The Divine and Felonious Nature of Cyber Security

( Introduction to DevSecOps )
BEYOND THE PHOENIX PROJECT

New!
In this transcript of the audio series, Gene Kim and John Willis present a nine-part discussion that includes an oral history of the DevOps movement, as well as discussions around pivotal figures and philosophies that DevOps draws upon, from Goldratt to Deming; from Lean to safety culture to learning organizations. The book is a great way for listeners to take an even

BEYOND THE PHOENIX PROJECT (AUDIOBOOK)

New!
Gene Kim and John Willis present this nine-part series that includes an oral history of the DevOps movement, as well as discussion around pivotal figures and philosophies that DevOps draws upon, from Goldratt to Deming; from Lean to Safety Culture to Learning

DEVOPS HANDBOOK (AUDIOBOOK)

New!
Increase profitability, elevate work culture, and exceed productivity goals through DevOps practices. More than ever, the effective management of technology is critical for business competitiveness. This non-fiction follow-up to The Phoenix Project shows leaders,

MYTHBUSTING DEVOPS IN THE ENTERPRISE

Anyone leading a company through a DevOps transformation will encounter minor to significant internal skepticism or lack concrete experience. This document lists the most common leadership and cultural traps and provides high-level reassurance and evidence that DevOps practices are generally applicable and plausibly successful in enterprise environments.
The Felonious Nature of Cyber Security
Infecting
1.5M Java Modules Per Week
5M Node Modules Per Week
31B Modules Per Year
230k Modules Per Enterprise/Per Year
8% Have Known Vulnerabilities
8M Average Cost to Remediate a Breach

modulecounts.com and 2017 Software Supply Chain Report
More Meta

- 28k new vulnerabilities in 2017
- Fintech 60%
- Most customers are using 2X more OSS than they thought.
- 85% unknown out of license compliance

The use of open source software is an essential part of application development

96% of applications scanned in this analysis utilized open source

The average app included 147 unique open source components
### SOURCES/11-dhcclient

```
@@ -7,7 +7,7 @@

interface=$1

eval "$(
-    declare | LC_ALL=C grep '^DHCP4_[A-Z_]*=' | while read opt; do
+    declare | LC_ALL=C grep '^DHCP4_[A-Z_]*=' | while read -r opt; do
      optname=${opt%*}"
    optname=${optname,,}
    optname=new_${optname#dhcp4_}
```
Actual Exploitation 2015 VZ DBIR

NOT ALL CVES ARE CREATED EQUAL.

If we look at the frequency of exploitation in Figure 11, we see a much different picture than what’s shown by the raw vulnerability count of Figure 12. **Ten CVEs account for almost 97%** of the exploits observed in 2014. While that’s a pretty amazing statistic, don’t be lulled into thinking you’ve found an easy way out of the vulnerability remediation rodeo. Prioritization will definitely help from a risk-cutting perspective, but beyond the top 10 are 7 million other exploited vulnerabilities that may need to be ridden down. And therein, of course, lies the challenge; once the “mega-vulns” are roped in (assuming you could identify them ahead of time), how do you approach addressing the rest of the horde in an orderly, comprehensive, and continuous manner over time?

About half of the CVEs exploited in 2014 went from publish to pwn in less than a month.

**Figure 11.** Cumulative percentage of exploited vulnerabilities by top 10 CVEs
COMPLAINT

Plaintiff, Securities and Exchange Commission (the “Commission”), files its complaint and alleges that:

SUMMARY

1. Defendant Jun Ying (“Ying”) committed securities fraud by engaging in illegal insider trading. After being entrusted with material, nonpublic information about a massive cyber-intrusion and data breach suffered by his employer, Equifax Inc. (“Equifax” or “the company”), Ying exercised all his vested Equifax stock options and sold the shares prior to the public announcement of the breach. By selling when he did, Ying avoided losses in excess of $117,000.
Anatomy of CVE-2017-5638

- Discovered 3/6/2017
- Announced 3/9/2017
- CVE created 3/10/2017
- Equifax Oracle Patches 4/2017
- Equifax Patches 6/30/2017
- Equifax discovers 7/29/2017
- Equifax announced 9/2017

```
curl http://127.0.0.1:8000/struts2-showcase/showcase.action -H "Content-Type: 
%{(#_="multipart/form-

com.opensymphony.xwork2.inject.ContainerImpl@dd2d200
```
Anatomy of CVE-2017-5638

- Discovered - 3/9/2017
- Action - 3/10/2017
- Remediation - 3/14/2017

@botchagalupe
Anatomy of CVE-2017-8046
(Fool Me Once)

• Published 9/21/17
• CVE created 01/04/2018
• Discovered 2/18/17
• Corrected 3/6/18

@botchagalupe

Current Description
Malicious PATCH requests submitted to spring-data-rest servers in Pivotal Spring Data REST versions prior to 2.5.12, 2.6.7, 3.0 RC3, Spring Boot versions prior to 2.0.0M4, and Spring Data release trains prior to Kay-RC3 can use specially crafted JSON data to run arbitrary Java code.
WEB APPLICATION SECURITY

Using Signal Sciences to Defend Apache Struts CVE-2018-11776

Cody Wood

Minor turned application security fanatic. Often straddling multiple departments and roles. Bit by the appsec bug in Houston, TX at the Threat Research Center working for Whitehat Security. Worked for brief periods in both builder and breaker roles. Currently focused on researching and...
Security and the Goldilocks Zone

• The fallacious nature of cyber security relates to the standard legacy security model specifically on the idea of perimeter security.

• This concept involves the implementation of a state-full firewall at a routed point within the network that very rarely gets looked at unless an operational change is required.

• The problem with having only premier security is that applications have changed significantly in the last ten years and the infrastructure they run upon is playing by the same old rules.
Devops Meets DevSecOps
“You build it, you run it”
- Werner Voegls (Amazon)
KEEP CALM AND PULL THE ANDON CHORD
Devops Automated Deployment Pipeline

Source: Wikipedia - Continuous Delivery
Devops Automated Deployment Pipeline

Google developer scale

30,000+ developers
1 billion files
800,000 builds per day

45,000 commits per workday
9 million source files

20,000 code reviews per workday
2 billion lines of code

2+ PB of build outputs per day

150 million test cases run per day

JFrog swampUP Melody Meckfessel, Senior Director of Engineering, Google Cloud Platform

@botchagalupe
Summary

- Agile took us from months to days to deliver software
- Devops took from months to days to deploy software
- Now security is the bottleneck
DevSecOps
You Build It, You Secure It
DevSecOps as Supply Chain?

Source: Wikipedia - Continuous Delivery

@botchagalupe
Software Supply Chain

DevOps Example

Delivery Team
Version Control
Build
Test
Release

Prod
Stage
Software Supply Chain

DevOps Example

Delivery Team

Version Control

Build

Test

Release

Stage

Prod

@botchagalupe
Security in the Software Supply Chain

DevOps Example

Delivery Team
Version Control
Build
Test
Release

DevOps and Security

Delivery Team
Version Control
Build
Test
Release

Selenium
CHEF
Prod
Stage

@botchagalupe
Security in the Software Supply Chain

DevOps Example
- Delivery Team
- Version Control
- Build
- Test
- Release

DevSecOps Example
- Delivery Team
- Version Control
- Build
- Test
- Release

Tools:
- Eclipse
- GitHub
- Jenkins
- CHEF
- OWASP
- University of Maryland
- JFrog Artifactory
- Twistlock
- OWASP ZAP
- Fortify
- Qualys
- Dome9

@botchagalupe
The New Goldilocks Zone (DevSecOps)

DevSecOps Supply Chain

- Delivery Team
- Version Control
- Build
- Test
- Release

Security Training
- Security Requirements
- Threat Modeling
- Architecture Review
- OWASP Top 10
- IDE Plugins
- Code Examples

Fail the Build
- SAST/DAST/IAST
- Configuration Analysis
- Application Module Scanning
- Threat Modeling as Unit Test

Automated Pen Testing
- Static Code Analysis
- Security Policy Testing
- Configuration Analysis
- Security Monitoring
- Configuration Monitoring

Stage
Prod

Evident.io

JFrog Artifactory
OWASP
Veracode
ZAP
Fortify
Qualys
Dome9

arachni
web application security
Best Practices for DevSecOps

• Train development teams to develop secure code
• Track security issues the same as software issues
• Security as code, Security Built In.
• Integrate security controls in the software pipeline
• Automate security test in the build process
• Detect known vulnerabilities during the pipeline
• Monitor security in production for known states
• Inject failure to ensure security is hardened

Knowing Adversities and Motivations

- Scanners
- Researchers
- Paid Noise
- Advanced Adversaries
- Information Brokerage
- Fame / Payment
- Continuous Payment
- Control / Payment

Shannon Lietz
Director, DevSecOps
Intuit
@devsecops
Knowing Adversities and Motivations

Adversary Return Rate

Shannon Lietz
Director, DevSecOps
Intuit
@devsecops
Delivering the Promise

- Average days to close a vulnerability improved by 74%
- Automated code quality scanning shows overall security code scores has increased by 10%
- More than 60% of application teams are performing security tests before release
- Critically vulnerable open source components (CVE 7.5+) downloaded has decreased from 18% to 6.25%
- ~ 55% of technical debt and security defects identified as a result of periodic testing have been dispositioned
- ~ 77% of older technical debt and security defects have been remediated, have a remediation plan in place, or have been addressed through managed retirements of assets
Example: Intrusion Detection Triggering Automation
DevSecOps Operational Tips

• Ruthlessly eliminate false positives to Developers
• Explain the vulnerabilities in business impact terms
• Devops the vulnerability (JIRA, backlog, Kanban)
• Open the code base to everyone in the organization
• Educate on how to fix
Organizational Fact Finding
Organizational Fact Finding - Meta Patterns

- Data Center vs Service Center
- Tribal vs Institutional Knowledge
- Security and Compliance Theater
- Work Visualization
- Organizational Workflow (WIP)
- Collaboration, Culture, and Performance
For this exercise we'll build this 15-piece puzzle several times, and experiment with ways to do it faster.
Graphical Storytelling

1. Map end-to-end process

2. Identify wastes, inefficiencies, bottlenecks

3. Identify countermeasures
The Divine
We don’t know precisely the circumstances, or even year, attending the last moments of the last dodo, so we don’t know which arrived first, a world that contained a Principia or one that had no dodos, but we do know that they happened at more or less the same time. You would be hard pressed, I would submit, to find a better pairing of occurrences to illustrate the divine and felonious nature of the human being—a species of organism that is capable of unpicking the deepest secrets of the heavens while at the same time pounding into extinction, for no purpose at all, a creature that never did us any harm and wasn’t even remotely capable of understanding what we were doing to it as we did it.
The Felonious Nature of Cyber Security

CONFESSIONS OF A HACKER

Confessions of a Hacker is a rare opportunity to hear from a professional hacker who will share their techniques and examples of how they have defeated the latest cyber security technologies as well as physical security measures. There is nothing that has stopped our hacker. They are your worst nightmare.

@botchagalupe