Exploiting Information Centric Networking to Build an Attacker-Controlled Content Delivery Network

Giulia Mauri*§, Riccardo Raspadori*‡, Mario Gerla†, Giacomo Verticale*§
* Department of Electronics, Information, and Bioengineering, Politecnico di Milano
§{name.surname}@polimi.it, ‡riccardo.raspadori@mail.polimi.it
† Department of Computer Science, University of California, Los Angeles
gerla@cs.ucla.edu

***Abstract*—The Information Centric Networking (ICN) model relies on the ubiquitous use of caching to improve performance and reduce bandwidth requirements. ICN also makes it possible for routers to fetch content from downstream nodes, such as when a content for a home user is fetched from a neighbor's home router, with significant performance improvement. This paper shows how an attacker using compromised hosts can easily gather a massive amount of low-cost, low-latency storage for malware, junk, and other attacker-controlled content. We conclude by considering a possible countermeasure, a blacklist fed by a honeypot, which we show to be effective.**

***Index Terms*—Information Centric Networking; Named Data Networking; ndnSIM; Cache Pollution; False Locality; Honeypot;**

I. Introduction

The Information Centric Networking (ICN) framework is emerging as a way to improve the performance of content delivery by leveraging on the pervasive use of caching. The ICN approach is used in several proposals and prototypes, among which the Content Centric Networking (CCN) protocol [1] and the Named Data Networking (NDN) project [2], to name a few. ICN nodes choose what content to keep in their Content Stores, the name used in CCN for caches, according to some reactive policy. This is an important difference with respect to the pre-provision algorithms of standard Content Delivery Networks. Thus, the information centric paradigm provides a more neutral service to the user, but leaves room for a malicious user to implement entirely new attacks to the cache management process. The literature identifies two broad classes of attacks: cache pollution attacks, in which the attacker forces an ICN node to keep unpopular contents in its cache in order to exhaust it [3], [4] and cache snooping, in which the cached content is used to get information about the downstream users [5], [6].

To our knowledge, most of the research on cache pollution attacks has focused on Denial-of-Service (DoS) attacks. This paper shows that attackers can do more and build a large storage network that can be used to store junk content, malware, and in general any kind of attacker-controlled content. This storage is naturally very near to the end user and can be exploited to serve such content with low latency. This ability provides a clear economic incentive for the adversary, and thus makes this kind of attack especially attractive.

978-1-4673-7306-7/15/$31.00 © 2015 IEEE

In our scenario, the attacker creates a *false locality* into the user caches by using compromised nodes to send a high number of requests for specific content objects. This way, the attacker can quickly spread its contents towards peripheral nodes. The attacker has an incentive in this attack because it can rent this storage to producers who, for any reason, cannot invest in infrastructure.

The attack is particularly effective when the network exploits the ability of CCN nodes of looking for content across several interfaces using for example a Nearest Replica Routing (NRR) as shown in [7]. A typical example would be an access network in which end-users are equipped with CCN set-top boxes. In such scenario, the access nodes could retrieve content objects requested by a user from the set-top box of another user in a peer-to-peer fashion, thus greatly enhancing the caching ability of the access network and reducing the bandwidth requirement of the core network. By exploiting the low cache churn rate of low-activity users, the attacker can easily obtain control of a large fraction of the storage available at these users' premises.

The next Section II reviews some of the related work about cache pollution attacks. The attacker model with its goals and capabilities is presented in Section III. While, the Section IV shows the evaluation scenario together with the parameters used for the attack analysis presented in the following Section V. In Section VI, we suggest a countermeasure to mitigate the attack. We conclude in Section VII.

II. Related Work

The cache pollution attacks are not new in the networking scenario. Since proxy caching servers are widespread in the IP-based networks, a DoS attack is easy to deploy. The paper [8] presents false-locality and locality-disruption attacks and efficient methods to detect them. A false-locality attack happens when the adversary continuously requests the same set of files, creating a false file locality at the caches. Instead, in a locality-disruption attack, the attacker generates requests for otherwise unpopular content. The authors suggest to use a metric, the byte damage ratio, to measure the effects of the previous attacks. Then, they describe how to detect and mitigate such pollution attacks. Although the paper is not focused on ICN, it is a starting point for our work.

The cache pollution attacks on CCN are investigated in [9].

The authors try to make caching in CCN more robust using a proactive mechanism called CacheShield. This mechanism is based on a shielding function that computes the probability for each content to be cached. Thus, the unpopular contents are excluded from caching. The solution is evaluated under different attacks in various network topologies. However, the effectiveness is limited to the scenarios taken into consideration. Instead, the paper [10] illustrates the inefficacy of CacheShield against realistic adversaries. The authors also present a new lightweight detection method for cache pollution attacks. The mechanism is tested with simulations and provides accurate results in different topologies. Nonetheless, the paper does not suggest countermeasures against this kind of attacks.

Another problem related to the caches is the content poisoning, as presented in [11]. In a content poisoning attack, the adversary injects junk content into the router caches. The paper analyzes the problem and suggests a ranking algorithm that allows routers to make decisions about the content validity. Instead, our paper does not consider the content validity but the problem of spreading attacker-controlled contents in the network caches.

As far as we know, our paper is the first that considers the attractiveness and the effectiveness of a cache pollution attack that aims to build an attacker-controlled Content Delivery Network. Moreover, our work highlights the incentive for the adversary in improving the caching ability of the access network and reducing the bandwidth requirement of the core network.

III. Attack Description

Our attacker is an entity whose goal is distributing content objects into end user storage, therefore the network can provide those contents with low latency to users asking for them. The attacker focuses its effort on the content objects whose name is in a list $\mathcal{A}$.

The attacker entity has the capability to compromise a set of consumer nodes, creating a botnet under its control. Particularly, the attacker can command:

- the frequency of requests sent by the compromised nodes and the distribution of the requests;
- the names of contents to be requested.

However, the attacker cannot change the cache size and the parameters related to the non compromised traffic: a compromised node sends interests according to the node owner's requests in addition to the attacker-controlled requests. Also, it is not able to compromise the router nodes, in particular it cannot modify their routing tables and Content Stores.

In order to prevent detection, the attacker must avoid starving the legitimate requests, therefore it should find a tradeoff between its gain and the service degradation caused to the compromised nodes. We do not discuss how to find this tradeoff, but evaluate the attack success depending on the number of compromised nodes.

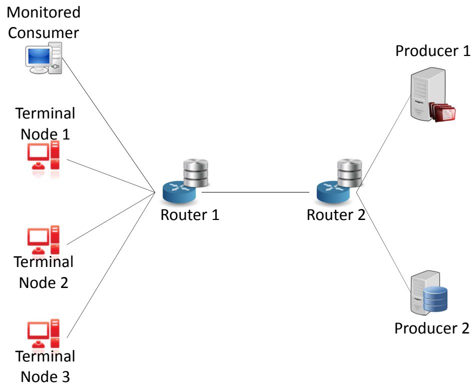

Fig. 1. The network topology

IV. Evaluation Scenario

To evaluate the impact of the attack, we consider the network scenario shown in Figure 1, in which the users have the following roles:

Producer 1. Its Content Store contains all and only the content objects whose name prefix is `/prefix1`. We assume that the names in list $\mathcal{A}$ are a subset of the names starting with this prefix.

Producer 2. Its Content Store contains all and only the content objects whose name prefix is `/prefix2`.

Monitored Consumer. It sends interests for objects whose name starts with `prefix1` with a probability according to the Zipf's law. Consequently, this Consumer may ask for contents in $\mathcal{A}$ or not.

Terminal Nodes. They request contents whose name starts with `prefix2` following the Zipf's distribution. Additionally, one or more of these nodes can be compromised. A Compromised Node (CN) also requests contents from $\mathcal{A}$.

Routers. They forward interests and data packets using a simplified version of the NRR policy. When a router receives a data packet with name n from the downstream interface i, it adds the interface i to its FIB as the next hop for n. Then, when an interests arrives for n, and n is not in the Content Store, the router forwards the interest towards i. If, after a short timeout, the corresponding data packet does not arrive, the interest is forwarded upstream towards the Producer and the association (n, i) is removed from the FIB.

V. Attack Analysis

In this Section, we answer the following questions:

1) Can the producer take advantage from the attack and how can we measure this advantage?
2) What are the network conditions that mostly influence the benefits for the attacker?
3) How much the performance of the compromised node is decreased?

We compare different attacker's strategies for choosing the most convenient set of nodes to compromise to reach the goal presented in Section III.

We conduct simulations using the open-source ndnSIM package [12], which implements the NDN protocol stack for the ns-3 network simulator. The simulations last 7200 s, after reaching the steady state. The shown results have a confidence of 95% or better.

We evaluate four scenarios that differ for the lack or presence of caches in the router nodes and for the RTT (Round Trip Time) between Users and Producers:

A. a short range network with RTT = 30 ms,
B. a long range network with RTT = 130 ms,

Both the scenarios can have the following cache configurations:

1. router content stores have negligible size.
2. both users and routers have content stores.

The link between each pair of nodes is 1 Gbit/s to avoid bottlenecks. Moreover, we suppose that the links do not introduce errors during transmission. The content catalog of each Producer comprises 10000 content objects. Each content has a size of 1000 bytes. The Content Store of the Monitored Consumer and of the Routers can store 100 contents, that is the 0.5% of the total. The Terminal Nodes have Content Stores of size 600 contents. All the Content Stores use a LRU policy, as mostly assumed by the literature [13], [14], [15].

The Monitored Consumer sends interests for content with name prefix `prefix1` following the Zipf's distribution with $\alpha = 0.9$ and request rate $\lambda_C = 100$ interests per second. The names in $\mathcal{A}$ correspond to the 1000 most popular objects with this prefix.

The Terminal Nodes request content with name prefix `prefix2` with rate λ_{CN} and with names chosen according to the Zipf's law. If a node is compromised, then it also requests contents from $\mathcal{A}$ with rate $\mu = 10$ interests per second. The names are chosen uniformly from $\mathcal{A}$. We define a *cache contention* parameter, C, as the ratio between the legitimate requests and the attacker-controlled requests: $C = \lambda_{CN}/\mu$. In order to avoid detection, the attacker must keep its request rate low, therefore we will only consider scenarios in which $C \geq 1$. In the following, we depict results relative to $C = 1, 5, 10$, meaning that $\lambda_{CN} = 10, 50, 100$.

A. Effectiveness of the Attack

Figure 2 shows the latency perceived by the Monitored Consumer in retrieving the 1000 most popular contents versus the cache contention, C, which measures the ratio between the legitimate traffic and the attacker-controlled traffic at the compromised nodes. Results are shown for different numbers of Terminal Nodes that become Compromised Nodes (CN).

With no compromised nodes, the latency is about 18 ms, the delay remains constant and it is higher than the other scenarios. Whereas, in presence of the attack, the delay increases with an higher level of contention, i.e. when the attack is less obvious. However, even with a high contention, the latency is always lower than the values of the scenario without the attack. Thus, this figure shows that the attacker gets an advantage in pushing the content it controls towards the consumers.

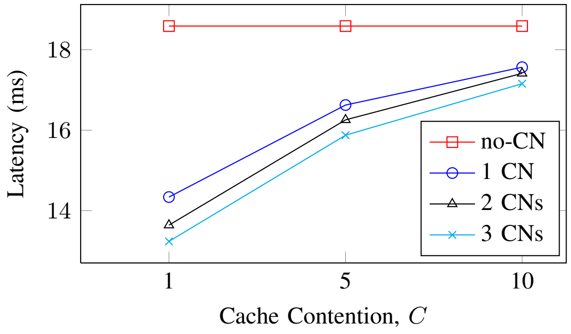

Fig. 2. Latency perceived by the Monitored Consumer.
Scenario A1: RTT=30ms, negligible Router caches.

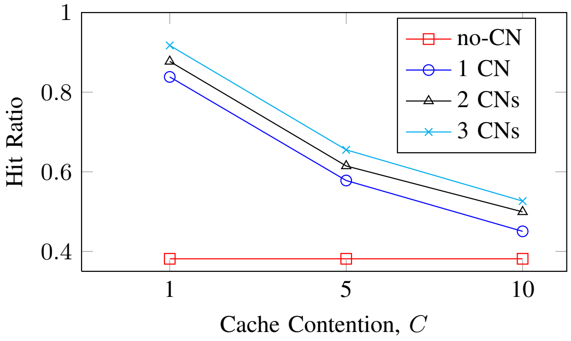

Fig. 3. Hit Ratio at the Monitored Consumer.
Scenario A1: RTT=30ms, negligible Router caches.

Figure 3 shows the cache hit ratio measured at the Monitored Consumer versus the cache contention, C. With no attack, the hit ratio is about 0.4.

In case of attack, the requests for attacker-controlled contents from the compromised nodes are routed also to the Monitored Consumer, increasing their perceived popularity and making them stay longer in the Content Store. When the attacker-controlled traffic is high ($C = 1$), the hit ratio grows significantly, to about 0.8 with a single compromised node up to almost 1 with three compromised nodes, meaning that attacker-controlled traffic is unlikely to be pushed out of the Content Store. Clearly, this behavior becomes weaker as the attack intensity decreases. With $C = 10$ and two or three compromised nodes, the effect of the attack on the Content Store of the Monitored Consumer can still be observed, while with a single compromised node, the effect of the attack is negligible.

Figure 4 presents the latency perceived by the Terminal Nodes in retrieving content from Producer 2, versus the cache contention, C. Results are shown for different numbers of Compromised Nodes (CN).

During the attack, the Terminal Nodes are forced to request content from Producer 1 and the perceived latency relative to content retrieved from Producer 2 grows. If the Terminal Nodes are not compromised, the latency is about 9 ms. In case of attack, the latency is always larger, showing that the attack bestows worse performance on the affected nodes. Clearly,

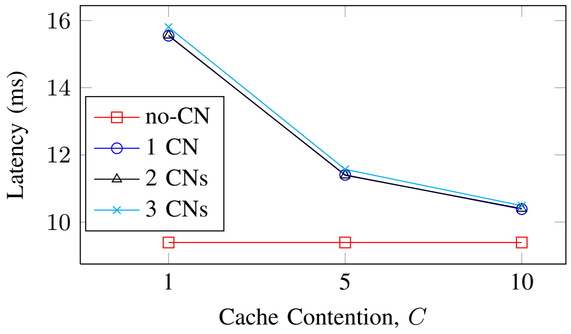

Fig. 4. Latency perceived by the Terminal Nodes.
Scenario A1: RTT=30ms, negligible Router caches.

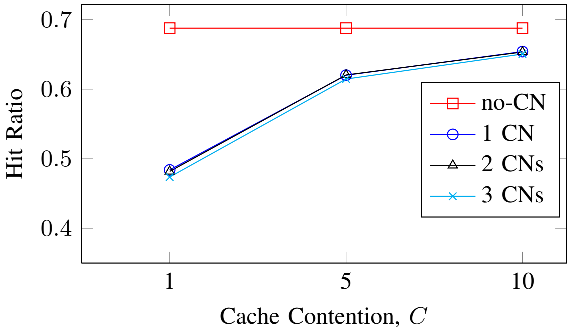

Fig. 5. Hit Ratio at the Terminal Nodes.
Scenario A1: RTT=30ms, negligible Router caches.

the latency decreases with a weaker attack, specifically with $C = 10$ the service degradation is small and the attacker might be more difficult to detect.

Figure 5 shows the cache hit ratio measured at the Terminal Nodes depending on cache contention, C. Results are shown for different numbers of Compromised Nodes (CN).

With no attack, the hit ratio is about 0.7. In case of attack, the hit ratio decreases due to the additional requests for attacker-controlled content. With an higher percentage of attacker-controlled traffic, the hit ratio lowers significantly up to about 0.5. While, with $C = 10$ the effect of the attack on the Compromised Nodes hit ratio is negligible. As it can be noticed, both the perceived latency by the Terminal Nodes and the hit ratio are inversely proportional to the perceived latency by the Monitored Consumer and its hit ratio, meaning that the better the service perception for the Monitored Consumer, the higher the service degradation for the Terminal Nodes.

Figure 6 describes the number of content packets sent per second by the Producer 1 versus the cache contention, C and for different numbers of Compromised Nodes (CN).

With no attack, the number of packets sent per second is around 75 packets/s. This number decreases when the Terminal Nodes start sending interest for attacker-controlled content. In particular, the smaller the number of Compromised Nodes, the smaller the number of content packets sent by the Producer 1. For example, the value with one Compromised Node is about 45 packets/s. As it can be expected, the results do not depend on the cache contention because the rate for the attacker-controlled contents, μ, is constant. Thus, this figure clearly depicts that the attacker, i.e. the malicious Producer, has benefits in terms of bandwidth savings. Indeed, the Compromised Nodes become responsible for pushing content to the requesting Consumer instead of the malicious Producer.

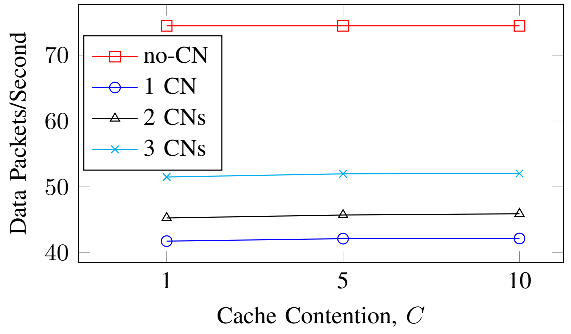

Fig. 6. Number of contents sent by the Producer 1.
Scenario A1: RTT=30ms, negligible Router caches.

The previous results prove that it is possible for a malicious Producer to launch an attack in order to improve the user experience and to reduce the bandwidth and storage requirements of the core network. In the previous setups, it is possible to get both the advantages with an appropriate choice of the network parameters. In particular, the Producer can choose the number of Terminal Nodes to compromise, and the percentage of attacker-controller traffic. At a glance, if $C = 1$, the scenario seems the most advantageous. Indeed, the number of deliveries by the Producer is smaller, and the perceived latency by the Monitored Consumer is decreased more than the scenario with no attack. However, the performance relative to the Terminal Nodes are worse, and consequently the attack is easily noticeable. Thus, the Producer should find a trade-off between the advantages and the possibility to be detected.

B. Results for Scenario B1

This scenario assumes RTT = 130 ms, and negligible caches in the router nodes.

The results are similar to the scenario A1 presented in subsection V-A. Figure 7 shows the metric relative to the latency perceived by the Monitored Consumer. The latency trend is constant with no attack and is about 80 ms. Also, the more the attacker-controlled traffic and the bigger the number of Compromised Nodes, the smaller the latency perceived by the Monitored Consumer. Thus, we can conclude that also if the distance, in terms of RTT, between the end nodes and the Producer is bigger, the attacker can launch an attack, that results to be effective, by controlling the behavior of some nodes.

We do not show the results for the scenarios A2 and B2 where router nodes have bigger caches, which provide similar figures and allow drawing similar conclusions.

Overall, the Producer can always take advantage from the attacker-controller traffic requested by the Terminal Nodes.

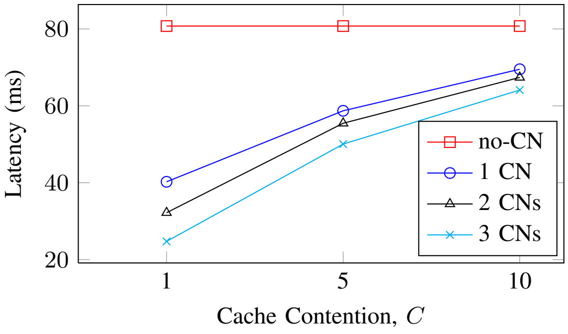

Fig. 7. Latency perceived by the Monitored Consumer.
Scenario B1: RTT=130ms, negligible Router caches.

However, it has to consider different aspects. First, it can choose to compromise only one Terminal Node with a small cache contention. In that case, the number of delivered contents is reduced, the latency perceived by the Monitored Consumer is lowered and consequently the hit ratio is increased. Nevertheless, the Compromised Nodes performance are worse than the other scenarios. Thus, the Producer could consider to compromise one or two Terminal Node(s) with medium cache contention, in order to take advantage and also to not be easily detected.

As a conclusion, we can assert that the malicious producer, exploiting the presented attack, can significantly increase its performance in terms of throughput. Thus, performing a attack can make a Producer save a significant bandwidth in the network. The number of Compromised Nodes and their cache contention, C, influence the results. In particular, the more beneficial is the effect on the Consumer, the more negative is the effect on the Terminal Nodes performance. Finally, the cache size and their location also provide advantages.

VI. Attack Mitigation

The previous Section shows how the attacker can exploit the ICN principles to pile up an attacker-controlled storage. In this Section, we suggest a simple countermeasure based on the idea that the list of names in $\mathcal{A}$ can be identified by means of a honeypot, and the nearest replica routing can be turned off for the identified names.

We install a honeypot over the Terminal Node 1, which we assume to have been compromised by deducing it from the degradation of its performance. This node also hides a monitoring station that collects the attacker-controlled requests. The list of these requests is then sent to Router 1 and stored in a blacklist. Thus, for the contents in this list, the standard NDN routing is used, instead of the modified one that exploits the nearest replica routing, thus reducing or eliminating the attacker's gain.

We evaluate the proposed countermeasure by means of simulations with ndnSIM. We consider the scenario A1 with RTT = 30ms, negligible caches in the routers, and $C = 1$. Differently from the evaluation scenario in the previous Section, we assume that the Terminal Node 1 does not make any legitimate request and, as a consequence, all the requests coming from that node are malicious, i.e. for content from $\mathcal{A}$. In order to simulate the blacklist collection and the choice of the forwarding algorithm on the basis of the blacklist, we compute the a-priori probability for a malicious content to be in the blacklist, $Pr(Bl)$. Whenever an interest packet arrives for an attacker-controlled content, the Router 1 chooses to use the standard NDN routing with a probability $Pr(Bl)$, otherwise it uses the nearest replica routing. First, we find the average number of contents in the blacklist, N_{Bl}, as follows:

$$N_{Bl} = E[Pois(\mu T)] = \frac{e^{-\mu T}}{\mu T!}.$$

The number of contents in the blacklist depends on a Poisson's distribution $Pois(\mu T)$, where μ is the request rate and T is the monitoring window, meaning that every T seconds the blacklist is updated and sent to the Router node.

Then, we compute the probability, $Pr(Bl)$, as:

$$Pr(Bl) = \frac{N_{Bl}}{|\mathcal{A}|} = \frac{\mu T}{|\mathcal{A}|} = 1 - (1 - \frac{1}{|\mathcal{A}|})^{\mu T}.$$

Assuming $\mu = 10$ interest/s, $T = 120s, |\mathcal{A}| = 1000$, we get $Pr(Bl) \simeq 0,7$. Thus, for contents in the blacklist there is a probability of 70% to be routed using the standard NDN routing instead of the nearest replica routing. Then, considering a monitoring window of $T = 230s$, we get $Pr(Bl) \simeq 0,9$, which corresponds to a probability of 90%.

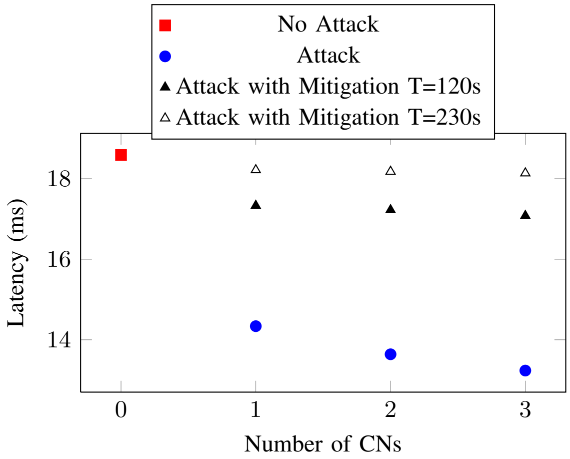

Fig. 8. Latency perceived by the Monitored Consumer in retrieving the attacker-controlled contents before the attack, during the attack and with the attack using the mitigation technique depending on the number of Compromised Nodes.

We measure the latency perceived by the Monitored Consumer in retrieving the attacker-controlled contents versus the number of Compromised Nodes (CN). The results are presented in Figure 8 for the different scenarios: with no attack, with attack, and with attack using the proposed mitigation technique.

The red square represents the latency perceived by the Monitored Consumer with no attack, and it is a little more than 18 ms, as already shown in Figure 2. While, the latency perceived during the attack is represented by the blue dots and decreases with the number of Compromised Nodes. The Figure highlights an improvement of about 20% relative to the scenario without the attack. Finally, we represent the perceived latency after exploiting the countermeasure to mitigate the attack, the results are depicted with the black triangles. In that case, the results do not depend on the number of Compromised Nodes. We notice that the latency goes near the scenario without the attack, i.e. the latency is about 17 ms for $T = 120s$ and about 18ms for $T = 230s$. Thus, the improvement becomes of only 5% or 2%, respectively. We recall that we choose a monitoring window of $T = 120s$ or of $T = 230s$, that leads to a probability of 70% and of 90% for a content to be in the blacklist. Overall, we can conclude saying that the countermeasure is effective in reducing the attacker's incentive by decreasing its gain. If we enlarged the monitoring window, we would get a bigger probability and so we could bring back the perceived latency to the case without the attack. This means that the attacker completely looses its gain, as expected. Thus, it has no incentives in taking the control of the Terminal Node(s) for requesting its contents becoming a simple DoS attack. This, however, comes at the expense of higher complexity and less robustness to changing conditions.

VII. Conclusion

This paper shows that, under realistic assumptions on the attacker ability, the ICN paradigm can be exploited by malicious users to take control of a large amount of storage near the end-user. This storage can be used to propagate junk, malware, or other content with low latency without requiring any investment in infrastructure. By means of simulations, we show that the attack is particularly effective, providing the attacker with bandwidth savings and excellent performance in exchange for a limited effort.

Fortunately, there are some countermeasures that can be used to reduce the effectiveness and thus, the attractiveness of the attack. In particular, we show with simulations that a monitoring station, i.e. a honeypot, can significantly alleviate the issue by sending a blacklist of contents likely to be attacker-controlled to the routers nodes.

References

[1] V. Jacobson, , D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. Braynard, "Networking named content," in *Proc. of the 5th international conference on Emerging networking experiments and technologies*, ser. CoNEXT '09. ACM, 2009, pp. 1–12.

[2] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, K. Claffy, D. Krioukov, D. Massey, C. Papadopoulos, T. Abdelzaher, L. Wang, P. Crowley, and E. Yeh, "Named data networking (ndn) project," University of California and Arizona, Palo Alto Research Center and others, Tech. Rep., October 2010.

[3] 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*, July 2013, pp. 1–7.

[4] A. Afanasyev, P. Mahadevan, I. Moiseenko, E. Uzun, and L. Zhang, "Interest flooding attack and countermeasures in named data networking," in *IFIP Networking Conference, 2013*, May 2013, pp. 1–9.

[5] 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.

[6] 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 2012 ACM Conference on Computer and Communications Security*, ser. CCS '12. New York, NY, USA: ACM, 2012, pp. 1001–1003.

[7] G. Rossini and D. Rossi, "Coupling caching and forwarding: Benefits, analysis, and implementation," in *Proc. of the 1st ACM SIGCOMM Conference on Information-Centric Netowrking*. ACM, sept 2014.

[8] L. Deng, Y. Gao, Y. Chen, and A. Kuzmanovic, "Pollution attacks and defenses for internet caching systems," *Comput. Netw.*, vol. 52, no. 5, pp. 935–956, Apr. 2008.

[9] M. Xie, I. Widjaja, and H. Wang, "Enhancing cache robustness for content-centric networking." in *INFOCOM*, A. G. Greenberg and K. Sohraby, Eds. IEEE, 2012, pp. 2426–2434.

[10] 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, information Centric Networking.

[11] 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.

[12] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for NS-3," NDN, Technical Report NDN-0005, October 2012. [Online]. Available: http://named-data.net/techreports.html

[13] G. Carofiglio, M. Gallo, L. Muscariello, and D. Perino, "Modeling data transfer in content-centric networking," in *Teletraffic Congress (ITC), 2011 23rd International*, Sept 2011, pp. 111–118.

[14] L. Muscariello, G. Carofiglio, and M. Gallo, "Bandwidth and storage sharing performance in information centric networking," in *Proceedings of the ACM SIGCOMM Workshop on Information-centric Networking*, ser. ICN '11. New York, NY, USA: ACM, 2011, pp. 26–31.

[15] I. Psaras, R. G. Clegg, R. Landa, W. K. Chai, and G. Pavlou, "Modelling and evaluation of ccn-caching trees," in *Proceedings of the 10th International IFIP TC 6 Conference on Networking - Volume Part I*, ser. NETWORKING'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 78–91.