APP SEC IN THE TIME OF DOCKER CONTAINERS

Akash Mahajan - Director Appsecco

COCON 2016

What is this talk about?

- Assertion that application security folks need to be aware of containers and what can they start with now
- Lets define typically what happens in application security
- Lets also quickly try and understand what are containers and especially docker containers
- Should we seriously start looking what is docker and related topics?
  - Our answer is yes for sure
- We also look at the fundamental reason docker is getting all the traction

SOMETHING ABOUT ME

- Docker n00b
- Director of Appsecco
  - Appsecco is a specialist application security company
- Author of ‘Burp Suite Essentials’
  - Burp Suite is the most popular software for security testing applications
- Community Champion
  - OWASP Bangalore Chapter Leader
  - n|u Co-Founder and Community Manager
- Co-Trainer of Xtreme Web Hacking Class
- Links
  - @makash | https://linkd.in/webappsecguy | akashm.com

> There is space for only 5 types of security approaches in this world
>
> *-Said no one ever*

Its always nice to start with a quote. So here is one.

HOW WE DO APPSEC CURRENTLY?

A bit of modern methods, process and approaches securing legacy stuff

AUTOMATED WEB APPLICATION SCANNERS

To a hammer everything looks like a nail approach

BUG BOUNTY BEGINNERS – WIN SOME – LOSE SOME

Be honest, we have all been here

BUG BOUNTY/PENTESTERS & EXPERTS MAKE IT LOOK SIMPLE

This may not be the case with all of us here.

SECURITY TESTERS PLOD AWAY USING CHECKLISTS & TOOLS

Test Name	Ref. Number	Status	Risk
Spiders, Robots and Crawlers	IG-001	Not Done	
Search Engine Discovery/Reconnaissance	IG-002	Done	L
Identify application entry points	IG-003	Done	H
Testing for Web Application Fingerprint	IG-004	Done	M
Application Discovery	IG-005	Done	H
Analysis of Error Codes	IG-006	Not Done	
SSL/TLS Testing (SSL Version, Algorithms, Key length, Digital Cert. Validity) - SSL Weakness	CM-001	Done	H
DB Listener Testing - DB Listener weak	CM-002	Not Done	
Infrastructure Configuration Management Testing - Infrastructure Configuration management weakness	CM-003	Done	H
Application Configuration Management Testing - Application Configuration management weakness	CM-004	Not Done	
Testing for File Extensions Handling - File extensions handling	CM-005	Not Done	
Old, backup and unreferenced files - Old, backup and unreferenced files	CM-006	Done	M
Infrastructure and Application Admin Interfaces - Access to Admin interfaces	CM-007	Not Done	
Testing for HTTP Methods and XST - HTTP Methods enabled, XST permitted, HTTP Verb	CM-008	Not Done	
Credentials transport over an encrypted channel - Credentials transport over an encrypted channel	AT-001	Not Done	

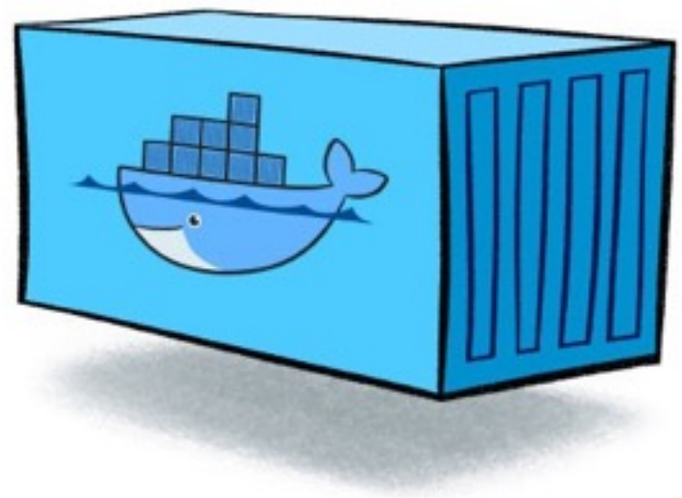
PORTSWIGGER
WEB SECURITY

OWASP ZAP

4.5		**Authentication Testing**
4.5.1	OTG-AUTHN-001	Testing for Credentials Transported over an Encrypted C
4.5.2	OTG-AUTHN-002	Testing for default credentials
4.5.3	OTG-AUTHN-003	Testing for Weak lock out mechanism
4.5.4	OTG-AUTHN-004	Testing for bypassing authentication schema
4.5.5	OTG-AUTHN-005	Test remember password functionality
4.5.6	OTG-AUTHN-006	Testing for Browser cache weakness
4.5.7	OTG-AUTHN-007	Testing for Weak password policy
4.5.8	OTG-AUTHN-008	Testing for Weak security question/answer
4.5.9	OTG-AUTHN-009	Testing for weak password change or reset functionaliti
4.5.10	OTG-AUTHN-010	Testing for Weaker authentication in alternative channel

4.3		**Configuration and Deploy Management Testing**
4.3.1	OTG-CONFIG-001	Test Network/Infrastructure Configuration
4.3.2	OTG-CONFIG-002	Test Application Platform Configuration
4.3.3	OTG-CONFIG-003	Test File Extensions Handling for Sensitive Information
4.3.4	OTG-CONFIG-004	Backup and Unreferenced Files for Sensitive Information
4.3.5	OTG-CONFIG-005	Enumerate Infrastructure and Application Admin Interfaces
4.3.6	OTG-CONFIG-006	Test HTTP Methods
4.3.7	OTG-CONFIG-007	Test HTTP Strict Transport Security
4.3.8	OTG-CONFIG-008	Test RIA cross domain policy

If you are thinking this slide is doing too much, yes you are absolutely right. Most security testers who do application security testing end up doing a bunch of manual and semi-automated tasks using checklists as references and also to convey what was covered etc.



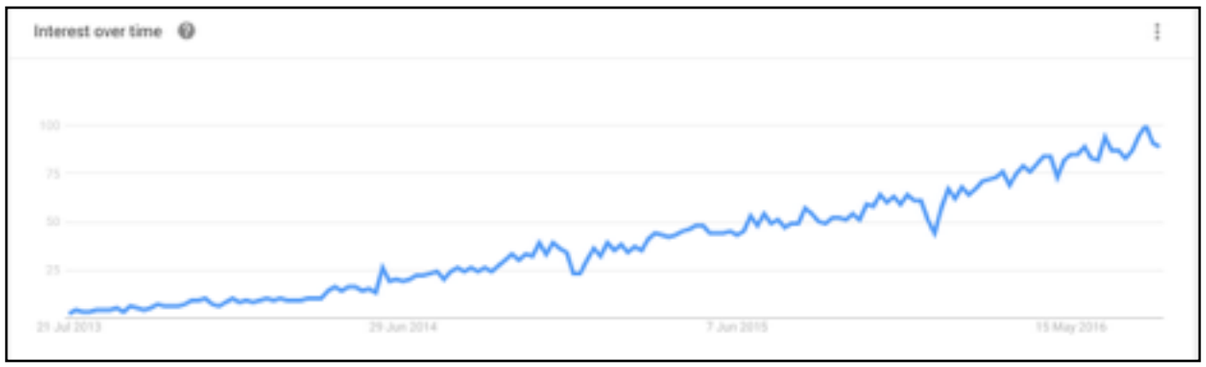
WHAT IS A DOCKER CONTAINER?

A DOCKER CONTAINER?

- A container allows a developer to package up and application and all of its dependent parts in a box
- This box is basically an isolated environment and the application has everything it needs to run inside of this environment

CONTAINERS ARE COMING

DOCKER IN GOOGLE TRENDS SINCE JUL 2013–PRESENT



A value of 100 is the peak popularity for a term

IF THE DRY GRAPH WASN'T ENOUGH TO CONVINCE YOU

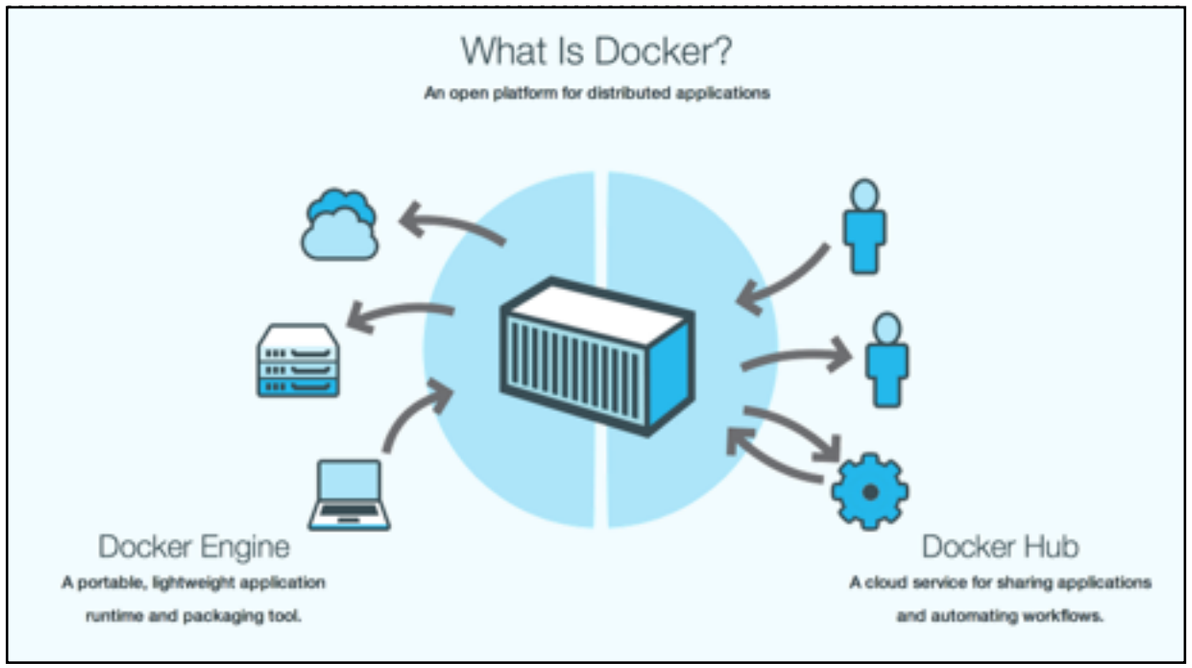
> Why has this change to docker become imminent?
>
> *-Me, when I started noticing how quickly the developer world was moving to docker*

Not using the word inevitable but imminent but is going to happen faster than most people can say OWASP Top 10 2017

Don’t worry, I am not going to jump up and down now.
How many of you get the reference?

> Regardless of how much security folks think their opinion matters, most of the developers don't give a fish about what we think
>
> *- Akash Mahajan, learning the truth the hard way*

THIS IS WHAT DEVELOPERS WANT – AN IT FREE WORLD



Attribution pending. Will update the slides once I know where I nicked this from!

http://www.infoq.com/cn/articles/docker-core-technology-preview

BUT ISN'T THIS JUST LIKE CHROOT?

```
# Mount Kernel Virtual File Systems
TARGETDIR="/mnt/chroot"
mount -t proc proc $TARGETDIR/proc
mount -t sysfs sysfs $TARGETDIR/sys
mount -t devtmpfs devtmpfs $TARGETDIR/dev
mount -t tmpfs tmpfs $TARGETDIR/dev/shm
mount -t devpts devpts $TARGETDIR/dev/pts

# Copy /etc/hosts
/bin/cp -f /etc/hosts $TARGETDIR/etc/

# Copy /etc/resolv.conf
/bin/cp -f /etc/resolv.conf $TARGETDIR/etc/resolv.conf

# Link /etc/mtab
chroot $TARGETDIR rm /etc/mtab 2> /dev/null
chroot $TARGETDIR ln -s /proc/mounts /etc/mtab
```

INSTALLING MUTILLIDAE (PHP+APACHE+MYSQL APP)

Pull image:

```
docker pull citizenstig/nowasp
```

Start with random mysql password:

```
docker run -d -p 80:80 citizenstig/nowasp
```

Or specify it as environment variable:

```
sudo docker run -d -p 80:80 -p 3306:3306 -e MYSQL_PASS="Chang3ME!"
citizenstig/nowasp
```

So a non-dev can use something like Kitematic and do this using a GUI

> If a developer has to choose between being productive or being secure, more or less she/he will chose being productive
>
> *- Something I should have said!*

WHAT CAN WE DO NOW TO GET ON THE BANDWAGON?

- For testing applications
  - We usually need the setup running somewhere (testing)
  - Being able to get the complete setup by just running a simple command, makes all of us “productive”
- For secure development
  - Pre-configured dockerfiles with selective containers which allow for secure configuration by default
- For secure operations
  - Running docker in secured, isolated instances

Depends on how you do application security

DOES DOCKER PROVIDE ISOLATION FROM THE HOST?

- Follow the CIS Docker Benchmark to get a checklist of things to do on
  - Host Configuration (15 list items)
  - Docker Daemon Configuration (13 list items)
  - Files, Permissions and configuration files for Docker Daemon (20 list items)
  - Container Images (5 list items)
  - Container Runtime (25 list items)
  - Follow Docker Security Operations Best Practices

Yes if you practice defence in depth

https://benchmarks.cisecurity.org/tools2/docker/CIS_Docker_1.11.0_Benchmark_v1.0.0.pdf
Play https://contained.af/
A good place to start is https://github.com/docker/docker-bench-security
And read this https://blog.docker.com/2015/05/understanding-docker-security-and-best-practices/

DOCKER HOST AND CONTAINER SECURITY GETTING STARTED

- [ ] Start by reading Understanding docker security and best practices https://blog.docker.com/2015/05/understanding-docker-security-and-best-practices/
- [ ] Use the Docker Bench Security script to automatically check best practices as outlined by the CIS Docker Benchmark version 1.11 https://github.com/docker/docker-bench-security
- [ ] Play this awesome game to break out of docker containers in your browser https://contained.af/
- [ ] Read the full CIS Docker 1.11.0 Benchmark report https://benchmarks.cisecurity.org/tools2/docker/CIS_Docker_1.11.0_Benchmark_v1.0.0.pdf
- [ ] Definitely read if you plan to run docker in prod or are guiding someone who does http://container-solutions.com/is-docker-safe-for-production/

https://benchmarks.cisecurity.org/tools2/docker/CIS_Docker_1.11.0_Benchmark_v1.0.0.pdf
Play https://contained.af/
A good place to start is https://github.com/docker/docker-bench-security
☑ And read this

NEW TOOLS, APPROACHES AND OPPORTUNITIES

- $$$ The configuration for docker containers will need to be tested in a continuous manner
- $$$ Auditing of existing deployments against security benchmarks like the CIS Docker Benchmark
- $$$ Following agile practices, dockers build built using CI/CD tools like Jenkins based on pre-commit and post-commit hooks
- $$$ Bring in your SAST, DAST, *ST analysis at any point in this pipeline
- $$$ Setting up and managing private registries
- $$$ Also setting up private repositories for nom etc.

All of the above can bring in the moolah and ensure your clients or your apps stay secure
Just a list from how I think these things will evolve, obviously all you talented folks will figure out even more options.

TO START WITH, THIS IS WHAT YOU SHOULD DO

- ☑ Test the application as you normally would
  - ☑ If you find appsec issues report these
- ☑ Do white box assessment with the docker security checklists
  - ☑ You already have a roadmap as mentioned in slide 21 & 22
- ☑ Keep track of any privilege escalation bugs in docker daemon or the underlying hypervisor/VM tech you are using
- ☑ Understand what is the software supply chain for the application & pick secure alternatives for orchestration itself
- ☑ Application containers make it simple for everyone so use them for training, best practices etc.

DOCKER FAILS

Couple of #devoops moments

No discussion is complete without talking about what didn’t work or what was FUBARed

TWITTER'S VINE SOURCE CODE DUMP BY @AVICODER

- @avicoder a bug bounty hunter, he spoke about this bug at a null/ OWASP/G4H Bangalore meet in June 2016
- He found an interesting sub domain for Vine (A twitter video app)
- He had stumbled upon a private docker registry being used
- He realised that the version being used didn't use any authentication and by querying the API he determined the docker files being hosted
- He did a docker pull of an image that contained the source code for the Vine App and got $$$$$ bounty
- https://avicoder.me/2016/07/22/Twitter-Vine-Source-code-dump/

DOCKER IMAGE INSECURITY

- This has been fixed now! Especially from docker version 1.10
- Earlier if an image had been compressed with xz (in C so not safety for memory)
- Docker Daemon would exec the xz binary as root user
- If there was a single vulnerability in xz, a docker pull could result in complete compromise
- Read more about the vulnerability https://titanous.com/posts/docker-insecurity
- Read more about how this was fixed https://titanous.com/posts/docker-insecurity

QUESTIONS

@makash | https://linkd.in/webappsecguy | akashm.com

APPSECCO
THE APPLICATION SECURITY COMPANY