Wide Area Network Acceleration in a High Assurance Enterprise

Kevin Foltz and William R. Simpson

Abstract— Bandwidth continues to be a problem for the active enterprise. One solution to bandwidth problems over long-haul distances with restricted bandwidth is the Wide Area Network (WAN) Accelerator. This accelerator works by tokenizing blocks of information that are sent multiple times in network traffic. Because many such communications include previously transmitted material, the accelerator traffic quickly damps out to transmissions that include tokens instead of the original communication. The tokens are reconstituted before delivery, and the receiver has a seamless connection and is unaware of the process. The acceleration is not without its drawbacks. The process does not work on encrypted traffic due to the random nature of encryption. For high assurance systems using an end-to-end paradigm, there are two main areas of concern. The first is security (how do we handle the decryption/re-encryption process?). The second is integrity (how do we maintain end-to-end integrity when encryption is broken?)

This paper discusses the current approach to WAN acceleration and the changes that are required by a high assurance end-to-end approach. The latter rely on a well-formed security paradigm for the enterprise.

Index Terms — Network Acceleration Appliance, Protection, IT Security, Encryption, Key Management, Wide Area Network, Integrity

I. INTRODUCTION

A WAN accelerator [1] is an appliance that optimizes bandwidth to improve the user's experience on a wide area network (WAN). The appliance, which can be a physical hardware component, a software program, or an appliance running in a virtualized environment, speeds up the time it takes for information to flow back and forth across the WAN by using compression and data deduplication techniques to reduce the amount of data that needs to be transmitted. An accelerator works by caching duplicate files or parts of files so they can be referenced instead of having to be sent across the WAN again. Many of the products have evolved beyond the core acceleration techniques. The WAN optimization controllers (WOC) further optimize the WAN link by accounting for known problems with common network protocols. Protocol optimization cleans up chatty protocols used in common enterprise standards such as Common Internet File System (CIFS), Microsoft Exchange, and even TCP/IP to eliminate the typical overhead found in these communication protocols. These optimizations require a deeper understanding of the protocols and can be accomplished only through significant collaboration with application vendors or reverse engineering by the WAN accelerator vendor.

WAN optimization encompasses [2]:

- Traffic shaping, in which traffic is prioritized and bandwidth is allotted accordingly.
- Data deduplication, which reduces the data that must be sent across a WAN for remote backups, replication, and disaster recovery.
- Compression, which shrinks the size of data to limit bandwidth use.
- Data caching, in which frequently used data are hosted locally or on a local server for faster access.
- Monitoring the network to detect non-essential traffic.
- Creating and enforcing rules about downloads and Internet use.
- Protocol spoofing, which is a method of bundling chatty protocols so they are, in effect, a single protocol.

WAN optimization vendors include Blue Coat Systems, Cisco, Expand Networks, F5 Networks, Juniper, and Riverbed Technology.

II. CURRENT WAN ACCELERATOR APPROACHES

Acceleration of unencrypted traffic is not of interest in the high assurance end-to-end paradigm, so we will concentrate on the steps involved in handling encrypted traffic. Initially, there is a need to decrypt the traffic since tokenization is ineffective for encrypted traffic. This is due to the randomization of bit streams that are a property of encryption. This is normally done by passing the private key of the server to the server-side WAN accelerator appliance, as shown in Fig. 1.

As shown in Fig. 2, the Transport Layer Security (TLS) keys are computed and then passed between the accelerator units. The premaster secret keying material that is passed is protected by the private key of the server. The sharing of the server’s private key allows the server-side appliance to extract the premaster secret and compute the session’s master secret, just as the endpoints compute it. This is used to generate the encryption keys and Message Authentication Code (MAC) secrets for each TLS connection within this session. As shown in Fig. 1, the session keys and MAC secrets are then transmitted to the client-side WAN accelerator.

Manuscript received January 1, 2015; revised January 21, 2015. This work was supported in part by the U.S. Secretary of the Air Force and The Institute for Defense Analyses (IDA). The publication of this paper does not indicate endorsement by any organization in the Department of Defense or IDA, nor should the contents be construed as reflecting the official position of these organizations.

K. Foltz is with the Institute for Defense Analyses. (email: kfoltz@ida.org)

William R. Simpson is with the Institute for Defense Analyses, 4850 Mark Center Drive, Alexandria, Virginia 22311 USA and is the corresponding author phone: 703-845-6637, FAX: 703-845-6848 (e-mail: rsimpson@ida.org).
A MAC is a cryptographic checksum on data that use a shared secret key to detect both accidental and intentional modifications of the data (for integrity). With this MAC secret and the encryption key and Initialization Vectors (IV), the client-side WAN accelerator can send TLS content that can be decrypted and validated properly by the client.

Fig. 1 shows the basic flows. This includes the logical end-to-end client-to-server TLS connection and its three secure component connections to, between, and from the WAN accelerators. It also includes TLS sessions to provide the session keys to the client-side accelerator, to transmit the tokenized packets to the client-side accelerator, and optionally to keep the tokenization stores synchronized. Although shown as separate TLS connections between the same endpoints, these may instead be implemented as separate logical flows within a single TLS connection. Aside from key management issues, the difference is not important.

Fig. 1 also shows the concern of integrity. The piecewise integrity of the component connections (I1, I2, and I3) does not amount to overall end-to-end integrity (E2E I).

\[ I_1 + I_2 + I_3 \neq E2E I \]  

(1)

In an end-to-end security paradigm, the goal is for the receiver to verify that the message sent from the server is identical to the message received at the client. Even when TLS MACs are included in all component connections, the integrity of the end-to-end transmission is not preserved unless we undertake some extraordinary measures. When a component connection retransmits data with a new MAC value, any changes made by that component to the original data will go undetected. In order to avoid explicit trust of the WAN processes, the TLS reconstruction of the server traffic by the client-side WAN accelerator must be modified. To understand this, we must examine the way in which TLS messages are formed.
III. AN ALTERNATIVE TO PRIVATE KEY PASSING

For most interactions using enterprise level security approaches, traffic content does not need to be inspected. Firewall functionality will still be available using the headers that are not encrypted. However, certain functions, including WAN acceleration, require content inspection. For these conditions we recommend an alternative to sharing private keys as follows:

1. Web application shares only the TLS session keys that are needed for each appliance to function

2. No shared private keys – each active entity has its own unique public/private key pair.

3. Web application is endpoint for browser requests.

Fig. 3 shows the alternative recommendation. HTTP traffic is encrypted using TLS from browser to web application. The WAN Accelerator uses the provided session keys to decrypt TLS traffic. The balance of the transaction works in the same way, with the exception of the integrity problem. For this we must examine the TLS package development process.

Fig. 4. TLS Transmission Steps
IV. INTEGRITY IN A TLS SESSION

The message breakdown and TLS packet construction [19-21] are shown in Fig. 4, including fragmentation of messages, addition of headers, compression, addition of the MAC and padding, and encryption. The MAC secret and the values that depend on it are indicated in orange. The encryption keys and IVs, and the values that depend on them, are indicated in red. The ciphertext has both red and orange, since it relies on both the MAC secret and the encryption keys and IVs.

Fig. 5 shows the reverse process, where the receiver decrypts, validates the received MAC, decompresses, extracts content, and defragments into the original messages.

One important feature of TLS is that the encryption key/IV and MAC secret are separate values with separate functions. With only the encryption key and IV, an intermediate node can view the content but not modify it. It cannot compute a valid MAC for the modified content without the MAC secret. Also relevant is that the fragmentation boundaries of the original messages are important for MAC computation. Identical messages that are fragmented differently will have different MACs since the MACs depend on the fragments to which they are appended.

Fig. 6. Flow from Server Side

---

**Fig. 5. TLS Reception Steps**

---

**Fig. 6. Flow from Server Side**
V. FLOWS IN A HIGH INTEGRITY SYSTEM

The overall recommended flows for the high assurance WAN Accelerator on the server-side are provided in Fig. 6. The TLS session keys are passed to the server-side WAN accelerator as shown in Fig. 3 and subsequently passed on to the client-side accelerator through the WAN Server / WAN Client TLS. The primary addition is the transmission of the original message fragmentation, padding, and MACs. These are needed to make an identical reconstruction of the original message. The original MACs can be added to the reconstructed message, for which the targeted end-point has the MAC key.

The original TLS MAC digests, paddings, and fragmentation are used to construct TLS packets in lieu of the values that would normally be computed by the TLS protocol. The client-side flows are given in Fig. 7 and show the same transmission of the fragmentation, paddings, and MACS for the reconstruction of the message on the server-side.

To ensure end-to-end integrity it is necessary that the MAC secret is not available to the WAN accelerator. The receiving WAN accelerator node uses the original content, as received and reconstructed from the WAN connection, the original TLS fragmentation information, and the associated MAC and padding values to construct valid TLS messages to the receiving endpoint. Because only the sending endpoint has the MAC secret, the receiving endpoint has an end-to-end integrity guarantee.

VI. SUMMARY

We have reviewed the basic approaches to WAN Acceleration security when dealing with encrypted traffic. These have been found lacking in the specific areas of key protection, and message integrity. Key protection is lacking in that out-of-band passing of the private keys of the servers violate specific security tenants. Message integrity is lacking in that piecewise integrity is substituted for overall end-to-end integrity. We have also described the high assurance architectures and protection elements they provide. In order to preserve the high assurance key security and integrity elements, changes to the basic flows must be made. Key security can be maintained by passing only the session key which provides a lower risk than passing the private key of the server and maintains the unique identity of the server. Overall message integrity can be maintained by conspicuously reconstructing the original messages, including the message authentication codes. The development of WAN accelerator mechanisms does not require distribution of a private key which is often done with today’s appliances. The distribution of private keys is a fundamental violation of a high assurance model. What remains is the need for high reliability and secure code for passing of session keys, as well as secure means of transporting TLS plans and MACs for the establishment of service interfaces on the appliances. Pilots are in the process of being undertaken, and we expect some modifications as we learn more about how these devices work.

This research is part of a body of work for high assurance enterprise computing using web services. Elements of this work include bi-lateral end-to-end authentication using PKI credentials for all person and non-person entities, a separate SAML credential for claims based authorization, fully encrypted at the transport layer and a defined federation process. Many of the elements of this work are described in [22-35].

REFERENCES

Proceedings of the World Congress on Engineering 2015 Vol I


