Monitoring Attack Surface to Secure DevOps Pipelines

Dan Cornell

@danielcornell

© 2016 Denim Group – All Rights Reserved

DENIM GROUP

Agenda

- Background
- Importance of Attack Surface
- What Does Attack Surface Have to Do with DevOps?
- Hybrid Analysis Mapping (HAM) Background
- Installation Instructions
- Use Cases
- Questions

© 2016 Denim Group – All Rights Reserved

My Background

- Dan Cornell, founder and CTO of Denim Group
- Software developer by background (Java, .NET, etc)
- OWASP San Antonio
- OWASP OpenSAMM Benchmark

© 2016 Denim Group – All Rights Reserved

Denim Group Background

- Secure software services and products company
  - Builds secure software
  - Helps organizations assess and mitigate risk of in-house developed and third party software
  - Provides classroom training and e-Learning so clients can build software securely
- Software-centric view of application security
  - Application security experts are practicing developers
  - Development pedigree translates to rapport with development managers
  - **Business impact: shorter time-to-fix application vulnerabilities**
- Culture of application security innovation and contribution
  - Develops open source tools to help clients mature their software security programs
    - *Remediation Resource Center, ThreadFix*
  - OWASP national leaders & regular speakers at RSA, SANS, OWASP, ISSA, CSI
  - World class alliance partners accelerate innovation to solve client problems

© 2016 Denim Group – All Rights Reserved

OWASP ZAP

- Open source web proxy and dynamic application security testing tool
- https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project

© 2016 Denim Group – All Rights Reserved

Example Codebases

- BodgeIt Store
  - Example vulnerable web application
  - https://github.com/psiinon/bodgeit
- Java Spring Petstore
  - Example Spring application
  - https://github.com/spring-projects/spring-petclinic
- Railsgoat
  - Example vulnerable web application
  - https://github.com/OWASP/railsgoat

© 2016 Denim Group – All Rights Reserved

ThreadFix Community Edition

- Application vulnerability management
  - And some other stuff
- https://github.com/denimgroup/threadfix

© 2016 Denim Group – All Rights Reserved

Downloads

- https://dl.dropboxusercontent.com/u/737351/endpoints-json.jar
- https://dl.dropboxusercontent.com/u/737351/threadfix-release-2.zap
- https://github.com/denimgroup/threadfix-examples/tree/master/web_app_attack_surface

© 2016 Denim Group – All Rights Reserved

Importance of Attack Surface

© 2016 Denim Group – All Rights Reserved

DENIM GROUP

Importance of Attack Surface

- This is where an attacker can “reach out and touch” your application
  - Web: Mostly in the HTTP request: URL, parameters, headers (cookies)
  - Mobile, IoT: More complicated
  - We will focus on web today
- Target for dynamic testing
  - Automated DAST
  - Manual assessment/penetration testing

© 2016 Denim Group – All Rights Reserved

What Does Attack Surface Have to Do With DevOps?

- ~~If you want your talk to be accepted, it has to have DevOps in the title~~

- Let's look at what we want from security in the DevOps pipeline

© 2016 Denim Group – All Rights Reserved

Security in the DevOps Pipeline

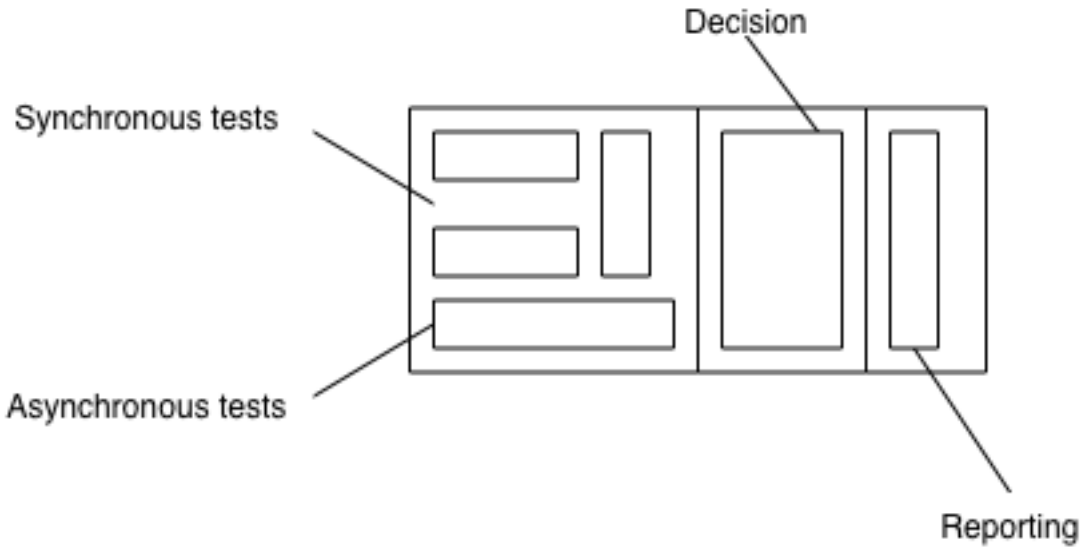
Organizations like Etsy and Netflix are doing *amazing* things to secure apps via their DevOps pipelines

© 2016 Denim Group – All Rights Reserved

DENIM GROUP

Security in the DevOps Pipeline

- Testing
  - Synchronous
  - Asynchronous
- Decision
- Reporting



© 2016 Denim Group – All Rights Reserved

DENIM GROUP

Focus on Testing in DevOps Pipeline

- Many security tools run too long to include in many pipeline builds
  - Full SAST, DAST
- Security testing also includes manual testing
  - Which is *way* too slow for most pipeline builds

- Tracking attack surface changes over time can help us:
  - Focus testing activities
  - Trigger testing activities

© 2016 Denim Group – All Rights Reserved

Hybrid Analysis Mapping

- Goal: Merge the results of SAST and DAST testing
- Funded via DHS S&T SBIR contracts
- Facilitated the creation of our attack surface modeling engine

© 2016 Denim Group – All Rights Reserved

Department of Homeland Security Support

- Currently in Phase 2 of a DHS S&T CSD SBIR
- Acronyms!
  - DHS = Department of Homeland Security
  - S&T = Directorate of Science and Technology
  - CSD = CyberSecurity Division
  - SBIR = Small Business Innovation Research
- Geared toward developing new technologies for Federal customers

Homeland Security

Science and Technology

- Hybrid Analysis Mapping (HAM)
- Technology has been included with ThreadFix
- Has also resulted in some other released components we will talk about today

- **Please do not assume this talk is endorsed by DHS**
  - **This is just me talking about what we have done**

© 2016 Denim Group – All Rights Reserved

Hybrid Analysis Mapping (HAM)

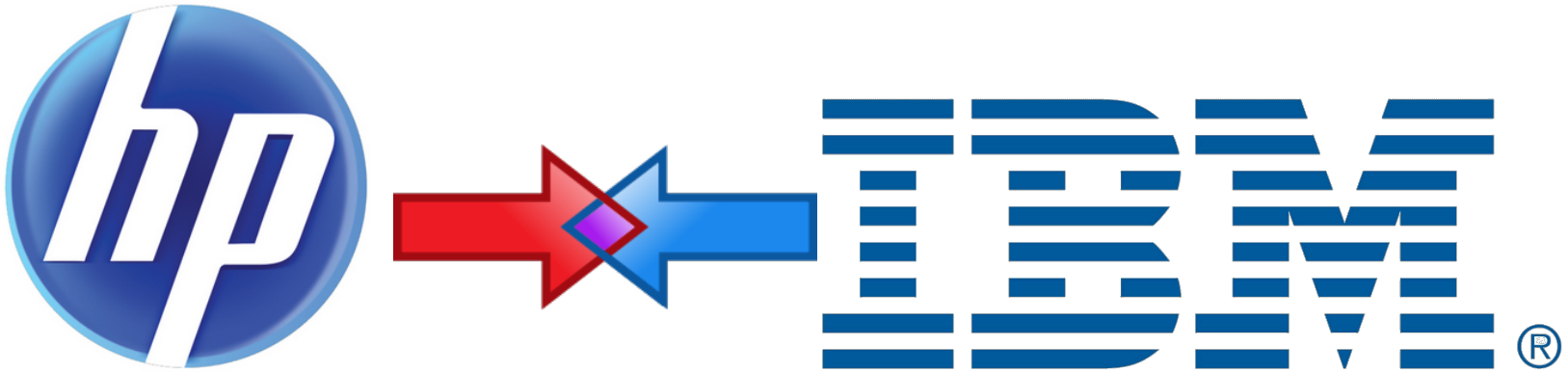
- Initial goal: Correlate and merge results from SAST and DAST

- After we made that work, we found other stuff we could do with the technology

© 2016 Denim Group – All Rights Reserved

Hybrid Analysis Mapping (HAM)

- **Determine the feasibility of developing a system that can reliably and efficiently correlate and merge the results of automated static and dynamic security scans of web applications.**



HP Fortify SCA Standard

IBM AppScan

© 2016 Denim Group – All Rights Reserved

Dynamic Application Security Testing (DAST)

- Spider to enumerate attack surface
  - Crawl the site like Google would
  - But with authentication / session detection
- Fuzz to identify vulnerabilities based on analysis of request/response patterns
  - If you send a SQL control character and get a JDBC error message back, that could indicate a SQL injection vulnerability
- A finding looks like (CWE, relative URL, [entry point])

© 2016 Denim Group – All Rights Reserved

Static Application Security Testing (SAST)

- Use source or binary to create a model of the application
  - Kind of like a compiler or VM
- Perform analysis to identify vulnerabilities and weaknesses
  - Data flow, control flow, semantic, etc
- A finding looks like (CWE, code/data flow)

```
String username = request.getParameter("username");
String sql = "SELECT * FROM User WHERE username = '" + username + "'";

Statement stmt;
stmt = con.createStatement();
stmt.execute(sql);
```

© 2016 Denim Group – All Rights Reserved

DENIM GROUP

Hybrid Analysis Mapping Sub-Goals

- Standardize vulnerability types
  - Settled on MITRE Common Weakness Enumeration (CWE)
- Match dynamic and static locations
  - Use knowledge of language/web framework to build attack surface database
- Improve static parameter parsing
  - Parse out of source code to match with DAST result

© 2016 Denim Group – All Rights Reserved

Information Used

- Source Code
  - Git, Subversion, Local Copy
- Framework Type
  - Java: JSP, Spring, Struts
  - C#: .NET WebForms, .NET MVC
  - Ruby: Rails
  - PHP: in progress
- Extra information from SAST results (if available)

© 2016 Denim Group – All Rights Reserved

Unified Endpoint Database

- EndpointQuery
  - dynamicPath
  - staticPath
  - Parameter
  - httpMethod
  - codePoints [List<CodePoint>]
  - informationSourceType

- EndpointDatabase
  - findBestMatch(EndpointQuery query): Endpoint
  - findAllMatches(EndpointQuery query): Set<Endpoint>
  - getFrameworkType(): FrameworkType

© 2016 Denim Group – All Rights Reserved

Merging SAST and DAST Results

- I have a DAST result:
  - (“Reflected XSS”, /login.jsp, “username” parameter)
- Query the Endpoint Database:
  - Entry point is com.something.something.LoginController.java, line 62
- Search the other findings for SAST results like:
  - (“Reflected XSS”, source at com.something.something.LoginController.java, line 62)

- If you find a match – correlate those two findings
- Magic!

© 2016 Denim Group – All Rights Reserved

That's Great But I Want More

- So our research produced a successful/valuable outcome
  - Hooray
- But – given these data structures, what else can we do?

- From an EndpointDatabase we can:
  - Get *all* of the application's attack surface
  - Map DAST results to a specific line of code

- Given those capabilities we can:
  - Pre-seed scanners with attack surface
  - Map DAST results to lines of code in a developer IDE
  - Map DAST results to lines of code in SonarQube

© 2016 Denim Group – All Rights Reserved

Scanner Seeding

- What if we could give the DAST spidering process a head start?
- Pre-seed with *all* of the attack surface
  - Landing pages that link in to the application
  - Hidden directories
  - Backdoor or "unused" parameters

- Currently have plugins for OWASP ZAP and BurpSuite
  - Plugin for IBM Rational AppScan Standard is in progress

https://github.com/denimgroup/threadfix/wiki/Scanner-Plugins

© 2016 Denim Group – All Rights Reserved

Final Thoughts on SBIR Work with DHS S&T

- Great use of the SBIR program
  - In my humble and *totally* unbiased opinion

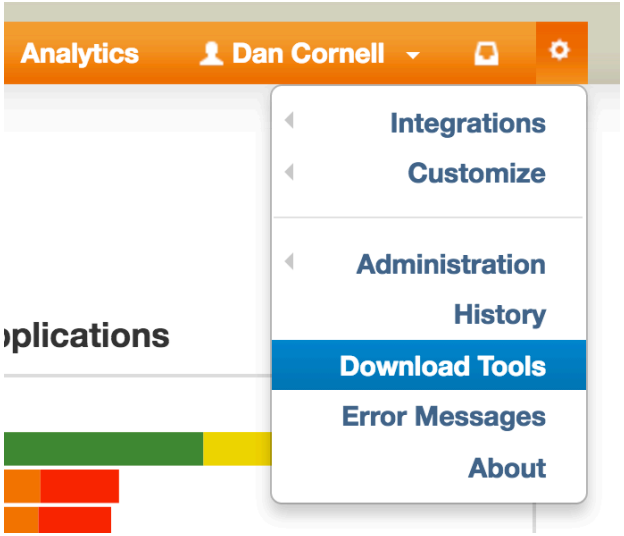
- Proved to be the tipping point to developing HAM
  - HAM was *interesting*, but required material investment

- Research produced a successful outcome (we think)
- We found other things we could do with the technology
- Released much of it open source to increase adoption

© 2016 Denim Group – All Rights Reserved

Getting the Plugin

- Main ThreadFix site
  - https://github.com/denimgroup/threadfix/
- ThreadFix build instructions
  - https://github.com/denimgroup/threadfix/wiki/Development-Environment-Setup
  - "Running ThreadFix Without an IDE"
- Download plugins from ThreadFix



© 2016 Denim Group – All Rights Reserved

Plugin Installation Instructions

- OWASP ZAP plugin installation instructions
  - https://github.com/denimgroup/threadfix/wiki/Zap-Plugin

- Plugins also available for:
  - Portswigger BurpSuite Professional
  - IBM Rational AppScan (soon)

© 2016 Denim Group – All Rights Reserved

Attack Surface Enumeration

- Find *all* of the attack surface
  - URLs
  - Parameters that will change application behavior
  - Future: Cookies, other HTTP headers
- Why is this a problem?
  - Hidden landing pages
  - Multi-step processes that automated crawls don't traverse
  - Unknown parameters
  - Debug/backdoor parameters (will discuss this further)
- Great for REST APIs support single-page web applications and mobile applications

© 2016 Denim Group – All Rights Reserved

Attack Surface Enumeration Benefits

- Reduce false negatives from scanners
  - Better coverage for standard fuzzing

- Pen test *all of* the application

© 2016 Denim Group – All Rights Reserved

Endpoints CLI Notes

- Syntax: java –jar [jar-name].jar /path/to/source

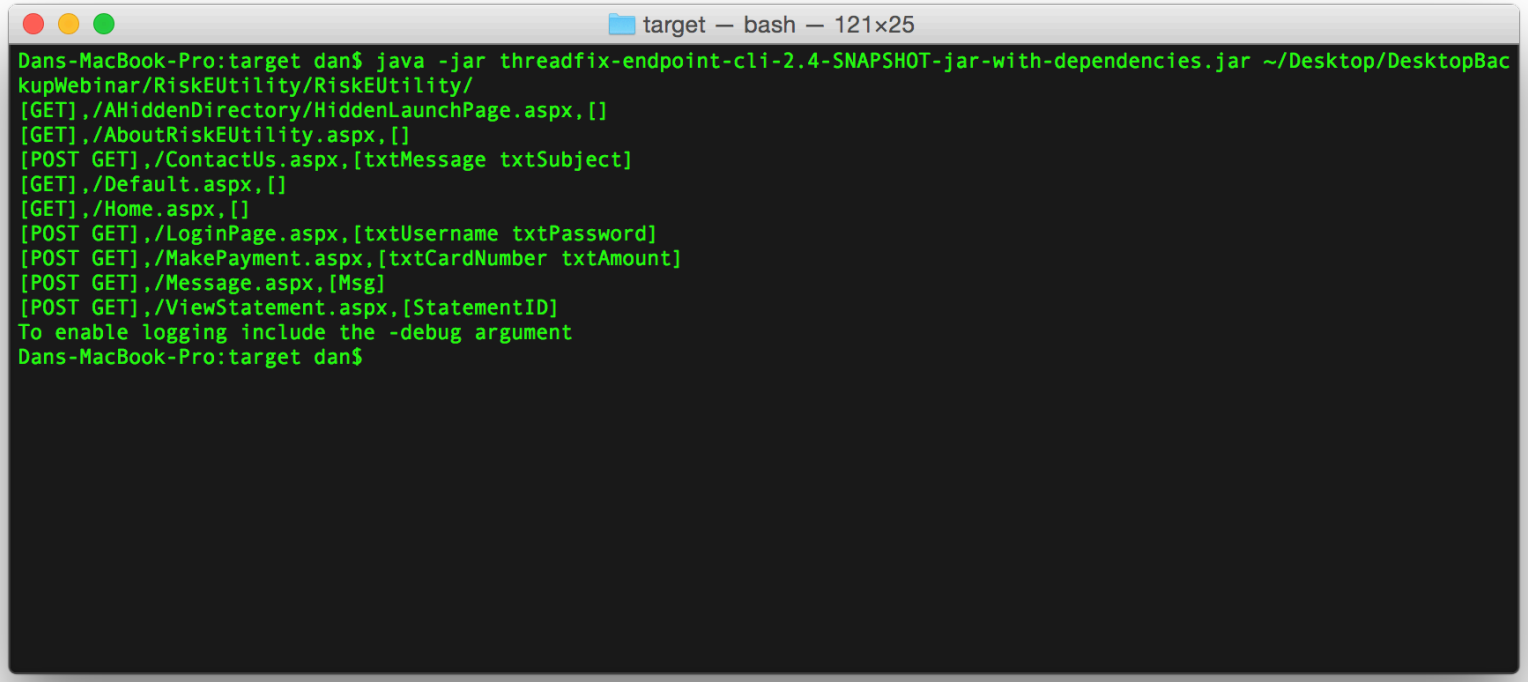
- JAR name will change based on build ID
- After Maven build, can also be found in: $GIT/threadfix/threadfix-cli-endpoints/target/
- You want the ”-jar-with-dependencies” JAR

- Will output list of HTTP methods, URLs and parameters based on analysis of the source code
- Attack surface!

- Add “-json” to the end of the command to get output in JSON format
  - Easier to manipulate

© 2016 Denim Group – All Rights Reserved

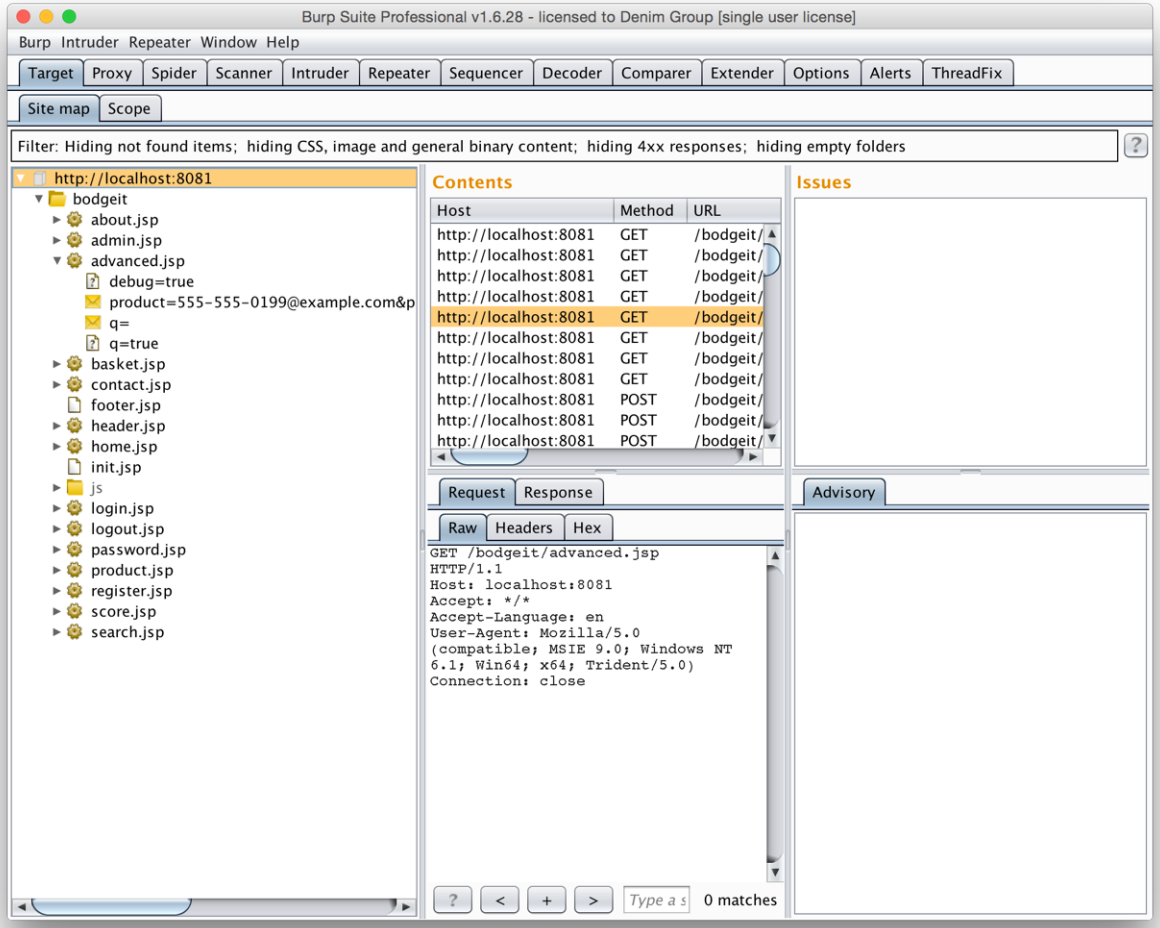
Command Line Demo



© 2016 Denim Group – All Rights Reserved

DENIM GROUP

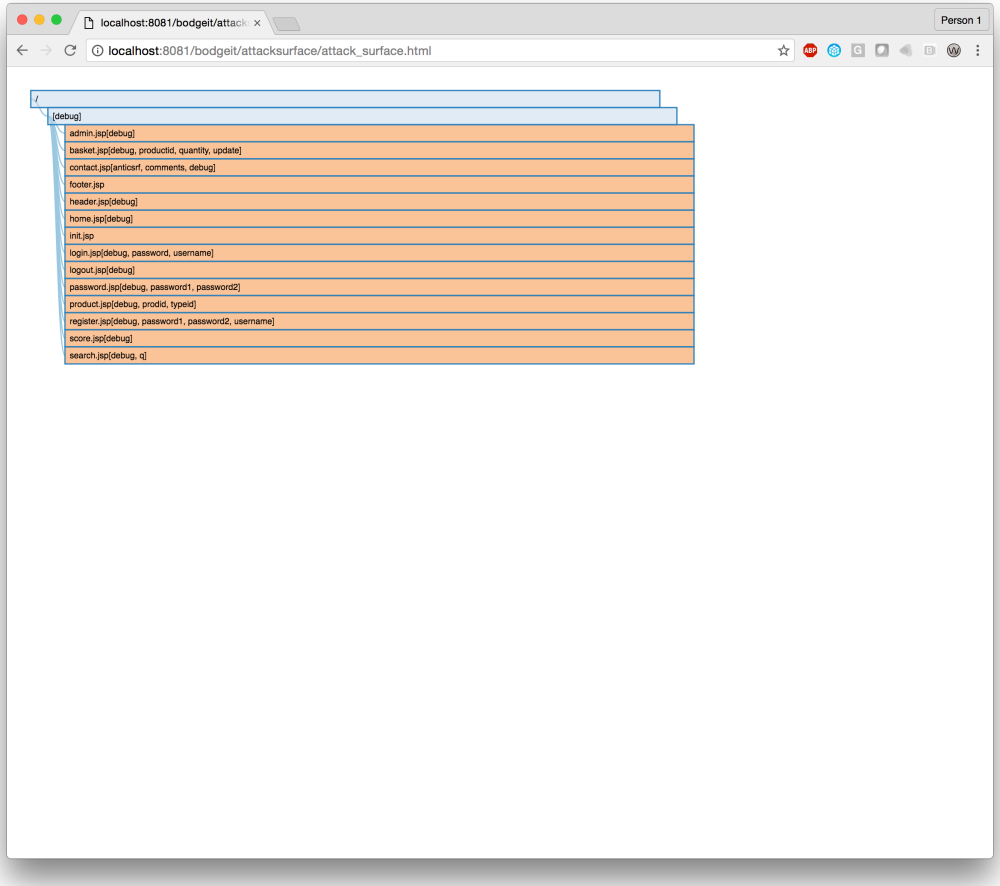
Scanner Attack Surface Seeding Demo



© 2016 Denim Group – All Rights Reserved

DENIM GROUP

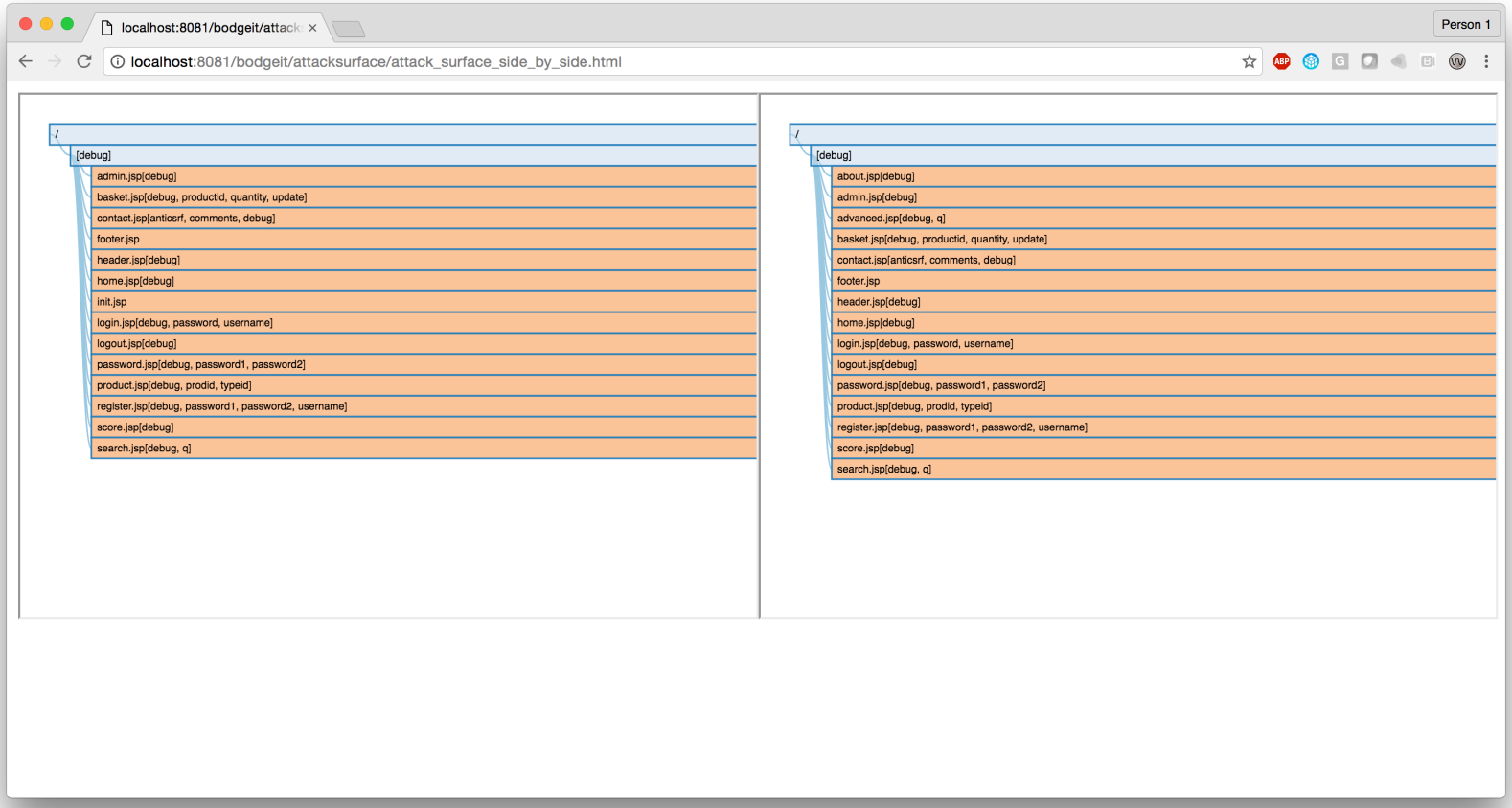
Attack Surface Visualization Demo



© 2016 Denim Group – All Rights Reserved

DENIM GROUP

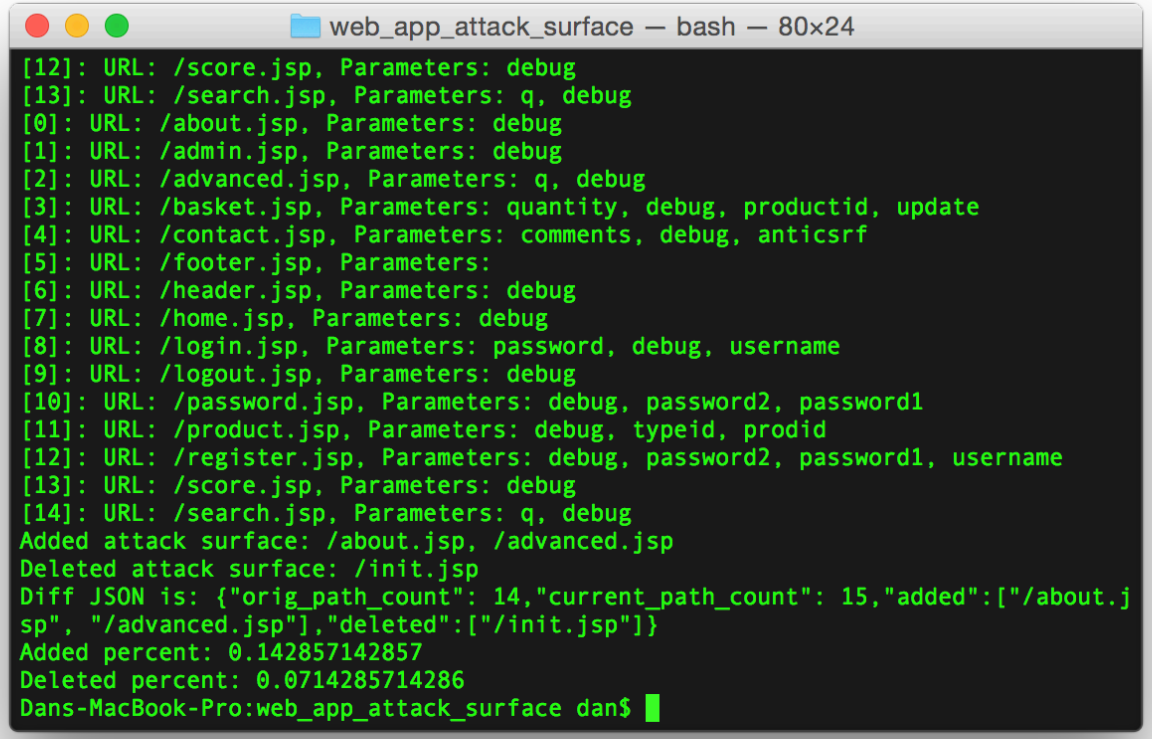
Attack Surface Comparison Visualization Demo



© 2016 Denim Group – All Rights Reserved

DENIM GROUP

Diffing Attack Surface Demo



© 2016 Denim Group – All Rights Reserved

DENIM GROUP

Applications for DevOps Pipelines

- Target DAST testing to focus on new attack surface in latest build
  - "Run an authenticated ZAP scan against the three new URLs added in the last commit"
- Set thresholds for when manual assessment/penetration testing is triggered
  - "Schedule a manual penetration test when the attack surface has increased by 10 URLs"
  - "Schedule a manual penetration test when the attack surface has increased by 5%"
  - Focus those efforts on new attack surface
- ChatOps: Attack surface delta notifications on commit
  - "Commit beb78c835706efe5d619148b9a8dc9e35ee9572b added attack surface: /advanced.jsp, /preferenes.jsp"

© 2016 Denim Group – All Rights Reserved

Next Steps

- Expand the model of application attack surface
  - Currently: Parameters, HTTP verbs
  - Working on: HTTP headers (cookies)
  - Future: Other application types: Mobile, IoT
- Better visualization
  - More details
  - Better granularity
- Native integrations: Jenkins, Slack, HipChat, etc
  - This is kind of “scripty” right now

© 2016 Denim Group – All Rights Reserved

Questions / Contact Information

Dan Cornell

Principal and CTO

dan@denimgroup.com

Twitter @danielcornell

(844) 572-4400

www.denimgroup.com

www.threadfix.it

© 2016 Denim Group – All Rights Reserved