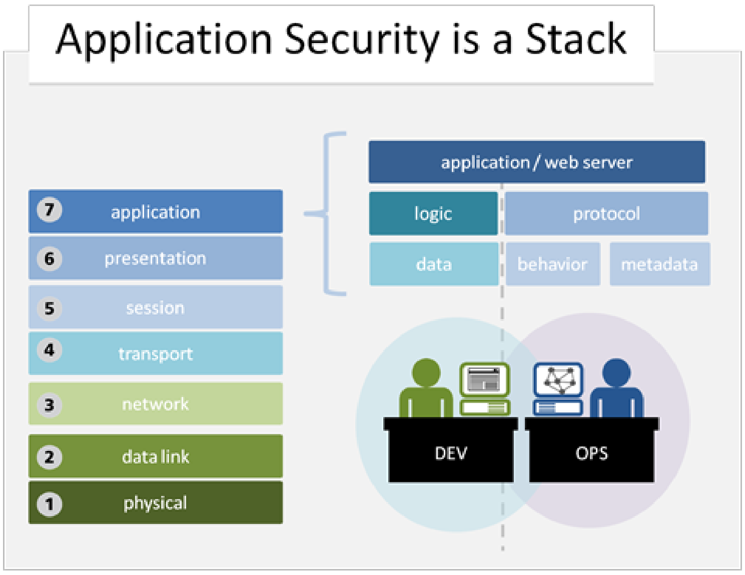
Application Security is a Stack

Lori MacVittie, 2012-11-07

#infosec #web #devops There's the stuff you develop, and the stuff you don't. Both have to be secured.



On December 22, 1944 the German General von Lüttwitz sent an ultimatum to Gen. McAuliffe, whose forces (the Screaming Eagles, in case you were curious) were encircled in the city Bastogne. McAuliffe's now famous reply was, "Nuts!" which so confounded the German general that it gave the 101st time to hold off the Germans reinforcements arrived four days later.

This little historical tidbit illustrates perfectly the issue with language, and how it can confuse two different groups of people who interpret even a simple word like "nuts" in different ways. In the case of information security, such a difference can have as profound an impact as that of McAuliffe's famous reply.

Application Security

It may have been noted by some that I am somewhat persnickety with respect to terminology. While of late this "word rage" has been focused (necessarily) on cloud and related topics, it is by no means constrained to that technology. In fact, when we look at application security we can see that the way in which groups in IT interpret the term "application" has an impact on how they view application security.

When developers hear the term "application security" they of course focus in on those pieces of an application over which they have control: the logic, the data, the "stuff" they developed. When operations hears the term "application security" they necessarily do (or should) view the term in a much broader sense. To operations the term "application" encompasses not only what developers tossed over the wall, but its environment and the protocols being used.

Operations must view application security in this light, because an increasing number of "application" layer attacks focus not on vulnerabilities that exist in code, but on manipulation of the protocol itself as well as the metadata that is inherently a part of the underlying protocol.

The result is that the "application layer" is really more of a stack than a singular entity, much in the same way the transport layer implies not just TCP, but UDP as well, and all that goes along with both. Layer 7 is comprised of both the code tossed over the wall by developers as well as the operational components and makes "application security" a much broader – and more difficult – term to interpret. Differences in interpretation are part of what causes a reluctance for dev and ops to cooperate. When operations starts talking about "application security" issues, developers hear what amounts to an attack on their coding skills and promptly ignores whatever ops has to say.

By acknowledging that application security is a stack and not a single entity enables both dev and ops to cooperate on application layer security without egregiously (and unintentionally) offending one another.

Cooperation and Collaboration

But more than that, recognizing that "application" is really a stack ensures that a growing vector of attack does not go ignored. Protocol and metadata manipulation attacks are a dangerous source of DDoS and other disruptive attacks that can interrupt business and have a real impact on the bottom line.

Developers do not necessarily have the means to address protocol or behavioral (and in many cases, metadata) based vulnerabilities. This is because application code is generally written within a framework that encapsulates (and abstracts away) the protocol stack and because an individual application instance does not have the visibility into client-side behavior necessary to recognize many protocol-based attacks. Metadata-based attacks (such as those that manipulate HTTP headers) are also difficult if not impossible for developers to recognize, and even if they could it is not efficient from both a cost and time perspective for them to address such attacks.

But some protocol behavior-based attacks may be addressed by either group. Limiting file-upload sizes, for example, can help to mitigate attacks such as slow HTTP POSTs, merely by limiting the amount of data that can be accepted through configuration-based constraints and reducing the overall impact of exploitation. But operations can't do that unless it understands what the technical limitations – and needs – of the application are, which is something that generally only the developers are going to know. Similarly, if developers are aware of attack and mitigating solution constraints during the design and development phase of the application lifecycle, they may be able to work around it, or innovate other solutions that will ultimate make the application more secure.

The language we use to communicate can help or hinder collaboration – and not just between IT and the business. Terminology differences within IT – such as those that involve development and operations – can have an impact on how successfully security initiatives can be implemented.

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