Web Application Security at the Edge is More Efficient Than In the Application

Lori MacVittie, 2009-28-09

If one of the drivers for moving to cloud-based applications is reducing costs, you should think twice about the placement of application security solutions.

There's almost no way to avoid an argument on this subject so I won't tiptoe around it: web application security in the cloud is better accomplished at the edge, with a web application firewall or similar solution, than it is ***inside*** the cloud in the application. This is true regardless of whether the cloud model is public or private; basically if you're being charged on a per-usage basis then placement of web application security solutions has an impact on the total cost of running your application. Now before you jump to the comments and start berating me, hear me out. There's a very good reason for this stance and if you're considering deployment of applications in the cloud as a cost-savings measure then you really should read on before you take me to task. After you're finished you're welcome to get crazy.

Ready? Here comes the logic…

WHERE ATTACKS ARE TURNED BACK IMPACTS COST OF DOING BUSINESS

When you write an application and include in it (as you should) security checks against attacks like SQLi and XSS your application – which are particularly prevalent against Web 2.0 applications where user-generated content is core to the applications - must execute in order to perform those checks for ***every request***, good or bad. If the request is "bad", i.e. contains an attack, then it should be rejected. But by the time you've figured that out in your application you've already incurred the cost of (a) executing the application and depending on the cloud model (b) the cost of the bandwidth required to transfer the request and your subsequent rejection of that request and quite probably (c) the cost of storage in terms of log file growth. If a web application firewall capable of detecting the same attacks is placed at the ***edge*** of the network, "bad" requests are turned back before they have to traverse the network to your application and ***before*** your application must execute in order to detect and reject it.

The counter-argument to this is that the bad requests are mixed in with good requests, and as you're paying on an hourly basis and not per-request it really doesn't impact the cost of running the application. Okay, let's go with that. But now the number of bad requests mixed in with good requests pushes you over the capacity of a single instance and, as it should happen, a second instance is launched to handle the load. Now, because of "bad" requests, you're paying twice as much as you should have to serve legitimate users. The only ***good*** thing about this situation is that now you have real hard dollars to attach to the cost of risk from web attacks.

Depending on the cloud model there's bandwidth costs to consider, too. If you're paying for bandwidth to and from your application then every bad request your application processes pushes your bandwidth usage higher and higher and eventually increases your monthly costs of doing business. This completely ignores the fact that bad requests using resources eventually degrades performance due to increased activity on the network (congestion) and the increasing cost of managing higher volumes of resources and connections on the server(s).

And then there's the cost of storage, which increases as the log file size increases from keeping track of all those bad requests. Yes, disk is cheap, but the cost of maintaining redundancy in storage as well as requirements associated with length of storage of ***all*** interactions by this regulation and that will add up over time.

IT'S NOT ABOUT WHO CAN DO IT BETTER IT'S ABOUT A MORE EFFICIENT ARCHITECTURE

Although this entire discussion is framed in the context of cloud computing, it's really not specific to cloud. The same principles apply to any architecture because the underlying premise is about architecting a more efficient infrastructure. It's about minimizing the cost of securing and delivering applications by reducing the load on the network, application, and storage infrastructure by stopping "bad" requests from ever getting inside the perimeter. It's about applying the principle of efficiency to the big picture rather than individual components; about recognizing the impact of a single malicious request on the entire architecture rather than on any single piece of it. Reducing a little resource consumption here and some bandwidth there and recovering some storage there adds up to a lot more operational efficiency in the end, and even in a traditional data center architecture that's going to translate into real dollars saved. When the scenario is cloud-based, however, the costs associated become more clear because each affected part of the infrastructure is a line-item on your monthly bill.

So whether you're looking at cloud-based applications or just improving the operational efficiency of your existing data center, take a long hard look at the impact of allowing bad requests to be processed by all the myriad components that make up your entire architecture and consider whether eliminating that burden from your data center would result in a more efficient, less costly infrastructure.

- Securing the Other Side of the Cloud
- Cloud Changes Cost of Attacks
- Layer 4 vs Layer 7 DoS Attack
- An Unhackable Server is Still Vulnerable
- The IT Security Flowchart
- Get your SaaS off my cloud
- New TCP vulnerability about trust, not technology
- 4 Reasons We Must Redefine Web Application Security

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