Poster: Secure VPN using Mobile IPv6 based Moving Target Defense

Vahid Heydari
Electrical and Computer Engineering Department
University of Alabama in Huntsville
Huntsville, AL 35899
Email: vahid.heydari@uah.edu

***Abstract*—A Virtual Private Network (VPN) extends a private network across a public network, such as the Internet. It is a big challenge to protect VPNs from remote exploits that take the advantage of a bug or vulnerability in order to gain unauthorized access to a remote vulnerable system. Although firewalls, Intrusion Detection Systems (IDSs), and IPsec can prevent unauthorized access, attackers have unlimited time to find a way to penetrate the server. For example, zero-day exploits can defeat the best firewalls and IDSs as a result of using undisclosed and uncorrected computer application vulnerability. Dynamic addressing limits an attacker's time to find a vulnerable attack vector. Having a permanent IP (home address, HoA) to avoid disrupting TCP sessions and a temporary IP for connecting to other nodes (care-of address, CoA) are used in Mobile IPv6 (MIPv6) [1]. Therefore, Mobile IPv6 was selected as the base of MTM6D [2], a moving target defense method explained in our previous work. In this research, a new version of MTM6D for secure VPN (MVPN) is explained that combines MTM6D with IDS to have a dynamic shuffling interval and uses multiple IPs on the server.**

1. Introduction

The first step of a remote attack is gathering information about a victim. MTM6D dynamically changes the care-of address of the server for moving targets. Note that a real mobility is not needed in the network. We had an static shuffling interval (10 seconds) for changing the IP address in MTM6D. However, in this research a combination of MTM6D with IDS is used. In fact, a long interval is selected as the default shuffling interval and if any attack is detected, the shuffling interval will be decreased. In MTM6D, we explained that we can isolate internal attackers by putting the attacker's IP in a black list and stop updating that with the new IP. However, a legal client may share the server's IP with an external attacker. So the server cannot find the malicious client and put it in the black list. To solve this problem, as the second contribution, multiple IPs is considered on the server and an IP per each client is assigned in this new version. Therefore, if IDS detects an attack to an specific IP of the server, we can find the client that is sharing the server's IP with the attacker and put it in the black list.

2. Design

The core of this approach involves the use of multiple IPv6 CoAs. The HoA is used as the permanent address of the server and the CoAs are used as the dynamic addresses. Each CoA is assigned to each client. A pseudo-random IP addresses are generated to dynamically rotate all CoAs of the server after each shuffling interval. During each of this shuffling interval, a new CoA is assigned to each client. The binding update mechanism is used to update clients with the new CoAs. According to the multiple CoA registration rules of MIPv6, the server (acting as if it were a mobile node) will send Binding Update (BU) messages to its clients to inform them of the new CoAs. When each client receives the BU, the HoA and CoA of the server are inserted into the binding cache. The server also removes the previous CoAs.

Because of using IPsec for route optimizations, the home agent is not needed, HoA is not accessible through the Internet, so a new client cannot start a connection to the server using the HoA of the server. Instead, the connection initiation is made by the server upon receiving an out-of-band request from a client (email can be used for this purpose). When a connection request is received from a new client, the server will ping the client and, according to the standard MIPv6 procedure, the server will start the route optimization mechanism and update the client with one of its active CoAs. The server has a table to save the list of clients and their mode includes *normal mode*, *suspicious mode*, and *malicious mode*. The default mode for a new client is *normal mode*. We have different shuffling interval for each mode. The shuffling interval for normal mode (t_n) is longer than the shuffling interval for suspicious mode (t_s).

For isolating attackers, an IDS should be installed on the server. If the IDS detects an attack, it reports the attacked IP to MVPN. When MVPN receives this attacked IP, it can find the responsible client because only one client knows this IP. However, it is too early to judge the client because the attacked IP might be detected by an attacker using IP scanner. So the server should put this client in *suspicious mode* and decrease the shuffling interval (using t_s instead of t_n) for this client as shown in Figure 1. If a new attack comes to the IP assigned to the suspicious client, then the server should put the client in *malicious mode* and remove the attacked IP and stop updating the malicious client with

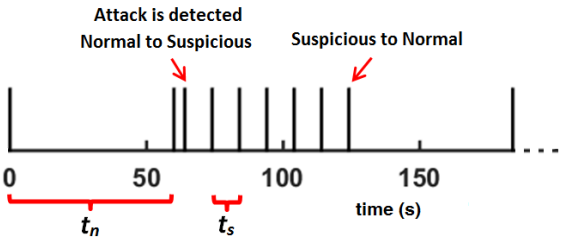

Figure 1: Effect of attack on shuffling interval.

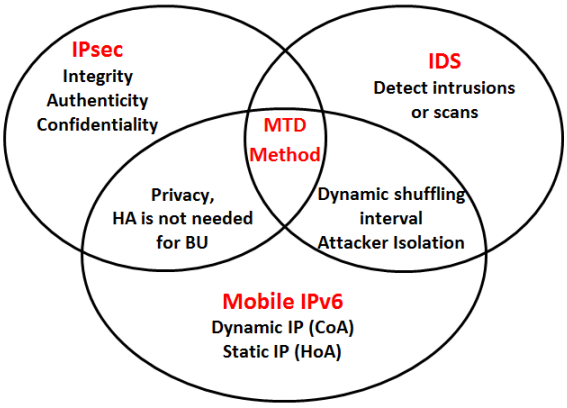

Figure 2: Design concept of MVPN approach.

the new CoA. The IP of the malicious client can be removed from the blacklist manually by the security administrator. The design concept of our overall approach is shown in Figures 2.

In view of scalability, according to the experimental results in [3], it is possible to have 55,000 IPv6 addresses bound to a single computer in suitable time. Note that their experiment was performed on a non-server-grade computer. For updating the shared symmetric key, IPsec with Internet Key Exchange version 2 (IKEv2) should be used. So the keys will be updated and we can also prevent replay attacks.

3. Implementation

Four routers and eight computers running Ubuntu 14.04 are used. An open source implementation of MIPv6 (UMIP) for Linux was used. Router R1 is used to emulate the heart of the Internet. The server's HoA does not have the same prefix with the advertised prefix of R2. So the server registered a CoA on R2 per each new client and updated the client with the new CoA as shown in Figure 3.

When the server changes its CoAs, it should update all clients with the new CoAs. During this procedure, all packets sent by clients will be dropped because the old CoAs are removed in the server's interface. For TCP test, a client generates and sends TCP packets to the server. During 50 seconds, the client sends 1000 TCP packets per second to the server. During the handoff delay TCP experiences timeout, resends the unacknowledged packet(s) and goes to slow start. This phenomenon is shown in our test results (Figure 4). The graph shows the handoff delay effect for both 10 second and 60 second shuffling intervals. We placed

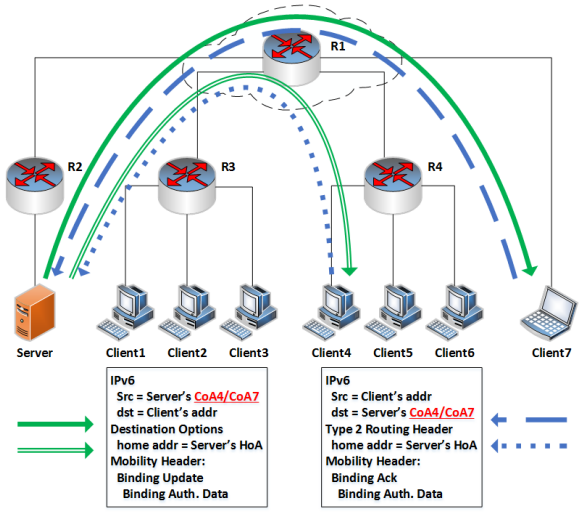

Figure 3: The Binding Update process.

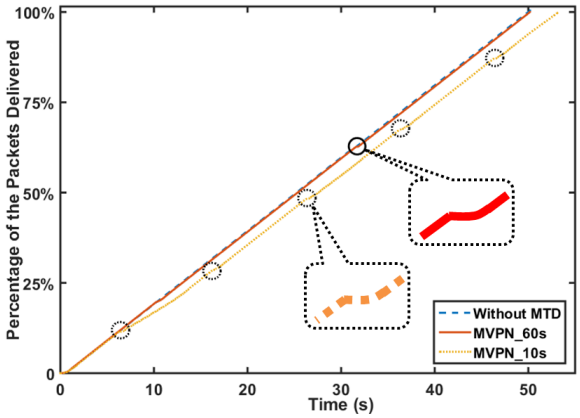

Figure 4: Percentage of TCP packets delivered over time.

circles around the areas of interest on each line that illustrate what we discussed above.

Acknowledgment

Thanks to my advisor, Dr. Seong-Moo Yoo, for assistance with this project.

References

[1] C. Perkins, D. Johnson, and J. Arkko, "Mobility support in ipv6," Internet Requests for Comments, RFC Editor, RFC 6275, July 2011, http://www.rfc-editor.org/rfc/rfc6275.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6275.txt

[2] V. Heydari and S.-M. Yoo, "Securing critical infrastructure by moving target defense," in *11th International Conference on Cyber Warfare and Security (ICCWS 2016)*, 2016.

[3] C. Morrell, J. Ransbottom, R. Marchany, and J. Tront, "Scaling ipv6 address bindings in support of a moving target defense," in *Internet Technology and Secured Transactions (ICITST)*, 2014.