DevOps and the Healthcare Giant

Aaron Rinehart
James Wickett

@aaronrinehart | @wickett | #devopsgiant | DOES 2017

About Aaron

Aaron Rinehart
Chief Enterprise Security Architect

UnitedHealth Group

Rinehart.Aaron@gmail.com

www.linkedin.com/in/aaronsrinehart/

@aaronrinehart

A Transformation Story

The Email that started it all

- The Need for Bold Ideas
- Demonstrate the Art of the Possible
- Inspire Transformation

The Challenge: Healthcare is Large & Complex

- **360+ Companies, 100+ Legal Entities, <20 Acquisitions /yr**
- **28,000+ Developers**
- **17,000+ Applications**
- **HIPAA, HITRUST, FISMA, MARS E 2.0, EU & Other Intl Privacy Reg++…….**
- **Diverse Technology Mix: from Mainframe to Machine Learning**
- **1000+ Security Professionals**
- **DevOps Teams – Varied Maturities**
- **Waterfall, Agile, and Scaled Agile Delivery**
- **Security Testing: Mostly Human Driven**

OPTUM®

UNITEDHEALTH GROUP®

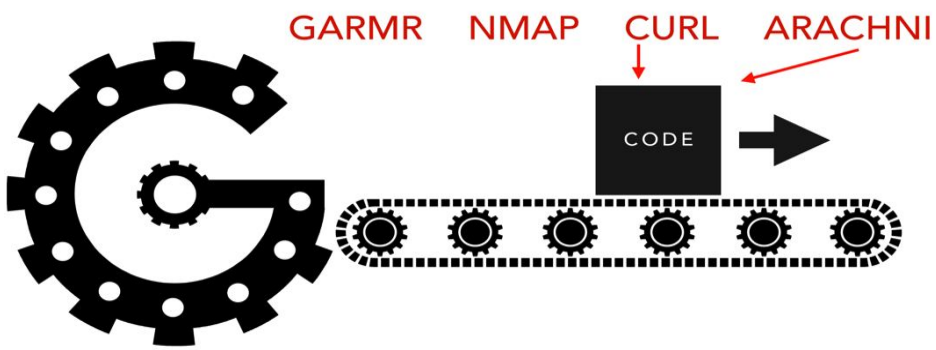
Our Journey: Developer Enablement

Develop the Tools, **Techniques** and **Processes** needed to **deliver security services** in a world of Continuous Delivery.

A New Paradigm: Bold Steps

- Drive **Security** as a Function of **Quality**
- Building a Better Model: *Continuous Delivery is Better Security*
  - Focus on **Delivering Value**
  - Continuous Security Model
  - Enable **DevOps Strategy and Automation**

Gauntlt: “Be Mean to Your Code”



Case Study: Driving Security Testing into the Pipeline: Automated Vulnerability Scanning

Automation is Important but “Don’t be Distracted by it”

Emphasize ….

Simplification & Standardization

….over Automation

Embrace Failure as a Friend

Plan and expect failure as a positive outcome. Encourage teams to fail quickly and learn from them.

James Wickett

- HEAD OF RESEARCH AT SIGNAL SCIENCES
- ORGANIZER OF DEVOPS DAYS AUSTIN
- LYNDA.COM AUTHOR ON DEVOPS
- BLOG AT THEAGILEADMIN.COM

My Journey

FIRST BIGCO JOB

- WEB AND ECOMM FOR $1B COMPANY
- BRUTAL ONCALL ROTATIONS
- +24HR DEPLOYMENTS
- WATERFALL, WATERFALL, WATERFALL
- FRIENDS ARE BORN FROM ADVERSITY

CLOUDING FOR PROFIT

- IN 2007 WENT STARTUP AND AWS CLOUD
- LEARNED A BIT ABOUT FAILURE AND HAPPINESS
- REJOINED OLD TEAM IN 2010 FOR NEW CLOUD VENTURE BACK IN BIGCO

ENTER DEVOPS

- LAUNCHED DEVOPS TRANSFORMATION AT NATIONAL INSTRUMENTS IN 2010 WITH ERNEST MUELLER, KARTHIK GAEKWAD, AND OTHERS
- INFRA AS CODE, DEV AND OPS ON SAME TEAM
- AT BIGCO DELIVERED 4 SAAS PRODUCTS IN 2 YEARS WITH DEVOPS AND CLOUD

DEVOPS AND SECURITY

- FOUND RUGGED SOFTWARE
- MET GENE KIM IN 2012 IN A BAR IN AUSTIN
- CREATED GAUNTLT
- LATER, JOINED SIGNAL SCIENCES

@aaronrinehart | @wickett | #devopsgiant | DOES 2017

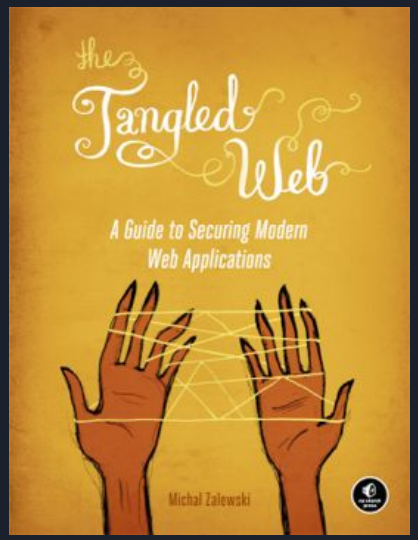
many security teams work with a worldview where their goal is to inhibit change as much as possible

Companies are spending a great deal on security, but we read of massive computer-related attacks. Clearly something is wrong.
The root of the problem is twofold:
we're **protecting the wrong things**,

and **we're hurting productivity** in the process.

THINKING SECURITY, STEVEN M. BELLOVIN 2015

[Security by 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



4 New Ways for Security Learned from our Journies

@aaronrinehart | @wickett | #devopsgiant | DOES 2017

OLD PATH VS. NEW PATH

Embrace Secrecy	Create Feedback Loops
Enforce Stability	Create Chaos
Slow Validation	Fast and Non-blocking
Certainty Testing	Adversity Testing

OLD PATH VS. NEW PATH

Embrace Secrecy	Create Feedback Loops

A security team who embraces openness about what it does and why, spreads understanding.
- Rich Smith

Runtime is arguably the most important place to create feedback loops

DETECT WHAT MATTERS

- ACCOUNT TAKEOVER ATTEMPTS
- AREAS OF THE SITE UNDER ATTACK
- MOST LIKELY VECTORS OF ATTACK
- BUSINESS LOGIC FLOWS


VAYAGIF.COM

Are you under attack?

Where?

Options: RASP, NGWAF or Web Protection Platform

OLD PATH VS. NEW PATH

Enforce Stability	Create Chaos

CHAOS ENGINEERING

- ADD IN CHAOS TO YOUR SYSTEM AND APPLICATION
- CHAOS MONKEY
- ANTI-FRAGILE
- RELEASE IT! BOOK

ChaoSlingr

Security is Chaotic

Chaos Defined

"Chaos Engineering is the discipline of experimenting on a distributed system in order to ***build confidence*** *in the system's ability to* ***withstand turbulent conditions****"*

Think Differently

"If it aint broke don't fix it"

"If it ain't broke, try harder

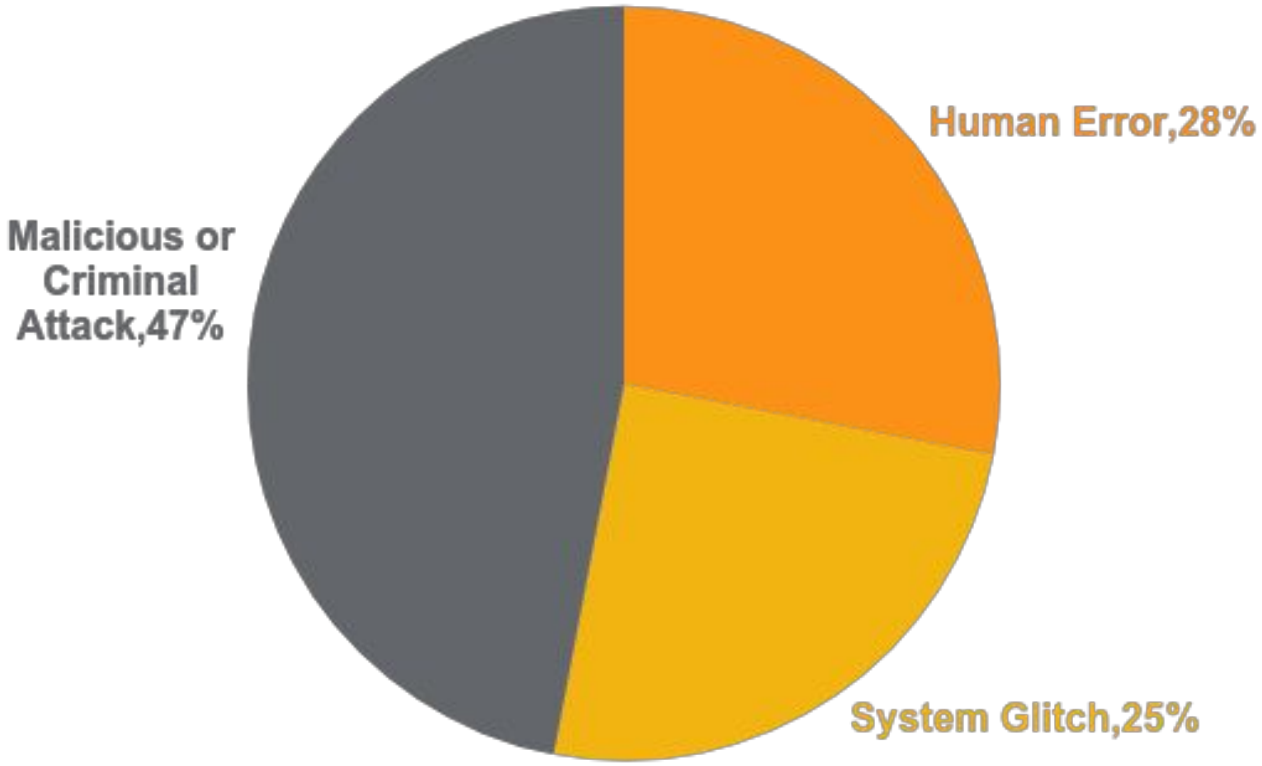
- Chaos Philosophy

@aaronrinehart | @wickett | #devopsgiant | DOES 2017

Failure Happens.

Ponemon INSTITUTE

2017 Causes of Data Breach



Bugs

Code Quality

Lack of Monitoring

Assumptions

Misconfiguration

Computer Error

Complexity in

Poor Testing Practices

Distributed Systems

Failure

Mistakes

Untested Scalability

Chang

e

Drift

Security Control

Single Points of Failure

Coverage Gaps

Vulnerabilities

Defect

Miscommunication

Failures in

External Dependencies

Human Error

Hope is Not a Strategy

@aaronrinehart | @wickett | #devopsgiant | DOES 2017

Security Incidents are **not** Detective Measures

A New Perception of Quality

Fragility: A risk of total failure/financial ruin
Resilient: Take damage, avoids total failure, recovers
Robust: Absorbs uncertainty, repels blows, avoids damage
Antifragility: Responds to stress by mutating, maintains fitness for purpose. Identity Change.
Rugged: Something built to last turbulent conditions

@aaronrinehart | @wickett | #devopsgiant | DOES 2017

CHAOS SLINGR

- ADDS MISCONFIG TO THE STACK AND CHECKS TO SEE IF IT GETS DETECTED
- NEW OPEN SOURCE TOOL!
- RUNS AS A LAMBDA

BUG BOUNTIES

- I AM BEING PEN TESTED ANYWAY, WHY NOT FIND OUT WHAT THEY ARE FINDING?
- 24/7 PEN TESTING
- BUILDS DEVELOPER CONFIDENCE
- FINDS MIX OF LOW HANGING FRUIT AND SOMETIMES MUCH MORE!

OLD PATH VS. NEW PATH

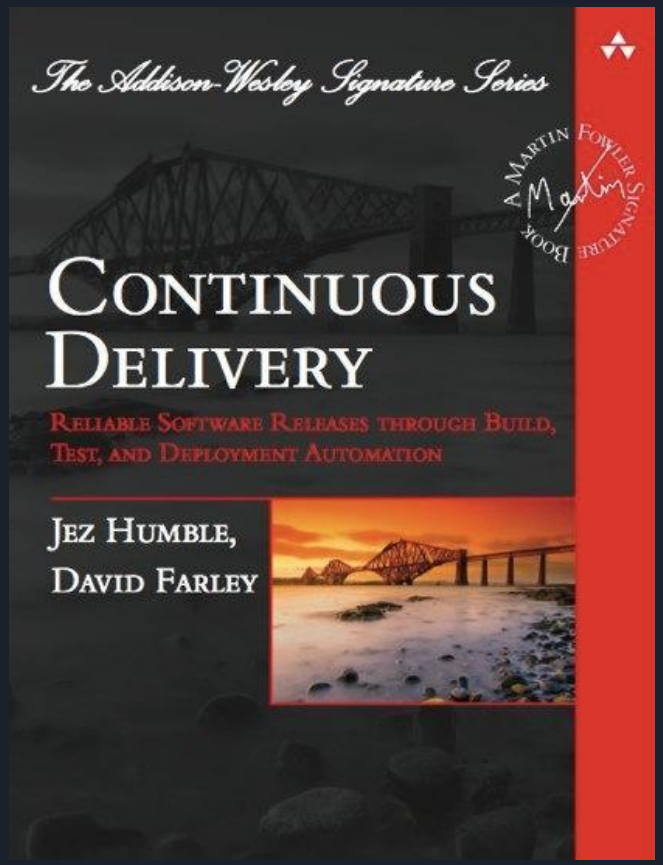
Slow Validation	Fast and Non-blocking

FAST AND NON-BLOCKING

- DON'T SLOW DELIVERY
- CONTINUOUS TESTING AND VALIDATION
- TESTING ON THE SIDE OF THE PIPELINE
- PENETRATION TESTING OUTSIDE OF DELIVERY

Currently, at Signal Sciences we do about 15 deploys per day

Roughly 10,000 deploys in the last 2.5 yrs



@aaronrinehart | @wickett | #devopsgiant | DOES 2017

CD is how little you can deploy at a time

We optimized for cycle time—the time from code commit to production

Gave power to the team to deploy

Signal Sciences is a software as a service company and a security company

Security is part of CI/CD and the overall delivery pipeline

PIPELINE PHASES

- DESIGN
- INHERIT
- BUILD
- DEPLOY
- OPERATE

SECURITY CONSIDERATIONS

- INHERIT — What have I bundled into my app that leaves me vulnerable?
- BUILD — Do my build acceptance tests and integration tests catch security issues before release?
- OPERATE — Am I being attacked right now? Is it working?

OLD PATH VS. NEW PATH

Certainty Testing	Adversity Testing

Be Mean to Your Code

@aaronrinehart | @wickett | #devopsgiant | DOES 2017

Compassion for Ops and for Security

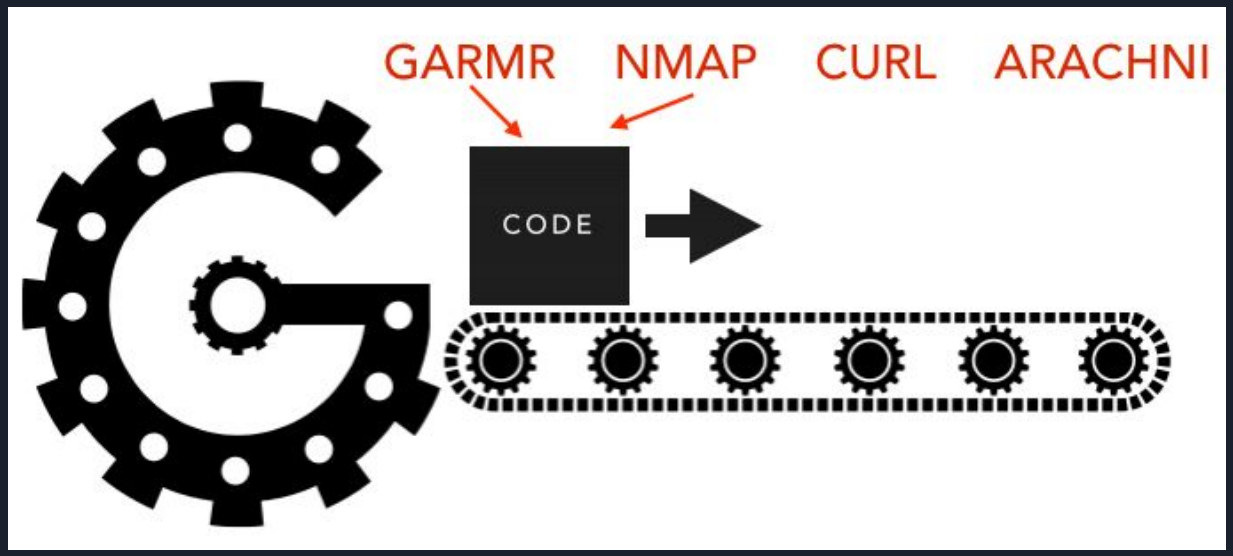
@aaronrinehart | @wickett | #devopsgiant | DOES 2017

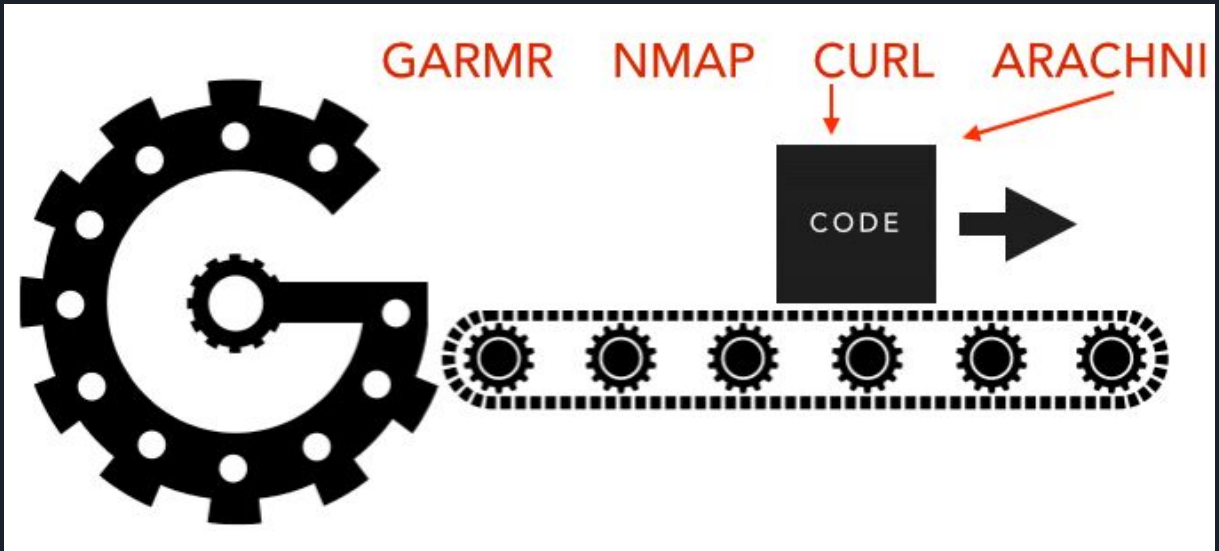
100:10:1

Dev:Ops:Sec

Gauntlt was born from trying to do Security inside a DevOps transformation inside an enterprise.

@aaronrinehart | @wickett | #devopsgiant | DOES 2017


GARMR
NMAP
CURL
ARACHNI
CODE


GARMR
NMAP
CURL
ARACHNI
CODE

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

What?

Given

When

Then

```
@slow @final
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 –check=xss* <url>
  """
  Then the output should contain "0 issues were detected."
```

gauntlt.org

http://bit.ly/2s8P1LI

WORKSHOP INCLUDES:

- 8 LABS FOR GAUNTLT
- HOW TO USE GAUNTLT FOR NETWORK CHECKS
- GAUNTLT FOR XSS, SQLI, OTHER APSES
- HANDLING REPORTING
- USING ENV VARS
- CI SYSTEM SETUP

The goal should be to come up with a set of automated tests that probe and check security configurations and runtime system behavior for security features that will execute every time the system is built and every time it is deployed.

Security tools are intractably noisy and difficult to use

A method of collaboration was needed for devs, ops and security eng.

There needed to be a new language to span the parties

Gauntlt Demo

This is a demo set of attacks that can be used to demo gauntlt and learn how to implement it. Each directory in `./examples` contains a specific type of attack that you might want to run. Inside each example you will find a README.md which will have a challenge and some hints on how to solve it. We recommend reading that first and then try to create an attack to solve the challenge.

Installation

```
$ git clone https://github.com/secure-pipeline/gauntlt-demo
$ cd ./gauntlt-demo
$ git submodule update --init --recursive
$ bundle
```

Start targets

This includes gruyere and railsgoat as a target to pratice against and in the future we will bundle other services. To start the default targets run the following.

```
$ bundle exec start_services

# For some reason railsgoat doesnt exit cleanly from a Ctl-C with service manager so yo
# will have to stop it manually
# ps -ef | grep rails
# kill -9 <PID>
# Please send a pull request if you know how to fix this
```

You can also run the following to start individual targets which include: railsgoat and gruyere

github.com/gauntlt/gauntlt-demo

Gauntlt Starter Kit

In the Gauntlt Starter Kit, you'll find scripts, examples, and some other great stuff to help you get started with Gauntlt.

How to use

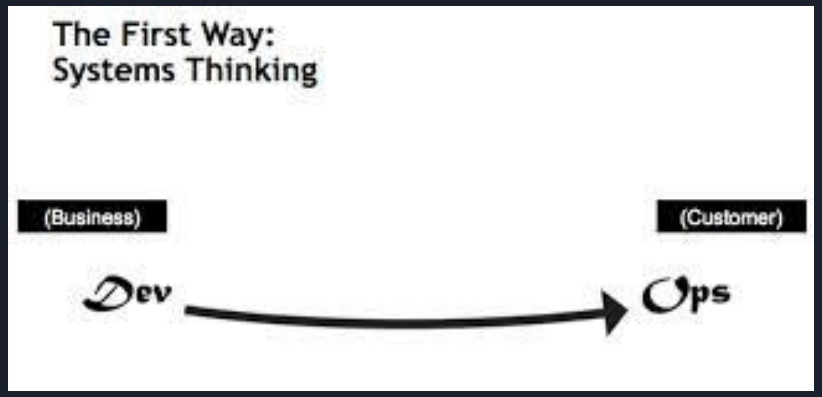
Start with a git clone of `git clone git@github.com:gauntlt/gauntlt-starter-kit` and run the following:

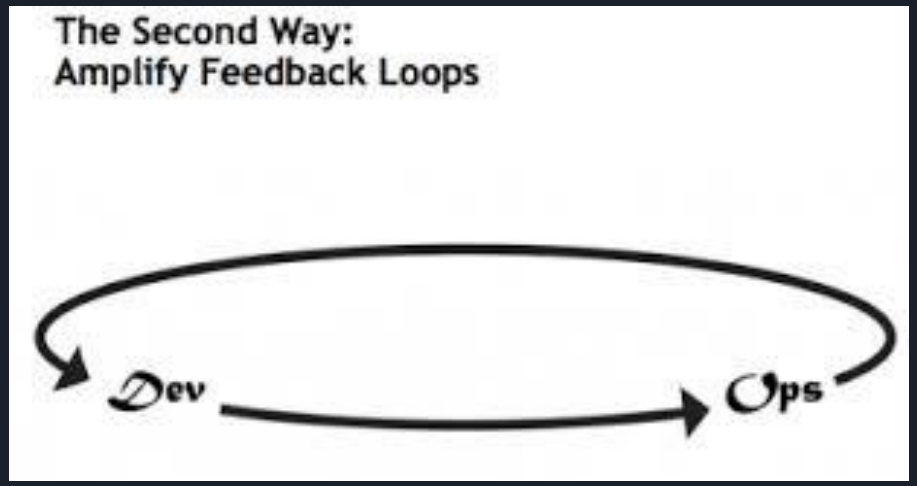
```
$ cd ./gauntlt-starter-kit/vagrant/gauntlt
$ vagrant up
$ vagrant ssh
```

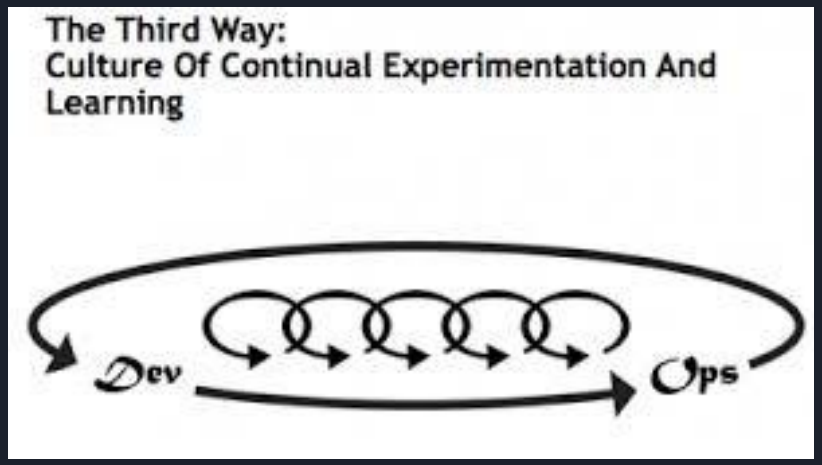
Pre-requisites

- Virtual Box
- Vagrant

github.com/gauntlt/gauntlt-starter-kit







SOURCE: THE THREE WAYS OF DEVOPS, GENE KIM

gauntlt/gauntlt-demo

build passing

this is a demo set of attacks that can be used to get started with gauntlt

Current | Build History | Pull Requests | Branch Summary

Build	51	Commit	a1e38a0 (master)
State	Passed	Compare	5c96e71da2fe...a1e38a0b1a6b
Finished	about 2 hours ago	Author	James Wickett
Duration	5 min 4 sec	Committer	James Wickett
Message	reorganizing these		

```
1   Using worker: worker-linux-10-1.bb.travis-ci.org:travis-linux-1
2
3   $ git clone --depth=50 --branch=master git://github.com/gauntlt/gauntlt-demo.git gauntlt/gauntlt-demo        git.1
11  $ cd gauntlt/gauntlt-demo
12  $ git checkout -qf a1e38a0b1a6b896265af8e21708f34ebfa1087bc                                               git.3
13  $ git submodule init                                                                                       git.4
18  $ git submodule update                                                                                     git.5
50  $ rvm use 1.9.3 --install --binary --fuzzy
51  Using /home/travis/.rvm/gems/ruby-1.9.3-p484
52  $ export BUNDLE_GEMFILE=$PWD/Gemfile
53  $ ruby --version
54  ruby 1.9.3p484 (2013-11-22 revision 43786) [x86_64-linux]
55  $ rvm --version
56
57  rvm 1.25.14 (version) by Wayne E. Seguin <wayneeseguin@gmail.com>, Michal Papis <mpapis@gmail.com> [https://rvm.io/]
58
59  $ gem --version
60  2.2.2
61  $ bundle --version
62  Bundler version 1.5.3
63  Applying fix for NPM certificates
64  $ git submodule update --init --recursive                                                                  before_install
65  $ bundle install                                                                                           install
133 $ sudo apt-get install nmap                                                                                before_script.1
161 $ sudo apt-get install wget                                                                                before_script.2
167 $ sudo apt-get install libcurl4-openssl-dev                                                               before_script.3
173 $ pwd                                                                                                      before_script.4
175 $ export SSLYZE_PATH="/home/travis/build/gauntlt/gauntlt-demo/vendor/sslyze/sslyze.py"                     before_script.5
176 $ export SQLMAP_PATH="/home/travis/build/gauntlt/gauntlt-demo/vendor/sqlmap/sqlmap.py"                     before_script.6
177 $ cd vendor/Garmr && sudo python setup.py install && cd ../..                                              before_script.7
256 $ cd vendor && wget http://downloads.sourceforge.net/project/dirb/dirb/2.03/dirb203.tar.gz && tar xvfz dirb203.tar.gz && cd dirb &&   before_script.8
```

```
459 $ export DIRB_WORDLISTS="/home/travis/build/gauntlt/gauntlt/vendor/dirb/wordlists"                before_script.
460 $ bundle exec rake
461 cd ./vendor/gruyere && ./manual_launch.sh && cd ../..
462 Gruyere started at 20097 PID and is available at localhost:8008
463 cd ./examples && bundle exec gauntlt --tags @final && cd ..
464 Using the default profile...
465 @final
466 Feature: hello world with gauntlt using the generic command line attack
467
468   Scenario:                                # ./hello_world/hello_world.attack:3
469     When I launch a "generic" attack with: # gauntlt-1.0.8/lib/gauntlt/attack_adapters/generic.rb:1
470       """
471       cat /etc/passwd
472       """
473     Then the output should contain:        # aruba-0.5.4/lib/aruba/cucumber.rb:147
474       """
475       root
476       """
477
478 @slow @final
479 Feature: Look for cross site scripting (xss) using arachni against a URL
480
481   Scenario: Using arachni, look for cross site scripting and verify no issues are found # ./arachni-xss/final_arachni-xss.attack:4
482     Given "arachni" is installed                                                         # gauntlt-1.0.8/lib/gauntlt/attack_adapters/arachni.rb:1
483     And the following profile:                                                           # gauntlt-1.0.8/lib/gauntlt/attack_adapters/gauntlt.rb:9
484       | name | value                |
485       | url  | http://localhost:8008 |
486     When I launch an "arachni" attack with:                                              # gauntlt-1.0.8/lib/gauntlt/attack_adapters/arachni.rb:5
487       """
488       arachni --modules=xss --depth=1 --link-count=10 --auto-redundant=2 <url>
489       """
490     Then the output should contain "0 issues were detected."                             # aruba-0.5.4/lib/aruba/cucumber.rb:131
491
492   Scenario: Using arachni, look for cross site scripting and verify no issues are found # ./arachni-xss/final_arachni-xss.attack:15
493     Given "arachni" is installed                                                         # gauntlt-1.0.8/lib/gauntlt/attack_adapters/arachni.rb:1
494     And the following profile:                                                           # gauntlt-1.0.8/lib/gauntlt/attack_adapters/gauntlt.rb:9
495       | name | value                |
496       | url  | http://localhost:8008 |
497 Running a arachni-simple_xss attack. This attack has this description:
498  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 specifiy depth.
499 The arachni-simple_xss attack requires the following to be set in the profile:
500  ["<url>"]
501     When I launch an "arachni-simple_xss" attack                                         # gauntlt-1.0.8/lib/gauntlt/attack_adapters/arachni.rb:9
502     Then the output should contain "0 issues were detected."                             # aruba-0.5.4/lib/aruba/cucumber.rb:131
                                                                                                                                  To top
```

Most teams use Gauntlt in Docker containers

https://github.com/gauntlt/gauntlt-docker

OLD PATH VS. NEW PATH

Embrace Secrecy	Create Feedback Loops
Enforce Stability	Create Chaos
Slow Validation	Fast and Non-blocking
Certainty Testing	Adversity Testing

DevOps and the Healthcare Giant

Aaron Rinehart
James Wickett

@aaronrinehart | @wickett | #devopsgiant | DOES 2017