Interest Flooding Attack Countermeasures Assessment on Content Centric Networking

Interest Flooding Attack Countermeasures Assessment on Content Centric Networking

Matteo Virgilio, Guido Marchetto and Riccardo Sisto
Department of Control and Computer Engineering
Politecnico di Torino
Torino, Italy
{first.last}@polito.it

***Abstract*— Content Centric Networking (CCN) has introduced new concepts and ideas in the next generation routing protocols research area, proposing an alternative approach to the well known and consolidated TCP/IP protocol suite. CCN envisions a network of smart caching devices that not only transport bits from one place to another but also support the network to provide end users with what they are really interested in: named data. However, while a large portion of the existing literature highlights the benefits of this new network paradigm, we focus on some specific security issues related to the opportunity of mounting distributed denial of service attacks, commonly known as Interest Flooding Attack (IFA). Our results confirm this possibility and assess the behavior of state of the art tools designed to mitigate this problem. We run different simulation campaigns in a real deployment scenario to support our evaluation.**

Content Centric Networking, interest flooding attack, security, DoS, simulation

I. INTRODUCTION

Networks have dramatically evolved towards content diffusion: services like on-demand video streaming, social networking and many others, have become the core business of the modern Internet. Despite this clear and straightforward evolution, the communication paradigm is essentially still based on point-to-point channels connecting two network nodes without any specific support for applications that do focus on content diffusion and retrieval. Starting from this evident dyscrasia, new ideas and protocols have been proposed to solve the mismatch: CCN [1] is certainly one of the most promising, proposing a network of nodes capable of delivering data by just looking at the name of the resources requested by clients and completely abandoning the approach based on location dependent host addresses. This vision augments the range of possible functionalities we can implement on board of routers, enabling either data caching at packet granularity and also the exploitation of smart management algorithms to increase network efficiency (from the user perspective) and maximize revenues (from the operator point of view).

While a wide part of the existing literature focuses and highlights the various benefits that may be achieved by the data centric proposal, this paper specifically tackles the security aspects related to a Denial of Service attack that could be implemented on top of CCN. This attack is also known in the literature as Interest Flooding Attack, meaning one or more end users flooding the network with Interests targeted at non existing resources, which only waste router memory and CPU resources. While traditional DoS attacks are usually targeted at the end user resources depletion or damaging, in CCN there exists the possibility to target intermediate nodes, i.e. routers belonging to the service provider network, thanks to the stateful nature of the proposed protocol. As a consequence, the potential impact of such type of attack is way greater and can potentially affect many more users. In this context, we aim to assess the existing countermeasures and evaluate their performance on a real ISP network, considering in particular the backbone network of Telecom Italia, a prominent Italian ISP. Simulations are designed to obtain a fair and significant comparison for all the considered countermeasures since the network topology and the network traffic are kept uniform during all the tests performed.

The rest of the paper is organized as follows. We first introduce the basic foundation of the CCN protocol functioning (Section II), we present the basic attack aimed at overloading CCN routers and give an overview about existing work and research directions (Section III). Then, we dive into the current frameworks designed to mitigate the abovementioned security threats and present our simulation results showing their performance by means of our custom implementation of such mechanisms over the ndnSim simulator [13] (Section IV). We finally conclude this work and draw up some possible future contributions in Section V.

II. BACKGROUND

CCN is built around the key concept of *name*. Each piece of data has a unique name through which it is uniquely identifiable and reachable throughout the network. These names have a hierarchical structure so as to make the routing infrastructure more efficient and scalable.

The packets used in the CCN universe belong to two categories: (*i*) the *Interest* is the packet used to request a piece of content. Basically, it has no payload and just carries the information about the wished content and some other information useful for packet processing and forwarding, (ii) the *Data* is the packet used by content providers to issue responses. It includes the name of the transported content and the digital signature of its payload. Security in CCN is managed by means of per packet signature which ensures data authenticity and integrity.

The focus of this paper is especially placed on the Pending Interest Table (PIT) because all the traffic state information are essentially stored there and the amount of memory destined to it cannot be more than some hundreds of MBs, given the actual RAM technology of modern high-end routers, as discussed in [4]. For this reason, the PIT quickly becomes a possible point of failure for the entire infrastructure and a powerful attack vector, hence we need to quantify this threat in a real world topology with a realistic workload.

III. INTEREST FLOODING ATTACK

In traditional IP networks, DDoS attacks usually plague end terminals since the connection information state is kept by these devices. On the other hand, CCN is hardly based on the fact that intermediate routers maintain per packet state. This feature allows the protocol to avoid routing loops since each Interest is recorded into the PIT table, and also to implement native multicast support because each node "remembers" who asked for what. However this feature arms attacking users because, as we will show in this section, there exist the possibility to artificially generate forged packets with the only aim of wasting router memory. In particular, let us consider the scenario depicted in Figure 1.

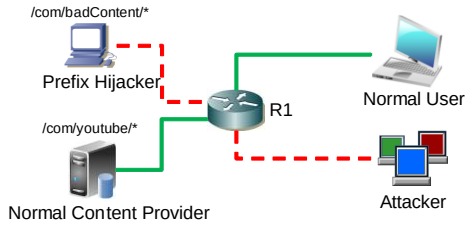

Figure 1 Interest Flooding Attack example

The attacker needs to announce a valid prefix to send fake Interests to; in our example, "/com/badContent/*" serves to this purpose and let us call that node "prefix Hijacker". In addition, the attacker needs one or more zombie clients (or even a large botnet) to start sending Interests targeted at the existing prefix but with non existing (and possibly long) resource names, for example "/com/badContent/abcdefg…z". Such packets will correctly reach the prefix Hijacker, i.e. the machine configured to receive these datagrams without generating any response, but no useful data will be sent back. With this simple procedure, all the Interests seen by R1 (and for which R1 creates an entry in the PIT) will remain in the device memory until the timeout, called LifeTime[1], expires.

The IFA is analyzed in several papers, which also discuss some possible solutions. For example, [5, 6, 7, 8, 9] contain a preliminary evaluation of possible security threats applicable to the CCN/NDN architecture. To the best of our knowledge, three main proposals are targeting the specific problem of IFA. The work by Afanasyev et al.[10] presents a framework named Satisfaction Based Pushback to limit the number of forwarded Interests for a given prefix depending on statistics about the Interest satisfaction ratio. Another countermeasure, referred to as Poseidon, is presented in [11]. This approach is similar to the previous one in that it gathers statistics about the traffic seen at each router but with a different activation procedure, as we will show in the following. The last solution we will consider, referred to as Traceback, is described in [12]. The idea is to activate a countermeasure after the memory usage has reached a predefined threshold. The algorithm consists in generating spoofed Data packets for the entries that are causing the memory overflow problem. Our goal is to evaluate and compare them in our use case scenario, i.e. the network of the main Italian service provider.

[1] The Lifetime is set by the client within the Interest packet so the router has no control over it.

IV. COUNTERMEASURES

A. *Satisfaction based pushback*

The Satisfaction based pushback algorithm is described in [10] and it works as follow: each router computes the Interest satisfaction metric as the ratio between the number of satisfied Interests over the number of forwarded Interests. This computation is performed on a per interface basis and it is an indication about the probability of an Interest coming from a certain interface to be satisfied. Such metric can be directly used to calculate the limits of Interests the router is willing to forward from each interface. After the computation, the router announces its limits to downstream neighbors in order to rate limit the incoming traffic, especially for those interfaces that are increasing the burden on the PIT memory occupancy. After some time, the router computes new statistics and clears its history with an exponential decay, in order to restore the original limits and give a chance to each interface to have more virtuous Interests forwarded again. Notice that metrics about the traffic may be collected at different granularities: prefixes, FIB entries and so on.

B. *Interest Traceback*

The Traceback algorithm is described in [12] and it is designed to release the unwanted PIT entries when the available amount of memory space falls under a predefined threshold. The detection phase is rather simple and only requires to monitor the PIT memory usage over time. After detecting an abnormal memory occupancy, the Traceback process is triggered and a set of spoofed Data packet are generated for those entries that remained unsatisfied for a long time. The spoofed Data packets carry the name needed to satisfy the offending Interests and are forwarded downstream to release resources all along the path.

In order to leverage, as much as possible, the memory space available to the PIT, we defined the threshold after which the Traceback is activated, as 90% of the occupied memory. Such aggressive limit avoids algorithm overreacting and allows the network to support temporary traffic peak without triggering any Interests blocking mechanism. Since some implementation details were omitted in the reference paper, we designed our code to meet as close as possible, the countermeasure description. In particular, we simulate for each router a monitoring process which is scheduled every second to check if the memory occupancy is over its alarming value. If (and only if) it exceeds our threshold (over 90% memory occupancy), we invoke the FindAndSend() function to generate spoofed Data packets and make them travel towards the attack initiator. A

simplified high level vision of our implementation can be seen in Algorithm 1.

```
void Traceback::FindAndSend()
{
    FOR EACH Entry in Pit
        IF IsOld(Entry)
            FOR EACH Face in Entry.FacesList
                IF Face.IsConnectedToEndUser()
                    BLOCK Face
                ELSE
                    GENERATE SpoofedData
                    SEND SpoofedData through Face
                END IF
            END LOOP
            RELEASE memory
        END IF
    END LOOP
}
```

Algorithm 1 Traceback sending spoofed Data

C. Poseidon

Poseidon[11] is a framework to mitigate the effect of the IFA on CCN/NDN networks. It shares some similarities with the previous approach since it also collects statistics by observing the forwarded traffic. The main difference is in the detection phase: Poseidon is triggered when two metrics exceed their corresponding thresholds. The two parameters used by Poseidon are defined as follows:

1) $\omega(r_i^j, t_k) = \frac{\#\ Interes\ ts\ from\ r_i^j\ at\ time\ t_k}{\#\ Data\ from\ r_i^j\ at\ time\ t_k}$

2) $\rho(r_i^j, t_k) = \#\ of\ PIT\ bytes\ used\ by\ r_i^j\ at\ time\ t_k$

In order for Poseidon to be activated, *both* of these two metrics must exceed the allowed value. The algorithm uses both in order to limit the number of false positive, namely the number of times it erroneously detects an in progress attack.

Poseidon reacts to an attack detection by imposing limits on the number of accepted Interests from the interface which exceeded both thresholds and lowering them for that interface. After some time, if the traffic becomes normal again, Poseidon will restore all the thresholds to their original values and the imposed limits are deactivated.

D. Simulation Scenario

Our simulation scenario is the network of the main Italian service provider, Telecom Italia (TI), whose logical topology is publicly available in [2] and exploited in [9], with PoP (Point of Presence) granularity. The connection between each user and its corresponding POP is modeled as an ADSL line with 7Mbps/1Mbps downlink/uplink bandwidth because these are very common values for TI domestic DSL contracts. The total number of customers is around 10 million and their distribution in the network is coherent with the population density of each province. The PIT size has been set to 1 GB in order to consider a static RAM based implementation and meet current hardware technology advances. See [4] for a deeper insight on this topic.

To load our network, we implement download arrivals at each client side and limit customers to download just one file at a time, for simplicity and scalability of the simulations. Each file to be requested is selected among the global resources catalog with a Zipf probability distribution having $\alpha = 0.55$ and $q = 25$ as in [9]. This traffic load represents our baseline for all the simulations.

The attacker model is rather simple since it generates Interest packets at the maximum speed allowed by its uplink bandwidth. We distributed many attackers around the network, targeting the same prefix in order to concentrate the effects on a central device, which, in our scenario, is the Rome PoP. For this reason and also for the sake of brevity, we provide results and metrics only for this network appliance. One prefix Hijacker node is directly connected to the Rome PoP to attract all the fake Interests and discard them. Such behavior makes PIT entries unsatisfied for the whole Lifetime thus wasting precious memory portions.

E. Simulation results

We run each scenarios with our customized version of the ndnSim simulator and provide simulation results either in terms of memory performance (RAM usage) and also in terms of the overall network functioning (percentage of retransmissions and total number of completed downloads). We report the RAM usage as the amount of memory occupied just by the PIT in a stable situation, i.e., after any transient has disappeared.

After implementing the countermeasures in the network, we run a simulation campaign for each of them and obtained the results depicted in Table 1. Attack bandwidth has been varied in the range [0 – 4] Gbps. Notice that the max attack bandwidth (4Gbps) is perfectly feasible since many security reports[3] confirm the possibility to obtain an aggregate attack bandwidth even higher than 10 Gbps. We start our analysis with the Pushback algorithm. As we can see from the results, an increasing attack bandwidth causes a worse network performance, especially considering the overall number of downloaded files. The surprising result is that the countermeasure limits the network also in case of low intensity attacks because the algorithm is designed to compute the maximum number of 'acceptable' Interests and announce it to downstream routers. Since the fake Interest packets mix with normal requests, the resulting limit computed for the interfaces of the routers along the path targeted by the attacker, involves also part of the legitimate traffic. The worse performance cannot be captured in the retransmissions computation since, as previously mentioned, it is performed only for finished file transfers thus not taking into account downloads in progress at the end of our simulations.

For what concerns the Traceback framework, results are definitely better and almost all the files are correctly delivered. Only in the last case, with an aggregate attack bandwidth of 4 Gbps, some downloads are not completed. The reason is that we have some transients between each attack detection and the countermeasure deployment, so that some regular Interests are initially discarded by on path routers. After reaching the threshold set for the Traceback process (this only happens in the last two rows where the attacker band is equal or higher than 2Gbps), the countermeasure is triggered and the spoofed Data immediately release the memory wasted on intermediate nodes.

Attack Bandwidth	*Retransmissions*			*RAM Usage*			*Total downloads*		
	Pushback	Traceback	Poseidon	Pushback	Traceback	Poseidon	Pushback	Traceback	Poseidon
0 bps	0%	0%	0%	0.2%	0.2%	0.2%	2841000	2841000	2841000
100 Mbps	0%	0%	0%	5%	5%	0.3%	2839500	2841000	2841000
500 Mbps	2%	0%	0%	0.7% – 6.4%	25%	0.3%	2707000	2841000	2841000
2Gbps	≈ 0%	≈ 0%	≈ 0%	0.7% – 6.4%	0.2%	0.3%	2706500	2841000	2841000
4Gbps	≈ 0%	13%	≈ 0%	0.7% – 6.4%	0.2%	0.3%	2706500	2837500	2841000

Table 1 Countermeasures simulations results with different attack bandwidths

The routers, which give connectivity to end users as well as attackers, quickly identify the attack originator and lock the link. In this way the attacker activity is completely denied and the network operating restored, as proved by the good performance achieved in terms of finished downloads and memory usage. The number of retransmissions is slightly higher because, during the transient, some downloads may experience a slow down.

The last framework under test is Poseidon. As evident from Table 1, simulation results are even better than the previous cases as confirmed by the total number of finished downloads, which is completely restored in all the considered attack scenarios. This is a positive consequence of the dynamic behavior of Poseidon and its combined usage of two parameters, ω e ρ.

The considered thresholds are automatically lowered while the attack is starting, resulting in less probability for the attacker to have its Interests forwarded upstream. After the statistics become normal again (with an exponential decay law), thresholds are raised again to progressively reopen the link to the attacker. However, this oscillation is never dangerous for the network performance as the system performs an early detection of this phenomenon thanks to the monitoring process that is triggered with 1 second frequency, and the regular traffic is definitely not affected by the fake traffic. This last result reveals that Poseidon is the most resilient framework against IFA and can successfully shield the considered network topology under the assumptions made for the attacker behavior.

V. CONCLUSION

CCN is a fertile research area, attracting many contributions from different research groups within the Academia as well as in the Industry. In this paper, we focused on the specific problem of assessing current frameworks to mitigate the possibility to implement IFA on CCN networks. The main outcome is that Traceback and Poseidon are promising mechanisms to alleviate the problem while not affecting normal clients performance. Especially Poseidon and its algorithm consisting of two independent thresholds, revealed to be effective in all the considered scenarios and a promising general approach to solve the problem.

Possible future work includes the analysis of more sophisticated attack behaviors, in order to prove the accuracy of these algorithms also in different use cases and with different network topologies. The final goal is to study further security issues related to the CCN protocol and ensuring an adequate level of security to the entire infrastructure, for example considering more PIT architectures and more traffic conditions.

REFERENCES

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard. 2009. Networking named content. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies* (CoNEXT '09). ACM, New York, NY, USA.

[2] A. Soldati. Telecom italia ip backbone and peering policies. In *Italian Peering Forum* (PFI 2008), 2008.

[3] Radware. Global application & network security report. Annual Report, 2011.

[4] D. Perino and M. Varvello. 2011. A reality check for content centric networking. In *Proceedings of the ACM SIGCOMM workshop on Information-centric networking* (ICN '11). ACM, New York, NY, USA.

[5] M. Wählisch, T. C. Schmidt, and M. Vahlenkamp. 2013. Backscatter from the data plane - Threats to stability and security in information-centric network infrastructure. *Comput. Netw*. 57, 16 (November 2013).

[6] 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*. July 30 2013-Aug. 2 2013

[7] S. Choi, K. Kim, S. Kim and B. Roh. Threat of DoS by interest flooding attack in content-centric networking. In *Information Networking (ICOIN), 2013 International Conference* 28-30 Jan. 2013

[8] K. Wang, J. Chen, H. Zhou, Y. Qin and H. Zhang. Modeling denial-of-service against pending interest table in named data networking. In *Int. J. Commun. Syst*.. 26 July 2013

[9] M. Virgilio, G. Marchetto, and R. Sisto. 2013. PIT overload analysis in content centric networks. In *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking (ICN '13)*. ACM, New York, NY, USA.

[10] A. Afanasyev, P. Mahadevan, I. Moiseenko, E. Uzun, and Lixia Zhang. Interest flooding attack and countermeasures in named data networking. In *IFIP Networking Conference*, 2013.

[11] A. Compagno, M. Conti, P. Gasti and G. Tsudik. Poseidon: Mitigating Interest Flooding DDoS Attacks in Named Data Networking. In *38th Annual IEEE Conference on Local Computer Networks (LCN 2013)*, Sydney, Australia

[12] H. Dai, Y. Wang, J. Fan, and B. Liu. Mitigate ddos attacks in ndn by interest traceback. In *NOMEN'13*, Turin, Italy, Apr. 2013

[13] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for NS-3," NDN, Technical Report *NDN-0005*, 2012