An Improved Defense Method Against Interest Flooding Attacks in CCN: Postprint

Authors: Wu Xun, Ling Jie

Date: 2019-01-28T00:00:00+00:00

Abstract

Interest Flooding Attack (IFA) in Content-Centric Networking (CCN) represents a hot research topic in CCN network security. To enhance the defense capabilities of CCN against IFA, this study investigates various defense methods and proposes an improved IFA defense method for CCN. Based on the flow balance principle of CCN, the method employs malicious prefix traceback to achieve rapid detection of IFA and implements defense against IFA by improving the Additive Increase Multiplicative Decrease (AIMD) algorithm. Security analysis demonstrates that the proposed method can respond more rapidly when confronted with IFA; furthermore, compared with other IFA defense methods, it reduces the computational overhead of CCN routers in IFA detection while maintaining security.

Full Text

An Improved Defense Method for Interest Flooding Attack in CCN

Wu Xun, Ling Jie
School of Computer, Guangdong University of Technology, Guangzhou 510006, China

Abstract

Interest Flooding Attack (IFA) in Content-Centric Networking (CCN) represents a critical research focus in CCN security. To enhance CCN' s defense capabilities against IFA, this paper investigates existing defense mechanisms and proposes an improved method for defending against interest flooding attacks in CCN. Based on the CCN flow balance principle, our approach employs malicious prefix traceback to achieve rapid IFA detection and implements defense

through an improved Additive Increase Multiplicative Decrease (AIMD) algorithm. Security analysis demonstrates that this method responds more quickly when facing IFA attacks and, compared to other IFA defense methods, reduces the computational overhead for CCN routers during IFA detection while maintaining security guarantees.

Keywords: interest flooding attack; malicious prefix traceability; content-centric networking; CCN; IFA

0 Introduction

Since its inception, Content-Centric Networking (CCN) has garnered widespread attention and emerged as one of the most promising architectures for future Internet infrastructure [1]. CCN names data directly, with consumers requesting content via interest packets and servers or routers returning corresponding data packets. Each CCN router maintains at least three data structures: Content Store (CS), Pending Interest Table (PIT), and Forwarding Information Base (FIB) [2]. The CS caches received data packets, the PIT records pending interest packets and their arrival interfaces to facilitate data packet responses, and the FIB handles routing and forwarding decisions.

While the CCN router architecture mitigates traditional IP network threats such as source address spoofing and host-targeted flooding attacks, its novel structure introduces new security vulnerabilities. The most significant threat to CCN is the Interest Flooding Attack (IFA). Since routers must record the content names of interest packets not satisfied by the CS in their PIT tables, attackers can exploit this characteristic by forging numerous malicious interest packets that will never be satisfied by the CS. By exhausting router PIT space and causing legitimate user requests to be dropped, attackers can successfully execute denial-of-service attacks.

IFA Defense Methods: Research Status

In response to this threat, numerous research institutions worldwide have conducted extensive experiments and studies to address IFA in CCN. Tang Jianqiang, Zhou Huachun, et al. proposed a collaborative defense method [3] that uses the AIMD algorithm to limit the forwarding rate of interest packets with abnormal content name prefixes. Professor Alexander and his team introduced a port traffic limitation-based IFA defense approach [4], which assigns specific limits to each outgoing port. When interest packets from a port exceed this threshold, they are queued for transmission. Upon authorization to transmit, queued packets are forwarded equally from the incoming port queues. Cheng et al. [5] proposed a satisfaction rate-based feedback defense mechanism that detects IFA by monitoring interest packet satisfaction rates and then defends against attacks through interface limiting. Gasti et al. [6] suggested leveraging the CCN flow balance principle to detect and defend against IFA attacks. Dai et al. [7] introduced an interest traceback mechanism where CCN nodes con-

struct suppression messages that propagate hop-by-hop toward attack sources. Upon identifying malicious prefixes, routers forge data packets to respond to malicious interest requests, satisfying illegal PIT entries with "corresponding data packets" and ultimately tracing back to access routers to limit the admission rate of malicious interest packets at the network edge. Compagno et al. [8] proposed Poseidon, which detects IFA on different interfaces based on the ratio of received interest packets to transmitted data packets and PIT utilization rates. Cheng et al. [9] demonstrated that network self-similarity characteristics effectively describe traffic patterns and can serve as a foundation for traffic anomaly detection, using traffic anomaly information to identify IFA occurrences. Garcia-Luna-Aceves et al. [10] proposed replacing existing CCN routing strategies with CCN-GRAM (Gathering of Routes for Anonymous Messengers) to fundamentally eliminate IFA attacks. While mandatory signature verification by intermediate routers could mitigate IFA attacks to some extent, such enforcement may cause privacy issues [11] and potentially enable computation-based attacks on routers.

Addressing these concerns, Ribeiro et al. [12, 13] proposed the CCNCheck mechanism, which requires routers to perform signature verification with dynamic probabilities. Additionally, since Software-Defined Networking (SDN) controllers possess a global network view and can rapidly collect interest packet information from monitoring nodes across the network, they can promptly detect anomalous attack traffic, avoid redundant detection and excessive responses, quickly suppress attack sources, and protect legitimate users [14, 15].

Most existing defense methods involve IFA detection techniques that first identify ongoing IFA attacks and determine which content name prefixes are malicious before implementing defense based on this malicious prefix information. This paper proposes a novel IFA detection method that fully leverages network device characteristics to rapidly identify interest packets with malicious prefixes in the network. The malicious prefix information is then disseminated to other routers through traceback. Upon receiving this information, network devices defend against interest packets with malicious content name prefixes using an improved AIMD algorithm [3]. Compared to [3], our approach eliminates the need to detect PIT information and perform complex calculations to determine IFA occurrences; instead, it simply checks the marker bits of data packets returned by other routers. Furthermore, we improve the AIMD algorithm from [3] to adapt to our traceback method, using shortest prefix matching to constrain malicious prefixes within a relatively small space without imposing significant storage burdens on routers.

2.1 Malicious Prefix Traceback

When forwarding interest packets, CCN routers query the FIB based on the longest prefix matching principle to determine appropriate interfaces, then forward interest packets to selected next hops according to forwarding strategies. If no matching entry exists in the FIB for a particular interest packet' s content

name, the packet is considered erroneous and discarded. Similarly, when an interest packet reaches the corresponding content server but the requested data does not exist, the same handling applies.

According to the CCN flow balance principle, CCN' s transmission mechanism maintains an intrinsic balance between interest packets and data packets: for each interest packet sent upstream, at most one data packet is returned downstream. The primary factor disrupting this balance is the direct discarding of interest packets with erroneous content names in routing forwarding strategies. Regardless of whether these erroneous interest packets are malicious, there always exists a unique device in the network that knows about them: the device that discards them, which could be either a CCN router or a content server. Based on this CCN characteristic, this paper proposes an improved IFA defense method.

Each CCN router maintains three data structures: CS, FIB, and PIT. Upon receiving an interest packet, if neither the CS nor PIT contains relevant information, the router queries the FIB. If a matching content name prefix domain exists, the interest packet is forwarded to the next hop and a new entry is added to the PIT. If the FIB contains no matching prefix domain, the router replies with a marked data packet and discards the interest packet. When a router receives this data packet, it checks the marker bit. If it indicates a discarded interest packet, the corresponding content name is identified as abnormal. The router first deletes the PIT entry and then applies the AIMD algorithm to limit the transmission rate of interest packets with that abnormal content name. If an interest packet reaches the server but the requested file does not exist, the server similarly responds with a marked data packet.

The specific processing flows are illustrated in Figures 1 [Figure 1: see original paper] and 2 [Figure 2: see original paper].

2.2 Malicious Prefix Defense

This paper improves the AIMD algorithm from [3] for processing received marked data packets. The improvements include: (1) adding a Queue[b,Rb,t] space to record the shortest prefix of abnormal content names, its forwarding rate, and lifetime; (2) using shortest abnormal prefix matching to constrain abnormal name prefixes; and (3) adding steps for marking traceback data packets, setting their content, and detecting markers to better adapt to our malicious prefix traceback method, as shown in Algorithm 1.

The figures illustrate router processing of interest packets and data packets in the malicious prefix traceback method. When a router receives an interest packet, it first checks the CS, PIT, and FIB. If the content name exists in the CS, PIT, or FIB, the packet is forwarded normally. If a longest prefix match is found in the Queue[], the interest packet is forwarded at the rate specified in the queue entry. If no match exists in any structure, a data packet with a special

field containing the shortest invalid content name is generated and returned, and the AIMD algorithm is applied to limit future transmissions.

We briefly introduce the concept of shortest abnormal prefix. Consider a content name N1/N2/N3/N4/···If the FIB contains an entry for domain N1/N2/, the interest packet with this content name can be forwarded via longest prefix matching to the next hop recorded in entry N1/N2/. However, if the next-hop router' s FIB contains no entry for domain N1/N2/N3/, the router discards the interest packet and, according to the malicious prefix traceback strategy, returns a data packet. This data packet' s content name matches the interest packet' s name (N1/N2/N3/N4/···), while its payload contains the shortest unreachable prefix N1/N2/N3/, which constitutes the shortest abnormal prefix.

Using the AIMD algorithm, each time an attack interest packet is received, the forwarding rate for the next attack interest packet with the same content name prefix decays by e^(-C). Whenever a new marked data packet arrives, Algorithm 1 compares its content with prefixes in Queue[]. If a Queue[] entry contains this shortest malicious prefix, the algorithm constrains that entry' s content name prefix to the new shortest malicious prefix. Thus, Queue[] effectively constrains shortest malicious prefixes. For attack interest packets with randomly generated malicious prefixes, this rapidly decays their forwarding rate, providing effective IFA defense.

```
Algorithm 1: AIMD-Based Rate Limiting for Abnormal Name Interest Packets

struct {
    string countName_short;      // Shortest abnormal prefix
    unsigned double forward_v;   // Forwarding rate
    unsigned int time;           // Time since last receipt of this data
    int find_long(string)        // Find position of longest prefix containing the string
    int find_short(string)       // Find position of shortest prefix contained in the string
    updata();                    // Automatically decrement time values; delete entries with
} Queue[];

struct {
    string countName;            // Datagram content name information
    int flag;                    // Special bit: 0=interest packet, 1=normal data packet, 2=
    name_short();                // Return shortest invalid content name
} Datagram;

switch(Datagram.flag):
    if CS.find(Datagram.countName) || PIT.find(Datagram.countName) || FIB.find(Datagram.coun
        forward(Datagram);       // Forward interest packet by default
        return 0;

    if Queue.find_long(Datagram.countName):
        int i = Queue.find_long(Datagram.countName);  // Return index of longest prefix matc
```

```
    forward(Datagram, Queue[i].forward_v);        // Forward at rate forward_v
    return 0;

// If no match in CS, PIT, FIB, or Queue
int flag = 2;
data = Datagram.name_short();
datapack = make_datapack(Datagram.countName, flag, data);  // Create packet with interes

// Return a datagram recording the anomaly
delete Datagram;
return datapack;

// Also need to check if corresponding prefix exists in Queue, delete if present, and ir
Queue.updata();
return Datagram;  // Return actual data packet

Queue.updata();   // Update queue

if !Queue.find_long(Datagram.data) && !Queue.find_short(Datagram.data):
    // If neither longest nor shortest match found, add new entry
    time = T;
    forward_v = init_forward_v;
    countName_short = Datagram.data;
    Queue.insert(countName_short, forward_v, time);  // Insert new entry
else:
    if Queue.find_short(Datagram.data):
        // C is a constant > 1 (e.g., C=10), T is default time
        // If shortest match found in Queue, update entry
        int i = Queue.find_short(Datagram.data);
        Queue[i].forward_v *= e^(-C);
        Queue[i].time = T;
        Queue[i].countName_short = Datagram.data;
    else:
        // If longest prefix match found, update that entry
        int i = Queue.find_long(Datagram.data);
        Queue[i].forward_v *= e^(-C);
        Queue[i].time = T;
```

2.3 Example

Figure 3 [Figure 3: see original paper] illustrates a simple traceback defense scenario. Assume all router CS tables are empty. Router CCN-R1' s FIB contains three entries: www/youku, www/baidu, and www/bilibili. CCN-R2' s FIB has one entry: www/youku/com. CCN-R3's FIB contains www/baidu/com, and CCN-R4' s FIB contains www/bilibili/com.

When an attacker sends an interest packet with content name www/baidu/cn to CCN-R1, the router forwards it to CCN-R3 based on longest prefix matching. Since the content name www/baidu/cn has no matching entry in CCN-R3' s FIB, conventional methods would simply discard the interest packet. Our approach instead replies with a marked data packet. When CCN-R1 receives this data packet and detects the traceback marker, it adds a new entry for prefix www/baidu/cn to the Queue[] via the AIMD algorithm and limits the forwarding rate for subsequent interest packets with this prefix.

Figure 4 [Figure 4: see original paper] demonstrates how the Queue[] constrains malicious content names. Numbers 1-4 represent different malicious content names generated by an attacker's program and propagated through the network, while 5-7 show the propagation paths of marked data packets returned by routers or servers. When interest packets with content names like a/b/c/[non-d]/[any-value] are sent, they are constrained to a single entry, reducing forwarding rates as shown in router c' s Queue[]. If a Queue[] entry receives a valid data packet, that prefix is deleted and its speed v is increased by C.

3.1 Malicious Prefix Traceback Analysis

The malicious prefix traceback concept was initially proposed in [7], which introduced an interest traceback mechanism where routers, after identifying malicious prefixes, forge data packets to respond to malicious interest requests. This satisfies illegal PIT entries with "corresponding data packets" and ultimately traces back to access routers to limit malicious interest packet admission rates at the network edge. However, [7] did not specify where to deploy the strategy and required prior detection of malicious interest packets before targeted defense could be implemented.

Compared to [7], our work first clarifies that the strategy should be deployed on all network routers and content producers. Second, our method does not require detecting malicious interest packets; it can directly trace back interest packets with malicious content names and combine this with the AIMD algorithm for effective IFA defense.

3.2 AIMD Algorithm Security Analysis

This paper improves the AIMD algorithm by adding a Queue[] space. While [3] also utilized a similar space, it was not described in detail. As shown in Figure 4, we provide a detailed description of this space, which records the shortest malicious prefix to constrain the numerous different malicious prefixes generated by attackers. The number of Queue[] entries relates to network hop count. Based on the hop limit in TCP/IP' s RIP protocol, real-world network hops remain within bounded values, so the required space does not impose significant storage overhead on routers.

Our method ensures security compared to [3]. The IFA defense strategy primar-

ily relies on the AIMD algorithm. Although we improved the algorithm, the core mechanism for limiting malicious prefix forwarding rates remains largely unchanged. Reference [3] proved that this rate-limiting approach for malicious prefix interest packets provides effective defense. Therefore, our improved IFA defense method guarantees security.

3.3 Efficiency Analysis

Our method achieves faster reaction speeds against IFA compared to [3]. The approach in [3] requires monitoring PIT utilization and interest packet satisfaction rates, waiting for PIT space to fill or approach capacity and for numerous interest packets to go unsatisfied before algorithmically determining an IFA attack.

This process primarily depends on two parameters: PIT utilization rate and interest packet satisfaction rate. Let $t_1, t_2, ..., t_n$ represent time points, with $I(t)$ and $D(t)$ denoting the numbers of interest packets and data packets arriving at the router during time period t, respectively. The historical average numbers can be expressed as:

$$\bar{I}(t_n) = (1-\alpha)\bar{I}(t_{n-1}) + \alpha I(t_n)$$

$$\bar{D}(t_n) = (1-\alpha)\bar{D}(t_{n-1}) + \alpha D(t_n)$$

where α is an inertia coefficient ($0 < \alpha < 1$) representing the sensitivity of long-term average data to current data quantities. The interest packet satisfaction rate at time t_n is:

$$S(t_n) = \bar{D}(t_n)/\bar{I}(t_n)$$

Since one interest packet can only be satisfied by one data packet in CCN, the satisfaction rate should normally be 100%. When PIT utilization exceeds a threshold, it indicates the PIT interest packet count has reached a warning level. When the satisfaction rate falls below a threshold, it signals an anomaly in the one-to-one correspondence between interest and data packets. These two metrics determine IFA occurrence.

Our method can rate-limit malicious prefix interest packets within a single round-trip time without the above computational process, simply waiting for the next node to return malicious content names. Since PIT entry expiration times far exceed average network round-trip times, the method in [3] cannot react within one round-trip time when facing IFA. Therefore, our approach offers advantages in IFA detection speed.

Table 1 compares our method with other IFA defense approaches, detailing their operational and deployment characteristics.

Notably, our method informs the entire network of malicious content names through marked traceback data packets. As most IFA defense methods require malicious content name information (as shown in the table), our approach can effectively cooperate with other IFA defense technologies. This collaboration enables other IFA defenses to operate without relying on IFA detection modules, improving their efficiency in detecting IFA.

For example, combining our method with collaborative defense strategies can rapidly disseminate new malicious prefix information to other routers by adding a collaborative defense packet module. Since our method only defends against malicious interest packets on the paths they traverse, this combination effectively enhances network-wide defense capabilities. When integrated with SDN, traceback data packets arriving at monitoring routers can forward malicious content name prefixes to the central controller, saving the time traditionally required for malicious content name detection.

4 Conclusion

The collaborative defense strategy in [3] effectively detects IFA by monitoring PIT utilization and interest packet satisfaction rates, then defends against detected malicious prefixes using the AIMD algorithm. This method offers excellent IFA defense capabilities for CCN. However, it has limitations in IFA detection. According to the CCN flow balance principle, devices in the network inherently possess knowledge of malicious prefix information, making it unnecessary to detect PIT utilization and satisfaction rates to obtain malicious interest packet prefixes.

Our approach leverages these knowledgeable devices to trace back malicious prefixes, enabling defense against malicious prefixes within a single round-trip time. Compared to [3], our method improves reaction speed against IFA attacks. Additionally, by adapting the AIMD algorithm to the new detection technique without altering its core mechanism, we maintain security guarantees. In summary, our method enhances reaction speed and reduces computational overhead for CCN routers during IFA detection while preserving the security assurances of [3].

References

[1] Li Y, Xin Y, Han Y, et al. Overview of DoS attacks in content center networks [J]. Journal of Information Security, 2017, 2(1): 91-108.

[2] Chen Z, Cao J, Yin H. Content center network architecture [M]. Beijing: Tsinghua University Press, 2014.

[3] Tang J, Zhou H, Liu Y, et al. Mitigating interest flooding attack based on prefix identification in content-centric networking [J]. Journal of Electronics & Information Technology, 2014, 36(7): 1735-1742.

[4] Afanasyev A, Mahadevan P, Moiseenko I, et al. Interest flooding attack and countermeasures in Named Data Networking [C]//Proc of IFIP Networking Conference. Piscataway, NJ: IEEE Press, 2013: 1-9.

[5] Cheng Y, Afanasyev A, Moiseenko I, et al. A case for stateful forwarding plane [J]. Computer Communications, 2013, 36(7): 779-791.

[6] Gasti P, Tsudik G, Uzun E, et al. DoS and DDoS in Named Data Networking [C]//Proc of the 22nd International Conference on Computer Communications and Networks. Piscataway, NJ: IEEE Press, 2012: 1-7.

[7] Dai H, Wang Y, Fan J, et al. Mitigate DDoS attacks in NDN by interest traceback [C]//Proc of Computer Communications Workshops. Piscataway, NJ: IEEE Press, 2014: 381-386.

[8] Compagno A, Conti M, Gasti P, et al. Poseidon: mitigating interest flooding DDoS attacks in named data networking [C]//Local Computer Networks. Piscataway, NJ: IEEE Press, 2013: 630-638.

[9] Cheng X, Xie K, Wang D. Network traffic anomaly detection based on self-similarity using hht and wavelet transform [C]//Proc of International Conference on Information Assurance and Security. Piscataway, NJ: IEEE Press, 2009: 710-713.

[10] Garcia-Luna-Aceves J J, Barijough M M. Content-centric networking using anonymous datagrams [C]//Proc of IFIP Networking Conference. Piscataway, NJ: IEEE Press, 2016: 171-179.

[11] Lauinger T. Security & scalability of content-centric networking [D]. Darmstadt: Technische Universität Darmstadt, 2010.

[12] Ribeiro I, Rocha A, Albuquerque C, et al. On the possibility of mitigating content pollution in Content-Centric Networking [C]//Local Computer Networks. Piscataway, NJ: IEEE Press, 2014: 498-501.

[13] Ribeiro I, Rocha A, Albuquerque C, et al. Content pollution mitigation for Content-Centric Networking [C]//Network of the Future. Piscataway, NJ: IEEE Press, 2017: 1-5.

[14] Salah H, Wulfheide J, Strufe T. Coordination supports security: a new defence mechanism against interest flooding in NDN [C]//Local Computer Networks. Piscataway, NJ: IEEE Press, 2016: 73-81.

[15] Salah H, Wulfheide J, Strufe T. Lightweight coordinated defence against interest flooding attacks in NDN [C]//Proc of IEEE Conference on Computer Communications Workshops. Piscataway, NJ: IEEE Press, 2015: 103-104.

Note: Figure translations are in progress. See original paper for figures.