone can reach to it but some internal Google server (e.g. database server) will be internal only within Google AS and won't be reachable to normal users.

For the sake of simplicity, we will call NFD with prefix /edu/ucla as *UCLA NFD*, /edu/ucla/cs as *UCLA CS NFD*, the globally reachable producer with prefix /edu/ucla/cs/webserver0/index.html as *webserver* and the locally reachable producer /edu/ucla/cs/alicelovecpp as *alicelovecpp*.

4 EVALUATION

We study four scenarios that show us usefulness of NFD Content Store, Interest Aggregation Technique, Interest Satisfaction Ratio and Knowledge of Traffic Composition to mitigate the DoS traffic from very beginning and also detect it.

4.1 Scenario 1: Locally Reachable Producer

In this scenario, we consider alicelovecpp as our target victim. alicelovecpp is a locally reachable producer. Since the attackers outside its local network know the prefix of the webserver, they do know that UCLA CS Network exists but except for the webserver, they cannot directly reach other producers as their prefixes are unknown. Hence, all such attackers will simply have to make a random guess. Without the exact prefix in interest name, these interest packets won't be able to reach alicelovecpp and gets dropped at UCLA CS NFD. Therefore, it is hard to overwhelm locally reachable producer by attackers outside its local subnet. However, one can still overwhelm the downstream NFD(s) (UCLA CS NFD in our topology). This is apparent from figure 2. The only traffic that alicelovecpp is receiving is from an attacker in its local network (with prefix /edu/ucla/cs/attacker).

Figure 3: Plot showing rate of incoming interests for UCLA NFD (left), UCLA CS NFD (center) and webserver (right)

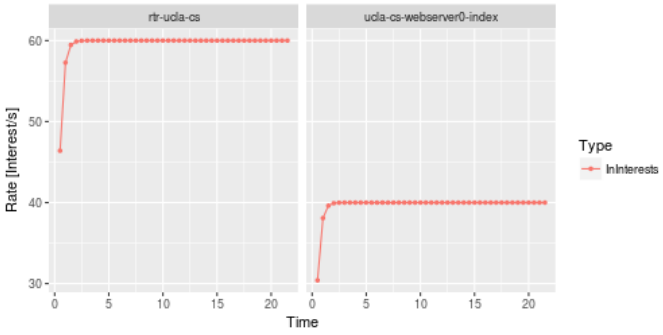

Figure 4: Plot showing rate of incoming interests for UCLA CS NFD (left) and webserver (right)

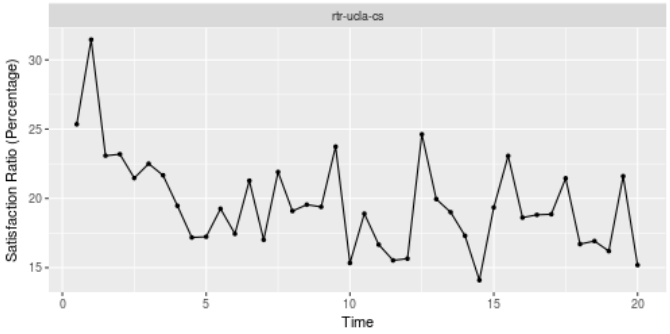

Figure 5: Plot showing interest satisfaction ratio for UCLA CS NFD

4.2 Scenario 2: Cache Can Withhold Much Traffic

In this scenario, we consider webserver as our target victim. The webserver is a globally reachable producer. This allows all the attackers, even the ones outside its local network, to reach webserver. As we saw in 2, the data satisfying consumer's interest gets cached in the routers along the way back to the consumer. If this router then receives the interest with same name corresponding to the cached data, it simply returns the data from its Content Store rather than forwarding the interest all the way to the producer. Hence, if attacker keeps requesting certain data only, eventually all of them will get cached in routers along the path and won't be able to overwhelm producer.

It is apparent from figure 3 that initially all NFDs in UCLA AS and webserver gets high traffic (incoming interests). It then drops fast because data corresponding to most of these interests gets cached in NFDs present in attacker's local AS. The rate of incoming interests almost drops down to zero and remains constant after few seconds because in our simulation we didn't account for cached content freshness period. All cached data in NFD has a timeout period. After the timeout period, the data in cache expires and is deleted. However, even with freshness period, we believe that caching will still help withhold DoS traffic. There could be some traffic bursts but overall the traffic received by producer will still be significantly lesser than what server would receive in traditional TCP/IP structure.

4.3 Scenario 3: Interest Aggregation

In this scenario, we again consider webserver as our target victim. Again as we mentioned in 2, if NFD sees that incoming interest name is already in its PIT, it simply adds the interface of incoming interest to the entry in the PIT and drops the interest packet. In this way, if NFD receives N interest packets with the same name within a small time window, it will aggregate interests and only forward one interest packet towards producer. In DoS, it is likely that multiple attacker nodes will send interests with the same name. But such interest aggregation will allow us to reduce traffic that gets forwarded towards producer. We can see this in figure 4. The rate of incoming interests at webserver is remaining lower than that at UCLA CS NFD. This is because interests from all 7 attackers are aggregated at UCLA CS NFD and only a couple gets forward to webserver.

This also indicates that NDN does stateful routing. Each NFD is aware of the interest name it is receiving. Through this, it can keep knowledge of the underlying traffic composition. For example, NFD can know that about 70% of traffic contains webserver's prefix in interest name while the rest 30% contains alicelovecpp's prefix. This allows detection of DoS attack in NDN.

4.4 Scenario 4: Interest Satisfaction Ratio

In this scenario, our target victim is again webserver. As described in 2, we make use of our custom consumer application "auto-append" feature to distinguish between valid and fake interest. All the attackers outside local UCLA CS network send fake interests to webserver while the local consumer (with prefix /edu/ucla/cs/attacker) sends valid interest to alicelovecpp. Since a majority of the interests in the PIT of UCLA CS NFD are fake, they won't get satisfied. Hence, UCLA CS NFD will observe very low interest satisfaction ratio. This is apparent from 5. The interest satisfaction ratio reaches as low as 5% - 20%. The spikes in the plot are simply there because of some legitimate interest packets sent by local consumer getting satisfied.

This way if some NFD starts seeing lower interest satisfaction ratio, it can warn the network operators about possible DoS attack.

5 RELATED WORK

As per our knowledge, this is the first DoS Simulation for NDN. [1] has been the first step towards assessment and possible mitigation of DoS in NDN. After identifying and analyzing several new types

of attacks, [1] investigates their variations, effects, and countermeasures. However, it doesn't simulate these attacks. [3] describes *Interest Flooding* attack. This is essentially what we do. However, our focus is different. We don't create any defense mechanisms rather we try to study how NDN's design model itself can provide in-built security and show the results proving the same.

6 CONCLUSION

"Security by design" is a primary goal for NDN, a newly proposed Internet architecture, and so it is important to study its in-built resilience to DoS traffic and also detection mechanism. We did analysis of key features of NDN architecture such as NFD Content Store and Interest Aggregation technique which helps mitigate DoS attacks from the very beginning as well as statistics like traffic composition and interest satisfaction ratio would help to detect such attacks.

As part of future work, more scenarios could be studied. For example, study how NDN uses load balancing when the attack occurs. One can also study how to control name prefix route announcements to stop DDoS by interest flooding. Further, one can simulate these attacks on real NDN testbed.

REFERENCES

[1] Paolo Gasti, et al. "DoS & DDoS in Named Data Networking." Computer Communications and Networks (ICCCN), 2013

[2] Lixia Zhang, et al. "Named Data Networking." ACM SIGCOMM Computer Communication Review (CCR), 2014

[3] Alexander Afanasyev, et al. "Interest Flooding Attack and Countermeasures in Named Data Networking." IFIP Networking Conference, 2013

[4] S. Mastorakis, et al. fiOn the Evolution of ndnSIM: an Open-Source Simulator for NDN Experimentation.fi ACM SIGCOMM Computer Communication Review (CCR), 2017

[5] Manos Antonakakis, et al. "Understanding Mirai Botnet." USENIX, 2017

[6] Kumar, et al. "Biggest-Ever DDoS Attack (1.35 Tbs) Hits Github Website.", The Hacker News, 2018.