Distributed versus Centralized Protection Schema for the Enterprise

Coimbatore Chandersekaran, Kevin Foltz and William R Simpson

***Abstract*— Entities in the enterprise are deployed with a standard configuration. Over time, patches, updates, new software versions, and mistakes or malicious activity all lead to deviations across the enterprise from this standard baseline. Malicious or unknown software on a system can cause harm or unexpected behavior.**

To mitigate these problems where possible, and help fix them in other cases, an enterprise plan for quality of protection is needed. This involves eliminating certain actions on machines that could harm the machine itself or the enterprise. The level of protection is dependent upon the type of enclave (an enclave is defined as a collection of entities with a common set of security and assurance mechanisms in place). Certain mitigations will be exercised based upon the cyber environment and enclave, and they may be exercised in different ways when communication is needed across enclaves of differing security and assurance. Mitigations include virus scanners and disabling of devices or interfaces. These mitigations also involve identifying and fixing issues that were not stopped. This requires a central visualization of the enterprise to quickly identify potential issues and a method of remotely taking action to either fix the affected system or freeze it until further action can be taken.

This paper discusses the current approach to centralized monitoring of communication as opposed to a more distributed approach. The latter relies on a well-formed security paradigm for the enterprise.

***Index Terms* — Appliance, Traffic Inspection, Protection, IT Security, Encryption, Key Management**

I. INTRODUCTION

Enterprise protection is based upon the device, the environment, and the enclave type. An enclave is a collection of entities and assurance mechanisms that uniformly employ the same security. Protection includes on-device software, in-line monitoring of communications and the particular security model that provides confidentiality, integrity, and availability. Many times this security model is compromised in trying to provide basic levels of protection. These compromises may include policy-based instructions to configure intrusion detection devices that provide the capability for selecting which attacks are being monitored. These policy selections can provide capabilities to select what responses will be taken for each detected intrusion.

II. CURRENT PROTECTION APPROACHES

Elements involved in implementing Quality of Protection are numerous and complicated. A wide range of appliances are used to provide functionality ranging from quality of service to the user or quality of protection of network resources and servers. These appliances are often placed in-line and some require access to content to provide their service. Figure 1 illustrates how these appliances are installed between the user and the application.

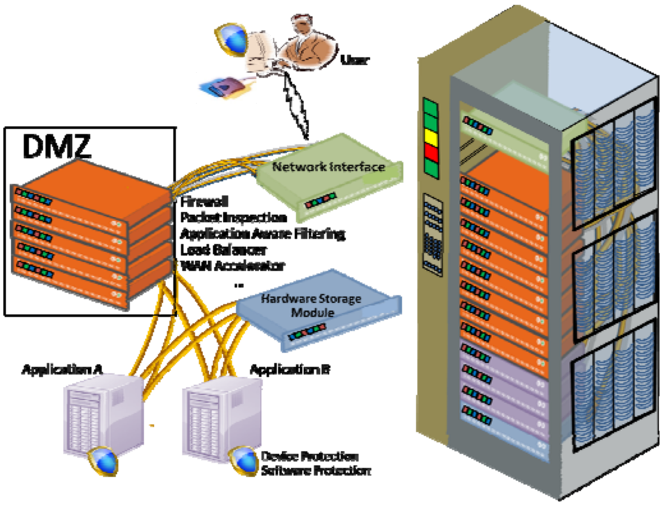

Fig. 1 End-Point Access

The number of appliances can be quite high [1-22]. Below is a partial list of functional types:

- Header-based scanner/logger:
  - Views only unencrypted portion of traffic
  - Synchronous or asynchronous operation
  - Scans for suspicious behavior, logs traffic
- Content-based scanner/logger:
  - Views decrypted content
  - Synchronous or asynchronous operation
  - Scans for suspicious behavior, logs traffic and/or content
- Header-based firewall:
  - Views only unencrypted portion of traffic
  - Synchronous operation
  - Scans for and blocks suspicious behavior
- Content-based firewall – block only:
  - Views decrypted content
  - Synchronous operation
  - Scans for suspicious behavior and blocks (terminates) connection
- Content-based firewall – modifies malicious content
  - Views decrypted content
  - Synchronous operation
  - Scans for suspicious content, and blocks connection or removes suspicious content while preserving the connection

Manuscript received April 17, 2014; revised May 18, 2014. 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

Coimbatore Chandersekaran and K. Foltz are with the Institute for Defense Analyses.(email: cchander@ida.org, 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)

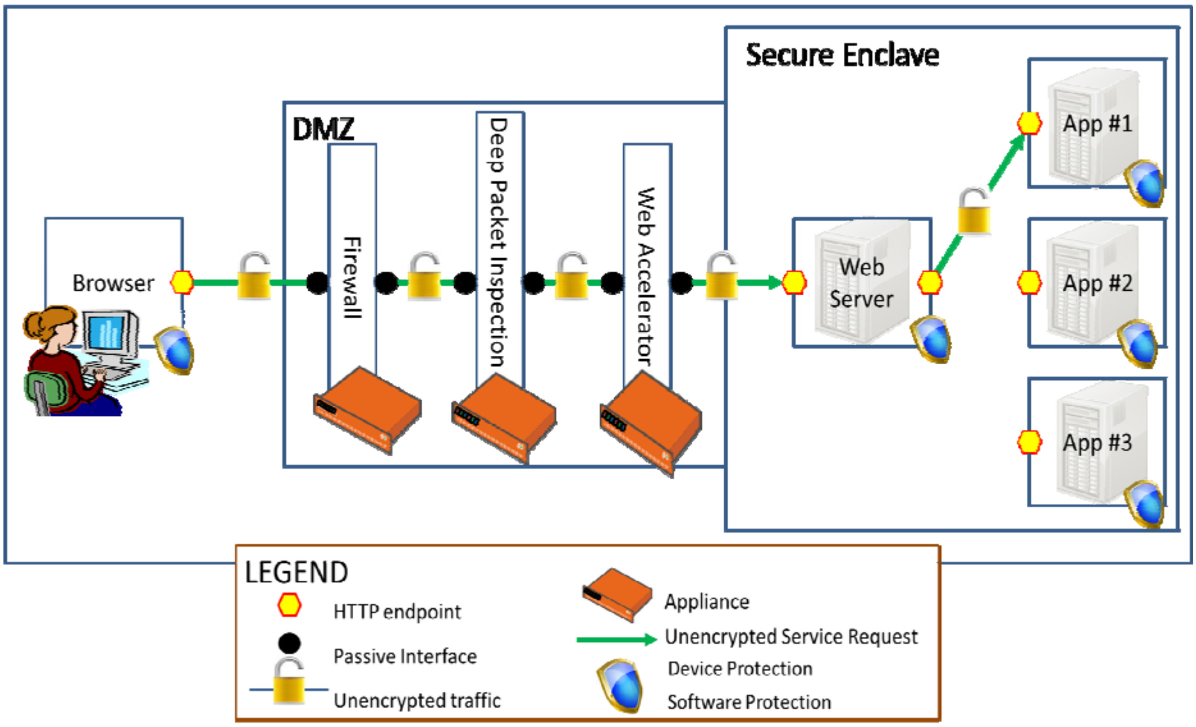

Fig. 2 Current Paradigm for Unencrypted Traffic

- Web accelerator:
  - Views decrypted content
  - Synchronous operation
  - Modifies content for performance
- WAN accelerator:
  - Views decrypted content
  - Multi-party system
  - Synchronous operation
  - Modifies content representation between parties, but no end-to-end modification
- Load Balancers:
  - Distributes load among destination end points to improve throughput and reduce latency.
  - May decrypt content:
    - ✓May combine encrypted flows through an "encryption accelerator"
    - ✓May distribute content by request to different servers based on load
    - ✓These load balancers are considered active entities.
  - May not decrypt content:
    - ✓Using "sticky" or end point balances may route all requests from an entity to the same server.
    - ✓These load balancers are considered passive entities.

Each of the appliances above offers some functionality and increases the threat exposure. None of these are bullet-proof from a security standpoint and they do increase the threat surface and the vulnerability space. Use of any appliance must be balanced by the increased functionality and the increased vulnerability. The situation is further complicated by vendor offerings of load balancers with firewall capability, "smart" accelerators that scan content, and software only offerings that provide most of these functionalities in a modular fashion.

A. *Current - Unencrypted Traffic*

To understand the current paradigm, a review of what is done through a portal for unencrypted traffic is provided. HTTP traffic is unencrypted from browser to portal, and unencrypted from portal to web applications providing content.

1. Examples:
   a. www.amazon.com
   b. www.va.gov
   c. www.af.mil
2. Man-in-the-middle (MITM) model for appliances (e.g., firewalls, deep packet inspection, accelerators) that perform analysis of headers and content (IP, TCP, HTTP, HTML, XML, JavaScript, etc.)
3. Portal is the endpoint for browser requests.

The process is shown in figure 2.

Note that while the traffic is unencrypted, the requester may or may not be authenticated using a smart card with public key infrastructure credentials. The traffic is pulled in-line through a number of appliances to protect ports, and inspect content. We have included a web accelerator in this figure, because they have a number of characteristics in common with these other appliances. Other appliances, including load balancers and WAN accelerators may also be included in this stack. Load balancers and some firewalls may be treated a passive entities in this treatment. WAN Accelerators are not covered in this paper.

B. *Current - Encrypted Traffic*

When traffic is encrypted, the same basic approach is adapted to handling traffic inspection. HTTPS traffic is successively decrypted and re-encrypted when needed. End-to-end HTTPS traffic is encrypted using Transport Layer Security (TLS) from browser to portal and using separate TLS sessions from portal to web applications

1. Example: a. https://www.mybank.com
2. MITM model for appliances
3. Some can function without decryption (e.g., firewall)
4. Some require decryption, using portal private key (e.g., deep packet inspection, accelerator)
5. Portal is the endpoint for browser requests.

The process is shown in figure 3.

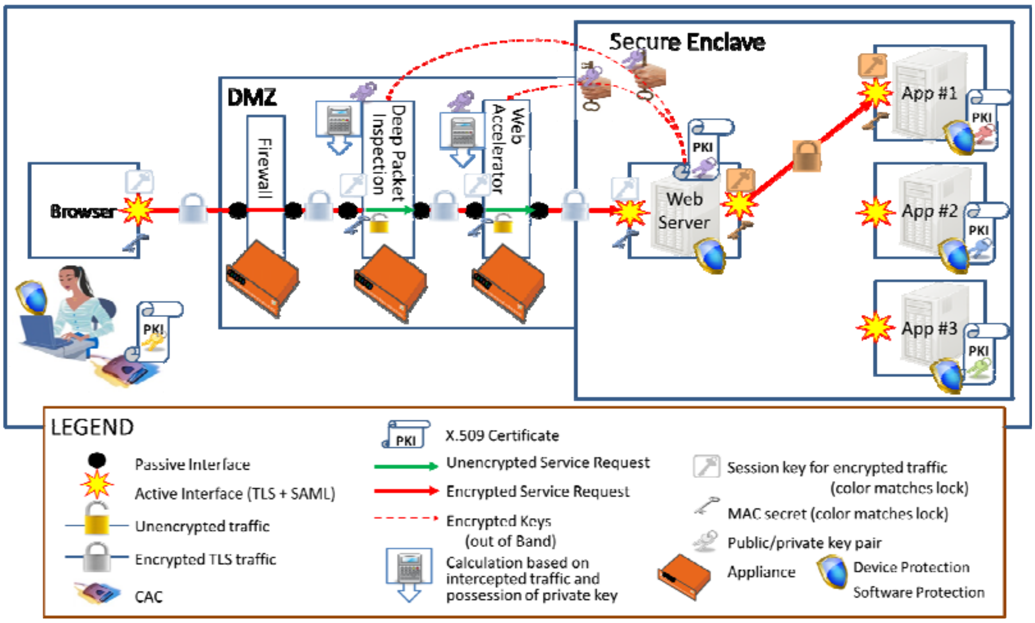
Secure Enclave
DMZ
Firewall
Deep Packet Inspection
Web Accelerator
Browser
Web Server
App #1
App #2
App #3
PKI
LEGEND
Passive Interface
Active Interface (TLS + SAML)
Unencrypted traffic
Encrypted TLS traffic
CAC
X.509 Certificate
Unencrypted Service Request
Encrypted Service Request
Encrypted Keys (out of Band)
Calculation based on intercepted traffic and possession of private key
Session key for encrypted traffic (color matches lock)
MAC secret (color matches lock)
Public/private key pair
Appliance
Device Protection
Software Protection

Fig. 3 Current Paradigm for Encrypted Traffic

In order to be able to decrypt the packages, the private key and initialization vectors (IVs) from the portal are provided to the in-line appliances. As such, they see the handshake and exchange and have access to the keying material sufficient to compute session and Message Authentication Code (MAC). A MAC is a cryptographic checksum on data that uses a session key to detect both accidental and intentional modifications of the data (for integrity). While sharing of private keys is an easy way to provide the appliance's visibility to the content, it is a singularly bad idea from a security standpoint. Loss of a private key will compromise identity and all sessions and will allow an adversary to impersonate the entity. Further, in high assurance systems, the private keys are locked in a Hardware Storage Module (HSM) and are not shareable. Loss of a session key will entail loss of session confidentiality only. The situation is a bit more complicated in that some of the devices need to see content and some do not. For example, the following data is available for all encrypted packets in the header without decryption:

1. IP Header:
   a. Time to live
   b. Source IP
   c. Destination IP
   d. IP version and flags
2. TCP Header:
   a. Source port
   b. Destination port
   c. Sequence number
   d. Acknowledgment number
   e. Windows size
   f. TCP flags
3. TLS Handshakes and Header:
   a. Version, cipher suite, compression algorithm, extensions
   b. Server certificate and partial chain to root CA
   c. Client certificate and partial chain to root CA
   d. Session IDs, client and server random values
   e. Alerts (full content) sent prior to encryption
   f. Message types and lengths
   g. Renegotiations
   h. Trusted CA list at server
   i. Client supported cipher suites list and preference
   j. Client supported compression algorithms
4. Firewall Capabilities not requiring Decryption:
   a. Blacklist/Whitelist based on IP/port numbers
   b. Block unacceptable TLS versions, cipher suites
   c. Block insecure TLS renegotiations
   d. Block compromised or unknown CAs

III. AN ALTERNATIVE TO PRIVATE KEY PASSING

For most interactions using Enterprise Level Security approaches, traffic does not need to be inspected. The firewall functionality will still be available using the headers that are not encrypted. However it is recognized that certain circumstances, including cyber-attack indications and/or insider suspicions, and others may require content inspection. For these conditions we recommend an alternative to the sharing of private keys as follows:

HTTP traffic is encrypted using TLS from browser to web application; gateway router at enclave boundary provides access to all internal IP addresses

1. Web app shares only the TLS session keys that are needed for each appliance to function.
   a. Firewall: no keys needed, uses headers.
   b. Deep Packet Inspection: encryption key only, no modification needed.
   c. Accelerator: encryption key and MAC secret, to inspect and modify content.
2. No shared private keys – each active entity has its own unique public/private key pair.
3. Web application is endpoint for browser requests.

Figure 4 shows the alternative recommendation. The figure illustrates the importance of key management. The user session (gray locks and keys) have a life equal to the user session. The web server to appliance keys (blue and brown locks and keys) have a fairly long life to accommodate passing of multiple session keys. The public and private keys have a life specified by the certificate.

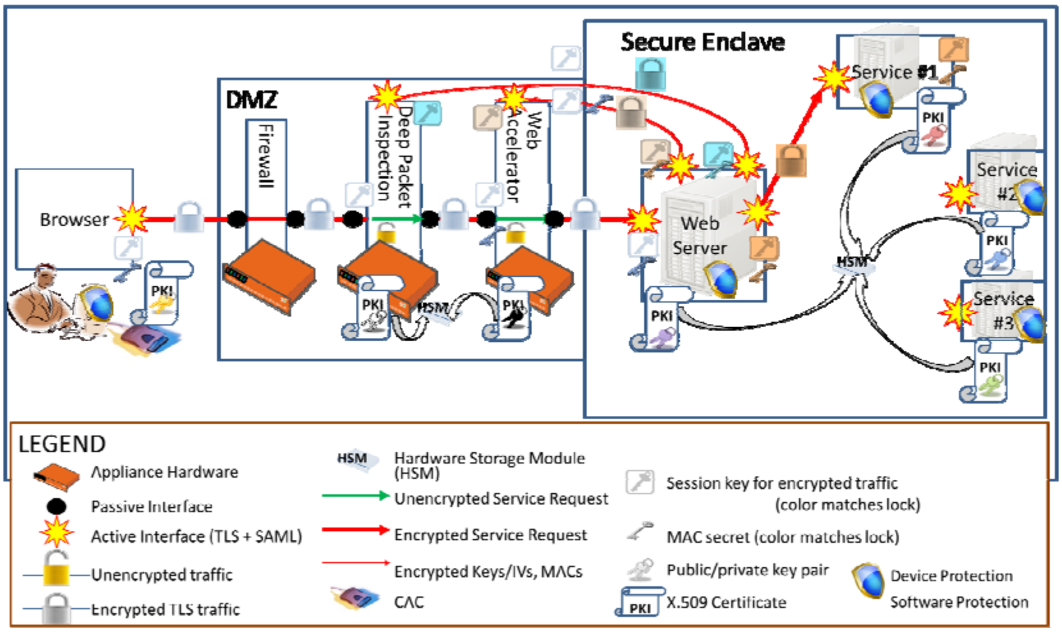

Fig. 4 Alternative Encrypted Web Server Communications

IV. A DISTRIBUTED PROTECTION SYSTEM IN A HIGH ASSURANCE MODEL

The protection system has the capability to monitor, filter, and shut down traffic to given ports. It scans for malicious code. It examines incoming and outgoing traffic for anomalies or known exploits. The protection system checks ports and protocols and can perform other functionality.

The protection system acts in the security context of the endpoint for both requester and provider and examines not only the encrypted traffic but also the unencrypted TLS traffic for malicious behavior or code. This requires access to the unencrypted traffic as well as the encrypted traffic. Not all of the checks are provided by the protection system. Figure 5 walks through checks in a high assurance enclave provided by the protection system, the server handlers, the service handlers and the service itself, minimizing the need for in-line appliances.

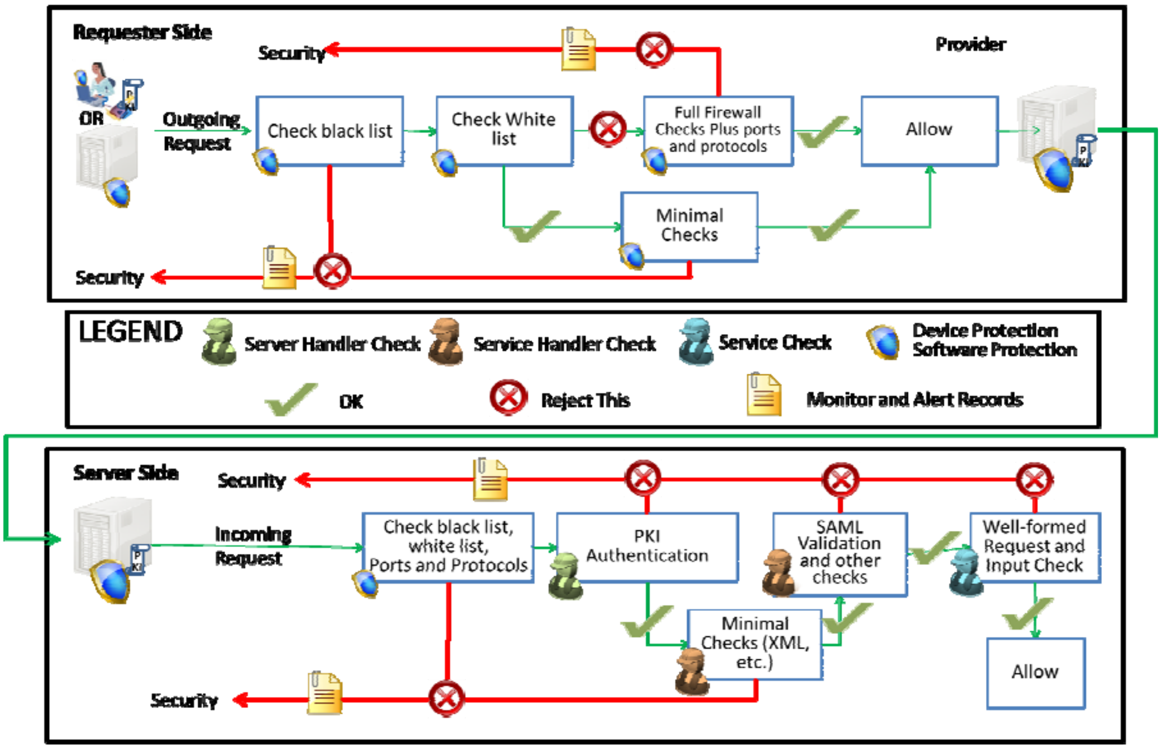
Requester Side
Security
Provider
OR
Outgoing Request
Check black list
Check White list
Full Firewall Checks Plus ports and protocols
Allow
Minimal Checks
Security
LEGEND
Server Handler Check
Service Handler Check
Service Check
Device Protection Software Protection
OK
Reject This
Monitor and Alert Records
Server Side
Security
Incoming Request
Check black list, white list, Ports and Protocols
PKI Authentication
SAML Validation and other checks
Well-formed Request and Input Check
Minimal Checks (XML, etc.)
Allow
Security

Fig. 5 Distributed Protection Provided without In-Line Appliances

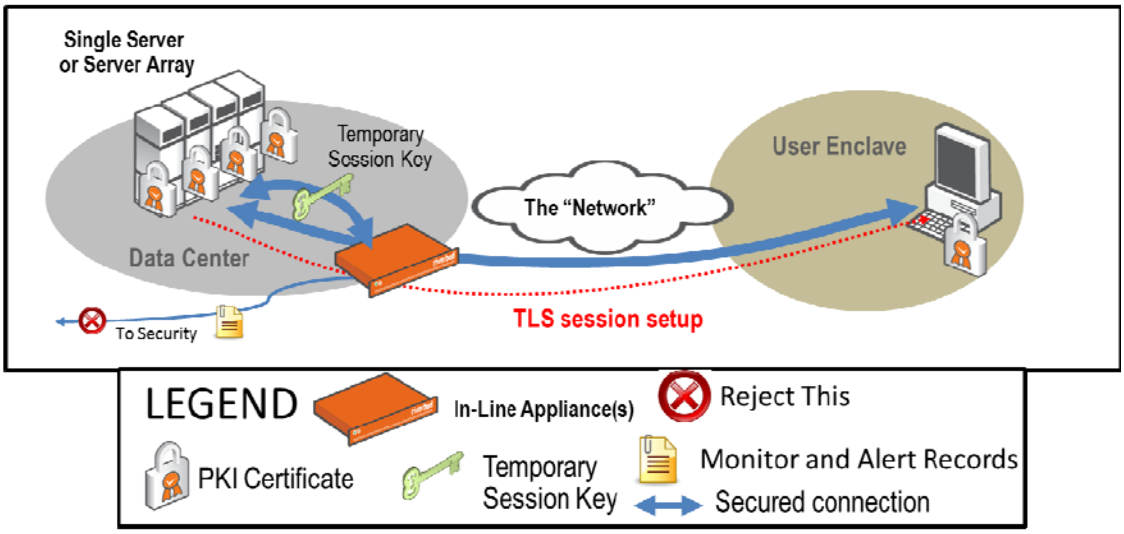

Fig. 6 In-Line Appliance Functionality

A. *Appliance Functionality In-Line*

For enclaves that are characterized by full bi-lateral based authentication, private keys should be required to be under the control of hardware storage modules, and the end-to-end paradigm cannot be broken by a passive entity. In-line appliances may be configured as active entities but this is not recommended. The in-line system can only observe headers, apply white and black lists and pass through encrypted content without inspection unless it is provided assistance from the service as shown in figure 6. This assistance is in the form of passing the session keys where appropriate.

The in-line device may decrypt and inspect incoming traffic. It should either deny the communication (with appropriate logging of information and/or security alerts as appropriate) or pass the communication (unmodified) on to the server. The device may also scan outgoing traffic.

B. *Appliance Functionality as a Service*

The appliance system may be used as a service by the application and as such follows full bi-lateral authentication and the TLS security paradigm as shown in figure 7.

In order to prevent attackers from gaining access to networks, each device must monitor DHCP requests and report to the central monitor all such requests. This provides listeners throughout the network that allow the central monitor to quickly identify the requesting entity, determine whether it is a known and trusted device or a rogue entity, and take action accordingly.

Any system that is found on the network, through DHCP or other traffic, must identify itself to the protection system before any services are provided to it. This identification is through protection system communications, through which each device authenticates to the central authority and also authenticates the central authority. All such traffic uses end-to-end security, and all devices and their protection systems are registered with enterprise. Unknown entities are not given services and are marked as rogue, which enables local devices to ignore their traffic.

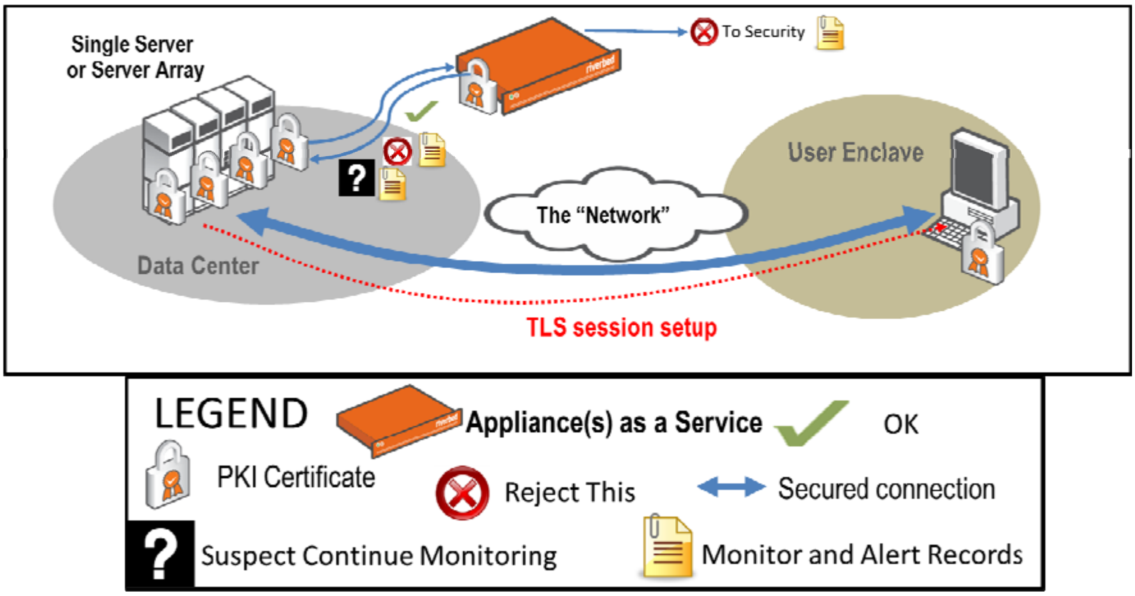

Fig. 7 Appliance Functionality as a Service

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

V. SUMMARY

We have reviewed the basic approaches to communication protection in computing environments. We have also described high assurance architectures and protection elements they provide. In many instances the high assurance elements provide equivalent protection. In cases where additional protective measures are needed, we have provided two mechanisms for their incorporation. Neither of these mechanisms requires 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 private keys, or the establishment of service interfaces on the appliances. This work is part of a body of work for high assurance enterprise computing using web services. Elements of this work are described in [23-36].

REFERENCES

[1] Oppliger, Rolf (May 1997). "Internet Security: FIREWALLS and BEYOND". *Communications of the ACM* **40** (5): 94.

[2] Ingham, Kenneth; Forrest, Stephanie (2002). "A History and Survey of Network Firewalls" (pdf).

[3] Firewalls by Dr.Talal Alkharobi

[4] Ingham, Kenneth; Forrest, Stephanie (2002). "A History and Survey of Network Firewalls" (pdf). p. 4. Retrieved 2011-11-25.

[5] William R. Cheswick, Steven M. Bellovin, Aviel D. Rubin (2003). "Google Books Link". *Firewalls and Internet Security: repelling the wily hacker*

[6] Aug 29, 2003 Virus may elude computer defenses by Charles Duhigg, Washington Post

[7] Conway, Richard (204). *Code Hacking: A Developer's Guide to Network Security*. Hingham, Massachusetts: Charles River Media. p. 281. ISBN 1-58450-314-9.

[8] Chang, Rocky (October 2002). "Defending Against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial.". *IEEE Communications Magazine* **40** (10): 42–43.

[9] Virgílio Almeida , Azer Bestavros , Mark Crovella , Adriana de Oliveira, Characterizing reference locality in the WWW, Proceedings of the fourth international conference on on Parallel and distributed information systems, p.92-107, December 18-20, 1996, Miami Beach, Florida, USA

[10] Mehmet Altinel , Christof Bornhövd , Sailesh Krishnamurthy , C. Mohan , Hamid Pirahesh , Berthold Reinwald, Cache tables: paving the way for an adaptive database cache, Proceedings of the 29th international conference on Very large data bases, p.718-729, September 09-12, 2003, Berlin, Germany

[11] Amiri, K., Tewari, R., Park, S., and Padmanabhan, S. 2002. On space management in a dynamic edge cache. In Proceedings of the Fifth International Workshop on the Web and Databases (WebDB 2002) (Madison, Wisc.). ACM, New York, 37--42.

[12] Jesse Anton , Lawrence Jacobs , Xiang Liu , Jordan Parker , Zheng Zeng , Tie Zhong, Web caching for database applications with Oracle Web Cache, Proceedings of the 2002 ACM SIGMOD international conference on Management of data, June 03-06, 2002, Madison, Wisconsin [doi>10.1145/564691.564762]

[13] Apache HTTP Server Project. 2003. Apache HTTP server. http://httpd.apache.org/.

[14] BEA Systems. 2003. Weblogic application server. http://www.bea.com.

[15] CacheFlow. 1999. Accelerating e-commerce with CacheFlow internet caching appliances (a CacheFlow white paper).

[16] Cain, B., Spatscheck, O., May, M., and Barbir, A. 2001. Request-routing requirements for content internetworking. http://www.ietf.org/internet-drafts/draft-cain-request-routing-req-03.txt.

[17] K. Selçuk Candan , Wen-Syan Li , Qiong Luo , Wang-Pin Hsiung , Divyakant Agrawal, Enabling dynamic content caching for database-driven web sites, Proceedings of the 2001 ACM SIGMOD international conference on Management of data, p.532-543, May 21-24, 2001, Santa Barbara, California, USA [doi>10.1145/375663.375736]

[18] Challenger, J., Dantzig, P., and Iyengar, A. 1999. A scalable system for consistently caching dynamic web data. In Proceedings of the 18th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM) (New York, N.Y.). IEEE Computer Society Press, Los Alamitos, Calif., 294--303.

[19] Carlos Cunha , Azer Bestavros , Mark Crovella, Characteristics of WWW Client-based Traces, Boston University, Boston, MA, 1995

[20] ESI Consortium. 2001. Edge side includes. http://www.esi.org.

[21] S. Gadde , M. Rabinovich , J. Chase, Reduce, Reuse, Recycle: An Approach to Building Large Internet Caches, Proceedings of the 6th Workshop on Hot Topics in Operating Systems (HotOS-VI), p.93, May 05-06, 1997

[22] Erich Gamma , Richard Helm , Ralph Johnson , John Vlissides, Design patterns: elements of reusable object-oriented software, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1995

[23] William R. Simpson, Coimbatore Chandersekaran and Andrew Trice, Electronic Digest of the 2008 System and Software Technology Conference, "A Persona-Based Framework for Flexible Delegation and Least Privilege", Las Vegas, Nevada, May 2008.

[24] William R. Simpson, Coimbatore Chandersekaran and Andrew Trice, The 1st International Multi-Conference on Engineering and Technological Innovation: IMET2008, "Cross-Domain Solutions in an Era of Information Sharing", Volume I, pp.313-318, Orlando, FL., June 2008.

[25] Coimbatore Chandersekaran and William R. Simpson, World Wide Web Consortium (W3C) Workshop on Security Models for Device APIs, "The Case for Bi-lateral End-to-End Strong Authentication", 4 pp., London, England, December 2008.

[26] William R. Simpson and Coimbatore Chandersekaran, The 2nd International Multi-Conf.on Engineering and Technological Innovation: IMETI2009, Volume I, pp. 300-305, "Information Sharing and Federation", Orlando, FL., July 2009.

[27] Coimbatore Chandersekaran and William R. Simpson, The 3rd International Multi-Conf. on Engineering and Technological Innovation: IMETI2010, Volume 2, "A SAML Framework for Delegation, Attribution and Least Privilege", pages 303-308, Orlando, FL., July 2010.

[28] William R. Simpson and Coimbatore Chandersekaran, The 3rd International Multi-Conference on Engineering and Technological Innovation: IMETI2010, Volume 2, "Use Case Based Access Control", pages 297-302, Orlando, FL., July 2010.

[29] Coimbatore Chandersekaran and William R. Simpson, The First International Conference on Computer Science and Information Technology (CCSIT-2011), "A Model for Delegation Based on Authentication and Authorization", Springer Verlag Berlin-Heildleberg, Lecture Notes in Computer Science 20 pp.

[30] William R. Simpson and Coimbatore Chandersekaran, The 16th International Command and Control Research and Technology Symposium: CCT2011, Volume II, pp. 84-89, "An Agent Based Monitoring System for Web Services", Orlando, FL., April 2011.

[31] William R. Simpson and Coimbatore Chandersekaran, International Journal of Computer Technology and Application (IJCTA), "An Agent-Based Web-Services Monitoring System" Vol. 2, No. 9, September 2011, page 675-685.

[32] William R. Simpson, Coimbatore Chandersekaran and Ryan Wagner, Lecture Notes in Engineering and Computer Science, Proceedings World Congress on Engineering and Computer Science 2011, Volume I, "High Assurance Challenges for Cloud Computing", pp. 61-66, San Francisco, October 2011.

[33] Coimbatore Chandersekaran and William R. Simpson, Lecture Notes in Engineering and Computer Science, Proceedings World Congress on Engineering 2012, The 2012 International Conference of Information Security and Internet Engineering, Volume I, "Claims-Based Enterprise-Wide Access Control", pp. 524-529, London, July 2012.

[34] William R. Simpson and Coimbatore Chandersekaran, Lecture Notes in Engineering and Computer Science, Proceedings World Congress on Engineering 2012, The 2012 International Conference of Information Security and Internet Engineering, Volume I, "Assured Content Delivery in the Enterprise", pp. 555-560, London, July 2012.

[35] William R. Simpson and Coimbatore Chandersekaran, Lecture Notes in Engineering and Computer Science, Proceedings World Congress on Engineering and Computer Science 2012, Volume 1, "Enterprise High Assurance Scale-up", pp. 54-59, San Francisco, October 2012.

[36] Coimbatore Chandersekaran and William R. Simpson, International Journal of Scientific Computing, Vol. 6, No. 2, "A Uniform Claims-Based Access Control for the Enterprise", December 2012, ISSN: 0973-578X, pp. 1-23.