Confessions of a lactose intolerant

late night coding, why cheese hates me and vulnerabilities!
Things to talk about ...

- Shell Scripts
- Docker
- SQL
- API
- Ruby
- Scan Tools
Once upon a time...
The charge of developers

Awareness

Educate

Ownership
The tool
fon-diddly-do

App Runner
Ruby on Rails API
PostgreSQL
The Cloner
Clone your targets

- Github API
- curl
- wget
- ... others
The Gatherer
Resolve Application Services

- Language
- Framework
- Database
- Caching
- Background Processing
- Versions
FROM ruby: <<RUBY_VERSION>>

RUN apt-get update -yqq \
    && apt-get install -yqq --no-install-recommends \
    build-essential \ 
    libqp-dev \ 
    Nodejs

WORKDIR /demo_app
COPY Gemfile* ./
COPY . .

CMD bundle exec unicorn -p 8081 -c .config/.unicorn.rb
version: “2”

volumes:

database-data:
  external: false

services:

cache:
  image: $<CACHE_IMAGE>

redis:
  image: $<REDIS_IMAGE>

database:
  image: $<DATABASE_IMAGE>

volumes:
  - database-data:/demo_data

env_file: .demo_env

build: .

volumes:
  - .:/demo_app

ports:
  - “8081:8081”

depends_on:
  - database
  - redis
  - cache
The Gatherer
Copy generic configs

1  default: &default
2    adaptor: <<DB_ADAPTOR>>
3    host: <%= ENV["DB_HOST"]%>
4    username: <%= ENV["DB_USER"]%>
5    password: <%= ENV["DB_PWD"]%>
6    encoding: utf8
7    min_message: warning
8    tool: 2
9    timeout: 5000
10
11 development:
12    <<: *default
13    database: <<DB_NAME>>

1  DB_HOST=localhost
2  DB_PORT=5432
3  DB_USER=db_user
4  DB_PASSWORD=db_pwd
5  REDIS_USER=rd_user
6  REDIS_PASSWORD=rd_pwd
7  SECRET_KEY_BASE=dev_secret_key
8  etc ...
The Gatherer

Build it up, scan it, tear it down

docker-compose up -d ...

scan-target (with opts ...)

docker-compose down

rm -rf ~/repos/demo_app
The Scanner
Burp

Automated

Driven from the command line

Run scanner with arguments

Generate well formed reports
class BurpExtender
    Include IBurpExtender, IHttpListner, IScannerListenner

def registerExtenderCallbacks(callbacks)
    # implement IBurpExtender
    @callbacks = callbacks
    end

def cmd_line_args
    params = @callbacks.getCommandLineArguments
    @target = params[0]
    @format = params[1]
    @name = params[2]
    end

def processHttpMessage(flag, request, message)
    # implement IHttpListner
    end

def newScanIssue(issue)
    # implement IScannerListener
    end
java -Xmx4g -Djava.awt.headless=true -jar ~/lib/burpsuite.jar $1 $2 $3

# add other script tasks ...
The Scanner
Scripting Burp

scan-target http://demo_app:8081 xml /tmp/demo_app.xml
The Importer
Rake tasks for data import

bundle exec rake "import:burp[demo_app.xml, demo_app, rails, 4.2.8, https://demo, demo@demo]"

bundle exec rake "import:brakeman[demo_app.xml, demo_app, rails, 4.2.8, https://demo, demo@demo]"
The Analysis
Organise, prioritize and fix
The Analysis
Organise, prioritize and fix
Further thoughts
Future development

- Improve gathering process
- Gatherers for different frameworks
- Interface for different scan-tools
- Javascript reporting server
- Scanning git history
- Vulnerability mitigation
The End

https://github.com/redshieldsecurityltd

Questions & Suggestions

appreggios.from.the.kitchen@gmail.com