Journal of Engineering and Applied Sciences 2 (12): 1823-1826, 2007
ISSN: 1816-949X

An Improved Technique that Employs a Secure Double Authentication (SDA) to the Routing Updates in Distance Vector Protocols

[1]R. Sabitha and [2]S.K. Srivatsa
[1]University of Sathyabama
[2]St. Joseph's College of Engineering, Jeppiaar Nagar, Chennai, 600 119, Tamil Nadu, India

Abstract: The Internet infrastructure security has been gaining importance in recent years due to growing concerns about cyber terrorism. The Internet routing infrastructures are vulnerable to various attacks due to the lack of strong authentication mechanisms, software vulnerabilities and software misconfigurations. Among different attacks, the routing table poisoning attack is the most important and least researched topic, which needs immediate research attention. Existing solutions do not solve the problem because they neither validate factual correctness of routing updates nor support incremental deployment. In this study, we propose a technique that employs a secure double authentication to the routing updates in distance vector protocols. In this technique, every router needs to sign the routing data twice with two different keys using a group-keying scheme, which is based on one-way hash function.

Key words: Secure Double Authentication (SDA), distance vector routing, Digital Signatures (DS)

INTRODUCTION

It is well known that today's Internet applications and the underlying routing infrastructures are vulnerable to a variety of attacks. Although a majority of incidents reported so far are realized by the exploitation of software vulnerabilities in client and server machines, it has been noted that abusing routing protocols may be the easiest way for launching attacks (Bellovin, 1989). Perlman (1992) pointed out that a single misbehaving router can completely disrupt routing protocols and cause disaster. This viewpoint has been more recently expressed by a group of network and security experts (Dering *et al*., 2000). There are many factors that make today's routing infrastructures insecure. Three of them as follow. First, there are no strong security services built into routing protocols. Many routing protocols only provide weak authentication mechanisms, e.g., plain-text password or system-wide shared keys, for authenticating peers or routing updates. As a result, it is easy for an adversary to gain access to the routing infrastructure and manipulate routing information. It is also easy for an insider to impersonate others. Second, software vulnerabilities and misconfigurations expose routing infrastructures to severe risks. Third, most routing protocols assume trustworthy environment. In the case where no authentication mechanisms are implemented., routing updates are accepted only with rudimentary validation-for example RIP (Malkin, 1998) one of the most popular distance vector routing protocols, only checks that a routing update is from an IP address of a neighbor node and that the source UDP port number is 520.When authentication mechanisms are present, routing updates are verified for the correctness of data origin and integrity only. However, after a route update is verified to be "authentic", the routing information conveyed in the update is trusted and used to update the recipient's routing table. This is risky since data origin authentication, which includes data integrity (Menezes *et al*., 1996) cannot guarantee the factual correctness of a message. A malicious entity or a compromised legitimate entity can send false information in a correctly signed message. A recipient can detect unauthorized alteration of the message, but cannot tell if the information conveyed in the message is factually correct unless the recipient has the perfect knowledge of what it expects to receive. For example, a malicious node can claim a longer distance (Hu *et al*., 2002) to avoid traffic without being detected unless the receiving node has the correct information of the network topology. In summary, cryptographic mechanisms can prevent several types of attacks, e.g., impersonation or unauthorized modification, but may not be able to prevent propagation of fraudulent information about network topology or network connectivity.

Corresponding Author: R. Sabitha, No 6, 2 Street, Abirami Nagar, Virugambakkam, Chennai- 600 092, Tamil Nadu, India

ASSUMPTIONS IN THE PROPOSED METHOD

In our method, we assume there exists an Intrusion Detection System (IDS) that can sense abnormal network events like the authentication failure, network traffic congestion or router fight back (Vetter *et al*., 1997). Here, we do not discuss the coordination mechanism between IDS and network routing protocols and how IDS makes decisions from the collected abnormal network events.

Our work is focused on the following goals: Present an authentication scheme that does not suffer form the performance issues of the public key scheme and expedite the detection of "bad" routers involved. In order to achieve our set forth goals, we propose a Secure Double Authentication (SDA) technique based on a group-keying scheme proposed in Huang and Medni (2002). This involves authentication of each Distance Vector (DV) twice, with two different keys using a one-way hash function. The basic idea behind our approach is to provide a degree of security as desired by a network designer at a relatively low cost.

SECURE DOUBLE AUTHENTICATION TECHNIQUE

Our SDA technique uses a one-way function instead of the asymmetric keying scheme. Moreover unlike Digital Signatures (DS), SDA does not use the end-to-end origin authentication. Instead, it sets up a authentication chain that follows the DV flooding path to provide data origin authentication. Here the router only needs to set up trust with its neighbors. This can decrease the number of keys used by symmetric cryptographic scheme. In this study, we present the keying scheme, which forms the basis of our proposed authentication scheme.

Keying scheme: In our scheme, the DVs that are being flooded are individually authenticated twice 2 different keys, i.e., each DV is signed twice by every router when it floods the DV to its neighbor(s). Authenticated codes are appended to the individual DVs. The first authentication code generated by the router can be verified by every other router except the neighbor(s) to which the DA is being flooded. This can prevent its neighbor(s) from altering the DV. The second authentication code can be used by its neighbor(s) to check the integrity of both the DV as well as the first authentication code. This is to ensure that if the DV and the first authentication code were altered before it reached the neighbor, it can be detected.

The keying scheme for our approach is based on Secure Group Communication Keying Scheme (SGCKS) (Huang and Medhi, 2002). In this scheme, each router has a set of Key Generation Seeds (KGS), which are used to generate the encryption/decryption key. A router uses its KGS to generate the sub-group key that is shared with everyone except its neighbor. The SGKCS scheme provides an efficient way to generate sub-group keys, thereby providing a way to reduce the number of authentication codes required for each DV.

In this study, we use x^i to represent router I and it's individual KGS as KGS^i. The individual key is represented as K^i, where $K^i = F(KGS)$. Here the function F is called key generation function. It is used to generate certain length of individual key and it is publicly known. F can be a bit wise logical operation or a one-way hash function depending on the implementation. The sub-group key for the sub-group S^i_j is represented as K^i_j and the sub-group key for S^i_j' as K^i_j'.

We enforce the security beyond the packet level. In our approach, every individual routing information data is authenticated. We use traditional authentication algorithm, for example Keyed-Hashing for Message Authentication (HMAC) (Krawczyk *et al*., 1997). The reason why we use HMAC and not DS is because the former is about a thousand times faster than the DS.

The scheme, SGKCS is suitable for authentication because each router has a secret KGS, which can be used to generate sub-group key K^i_j for S^i_j communication between router x^i and its neighbor x^j. Each DV is authenticated twice by a router using two sub-group keys discussed above. The first authentication code generated by key K^i_j is used by its neighbor to verify the routing information data. The second authentication code generated by key K^i_j' is used to guarantee that the neighbor does not alter the data. These authentication codes are appended at the end of each DV. Router x^j forwards the second authentication code to it's neighbors so that they can verify it's integrity. This is a distributed scheme where each router needs to maintain the trust-relation only between its neighbors. The trust is built up when the router is added into the network by assigning a KGS from key distribution center.

Generating the SDA: The SDA generation rules are listed below:

- When the router x^i creates an DV, it sends the DV to its neighbor x^j as $DV^i_{(j)}$', $_{(j)}$.
- When router x^j receives an DV from its neighbor router x^i, it forwards it to its neighbor x^k as $DV^j_{(k)',(j)',(k)}$.

We present an example shown in Fig. 1.

The superscript on DV identifies the router that flooded thd DV. In rule 1, when router x^i creates the DV, it

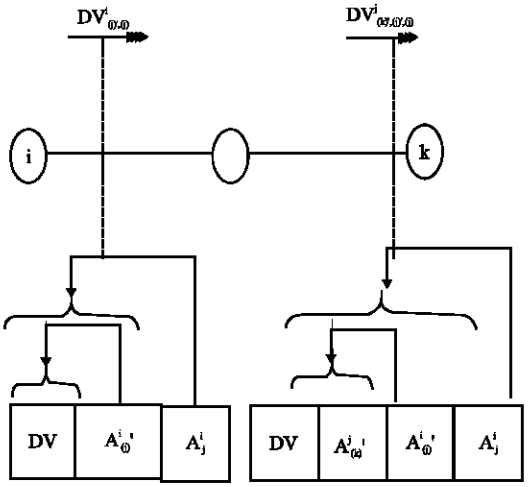

Fig. 1: The SDA example

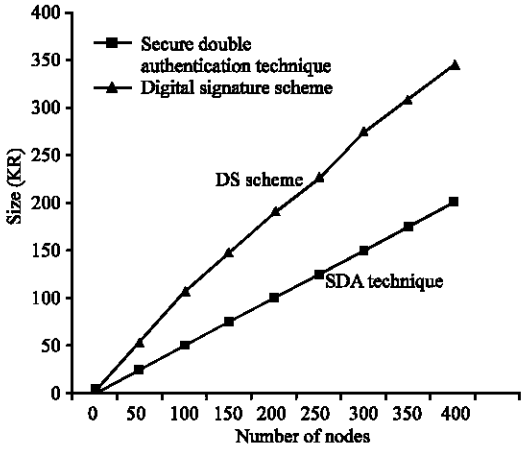

Fig. 2: Communication overhead

needs to attach two authentication codes to it. We use subscript (j)' and (j) to represent authentication codes generated by key $K^i_{j'}$ and $K^i_{,j}$ respectively. Note that authentication code A^i_j should authenticate both the DV and authentication code.

$A^i_{(j)'}$. This will help to detect if the DV and $A^i_{(j)'}$ have been altered before reaching the neighbor, as any change in the information will be reflected in A^i_j. This is the reason $_{(j)}$ is at the rightmost side in our representation. This is the same for rule 2 where the last authentication code A^j_k will cover previous two authentication codes and the DV.

When x^j receives an DV, it first detects that the source of DV is its neighbor x^i. Then, router x^j uses sub-group key K^i_j to verify authentication code A_j .^{i} Once authenticated, router x^j uses the sub-group keys K^i_k and $K^i_{k'}$ to generate authentication codes for both its neighbor x^k and x^k's neighbors, before forwarding this DV to x^k. Note, as described above, authenticated code $A^i_{(j)'}$ is attached after $A^i_{(k)'}$ as a part of the authentication data and forwarded to x^k. This can be used by x^k to verify that x^j has not altered the DV. Authentication code $A^i_{(j)'}$ needs to be attached after authentication code $A^i_{(k)'}$. This is because when x^k forwards the DV to further hops, authentication code $A^i_{(j)'}$ is not attached. So, authentication code $A^i_{(k)'}$ should not cover it. But these two authentications should be covered by code A^i_k. This will help detect any active attackers who might alter the DV and authentication codes. In our scheme only the originator sends the DV with two authentication codes attached to it. The intermediate routers generate two authentication codes but forward three codes.

Finally, router x^j forwards the DV $A^j_{(k)',(j)',(k)}$ to router x^k. When router x^k receives this DV, it first verifies authentication code A^j_k and then checks $A^i_{(j)'}$. If authenticated, it just follows same steps as router x^j does in order to generate new authentication codes before forwarding the DV to its neighbors.

SIMULATION ENVIRONMENT

We implemented the technique in NS-2. The authentication scheme for distance vector protocols that have been proposed is DS scheme. The SDA scheme is inspired by the DS scheme where authentication is provided for each DV. In this study, we draw a comparison between these schemes based on the communication overhead in terms of security that each provides.

Communication overhead: It is the size of the information that has to be carried along with the packet in order to support various schemes. For DS scheme, the signature for each DV has a variable size. This message size can technically vary from 0 to 1024 bits for 1024-bit RSA scheme. Assuming each size is equally likely, we find that on an average the length of the signature that has to be attached for each DV is 512 bits.

In case of SDA technique, authentication codes generated by three different hash keys need to be attached for each DV by the intermediate node. The DV originator attaches only two authentication codes generated by two different hash keys. Assuming that we use a 128-bit hashing algorithm, the total size of authentication codes per DV will be 384 bits for intermediate routers and 256 bits for the originator.

Hence, the communication overhead is maximum for DS scheme when it is compared with the SDA technique. The Fig. 2 shows the results.

CONCLUSION

In this study, we have presented a Secure Double Authentication Technique for Distance Vector Protocols. This method provides authentication with two different keys using a one-way hash function. In our approach, a single subverted router can be easily detected by its neighbors. When the network is partitioned by subverted routers, the SDA can only be effective within a single partition. So our SDA can be as strong as Digital Signature which provides source authentication when there is single or multiple subverted routers that do not wok together. We should provide strong security to those network nodes that can easily partition the network. In this regard, SDA technique fairs well, with low communication overhead, though it has higher memory requirements. Reducing the space complexity is one of the intended future works.

REFERENCES

Bellovin, S.M., 1989. Security Problems in the TCP/IP Protocol Suite. ACM Computer Commun. Rev., 19: 32-48.

Dering, S., S. Hares, C. Perkins and R. Perlman, 2000. Overview of the 1998 IAB Routing Workshop (RFC 2902).

Hu, Y.C., A. Perrig and D.B. Johnson, 2002. Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks In Proceedings of the Eight ACM International Conference on Mobile Computing and Networking.

Huang, D. and D. Medhi, 2002. A Flat Group Keying Scheme to Support 'Any to Any' Secure Subgroup Communication, technical report, University Missouri Kansas City.

Krawczyk, H., M. Bellare and R. Canetti, 1997. HMAC: Keyed-Hashing for Message Authentication, RFC2104.

Malkin, G., 1998. RIP Version 2. RFC 2453 (Standard).

Menezes, A.J., P.C. van Oorschot and S. Vanstone, 1996. Handbook of Applied Cryptography. CRC Press.

Perlman, R., 1992. Interconnections: Bridges and Routers. Addison-Wesley.

Vetter, B., F. Wang and S.F. Wu, 1997. An Experimental study of Insider Attcak for Routing Protocols. IEEE. Int. Con. Network Protocols, pp: 293-300.