The Web 2.0 API: From collaborating to compromised

Lori MacVittie, 2009-01-04

Are you protecting your Web 2.0 APIs?

As Web 2.0 applications continue to expand from connected to collaborative via the extensive use of APIs it behooves developers and security professionals alike to consider the ramifications of providing this necessary yet dangerous avenue of entry into their application infrastructure.

Too many discussions around web application security are focused on the user-facing web interfaces and ignore the potentially more dangerous collaboration-focused interfaces that make up the API. What makes them more dangerous is that they almost always offer an XML exchange format, but it is rare that these services are thoroughly tested against XML-focused attacks. The problem with this is that the XML API provides a nice back-door through which attacks can be executed that are designed to affect the users. This is because the XML API is simply another means by which the application can be accessed by third-party developers and applications.

And, of course, miscreants.

If your XML APIs are less-well tested than traditional web-interfaces it may be because of the differences inherent in the tools and skill-sets necessary to properly test them against attacks. An SQL injection attack may look the same regardless of whether it's transported via an HTTP POST response or XML, but the way in which the encompassing data, the envelope if you will, must be interpreted and parsed in order to discover that attack requires very different methods and skill sets. There are also a variety of XML-focused attacks that are not prevalent in traditional web-based application data exchanges such as XPath injection, schema poisoning, and exploitation of required parsing mechanisms.

Fortunately, there are a plethora of tools and options available that address both the difference in tools and knowledge necessary to test and protect XML APIs. Many of them are freely available and contain the combined knowledge of many experts, such as those maintained by OWASP.

OWASP and XML

OWASP, the Open Web Application Security Project, is often overshadowed by other efforts to improve web application security. The depth and breadth of information available through OWASP is often a surprise to even the most seasoned security folks. OWASP is a non-profit organization that relies on contributions from an active, knowledgeable community to create and maintain information and tools on across a wide variety of web application security concerns.

An excellent example of the usefulness of OWASP is in the area of XML-based exploits and testing. An example of the thoroughness of OWASP's security expertise can be seen in its documentation around one of the most simplest of XML security concerns: enforcing well-formedness. What is often lacking in discussions of security is ***actionable help*** for testing. After all, most developers aren't malicious miscreants, so coming up with a test-case to determine the security – or lack thereof – of an application against a particular attack can be difficult. OWASP solves this problem by providing examples that can be used to not only enable a developer to understand the core of the attack, but that can be used in testing. In the case of enforcing well-formedness, OWASP provides the following simple but effective example:

```

OWASP

EOIN
```

```
I am Malformed
```

```
Don’t forget me this weekend!
```

```
```

OWASP also offers tools, free of charge, to assist in the security testing of XML and web-services. WSDigger, for example, comes with sample plug-ins that can generate several XML attacks for use in testing Web 2.0 APIs or any other XML-based application. WSDigger can help automate tests for:

- SQL injection
- cross site scripting
- XPATH injection attacks

What’s great about OWASP is that goes further than just telling you ***how*** to test web applications and APIs, it also provides guidance and assistance in remediation. That’s important because once you find out your application is vulnerable you need to address the vulnerability. Secure coding is the goal, after all, and having concrete examples of what to look for ***and how to fix it*** aids in the skill set necessary for developers to improve their secure coding techniques. For developers who really want to dig in and learn secure coding best practices, which are unfortunately not taught in most universities today, OWASP is the organization behind the WebGoat Project, “a deliberately insecure J2EE web application maintained by OWASP designed to teach web application security lessons”. And best of all given today’s tight budgets, it’s completely free.

Some might view OWASPs focus on secure coding as a condemnation of web application firewalls. Not true at all. While the primary goal of OWASP is to encourage secure coding practices through education and information, it also recognizes that a web application firewall may be the primary or secondary line of defense for an organization. To help organizations choose a WAF, OWASP offers guidance on the selection criteria it believes is most important, and encourages the use of the Web Application Firewall Evaluation Criteria.

IF YOU AREN’T TESTING THAT API, YOU SHOULD

Given the amount of information, tools, and assistance provided for free from OWASP and for nominal fees at other organizations, there’s no valid reason to ***not*** be thoroughly testing Web 2.0 APIs. The use of XML as a primary method of data transport across a growing number of sites simply increases the attack surface across which miscreants can attempt to spread malicious data.

Given the connectedness inherent in social networking and Web 2.0, the sharing of malicious data from one application can potentially affect not just thousands of users, but thousands of users of connected applications, essentially turning a large portion of the Web 2.0 network into a giant, automated bot-net like distribution network for miscreants to use and abuse at will.

So be certain to evaluate the security posture of your web application APIs as thoroughly as you would its user-interface.

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

©2017 F5 Networks, Inc. All rights reserved. F5, F5 Networks, and the F5 logo are trademarks of F5 Networks, Inc. in the U.S. and in certain other countries. Other F5 trademarks are identified at f5.com. Any other products, services, or company names referenced herein may be trademarks of their respective owners with no endorsement or affiliation, express or implied, claimed by F5. CS04-00015 0113