Building an AppSec Program From the Ground Up: An Honest Retrospective

...

John Melton
@_jtmelton

work

- AppSec Lead at Oracle NetSuite
- Netsuite
  - “One unified business management suite, encompassing ERP/Financials, CRM and ecommerce for more than 40,000 organizations.”
  - 5,000 employees
  - $1B Revenue
  - 1,200 Developers

whoami

- Past: Dev/Security engineering in defense, finance, technology companies
- Current: Leading AppSec at technology company (Oracle NetSuite)
- Side: OWASP*, AppSensor, Manna
- Opinions are my own, not my employers

Thanks for being here!

Agenda:

An honest retrospective of the last ~2 years building an appsec program from scratch: the good, the bad, and the ugly ...

With an eye towards immediate applicability of lessons learned ...

And wrapping up with my observations and beliefs about what I would do if I had it to do all over again


Who?

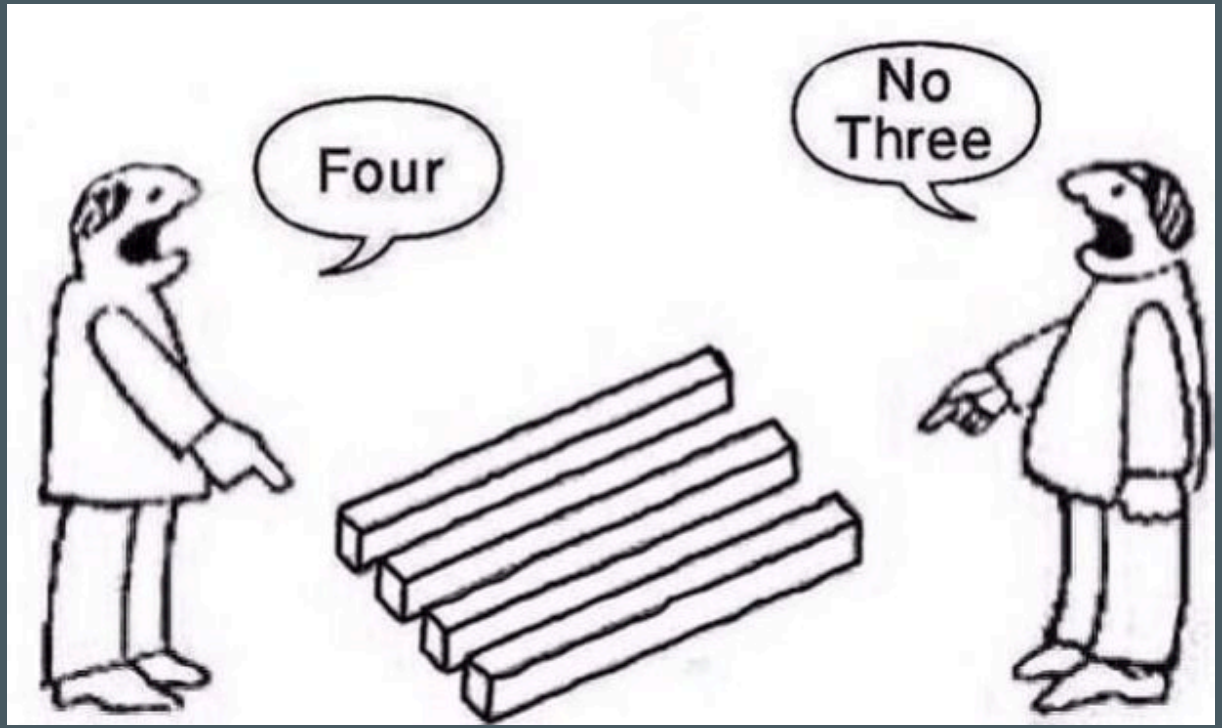

STORY

Gather round y’all, it’s ...

story time


WWW.OGRISH.COM

Context Matters



Context

Environment	Management	Me
15yr old (acquired) startup	20yr old (acquirer) startup	The new guy
Very smart engineers	Very smart engineers	meh
No “security” people	Full security team	“The” security team
CI/CD µServices / DevOps	Monolith, traditional ops	I’ve done both
Very culture-protective	Broad, varied culture	Need to get stuff done
~350 people	~5,500 people	1 person

Tasks

ToDo List (Aiming for 2yr timeline)

- Learn environment
- Embed into team / culture
- Translate
- Secure everything
- Prep for compliance
- Share / leverage successes upstream

Resources

- Tooling
- Training material
- “Policies”

Day 0


EVERYONE CALM DOWN
I GOT THIS
POSTED ON LITTLEFUN.ORG

Day 1

Lessons Learned

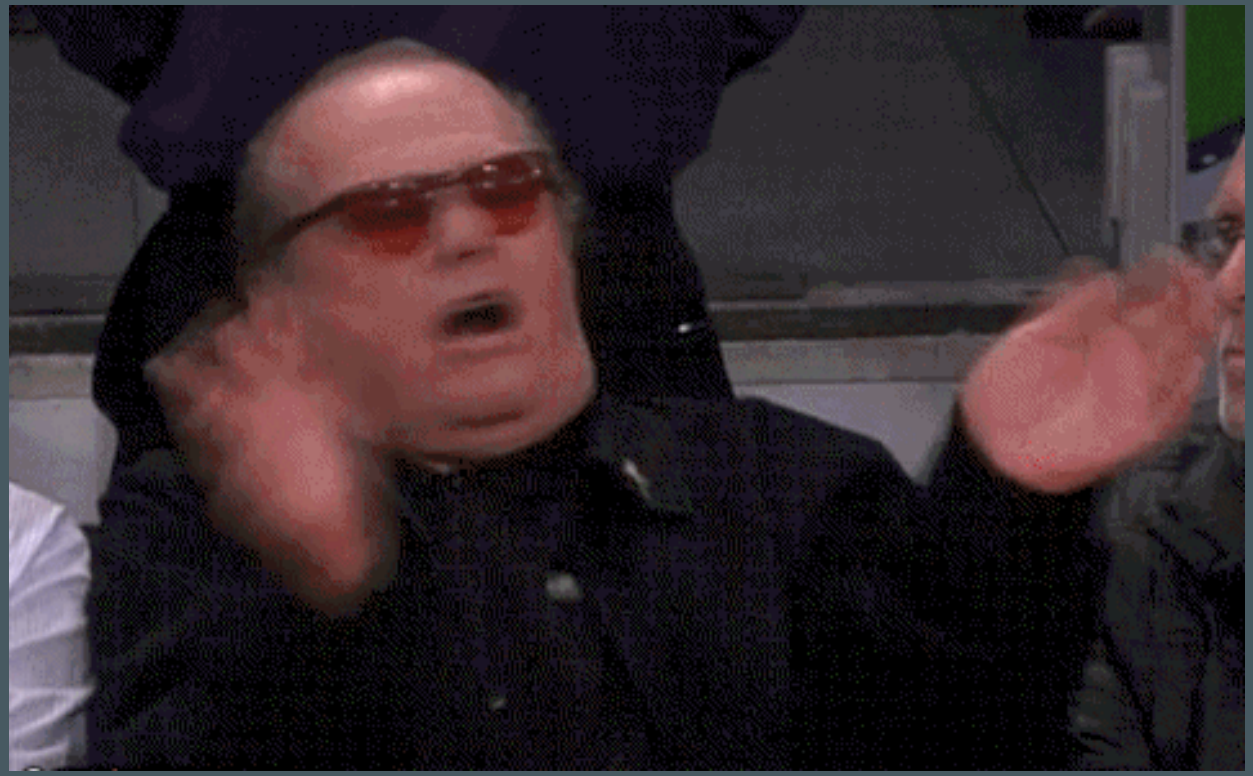
- **Be humble**

Week 1

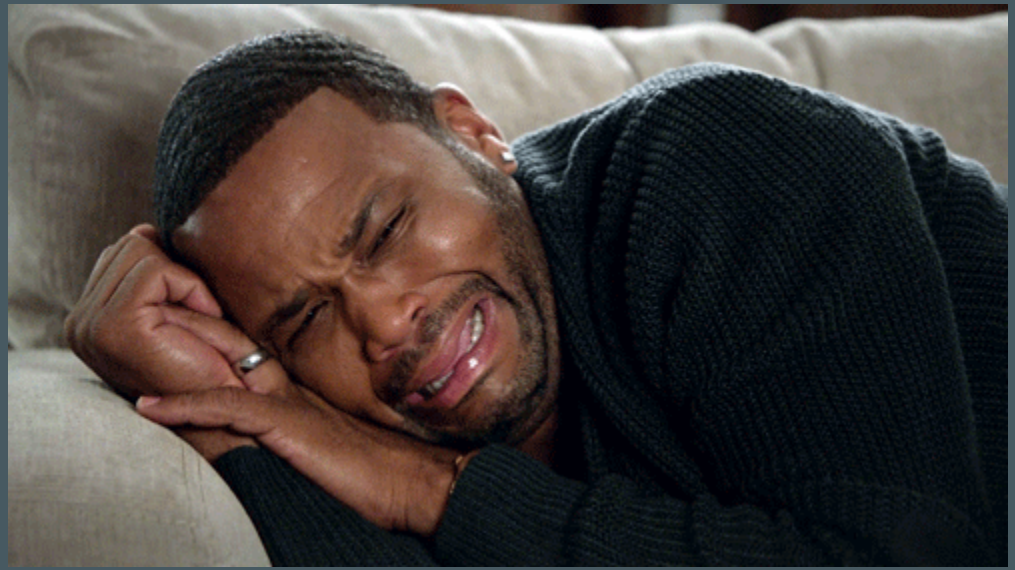

OURGANGAPPRECIATION


blameitonthecoffee.tumblr


whendoiturnbackintoapumpkin




MAKE GIFS AT GIFSOUP.COM



The First Weekend

Lessons Learned

- **The business is still running**
- **They wouldn’t be paying you if all the work was already done**
- **Take a breath**

Q1

Proposed deliverables

Introductions	• Get to know people • Get to know tech/processes
Update Training	• I've written training before • We have a pretty good base to start from
Dependency Analysis	• DependencyCheck is awesome • I've written training before • We have a pretty good base to start from

Results

Introductions (100%)	• People are great • Tech is great, Processes are good / maturing
Update Training (30%)	• Updated team HR training • Made a dent in tech training
Dependency Analysis	• DependencyCheck is awesome
	• I've written training before • We have a pretty good base to start from

Lessons Learned

- **Find key stakeholders**
- **Learn existing processes**
- **Learn the environment / tools**

Q2

Proposed deliverables

Static Analysis	• We already have a tool • I built one - setting one up should be easy!
Dynamic Analysis	• ZAP is awesome • It's just a script right?
Dependency Analysis	• DependencyCheck is awesome • Need to get a handle on these 3rd party libs
Update Training	• I've written training before • We have a pretty good base to start from

Results

Static Analysis (80%)	• A factory is hard work (especially on * apps) • Oh yeah, we should vet and explain results
Dynamic Analysis (0%)	• ZAP is still awesome • No time
Dependency Analysis (100%)	• DependencyCheck is still awesome • There's a lot of low-hanging fruit here
Update Training (-10%)	• Way more stakeholders than I thought • More work to do than initially planned

Lessons Learned

- **Start to formulate questions you want to answer (metrics) (fail)**
- **Start an application inventory (hard fail!)**
- **Tools _may_ not be the right place to start (fail)**
- **CI/CD is security's friend**
- **Get a handle on 3rd party libraries - it's low hanging (often rotten) fruit**

Q3

Proposed deliverables

Credential Storage	• Need to clean this up • Like Vault as a base
Dynamic Analysis	• Work with QA / Selenium • Need help from project leader
Champions	• Extend relationship with interested folks • Offer extra training and security info
Update SDLC	• The existing SDLC is restrictive • Need to separate policy from standard

Results

Credential Storage (100%)	• Good planning, coordination • Vault operationally works in our env
Dynamic Analysis (100%)	• ZAP/Selenium/QA automation are powerful • Project leaders want to help you succeed
Champions (50%)	• People are interested in security • No formal program, small sharing network
Update SDLC (20%)	• This is complex, and needs high-level approval • SDLC touches everything

Lessons Learned

- Pick operationally compatible tools for your env (work with eng team for input)
- Leverage open source where possible (also contribute back code, docs, stories, etc.)
- Reach out to people on github, twitter, phone, etc. - they will actually help (fail, see BSIMM)
- Share security knowledge freely (fail)
- Policies matter *

Q4

Proposed deliverables

Metrics	• Need to measure progress on desired goals • Should be easy to collect and drive change
Threat Modeling	• We should fix arch/design issues too • Aiming to start simply
Track Attack Surface	• Lots of micro-services • How are they changing?
Update SDLC	• Work is now scoped • Know stakeholders for approval

Results

Metrics (20%)	• There are lots of bad metrics • Lots of stakeholders with significant input
Threat Modeling (100%)	• Starting small was key • A picture and a list of threats
Track Attack Surface (75%)	• Surprisingly simple to support • Really helpful over time and good signal
Update SDLC (100%)	• This was a lot of work • A solid, clear policy is really helpful

Lessons Learned

- **Measuring what you are doing is critical. Communicating that information well is a challenge.**
- **Security talks about the “what’s wrong” all the time, but rarely about proactive controls (fail)**
- **People like threat modeling (be the attacker)**
- **Threat modeling gives developers power to communicate and address what keeps them up at night (fail)**


YOU'RE
HALFWAY
THERE

Progress Report

ToDo List

- Learn environment - **Done**
- Embed into team / culture - **Done, solid champions core team**
- Translate - **Mostly done**
- Secure everything - **Basics are in place, still plenty of room for improvement**
- Prep for compliance - **A little bit done**
- Share / leverage successes upstream - **Very little upstreamed**

Q5
(Acquired)

Proposed deliverables

App Deployment Tool	• Needs a refresh • Let's upgrade security at the same time
Containers	• Chance to affect greenfield deployment • Aim for reasonable default security bump
Runtime Intelligence	• Get runtime feedback about security • Empower / delegate developers to monitor
Operational Tasks	• Integrate with other teams • Produce / consume useful data

Results

App Deployment Tool (90%)	• Awesome tooling people are awesome • Platform defaults matter
Containers (75%)	• Set strong, safe defaults for great bump • Security features not always well-tested
Runtime Intelligence (10%)	• Introduced the idea and tools • Needs a “light bulb” moment
Operational Tasks (100%)	• This data is really valuable • We don’t think about this enough

Lessons Learned

- **Vault useful in many contexts, operational ability is important***
- **Reminder: availability is part of CIA triad - it is a security issue**
- **We need more runtime intel in our applications (fail)**
- **AppSec <--> OpSec is an area ripe for exploration and exploitation (fail)**
- **Went to BSIMM (world changing moment) (fail)**

* http://blog.bronto.com/engineering/tooling-microservices-for-scale-and-access/

http://blog.bronto.com/engineering/microservices-deployment-security-flexibility/

Q6

Proposed deliverables

Core Security Libraries	• Stop squashing individual bugs • Need consistent mechanisms for devs
Source Code Attestation	• Verify steps from commit -> ops • Auditability
Training Refresh	• Function-specific training • Continue simplification
CI Upgrade	• Tooling upgrade & apis • Versioned CI configuration

Results

Core Security Library (50%)	• Built/extended core systems • Partial custom rules to match
Src Code Attestation (100%)	• Requires integration work • Audit log is powerful
Training Refresh (25%)	• Customization benefits from modularity
CI Upgrade (75%)	• Config versioning is powerful • Declarative CI is powerful

Lessons Learned

- **** Killing bug classes is the useful engineering work (Fail)**
- **(git/web) Hooks are great integration points (Fail)**

Q7

Proposed deliverables

Tool Additions	• Need broader coverage • Tool vendors always behind
Immutable Infra	• Joint effort with eng • Increase stability & security
Compliance	• Has to happen • Supports business
Fast/Slow Checks	• Faster feedback on high confidence issues • Block certain classes of issues

Results

Tool Additions (100%)	• Lots of tools available these days • Single purpose tools are nice
Immutable Infra (80%)	• Big eng win • Big security win
Compliance (100%)	• Sanity check • Low bar
Fast/Slow Checks (100%)	• Further left • Lots of low-hanging fruit here

Lessons Learned

- **Completely fixing and forever preventing an issue (even a small one) is a win - combine these to get momentum**
- **Create minimum assertions and raise the bar**
- **Config mgmt / Infra mgmt are powerful**
- **The faster devs see the issue relative to coding it, the faster (and better!) the fix**

——**(Fail)**

Q8

Proposed deliverables

Refresh SDLC	• Maturity step • Chance to move left / increase visibility
App Portfolio	• Do a better job collecting metadata / metrics • Single, consistent view
Compliance	• More, more, more
Fast/Slow Checks	• Faster feedback on high confidence issues • Block certain classes of issues

Proposed deliverables

Refresh SDLC (100%)	• Tailor to different stakeholders • Talk about privacy alongside security
App Portfolio (20%)	• Complex area • Missing tool support
Compliance (100%)	• More, more, more
Fast/Slow Checks	• Faster feedback on high confidence issues • Block certain classes of issues

Lessons Learned

- **Privacy is something everybody cares about (Fail)**
- **App Portfolio is ripe for a solution, many people are struggling in that area**
- **Hard to secure what you don’t know about**

Progress Report

ToDo List

- Learn environment - **Done**
- Embed into team / culture - **Done, solid champions core team**
- Translate - **Done**
- Secure everything – **We’re further, but always room for improvement**
- Prep for compliance – **Oh boy … did we ever**
- Share / leverage successes upstream – **Good progress**


What am I supposed to do now?
STOP
WRONG WAY
DO NOT ENTER

TODOs - People

- Work with the best people you can
- Do small, focused, context-sensitive training
- Connect with tribal knowledge owners
- Build a real champions program
- Say “no” rarely
- Know your place and stay humble - security is not the only business concern
- Connect with others doing your job in security at cons, social media, etc. Ask them questions.
- Talk about security in regular life (e.g. https://securityplanner.org/)
- Talk about security often (chat, email, presentations, etc.)
- Talk about privacy
- Talk issues at the highest level possible - exec buy-in is critical

TODOs - Process

- Have office hours, chatops, email list - be available
- Never start with “no”. Default to “how can we get to yes”?
- Use the champions program
- Actively build relationships with other teams
- Collect useful data to improve and build a data-driven process
- Live and breathe threat modeling
- Inject into the standard SDLC (not as a blocker though)
- Reqs > Arch > Design > Code
- Use consistent terminology (words matter)
- Develop method for ranking apps
- Develop method for ranking vulns
- Meet customers where they are (chat, email, wiki, bugtracker, etc.)

TODOs - Technology

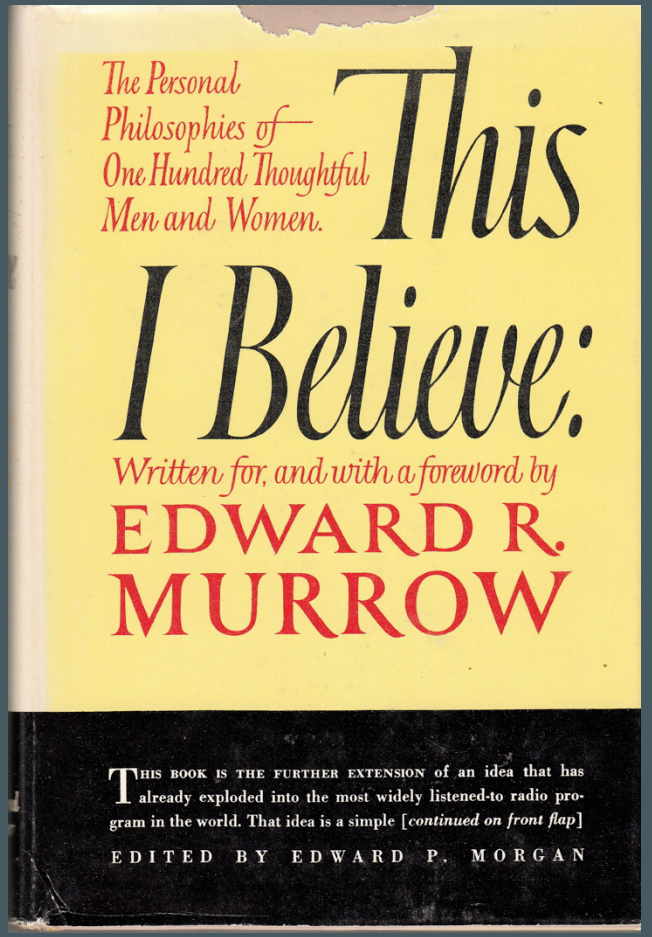
- Don’t have tech envy
- Isolate security services (e.g. encryption as a service)
- Exploit CI (fast/slow lanes)
- Squash bug classes, not bugs (bug of the month/quarter, top “1”)
- Support / amplify good tech from devs (containers, cloud, etc.)
- Build a solid app inventory
- Focus on and invest in self-service
- Spend time on the “big” things (cloud, 2/MFA, IAM, Authn/z, crypto)
- Limit crypto primitives (e.g. nacl)
- Support primary tech stacks well
- Work with dev and ops to get runtime info, and create a feedback loop
- Build self-defending apps (appsensor)

Some Homework

- “Starting Up Security” by Ryan McGeehan (https://medium.com/starting-up-security)
- “Preventing Security Bugs Through Software Design” by Christoph Kern (https://www.youtu.be/ccfEu-Jj0as)
- Measuring End-to-End Security Engineering by Garrett Held (https://www.youtu.be/MLmQ4uSi4EU)
- “Software Security Metrics” (https://youtu.be/50vOxExpAOU) and “Effective AppSec Metrics (https://youtu.be/dY8IuQ8rUd4), by Caroline Wong
- Starting a Metrics Program by Marcus Ranum (https://youtu.be/yW7kSVwucSk)
- Enabling Product Security with Culture and Cloud by Astha Singhal and Patrick Thomas (https://youtu.be/L1WaMzN4dhY)

Summary

This I Believe



This I Believe

You can’t do appsec effectively and not understand code. You should be able to read & write it. Same with design & architecture.

This I Believe

Security teams don’t scale effectively, even with significant automation efforts. Build a champions program, and move to self-service.

This I Believe

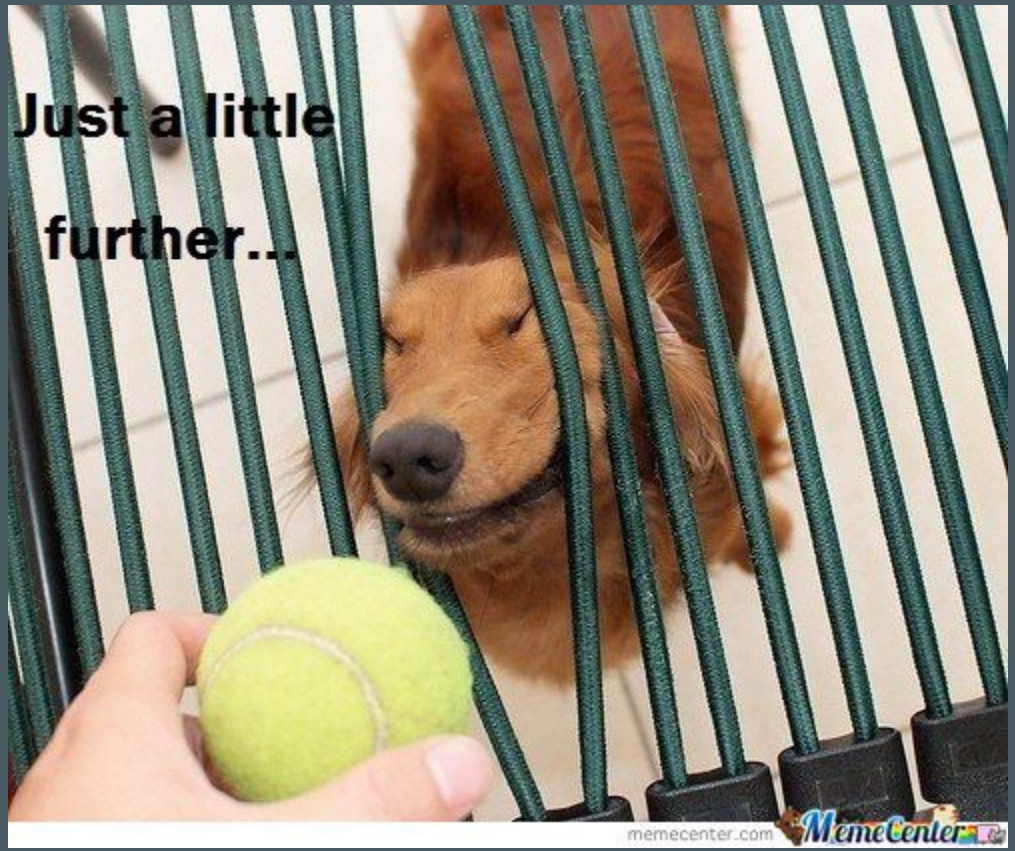
We need more focus on detection and response, not just prevention.

This I Believe

Certain solutions (executive support/ buy-in, 2FA, CI/CD automation, threat modeling) move the needle _much_ farther - focus on those.

This I Believe

You cannot protect what you don’t know about - build an app inventory.


Just a little
further...
memecenter.com
MemeCenter

One more thing...


GIVE
BACK

Open Source

- We don’t all have huge teams
- Most/all of us use open source docs, software
- Many of us build useful things
- Some of us can contribute at work
- Many of us can contribute at home

- You benefit the community
- You grow
- You build awareness of issues
- You help drive solutions
- You benefit personally and professionally

Introducing …

Manna

Create Sample.java

Browse files

master

jtmelton committed 24 days ago Verified

1 parent d06a931 commit 44a6bb8d31dc5ac32be90ff040656f1eaf06df47

Showing 1 changed file with 26 additions and 0 deletions.

Unified Split

26 src/main/java/com/jtmelton/testme/Sample.java View

```
@@ -0,0 +1,26 @@
+package com.jtmelton.testme;
+
+public class Sample {
+    public static void main(String[] args) {
+        boolean s = true;
+
+        if (s = false) {  //should produce finding - HERE
+            int i1 = "555";
+            System.err.println("assignment");
+        }
+
+        if (s == false) {
+            System.err.println("compare");
+        }
+
+        if (s = true || 1 == 1) { //should produce finding - HERE
+            System.err.println("a1");
+        } else if (1 == 2 || (s = true)) {  //should produce finding - HERE
+            System.err.println("a2");
+        } else if (1 == 3 && (s = false || 1 == 4)) { //should produce finding - HERE
+            System.err.println("a3");
+        }
+
+        System.out.println("Hello World!");
+    }
+}
```

[manna] Automated Security Analysis #3

Edit

Open **manna-bot** wants to merge 1 commit into `jtmelton:master` from `manna-bot:master`

Conversation 0 | Commits 1 | Files changed 1

Changes from **all commits** | Jump to... | +5 −4 | Unified | Split | Review changes

9 src/main/java/com/jtmelton/testme/Sample.java — View

```
@@ -4,7 +4,7 @@
4  4      public static void main(String[] args) {
5  5          boolean s = true;
6  6
7     -        if (s = false) {  //should produce finding - HERE
   7  +        if (s == false) {  //should produce finding - HERE
8  8              int i1 = "555";
9  9              System.err.println("assignment");
10 10         }
@@ -13,14 +13,15 @@ public static void main(String[] args) {
13 13             System.err.println("compare");
14 14         }
15 15
16    -        if (s = true || 1 == 1) { //should produce finding - HERE
   16 +        if (s == true || 1 == 1) { //should produce finding - HERE
17 17             System.err.println("a1");
18    -        } else if (1 == 2 || (s = true)) {  //should produce finding - HERE
   18 +        } else if (1 == 2 || (s == true)) {  //should produce finding - HERE
19 19             System.err.println("a2");
20    -        } else if (1 == 3 && (s = false || 1 == 4)) { //should produce finding - HERE
   20 +        } else if (1 == 3 && (s == false || 1 == 4)) { //should produce finding - HERE
21 21             System.err.println("a3");
22 22         }
23 23
24 24         System.out.println("Hello World!");
25 25     }
26 26 }
   27 +
```

```
7   -       if (s = false) {  //should produce finding - HERE
7   +       if (s == false) {  //should produce finding - HERE
```

This I Believe

We all have something useful to contribute to the community.

This I Believe

- Docs - https://medium.com/starting-up-security
- Talks - https://www.youtube.com/watch?v=ccfEu-Jj0as
- Scripts - https://github.com/jgamblin/AWSScripts
- Code - https://github.com/jeremylong/DependencyCheck
- Organization - https://github.com/sbilly/awesome-security and https://github.com/paragonie/awesome-appsec
- Work - https://github.com/nccgroup/Scout2 and https://www.nccgroup.trust/us/our-research/understanding-and-hardening-linux-containers/

This I Believe

Manna

Simple static analysis & automatic remediation

- **https://github.com/manna-security**
- **Open Source (Apache 2)**
- **Alpha release (only 1 rule)**
- **Please contribute!**

Some Homework

- Champions Program
- Self-Service Software
- Focus on Detection and Response
- Need an App Inventory Solution
- Contribute to Open Source


That's all Folks!

Questions ?