An Authentication Based Source Address Spoofing Prevention Method Deployed in IPv6 Edge Network

Lizhong Xie, Jun Bi, and Jianpin Wu

Network Research Center, Tsinghua University,
Beijing, 100084, China
junbi@tsinghua.edu.cn

Abstract. In today's Internet routing architecture, the router doesn't validate the correctness of the source address carried in the packet, nor keep the state information when forwarding the packet. Thus the DDoS attacks with spoofed IP source address can cause security problems. In this paper, we aim to prevent the attackers from attacking somewhere outside the IPv6 edge network with forged source address in the fine granularity. The proposed methods include source address authentication by using session key and hash digest algorithm, and replay attack prevention by combining the sequence number method and the timestamp method. This paper presents the algorithm design and evaluates its feasibility and correctness by simulation experiments.

Keywords: Source Address Spoofing, IPv6, Edge Network.

1 Introduction

Today's Internet is vulnerable to a lot of security threats, such as DDoS attack TCP SYN attack, Smurf attack, etc. These attacks greatly rely on the IP source address spoofing. According to the statistics of US CERT, the increase rate of the Internet security attacks is much quicker than the development of the Internet itself.

There are several approaches to tackle the IP source address spoofing, including:

1. The end-to-end cryptographic authentication based approaches, such as IPSec, SPM [1]. IPSec supports host-to-host cryptographic authentication; and SPM is a kind of AS-to-AS authentication.
2. The tracing back based approaches. In these methods, some useful information is recorded by the routers, the ICMP messages or the packets themselves. When the receiver finds some packets whose IP source addresses are forged, it can trace back to the forger by using the recorded information. SPIE [2], iTrace [3], PPM [4], APPM [5], PPPM [6], and DPM [7] all belong to this kind of methods.
3. The filtering based approaches. In these methods, the routers filter the forwarding packets according to some filtering rules (such as the IP prefixes). These methods include Ingress filtering [8], DPF [9], and SAVE [10].

However, all of these approaches have some drawbacks:

1. In the end-to-end authentication methods, the main problem of IPSec lies in: the routers can't validate the sender's IP address, which is, in some extent, under some

Y. Shi et al. (Eds.): ICCS 2007, Part IV, LNCS 4490, pp. 801–808, 2007.

possible security threats on the routers. SPM has a rough granularity of source spoofing prevention because it just supports authentication in the AS granularity.

2. The tracing back approaches have three main flaws. Firstly, they are passive methods because they don't take action until attack has occurred. Secondly, these algorithms of the tracing back are too complicated to deploy. Thirdly, the effectiveness of this method always relies on the sensitivity of the intrusion detection.

3. The current filtering based approaches are the most effective because they can proactively prevent the source address spoofing attacks. However these methods can only filter the packets based on the IP address without validation (which means an attacker that use a legal source address won't be filtered). Therefore, it can only prevent source address spoofing in a rough granularity.

The emerging IPv6 network grants us an opportunity to redesign the trustworthy network infrastructure. In this paper, we aim to prevent the attackers from attacking somewhere outside the IPv6 edge network with forged source address in the fine granularity. If an attacker forges the source address of another host in the same edge network, the ingress filtering method won't work. We also need to prevent the replay attacks, because attackers can send the detected victim's packets to the outside of the edge network since its source address is valid. In our approach, if an attacker sends packets to somewhere outside the edge network by forging the IP address of another host or replays the victim's packets, these malicious packets will be filtered out and can't damage the outside of the edge network. Moreover, the method can work with other effective but rough-granularity methods such as ingress filing and SPM to form multi-fence defense architecture for the next generation Internet.

The rest of this paper is organized as follows: Section 2 describes the algorithm, including: the mechanism of source address validation and the mechanism of replay attacks prevention; Section 3 presents simulation results on the feasibility and the correctness of our approach; Section 4 discusses the future work and concludes the paper.

2 The Algorithms

2.1 Method Description

Our approach of source address spoofing prevention in the IPv6 edge network is based on filtering and authentication. As shown in Figure 1, we deploy a security gateway to carry out the authentication algorithm. The authentication algorithm includes two main mechanisms: the source address validation mechanism which verifies the source address by using the session key and hash digest algorithm (such as MD5 [11], SHA-1 [12], etc.), and the anti-replay mechanism which combines the sequence number method and the timestamp method.

Initially, each host in the edge network needs to be authenticated by the security gateway. This action also binds the host's IP address and its session key which is shared by the host and security gateway. If the access authentication succeeds, each packet sent to somewhere outside the edge network will carry a signature. Then, the packet passes through the source address validation and the replay check of security gateway. If both the authentications succeed, the packet will be forwarded to the edge router of the edge network; otherwise, the packet will be dropped.

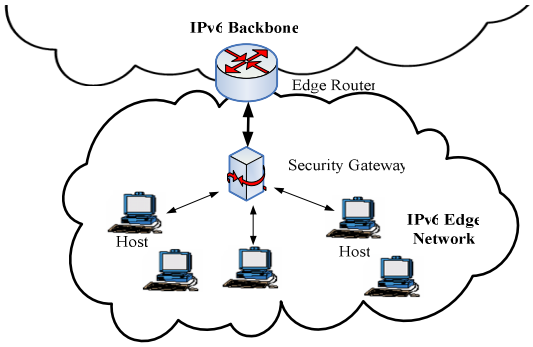

Fig. 1. The deployment scenario

Our approach mainly includes the following steps:

1. When a host wants to access the Internet, it firstly accesses the security gateway for authentication. This process can use the existing access authentication mechanism such as RADIUS [13], Kerberos [14].
2. The host generates a session key and sends it to the security gateway via some key exchange mechanisms such as IKE [15], IKE2 [16]. The security gateway binds the session key and the host's IP address.
3. When the host sends packets to somewhere outside the edge network, it needs to generate one signature for each packet by using the hash digest algorithm. The signature is carried in a new IPv6 extension header, named as "source address validation header".
4. The security gateway authenticates the signature carried in the packet to validate the source address.
5. The security gateway identifies the replay packets by checking whether the sequence number of the packet is increasing within the life time of the session key.

In addition, the session key will be changed frequently for the security purpose.

2.2 Source Address Validation Algorithm

Because every packet needs to be authenticated when they are sent to somewhere outside the edge network, the primary design requirement of source address validation algorithm is the high performance. We evaluate several authentication algorithms and finally choose the algorithm which verifies the source address by using the session key and hash digest algorithm (such as MD5, SHA-1, etc.). In this algorithm, the host certifies its ownership of a certain IP address via showing the security gateway its secret session key which is shared with the security gateway. The session key is a random number that is at least 12 bytes long. The host generates a signature by using the hash digest function with the session key and some certain data as the input. The security gateway checks the signature to validate the source address. Since the secret session key itself is not transferred in the plaintext form, the attacker is thus impossible to modify or forge another host's packets. Therefore this mechanism can support the authentication effectively.

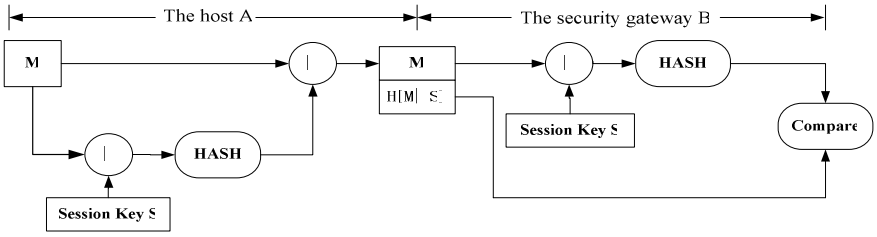

Fig. 2. The authentication process of source address validation

The authentication process shown in Figure 2 is as follows:

1. Host *A* sends a packet *M* to the security gateway *B*. *M* carries a signature *H*[*M*||*S*] which is computed by the hash digest function by using the session key *S* and the certain part of the packet *M* (source address, destination address, sequence number, etc.) as the input.

When the security gateway *B* receives the packet *M*, *B* can re-compute the hash value *HB* according to the packet *M*, since *B* also knows the session key *S*. If *HB* is equal to the signature *H* [*M*||*S*] carried in the packet *M*, the security gateway *B* can confirm that the packet has the valid source address; Otherwise, *B* can conclude that the packet's source address is forged and then drops it.

2.3 The Anti-replay Algorithm

Our anti-replay algorithm combines the timestamp method and the sequence number method. Both the timestamp method and the sequence number method are the prevailing anti-replay algorithms. The timestamp method works as follows: when the host *A* sends a packet *M* to the security gateway, the packet *M* is marked with a timestamp T_a , which represents the sending time of the packet *M*. Once the security gateway receives the packet *M*, it reads its local time T_b. If $|T_b-T_a|>\triangle T$, where $\triangle T$ is the admission time window, the security gateway can conclude that the packet *M* is a replay one then drops it. However, it's hard to synchronize the clocks of the host and the gateway exactly. Moreover, the transfer time of the packet in the network is also uncertain. Therefore, the admission time window $\triangle T$ is always larger than the real transfer time of the packet. This feature makes the timestamp method unfaithful for anti-replay. When $|T_b-T_a|<\triangle T$, the packet should be a non-replay one. But afterwards, if the replay packet is received in the margin time ($\triangle T-|T_b-T_a|$), the security gateway will wrongfully regard it as a normal packet.

The main idea of the sequence number method is: when the host *A* sends packets to the security gateway, each packet carries an incremental sequence number. If the latest packet's sequence number is greater than the previous one, the packet is normal; otherwise, the packet is a replay one. However, this method may not identify some replay packets when the sequence number is used in a cycle way. For example, assuming the length of the sequence number is 16 bits, once the sequence number reaches the maximum 65535, it will return to 0 and increase as the previous cycle. In this case, if the attacker keeps a packet of the n^{th} cycle and replays it in the $(n+1)^{th}$ cycle, the security gateway can't identify the replay packet.

In order to overcome the drawbacks of the timestamp method and sequence method, we combine these two methods to prevent the replay attack. The timestamp method can use the sequence number mechanism to identify the replay packets in the admission time window $\triangle T$; And the sequence method can avoid the confusion between the normal and the replay packet by limiting the period of the sequence number cycle within the admission time window $\triangle T$.

However, in our approach it is not necessary to mark a real timestamp in the packet. For the convenience of updating the session key, the packet carries the session key version when it is sent to somewhere outside the edge network. We can regard the session key version as the timestamp and the life time of the session key as the admission time window $\triangle T$. So what we need to do is just setting the life time of the session key less than the period of the sequence number cycle.

2.4 IPv6 Source Address Validation Header

In our approach, we design a new IPv6 extension header to carry the signature, the sequence number and other useful information. We call this new extension header "source address validation header".

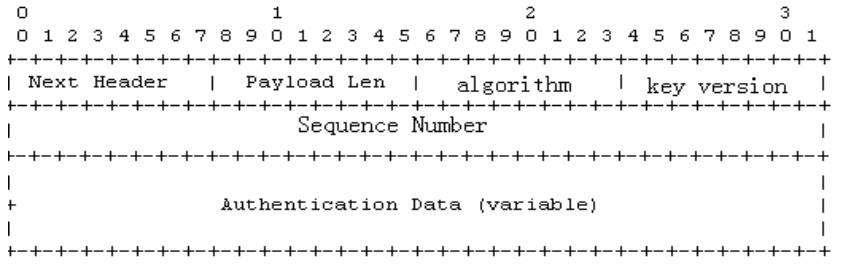

Fig. 3. The format of the source address validation header

The format of the extension header is shown in figure 3:

- Next Header: 8-bit. Indicate either the type of the next extension header or the protocol type of the payload (TCP/UDP).
- Payload Len: 8-bit. Length of the source address validation header in 8-octet units, not including the first 8 octets.
- Algorithm: 8-bit. Point out the hash digest algorithm. For example, MD5 is set to 1.
- Key Version: 8-bit. Key version refers to the version of the current using session key, for the convenience of updating the session key. We also regard it as a timestamp.
- Sequence Number: 32-bit. It is used to anti-replay as described above.
- Authentication Data: 128 bits if using MD5 as the hash digest algorithm. The authentication data is computed by the hash digest algorithm. The input of the hash digest algorithm includes: IPv6 source address, IPv6 destination address, sequence number, session key and the session key version.

The usage of IPv6 source address validation header:

1. Each packet which is sent to somewhere outside the edge network needs to carry a source address validation header. The header is inserted after the IPv6 header and before all the other extension headers.
2. The security gateway does the following validation when receiving the packet:

– Drop the packet directly if there is no source address validation header in the packet.
– Check the authentication data in the source address validation header as described above. Drop the packet if this process is failed.
– Affirm that the sequence number is greater than the previous one within the life time of the session key. If that is not true, drop the packet.

If the whole process in the step 2 passes, the security gateway needs to remove the source address validation header from the packet. Considering the partial deployment, if we don't remove the source address validation header, the hosts in other edge network may drop the packet due to misunderstanding of the new extension header. This step is unnecessary if our approach has been global deployed.

3 Experiment Evaluation

The performance and the correctness of our approach have been evaluated by simulation.

3.1 The Performance Evaluation

As described above, the performance is the primary requirement of source address validation algorithm. In our approach, this requirement means whether the performance of the hash digest algorithm is high enough. We do some experiments to testify it. Table 1 shows our experimental results, which are evaluated in the platform of Intel P4 2.0G CPU and 512M memory.

Table 1. The performance compare of two main hash digest algorithms

HASH Digest algorithm	The capacity per second (MB/S)
MD5	204.346
SHA-1	65.963

The results show that the performance of MD5 is about 1.63 Gbps (204.346MB/s $\times$ 8). This performance can satisfy the requirements of most edge networks. We should note that this result is gotten from the MD5 algorithm implemented in the software. If we implement the MD5 algorithm by using hardware, we will get a higher performance. Therefore, the source address validation algorithm in our approach is completely feasible.

3.2 The Correctness Evaluate

In order to test the correctness and the effectiveness of the source address validation mechanism and the anti-replay mechanism, we performed a simulation experiment.

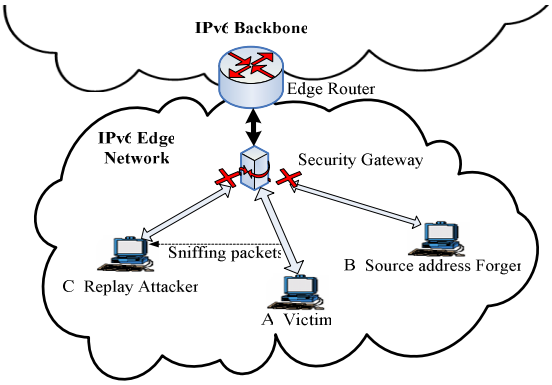

Fig. 4. The scenario of the simulation experiment

As shown in Figure 4, host *A* is the victim; host *B* sends packets to somewhere outside the edge network using the host *A*'s IP address as its source address; host *C* sniffs the packets sent by *A* and replays them. Initially, the functions of the security gateway are turned off. All the forged packets produced by *B* and the replay packets produced by *C* can be sent outwards. Once we turn on the functions of the security gateway, all these malicious packets are filtered by the security gateway (We build 10,000,000 malicious packets during the experiment.).

The experiment results show that both the source address validation mechanism and the anti-replay mechanism are effective.

4 Conclusion and Future Work

In this paper, we aim to prevent the attackers from attacking somewhere outside the IPv6 edge network with forged source address in the fine granularity. The proposed methods include source address authentication and the mechanism of anti-replay. The authentication algorithm uses the signature generated by the hash digest function (such as MD5, SHA-1, etc.) with the session key and the certain part of the packet. The anti-replay mechanism combines the sequence number method and the timestamp method to prevent the replay attack more reliably.

We evaluate our approach by using the simulation experiments. The experiment results show that our approach can prevent the source address spoofing and the replay attack effectively; and the performance of our approach is high enough to satisfy the requirement of the most edge networks. Moreover, our approach supports partial deployment.

The proposed fine-granularity method can work with other effective but rough-granularity methods such as ingress filing and SPM to form multi-fence defense architecture for the next generation Internet. We have implemented the prototype system and are deploying those mechanisms in CERNET2 (China Education and Research Network) IPv6 network.

The proposed method doesn't consider the multihoming situation yet, which refers to that an edge network obtains two or more simultaneous IP connectivity. In the

multihoming environment, a host in the edge network may have several IP addresses and the edge network may have several outbound links to different ISPs. In this case, we are studying whether we should deploy security gateway for each outbound link or just use one security gateway for all outbound traffic. We are also considering a gateway backup mechanism to protect the security gateway from attacking and becoming a bottleneck of the user traffic.

References

1. Bremler-Barr, A. and Levy, H.: Spoofing Prevention Method, INFOCOM (2005)
2. Snoeren, C. and Luis, A.: Hash-based IP traceback, SIGCOMM (2001)
3. Bellovin, S.: ICMP Traceback messages, IETF Internet Draft draft-ietf-itrace -03.txt (2003)
4. Savage, S., Wetherall, D., Karlin, A. and Anderson, T.: Pratical network support for IP traceback, SIGCOMM (2000)
5. Rizvi, B.: Analysis of Adjusted Probabilistic Packet Marking, IPOM 2003
6. Al-Duwairi, B. and Manimaran, G.: A Novel Packet Marking Scheme for IP Traceback, ICPADS (2004)
7. Belenky, A. and Ansari, N.: Tracing multiple attackers with deterministic packet marking (DPM),PACRIM (2003)
8. Ferguson, P. and Senie, D.: Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing , RFC2827, (2000)
9. Park, K. and Lee, H.: On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law internets, SIGCOMM (2001)
10. Li, J., Mirkovic, J., Wang, M., Reiher, P., and Zhang, L.: SAVE: Source Address Validity Enforcement Protocol, INFOCOM (2002)
11. Rivest, R.: The MD5 Message-Digest Algorithm, RFC1321, (1992)
12. Eastlake, D. and Jones, P.: US Secure Hash Algorithm 1 (SHA1), RFC 3174, (2001)
13. Rigney, C., Willens, S., Rubens, A. and Simpson, W.: Remote Authentication Dial In User Service (RADIUS), RFC2865, (2000)
14. Kohl, J., and Neuman, C.: The Kerberos Network Authentication Service (V5),RFC 1510, September (1993)
15. Harkins, D. and Carrel, D.: The Internet Key Exchange (IKE), RFC 2409, (1998)
16. Kaufman, C.: Internet Key Exchange (IKEv2) Protocol, RFC 4306, (2005)