ASTES

Advances in Science, Technology and Engineering Systems Journal Vol. 2, No. 3, 30-35 (2017)

www.astesj.com

Special Issue on Recent Advances in Engineering Systems 2017

ASTESJ

ISSN: 2415-6698

Zero Trust Cloud Networks using Transport Access Control and High Availability Optical Bypass Switching

Casimer DeCusatis [*,1], Piradon Liengtiraphan [1], Anthony Sager [2]

[1]*Marist College, School of Computer Science and Mathematics, Poughkeepsie, NY 12603 USA*

[2]*BlackRidge Technologies, Research & Development Division, Reno NV 89433 USA*

ARTICLE INFO

Article history:
Received: 02 March, 2017
Accepted: 25 March, 2017
Online: 04 April, 2017

Keywords:
Cybersecurity
Authentication
Bypass

ABSTRACT

Cyberinfrastructure is undergoing a radical transformation as traditional enterprise and cloud computing environments hosting dynamic, mobile workloads replace telecommunication data centers. Traditional data center security best practices involving network segmentation are not well suited to these new environments. We discuss a novel network architecture, which enables an explicit zero trust approach, based on a steganographic overlay, which embeds authentication tokens in the TCP packet request, and first-packet authentication. Experimental demonstration of this approach is provided in both an enterprise-class server and cloud computing data center environment.

1. Introduction

Network-based cybersecurity attacks against cloud data centers have been increasing in both frequency and severity, far outstripping traditional defense methods. In an effort to combat this problem, this paper is an extension of work originally presented at the recent IEEE International Conference on Smart Cloud [1] which introduced the use of first packet authentication in a zero-trust cloud network. We further develop the use of transport access control (TCP) for authentication in cloud environments, such as Amazon Web Services (AWS). We then introduce a high-speed optical bypass switch into the system architecture, and characterize its performance under realistic load conditions.

Security remains a significant concern for cloud computing environments. According to recent industry survey data [2], 27% of connected third-party cloud applications introduced into enterprise environments in 2016 posed a high security risk. Authentication is a particular problem in cloud environments, since the traditional security perimeter is effectively virtualized. Many cloud applications cannot be effectively segregated from the corporate infrastructure, and communicate freely with software-as-a-service (SaaS) platforms [2]. Cloud data centers are under constant attack from a variety of bad actors; a moderately-sized commercial data center network can experience over 100,000 security events per day [2,3]. These attackers range from individual hacktivists and cyber-gangs motivated by creating social disruption to large, well organized groups with political or financial motivations who are backed by nation-states. Increasingly, these attacks have multiple goals, including compromising critical network resources such as the network controller.

In response to the growing number and sophistication of cybersecurity threats, a United States Presidential Executive Order tasked the National Institute of Standards and Technology (NIST) with creating a set of voluntary policies and guidelines to help develop the U.S. cybersecurity framework. This eventually led to the so-called "zero trust model" for information security [4]. Traditional security models are based on a perimeter security model (also known as an implicit trust model or "trust but verify" approach), in which all communication is trusted between devices within a specified security group. This relatively static approach is based on segmenting the network and creating a demilitarized zone (DMZ) between trusted and untrusted portions of the network. Perimeter security breaks down in modern cloud computing environments, leading to a new approach that explicitly verifies all network connections. These zero trust networks assume that all traffic is a threat until it is authenticated and inspected. Zero trust is intended to provide a scalable security infrastructure that redefines the approach to resource

*Corresponding Author C. DeCusatis, Marist College, Poughkeepsie, NY, Casimer.decusatis@marist.edu

segmentation, a fundamental principle in which resources to be protected are grouped together and securely isolated or partitioned to limit unauthorized access.

While a full zero trust network has not yet been commercially realized, significant progress has been made towards the development of enabling technologies. It is generally recognized that long-standing segmentation techniques such as VLANs no longer provide sufficient cloud network security [2-6]. Many organizations attempt to segment their networks in a coarse granularity fashion to reduce risk, subject to limitations imposed by legacy hardware, complex virtualization software, and a lack of programmable resources [6]. By contrast, a zero trust network security architecture incorporates an explicit trust model and dynamic, automated security policy that extends across conventional security boundaries but still provides fine granularity segmentation and isolation of critical resources. All traffic needs to be validated, even between virtual machines (VMs) sharing a common physical host. Explicit security is part of a layered, defense-in-depth approach, which avoids kill chains and thus prevents single points of failure from compromising the entire security defense system. Fine grain segmentation improves management visibility and makes it feasible to disrupt network attacks as early as possible in the attack process, preferably to prevent data reconnaissance techniques from even identifying the resources which are being protected, much less being able to fingerprint these resources in preparation for an attack.

Micro-segmentation of a zero-trust network should preferably include authentication of not just users or applications, but also extend to the level of authenticating individual packets. Conventional networks assert the identity of a user or application based on a series of attributes such as network addresses, which may be forged [5]. Such networks may decide to trust a user or application based on some criteria, but the concept of trust does not apply to conventional network packets, which are the fundamental building blocks of any network. Our research implements a form of authentication with packet level granularity, which offers several advantages. First, a finely grained approach (at the packet level) improves visibility, particularly when combined with analysis of management plane data logs. Other potential benefits of our approach include simpler, vendor agnostic architectures, better scalability, and improved application portability.

Our approach helps avoid unauthorized awareness (a request for access to the network should not only be denied, it should avoid providing the requestor with any information about the nature of resources that are connected to the network). For example, modern data center networks are subjected to a constant stream of access requests, since even a denied TCP connection request will return some information about the nature of the network, thereby assisting attackers in fingerprinting the target system [5, 6]. This means that potential attackers can gather useful information about a potential target by repeatedly trying to complete a connection request, even if the request attempt itself does not succeed. The information collected in this manner may be used to plan future attacks or identify weaknesses in the perimeter defenses. The approach demonstrated in this paper prevents error message reconnaissance information from reaching a potential attacker, without compromising performance of the remaining system.

While the theoretical approach to zero trust data center architectures encompasses a wide range of components, for the remainder of this paper we will concentrate on disruptive network technologies. For example, zero trust networks benefit from the centralized management plane and dynamic configurability offered by software-defined networks (SDN). While the definition and basic operating principles of SDN networks are well known [7], we will briefly mention several of their useful features. Programmable SDN controllers are able to implement dynamic network segmentation based on data collected from sources outside the network itself, such as honeypots, security analytic engines, and other sources. The application of security analytics to monitoring or management data sets enables the creation of actionable threat intelligence, allowing an SDN network to proactively discourage security threats and respond in near real time when new threats become apparent. This approach is particularly effective when combined with virtualized network functions (VNFs) such as virtual routers, firewalls, or other appliances [8, 9].

In this paper, we describe an approach that enables zero trust networks by providing first-packet based authentication, combined with transport layer access control, and experimentally demonstrate the use of this approach in defending an SDN controller from cyberattacks. We describe a steganographic overlay that embeds network authentication tokens in a TCP connection request, and blocks unauthorized traffic from completing a request. Resources protected in this manner are effectively concealed from reconnaissance attempts by attackers. We also demonstrate an approach to transport layer identity management and authentication. We show that this approach prevents fingerprinting of key network resources (such as the SDN controller) by blocking any response to unauthorized packets at the transport layer and below. We experimentally demonstrate this approach in both large, enterprise-class servers and a cloud computing test bed.

This paper is organized as follows. After the introduction, we describe the operation of a transport layer identity management scheme in section 2. We then present experimental results for the enterprise server and cloud test bed use cases in section 3. Finally, section 4 presents our conclusions and recommendations for future work.

2. Transport Access Control Architecture

Our approach independently authenticates each network session at the transport layer, prior to granting any access to the network. This is implemented through a combination of two technologies, namely transport access control (TAC) and first packet authentication. To our knowledge, these approaches have not previously been combined in this manner. Unauthorized session requests are completely rejected, and there is no feedback to a potential attacker attempting to fingerprint the network. Explicit trust is established by generating a network identity token during session setup. The network token is a 32 bit, cryptographically secure, single use object which expires after four seconds. Tokens are associated with identities from existing Identity Access Management (IAM) systems and credentials, such as Microsoft Active Directory or the IAM system used by Amazon Web Services [10]. Explicit trust is established by authenticating these identity tokens as early as possible, namely on the first packet of a TCP connection request (see Figure 1).

Tokens are generated for each unique entity (user or device) requesting access to network resources. An in-line virtual appliance (the TAC gateway) is installed between the equipment

being protected (the protected resource) and the rest of the network; a second gateway is installed between the authorized or trusted user and the rest of the network. When the trusted user attempts to access the protected resource, the first gateway inserts an identity token into the first packet of the TCP connection request. When the second gateway receives a connection request, it extracts and authenticates the inserted identity token and then applies a security policy (such as forward, redirect, or discard) to the connection request based on the received identity. This gateway acts as a policy enforcement point transparent to the rest of the system architecture and backwards compatible with existing network technologies. If the network access token for a TCP request fails to resolve to an identity or resolves to an identity that lacks the authority to access the requested resource, then the connection request is rejected without providing any further response to the requestor, effectively cloaking the presence of the protected resource. Failed access attempts are logged in an external Syslog server, which allocates enough memory to avoid wrapping and over-writing log entries. Existing tools such as SIAM can be used to analyze the logs or generate alerts of suspicious activity. We note that continuous logging of all access attempts is consistent with the approach of a zero-trust network (i.e. not allowing any access attempts to go unmonitored). Both the identity insertion gateway and identity authentication gateway appliances can be hosted as VNFs hosted on a virtual server, router, or other compatible piece of networking equipment.

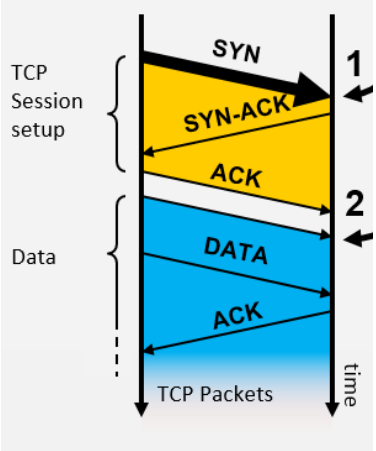

Figure 1: Transport Layer Authentication handshake; first packet authentication is performed at position 1 (as early as possible); traditional session authentication is performed at position 2 (after the session is already established).

This approach offers several advantages, including separation of security policy from the network design (addresses and topologies). This approach works for any network topology or addressing scheme, including IPv4, IPv6, and networks that use the Network Address Translation (NAT) protocol, and is compatible with dynamic addressing often used with mobile devices. This approach extracts, authenticates, and applies policy to the connection requests, not only protecting against unauthorized external reconnaissance of the network devices but also stopping any malware within the protected devices from calling home (exfiltration). Security policies can be easily applied at the earliest possible time to conceal network-attached devices from unauthorized awareness. By preventing unauthorized awareness and access, transport access control blocks both known and unknown attack vectors. This approach is low latency and high bandwidth since packet content is not inspected. Since the network tokens are embedded in the TCP session request, they do not consume otherwise useful data bandwidth. The combination of transport access control and a segmented, multi-tenant network implements a layered defense against cybersecurity threats, and contributes to non-repudiation of archival data. These techniques are also well suited to protecting public and hybrid cloud resources, or valuable, high performance cloud resources such as enterprise-class mainframe computers. Further, this approach can be applied to protecting the centralized SDN network controller from unauthorized access, and enable only authorized SDN controllers to manage and configure the underlying network. TAC uses an innovative identity token cache to provide high scalability and low, deterministic latency. The token cache is tolerant of packet loss and enables TAC deployments in low bandwidth and high packet loss environments.

3. Cloud test bed experimental results

The experimental cybersecurity cloud test bed is illustrated in Figure 2. A protected resource (in this case, the SDN controller) is intended to be accessible only from either of two trusted clients. A BlackRidge hardware appliance gateway, which implements TAC with first packet authentication [11], is placed in-line with the trusted clients, where it inserts tokens in the transport frame headers. Tokenized packets flow through the untrusted network, which eventually routes them to the SDN controller. A virtual gateway appliance is placed between the network and the SDN controller, which will authenticate the tokenized packets and only allow authenticated and authorized packets to pass through to the SDN controller. Any packets without Tokens or identified traffic without the authority to access the SDN controller will be dropped. Our test configuration uses a hardware appliance to insert Tokens and a Virtual Appliance running on VMWare ESXi to authenticate Tokens. The hardware and software appliances are only addressable through their management ports and use the management ports to access the required network time protocol (NTP) Servers. A list of trusted devices to be allowed access is provisioned in the TAC gateways, and the list of trusted devices can be edited using the gateway management ports.

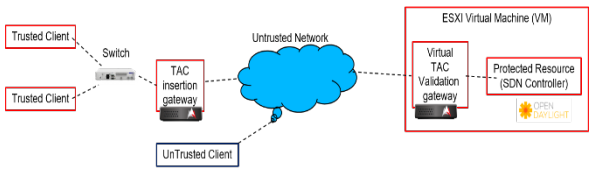

Figure 2 – Experimental cybersecurity cloud test bed

The specifications for the gateway used in this testbed are given in Table 1. This gateway has three modes of operation, known as Bridge, Enforce, and Monitor. In Bridge mode the gateway does not perform authentication or insert tokens into the data packets; rather, it simply functions as a two port, Layer 2 bridge device. Enforce mode will perform authentication and insert tokens into the 32 bit sequence and acknowledgement number fields of a TCP frame, according to the established address list policy. Monitor mode has the same functionality as Enforce mode, with the exception that it does not enforce the security policy. Monitor mode is useful to validate configurations during installation and setup for a new gateway. By toggling a configured gateway between Bridge and Enforce modes, it is possible to observe the effects of turning token-based authentication off and on. The gateway architecture is a “bump

in the wire" approach, and the gateway device is only addressable through its own dedicated device management access port.

Table 1 – TAC gateway capabilities

Performance	Branch Gateway	1G VM, z/VM, and AWS Gateway[1]	1G 1RU Gateway	10G VM Gateway[1]	10G 1RU Gateway	z Systems LPAR Gateway[1]
Throughput	1 Gb/sec	1 Gb/sec	1 Gb/sec	10 Gb/sec	10Gb/sec	1Gb/sec
Latency	<200 µsec	<100 µsec	<80 µsec	<100 µsec	<12 µsec	TBD
Max Capacity	**Branch Gateway**	**1G VM/KVM Gateway**	**1G 1RU Gateway**	**10G VM Gateway**	**10G 1RU Gateway**	**z Systems LPAR Gateway**
Identities[2]	1K	10K	10K	30K	40K	10K
Concurrent Sessions	100K	1M	1M	4M	4M	1M

1. Virtual appliance performance is server platform dependent. Required resources:
   - 1GbE VM, z/VM, AWS: 4 core CPU, 4-GB memory, 8GB storage
   - Branch Gateway 2 cores 8gb memory, 256GB SSD storage
   - 10GbE VM: 8 core CPU, 16 gb memory, 8GB storage
   - z Systems LPAR Gateway: 1 IFL, 16 GB memory, 32GB storage
2. Total Identities is limited by the Enterprise IDMS

We configured the test bed and toggled the gateway between Bridge and Enforce modes. This allowed us to verify that Enforce mode would only permit tokenized packets from one of the trusted clients to reach the SDN controller. We then attempted a reconnaissance scan of the SDN controller from an untrusted client. These scans were conducted using several industry standard tools, including Metasploit, HTTPrint, Firewalk, and PuTTY [5, 6]. When the TAC gateways were in Bridge mode, we were able to successfully fingerprint the SDN controller, and determine that it was an instance of Open DayLight (Helium release) running OpenFlow 3.2 in this example. We then repeated the scans with the gateway in Enforce mode, and as expected we were unable to identify even the presence of an SDN controller since The TAC authentication gateway blocks all potential responses at and below the transport layer. We cannot even determine if there is a TAC gateway present, as TAC was also used to protect the management port of the gateway. This implements both packet level authentication and unauthorized awareness, both desirable properties in a zero trust architecture.

To evaluate the effectiveness of the TAC gateway in defending against denial of service (DoS) attacks, we launched a DoS attack at the network protocol layer from the untrusted client in Figure 2. Common DoS simulation tools require knowledge of the target IP address, but as previously demonstrated the TAC gateway effectively cloaks the IP address for our SDN controller. For test purposes, we assume that an attacker has somehow obtained the SDN controller IP address through outside channels (perhaps a spear phishing attack on the network administrator) and we proceed to launch a DoS attack against the controller. Using a standard tool such as Low Orbit Ion Cannon (LOIC), we launched an attack against the IP address of the gateway data port, management port, and SDN controller. All packets were blocked by the TAC gateway without providing any additional intelligence to the attacker.

The gateway can also be used in a Layer 3 operating mode, which performs NAT for selected ports on the gateway. This is useful in cloud computing environments, allowing the gateway to present a public IP address on its client facing, untrusted port and a private IP address on its trusted port. In this case, the insertion and authentication of tokens is performed before NAT. We have demonstrated this approach in a public cloud deployment, similar to figure 2 except that the protected resource was located within an Amazon Web Services cloud. The cloud service provider infrastructure is located on the right-hand side of figure 2 (replacing the ESXi network), with the public Internet acting as the untrusted network and the cloud users on the left side of figure 2. Public IP addresses are used on ingress ports facing the untrusted network and the cloud service provider's protected resource connect to a trusted egress port.

The virtual TAC gateways were also tested in a enterprise-class environment using the IBM zCloud (a cloud based on a highly virtualized IBM Z Systems mainframe). These servers are commonly used in public, private, and hybrid cloud environments for Fortune 500 applications (particularly in the financial markets) as well as within cloud service providers such as SoftLayer. As shown in figure 3, an IBM model z13 enterprise server was provisioned into four logical partitions, with two partitions running the z/OS operating system, and two partitions running zLinux. For each operating system, one partition served as the protected resource while the other served as the trusted host. All four partitions share common physical network interfaces, provided by an Open System Adapter (OSA) card running 1 Gbit/s Ethernet. The virtual appliances were hosted in two additional logical partitions (LPARs), interconnected with the protected resources, trusted hosts, and OSAs as shown in figure 3. Additional OSA cards were provisioned to serve as interfaces for the network management ports on the virtual appliances.

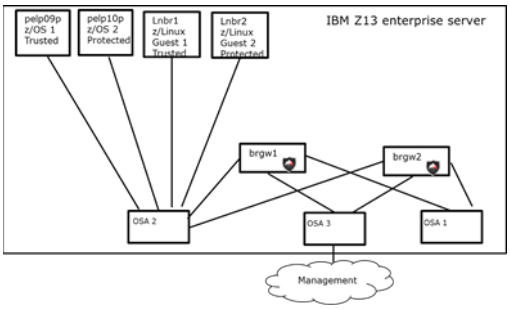

Figure 3 – Enterprise server use case test configuration

Three use cases were tested using this configuration. First, the gateways were configured to allow connectivity only between Linux partitions 1 and 2. We confirmed normal operation of resource connectivity including SSH, SCP, iperf, sftp, and wget functions, and verified that untrusted hosts such as zOS-1 could not access the Linux partitions. Second, the gateways were configured to allow connectivity between Linux-1 and zOS-2 partitions (note that the gateway authentication is independent of the operating system running in either the supplicant or the trusted resource). As before, we verified basic functionality (including multiple sftp file transfers between the trusted host and protected resource) and confirmed that other partitions, such as zOS-1, could not access the protected \resource in this configuration. Third, the gateway was configured to allow connectivity between the two zOS partitions. As in the previous use cases, we verified basic functionality (including multiple sftp file transfers between the protected resource and trusted hosts) and confirmed that the Linux-1 partition was unable to access protected resources in this configuration. These three test cases established that the gateways could be configured to enable or disable applications running between any two partitions on the same physical server, even if the gateway itself is hosted in a partition on the same physical server. This approach directly supports the zero-trust architecture we intended to implement.

The gateway functionality was further demonstrated in a pre-production test environment at Marist College, part of the New York State Cloud Computing and Analytics Center (CCAC).

In this case, the network spans multiple buildings on the campus MAN (about 1 km apart) as shown in figure 4. In the first building, ten sysadmin terminals were interconnected to a Layer 2 switch, and one switch port was connected to the insertion gateway. In this configuration, all terminals connected to the trusted switch port receive authentication tokens and will be allowed to access the protected resource. In a second building, the second gateway was configured in Layer 3 mode, which was then attached to the protected resource (a sysadmin application (Syswiki) running in a SUSE Linux guest VM on an IBM z144 enterprise server (mainframe).

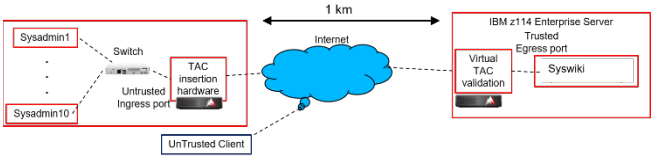

Figure 4 – syswiki BlackRidge

Using this configuration, we verified that the gateway allowed the Sysadmin trusted host terminals to perform operations including SSH, scp, and http post/get functions to the protected resource. We also verified that other terminals (untrusted hosts) on the same network were unable to access the protected resource when the gateways were set to Enforce mode, but could access the protected resource when the gateway was put into Bridge mode. The Sysadmins showed no measurable degradation in response time or performance of the protected resource application with and without the gateway operating in enforce mode.

Further, there was no measurable performance impact when accessing other resources on the campus network or accessing Internet resources when the gateway was in Enforce or Bridge mode. We also conducted a port scan of the gateway using Nmap/Zenmap tools from an untrusted terminal, and were unable to identify any open ports or fingerprint the Syswiki when the gateways were in Enforce mode.

4. Optical Bypass Switch Testing

For the next phase of testing, we incorporate a high-speed optical bypass switch into the system architecture, so that in the event of a hardware or software failure in the gateway, the TAC appliance will not obstruct network traffic. One advantage of an optical bypass switch is that it continues to pass traffic even if a power failure or firmware problem disables the functionality of the TAC gateway. Similar approaches have been suggested in high performance computing applications, where network traffic bypasses firewalls and other security features in order to maintain throughput for the application [12]. For these experiments, we used a 10 Gbit/s traffic stream while evaluating the impact of protection switching on the gateway.

In our environment, the TAC gateway incorporated a 10 Gbit/s multimode Silicom brand optical switch (model PE210G2BP) with optical bypass switch driver ixgbe ver 4.3.15. The optical bypass testbed is shown in figure 5. We configure two ESXi server instances, each driving 10 Gbit/s of traffic (loadgen 1 and loadgen 2, respectively). These servers run the Iperf benchmark test, enabling us to test switch times and throughput in both directions. As shown, the virtual switch on server loadgen1 connects to an untrusted port on the TAC gateway, while the virtual switch on loadgen2 connects to the corresponding trusted port.

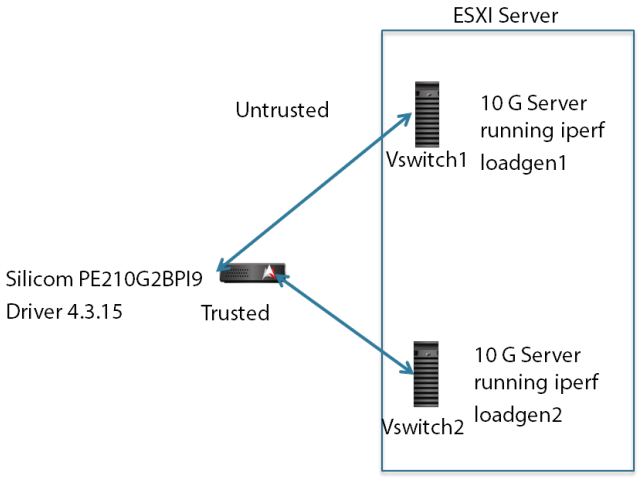

Figure 5 – Optical Bypass Switch testbed

Each test was repeated five times to obtain a meaningful average result, as shown in table 2. We can see that the optical bypass can sustain a data rate of about 8.3 Gbit/s while running Iperf, and about 8.9 Gbit/s in bypass mode (as expected, the bandwidth in bypass mode is somewhat higher).

Table 2 – optical bypass switch sustainable bandwidth measurements

Bypass mode, Gigabits per second	Iperf, Gigabits per second
8.88	8.39
8.95	8.24
8.92	8.33
8.91	8.05
8.85	8.31

In order to determine how long it would take for the optical bypass to tansfer traffic under different conditions, another server was inserted as a tap on the untrusted link, as shown in figure 6. By running WireShark on a 10 Gbit/s interface, this server is able to measure the effective switching times. A representative WireShark trace, shown in figure 7, confirms that the gateway bypass switch exhibits 110 ms recovery time upon a power failure to the gateway. We found that recovery time was faster when simulating a firmware outage (53 ms) and slightly longer when the TAC cloaking function was suspended due to a firmware problem (257 ms).

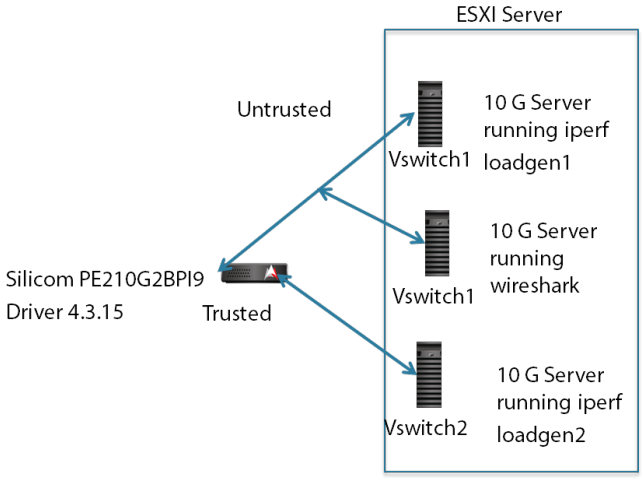

Figure 6 – Optical Bypass swithing time test bed

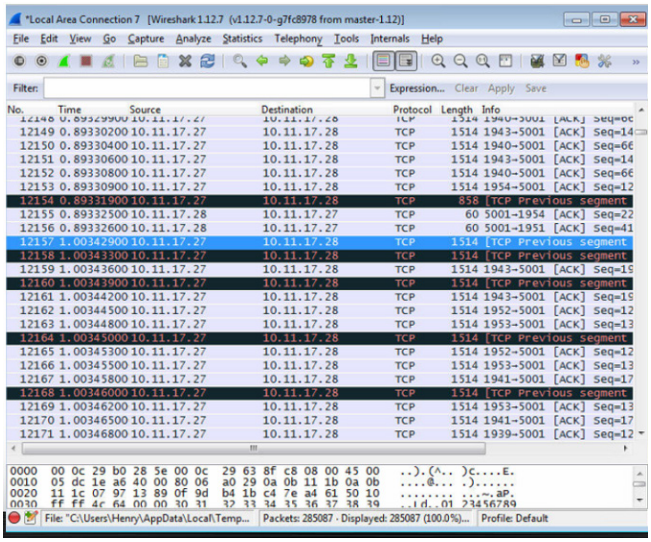

Figure 7 – Representitive WireShark trace from Optical Bypass switching time test bed

Finally, we reconfigured the gateway to conduct syslog benchmark testing, as shown in figure 8. The WireShark server is now monitoring traffic through the TAC gateway with optical bypass. We can run an nmap port scan simultaneously on servers loadgen1 and loadgen 2 using this configuration. For this testing, we configured the gateway to automatically blacklist any IP address which makes more than 100 access attempts in 60 seconds (an effective defense against some types of brute force dictionary attacks). We can keep an address blacklisted for different periods, ranging from 30 seconds to hours or even 24 hours. A blacklisted address generates a specific message type in the syslog, so we can easily determine the number of blacklist events in a given period.

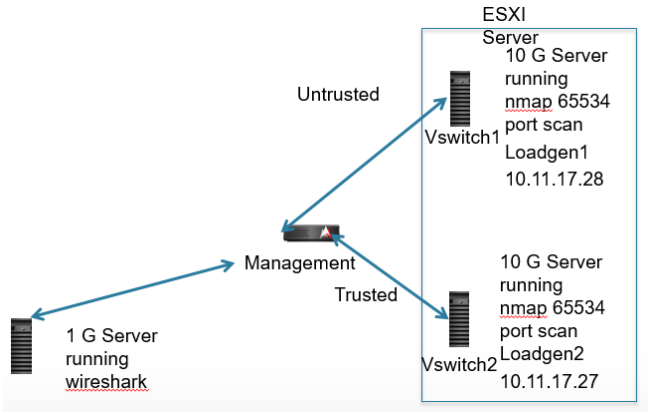

Figure 8 – Optical Bypass syslog test bed

For our tests, the initial default setting was a rate limit of 10 messages within 5 seconds, or a peak rate of 5 Mbit/second. Based on existing data about typical syslog messaging patterns [2, 3, 12], we expect the performance of our system to be well within normal commercial operating parametes. The peak data throughput measured during this experiment was 14 Mbit/second.

5. Conclusions

The growing cybersecurity treat requires an architectural redesign of the data center network, based on the principles of an explicit zero trust network. We have demonstrated several principles of zero trust using a transport access control system, based on a steganographic overlay, which embeds authentication tokens in the TCP packet request and first-packet authentication. The system was tested on both x86 and Z Systems platforms in private cloud environments, and using AWS in a public cloud environment. This system can provide enhanced security in both enterprise computing and cloud environments as part of a defense-in-depth strategy and prevents unwanted fingerprinting of protected resources. An optical bypass switch was also characterized as part of a high availability architecture.

Conflict of Interest

The authors declare no conflict of interest.

Acknowledgment

The authors gratefully acknowledge the support of the National Science Foundation (NSF) grant 1541384, Campus Cyberinfrastructure – Data, Networking and Innovation Program (CC-DNI), per NSF solicitation 15-534, for the project entitled CC-DNI (Integration (Area 4): Application Aware Software-Defined Networks for Secure Cloud Services (SecureCloud). The authors also gratefully acknowledge the support of Marist College and the New York State Cloud Computing and Analytic Center (CCAC).

References

[1] C. DeCusatis, P. Liengtiraphan, A. Sager, and M. Pinelli, “Implementing zero trust networks with transport access control and first packet authentication”, Proc. IEEE International Conference on Smart Cloud 2016, New York, NY (November 18-20, 2016)

[2] “Cisco 2017 annual security report”, published by Cisco System Inc., https://www.cisco.com/web/offer/gist_ty2_asset/Cisco_2017_ASR.pdf . (Last accessed February 9, 2015)

[3] “IBM X-Force trend and risk report”, published by IBM Corporation, October 2013http://www-03.ibm.com/security/xforce/downloads.html (last accessed December 18, 2015)

[4] NIST report, “Developing a framework to improve critical infrastructure cybersecurity”, submitted by Forrester Group, 18 pp. http://csrc.nist.gov/cyberframework/rfi_comments/040813_forrester_research.pdf (April 2013; last accessed January 5, 2015).

[5] R. Smith, Elementary Information Security, 2nd edition, Jones and Bartlett, Burlington, MA (2016)

[6] S. Oriyano, Hacker Techniques, Tools, and Incident Handling, 2nd edition, Jones and Bartlett, Burlington, MA (2014)

[7] S. Cherukuri, “NFV architecture and orchestration for cloud based virtual managed services”, Cisco Live 2015 paper BRKSDN-2065 https://www.ciscolive.com/online/connect/sessionDetail.ww?SESSION_ID=83663&tclass=popup (last accessed May 20, 2016)

[8] M. Casado, N. Foster, A. Guha. “Abstractions for Software-Defined Networks”. Vol. 57, pp 86-95. Communications of the ACM. 2014

[9] J. Kindervag and R. Harrison, “Orchestrate a zero trust network”, https://www.brighttalk.com/webcast/9591/186577?utm_campaign=communication_missed_you&utm_medium=email&utm_source=brighttalk-transact&utm_content=webcast (last accessed April 28, 2016)

[10] Amazon Web Services Identity and Access Management, April 2016 https://aws.amazon.com/iam/ (last accessed May 20, 2016)

[11] BlackRidge white paper, “Dynamic network segmentation”, August 2012http://www.blackridge.us/images/site/page-content/BlackRidge_Dynamic_Network_Segmentation.pdf (last accessed April 27, 2016)

[12] T. Hwang, “Secure Science DMZ using event driven SDN”, Proc. Internet 2 symposium, https://meetings.internet2.edu/media/medialibrary/2015/10/05/20151005-hwang-Cisco-Secure-Science-DMZ.pdf (October 5, 2015, Denver, CO; last accessed March 3, 2016).