OWASP
Open Web Application
Security Project

Using JIRA to manage Risks and Security Champions activities

OWASP AppSecEU, Rome, 2016

Me

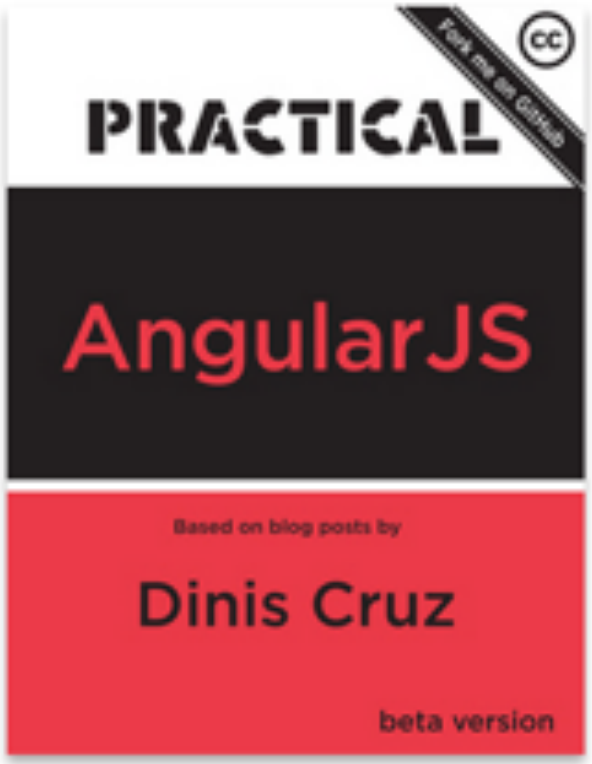
- Developer for 25 years
- AppSec for 13 years
- Day jobs:
  - Leader OWASP O2 Platform project
  - Application Security Training for JBI Training
  - Part of AppSec team of:
    - The Hut Group
    - BBC
- AppSec Consultant and Mentor

Books Published

- @Leanpub (buy for 0$)
- http://leanpub.com/u/DinisCruz

Books under development

Ideas shown in this presentation
and a lot more



Major revision with lots of new content
(based on Maturity Models app)

Twitter, Inc. [US] https://twitter.com/DinisCruz

Home Notifications Messages Search Twitter

TWEETS 11.9K | FOLLOWING 264 | FOLLOWERS 2,731 | LIKES 1,375

Edit profile

Dinis Cruz
@DinisCruz

Tweets | Tweets & replies | Media

Pinned Tweet

blog.diniscruz.com

Dinis Cruz Blog

A personal blog about: transforming Web Application Security into an 'Application Visibility' engine, the OWASP O2 Platform, Application/Data interoperability and a lot more

Home | Hire me | AppSec Jobs and Projects | OWASP O2 Platform | Real-Time Vulnerability Feedback in VisualStudio | About

See also:

http://blog.diniscruz.com/2016/03/new-era-of-software-with-modern.html

NEW ERA OF SOFTWARE WITH MODERN APPLICATION SECURITY

My thesis is that

Application Security can be used to define and measure Software Quality

THE POLLUTION ANALOGY

These tools/techniques are designed to

A) Improve code Quality

B) Make AppSec possible

MODERN APPLICATION SECURITY

- TDD with Code Coverage
- Threat Models
- Docker and Containers
- Test Automation
- SAST/DAST/IAST/WAF
- Clever Fuzzing
- JIRA Risk workflows
- Kanban for Quality fixes
- Web Services visualisation
- ELK

TECHNICAL DEBT IS A BAD ANALOGY

- The developers are the ones who pays the debt
- Pollution is a much better analogy
- The key is to make the business accept the risk (i.e the debt)
  - Which is done using the JIRA RISK Workflows

APPSEC AND DEVELOPERS

Disclamers

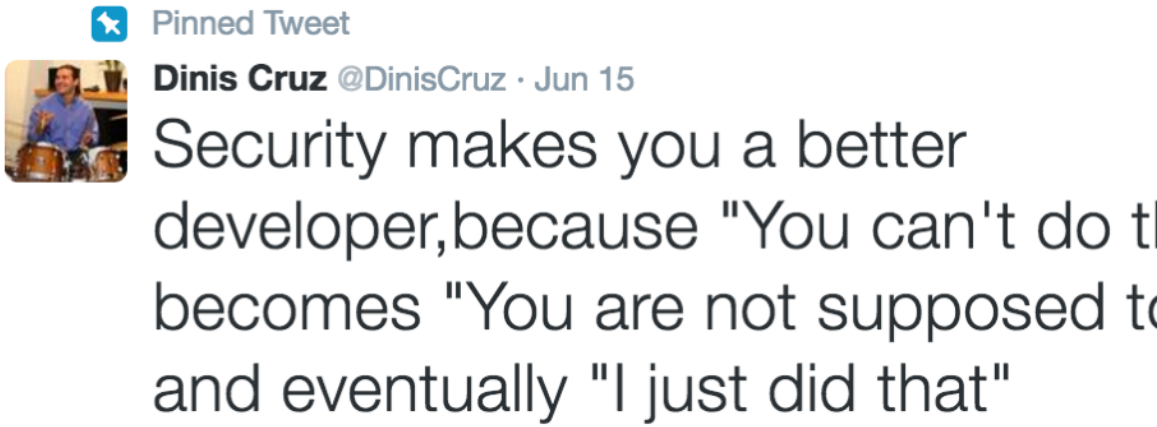
- **(unit) Test** - For me a test is anything that can be executed with one of these Unit Test Frameworks: https://en.wikipedia.org/wiki/List_of_unit_testing_frameworks
- **RISK** - Abuse the concept, found RISK to be best one for the wide range of issues covered by AppSec, while being understood by all players
- **100% Code Coverage** - not the summit, but base-camp (i.e. not the destination). And 100% code is not enough, we really need 500% or more Code Coverage)
- **AppSec ~= Non Functional requirements** - AppSec is about understanding and controlling app's unintended behaviours

AppSec vs InfoSec

- This presentations is about AppSec
- AppSec is about:
  - code, apps, CI, secure coding standards, threat models, frameworks, code dependencies, QA, testing, fuzzing, dev environments, DevOps, ….
- InfoSec is about:
  - Networks, Firewalls, Server security, Anti-virus, IDS, Logging, NOC, Policies, end-user security, mobile devices, AD/Ldap management, user provisioning, DevOps, ….
- If your 'InfoSec' team/person cannot code (and would not be hired by the Dev team), then that is NOT AppSec.
- InfoSec is also very important (workflow described here can also be used by them)

Developers we need you to join AppSec

- You will become a better developer



- You will be paid better

MATURITY MODELS APP

Maturity Models

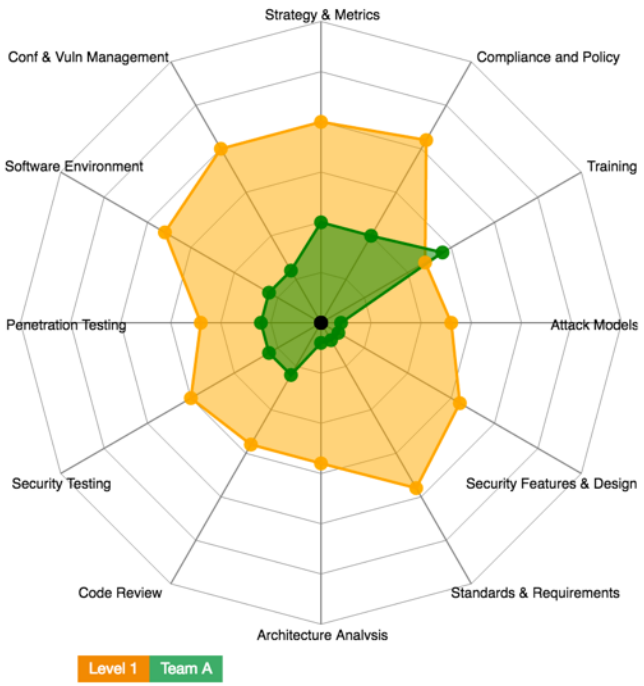
- App used on the JIRA tickets examples
- Open Source (https://github.com/DinisCruz/Maturity-Models)
- Based on real world mapping of BSIMM on large organisation
- Starting to be compatible with OWASP OpenSAMM (help needed)
- Coded in NodeJS and AngularJS (v1) with 90%+ code coverage and full automated CI

Visualise Maturity Models

localhost:3000/view/bsimm/team-A

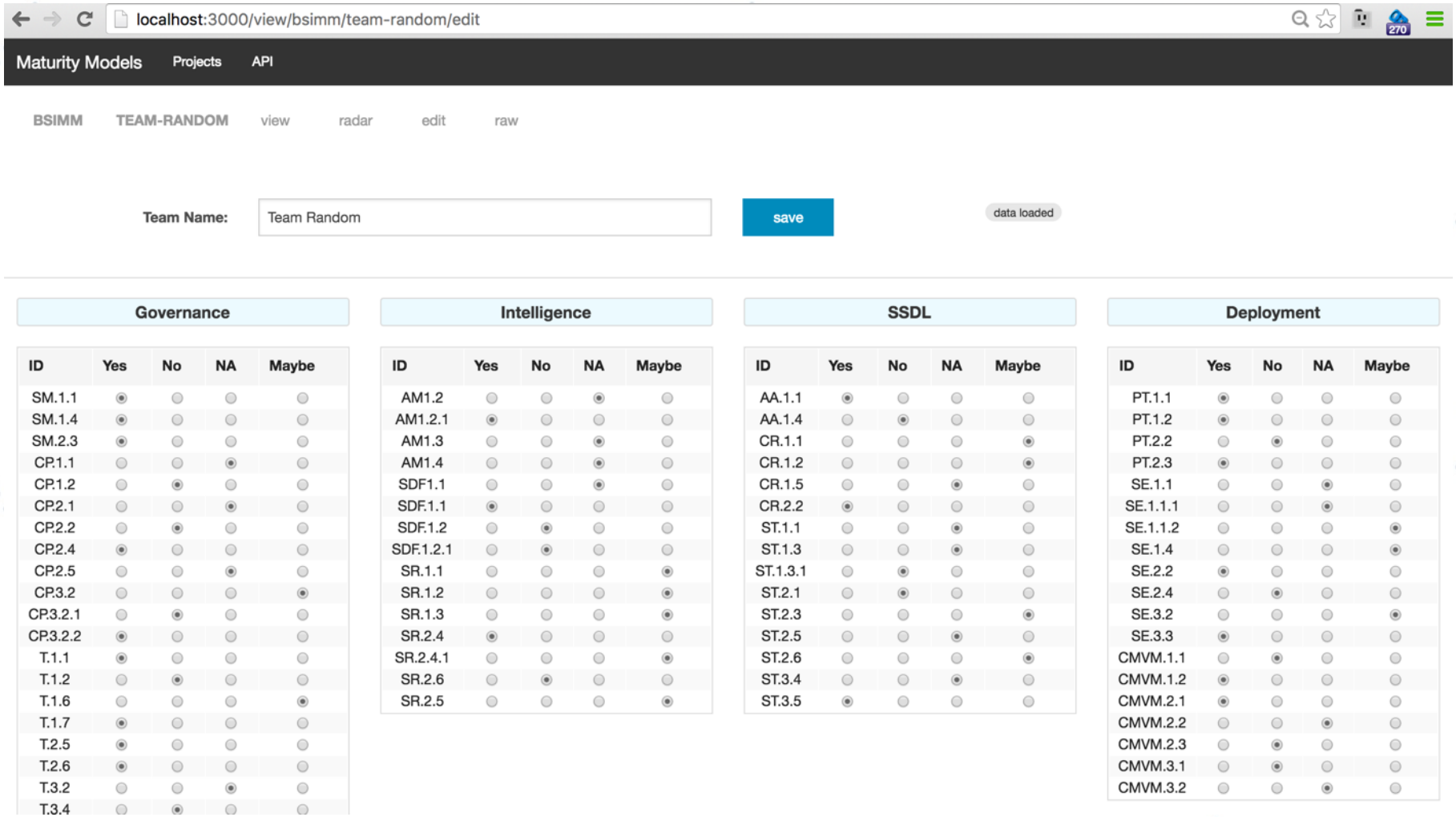
Maturity Models | Projects | API

BSIMM | TEAM-A | view | radar | edit | raw

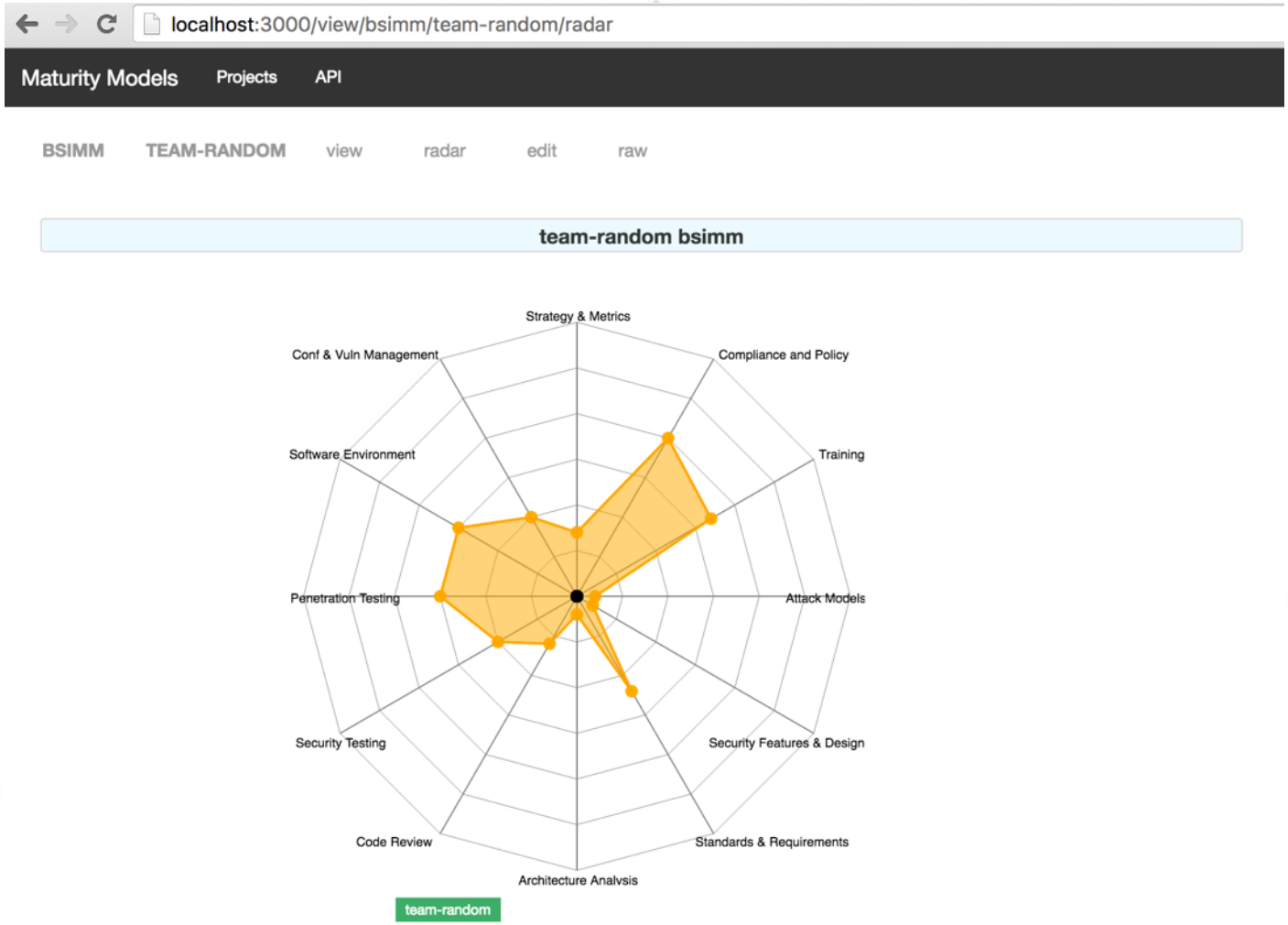


Governance		Intelligence		SSDL		Deployment	
Activity	Status	Activity	Status	Activity	Status	Activity	Status
SM.1.1	Yes	AM1.2	Maybe	AA.1.1	Maybe	PT.1.1	Maybe
SM.1.4	No	AM1.2.1	Maybe	AA.1.4	Maybe	PT.1.2	Maybe
SM.2.3	Yes	AM1.3	Maybe	CR.1.1	Maybe	PT.2.2	Maybe
CP.1.1	Maybe	AM1.4	Maybe	CR.1.2	Maybe	PT.2.3	Maybe
CP.1.2	Yes	SDF1.1	Maybe	CR.1.5	Maybe	SE.1.1	Maybe
CP.2.1	Yes	SDF.1.1	Maybe	CR.2.2	Yes	SE.1.1.1	Maybe
CP.2.2	Maybe	SDF.1.2	Maybe	ST.1.1	Maybe	SE.1.1.2	Maybe
CP.2.4	Maybe	SDF.1.2.1	Maybe	ST.1.3	Maybe	SE.1.4	Maybe
CP.2.5	Maybe	SR.1.1	Maybe	ST.1.3.1	Maybe	SE.2.2	Maybe
CP.3.2	No	SR.1.2	Maybe	ST.2.1	No	SE.2.4	Maybe
CP.3.2.1	Maybe	SR.1.3	Maybe	ST.2.3	Maybe	SE.3.2	Maybe
CP.3.2.2	Maybe	SR.2.4	No	ST.2.5	Maybe	SE.3.3	Yes
T.1.1	Maybe	SR.2.4.1	No	ST.2.6	Maybe	CMVM.1.1	Maybe
T.1.2	No	SR.2.6	Maybe	ST.3.4	Yes	CMVM.1.2	Yes
T.1.6	Yes	SR.2.5	Maybe	ST.3.5	Maybe	CMVM.2.1	Maybe
T.1.7	No	CMVM.2.2	Maybe				
T.2.5	Yes	CMVM.2.3	Maybe				

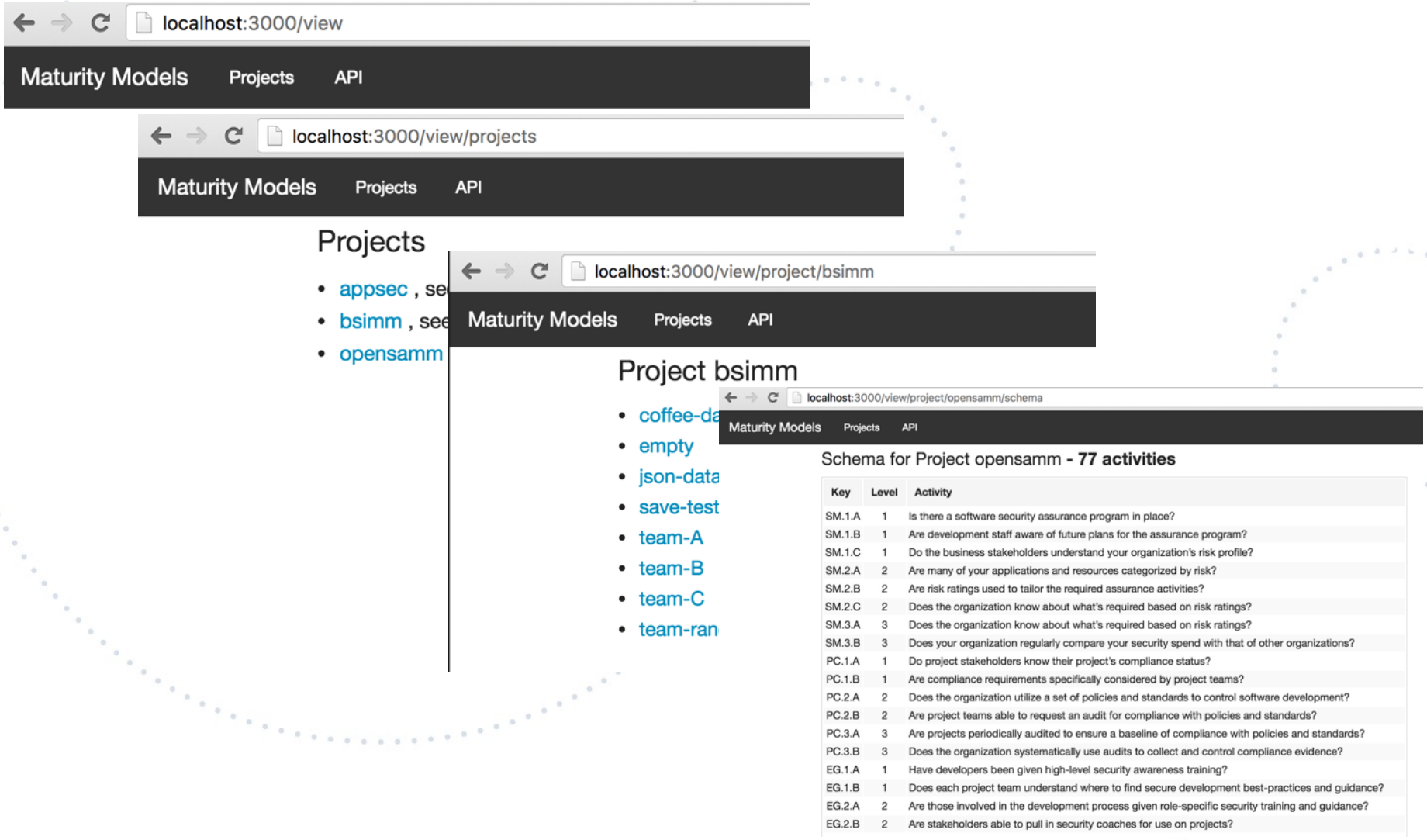
Edit Maturity Model



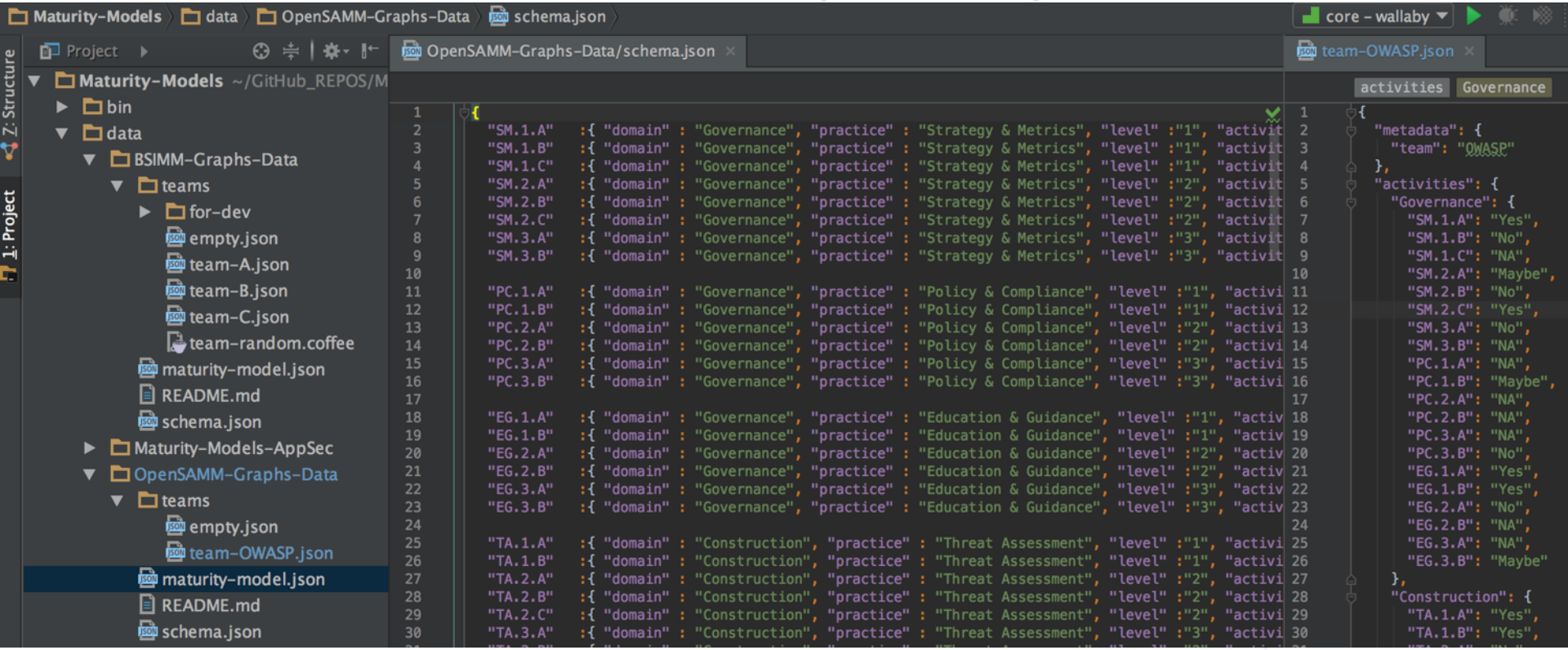
View Maturity Model Radar chart



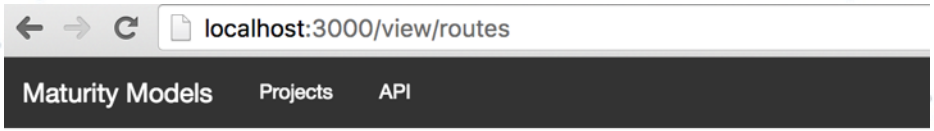
View projects and schema



All data stored in JSON (git repo)



Mapped Attack Surface



API methods

/
/api/v1/data/bsimm/coffee-data/radar
/api/v1/data/bsimm/empty/radar
/api/v1/data/bsimm/json-data/radar
/api/v1/data/bsimm/save-test/radar
/api/v1/data/bsimm/team-A/radar
/api/v1/data/bsimm/team-B/radar
/api/v1/data/bsimm/team-C/radar
/api/v1/data/bsimm/team-random/radar
/api/v1/logs/file/:index
/api/v1/logs/list
/api/v1/logs/path
/api/v1/project/get/:project
/api/v1/project/list
/api/v1/project/schema/:project
/api/v1/routes/list
/api/v1/routes/list-raw
/api/v1/table/bsimm/coffee-data
/api/v1/table/bsimm/empty
/api/v1/table/bsimm/json-data
/api/v1/table/bsimm/save-test
/api/v1/table/bsimm/team-A
/api/v1/table/bsimm/team-B
/api/v1/table/bsimm/team-C
/api/v1/table/bsimm/team-random
/api/v1/team/:project/list
/api/v1/team/bsimm/get/coffee-data
/api/v1/team/bsimm/get/empty
/api/v1/team/bsimm/get/json-data
/api/v1/team/bsimm/get/save-test
/api/v1/team/bsimm/get/team-A
/api/v1/team/bsimm/get/team-B
/api/v1/team/bsimm/get/team-C
/api/v1/team/bsimm/get/team-random
/api/v1/team/bsimm/save/coffee-data
/api/v1/team/bsimm/save/empty
/api/v1/team/bsimm/save/json-data
/api/v1/team/bsimm/save/save-test
/api/v1/team/bsimm/save/team-A
/api/v1/team/bsimm/save/team-B
/api/v1/team/bsimm/save/team-C
/api/v1/team/bsimm/save/team-random
/ping
/view*

Continuous Integration (CI)

1. Dev pushes code to GitHub
2. Github (main code repo)
   - sends web hook to Travis
3. Travis
   - clones repo, runs tests (API and UI)
   - builds Docker Image (if all tests pass)
   - push Docker Image to Docker Hub
   - clones QA repo fork, sync with QA repo, adds extra commit to QA repo fork, pushes to QA repo Fork
4. Docker Hub
   - sends web hook to Docker Cloud
5. Docker Cloud
   - contacts mapped Node (Digital Ocean VM with Docker installer)
   - docker host pulls image from Docker cloud
   - docker container starts
6. Github (QA fork repo)
   - sends web hook to Travis
7. Travis
   - clones repo, runs tests (QA against deployed docker image on Digital ocean)
   - (in the future) will send web hook to deploy to production (if all tests pass)

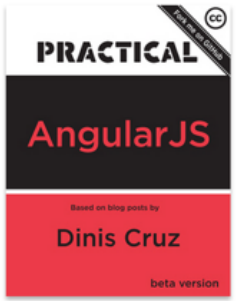
Technologies used (40x)

This section contains all the different technologies that I used in the development of the Maturity-Models project

In alphabetical order

- [ ] Agile-using-Kandan-WIP
- [ ] Bower
- [ ] CoffeeScript
- [ ] Css
- [ ] D3
- [ ] Digital Ocean
- [ ] Docker
  - [ ] Docker Hub
  - [ ] Docker Cloud
- [ ] Electrium
- [ ] Foundation
- [ ] Git
  - [ ] Git Branches
  - [ ] Git Commits
  - [ ] Git SourceTree
  - [ ] Git Submodules
  - [X] Git Tags
  - [ ] Git Rebase
- [ ] GitHub
- [ ] Gulp
- [ ] Html
- [ ] Javascript
- [ ] Jira
- [ ] JQuery
- [ ] Node
  - [ ] Node Express
  - [ ] Node Fluentnode
  - [ ] Node JsDom
  - [ ] Node Karma
  - [ ] Node Mocha
  - [ ] Node Morgan
  - [ ] Node Npm
  - [ ] Node PhantomJS
  - [ ] Node modules
  - [ ] Node Supertest
- [ ] Open Source
- [ ] Pug
- [ ] Travis
- [ ] WallabyJs
- [ ] WebStorm

see book for details on each of these technologies



SECURITY CHAMPIONS

Security Champions (SC)

What are Security Champions and what do they do?

Security Champions are a key element of an AppSec team, since they create an cross-functional team focused on Application Security

Here is an good definition for you to customise to your culture and workflows:

What is an Security Champion?

- Security Champions are active members of a team that may help to make decisions about when to engage the Security Team
- Act as the "voice" of security for the given product or team
- Assist in the triage of security bugs for their team or area

What do they do?

- Actively participate in the AppSec JIRA and WIKI
- Collaborate with other security champions
    - Review impact of 'breaking changes' made in other projects
- Attend weekly meetings
- Are the single point of contact for their assigned team
- Ensure that security is not a blocker on active development or reviews
- Assist in making security decisions for their team
    - Low-Moderate security impact
        - Empowered to make decisions
        - Document decisions made in bugs or wiki
    - High-Critical security impact
        - Work with AppSec team on mitigations strategies
- Help with QA and Testing
    - Write Tests (from Unit Tests to Integration tests)
    - Help with development of CI (Continuous Integration) environments

http://blog.diniscruz.com/2015/10/what-are-security-champions-and-what-do.html

If you don't have an SC, get a Mug

Basically that 'Security Expert' Mug should represent the fact that at the moment when a developer has an Application Security question, he might as well ask the dude on that Mug for help :)

I also like that it re-enforces the idea, that for most developer teams, **just having somebody assigned to application security,** is already a massive step forward!!

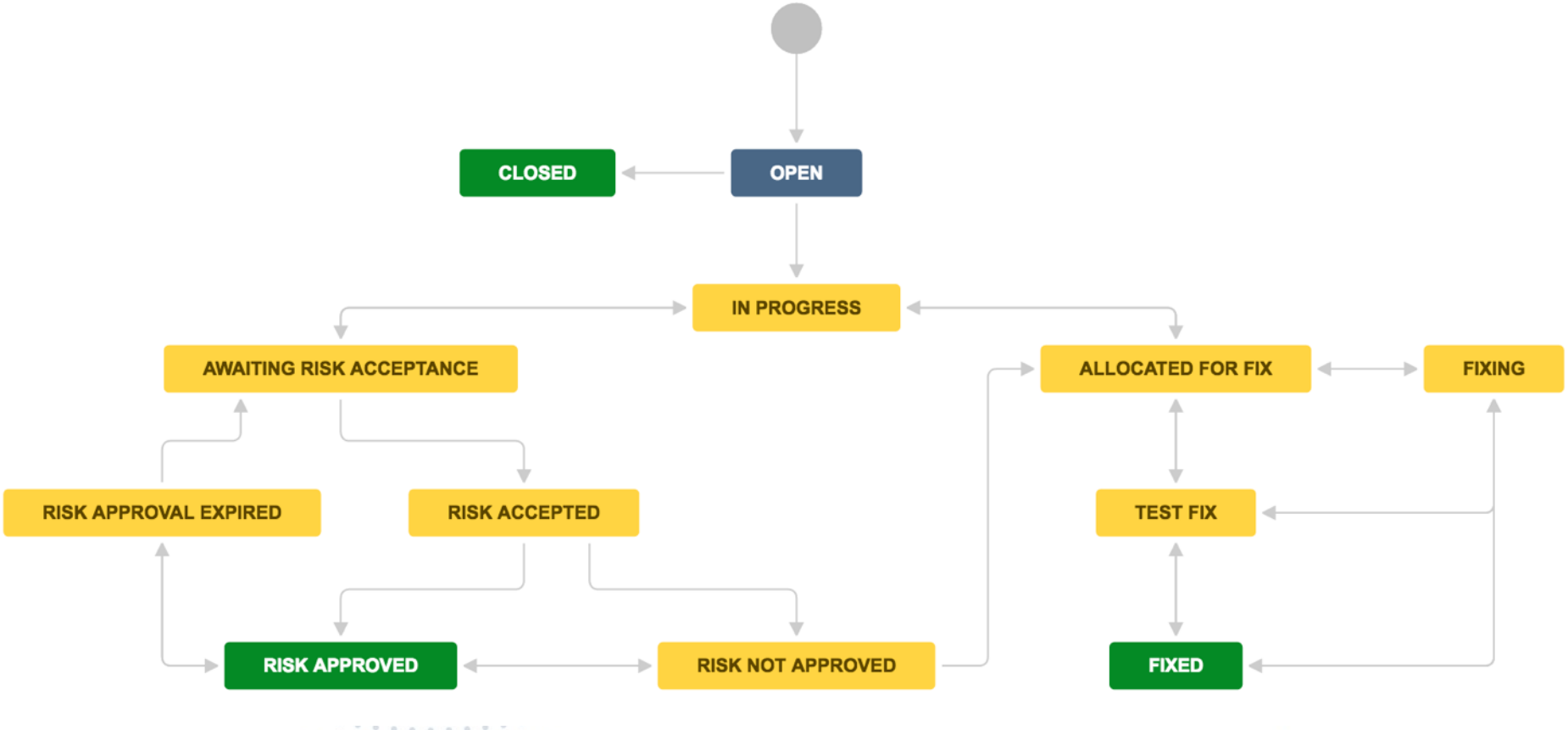
Basically we have such a skill shortage in our industry for application security devs that ***'if you have a heart-beat you qualify'***

JIRA WORKFLOW

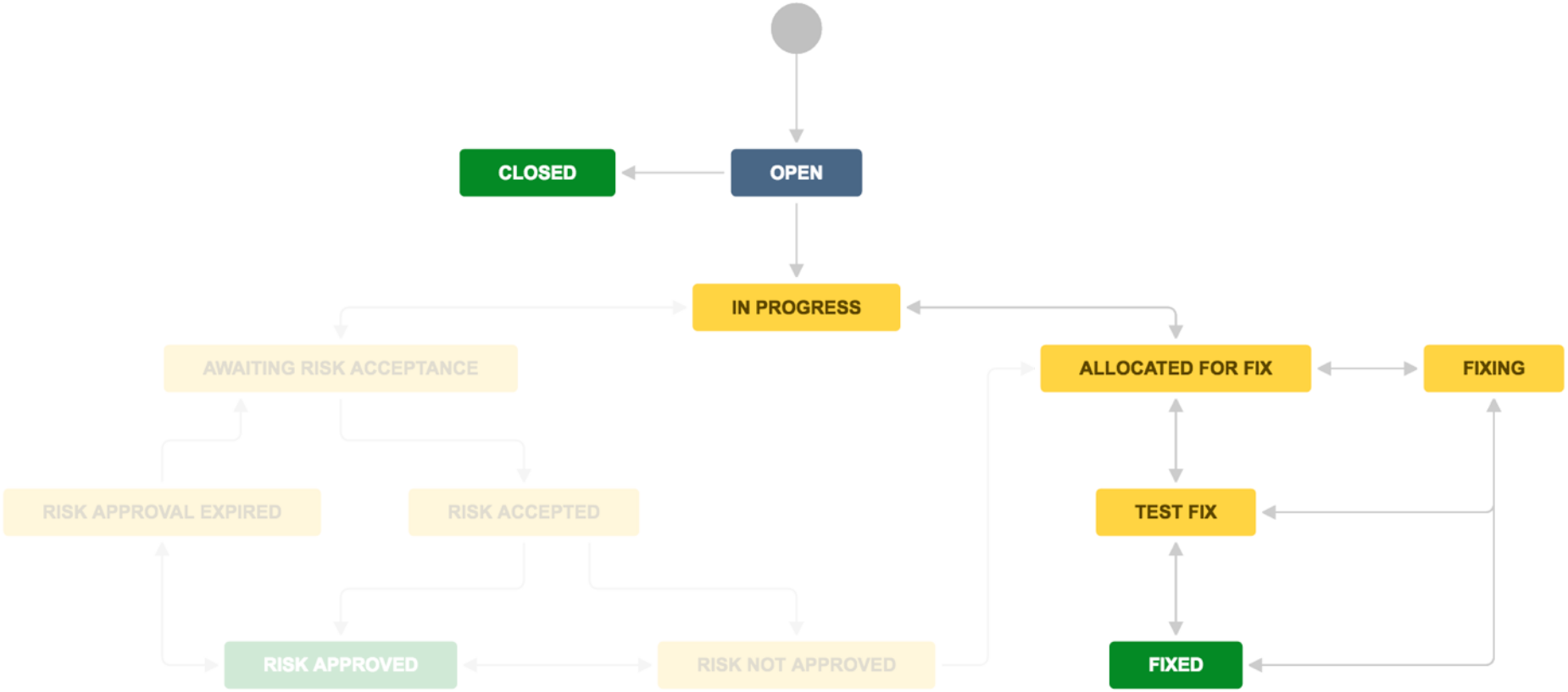
Proposed JIRA workflow

1. Open JIRA issues for all AppSec issues
2. Write passing tests for issues reported
3. Manage using AppSec RISK workflow
    1. **Fix Path:** Open, Allocated for Fix, Fix, Test Fix, Close
    2. **Accept Risk Path:** Open, Accept Risk, Approve Risk, (Expire Risk)
4. Automatically report RISK's status

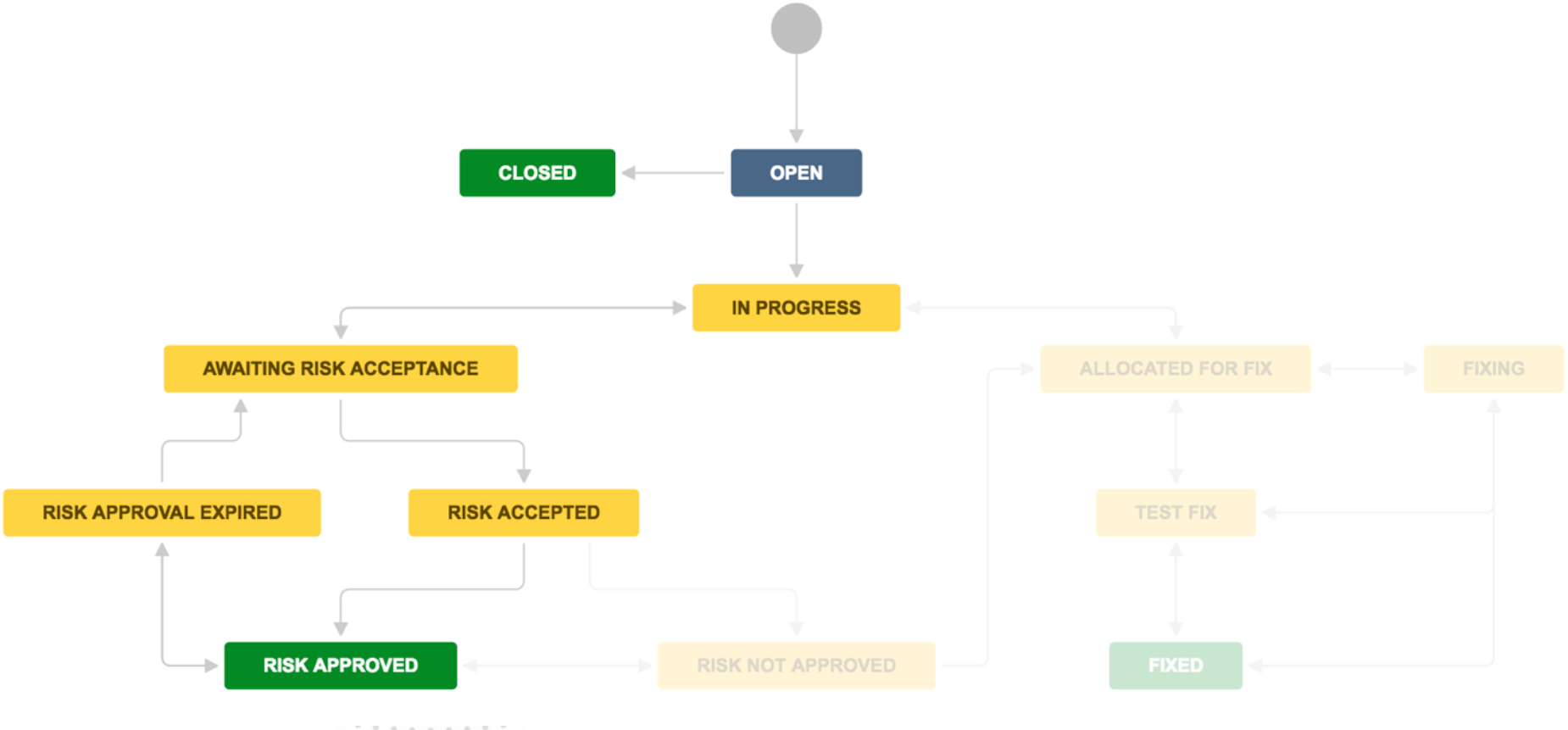
RISK Workflow (using JIRA in Cloud)



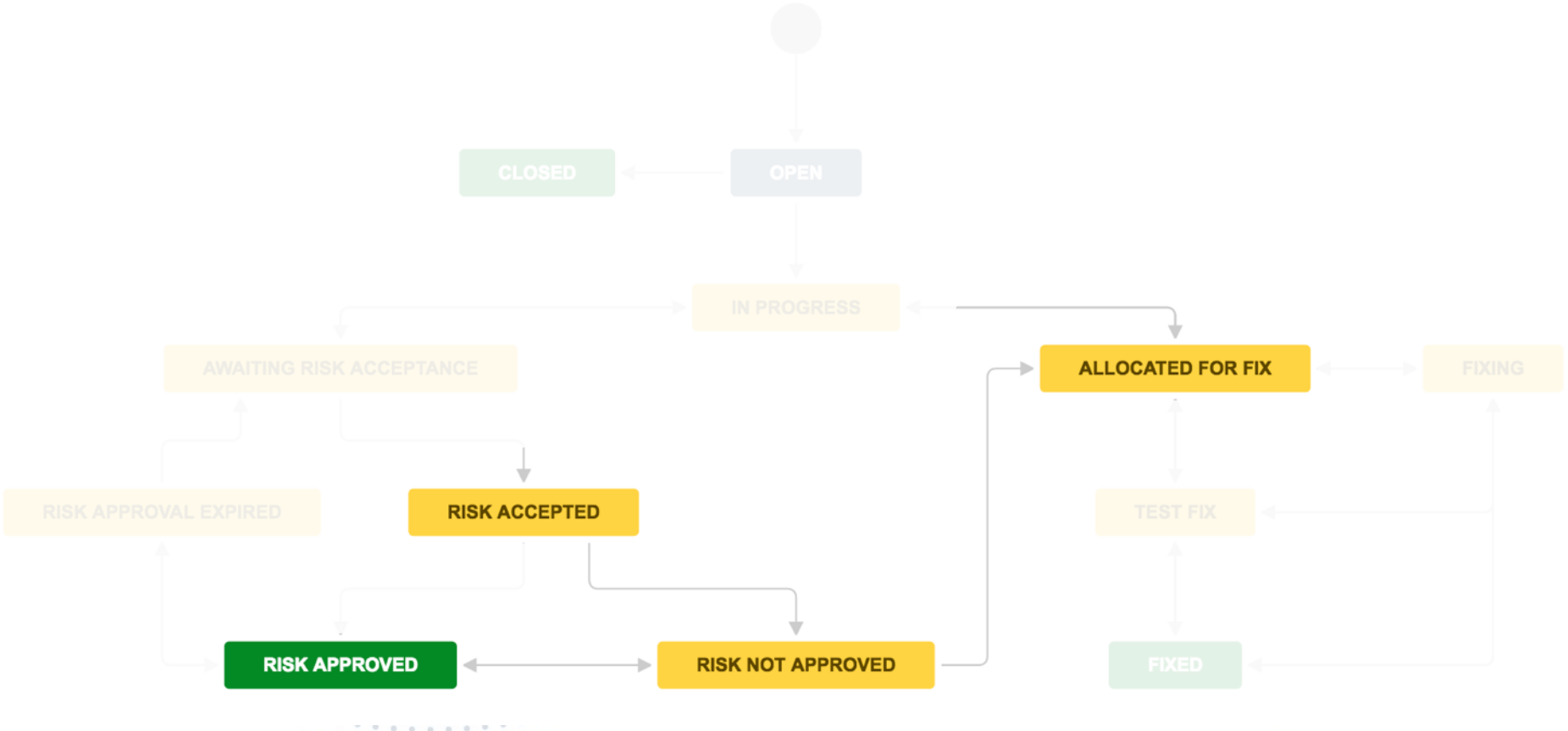
PATH #1 - Fix issue



PATH #2 - Accept and Approve RISK

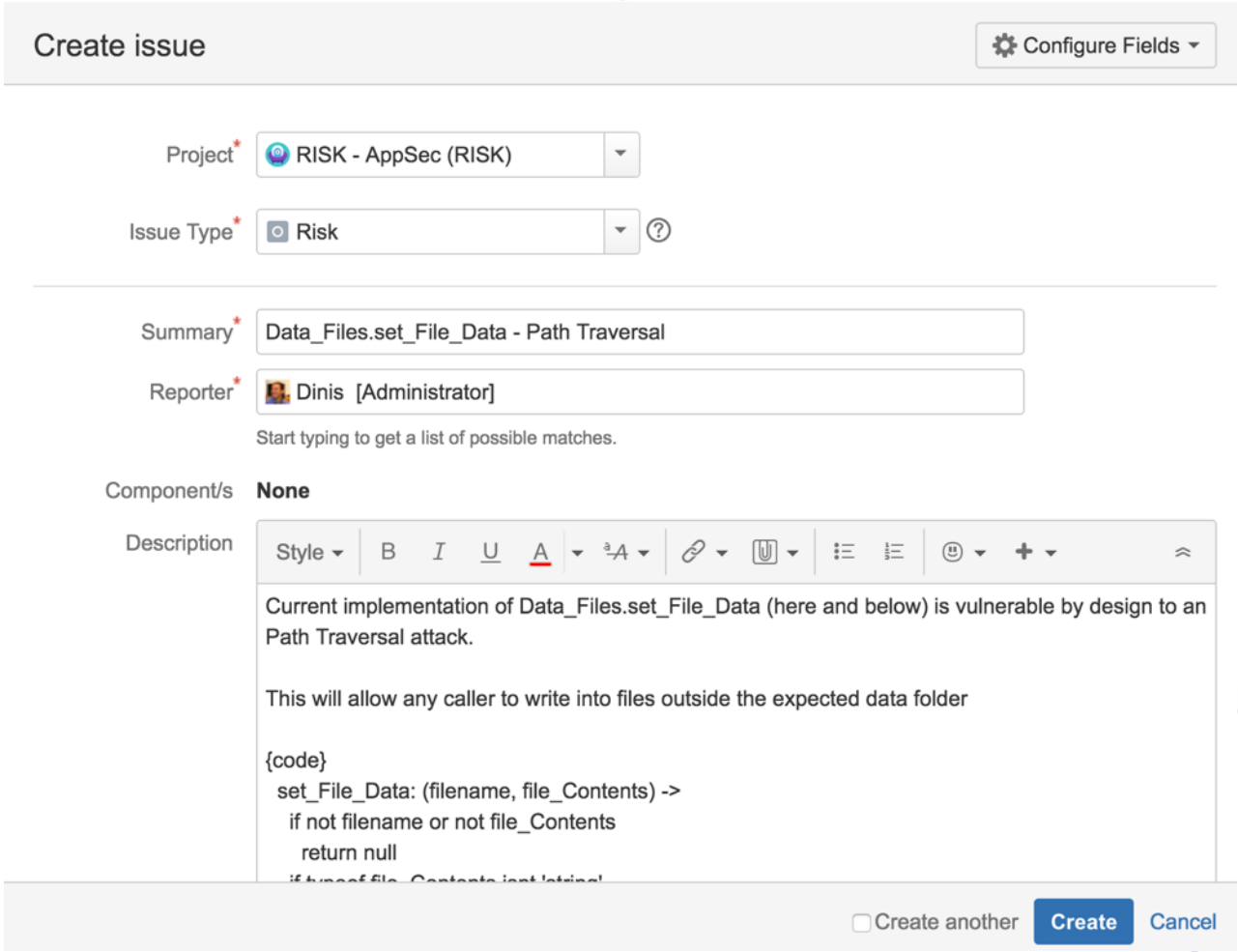


PATH #2 - Variation when risk not approved



‘FIX’ PATH

Issue: Data_Files.set_File_Data - Path Traversal



Status: OPEN

RISK - AppSec / RISK-4

Data_Files.set_File_Data - Path Traversal

Edit | Comment | Assign | More | Close | to 'In Progress' | Admin | Export

Details

Type:	Risk	Status:	OPEN (View Workflow)
Priority:	Medium	Resolution:	Unresolved
Labels:	None		

People

Assignee:	Unassigned Assign to me
Reporter:	Dinis [Administrator]
Votes:	0
Watchers:	1 Stop watching this issue

Dates

Created: Just now

Description

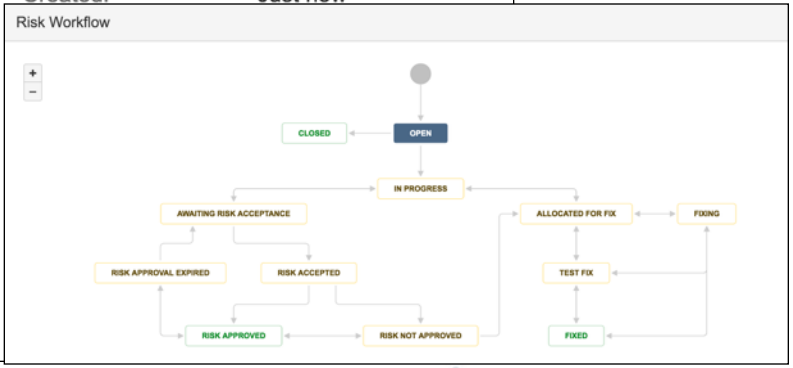
Current implementation of Data_Files.set_File_Data (here and below) is vulnerable by design to an Path Traversal attack.

This will allow any caller to write into files outside the expected data folder

```
set_File_Data: (filename, file_Contents) ->
  if not filename or not file_Contents
    return null
  if typeof file_Contents isnt 'string'
    return null
  file_Path = @.find filename
  if file_Path is null or file_Path.file_Not_Exists()
    file_Path = @.data_Path.path_Combine filename
  file_Path.file_Write file_Contents
  return file_Path
```

At the moment this method is not wired to a controller, but that is exactly what will happen next (since the point of this method is to allow the existing BSIMM mappings to be edited

Risk Workflow



Status: IN PROGRESS

RISK - AppSec / RISK-4

Data_Files.set_File_Data - Path Traversal

Edit | Comment | Assign | More | Allocate for Fix | Request Risk Acceptance | Admin | Export

Details

Type:	Risk	Status:	IN PROGRESS (View Workflow)
Priority:	Medium	Resolution:	Unresolved
Labels:	None		

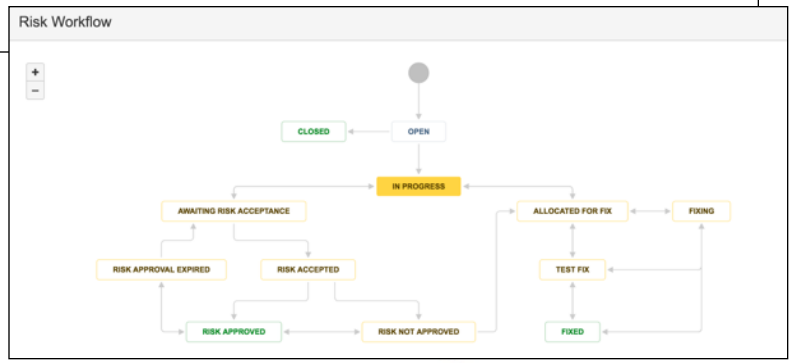
People

Assignee:	Unassigned Assign to me
Reporter:	Dinis [Administrator]
Votes:	0
Watchers:	1 Stop watching this issue

Description

Current implementation of Data_Files.set_File_Data (here and below) is vulnerable by design to an Path Traversal attack.

This will allow any caller to write into files outside the expected data folder



Status: ALLOCATED FOR FIX

RISK - AppSec / RISK-4

Data_Files.set_File_Data - Path Traversal

Edit | Comment | Assign | More | Fixing | Test Fix | to 'In Progress' | Admin | Export

Details

Type:	Risk	Status:	ALLOCATED FOR FIX (View Workflow)
Priority:	Medium	Resolution:	Unresolved
Labels:	None		

People

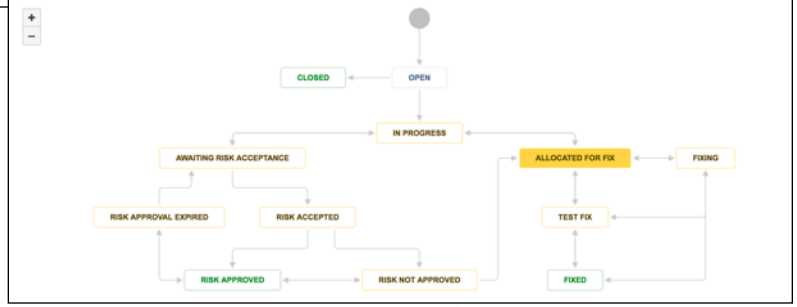
Assignee:	Unassigned Assign to me
Reporter:	Dinis [Administrator]
Votes:	0
Watchers:	1 Stop watching this issue

Description

Current implementation of Data_Files.set_File_Data (here and below) is vulnerable by design to an Path Traversal attack.

This will allow any caller to write into files outside the expected data folder

Risk Workflow



Status: FIXING

RISK - AppSec / RISK-4

Data_Files.set_File_Data - Path Traversal

Edit | Comment | Assign | More | Fixed | Test Fix | back to 'Allocated... | Admin | Export

Details

Type: Risk
Priority: Medium
Labels: None
Status: FIXING (View Workflow)
Resolution: Unresolved

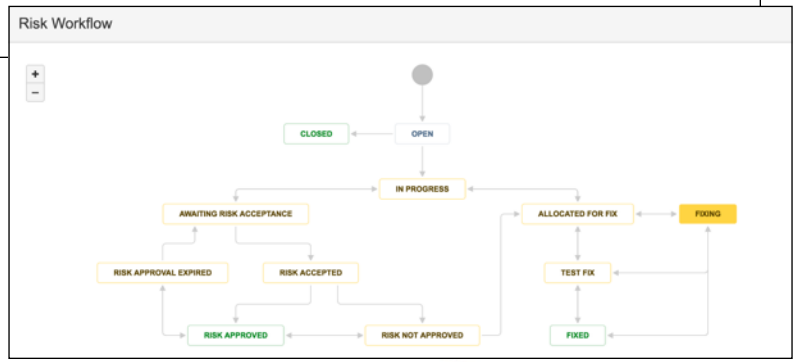
People

Assignee: Unassigned
Assign to me
Reporter: Dinis [Administrator]
Votes: 0
Watchers: 1 Stop watching this issue

Description

Current implementation of Data_Files.set_File_Data (here and below) is vulnerable by design to an Path Traversal attack.

This will allow any caller to write into files outside the expected data folder



Status: TEST FIX

RISK - AppSec / RISK-4

Data_Files.set_File_Data - Path Traversal

Edit | Comment | Assign | More | Fixed | back to 'Allocated... | back to 'Fixing' | Admin | Export

Details

Type:	Risk	Status:	TEST FIX (View Workflow)
Priority:	Medium	Resolution:	Unresolved
Labels:	None		

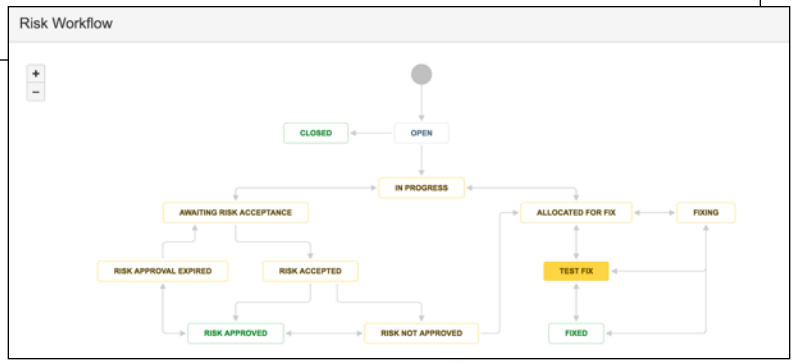
People

Assignee:	Unassigned Assign to me
Reporter:	Dinis [Administrator]
Votes:	0
Watchers:	1 Stop watching this issue

Description

Current implementation of Data_Files.set_File_Data (here and below) is vulnerable by design to an Path Traversal attack.

This will allow any caller to write into files outside the expected data folder



Status: FIXED

RISK - AppSec / RISK-4

Data_Files.set_File_Data - Path Traversal

Edit | Comment | Assign | More | back to 'Fixing' | back to 'Test Fix&...' | Admin | Export

Details

Type:	Risk	Status:	FIXED
Priority:	Medium		(View Workflow)
		Resolution:	Unresolved
Labels:	None		

People

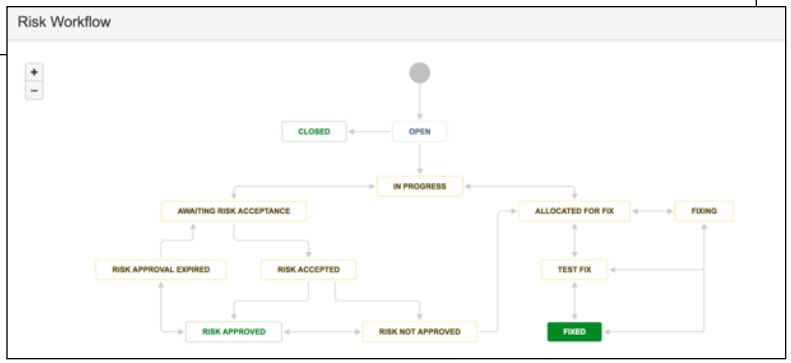
Assignee:	Unassigned
	Assign to me
Reporter:	Dinis [Administrator]
Votes:	0
Watchers:	1 Stop watching this issue

Description

Current implementation of Data_Files.set_File_Data (here and below) is vulnerable by design to an Path Traversal attack.

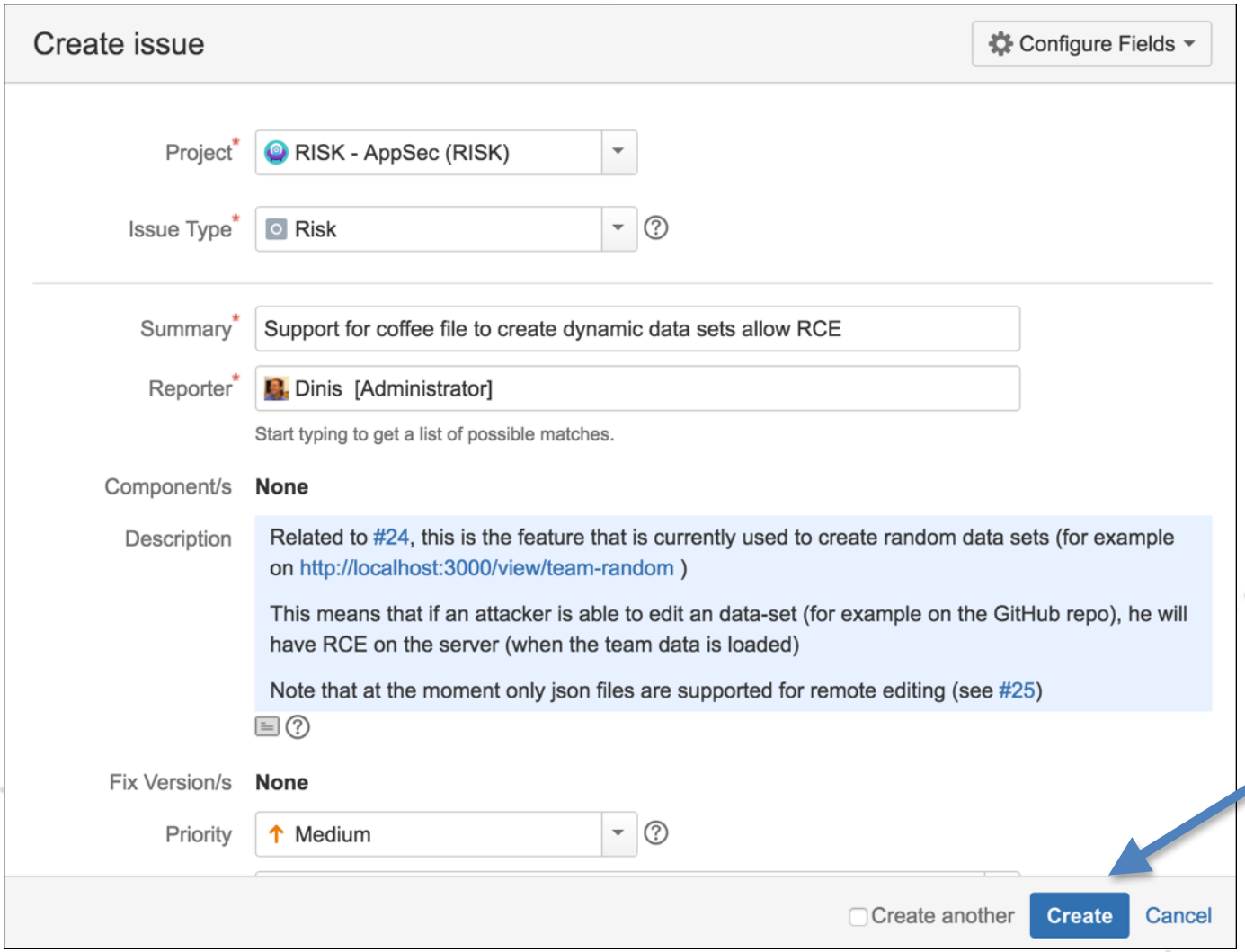
This will allow any caller to write into files outside the expected data folder

Risk Workflow



PATH 'RISK ACCEPT/APPROVE'

RISK: Support for coffee allows RCE



Status: OPEN

RISK - AppSec / RISK-3

Support for coffee file to create dynamic data sets allow RCE

Edit | Comment | Assign | More | Close | to 'In Progress' | Admin | Export

Details

Type:	Risk	Status:	OPEN (View Workflow)
Priority:	Medium	Resolution:	Unresolved
Labels:	None		

People

Assignee:	Unassigned Assign to me
Reporter:	Dinis [Administrator]
Votes:	0
Watchers:	1 Stop watching this issue

Description

Related to #24, this is the feature that is currently used to create random data sets (for example on http://localhost:3000/view/team-random)

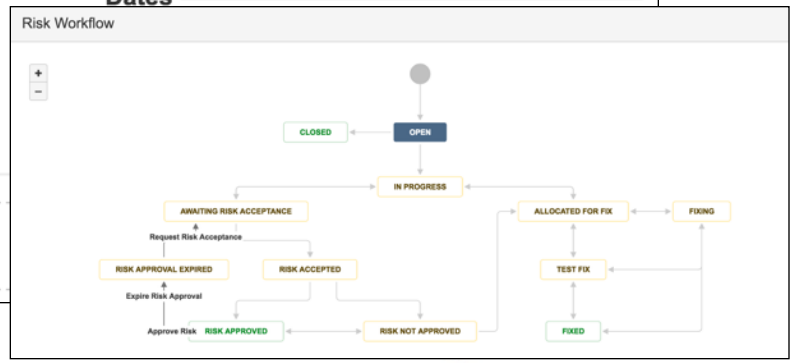
This means that if an attacker is able to edit an data-set (for example on the GitHub repo), he will have RCE on the server (when the team data is loaded)

Note that at the moment only json files are supported for remote editing (see #25)

Dates

Attachments

Drop files to attach, or browse.



Status: IN PROGRESS

RISK - AppSec / RISK-3

Support for coffee file to create dynamic data sets allow RCE

Edit | Comment | Assign | More | Allocate for Fix | Request Risk Acceptance | Admin | Export

Details

Type:	Risk	Status:	IN PROGRESS (View Workflow)
Priority:	Medium	Resolution:	Unresolved
Labels:	None		

People

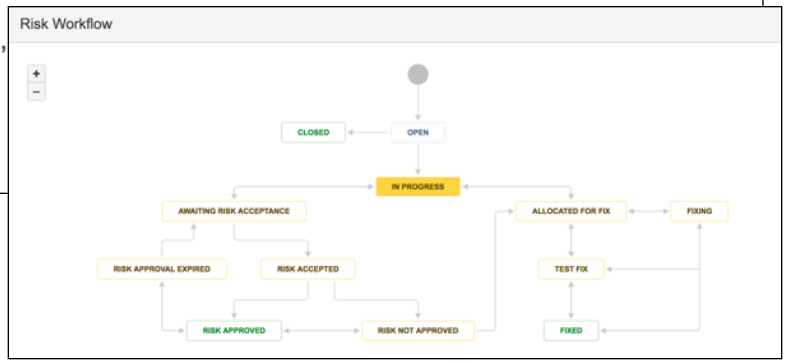
Assignee:	Unassigned Assign to me
Reporter:	Dinis [Administrator]
Votes:	0
Watchers:	1 Stop watching this issue

Description

Related to #24, this is the feature that is currently used to create random data sets (for example on http://localhost:3000/view/team-random)

This means that if an attacker is able to edit an data-set (for example on the GitHub repo), will have RCE on the server (when the team data is loaded)

Note that at the moment only json files are supported for remote editing (see #25)



Status: AWAITING RISK ACCEPTANCE

RISK - AppSec / RISK-3

Support for coffee file to create dynamic data sets allow RCE

Edit | Comment | Assign | More | Accept Risk | to 'In Progress' | Admin | Export

Details

Type: Risk
Priority: Medium
Labels: None
Status: AWAITING RISK ACC... (View Workflow)
Resolution: Unresolved

People

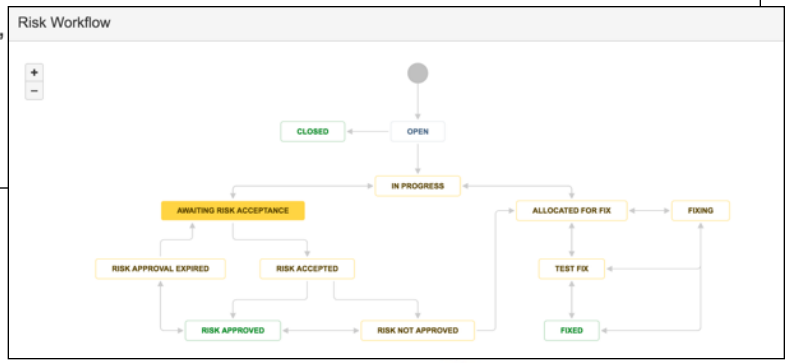
Assignee: Unassigned
Assign to me
Reporter: Dinis [Administrator]
Votes: 0
Watchers: 1 Stop watching this issue

Description

Related to #24, this is the feature that is currently used to create random data sets (for example on http://localhost:3000/view/team-random)

This means that if an attacker is able to edit an data-set (for example on the GitHub repo), will have RCE on the server (when the team data is loaded)

Note that at the moment only json files are supported for remote editing (see #25)



Status: RISK ACCEPTED

RISK - AppSec / RISK-3

Support for coffee file to create dynamic data sets allow RCE

Edit | Comment | Assign | More | Approve Risk | Don't Approve Risk | Admin | Export

Details

Type:	Risk	Status:	RISK ACCEPTED (View Workflow)
Priority:	Medium	Resolution:	Unresolved
Labels:	None		

People

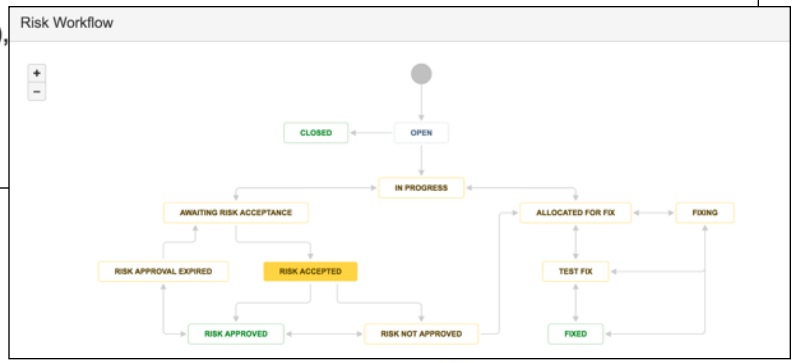
Assignee:	Unassigned Assign to me
Reporter:	Dinis [Administrator]
Votes:	0
Watchers:	1 Stop watching this issue

Description

Related to #24, this is the feature that is currently used to create random data sets (for example on http://localhost:3000/view/team-random)

This means that if an attacker is able to edit an data-set (for example on the GitHub repo), will have RCE on the server (when the team data is loaded)

Note that at the moment only json files are supported for remote editing (see #25)



Status: RISK APPROVED

RISK - AppSec / RISK-3

Support for coffee file to create dynamic data sets allow RCE

Edit | Comment | Assign | More | Expire Risk Approval | Don't Approve Risk | Admin | Export

Details

Type: Risk
Priority: Medium
Labels: None
Status: RISK APPROVED (View Workflow)
Resolution: Unresolved

People

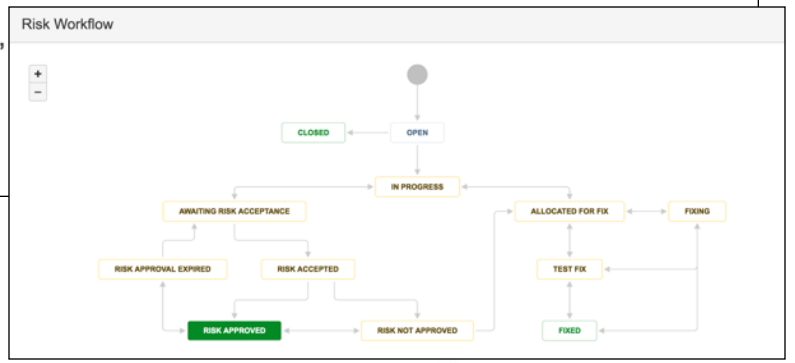
Assignee: Unassigned
Assign to me
Reporter: Dinis [Administrator]
Votes: 0
Watchers: 1 Stop watching this issue

Description

Related to #24, this is the feature that is currently used to create random data sets (for example on http://localhost:3000/view/team-random)

This means that if an attacker is able to edit an data-set (for example on the GitHub repo), will have RCE on the server (when the team data is loaded)

Note that at the moment only json files are supported for remote editing (see #25)



Status: RISK APPROVED EXPIRED

RISK - AppSec / RISK-3

Support for coffee file to create dynamic data sets allow RCE

Edit | Comment | Assign | More | Request Risk Acceptance | Approve Risk | Admin | Export

Details

Type:	Risk	Status:	RISK APPROVAL EX... (View Workflow)
Priority:	Medium	Resolution:	Unresolved
Labels:	None		

People

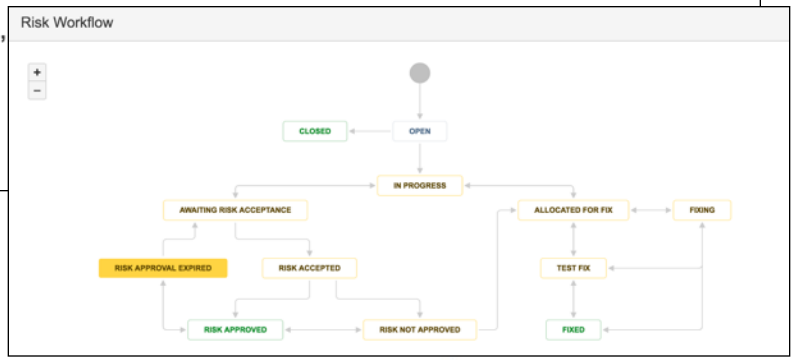
Assignee:	Unassigned Assign to me
Reporter:	Dinis [Administrator]
Votes:	0
Watchers:	1 Stop watching this issue

Description

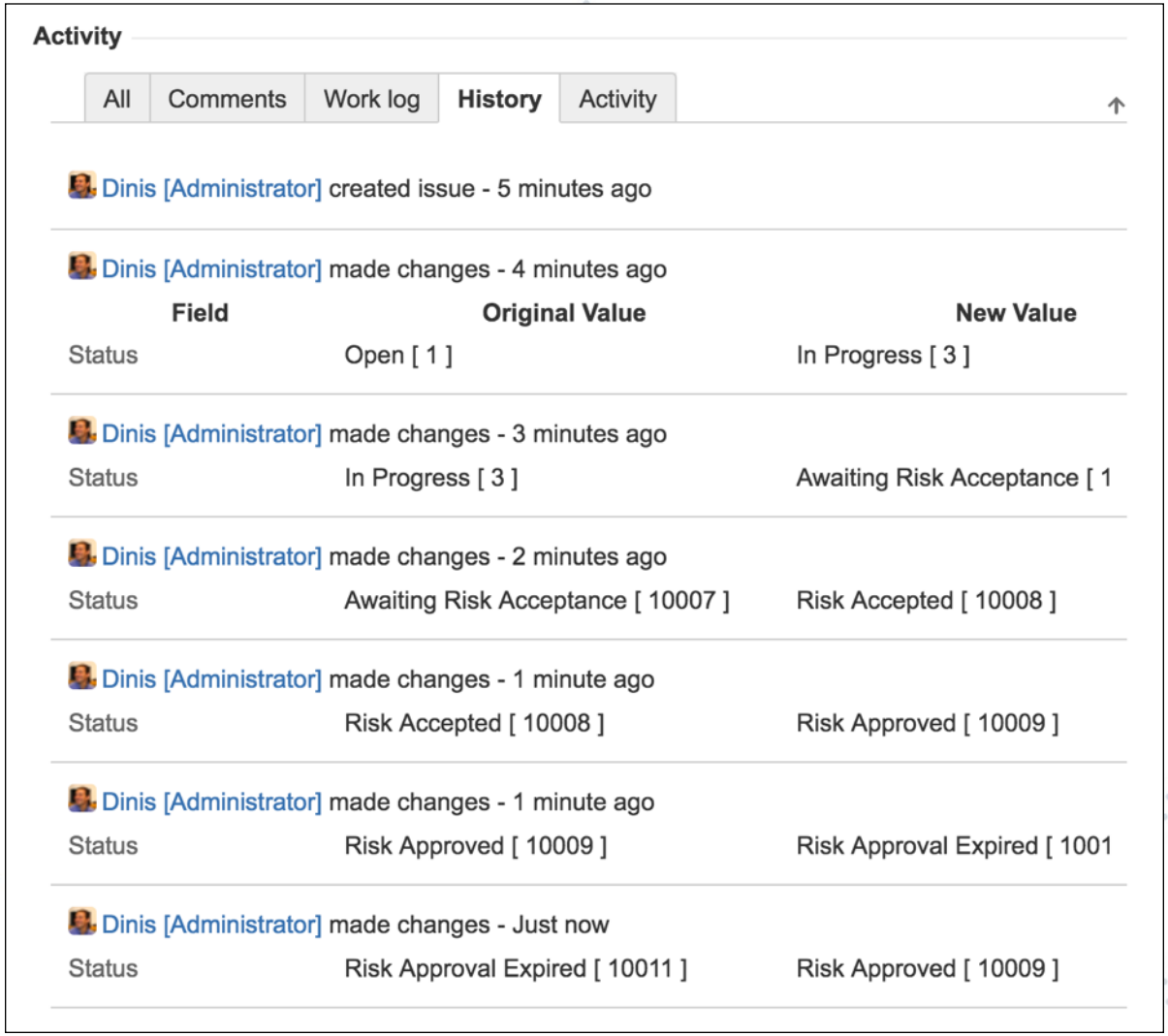
Related to #24, this is the feature that is currently used to create random data sets (for example on http://localhost:3000/view/team-random)

This means that if an attacker is able to edit an data-set (for example on the GitHub repo), will have RCE on the server (when the team data is loaded)

Note that at the moment only json files are supported for remote editing (see #25)



All status changes are tracked



Activity

All | Comments | Work log | **History** | Activity

Dinis [Administrator] created issue - 5 minutes ago

Dinis [Administrator] made changes - 4 minutes ago

Field	Original Value	New Value
Status	Open [1]	In Progress [3]

Dinis [Administrator] made changes - 3 minutes ago

Status	In Progress [3]	Awaiting Risk Acceptance [1

Dinis [Administrator] made changes - 2 minutes ago

Status	Awaiting Risk Acceptance [10007]	Risk Accepted [10008]

Dinis [Administrator] made changes - 1 minute ago

Status	Risk Accepted [10008]	Risk Approved [10009]

Dinis [Administrator] made changes - 1 minute ago

Status	Risk Approved [10009]	Risk Approval Expired [1001

Dinis [Administrator] made changes - Just now

Status	Risk Approval Expired [10011]	Risk Approved [10009]

CASE STUDY: WHEN I CREATED A VULNERABILITY

Feature request: Allow data editing on UI

- Here is the code I wrote (at the Data Layer)

```
56    set_File_Data: (filename, file_Contents) ->
57      if not filename or not file_Contents
58        return null
59      if typeof file_Contents isnt 'string'
60        return null
61      file_Path = @.find filename
62      if file_Path is null or file_Path.file_Not_Exists()
63        file_Path = @.data_Path.path_Combine filename
64      file_Path.file_Write file_Contents
65      return file_Path
```

- This method is designed to be called by the controller (i.e. rest api endpoint):

Data_Files.set_File_Data - Path Traversal #19

Closed **DinisCruz** opened this issue 27 days ago · 2 comments

DinisCruz commented 27 days ago • edited Owner

Current implementation of Data_Files.set_File_Data (here and below) is vulnerable by design to an Path Traversal attack.

This will allow any caller to write into files outside the expected data folder

```
set_File_Data: (filename, file_Contents) ->
  if not filename or not file_Contents
    return null
  if typeof file_Contents isnt 'string'
    return null
  file_Path = @.find filename
  if file_Path is null or file_Path.file_Not_Exists()
    file_Path = @.data_Path.path_Combine filename
  file_Path.file_Write file_Contents
  return file_Path
```

Regression test that passes on issue

```
describe '_securtiy | A1 - Injection', ->

  # https://github.com/DinisCruz/BSIMM-Graphs/issues/21
  it 'Issue 19 - Data_Files.set_File_Data - Path Traversal', ->
    using new Data_Files(), ->
      folder_Name  = 'outside-data-root'
      file_Name    = 'some-file.txt'
      file_Content = 'some content'
      target_Folder = @.data_Path.path_Combine('../' + folder_Name)        # Create target
                                 .folder_Create()
                                 .assert_Folder_Exists()                   # Confirm it ex

      target_Folder.path_Combine(file_Name)                                # Create target
                   .file_Write(file_Content)
                   .assert_File_Exists()                                   # Confirm it ex

      payload     = "../#{folder_Name}/#{file_Name}"
      new_Content = 'new - content'

      @.data_Path.path_Combine(payload)
                 .file_Contents().assert_Is file_Content                   # Confirm origi

      @.set_File_Data payload, new_Content

      @.data_Path.path_Combine(payload)
                 .file_Contents().assert_Is_Not file_Content               # Confirm origi
                                 .assert_Is new_Content                    # Confirm that

      target_Folder.folder_Delete_Recursive().assert_Is_True()             # Delete temp f
```

Data_Files.set_File_Data - DoS via filename #20

Closed **DinisCruz** opened this issue 27 days ago · 1 comment

DinisCruz commented 27 days ago • edited Owner

As seen in #19 the `set_File_Data: (filename, file_Contents)` method does not check the size (and contents) of the filename and file_Contents variables.

The problem is that they are strings, which means that they can be huge:

- http://appsandsecurity.blogspot.co.uk/2013/05/should-string-be-abstract-class.html
- http://1raindrop.typepad.com/1_raindrop/2013/04/security-140-conversation-with-john-wilander.html
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String
- http://stackoverflow.com/questions/2219526/how-many-bytes-in-a-javascript-string
- http://stackoverflow.com/questions/24153996/is-there-a-limit-on-the-size-of-a-string-in-json-with-node-js

And since those values are used to on the name and contents of files written on disk, in addition to possible probs in the Node Heap, this function can be used to fill up the disk

Here is the test for this issue which proves that we can create large files and also detects some weird behaviours on the file name size (which is different in wallaby, mocha and travis)

```
it.only 'Issue 20 - Data_Files.set_File_Data - DoS via filename and file_Contents', ->
  using new Data_Files(), ->
    create_File = (file_Size, content_Size, should_Work)=>
      file_Name     = file_Size   .random_String()
      file_Contents = content_Size.random_String()
      file_Path     = @.data_Path .path_Combine(file_Name)

      file_Path.assert_File_Not_Exists()                    # confirm file doesn't exist

      @.set_File_Data file_Name, file_Contents              # PAYLOAD: create file

      if should_Work                                        # if it should work
        file_Path.assert_File_Exists()                      #   confirm file exists
        file_Path.file_Delete().assert_Is_True()            #   delete temp file
        file_Path.assert_File_Not_Exists()
      else                                                  # if not
        file_Path.assert_File_Not_Exists()                  #   confirm creation failed


    # testing multiple file sizes
    create_File 10 ,10 , true
    create_File 100,10 , true
    create_File 156,10 , true
    #create_File 157,10 , false                              # interesting in wallaby, af
    #create_File 208,10 , false                              #             in mocha, it's
    create_File 512,10 , false                              #             in travis the

    # testing multiple file contents
    create_File 10 ,10 , true                               # 10 bytes
    create_File 10 ,100 , true                              # 100 bytes
    create_File 10 ,10000 , true                            # 10 Kb
    create_File 10 ,1000000 , true                          # 1 Mb
    create_File 10 ,10000000 , true                         # 10 Mb - will work and take
    create_File 10 ,100000000 , true                        # 100 Mb - will work and tak
```

Data_Files.set_File_Data - allows creation of files with any extension #23

Closed **DinisCruz** opened this issue 27 days ago · 1 comment

DinisCruz commented 27 days ago • edited Owner

Related to #19 and #20 , at the moment there is no limitations on the type of files that can be saved.

According with the current design, the only file paths that should be supported are `.json` files

Here is the test that proves the issue

```
it 'Issue 23 - Data_Files.set_File_Data - allows creation of files with any extension', -
  using new Data_Files(), ->
    create_File = (extension)=>
      file_Name     = 10.random_String() + extension
      file_Contents = 10.random_String()
      file_Path     = @.data_Path .path_Combine(file_Name)

      @.set_File_Data file_Name, file_Contents              # PAYLOAD: create file

      file_Path.assert_File_Exists()                        #   confirm file exists
               .file_Delete().assert_Is_True()              #   delete temp file

    create_File '.json'                                     # these are the ones that sh

    create_File '.json5'                                    # these are the ones that sh
    create_File '.coffee'
    create_File '.js'
    create_File '.exe'
    create_File '.html'
    create_File '.css'
    create_File '...'
```

Data_Files.set_File_Data - allows editing of coffee-script files (RCE) #24

Edit | New issue

Closed **DinisCruz** opened this issue 27 days ago · 3 comments

DinisCruz commented 27 days ago • edited — Owner

Related to #23 it will be possible to do RCE on the server by editing one of the existing data coffee-scripts files (for example the one used to create random data)

Here is the code from `Data-Files` that creates the security issue, note how the file is updated and the code is executed

```
it 'Issue 24 - Data_Files.set_File_Data - allows editing of coffee-script files (RCE)', -

  using new Data_Files(), ->
    # PREPARE
    new_File_Contents = 'module.exports = ()-> 40+2'
    file_Name         = 'coffee-data'
    file_Path         = @.find_File file_Name
    file_Contents = file_Path.file_Contents()
    @.get_File_Data(file_Name).user.assert_Is 'in coffee'        # confirm original data

    # TEST
    @.set_File_Data file_Name, new_File_Contents                 # PAYLOAD make change
    file_Path.file_Contents().assert_Is new_File_Contents        # confirm it was changed
    delete require.cache[file_Path]                              # clean the node cache
    @.get_File_Data(file_Name).assert_Is '42'                    # it should be 42 now (v

    # CLEAN
    @.set_File_Data file_Name, file_Contents                     # restore file contents
    file_Path.file_Contents().assert_Is file_Contents            # confirm it was reset c
    delete require.cache[file_Path]                              # clear the cache again
    @.get_File_Data(file_Name).user.assert_Is 'in coffee'        # confirm original data
```

Labels

- risk - fixed
- risk - high
- security

Milestone

No milestone

Assignees

No one—assign yourself

1 participant

Notifications

Unsubscribe

You're receiving notifications because you modified the open/close state.

Lock conversation

Fix for Path transversal

DinisCruz commented 27 days ago • edited Owner

This has now been fixed.

Here is the updated version of this method that doesn't have the path traversal issue

```
set_File_Data: (filename, file_Contents) ->

  if not filename or not file_Contents                # check if both values are set
    return null

  if typeof file_Contents isnt 'string'               # check if file_Contents is a strir
    return null

  file_Path = @.find_File filename                    # resolve file path based on file r

  if file_Path is null or file_Path.file_Not_Exists() # check if was able to resolve it
    return null

  file_Path.file_Write file_Contents
```

Regression test

For reference here is the regression test that confirms that it is not possible to write to files outside the data folder:

```
describe '_regression | A1 - Injection', ->

  # https://github.com/DinisCruz/BSIMM-Graphs/issues/21
  it 'Issue 19 - Data_Files.set_File_Data - Path Traversal', ->
    using new Data_Files(), ->
      folder_Name  = 'outside-data-root'
      file_Name    = 'some-file.txt'
      file_Content = 'some content'
      target_Folder = @.data_Path.path_Combine('../' + folder_Name)        # Create target
      .folder_Create()
      .assert_Folder_Exists()                    # Confirm it exists

      target_Folder.path_Combine(file_Name)                                # Create target
      .file_Write(file_Content)
      .assert_File_Exists()                                  # Confirm it exists

      payload     = "../#{folder_Name}/#{file_Name}"
      new_Content = 'new - content'

      @.data_Path.path_Combine(payload)
      .file_Contents().assert_Is file_Content                  # Confirm original content

      assert_Is_Null @.set_File_Data payload, new_Content                  # PAYLOAD: Creat

      @.data_Path.path_Combine(payload)
      .file_Contents().assert_Is file_Content                  # Confirm original content

      target_Folder.folder_Delete_Recursive().assert_Is_True()             # Delete temp fo
```

LET'S SEE HOW IT LOOKED IN THE CODE

...before the vuln is created

```
51  #set_File_Data: fileName
52
53  list: ()=>
54    @.files().file_Names()
55
56  files: =>
57    values = []
58    for file in @.data_Path.files_Recursive()
59      if  file.file_Extension() in ['.json', '.json5', '.coffee']
60        values.push file.remove(@.data_Path)
61    values
```

...when the vuln is created

```
56    set_File_Data: (filename, file_Contents) ->
57      if not filename or not file_Contents
58        return null
59      if typeof file_Contents isnt 'string'
60        return null
61      file_Path = @.find filename
62      if file_Path is null or file_Path.file_Not_Exists()
63        file_Path = @.data_Path.path_Combine filename
64      file_Path.file_Write file_Contents
65      return file_Path
```

… adding comments

```
56  set_File_Data: (filename, file_Contents) ->            # todo: add security issue: that this method will allow the writing
57    if not filename or not file_Contents                 #       of any file (not just the files in the data
58      return null                                        #       folder, which are the ones that should be edited)
59    if typeof file_Contents isnt 'string'
60      return null
61    file_Path = @.find filename                          # todo: add security issue: filename is not validated
62    if file_Path is null or file_Path.file_Not_Exists()
63      file_Path = @.data_Path.path_Combine filename       # todo: add security issue: directory transvesal
64    file_Path.file_Write file_Contents                    # todo: add security issue: no authorization, will write outside d
65    return file_Path
```

...after issues are created

```
54    # Issue 19 - Data_Files.set_File_Data - Path Traversal
55    # Issue 20 - Data_Files.set_File_Data - DoS via filename and file_Contents
56    # Issue 23 - Data_Files.set_File_Data - allows creation of files with any extension
57    set_File_Data: (filename, file_Contents) ->
58      if not filename or not file_Contents
59        return null
60      if typeof file_Contents isnt 'string'
61        return null
62      file_Path = @.find filename
63      if file_Path is null or file_Path.file_Not_Exists()
64        file_Path = @.data_Path.path_Combine filename
65      file_Path.file_Write file_Contents
66      return file_Path
```

...improving comments

```
54    # Issue 19 - Data_Files.set_File_Data - Path Traversal
55    # Issue 20 - Data_Files.set_File_Data - DoS via filename and file_Contents
56    # Issue 23 - Data_Files.set_File_Data - allows creation of files with any extension
57    set_File_Data: (filename, file_Contents) ->
58
59      if not filename or not file_Contents                # check if both values are set
60        return null
61
62      if typeof file_Contents isnt 'string'               # check if file_Contents is a string
63        return null
64
65      file_Path = @.find_File filename                    # resolve file path based on file name
66
67      if file_Path is null or file_Path.file_Not_Exists() # check if was able to resolve it
68        return null
69
70      file_Path.file_Write file_Contents
```

...updating issues after 1st fix

```
60    # Issue 24 - Data_Files.set_File_Data - allows editing of coffee-script files (RCE)
61    # Issue 25 - Refactor set_File_Data to Set_File_Data_JSON
62    # Issue 26 - Data_Files.set_File_Data - DoS via file_Contents
63    set_File_Data: (filename, file_Contents) ->
64
65      if not filename or not file_Contents                # check if both values are set
66        return null
67
68      if typeof file_Contents isnt 'string'               # check if file_Contents is a string
69        return null
70
71      file_Path = @.find_File filename                    # resolve file path based on file name
72
73      if file_Path is null or file_Path.file_Not_Exists() # check if was able to resolve it
74        return null
75
76      file_Path.file_Write file_Contents
```

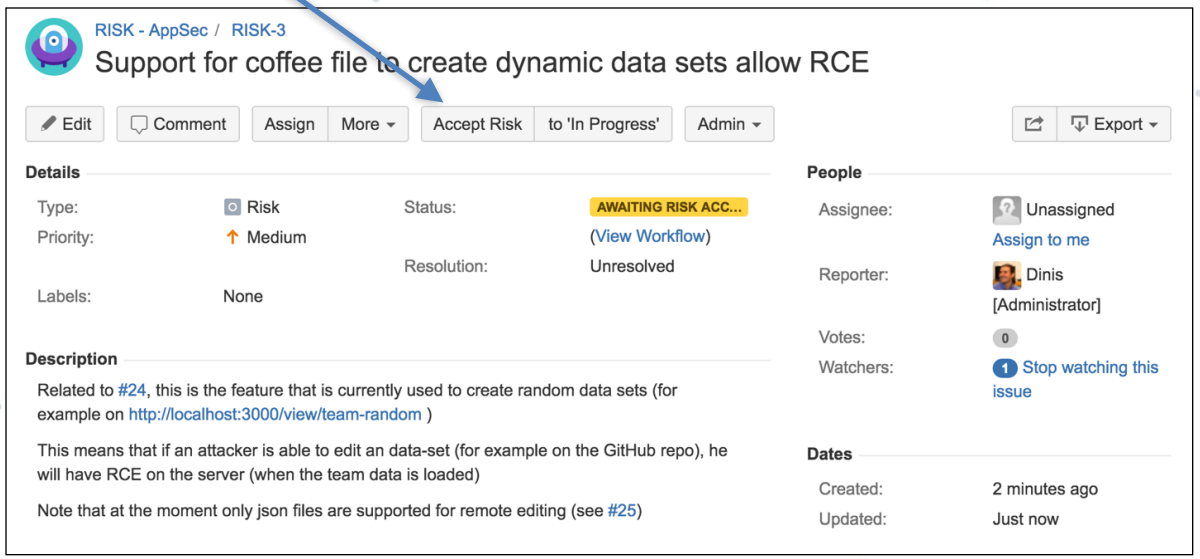
… after final fix

```
60  # Issue 26 - Data_Files.set_File_Data - DoS via file_Contents
61  set_File_Data_Json: (filename, json_Data) ->
62
63    if not filename or not json_Data                    # check if both values are set
64      return null
65
66    if typeof json_Data isnt 'string'                   # check if file_Contents is a string
67      return null
68
69    try                                                 # confirm that json_Data parses OK into JSON
70      JSON.parse json_Data
71    catch
72      return null
73
74    file_Path = @.find_File filename                    # resolve file path based on file name
75
76    if file_Path is null or file_Path.file_Not_Exists() # check if was able to resolve it
77      return null
78
79    if file_Path.file_Extension() isnt '.json'          # check that the file is .json
80      return null
81
82
83    file_Path.file_Write json_Data                      # after all checks save file
84
85    return file_Path.file_Contents() is json_Data       # confirm file was saved ok
```

KEY CONCEPTS FOR JIRA RISK WORKFLOW

Key for AppSec JIRA workflow is this button

Accept Risk



Separate JIRA project

- This is a separate JIRA repo from the one used by devs
  - I like to call that project 'RISK'
  - This avoids project 'issue creation' politics and 'safe harbour for:
    - known issues
    - 'shadow of a vulnerability' issues
    - 'this could be an problem…' issues
    - 'app is still in development' issues
  - When deciding to fix an issue:
    - that is the moment to create an issue in the target project JIRA (or whatever bug tracking system they used)
  - When issue is fixed (and closed on target project JIRA):
    - AppSec confirms fix and closes RISK

Always moving until fix or acceptance

- Key is to understand that issues need to be moving on one of two paths:
  - Fix
  - Risk Accepted (and approved)
- Risks (i.e. issues) are never in 'Backlog'
- If an issue is stuck in **'allocated for fix'**, then it will be moved into the **'Awaiting Risk Acceptance'** stage

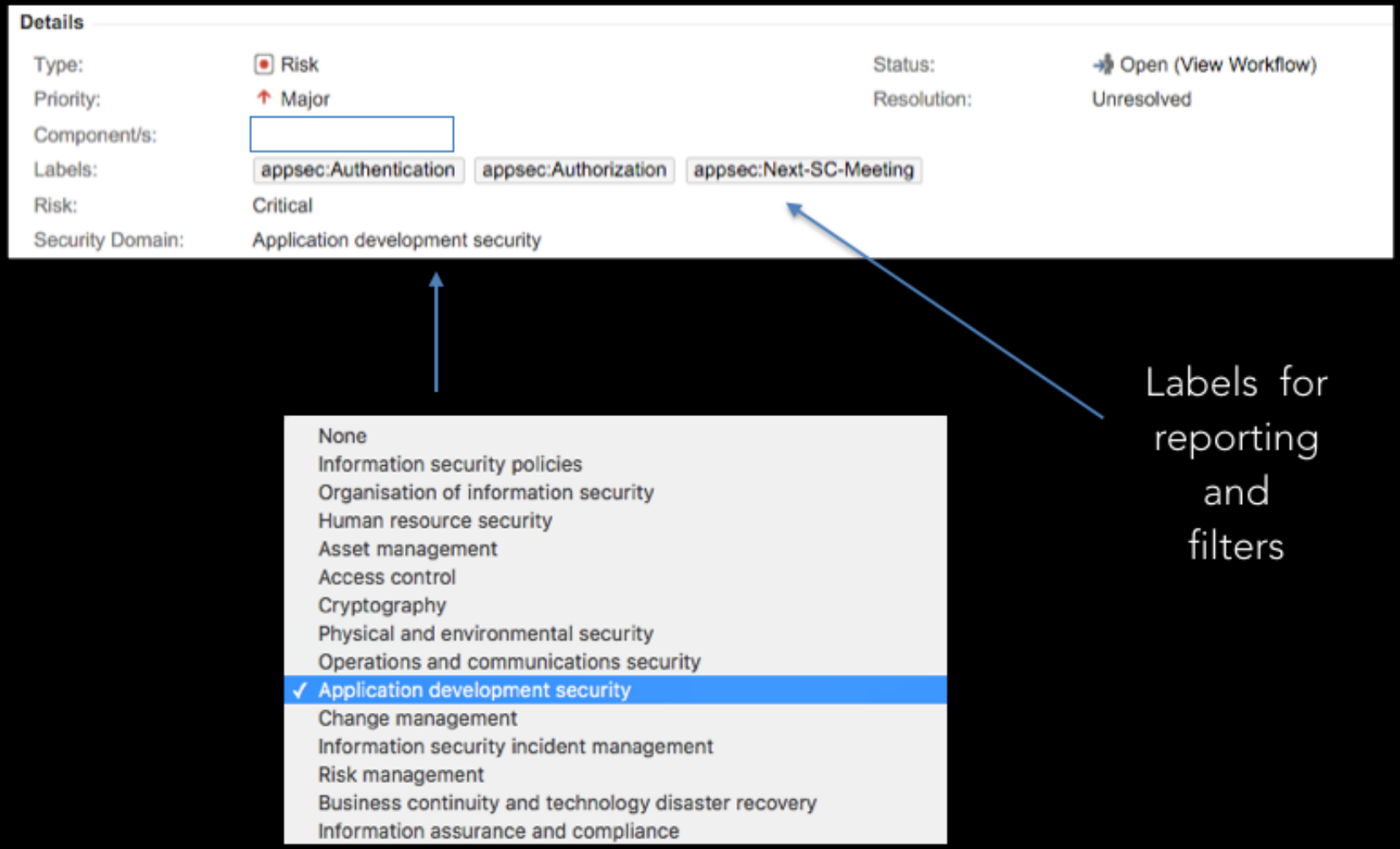
You need volume

- If you don't have 350+ issues on your JIRA RISK Project, you are not playing (and don't have enough visibility into what is really going on)
- Allow team A to see what team B had (and scale due due to issue description reuse)
- Problem is not teams with 50 issues, prob is team with 5 issues
- This is perfect for Gamification and to provide visibility into who to reward (and promote)

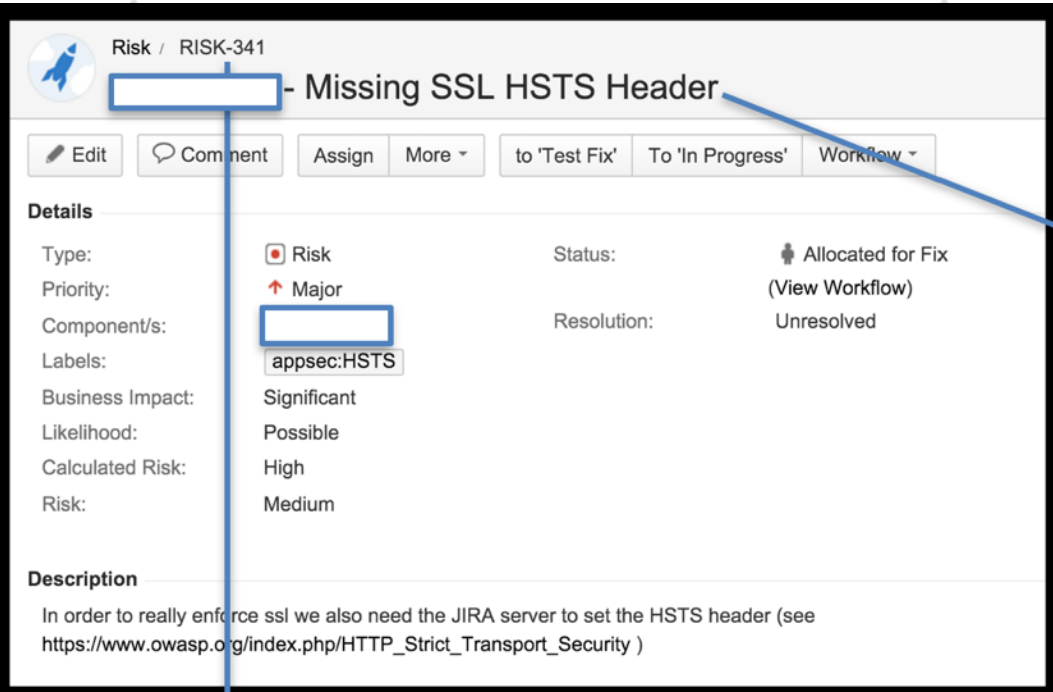
Threat model

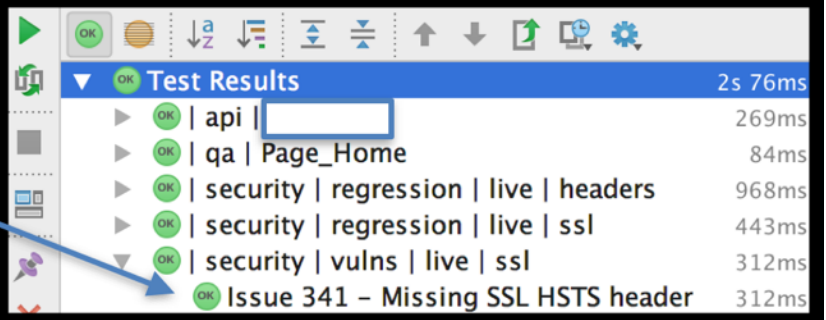
- All issues identified in Threat Models are added to the JIRA RISK project
- Create Threat models by
  - layer
  - feature
  - bug
- … that is a topic for another talk

Mapping to InfoSec risks



Mapping JIRA Tickets to Tests





vulns.live.ssl.coffee

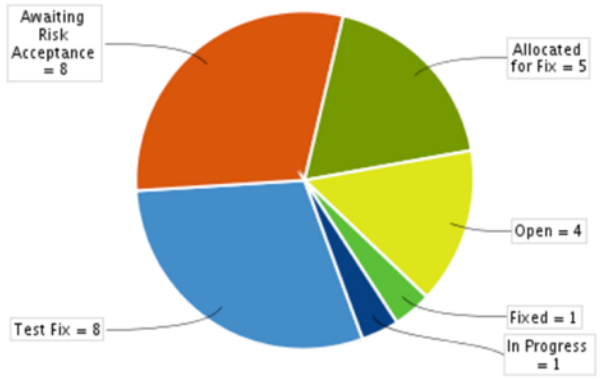
```
request = require 'request'

describe ' security | vulns | live | ssl', ->
  it 'Issue 341 - Missing SSL HSTS header', (done)->
    url = '
    request.get url, (error, res, data)->
      res.headers.keys().size().assert_Is 13
      res.headers.keys().assert_Is  [ 'date', 'x-frame-options', 'set-cookie','expires','cache-control','pragma',
                                      'access-control-allow-origin','access-control-allow-methods',
                                      'access-control-allow-headers',
                                      'keep-alive','connection', 'transfer-encoding','content-type' ]
      assert_Is_Undefined res.headers['strict-transport-security']
      done()
```

JIRA AppSec Dashboards

AppSec - [redacted]

Tools

Pie Chart: RISK - Blogs



Total Issues: **27** Statistic Type: **Status**

Filter Results: Risk - [redacted] Awaiting Risk Acceptance

Key	Summary	Risk	Status
RISK-	[redacted]		Awaiting Risk Acceptance
RISK-			Awaiting Risk Acceptance
RISK-			Awaiting Risk Acceptance
RISK-			Awaiting Risk Acceptance
RISK-			Awaiting Risk Acceptance
RISK-			Awaiting Risk Acceptance
RISK-			Awaiting Risk Acceptance
RISK-			Awaiting Risk Acceptance

1–8 of 8

Filter Results: Risk - [redacted] Allocated for Fix

Key	Summary	Risk	Status
RISK-	[redacted]	High	Allocated for Fix
RISK-		Low	Allocated for Fix
RISK-		Medium	Allocated for Fix
RISK-		Medium	Allocated for Fix
RISK-		Medium	Allocated for Fix

Filter Results: Risk - [redacted] Risk Accepted

No matching issues found.

Filter Results: Risk - [redacted] Test Fix

Key	Summary	Risk	Status
RISK-288	[redacted]	Medium	Test Fix
RISK-284		Medium	Test Fix

Weekly emails with Risk status

[redacted] Today at 08:19 BP

To: [redacted] Cc: Dinis Cruz

[redacted] Risk Accepted

Hi,

We are reviewing the areas of the risk that has been accepted.

The following screenshot shows your team's related ticket.

Filter Results: [redacted] - Risk Accepted

Key	Summary	Assignee	Risk Accepted by
RISK-1	[redacted]	[redacted]	[redacted]
RISK-2			
RISK-2			
RISK-2			
RISK-2			
RISK-1			
RISK-1			
RISK-1			
RISK-1			
RISK-1			

1–10 of 27

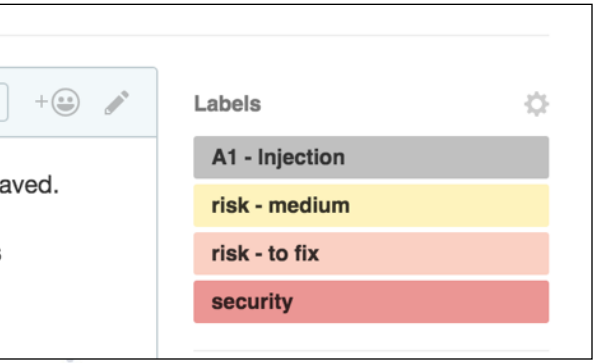
1 2 3

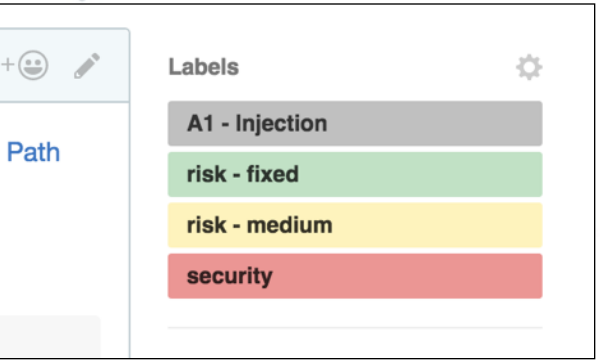
Other powerful JIRA features

- Components (one per team or project)
- Labels (to add metadata to issues, for OWASP Top 10)
- Links
  - connect with internal/external issues and
  - external resources
- Auto emails
- Copy and paste of images into description
- Markdown
- Security restrictions (use with care)
- Security lock certain actions
- Extra workflow actions for example when moving state)
- Create APPSEC JIRA project for AppSec related tasks (like 'Create Threat Model for app XYZ')

GITHUB RISK WORKFLOW

Using GitHub (instead of JIRA)





- **Server web root (i.e. path) is exposed by API** A6 - Sensitive Data Exposure risk - accepted risk - low security test needed
  #31 opened 26 days ago by DinisCruz
- **All server logs are exposed via API** A6 - Sensitive Data Exposure risk - accepted risk - low security test needed
  #30 opened 26 days ago by DinisCruz
- **Data_Files.set_File_Data - DoS via file_Contents** A1 - Injection risk - accepted risk - low security 1
  #26 opened 27 days ago by DinisCruz
- **Data_Files.set_File_Data - allows editing of coffee-script files (RCE)** risk - fixed risk - high security 3
  #24 opened 27 days ago by DinisCruz
- **Data_Files.set_File_Data - allows creation of files with any extension** A1 - Injection risk - medium risk - to fix security 1
  #23 opened 27 days ago by DinisCruz
- **Write regression test to prove that Data-Files.find method is not vulnerable to A1-Injection** A1 - Injection risk - accepted security test needed 1
  #22 opened 27 days ago by DinisCruz
- **Data_Files.set_File_Data - DoS via filename** risk - fixed risk - medium security test needed 1
  #20 opened 27 days ago by DinisCruz
- **Data_Files.set_File_Data - Path Traversal** A1 - Injection risk - fixed risk - medium security 2
  #19 opened 27 days ago by DinisCruz
- **Api-Controller - filename is a string and it is not validated** risk - accepted security test needed 1
  #18 opened 28 days ago by DinisCruz
- **There is no data classification of assets used** A6 - Sensitive Data Exposure risk - accepted risk - low security 1
  #17 opened 28 days ago by DinisCruz

Data_Files.set_File_Data - allows editing of coffee-script files (RCE) risk - fixed risk - high security

#24 opened 27 days ago by DinisCruz

Data_Files.set_File_Data - DoS via filename risk - fixed risk - medium security test needed

#20 opened 27 days ago by DinisCruz

Data_Files.set_File_Data - Path Traversal A1 - Injection risk - fixed risk - medium security

#19 opened 27 days ago by DinisCruz

Example with DoS issue

Project list gets data from File System and allows DoS (with large amounts of requests) #72

Edit | New issue

Closed **DinisCruz** opened this issue 17 days ago · 3 comments

DinisCruz commented 17 days ago • edited — Owner — Labels

this code will transverse the file system

```
# returns a list of current projects (which are defined by a folder contain
list: ()=>
  projects = {}
  for folder in @.data_Path.folders_Recursive()
    config_File = folder.path_Combine @.config_File
    if config_File.file_Exists()
      data = config_File.load_Json()
      if data and data.key
        projects[data.key] =
          path: folder
          data: data
  projects
```

Here is the test that confirms the DoS prob

```
describe '_regression | A11 - DoS', ->

  # https://github.com/DinisCruz/BSIMM-Graphs/issues/72
  it 'Issue 72 - Project list gets data from File System and could cause DoS', ()->
    using new Data_Project(), ->
      start = Date.now();
      test_List = (index, next)=>
        @.list().assert_Is_Object()
        next()

      #items = [0..0     ]   # 1 takes 15ms
```

DinisCruz added test needed | risk - to accept | security | A11 - DoS | risk - accepted and removed risk - to accept labels 17 days ago

DinisCruz commented 17 days ago — Owner

accepting risk

DinisCruz closed this 17 days ago

or 10 it should take between 20ms and 80ms

TDD

TDD

- For TDD to be productive you need
  - Real time unit test execution (when hands lift)
  - Real time code coverage
- TDD focus needs to be on
  - making developers more productive
  - preventing developers from switching context
- If 99% code coverage doesn't happen 'by default' TDD workflow is not working

TDD in WebStorm with WallabyJS

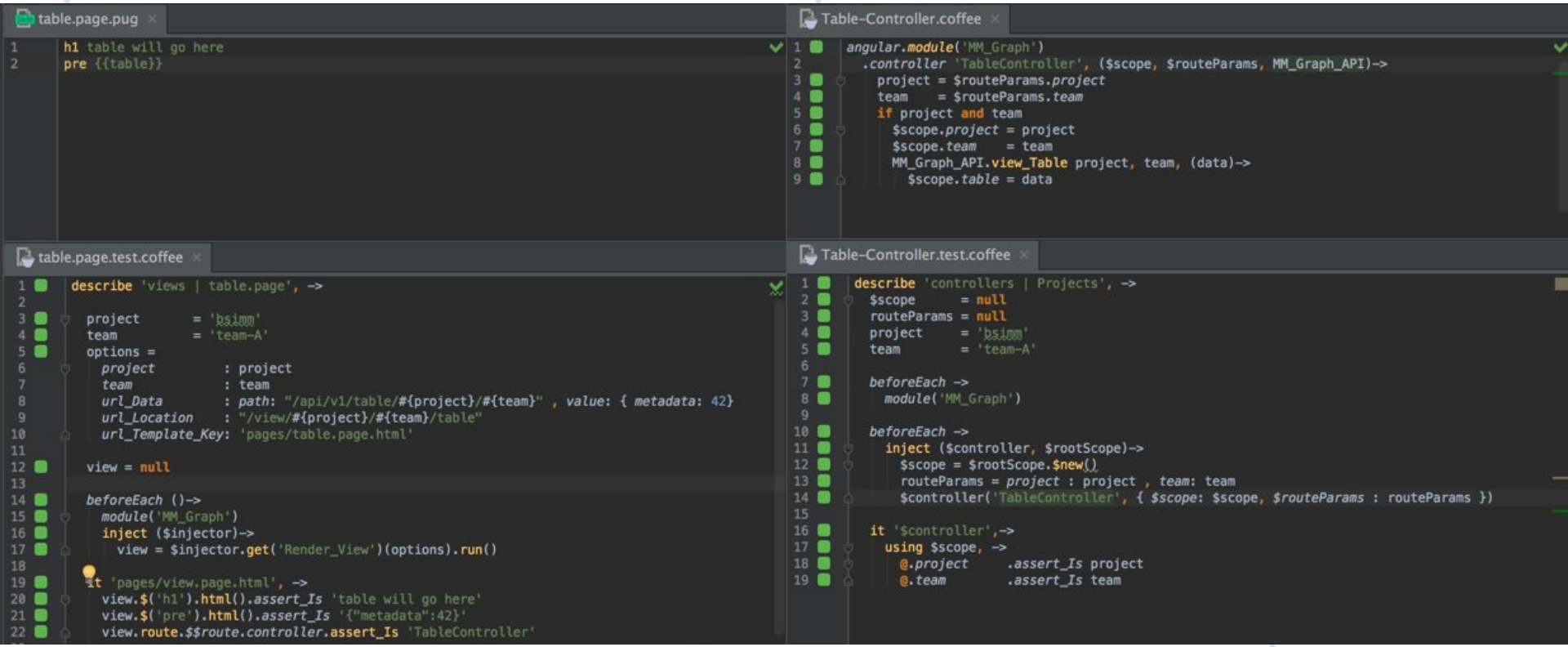


table.page.pug

```
h1 table will go here
pre {{table}}
```

Table-Controller.coffee

```
angular.module('MM_Graph')
  .controller 'TableController', ($scope, $routeParams, MM_Graph_API)->
    project = $routeParams.project
    team    = $routeParams.team
    if project and team
      $scope.project = project
      $scope.team    = team
      MM_Graph_API.view_Table project, team, (data)->
        $scope.table = data
```

table.page.test.coffee

```
describe 'views | table.page', ->

  project       = 'bsimm'
  team          = 'team-A'
  options =
    project         : project
    team            : team
    url_Data        : path: "/api/v1/table/#{project}/#{team}" , value: { metadata: 42}
    url_Location    : "/view/#{project}/#{team}/table"
    url_Template_Key: 'pages/table.page.html'

  view = null

  beforeEach ()->
    module('MM_Graph')
    inject ($injector)->
      view = $injector.get('Render_View')(options).run()

  it 'pages/view.page.html', ->
    view.$('h1').html().assert_Is 'table will go here'
    view.$('pre').html().assert_Is '{"metadata":42}'
    view.route.$$route.controller.assert_Is 'TableController'
```

Table-Controller.test.coffee

```
describe 'controllers | Projects', ->
  $scope      = null
  routeParams = null
  project     = 'bsimm'
  team        = 'team-A'

  beforeEach ->
    module('MM_Graph')

  beforeEach ->
    inject ($controller, $rootScope)->
      $scope = $rootScope.$new()
      routeParams = project : project , team: team
      $controller('TableController', { $scope: $scope, $routeParams : routeParams })

  it '$controller',->
    using $scope, ->
      @.project     .assert_Is project
      @.team        .assert_Is team
```

What happens when you increase attack surface

Api-Logs.coffee

```coffee
Api_Base = require './Api-Base'

class Api_Logs extends Api_Base
  constructor: (options)->
    @.options     = options || {}
    @.logs_Folder = __dirname.path_Combine('../../logs')
    super()

  add_Routes: ()=>
    @.add_Route 'get', '/logs/path'        , @.path
    @.add_Route 'get', '/logs/list'        , @.list
    @.add_Route 'get', '/logs/file/:index', @.file
    #@.add_Route 'get', '/logs/path_2'      , @.path
    @

  list: (req, res)=>
    res.send @.logs_Folder.files().file_Names()

  file: (req, res)=>
    index = parseInt(req.params?.index)
    if is_Number(index)
      file_Name = @.logs_Folder.files().file_Names()[index]
      if file_Name
        file_Path = @.logs_Folder.path_Combine file_Name
        if file_Path.file_Exists()
          return res.send  file_Path.file_Contents()

    res.send 'not found'

  path: (req, res)=>
    res.send @.logs_Folder
```

Server.test.coffee

```coffee
    @.server_Url().assert_Is 'http://localhost:12345'

it 'routes', ()->
  using server, ->
    @.setup_Server()
    @.add_Controllers()
    @.add_Redirects()
    @.add_Angular_Route()
    version = '/api/v1'
    expected_Routes = [ '/', '/ping',  '/view*'
                        "#{version}/data/:project/:team/radar"
                        "#{version}/logs/path"
                        "#{version}/logs/list"
                        "#{version}/logs/file/:index"
                        "#{version}/team/:project/list"
                        "#{version}/team/:project/get/:team"
                        "#{version}/team/:project/save/:team"
                        "#{version}/project/list"
                        "#{version}/project/get/:project"
                        "#{version}/project/schema/:project"
                        "#{version}/routes/list"
                        "#{version}/routes/list-raw"
                        "#{version}/table/:project/:team"]
    current_Routes = @.routes()

    for route in expected_Routes
      current_Routes.assert_Contains route

    for route in current_Routes
      #console.log route
      expected_Routes.assert_Contains route
```

You want a test to fail

Api-Logs.coffee

```coffee
Api_Base = require './Api-Base'

class Api_Logs extends Api_Base
  constructor: (options)->
    @.options     = options || {}
    @.logs_Folder = __dirname.path_Combine('../../logs')
    super()

  add_Routes: ()=>
    @.add_Route 'get', '/logs/path'       , @.path
    @.add_Route 'get', '/logs/list'       , @.list
    @.add_Route 'get', '/logs/file/:index', @.file
    @.add_Route 'get', '/logs/path_2'     , @.path
    @

  list: (req, res)=>
    res.send @.logs_Folder.files().file_Names()

  file: (req, res)=>
    index = parseInt(req.params?.index)
    if is_Number(index)
      file_Name = @.logs_Folder.files().file_Names()[index]
      if file_Name
        file_Path = @.logs_Folder.path_Combine file_Name
        if file_Path.file_Exists()
          return res.send  file_Path.file_Contents()

    res.send 'not found'

  path: (req, res)=>
    res.send @.logs_Folder
```

Server.test.coffee

```coffee
    @.server_Url().assert_Is 'http://localhost:12345'

it 'routes', ()->
  using server, ->  [assert_Contains]
    @.setup_Server()
    @.add_Controllers()
    @.add_Redirects()
    @.add_Angular_Route()
    version = '/api/v1'
    expected_Routes = [ '/', '/ping',  '/view*'
                        "#{version}/data/:project/:team/radar"
                        "#{version}/logs/path"
                        "#{version}/logs/list"
                        "#{version}/logs/file/:index"
                        "#{version}/team/:project/list"
                        "#{version}/team/:project/get/:team"
                        "#{version}/team/:project/save/:team"
                        "#{version}/project/list"
                        "#{version}/project/get/:project"
                        "#{version}/project/schema/:project"
                        "#{version}/routes/list"
                        "#{version}/routes/list-raw"
                        "#{version}/table/:project/:team"]
    current_Routes = @.routes()

    for route in expected_Routes
      current_Routes.assert_Contains route

    for route in current_Routes
      console.log route  /api/v1/logs/path_2
      expected_Routes.assert_Contains route  [assert_Contains]
```

TDD in WebStorm with WallabyJS

Server.test.coffee | A1-Injection.coffee

```coffee
Data_Files = require '../../src/backend/Data-Files'

# These tests represent Injections Security Issues
# See https://www.owasp.org/index.php/Top_10_2013-A1-Injection for more references
describe '_security | A1 - Injection', ->

  # https://github.com/DinisCruz/BSIMM-Graphs/issues/20
  it 'Issue 26 - Data_Files.set_File_Data - DoS via file_Contents', ->

    using new Data_Files(), ->
      project = 'bsimm'
      file_Name = "save-test"
      file_Path = @.find_File(project,file_Name)
      create_File = (size)=>
        new_File_Contents = { data: size.random_String() }.json_Str()
        file_Path.assert_File_Exists()                              # confirm file exist (since the current version only allows files to be modified

        @.set_File_Data_Json project,file_Name, new_File_Contents   # set file with json data
                  .assert_Is_True()

        file_Path.file_Contents().assert_Is  new_File_Contents      # confirm file was changed
                                 .assert_Size_Is (size + 11)        # to the expected size (+11 from json data field and padding)

      restore_File = ()=>
        original_Content = { "will-be": "changed by tests" }
        @.set_File_Data_Json project,file_Name, original_Content.json_Pretty()
            .assert_Is_True()

      # testing multiple file contents
      create_File 10 ,10                                            # 10 bytes
      create_File 10 ,100                                           # 100 bytes
      create_File 10 ,10000                                         # 10 Kb
      create_File 10 ,1000000                                       # 1 Mb
      #create_File 10 ,10000000 , true                              # 10 Mb - will work and take about 250 ms
      #create_File 10 ,100000000 , true                             # 100 Mb - will work and take about 2 secs
      restore_File()
```

- … but is a topic for another talk :)

OWASP
Open Web Application Security Project

Thanks, any questions

@diniscruz

dinis.cruz@owasp.org