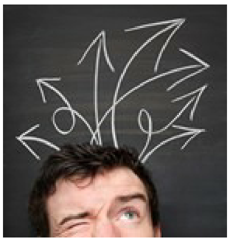
Infrastructure Architecture: Removing Blinders from Security Infrastructure

Lori MacVittie, 2011-26-10

Infrastructure architecture is often the answer to many of IT's most challenging issues.



It is a fact of IT that different businesses have different technical requirements in terms of security, processing, performance, and even storage. In many organizations, particularly those that transport sensitive personal or financial information, end-to-end encryption is a must. At first glance this seems to be a fairly simple thing – enable a secure transport from client to server and vice-versa and voila! But further exploration reveals that this isn't the case, primarily because it's never a straight shot between the client and the server – there are a variety of critical functions and processes that must be applied to exchanges in flight, many of which are impeded by the presence of encrypted traffic.

INFRASTRUCTURE ARCHITECTURE

One solution, of course, is to get rid of the encryption requirement and ensure that all functions and processes in the data path can perform their functions uninhibited.

That's rarely an option, however, so we need to use an architectural approach to provide those components with unencrypted traffic upon which they can perform their tasks in a way that preserves the security of the data, which is certainly the intent behind encryption in the first place.

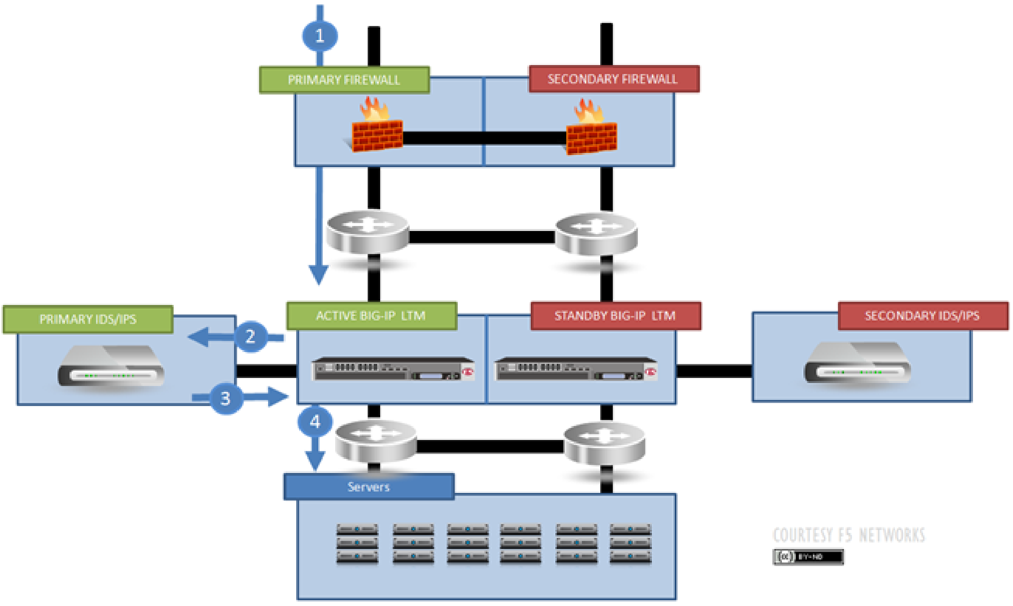
A side-arm (or one-arm) architectural configuration provides the answer to this problem, in conjunction with a terminating-capable application delivery controller. This, in itself, is somewhat ironic as load balancers (from which application delivery controllers evolved) were often deployed in one-armed configurations in the early days of web-enabling network architecture. Thus it is amusing that it is now the application delivery controller that serves as the "body" for a one-armed configuration for other infrastructure components.

Or maybe I just have a really odd sense of humor. That, too, is likely.

In any case, this popular infrastructure architecture solution is fairly straightforward, comprising an application delivery controller - provides load balancing services, multi-layer threat mitigation for the applications, and of course SSL termination and re-establishment – and the IDS/IPS infrastructure. The application delivery controller, acting as the topological endpoint for the client, terminates the SSL session and decrypts the data before sending it to the IDS/IPS for evaluation. When the data is returned, it is re-encrypted and traverses its normal path to the web/application servers for processing.

The details are as follows (some assumptions are made to keep this simple):

Logical Network Topology with Traffic Flow



1. Client request is made to the application (i.e. https://www.customer.com). This TCP/IP request is terminated by the application delivery controller as follows:

> a. TCP/IP Optimizations are applied for client-side traffic.
>
> b. SSL decryption occurs using CA-signed cert/key pairs as appropriate.
>
> c. A load-balancing decision is made for the web server per the configured algorithm.
>
> d. Persistence is applied as configured (typically encrypted HTTP cookie injection) to maintain “stickiness” of client to the selected webserver.
>
> e. HTTP is processed and optimized as appropriate (header injection/scrubbing, caching, compression, etc…)
>
> f. Application firewalling policy (via web application firewall services) is applied as desired

2. The application delivery controller initiates a separate TCP/IP connection to the chosen server. This connection (optimized for TCP/IP multiplexing and LAN characteristics) remains decrypted and exits towards the IDS/IPS appliance for inspection.

3. Return traffic from the IDS/IPS re-enters the application delivery controller after it has been inspected.

4. The application delivery controller re-encrypts the traffic and routes it to the appropriate webserver.

Return traffic flows through the same path in reverse. This architecture is fully redundant and would survive a failure of any given device in the path.

SERVICE-ORIENTED INFRASTRUCTURE

You’ll note that the data path flow from the application delivery controller to the IDS/IPS is similar to what we’d call in the application development world a “lookup” or a “callout".

From the admittedly high-level perspective of an architectural flow it’s a service call, the integration of an externally provided function / operation into an operational process. It’s service-oriented in theory and practice, if not actual implementation. It’s also near-stateless, with the routed flow of traffic implying the policy application rather than an explicitly stated instruction.

This is a simple architectural solution to a common problem, one that’s plagued IT since the introduction of encrypted communication as a standard practice. More often than not, the solution to many of IT’s problems can be found in a collaborative architectural approach. If cloud computing and virtualization are doing anything, it’s bringing this reality to the fore by forcing the conversation up the stack.

- Cloud Infrastructure Integration Model: Virtualization
- Cloud Infrastructure Integration Model: Bridging
- Cloud, open source, and new network models: Part 1
- Live Migration versus Pre-Positioning in the Cloud
- Cloud is an Exercise in Infrastructure Integration
- IT as a Service: A Stateless Infrastructure Architecture Model
- Cloud is the How not the What
- Cloud-Tiered Architectural Models are Bad Except When They Aren't

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com