Using Transport Layer Multihoming to enhance Network Layer Moving Target Defenses

[Extended Abstract]

Reese Moore, Stephen Groat, Randy Marchany, and Joseph Tront
IT Security Lab
Bradley Department of Electrical and Computer Engineering
Virginia Tech
Blacksburg, VA 24061
{ram, sgroat, marchany, jgtront}@vt.edu

ABSTRACT

As systems and networks begin to transition to the Internet Protocol version 6 (IPv6), the immense address space available in the new protocol allows for devices to maintain multiple addresses and to change addresses frequently. These new capabilities encourage network layer moving target defenses in IPv6. Yet, common transport layer protocols, such as the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP), create sockets that are bound to a single IP address and that require significant amounts of system and network overhead per session, discouraging their use for communication over multiple addresses. Stream Control Transmission Protocol (SCTP) is a transport layer protocol that allows for network sockets to use multiple IP addresses, referred to as multihoming. SCTP was tested with the Moving Target Defense for IPv6 (MT6D), a network layer moving target defense that was originally designed using UDP to dynamically change IPv6 addresses while maintaining sessions. By switching from UDP to SCTP, MT6D will improve performance and show the capability of multihomed transport layer protocols, such as SCTP, in moving target defenses.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*; C.4 [**Performance of Systems**]: Reliability, availability, and serviceability

General Terms

Security, Design

1. INTRODUCTION

As Internet Protocol version 4 (IPv4) addresses are depleted and Internet Protocol version 6 (IPv6) is globally deployed, the immense addresses space in IPv6 brings new opportunities for moving target defenses. Network layer moving target defenses in an IPv4 environment are difficult, expensive, and impractical to implement due to the limited addresses space and ease of network scanning and targeting. To combat the shortage of addresses in IPv4, IPv6 employs 128-bit addresses. With the current number of Internet-ready devices on the network, the immense address space is sparsely populated and allows for hosts to use multiple IP addresses simultaneously and to change IP addresses frequently. Implementing a network layer moving target defense in the new, dynamic IPv6 network environment has become a reality.

Network layer moving target defenses protect systems by dynamically changing the network locations of hosts while maintaining active sessions. This allows for users to have seamless communications and prevents attackers from successfully targeting a system for exploitation. In IPv4, the limited addresses space allows for subnets to be scanned in seconds by a single host and the entire IPv4 Internet to be scanned in minutes when distributed over multiple systems. Even if hosts dynamically change addresses in IPv4, the limited address space and ease of scanning makes targeting the host for exploitation possible. The large address space of IPv6 makes this scanning impossible, since scanning a single subnet takes approximately $8.77 \cdot 10^{10}$ years [5]. By dynamically changing addresses in IPv6, targeting a host becomes difficult; reacquiring the target after an address change becomes practically impossible. Yet, the current transport layer protocols, such as Transmission Control Protocol (TCP) and User Datagram Protocol (UDP), are designed for IPv4 and prevent network layer moving target defenses from being effective.

Current popular transport layer protocols, such as TCP and UDP, prevent moving target defenses from being effective by restrictively binding network sockets to a single address. Originally designed for IPv4, these protocols cannot dynamically change addresses without creating new sessions, significantly increasing network and system overhead for communications involving multiple addresses. Transport layer protocols capable of using multiple IP addresses in a single session have been designed and implemented, deploying a feature termed multihoming. Originally designed for increased performance and reliability, multihoming protocols, such as the Stream Control Transmission Protocol

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
CSIIRW '12, October 30 - November 2, Oak Ridge, Tennessee, USA
Copyright 2012 ACM 978-1-4503-1687-3 ...$15.00.

(SCTP), allow for a single network socket to use multiple IP addresses during the same session. In a moving target defense, multihoming transport layer protocols can increase flexibility and performance by dynamically changing addresses while maintaining a single session.

This paper discusses how multihoming transport layer protocols can improve network layer moving target defenses. First, we provide background on IPv6, multihoming, and moving target defenses in Section 2. Related work on transport layer security and moving target defenses is discussed in Section 3. Section 4 explains the design of MT6D, a network layer moving target defense using UDP. After switching MT6D to a multihoming transport layer protocol in Section 5, the results are analyzed in Section 6. In Section 7 we discuss future work and conclude in Section 8.

2. BACKGROUND

The tremendous growth of the Internet has driven the need for a new version of the Internet Protocol, version 6. With the new, larger address space in IPv6, the deployment of single hosts with multiple addresses has become a reality. This large and dynamic address space also encourages new network defenses, including moving target defenses. Specifically, a new system named the Moving Target Defense for IPv6 (MT6D) creates hosts that communicate through multiple IPv6 addresses during a single session. Yet, these network defenses are limited by popular transport layer protocols limiting a single address per network socket. To allow for a host to use multiple addresses on a single network socket, new transport layer protocols have been developed to allow for multihoming. Multihoming allows for a single network socket to use multiple network addresses, enhancing performance for network layer moving target defenses.

2.1 IPv6

As previously mentioned, IPv6 was primarily developed to create more address space in the Internet. An IPv4 address consists of 32 bits, which provides a little over 4.2 billion possible addresses. This address space is not enough to support the myriad of Internet capable devices. Therefore the IPv6 address was expanded to 128 bits. This new address size allows for 2^{128} possible addresses, approximately $7.92 \cdot 10^{28}$ addresses for every one IPv4 address.

2.2 Moving Target Defense

A moving target defense is a network protection system that that proactively prevents attackers from exploiting hosts. Current network addressing systems use static addresses that allow users to easily identify systems. These static addresses, though, allow for attackers to easily target and exploit systems. By using a moving target addressing system and frequently changing network addresses, networks can avoid attacks by constantly changing the attack surface. This defense is also effective since it creates a dynamic element at a very low layer of the Open Systems Interconnection (OSI) model. By securing the network and transport layer, network layer moving target defenses provides inherent security for the higher layers, including the application layer. Since the attacker must continually reacquire the target as the target continues to logically move within the network, the attackers must use more resources and time to find the system. By increasing the cost of network attacks, network administrators and network security professionals can avoid the need to analyze an excessive amount of basic, weak attacks launched by individuals and focus defenses and efforts of dangerous threats perpetuated by well resourced attackers, possibly nation states. Conversely, static network systems are essentially free for attackers to find. Also, by forcing attackers to remain in the planning and target acquisition phase of an attack, targets can avoid exploitation.

The Moving Target Defense for IPv6 (MT6D) is a network layer moving target defense protocol that dynamically changes network addresses while maintaining sessions. Exploiting the large address space in IPv6, the network defense proactively changes IP addresses between communicating hosts. Using StateLess Address Auto Configuration (SLAAC) and Neighbor Discovery Protocol (NDP) in IPv6, hosts can self obtain and use multiple network addresses simultaneously, logically moving throughout the network to avoid attack. Yet, the system's use of UDP as a transport layer protocol reduces the protocol's performance.

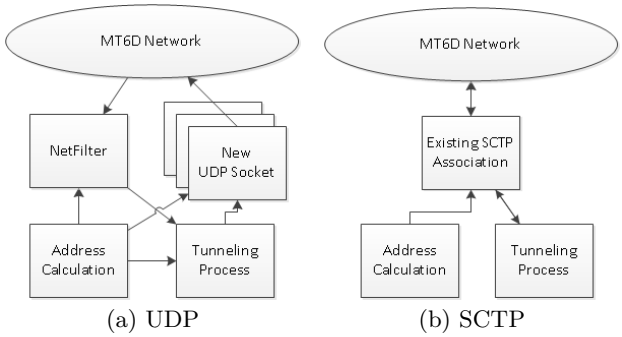
2.3 Multihoming

In current popular transport layer protocol such as TCP and UDP, sockets can only be bound to a single addresses at a time. When trying to create a moving target defense, TCP is especially detrimental when it creates a significant amount of system and network overhead for connection creation and teardown. Moving target defenses require flexibility in the transport layer to allow for multiple address on a single socket. By being able to have a single socket with multiple addresses managed by the operating system, programmers can port applications to moving target defense protocols, such as MT6D, without significant rearchitecture of their work. Multihoming in the transport layer creates the required flexibility without requiring changes to existing layer 3 infrastructure.

In transport layer protocols, multihoming was originally designed to allow for systems and sockets to choose the best available network for communication. To prevent a single point of failure in a network, a multihomed host could have multiple network connections seamlessly available for a single network socket, assuring that if one network connection was disabled, other connections would be available to transport the payload. Simultaneously, the receiving host did not have to manually manage the different addresses of the sender; packets received by the socket would be bound for the application. In theory, a single multihomed socket could be connected to multiple networks, including networks of different speeds and connection states. Multihoming sockets can also have addresses dynamically added after their creation. The Stream Control Transmission Protocol (SCTP) [8] is a transport layer protocol that supports multihoming. It has been widely implemented in the Linux kernel, though Microsoft Windows does not natively support the protocol.

3. RELATED WORK

Security vulnerabilities in the transport layer and multihoming have been thoroughly addressed, respectively, yet little work has been done on exploiting the defensive security advantages of multihoming in the transport layer. Significant work has been done on transport layer security, applying cryptography to the transport layer payloads. The Transport Layer Security protocol [3] is an industry-standard protocol to add security to TCP connections. Work has also been done to address security vulnerabilities in mul-



(a) UDP (b) SCTP

Figure 1: MT6D Transport Layer Designs

tihomed environments and SCTP, specifically. Nordmark [6] analyzed multihoming in the IPv6 protocol and the potential network layer concerns of allowing multiple addresses on a single interface. Aura et al. [2] analyzed SCTP mobility and multihoming functionality, finding multiple attacks and exploits in the protocol. Yet, few have looked at the potential defensive advantages offered by multihoming. Atkinson et. al [1] presented a new security naming scheme, using the additional security offered by multihoming. This work differs by using the performance and architecture benefits of transport layer multihoming in a network layer moving target defense.

While others have looked at creating network layer moving target defenses, no current implementations of moving target defenses have progressed to a state where analysis of their transport layer mechanisms have been completed. A technique by Sheymov [7] was designed with the goal of dynamically obscuring cyber coordinates, which can represent any piece of information in cyber space. Since this defense uses a custom transport layer protocol and custom hardware, it suffers from easy identification in packet traces and would not be effective in today's network environments. Fink et al. [4] also proposed a technique for dynamically obscuring host addresses called Adaptive Self-Synchronized Dynamic Address Translation (ASD). ASD is similar to MT6D in that its objective is to hide the location of communicating hosts. It does this through a handshake process between a trusted sender and receiver enclave to assign source and destination addresses. Obscured addresses are selected from those available to the ASD enclave. ASD also creates a custom transport layer protocol, mimicking some of the features of TCP and SCTP. This research looks to apply available transport layer protocols to network layer moving target defenses to increase the security and anonymity of the defensive.

4. CURRENT DESIGN

MT6D creates a network tunnel, allowing traffic to be sent across potentially hostile networks in a secure and obfuscated manner [5]. The MT6D node acts as a gateway for hosts communicating with the remote network. The packets are received by this gateway node, encapsulated in an end-to-end virtual private network with dynamically changing network addresses and sent the remote gateway. At the remote gateway, the packets are reassembled in their original form and forwarded to their destination.

This transit is transparent to the communicating hosts; it appears that they are sending and receiving packets through a regular network gateway. To an attacker who is attempting to monitor their network traffic, however, all communications are obfuscated and it becomes impossible to reliably target any of the endpoints in the communication.

The MT6D protocol was originally designed to use UDP as the transport layer protocol for tunneled messages. UDP was created for communication between two hosts with fixed network addresses and does not, as part of the protocol or in common implementations, support adding multiple addresses to an association. The problem of creating a network layer moving target defense is fundamentally equivalent to the problem of managing a multiple address multihomed association between nodes.

When designing a network layer moving target defense, multihoming is necessary. If the transport layer protocol does not support it natively, it is necessary to implement it at the application layer. This creates difficulties when working with the socket abstraction. Among those difficulties are the need to manage shared state and to efficiently receive all messages sent through the tunnel. The MT6D solution to these problems is illustrated in Figure 1(a).

Forwarding a message requires the application to create a new network socket with the appropriate addressing information for each packet. This requires the tunneling routines to read the current address state and pass that information into the operating system for every send operation. In practice, MT6D creates a new UDP socket for each packet sent. This is possible because there is little overhead associated with opening a UDP socket for transmission, but would be infeasible in other transport layer protocols such as TCP which require more system resources on initialization.

Receiving messages is particularly difficult when managing the multihomed association at the application layer. The application cannot create a socket and block on a read operation until data has arrived, because it will have to repeatedly remove and add sockets to the set that it is monitoring. In addition to being difficult for the programmer to manage conceptually, the repeated creation and destruction of listening sockets is an unacceptable use of system resources. Instead, the application must resort to using more complex methods of receiving packets. MT6D utilizes the Linux kernel's netfilter system to receive every packet destined for the intended address.

5. MULTIHOMING DESIGN

MT6D implemented using SCTP as the transport layer protocol provides a more efficient and simpler design while retaining all of the features discussed in Section 4. SCTP allows MT6D to offload work into the transport layer, which reduces code size and complexity and moves processing from user space to kernel space.

Using transport layer protocols that natively support multihoming, such as SCTP, the design of a network layer moving target defense is simplified by passing the responsibility of sending and receiving messages with the proper address pair to the underlying operating system. The program can then be split into separate processes that are designed to do a single task efficiently.

The design consists of two separate processes: a tunneling process and an address management process. The tunneling process sees a single persistent connection that sends and receives messages using standard socket API calls. The ad-

dress management process sees a network association to add and remove addresses using the protocol's library. In practice, MT6D using SCTP has been implemented on Linux using the Linux Kernel Streaming Control Transmission Protocol (LKSCTP) project SCTP library.

These processes do not directly share state, except through the association that is mediated by the operating system, as seen in Figure 1(b). Both processes can be designed using existing and accepted networking and system programming paradigms, simplifying their implementation, better utilizing system resources, and making them more efficient.

Utilizing multihomed transport layer protocols for moving target defenses is not without its drawbacks, however. These newer protocols, such as SCTP, have not reached the adoption levels of common protocols such as TCP and UDP. As such, they appear to be anomalous, and may draw more attention to the users. Further, certain networking appliances may not be configured to work with less common transport layer protocols. These are current concerns, however, they will be less relevant as modern transport layer protocols become more prevalent.

6. ANALYSIS

For a moving target defense, some form of multihoming is required in order to manage the apparently separate streams of communication utilized. While it is possible to implement multihoming at the application layer, as is the case in MT6D over UDP, utilizing a protocol that allows the application to use an existing multihoming implementation at the transport layer has many benefits over the application layer design.

Using a protocol that natively supports multihoming allows the application to be written in the style of other networking programs. Not only does this mean that programmers will be familiar with the style and therefore more likely to produce correct code, but modern operating systems are designed to work efficiently with this model. Additionally, this allows existing applications to be easily ported to use a moving target defense with minimal software changes. TCP based applications can take advantage of SCTP through the use of TCP-to-SCTP wrappers.

When receiving messages, the application can simply block on a read and the operating system will not devote systems resources to that thread until data arrives. When sending messages, the application simply has to write data to the socket, and the operating system already has the address pair and will send the packet appropriately.

When designing a multihoming interface at the application layer, using operating system specific methods and subsystems are necessary in order to get acceptable performance. In simplifying the design, the application more closely adheres to standard API calls, meaning it is more easily ported to other systems.

By pushing the additional processing into the operating system, the design becomes much simpler. This allows for a cleaner implementation as well as the ability to use system resources more efficiently to achieve the same results.

7. FUTURE WORK

In the next stage of our research, we will utilize privileged information about the SCTP association to prevent the protocol from leaking information in the absence of Internet layer encryption, such as IPsec. This will be implemented in such a way that the application is oblivious to the modifications, and therefore doesn't complicate application design.

Once that is implemented, we will be comparing and contrasting several transport layer protocols in network layer moving target defenses. TCP, UDP, and SCTP are expected to exhibit different behavior when being used as the transport layer for a moving target defense tunneling application. We will deploy several implementations that utilize different transport layer protocols across Virginia Tech's production scale IPv6 network and the global IPv6 Internet to gather concrete performance data. This will allow us to analyze the strengths and weaknesses of those protocols for use in network layer moving target defenses.

8. CONCLUSION

The increased size of the Internet in IPv6 allows for network-layer moving target defenses to be a highly effective method of security. Yet, the transport-layer protocols needed for these defenses were originally designed for IPv4 and, due the limitation of being restricted to a single address per a session, are ineffective for moving target defenses. By switching from transport layer protocols such as UDP to multihoming capable protocols which can use more than one address per a session, such as SCTP, moving target defenses can be successfully deployed and adapted to new systems with minimal overhead. MT6D, a current network-layer moving target defense, has been adapted to use SCTP from UDP and gained increase performance, transparancy, and system flexibility.

9. REFERENCES

[1] R. Atkinson, S. Bhatti, and S. Hailes. Ilnp: mobility, multi-homing, localised addressing and security through naming. *Telecommunication Systems*, 42:273–291, 2009. 10.1007/s11235-009-9186-5.

[2] T. Aura, P. Nikander, and G. Camarillo. Effects of mobility and multihoming on transport-protocol security. In *Security and Privacy, 2004. Proceedings. 2004 IEEE Symposium on*, pages 12 – 26, may 2004.

[3] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (Proposed Standard), Aug. 2008. Updated by RFCs 5746, 5878, 6176.

[4] R. A. Fink, M. A. Brannigan, S. A. Evans, A. M. Almeida, and S. A. Ferguson. Method and apparatus for providing adaptive self-synchronized dynamic address translation. Patent, May 2006. US 7,043,633.

[5] S. Groat, M. Dunlop, R. Marchany, and J. Tront. Using dynamic addressing for a moving target defense. In *the 6th International Conference on Information Warfare and Security (ICIW 2011)*, Mar. 2011.

[6] E. Nordmark and T. Li. Threats Relating to IPv6 Multihoming Solutions. RFC 4218 (Informational), Oct. 2005.

[7] V. I. Sheymov. Method and communications and communication network intrusion protection methods and intrusion attempt detection system. Patent, Feb. 2010. US 2010/0042513 A1.

[8] R. Stewart. Stream Control Transmission Protocol. RFC 4960 (Proposed Standard), Sept. 2007. Updated by RFCs 6096, 6335.