Malicious Dropping Attack in the Internet: Impacts and Solution

Lata L. Ragha
Ramrao Adik Institute of Technology, Department of Computer Engineering, Nerul, Navi Mumbai, India

B. B. Bhaumik
Jadavpur University, Department of CSE, Kolkata, India

S. K. Mukhopadhyay
CA-30, Salt Lake, Sector-1, Kolkata-700064, India

ABSTRACT
The unprecedented growth of the Internet over the last years, and the expectation of an even faster increase in the numbers of users and networked systems, suggest that in the near future the Internet may become the single integrated communication media. However, as the dependence on the networking infrastructure grows, its security becomes a major concern, in light of the increased attempt to compromise the infrastructure. In particular, the routing operation is a highly visible target that must be shielded against a wide range of attacks. The current interdomain routing protocol, the Border Gateway Protocol (BGP), is limited in implementations of universal security. Because of this, it is vulnerable to many attacks at the Autonomous System (AS) to AS routing infrastructure. Initially, the major concern about BGP security is that malicious BGP routers can arbitrarily falsify BGP routing messages and spread incorrect routing information. Recently, some authors have pointed out another kind of attack, called malicious dropping attack that has not studied before. The malicious draping attack can result in data traffic being blackholed or trapped in a loop. However, the authors did not elaborate on how one can detect such attacks. In this paper, we discuss and analyse a method that can be used to detect malicious dropping attacks in the Internet.

In this paper, we describe the formatting guidelines for IJCA Journal Submission.

Keywords
AS, BGP, Malicious Dropping Attacks, Monitoring point, instability.

1. INTRODUCTION
Border Gateway Protocol (BGP) is the de facto interdomain routing protocol [1]. Current Internet consists of many Autonomous Systems (ASes) connected by interdomain (inter-AS) links. Each AS is a set of routers that have the same routing policy within a single administrative domain. BGP is responsible for discovery and maintenance of paths between ASes in the Internet. BGP routers exchange routing information via two types of UPDATE messages: namely route withdrawal and route announcement. When a BGP router receives an UPDATE from its neighboring BGP router, this message will be processed, stored, and redistributed in accordance with both BGP specification and the routing policies of the local AS.

Previously, the major concern about BGP security is the authenticity and integrity of BGP UPDATES, especially route origin information and AS path information stored in the AS_PATH attribute. Incorrect UPDATES, due to either BGP router misconfiguration or malicious attack, may cause serious problems to the global Internet. Some countermeasures have been proposed to mitigate BGP vulnerabilities. To protect BGP session from spoofed BGP UPDATES sent by outsiders, TCP MD5 signature [2] using shared secret key between two BGP routers was proposed. S-BGP[3] and SoBGP[4] apply cryptography to prevent an attacker (either insider or outsider) from advertising faulty BGP messages or tampering normal messages. However, as noted in [5], [6], [7], cryptography-based security mechanisms, cannot protect routing protocols against some kind of attacks. In [7], the authors describe one such attack, namely the malicious or selective dropping attack, which can cause data traffic blackhole and persistent traffic loop. However, the authors do not present any technique to detect such attacks.

In this paper, we propose a very simple method used for detecting the malicious dropping attacks in the Internet. This scheme is based on the algorithms used to detect source of instability when the link failure occurs in the Internet. Monitoring point is considered to collect BGP update messages from the routers and this route change information is used to pinpoint the culprit ASes where the instabilities have originated. Once the source of instability is identified, the stable route database will check to see if a malicious dropping attack is embedded within this burst of BGP updates. If an attack is suspected, then an Alarm message will be flooded (with limited scope) across the BGP routers in the Internet.

This paper is organised as follows: Section II explains the concept of malicious dropping attack as described in [7]. Section III covers about the impacts of malicious dropping attack. Section IV describes the proposed methodology for detecting malicious dropping attack. Section V includes simulation results for finding damage costs of the link with and without malicious dropping attack. Finally Section VI concludes the paper.

2. MALICIOUS DROPPING ATTACK
BGP is a policy routing, path vector protocol. According to the inbound and outbound policies, BGP router may legitimately suppress some UPDATES. The authors in [7] define two consistency properties for correct BGP operation. In

this model the notations: **peer(*u*)** denotes the set of peers for node (AS) *u*, **rib-in (*u*<=*w*)** denote node *u*'s most recently received message from peer *w*, **rib(*u*)** denotes the best path that *u* adopts and stores in the local-RIB, **rib-**out(*u*=>*w*) denotes the route that *u* advertises to *w*.

The properties defined by the authors in [7] are duplicated below:

1. a) If rib(u) ≠ ε, there must ∃v∈ peers(u) [rib-in(u<=v)=rib(u)].

   b) If rib(u) =ε , rib-in(u<=v) can be arbitrary.

2. a) For any w ∈ peers(u), if rib-out(u=>w) ≠ ε, then rib-out(u=>w) = u o rib(u).

   b) It is possible that when rib(u) ≠ ε , there exists rib-out(u=>w) = ε where w ∈ peers(u).

The two properties listed above are legitimate properties that allow a BGP router at node *u* to drop BGP UPDATES. Property 1(a) implies node *u* can select a route from one peer but drop the routes it received from the others. Property 1(b) indicates that node *u* does not have to use the route announced by node *v* to reach a particular destination even though node *u* has no route. Property 2(a) guarantees that no policy allows node *u* to use one route but announce the other route to its peers. Property 2(b) indicates a policy to authorize node *u* not to transit the traffic for node *v* even though node *u* can reach a particular destination.

Any BGP dropping that is not consistent to these two properties will be classified as malicious dropping and this can result in traffic blackhole and persistent traffic loop.

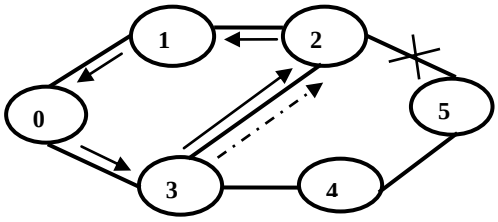

Figure 1. Example Network to show the impacts of malicious dropping attack.

3. IMPACTS OF MALICIOUS DROPPING ATTACK

In the Internet the best path is selected by the BGP to transmit the data from source to destination. The malicious router has to be selected for some prefixes on the default path to show the possible routing problems caused by Malicious dropping attack. The impacts of malicious dropping attack are: a traffic blackhole, sub-optimal routing and persistent traffic loop.

3.1 Blackhole

In the example network shown in Figure 1, we assume that each node represents an AS, there is a BGP router associated with each AS.

The target network is owned by AS5. In the initial stable state, R0 uses (AS3, AS2, AS5)as the best AS path to reach the target network. R0 uses R3 as the next hop. When the link between R3 and R5 is cut, under normal circumstances, R0 will remove (AS3, AS2, AS5) from the BGP routing table and select (AS3, AS4, AS5) the best path. This path will be announced to AS1, which will use the path (AS0, AS3, AS4, AS5). In the forwarding table, for the entry of the particular network of AS5, R0 sets R3 and R3 sets R4 as the next hop, R2 and R3 set R0 as the next hop.

However, if R3 is malicious, it can hijack the normal traffic to the target network by malicious dropping attack. In this example, we let R2 hold the withdrawal messages to R3 and only send a withdrawal message to R1. Consequently, although R0 receives the route withdrawal from R1, it will still use R3 then R2 as next hop to deliver traffic to the network of AS5. Therefore, all the traffic from AS0 will be blackholed by R2. That is indicated by dotted arrow in figure 1.

If the malicious router drops the incoming withdrawal or outgoing withdrawal messages for a particular prefix, then a blackhole for that traffic may be formed. Because as long as the downstream AS is not informed that its best path is not valid any more, the downstream AS will continue to deliver data traffic along its "best" path to the malicious router. Consequently, all the packets may be dropped by the malicious router and network bandwidth has been wasted. Even though it is possible for the downstream router to receive withdrawal messages from other routers, those withdrawals will not cause the downstream router to remove or replace its "best" path.

3.2 Traffic Loop

If the router drops incoming routes, either announcement or withdrawal, persistent packet forwarding loop can occur. The same example network is used to explain about the persistent routing loop. The target network is still considered in AS5. The major difference from the first example network is that R2 is a normal node whereas R3 is malicious. R3 selectively drops outgoing route announcements to R0 in the beginning. In the example, R3 sets (AS2, AS5) as the best path, yet drops the route update to R0. It announces (AS3, AS4, AS5) to R0 instead of announcing the current route stored in the Loc-RIB. R2 announces (AS2, AS5) to R1. R1 sets a larger local preference value to the route learned from R0 than the route learned from R2 so that R1 uses (AS0, AS3, AS4, AS5) as the best AS path. Initially, in the stable state, every router chooses the correct next hop for the destination network.

Same as the first example, we cut the link between R2 and R5. Consequently, R2 sends route withdrawal to R1 and R3. R3 maliciously drops this incoming message and still uses the route (AS3, AS2, AS5). When R1 receives the withdrawal message, R1 uses the route (AS0, AS3, AS4, AS5) and sends this route back to R2. Finally, R2 uses the route (AS1, AS0, AS3, AS4, AS5). From the routing policies, we can see that the

loop has been formed. For the route to AS5, R2 sets R1 as next hop, R1 sets R0 as next hop, R0 sets R3 as next hop and finally R3 sets R2 as next hop.

3.3 Suboptimal Routing

If the malicious router drops the outgoing route announcement, the attack may cause the sub-optimal routing. Because the routing conditions have changed, the downstream routers need to reevaluate all possible paths and select the best one for optimal routing. However, by removing such routing signals, dropping updates will cause downstream routers keep using the previous path which may not be the best path anymore.

4. SOLUTIONS FOR MALICIOUS DROPPING ATTACK

In this section, we give an overview of the Algorithm that we propose to detect and mitigate against BGP malicious dropping attacks. These attacks can be induced by the malicious router while transmitting withdrawal messages (with link fails) in the internet.

A BGP instability is an event that impacts inter-AS routing based on link/node failures/additions may cause changes in BGP session availability. We consider the EBGP update message as a consequence of some instability. That is, in response to BGP instability, a BGP speaking router initiates a BGP update that propagates an attribute change from one BGP peer to another. When the link fails in the internet, the malicious router may not propagate this withdrawal message to all its peering ASes can cause malicious dropping attack. If we find instabilities and source of instability (failed link), we can easily detect the malicious router.

We can identify faulty link by considering either single or multiple monitoring points. Monitoring point is any Node in the network which can be able to collect all update information from a number of ASes. Projects such as University of Oregon's Route Views [8] and RIPE [9] collect BGP updates from a number of Autonomous Systems.

4.1 Identifying Faults Using a Single monitoring Point

The monitoring points set up peering sessions with collaborating routers and passively collect all the updates generated by the routers. To the router being monitored, the monitoring point appears to be simply another BGP peer router. The monitoring point logs update data and do not advertise any paths to other ASs.

We first consider only the view from a single monitoring point, *M*, and provide an algorithm that gives *a possible* explanation for the routing changes observed at *M*. More formally, we are given two shortest path trees rooted at *M*: $T_0 = (V_0, E_0)$ is the shortest path tree at time t_0 and $T_1 = (V_1, E_1)$ is the shortest path tree at time t1. Both T_0 and T_1 were computed in some unspecified graph G = (V, E) and $T_0 \# T_1$ if, some link(s) in graph must have failed (or recovered). Our objective is to identify some scenario of failed (and/or recovered) links that explain the change from T_0 to T_1. Although there may be many possible explanations for the routing changes observed by, we first seek the simplest possible explanation. In the best case, a large number of route changes can be caused by a single link failure or recovery. We have considered the case of single-fault scenario.

Algoritm *FindChange()* takes tree $T_0 = (V_0, E_0)$ and $T_1 = (V_1, E_1)$ as input and labels each edge in $E_0 \cup E_1$ as either *unchanged*, *vanished*, or *appeared*. A *vanished* link is present in T_0 , but not present in T_1. This can occur due the failure of the link or the link may *v*anish as a consequence of some other

Algorithm 1: FindChange(T_0, T_1)

Input: $T_0 = (V_0, E_0)$ //Shortest path tree from M at t_0;
$T_1 = (V_1, E_1)$ //Shortest path tree from M at t_1;
Output: Marked edges: *unchanged*, *vanished*, *failed*, *added*;

```
Let V = V0 U V1;   Eadd = Efail = E = E0 U E1;
Let Tadd = Tfail = SPT (V, E);
for each e ∈ E in BFS order do
 {  if e ∈ E0 ∩ E1 then
       e = unchanged;
    else e ∈ E0 then
       e = vanished;
       if e ∈ Tfail then
          { e = failed;
            Efail = Efail - e;
            Tfail = SPT (V, Efail);
          }
    else e ∈ E1 then
       { e = appeared;
       if e ∈ Tadd then
        { e = added;
           Eadd = Eadd - e;
           Tadd = SPT (V, Eadd);
         }
       }
```

Algorithm 2: FindPath(T_0, T_1);

```
Let e = (u,x) be the only edge marked failed in
                FindChange(T0, T1);
Initialize P = {e};
Let p = x;
if tag(e) == vanished then
{   while p has only one outgoing edge(p, q) ∈ E0 - E1
do
      {  Add (p, q) to P;
         Set tag (p, q) = failed;
         Set p = q;
       }
}
else if tag(e)==appeared then
{   while p has only one outgoing edge(p, q) ∈ E1 - E0
do
      {  Add (p, q) to P;
         Set tag (p, q) = added;
         Set p = q;
       }
```

change. *Findpath()* starts from failed or added path and moves down the tree until all edges that could have failed are marked *failed. Findpath()* returns the entire set of links that constitute this path. The failure of any of these links alone would explain the path changes from T0 to T1.

In the proposed method shown in figure 3, we have used the algorithms FindChange( ) and FindPath( ) described in [10] shown as Algorithm 1 and Algorithm 2 to get the list of failed edges for every instable path. The notations used in the proposed method are described below:

instability even: path change occurred;
r_p *:* Previous route
r_n *:* New route
Ed_0 : links between ASes at time t0
Ed_1 : links between ASes at time t1
Vt_0 : set of ASes in the path at time t0
Vt_1 : set of ASes in the path at time t1

Monitoring point will get all the routing updates from different routers. Based on changes in routing tables of each router graphs will be generated for both new and old routes. Then labelled graph is generated by using Algorithm 1 and Algorithm 2. Finally the intersection of failed paths from every prefix (where old paths are replaced by new paths) results a single failed link which is the source of instability.

```
// an algorithm used to find instability
At Monitoring point M,
 flag = true;
for every eBGP router's table
    Ed0 = Ed1 = Vt0 = Vt1 = null;
      for each instability event
            if route changes from rp to rn
                { Ed0 = list of links in rp
                  Ed1 = list of links in rn
                  Vt0 = list of routers in rp
                  Vt1 = list of routers in rn
                  E0 = E0 U Ed0;   E1=E1 U Ed1;
                  V0 = V0 U Vt0;   V1=V1 U Vt1;
                }
       T0 = (V0, E0);  T1 = (V1, E1);
       FindChange(T0,T1);
       Let CF be the set of edges marked as failed
       if flag
         { CF = FindPath (T0,T1);
           flag = false;      }
      CF = CF ∩ FindPath (T0,T1);
```

Figure 3. Algorithm to detect source of instability

```
Detect_Dropping(u,t)//u is a cluster of updates at time period t
{    // this is used to monitor AS to report instability and dropper
   instability=Locate(u); // running locating algorithm try to
                          // find instability
   if (instability == NULL) // if not found
       {   error("can't find instability!");
           return;
       }
    else  report instability, t; // if instability found
     dropper=Check_RT(instability, current_node);
     if (dropper != NULL)
       { report dropper, t;
         return;
        }
   return;   }
```

Figure 4. Algorithm to detect Malicious Router

Then this monitoring router will check all the routing tables of the routers, those are under observation, to see if the troubled inter AS link is used in any best-path route. The module returns a true (indicating that there is a possible malicious dropping attack). Otherwise, the detection module returns a false (no indication of malicious dropping attack). The pseudo code of the malicious dropping attack detection is shown in figure 4.

We explain our attack detection method using an example based on the network topology shown in Figure 5. Assume that link 6-4 is broken and this results in a burst of BGP update messages. If there is no malicious dropping attack, all the nodes can locate the source of instability. However, if the link 6-4 is broken and at the same time, the router in AS-4 launches a malicious dropping attack towards AS-2, then ASes 2, 1, 11, 12, cannot locate the source of instability.

Upon receiving the information from monitoring point, AS2 was able to check its routing table and discover a discrepancy about the status of the inter-AS link 2-4. Thus, AS2 was able to detect a potential malicious dropping attack. AS2 can then issue an Alarm message which is propagated with limited scope across the network. The Alarm message contains information about the identifier of the malicious router and any suspected broken links that are not reported. Such Alarm messages are authenticated so that attackers will not issue forged Alarm messages to confuse neighbors. BGP router identity authentication approach proposed in SBGP [3] can be used for authenticating Alarm messages. In the example described earlier, without using the proposed method many AS routes will use an AS path that goes through the broken link as a result of the malicious dropping attack. With the proposed method, few of the routes will use an AS path that goes through the broken link if no Alarm message is issued. If the Alarm message is flooded across the whole network, then no AS will utilize a path that goes through the broken inter-AS link and hence the damaged cost is reduced to zero.

5. EXPERIMENTAL EVALUATIONS USING NS2 SIMULATOR

We conducted experiments using NS2-BGP simulator to evaluate the effectiveness of our proposed algorithm. In this experiment we have consider only one prefix per AS and only one link failure at a time.

The flat AS topology is generated. For this topology, we introduce 3 inter-AS link failures; 4-6, 4-8, 6-7 nearer to malicious router with and without malicious dropping attacks. For each link failure, we evaluated the damage cost. The damage cost is the ratio of the number of stable paths that utilize the broken inter-AS link to the number of total stable-paths. When the failed link is away from the malicious router, the observed damage cost is very less. The monitoring point is considered to get the routing updates from all the routers. Based on the results of our experiments, we are able to find the malicious Router-4. Routers 1, 2, 11 and 12 will generate Alarm messages and flooded across the whole network so that not a single best path will utilize the failed link as well as Router-4 will be isolated from the network.

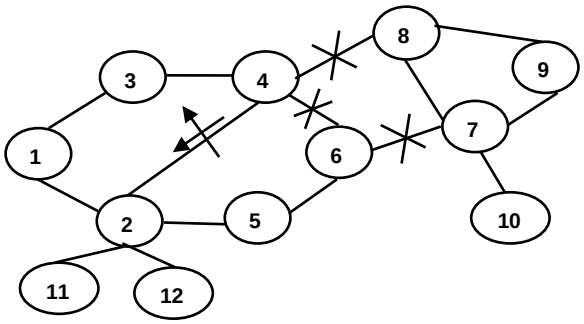

Figure 5. Example network for analysis

Our results with the malicious (selective) dropping attacks are shown below in figure 6 with bar graph. With malicious dropping attack, the damaged cost without deploying the proposed method for these three instabilities range from 6.1 to 9.1%. In the proposed method without Alarm message is deployed, then the damage cost reduces (range from 3.03 to 4.55%). With the additional Alarm messages, the damage cost is reduced to 0.

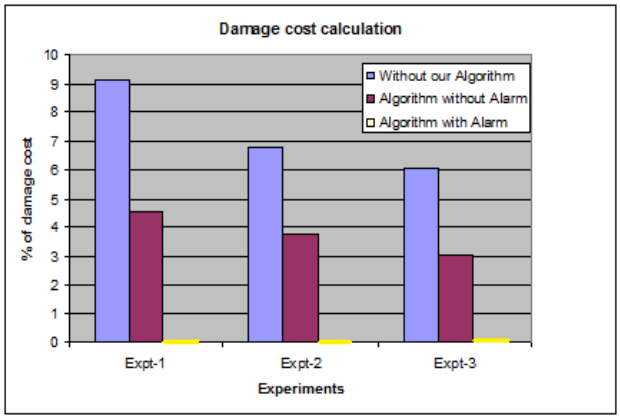

Figure 6. Experiment results with damage cost of the root

6. CONCLUSIONS

In this paper, we have described a method used for detecting instabilities in the internet along with detection of root cause location. Based on this information the malicious router can be detected and isolated. The example network topology has been analyzed to evaluate the effectiveness of our proposed scheme. It is further analyzed that when the method detects the presence of potential malicious router, the damage cost can be reduced to certain percentage without deploying the Alarm message. With the Alarm message, the damaged cost can be reduced to zero. We implemented and analyzed this proposed method by using NS-BGP simulator.

7. REFERENCES

[1] Y. Rekhter and T. Li, "Border Gateway Protocol 4", RFC 1771, SRI Network Information Center, July, 1995.

[2] A. Heffernan, "Protection of BGP sessions via the TCP MD5 signature option", RFC 2385, SRI Network Information Center, August, 1998

[3] S. Kent et al, "Secure Border Gateway Protocol (S-BGP)", IEEE Journal on Selected Areas in Communications, Volume18 Issue 4, April, 2000

[4] C. Kruegel et al, "Topology-based detection of anomalous BGP messages", Proceedings of 6th Symposium on Recent Advanced in Intrusion Detection (RAID), 2003.

[5] S. M. Belovin and E. R. Gansner, "Using Link Cuts to attack Internet Routing", draft, May 2003.

[6] S. M. Belovin, "Routing Security", Talk at British Columbia Institute of Technology, June 2003.

[7] K. Zhang, X. Zhao, F. Wu, "An analysis of Selective Dropping Attack in BGP", Proceedings of IEEE IPCCC, April, 2004.

[8] RIPE. Routing Information Service Project. http://www.ripe.net/ripencc/pub-services/np/ris-index.html.

[9] University of Oregon. The Route Views Project. http://www.routeviews.org.

[10] Mohit Lad, Akash Nanavati, Dan Massey, Lixia Zhang, "An Algorithmic Approach to Identifying Link Failures", Proceedings of the 10th IEEE PRDC-04, pp: 25 – 34, 2004.