The Ascendancy of the Application Layer Threat

Lori MacVittie, 2012-17-01

#adcfw #RSAC *Attackers have outflanked your security infrastructure*

Many are familiar with the name of the legendary Alexander the Great, if not the specific battles in which he fought. And even those familiar with his many victorious conquests are not so familiar with his contributions to his father's battles in which he certainly honed the tactical and strategic expertise that led to his conquest of the "known" world.

In 339 BC, for example, then Macedonian King Phillip II – the father of Alexander the Great – became engaged in a battle at Chaeronea against the combined forces of ancient Greece. While the details are interesting, they are not really all that germane to technology except for commentary on what may be* Phillips' tactics during the battle, as suggested by the Macedonian author Polyaenus:

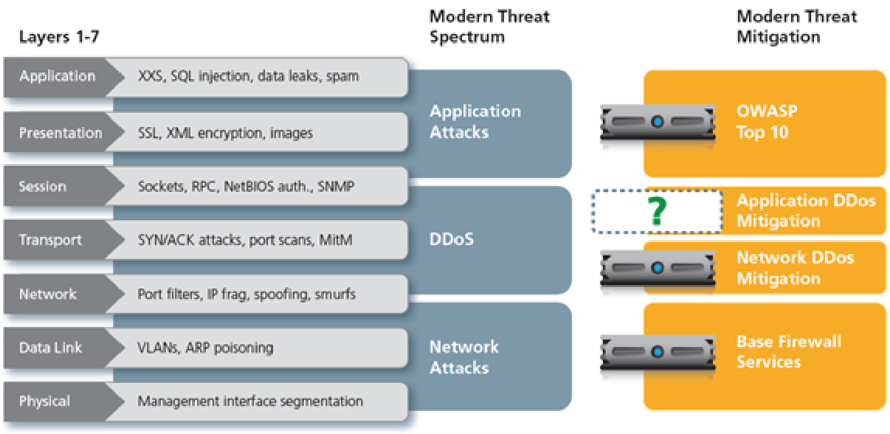
> In another 'stratagem', Polyaenus suggests that Philip deliberately prolonged the battle, to take advantage of the rawness of the Athenian troops (his own veterans being more used to fatigue), and delayed his main attack until the Athenians were exhausted.
>
> -- Battle of Chaeronea (338 BC) (Wikipedia)

This tactic should sound familiar, as it akin in strategy to that of application DDoS attacks today.

THE RISE of APPLICATION LAYER ATTACKS

Attacks at the application layer are here to stay – and we should expect more of them. When the first of these attacks was successful, it became a sure bet that we would see more of them along with more variations on the same theme. And we are. More and more organizations are reporting attacks bombarding them not just at the network layer but above it, at the transport and application layers.

Surely best practices for secure coding would resolve this, you may think. But the attacks that are growing to rule the roost are not the SQLi and XSS attacks that are still very prevalent today. The attacks that are growing and feeding upon the resources of data centers and clouds the globe over are more subtle than that; they're not about injecting malicious code into data to be spread around like a nasty contagion, they're DDoS attacks. Just like their network-focused DDoS counterparts, the goal is not infection – it's disruption.



These attacks exploit protocol behavior as well as potentially missed vulnerabilities in application layer protocols as a means to consume as many server resources as possible using the least amount of client resources. The goal is to look legitimate so the security infrastructure doesn't notice you, and then slowly leech compute resources from servers until they can't stand – and they topple.

They're Phillip's Macedonians; wearing out the web server until it's too tired to stand.

These attacks aren't something listed in the OWASP Top Ten (or even on the OWASP list, for that matter). These are not attacks that can be detected by IPS, IDS, or even traditional stateful firewalls. These technologies focus on data and anomalies in data, not behavior and generally not at the application protocol layer.

For example, consider HTTP Fragmentation attacks.

In this attack, a non-spoofed attacker establishes a valid HTTP connection with a web server. The attacker then proceeds to fragment legitimate HTTP packets into tiny fragments, sending each fragment as slow as the server time out allows, holding up the HTTP connection for a long time without raising any alarms. For Apache and many other web servers designed with improper time-out mechanisms, this HTTP session time can be extended to a very long time period. By opening multiple extended session per attacker, the attacker can silently stop a web service with just a handful of resources.

Multiple Methods in a Single Request is another fine example of exhausting a web server's resources. The attacker creates multiple HTTP requests, not by issuing them one after another during a single session, but by forming a single packet embedded with multiple requests. This allows the attacker to maintain high loads on the victim server with a low attack packet rate. This low rate makes the attacker nearly invisible to NetFlow anomaly detection techniques. Also, if the attacker selects the HTTP method carefully these attacks will bypass deep packet inspection techniques.

There a number of other similar attacks, all variations on the same theme: manipulation of valid behavior to exhaustion of web server resources with the goal of disrupting services. Eventually, servers crash or become so slow they are unable to adequately service legitimate clients – the definition of a successful DDoS attack.

These attacks are not detectable by firewalls and other security infrastructure that only examine packets or even flows for anomalies because no anomaly exists. This is about behavior, about that one person in the bank line who is acting oddly – not enough to alarm most people but just enough to trigger attention from someone trained to detect it. The same is true of security infrastructure. The only component that will detect such subtle improper behavior is one that's been designed to protect it.

* *It was quite a while ago, after all, and sources are somewhat muddied. Whether this account is accurate or not is still debated.*

- The Pythagorean Theorem of Operational Risk
- F5 Friday: When Firewalls Fail…
- F5 Friday: Mitigating the THC SSL DoS Threat
- F5 Friday: If Only the Odds of a **Security** Breach were the Same as Being Hit by Lightning
- F5 Friday: Multi-Layer **Security** for Multi-Layer Attacks
- When the Data Center is Under Siege Don't Forget to Watch Under the Floor
- Challenging the Firewall Data Center Dogma
- What We Learned from Anonymous: DDoS is now 3DoS
- The Many Faces of DDoS: Variations on a Theme or Two

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