



May 2010

Hello readers and welcome to another edition of Security Compass's quarterly newsletter, Navigations.

In this issue, we will be analyzing the How's and Why's of Google's much-publicized breach in [Dragons in the Box](#), this issue's feature article. As well, we talk about the difficulty of presenting security requirements to a skeptical board of executives in this edition's [Security Scenarios](#). We also bring for your reading pleasure, some of the headlines that caught our eye in the newly-redesigned [Updates section](#).

As always, please give us any thoughts that you may have on Navigations or how we may improve it by dropping a line or two in an email with your favorite client pointing to [navigations@securitycompass.com](mailto:navigations@securitycompass.com).

## Trainings

Secure Coding in Java/JEE: Developing Defensible Applications, SANSFIRE, Baltimore, MD

Secure Code Review for Java Web Apps, SANSFIRE, Baltimore, MD

Web Application Security Awareness, SecTor, Toronto, ON

## Conferences

BlackHat USA 2010

Defcon 18 2010

SecTor 2010

OWASP AppSec USA 2010

Recon 2010

## Blogroll

Security Compass YouTube Channel

SDLC Security Audit Framework

Secure Web Application Framework Manifesto - Draft

XSLT Command Execution Exploit

## Links

OWASP

Web Application Security Consortium

Threatpost

## Feature Article

Dragons in the Box, The Necessities for Operation Aurora:



We will be taking a look at the recent attacks on Google and other companies (so-called Operation Aurora) and what made them possible. As well, we examine how the intrusions occurred and what companies can do to protect themselves from future attacks.

"What were once rare occurrences of mischievous cyber-vandalism may now be becoming commonplace industry-wide acts of corporate espionage. In January, Google was hacked as part of a large-scale operation that compromised an estimated number of over 30 companies. The attacks generally targeted corporate IP mostly comprised of application code from SCM systems..."

[Read more, Dragons in the box...](#)

## Security Scenarios

Security scenarios are modeled after the Harvard Business Review Case Studies - they're real world scenarios based on actual challenges faced by practitioners on the ground. Each scenario describes a fictional predicament faced by somebody involved in application security.

This Issue's Scenario, The Skeptic:



This issue's scenario is about James, manager of information security at a large healthcare company. After several years of primarily running penetration testing, and a few limited source code reviews, James successfully made the case to internal IT leadership that security needs to come earlier in the software development life cycle. Although most leadership was on board, there remained a few skeptics.

[Read the rest of James' dilemma...](#)

How would you make the case to the skeptics that security should be brought earlier into the SDLC? Send in your responses via email to [navigations@securitycompass.com](mailto:navigations@securitycompass.com). The best reader response will be showcased in the next issue.

\* \* \*

Last Issue's Scenario, The Falling Stock of Appsec:

Our last scenario dealt with the issue of justifying application security in the face of falling market value. How would you justify application security? [Read more, Case Study: The Falling Stock of Appsec...](#)

Best Reader Response, Jason Lam: "In Jamie's shoes, I would first estimate the cost of one single incident based on previous incidents in similar organizations and the likelihood of an incident happening...". Read Jason's entire response on the scenario page [here](#).

Security Compass's Response, Nish Bhalla: "Once, however, organizations realize the impact of the potential issues (like DG & S), they end up hiring someone like Jamie to help reduce their risk...". Read Nish's entire response on the scenario page [here](#).

## Updates



New OWASP Top Ten, top security threats for 2010:

OWASP has released a new listing of the top security threats for 2010. Among the changes are some new threats and a reordering of already existing threats. Injection attacks seem to have become much more prominent as they have been moved to the top of the new list, displacing previous leader, XSS attacks, which end up in second place.

In an interesting move, Malicious File Execution was taken off the list, even though it was the third highest ranking vulnerability on OWASP's last listing; this was a big issue in 2007 as many PHP applications were vulnerable but now PHP ships with more secure settings by default resulting in the issue dropping in significance. 2007's A6, Information Leakage and Improper Error Handling, has been incorporated into 2010's A6, Security Misconfiguration. This issue saw its debut on 2004's listing, but was dropped in 2007 due to it not being considered a software issue; this issue is extremely prevalent however and so it was re-included in 2010's listing. Get your copy of the listing and read more about this and older releases of OWASP's top ten security threats at the OWASP Top Ten Project page.

Source: [OWASP Top Ten Project](#)



Java EE design pattern security explored:

The security of common Java design patterns is something that developers are now paying more attention to. Aiding in the design of secure code, a new OWASP initiative wishes to document the methods by which Java design patterns could be secured. While the project is still in its infancy, developers wishing to check their code for vulnerabilities might benefit from giving the project a look.

Source: [Security Analysis of Core J2EE Design Patterns](#)



Skipfish, Security testing for the masses:

Skipfish promises to be a fast, easy and cutting edge web application security 'reconnaissance' tool. Written in C and optimized for HTTP handling, Skipfish boasts an impressive 2000+ requests per second on responsive hosts. If you're in the business of creating or maintaining web applications, then it might be worth your while to give Skipfish a once-over. Note that Skipfish is still considered experimental, handle with care.

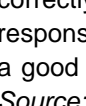
Source: [Skipfish](#)



Pinata, CSRF proof of concept on demand:

Judging a site's vulnerability to CSRF attacks can be difficult without proper understanding of what a CSRF attack is or how it may apply to your site. Pinata promises to alleviate this issue by giving web developers on-demand proof of concept CSRF attacks from a given HTTP request. Note that you will need a correctly configured HTTP proxy (like [Paros](#)) in order to capture the appropriate response; more on that [here](#). Note that Pinata is still experimental and lacking of a good GUI interface (or even a command line interface, for that matter.)

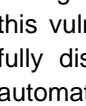
Source: [Pinata](#)



Java zero-day patched out of cycle:

[Threatpost](#) has been closely following the story of an unpatched Java exploit originally disclosed on April 9. In an unexpected move by Oracle/Sun, an out of cycle patch was released for April 15th for Java 6 disabling the exploit that's been making its rounds for the past week. The root of the vulnerability is that the Java web start deployment kit (javaws.exe, runs in user mode) does not sanitize command line parameters. This has led to websites popping up exploiting users through [drive-by-downloads](#). The vendor initially declined to release a patch for this vulnerability citing it as insignificant. This forced researcher Tavis Ormandy to fully disclose the vulnerability on April 9. The update is available through the automatic Java updater or [here](#).

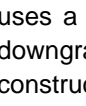
Source: [Inside the java 0-Day Exploit | Threatpost](#)



Chip and PIN is broken:

Users of chip and pin technology in credit cards were in for a shock back in February, when a research team disclosed the details of a critical flaw in the chip and PIN algorithm, allowing an attacker to use a stolen credit card without knowing the PIN. According to the [whitepaper](#), several cases of claimed stolen credit card fraud have been declined by the respective banks on the grounds that "Verified by PIN" positively identified a customer through their PIN. This exploit uses a miscommunication between the credit card chip and the reader to force a downgrade attack. The researchers claim that while this exploit requires the construction of a small PCB and the use of a laptop, the creation and selling of such "kits" in the underground community is not farfetched and may already be happening.

Source: [Chip and PIN is broken](#)



Corporate Hacking, the new face of Facebook:

According to a story on [USATODAY.com](#), cybercriminals are increasingly using Facebook to data-mine corporations. In a brazen attempt to siphon data from an unnamed corporation, cybercriminals purportedly rummaged around a corporate server for two weeks before being discovered. This after managing to break into a worker's Facebook account and from there infecting one of his/her co-worker's computer with a keylogger that yielded the username and password to the corporate server.

Source: [How cybercriminals invade social networks, companies - USATODAY.com](#)