S. Felix Wu, W. Huang
UCDavis
Dan Massey, Allison Mankin
USC/ISI
C.L. Wu, X.L. Zhao
NCSU
Lixia Zhang
UCLA

Intention-Driven ICMP Trace-Back

draft-wu-itrace-intention-02.txt

Status of this memo

Abstract

This draft describe an enhancement over the current iTrace proposal such that we can trace more closely to the DDoS

slaves faster.

1 Introduction

The probability of iTrace message generation on a particular router in the current internet draft is static and small (about 1 over 20,000 packets) such that the overhead introduced by the iTrace messages is small. However, if each DDoS slave produces a relatively small amount of attack traffic, then, it might take a long time for a nearby

router to generate a valuable iTrace packet. Statistically, routers closer to the victim will generate "useful" iTrace messages toward the true victim faster than the routers closer to the true slaves.

For example, we have one DDoS victim, X and R is one of the router forwarding DDoS traffic toward X. When the router R generates an iTrace message, the iTrace probability 1/20,000 is for all the packets being sent through e[x]. If the packet rate for e[x] is 10,000 packets per second, then, statistically one useful iTrace packet will be generated in about 2 seconds. However, if the packet rate is 100 packets per second, then the time will be 200 seconds. If a slave (attacking X) is in the campus of UCDavis, then routers closer to X will statistically generate an iTrace message toward X right after the attack. On the other hand, it will take maybe a few minutes before the victim will see the first iTrace message from the border router of UCDavis.

A related problem to the above scenario is that many "useless" iTrace messages might be generated. While it might not be a big problem for a non-victim to receive iTrace messages, resources (CPU cycles, for example) are wasted and the tracing activities toward the true victim are delayed.

We propose a mechanism, "intention-driven" iTrace, to enhance the iTrace performance. Our objectives are:

- o With a specified probability, the resources for generating iTrace messages will be spent, more likely, on packets toward DDoS victims. In other words, a selected set of destinations will have more than 1/20,000 probability to get iTrace messages.

- The total number of iTrace messages generated by each router remains the same. I.e., statistically still 1 over 20,000.

- The new mechanism is compatible with the current iTrace scheme such that we do not require every router to support the new mechanism.

- This new mechanism must be scaleable and simple to implement.

2 Intention-Driven iTrace:

2.1 **Prob(iiT-Control):** Probabilistic Control of Invoking "Intention" iTrace

For each router, a probability Prob(iiT-Control) (probability for intention iTrace) is given. Every time, when a packet is selected, through the standard iTrace scheme (1/20,000 probability), we will flip another random coin, controlled by Prob(iiT-Control), to determine

whether this iTrace message should be sent "as it is" or we should invoke the "intention" iTrace mechanism.

For example, if Prob(iiT-Control) is 0.5, then 50% of the iTrace messages will be used to handle the "original" iTrace scheme, while the rest will be used for only those who really want to receive iTrace messages. While we dedicate 50essentially reduce the 1/20,000 iTrace probability to 1/40,000 statistically for all the traffic.

2.2 iTrace Intention Bit (iiB) and iTrace Execution Bit (ieB)

A router conceptually has two tables: routing information table and packet forwarding table. We propose to associate each routing entry with one extra bit called "iTrace intention bit (iiB)", and for each forwarding entry, we introduce a new bit called "iTrace execution bit (ieB)". In a routing table, more than one entries might have "iiB" on, while, in a forwarding table, normally at most one entry can have "ieB" on. However, if the ieB of a particular forwarding table entry is on but no data packets use this entry before the next iTrace trigger, then it is possible to have multiple one's in the forwarding table as well.

The "iTrace intention bit (iiB)" will be distributed via routing information protocols such as BGP. If the iiB for a particular route entry is 1, then the network destination under this route entry indicates its desire in receiving iTrace messages. For instance, some of them might be currently under serious DDoS attacks and they have running applications that will utilize the information being carried by iTrace messages. On the other hand, if the iTrace intention bit is 0, then it indicates that either the destination's intrusion detection system does not believe that its network is under DDoS attacks or it believes that iTrace messages would not help to handle the attacks it observed. For instance, if an intrusion detection system detects that its network is under reflective DDoS attacks, then the normal (so called) "forward" iTrace messages will not help because iTrace messages will only lead to a hugh number of reflectors, but not to the real DDoS slaves.

The "iTrace execution bit (ieB)" indicates that the very next data packet going through that forwarding entry must be iTraced. And, usually, right after this happens, this ieB bit will be cleared. Please note that the iTrace execution bit might be only conceptually associated with the forwarding table. In implementation, this extra bit might be completely outside of the packet forwarding process.

3 Packet Forwarding for Intention iTrace

In the original iTrace proposal, an iTrace message will be generated with a small probability. When an iTrace message is about to generate

but the Prob(iiT-Control) control determines to invoke intention iTrace, instead of sending a normal iTrace message, an "iTrace trigger" will be generated. The frequency of iTrace trigger can happen at most 1/20,000 (i.e., if Prob(iiT-Control) is 1.0).

Under the intention iTrace, we separate the implementation into two parts: the processing for each incoming data packet and the processing for each iTrace trigger. Please note that it is desirable to have a very efficient per-data packet process, while it is more tolerable to spend more processing time for per-iTrace trigger processing.

3.1 Per-Data Packet Processing:

When an IP packet arrives, the router will first decide which forwarding entry should be used to forward this packet. If the iTrace execution bit for this route entry is 1, then an iTrace message is generated toward this particular packet. After the iTrace message is sent, the ieB bit for this entry will be set to 0 again. The following is the pseudo code (in C) for this part of processing:

```
int
perDataPacketProcess
(IP *p)
{
         ForwardEntry *fe = findForwardEntry(p->destinationIPaddr);
         if (fe == NULL) return -1;
         if (fe->ieB == 1)
         {
                  sendiTrace(p);
                  fe->ieB = 0;
         }
         regularPacketForwarding(p, fe);
         return 0;
}
```

3.2 Per-iTrace-Trigger Processing:

When an iTrace trigger (i.e., a packet has been chosen for the regular iTrace, but the Prob(iiT_Control) control determines to invoke intention iTrace) occurs, instead of directly sending an iTrace message, the router will randomly choose one entry among all the "route" entries (not "forwarding" entries) with the "iTrace intention bit (iiB)" on. This random selection process can be as simple as a random number generation, or a round-robin fashion of selection (maybe based on traffic distribution) to enhance fairness.

The result of the selection process is a particular route entry with iiB on. Then, the corresponding forwarding entry will have its ieB

set on. The following pseudo code is an "example" of how the selection "might" work:

```
int
periTraceTriggerProcess
(RouteEntry *RETable)
{
        int i, iiB_count;
        int iTr_rand;


        iiB_count = 0;
        for(i=0; i<N; i++)
                if (RETable[i].iiB == 1) iiB_count++;


        if (iib_count == 0)
        {
                invokeRegulariTrace();
                return 0;
        }


        iTr_rand = rand() % iiB_count;
        for(i=0; i<N; i++)
        {
                RouteEntry *re = &(RETable[i]);
                if (re->iiB == 1)
                {
                        if (iTr_rand == 0)
                        {
                                re->fe->ieB = 1;
                                return 1;
                        }
                        iTr_rand--;
                }
```

```
         }
         return -1;
}
```

3.3 The iTrace probability attribute

Instead of one constant probability, Prob(iiT-Normal), (e.g., 1/20000) in the normal iTrace, two different probability values are defined for intention driven iTrace: Prob(iiT-Intention) and Prob(iiT-Other). To "approximately" obtain these values, we propose the following method:

1. Assume that the traffic ratio for iiT-intention and iiT-other is Ratio(intention) and Ratio(other), respectively. Ratio(intention) + Ratio(other) = 1.0.

2. Prob(iiT-Other) = Prob(iiT-Normal) * (1 - Prob(iiT-Control))

3. Prob(iiT-Intention) = (Prob(iiT-Noraml) - Ratio(other) * Prob(iiT-Other)) /Ratio(intention)

4 How to Distribute the iTrace Intention Value?

In this section, we present a way to distribute the Intention value for each router and for each route entry. We propose to have two new BGP community string values for the iTrace intention.

BGP community string, an existing BGP attribute, is a 32 bit unsigned integer. We would like to use one value to signal the positive intention to receive iTrace messages. In other words, including this intention commuity string means 1 for iiB. Only the originating AS may set the intention community string. The originating AS MUST apply a local dampening rule to limit the frequency of changes in the intention community string.

When a BGP router advertises the reachability of some network address, it will also attach the iTrace intention bit for that network. Therefore, when this BGP update is received by some downstream BGP routers, the intention value of the corresponding route entry will be updated the attached iiB. Furthermore, when BGP updates are aggregated, the iiB will also be aggregated.

Therefore, when a particular network is under DDoS attack, the intrusion detection system will inform the BGP router to boost up its iTrace receiving intention. This new iTrace intention bit will be distributed through the whole internet using BGP (or piggyback on BGP updates).

The exact value of the new community attribute values will be given in the later version of this draft.

5 Evaluation:

The proposed scheme will enhance the probability of receiving iTrace messages closer to the DDoS slaves.

The new extension is fairly simple to implement (as shown in the pseudo code) and the per-data packet processing overhead is reasonably small - one comparison operation. The memory cost for ieB might be a concern as we conceptually need one bit for each forwarding entry. However, we potentially can implement the same scheme without the requirement of adding one bit to the forwarding table. For example, we can choose a packet to iTrace among a pool of logged packets. The per-iTrace trigger process part is more expensive then the original iTrace proposal. While the overhead here mainly involves a random

number generation (hardware can certainly speed up), this process will happen very rarely (e.g., 1 in 20,000 packets at most).

Our proposal to distribute the intention values requires a small change to BGP. Furthermore, because of the nature of community string, it is not necessary to have all the routers supporting the new feature. I.e., even if we have sparsely a few new BGP routers supporting intention iTrace, these routers can still utilize the intention bits, while others traditional routers will still pass the intention bits around, Because of BGP route aggregation, our proposal will not increase the number of entries in the routing or forwarding tables.

6 Security Consideration:

Since our scheme will introduce exactly the same amount of iTrace messages as the original iTrace proposal, our proposal will not introduce any new vulnerability related to denial of service attacks based on the iTrace messages themselves.

Since we propose using BGP to distribute the intention values, our scheme is subject to the same security risks as BGP. The risks with respect to intention values would be that an attacker who can tamper with the BGP contents could modify the behavior of itrace to divert itrace away from the attacker's location.

With the P_IIT control, destinations without the intention iTrace support can still receive iTrace messages but with a smaller probability such as 1 over 40,000. A malicious destination can distribute positive intention bits to attract more intention iTrace messages, but it still need to compete probabilistically against other DDoS victim over about 50

7 Intellectual Property

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the tech- nology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to iden- tify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11.

Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Sec- retariat.

The IETF invites any interested party to bring to its atten- tion any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the infor- mation to the IETF Executive Director.

8 References

[iTrace] S. Bellovin, M. Leech, "ICMP Trace-Back", Internet-Draft, July 2001.

9 Acknowledgements

Our research is sponsored by DARPA under the fault tolerant networking program.

10 Author Information

S. Felix Wu <wu@cs.ucdavis.edu>
Computer Science Department
University of California at Davis
One Shields Avenue
Davis, CA 95616
phone: +1 530 754 7070

Wayne Huang <ywhuang@ucdavis.edu>

Computer Science Department
University of California at Davis
One Shields Avenue
Davis, CA 95616

Dan Massey <masseyd@isi.edu>
USC/ISI
3811 **N. Fairfax Drive, Suite 200**
Arlington VA 22203

Allison Mankin <mankin@isi.edu>
USC/ISI
3811 **N. Fairfax Drive, Suite 200**
Arlington VA 22203

Chien-Lung Wu <cwu4@unity.ncsu.edu>
NCSU
Box 7550, NCSU Centennial Campus
Raleigh, NC 27695

Xiaoliang Zhao <xzhao@unity.ncsu.edu>
NCSU
Box 7550, NCSU Centennial Campus
Raleigh, NC 27695

Lixia Zhang <lixia@cs.ucla.edu>
Computer Science Department
UCLA
Los Angeles, CA

Full Copyright Statement

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."