How to effect change in the Epistemological Wasteland of Application Security

James Wickett
Let us know what you think

Click ‘engage’ to rate session.
Rate 12 sessions to get the supercool GOTO reward

Join the conversation gotoidn
HOW TO EFFECT CHANGE IN THE EPISTEMOLOGICAL WASTELAND OF APPLICATION SECURITY

- @WICKETT
JAMES WICKETT

Sr. Engineer, Signal Sciences

AUSTIN, TX

Hands-on Gauntlt Book

DevOps Days Global Organizer

LASCON Organizer

@WICKETT  #RUGGEDDEVOPS
Application Security Monitoring and Instrumentation

Application Security you can use!

An approach that integrates with devops organizations

Productizing the Etsy security approach
Make security visible.

Prioritize your defensive efforts where your applications are actually targeted.

LEARN MORE  REQUEST MORE INFO

signalsciences.com
Summary

Software development has been a constant experiment in how we know anything.

Application Security abdicated runtime responsibility and effectively abdicated development responsibility through incoherent philosophical approaches and fostering organizational silos.

DevOps is here to stay, and security can actually be a part of it.

Ops found a way to add value, security needs to find that same path.

There are three ways we can add value: at development, at deploy, at runtime.
A study in how we know anything in Application Security

@WICKETT

#RUGGEDDEVOPS
Spoiler Alert: We don't!
ONCE UPON A TIME...
Epistemological Problem of Software Development
WE OPTIMIZE FOR THE PROBABLE
UNIT TESTING
INTEGRATION TESTING
Happy Path Engineering
We also optimize for the possible
OVER ENGINEERING
The scaling algo that never got used...
There is too much to choose from in the realm of possible
Actually, we optimize for the perceived probable
How do we know what to create?
This is the problem
Epistemological Problem of Software Development
WE GATHER DATA AND RHETORIC TO SUPPORT OUR THEORIES
There are 3 major arcs in the history of Software Development
First Arc: Agile

@Wickett

#RuggedDevOps
AGILE AVOIDS THE PROBLEM
AGILE REMINDS THAT
WE DON'T KNOW WHAT
WE ARE BUILDING
Behavior Driven Development
BDD = AGILE + FEEDBACK
Behavior Driven Development is a second-generation, outside–in, pull-based, multiple-stakeholder, multiple-scale, high-automation, agile methodology. It describes a cycle of interactions with well-defined outputs, resulting in the delivery of working, tested software that matters.

Dan North, 2009
Agile emphasizes feedback to developers from their overlords and sometimes even customers.
TLDR: Rapid Iterations Win
Agile is our guiding light.
The world has changed since Agile
We don't sell CD's anymore
Software as a Service
The last fifteen years have brought a complete change in our delivery cadence, distribution mechanisms and revenue models
Second Arc: DevOps

@wickett

#ruggeddevops
DEVOPS IS THE APPLICATION OF AGILE METHODOLOGY TO SYSTEM ADMINISTRATION

- THE PRACTICE OF CLOUD SYSTEM ADMINISTRATION BOOK
DEVOPS
AGILE INFRASTRUCTURE
10 deploys per day
Dev & ops cooperation at Flickr

John Allspaw & Paul Hammond
Velocity 2009

http://www.slideshare.net/jallspaw/10-deploys-per-day-dev-and-ops-cooperation-at-flickr
Less WIP

Less Technical Debt

@Wickett

#RUGGEDDEVOPS
Customers actually using the feature while the developer is working on it
Great side effect: produces happy developers
I want change!

Wall of Confusion

I want stability!

Development  Operations

@WICKETT  #RUGGEDDEVOPS
WORKED FINE IN DEV

OPS PROBLEM NOW

@WICKETT  #RUGGEDDEVOPS
DevOps realized that ops doesn't know what devs know and vice versa.
Dev : Ops
10 : 1

@WICKETT
#RUGGEDDEVOPS
DevOps is an epistemological breakthrough joining people around a common problem.
Culture is the most important aspect to DevOps succeeding in the enterprise

- Patrick DeBois
Culture is shaped in part by values.
Mutual Understanding
Shared Language
Shared Views
Collaborative Tooling
DEVOPS IS THE INEVITABLE RESULT OF NEEDING TO DO EFFICIENT OPERATIONS IN A [DISTRIBUTED COMPUTING AND CLOUD] ENVIRONMENT.

- TOM LIMONCELLI
TLDR;

High-performing IT organizations experience 60X fewer failures and recover from failure 168X faster than their lower-performing peers. They also deploy 30X more frequently with 200X shorter lead times.
CULTURE
AUTOMATION
MEASUREMENT
SHARING

- @damonedwards, @botchagalupe
DevOps gone wrong

WANTED DEVOPS

BUILT A "DEVOPS TEAM"

@WICKETT  #RUGGEDDEVOPS
“THAT THE WORD #DEVOPS GETS REDUCED TO TECHNOLOGY IS A MANIFESTATION OF HOW BADLY WE NEED A CULTURAL SHIFT”
- @PATRICKDEBOIS

http://www.slideshare.net/cm6051/london-devops-31-5-years-of-devops
Third Arc: Continuous Delivery

@WICKETT  #RUGGEDDEVOPS
Continuous Delivery is not merely how often you deliver but how little you can deliver at a time.
DELIVERY PIPESLINES ARE RAD!
Batch Size of 1
Separation of Duties Considered Harmful

@WICKETT #RUGGEDDEVOPS
Give power to the developers to deploy
Reduce Code Latency
Increase Code Velocity
3 ARCS:

AGILE

DEVOPS

CONTINUOUS DELIVERY
The next arc:
Security
RUGGED
“...THOSE STUPID DEVELOPERS”

- Security person
“Security prefers a system powered off and unplugged”
- Developer
CULTURAL UNREST WITH SECURITY IN MOST ORGANIZATIONS
Compliance Driven Culture
“[RISK ASSESSMENT] INTRODUCES A DANGEROUS FALLACY: THAT STRUCTURED INADEQUACY IS ALMOST AS GOOD AS ADEQUACY AND THAT UNDERFUNDED SECURITY EFFORTS PLUS RISK MANAGEMENT ARE ABOUT AS GOOD AS PROPERLY FUNDED SECURITY WORK”
Security is where ops was 5 years ago...
Dev : Ops : Sec
100 : 10 : 1
Understaffing means no one thinks security helps the business win
DevOps changed that for Ops, security can change too.

@WICKETT  #RUGGEDDEVOPS
Netflix demonstrated that people care about resiliency.
Innately, we all care

@WICKETT #RUGGEDDEVOPS
Rugged Software Movement

Rugged Software Development
Joshua Corman, David Rice, Jeff Williams
2010

@wickett #ruggeddevops
Put Your Robots to Work: Security Automation at Twitter

Speakers: Justin Collins, Security Engineer, Twitter
Neil Matatall, Information Security Engineer, Twitter
Alex Smolen, Security Engineer, Twitter

https://vimeo.com/54250716
http://www.youtube.com/watch?v=jQblKuMuS0Y
Security’s way forward is to help developers and help operations
START THERE
Let’s review Security’s approach thus far
Bad Idea #1
Applications can't be defended—Web App Firewalls Suck!
Let's do developer training
Windows Internet Explorer

XSS Attack!

OK
THEY SEE ME ROLLIN'
THEY PATCHIN'

@WICKETT

#RUGGEDDEVOPS
Awareness Campaign
OWASP Top Ten

@WICKETT

#RUGGEDDEVOPS
We abandoned knowing anything useful about the runtime
Instead add defense based on behaviors
BAD IDEA #2

Developers can't figure it out. Let's scan for vulnerabilities instead.

@WICKETT #RUGGEDDEVOPS
“HERE IS A 400 PAGE PDF OF OUR FINDINGS TO PROVE YOUR DEVELOPERS DON'T GET IT!”

- The Pen Tester
Even with the emphasis on appsec training, in practice we made it a dark art.
INTEGRATED RUGGED TESTING SHOULD SIT INSIDE THE PIPELINE
BAD IDEA #3

With the new alignment to vulnerability scanning, there is a tendency to fix the low-hanging fruit.
What if when we swat flies,
We're killing only slow ones...
So there's only fast ones to breed?

@WICKETT

#RUGGEDDEVOPS
WE STILL DON'T KNOW WHO IS ATTACKING US
WE STILL DON'T ACTUALLY KNOW WHAT THEY ARE ATTACKING
Real Threats go Unknown so Developers fix what the automated tooling detected at a certain point in time

@Wickett

#RUGGEDDEVOPS
Add Application Security Telemetry
BAD IDEA #4

PUT IN TOOLING THAT NO ONE OUTSIDE OF SECURITY CAN UNDERSTAND
Usually in the name of compliance
“Get a Web App Firewall dude!”
– PCI-DSS REQ 6.6
Choose your own adventure...
SMALLEST POSSIBLE SOLUTION YOU CAN CONSIDER A WAF...
Our CDN added ModSecurity Ruleset
Huzzah!

@WICKETT  
#RUGGEDDEVOPS
AN APPLIANCE THAT BLOCKS ALL THE THINGS
And now you wonder why no one eats lunch with you anymore.
“Every aspect of managing WAFs is an ongoing process. This is the antithesis of set it and forget it technology. That is the real point of this research. To maximize value from your WAF you need to go in with everyone’s eyes open to the effort required to get and keep the WAF running productively.”

- A whitepaper from a WAF vendor
OK, Security has to change... how do we add value already?
Two ways!
Add value to Devs
Add value to ops
Pray that someone notices
Pro-Tip #1
Automate security tooling to run in testing
Start with adding just one test for XSS on a few pages in your app.
GAUNTLET AUTOMATES SECURITY TOOLS
Gauntlt

Open source, MIT License

Gauntlt comes with pre-canned steps that hook security testing tools

Gauntlt does not install tools

Gauntlt wants to be part of the CI/CD pipeline

Be a good citizen of exit status and stdout/stderr
Feature: nmap attacks for example.com

Given
  Given "nmap" is installed
  And the following profile:
  | name   | value     |
  | hostname | example.com |

Scenario: Verify server is open on expected ports
  When I launch an "nmap" attack with:
  
  nmap -F <hostname>

  Then the output should contain:
  
  80/tcp  open  http

Scenario: Verify that there are no unexpected ports open
  When I launch an "nmap" attack with:
  
  nmap -F <hostname>

  Then the output should not contain:
  
  25/tcp
HERE'S AN XSS ATTACK YOU CAN USE
Feature: Look for cross site scripting (xss) using arachni against a URL

Scenario: Using arachni, look for cross site scripting and verify no issues are found
  Given "arachni" is installed
  And the following profile:
    | name    | value                          |
    | url     | http://localhost:8008          |
  When I launch an "arachni" attack with:
    ""
    arachni --modules=xss --depth=1 --link-count=10 --auto-redundant=2 <url>
    ""
  Then the output should contain "0 issues were detected."
github.com/gauntlt/gauntlt-demo
Hands-on Gauntlt Book for Goto Attendees

Email book@gauntlt.org before the end of the day for a review copy

@wickett #ruggeddevops
Pro-tip #2
Put security testing in your continuous integration system
this is a demo set of attacks that can be used to get started with gauntlt

```
Using worker: worker-linux-10-1.bb.travis-ci.org:travis-linux-1

$ git clone --depth=50 --branch=master git://github.com/gauntlt/gauntlt-demo.git gauntlt/gauntlt-demo
$ cd gauntlt/gauntlt-demo
$ git checkout -qf a1e38a0b1a6b896265af8e21708f34ebfa1087b0
$ git submodule init
$ git submodule update
$ rvm use 1.9.3 --install --binary --fuzzy
Using /home/travis/.rvm/gems/ruby-1.9.3-p484
$ export BUNDLE_GEMFILE=$PWD/Gemfile
$ ruby --version
ruby 1.9.3p484 (2013-11-22 revision 43786) [x86_64-linux]
$ rvm --version
rvm 1.25.14 (version) by Wayne E. Seguin <wayneeseguin@gmail.com>, Michal Papis <mpapis@gmail.com> [https://rvm.io/]
$ gem --version
2.2.2
$ bundle --version
Bundler version 1.5.3
Applying fix for NPM certificates
$ git submodule update --init --recursive
$ bundle install
$ sudo apt-get install nmap
$ sudo apt-get install wget
$ sudo apt-get install libcurl4-openssl-dev
$ pwd
$ export SSLYZE_PATH="/home/travis/build/gauntlt/gauntlt-demo/vendor/sslyze/sslyze.py"
$ export SQLMAP_PATH="/home/travis/build/gauntlt/gauntlt-demo/vendor/sqlmap/sqlmap.py"
$ cd vendor/Garpur && sudo python setup.py install && cd .....
$ cd vendor && wget http://downloads.sourceforge.net/project/dirb/dirb/2.03/dirb203.tar.gz && tar xvzf dirb203.tar.gz && cd dirb &&...
```
Feature: hello world with gauntlt using the generic command line attack

Scenario:

- When I launch a "generic" attack with: `# gauntlt-1.0.8/lib/gauntlt/attack_adapters/generic.rb:1`
  ```
  ***
  cat /etc/passwd
  ***
  ```
  Then the output should contain: `# aruba-0.5.4/lib/aruba/cucumber.rb:147`
  ```
  ***
  root
  ***
  ```

Feature: @slow @final

Scenario: Look for cross site scripting (xss) using arachni against a URL

Given "arachni" is installed

And the following profile:

<table>
<thead>
<tr>
<th>name</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td><a href="http://localhost:8008">http://localhost:8008</a></td>
</tr>
</tbody>
</table>

When I launch an "arachni" attack with:

```
***
arachni --modules=xss --depth=1 --link-count=10 --auto-redundant=2 <url>
***
```

Then the output should contain "0 issues were detected."

Scenario: Using arachni, look for cross site scripting and verify no issues are found

Given "arachni" is installed

And the following profile:

<table>
<thead>
<tr>
<th>name</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td><a href="http://localhost:8008">http://localhost:8008</a></td>
</tr>
</tbody>
</table>

Running an arachni-simple_xss attack. This attack has this description:

This is a scan for cross site scripting (xss) that only runs the base xss module in arachni. The scan only crawls one level deep which makes it faster. For more depth, run the gauntlt attack alias 'arachni-simple_xss_with_depth' and specify depth.

The arachni-simple_xss attack requires the following to be set in the profile:

```
{"<url>"}
```

When I launch an "arachni-simple_xss" attack

Then the output should contain "0 issues were detected."
BATTLE-TESTED CODE WITHOUT THE BATTLE

SECURITY TESTING AND CONTINUOUS INTEGRATION

James Wickett and Gareth Rushgrove
Velocity 2014

https://speakerdeck.com/garethr/battle-tested-code-without-the-battle
Pro-Tip #3
Add Application Security Telemetry to Devs and Ops
Convert App Security Logs into Metrics in the Systems Dev and Ops Use StatsD
RUNTIME CORRELATION BETWEEN BIZ, OPS, DEV, SEC
SQLi Attempts + HTTP 500's
or
Login spikes + Transaction decrease
Runtime Instrumentation for Application Security

Signal Sciences

@WICKETT #RUGGEDDEVOPS
Pro-Tip #4
Get hugs from the auditors and add hardening and audit using config management

@WICKETT #RUGGEDDEVOPS
Open Source Hardening Framework
CheF/Puppet/Ansible

http://hardening.io/
Run Nightly Audits of your Hardening using Config Management (Chef Audit Mode)

https://www.chef.io/blog/2015/04/09/chef-audit-mode-cis-benchmarks/
OS AND CONFIG MANAGEMENT

threat stack

@WICKETT #RUGGEDDEVOPS
REVERSE THE TREND
ADD VALUE TO DEVS
ADD VALUE TO OPS
Software development has been a constant experiment in how we know anything.

Application Security abdicated runtime responsibility and effectively abdicated development responsibility through incoherent philosophical approaches and fostering organizational silos.

DevOps is here to stay, and security can actually be a part of it.

Ops found a way to add value, security needs to find that same path.

There are three ways we can add value: at development, at deploy, at runtime.
Please remember to rate this session.

Thank you!

Join the conversation #gotoLDN
Thanks !