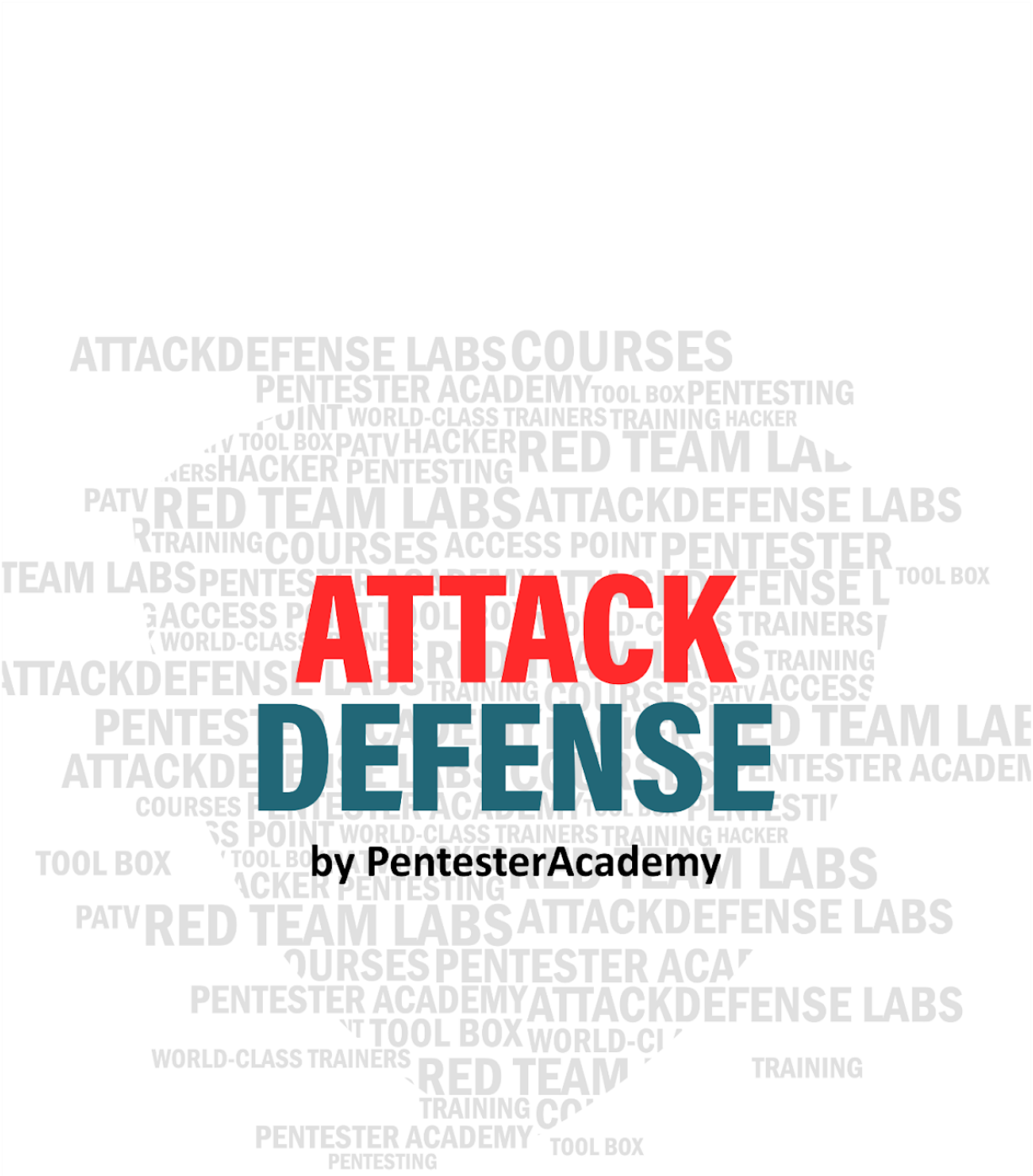
ATTACK
DEFENSE
by PentesterAcademy

Name	Brakeman: Finding Bugs in Ruby on Rails
URL	https://www.attackdefense.com/challengedetails?cid=2155
Type	DevSecOps Basics: Static Application Security Testing

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Challenge Description

The Brakeman tool is used to find vulnerabilities in Ruby on Rails-based applications.

A Kali CLI machine (kali-cli) is provided to the user with brakeman installed on it. The source code for three sample applications is provided in the home directory of the root user.

Objective: Use the brakeman utility to find vulnerabilities in the applications!

Instructions:

- The source code of applications is provided at /root/github-repos

Solution

Step 1: Check the provided applications.

Command: ls -l github-repos/

```
root@attackdefense:~# ls -l github-repos/
total 12
drwxrwxr-x 16 root root 4096 Nov 16 06:05 Autolab
drwxrwxr-x 17 root root 4096 Nov 16 06:05 ciao
drwxrwxr-x 14 root root 4096 Nov 16 06:05 CodeTriage
root@attackdefense:~#
```

We will take one example at a time and run the tool on that.

Example 1: Autolab

Step 1: Navigate to the Autolab directory.

Commands:
cd ~/github-repos/Autolab
ls

```
root@attackdefense:~# cd ~/github-repos/Autolab/
root@attackdefense:~/github-repos/Autolab#
root@attackdefense:~/github-repos/Autolab# ls
app     config.ru        db          docs      Gemfile.lock  mkdocs.yml  README.md  templates
bin     config.zip       docker      examples  lib           public      script
config  CONTRIBUTING.md  Dockerfile  Gemfile   LICENSE       Rakefile    spec
root@attackdefense:~/github-repos/Autolab#
```

Step 2: Run the brakemen tool in order to identify vulnerabilities in the application.

Command: brakeman

```
root@attackdefense:~/github-repos/Autolab# brakeman
Loading scanner...
Processing application in /root/github-repos/Autolab
Processing gems...
[Notice] Detected Rails 5 application
Processing configuration...
[Notice] Escaping HTML by default
Parsing files...
```

```
root@attackdefense:~/github-repos/Autolab# brakeman
Loading scanner...
Processing application in /root/github-repos/Autolab
Processing gems...
[Notice] Detected Rails 5 application
Processing configuration...
[Notice] Escaping HTML by default
Parsing files...
Processing initializers...
Processing libs...sed
Processing routes...
Processing templates...
Processing data flow in templates...
Processing models...
Processing controllers...
Processing data flow in controllers...
Indexing call sites...
```

```
Checks finished, collecting results...
Generating report...

== Brakeman Report ==

Application Path: /root/github-repos/Autolab
Rails Version: 5.2.0
Brakeman Version: 4.10.0
Scan Date: 2020-11-16 13:27:41 +0000
Duration: 12.292359102 seconds
Checks Run: BasicAuth, BasicAuthTimingAttack, CSRFTokenForgeryCVE, ContentTag, CookieSerialization, CreateWit
h, CrossSiteScripting, DefaultRoutes, Deserialize, DetailedExceptions, DigestDoS, DynamicFinders, EscapeFunct
ion, Evaluation, Execute, FileAccess, FileDisclosure, FilterSkipping, ForgerySetting, HeaderDoS, I18nXSS, JRu
byXML, JSONEncoding, JSONEntityEscape, JSONParsing, LinkTo, LinkToHref, MailTo, MassAssignment, MimeTypeDoS,
ModelAttrAccessible, ModelAttributes, ModelSerialize, NestedAttributes, NestedAttributesBypass, NumberToCurre
ncy, PageCachingCVE, PermitAttributes, QuoteTableName, Redirect, RegexDoS, Render, RenderDoS, RenderInline, R
esponseSplitting, RouteDoS, SQL, SQLCVEs, SSLVerify, SafeBufferManipulation, SanitizeMethods, SelectTag, Sele
ctVulnerability, Send, SendFile, SessionManipulation, SessionSettings, SimpleFormat, SingleQuotes, SkipBefore
Filter, SprocketsPathTraversal, StripTags, SymbolDoSCVE, TemplateInjection, TranslateBug, UnsafeReflection, V
alidationRegex, WithoutProtection, XMLDoS, YAMLParsing
```

```
== Warning Types ==

Authentication: 1
Command Injection: 14
Cross-Site Request Forgery: 1
Cross-Site Scripting: 8
```

```
Dangerous Eval: 1
Dynamic Render Path: 5
File Access: 20
Format Validation: 1
Mass Assignment: 1
Redirect: 1
SQL Injection: 1
Session Setting: 2
```

```
== Warnings ==

Confidence: High
Category: Command Injection
Check: Execute
Message: Possible command injection
Code: system("fs sa #{File.join(@course.assessments.find_by(:name => params[:name]).remote_handin_path, ((Use
r.find_by(:email => params[:user]).email + "_remote_handin_") + @course.assessments.find_by(:name => params[:
name]).name))} #{User.find_by(:email => params[:user]).email} rlidw")
File: app/controllers/assessment/handin.rb
Line: 210
```

```
Confidence: Medium
Category: Cross-Site Request Forgery
Check: CSRFTokenForgeryCVE
Message: Rails 5.2.0 has a vulnerability that may allow CSRF token forgery. Upgrade to Rails 5.2.4.3 or patch
File: Gemfile.lock
Line: 218

Confidence: Medium
Category: Cross-Site Scripting
Check: CrossSiteScripting
Message: Unescaped model attribute
Code: Course.find_by(:name => ((params[:course_name] or (params[:controller] == "courses") ? (params[:name])
: (nil)))).assessments.find_by!(:name => ((params[:assessment_name] or params[:name])))).embedded_quiz_form_da
ta
File: app/views/assessments/show.html.erb
Line: 259
```

```
Confidence: Medium
Category: SQL Injection
Check: SQL
Message: Possible SQL injection
Code: connection.execute("INSERT INTO #{table_name} (#{params.keys.join(", ")}) VALUES (#{params.values_at(*p
arams.keys).join(", ")})")
File: app/models/assessment_user_datum.rb
Line: 227
```

Warnings Detected:

- Command Injection
- Cross-Site Scripting
- SQL Injection

Example 2: Ciao

Step 1: Navigate to the Ciao directory.

Commands:
cd ~/github-repos/ciao
ls

```
root@attackdefense:~/github-repos/Autolab# cd ~/github-repos/ciao
root@attackdefense:~/github-repos/ciao# ls
app              db           LICENSE           README.md              tmp
babel.config.js  Dockerfile   log               scripts                vendor
bin              Gemfile      package.json      smtp_configuration.md  webhook_configuration.md
chart            Gemfile.lock postcss.config.js start.sh              yarn.lock
config           guestbook.md public            storage
config.ru        lib          Rakefile          test
root@attackdefense:~/github-repos/ciao#
```

Step 2: Run the brakemen tool in order to identify vulnerabilities in the application.

Command: brakeman -q

```
root@attackdefense:~/github-repos/ciao# brakeman -q

== Brakeman Report ==

Application Path: /root/github-repos/ciao
Rails Version: 6.0.3.3
Brakeman Version: 4.10.0
Scan Date: 2020-11-16 14:08:11 +0000
Duration: 0.519769373 seconds
Checks Run: BasicAuth, BasicAuthTimingAttack, CSRFTokenForgeryCVE, ContentTag, CookieSerialization, CreateWit
h, CrossSiteScripting, DefaultRoutes, Deserialize, DetailedExceptions, DigestDoS, DynamicFinders, EscapeFunct
ion, Evaluation, Execute, FileAccess, FileDisclosure, FilterSkipping, ForgerySetting, HeaderDoS, I18nXSS, JRu
byXML, JSONEncoding, JSONEntityEscape, JSONParsing, LinkTo, LinkToHref, MailTo, MassAssignment, MimeTypeDoS,
ModelAttrAccessible, ModelAttributes, ModelSerialize, NestedAttributes, NestedAttributesBypass, NumberToCurre
ncy, PageCachingCVE, PermitAttributes, QuoteTableName, Redirect, RegexDoS, Render, RenderDoS, RenderInline, R
```

```
esponseSplitting, RouteDoS, SQL, SQLCVEs, SSLVerify, SafeBufferManipulation, SanitizeMethods, SelectTag, Sele
ctVulnerability, Send, SendFile, SessionManipulation, SessionSettings, SimpleFormat, SingleQuotes, SkipBefore
Filter, SprocketsPathTraversal, StripTags, SymbolDoSCVE, TemplateInjection, TranslateBug, UnsafeReflection, V
alidationRegex, WithoutProtection, XMLDoS, YAMLParsing
```

```
Cross-Site Scripting: 2

== Warnings ==

Confidence: Weak
Category: Cross-Site Scripting
Check: LinkToHref
Message: Potentially unsafe model attribute in `link_to` href
Code: link_to(Check.new.url, Check.new.url)
File: app/views/checks/index.html.erb
Line: 48

Confidence: Weak
Category: Cross-Site Scripting
Check: LinkToHref
Message: Potentially unsafe model attribute in `link_to` href
Code: link_to(Check.new(check_params).url, Check.new(check_params).url)
File: app/views/checks/show.html.erb
Line: 30
```

Warnings Detected

- Cross-Site Scripting

Example 3: Code Triage

Step 1: Navigate to the CodeTriage directory.

Commands:

cd ~/github-repos/CodeTriage
ls

```
root@attackdefense:~/github-repos/ciao# cd ~/github-repos/CodeTriage
root@attackdefense:~/github-repos/CodeTriage# ls
app                  config.ru        Gemfile.lock       log             Procfile.concurrency      README.md
app.json             CONTRIBUTING.md  heroku.yml         package.json    Procfile.development      scratch.rb
bin                  db               ISSUE_TEMPLATE.md  perf.rake       public                    test
CODE_OF_CONDUCT.md   doc              lib                perf_scripts    PULL_REQUEST_TEMPLATE.md  vendor
config               Gemfile          LICENSE.txt        Procfile        Rakefile
root@attackdefense:~/github-repos/CodeTriage#
```

Step 2: Run the brakemen tool in order to identify vulnerabilities in the application.

Command: brakeman -w3 -q

```
root@attackdefense:~/github-repos/CodeTriage# brakeman -w3 -q

== Brakeman Report ==

Application Path: /root/github-repos/CodeTriage
Rails Version: 6.0.3.2
Brakeman Version: 4.10.0
Scan Date: 2020-11-16 14:12:26 +0000
Duration: 1.557348099 seconds
Checks Run: BasicAuth, BasicAuthTimingAttack, CSRFTokenForgeryCVE, ContentTag, CookieSerialization, CreateWit
h, CrossSiteScripting, DefaultRoutes, Deserialize, DetailedExceptions, DigestDoS, DynamicFinders, EscapeFunct
ion, Evaluation, Execute, FileAccess, FileDisclosure, FilterSkipping, ForgerySetting, HeaderDoS, I18nXSS, JRu
byXML, JSONEncoding, JSONEntityEscape, JSONParsing, LinkTo, LinkToHref, MailTo, MassAssignment, MimeTypeDoS,
ModelAttrAccessible, ModelAttributes, ModelSerialize, NestedAttributes, NestedAttributesBypass, NumberToCurre
ncy, PageCachingCVE, PermitAttributes, QuoteTableName, Redirect, RegexDoS, Render, RenderDoS, RenderInline, R
esponseSplitting, RouteDoS, SQL, SQLCVEs, SSLVerify, SafeBufferManipulation, SanitizeMethods, SelectTag, Sele
ctVulnerability, Send, SendFile, SessionManipulation, SessionSettings, SimpleFormat, SingleQuotes, SkipBefore
Filter, SprocketsPathTraversal, StripTags, SymbolDoSCVE, TemplateInjection, TranslateBug, UnsafeReflection, V
alidationRegex, WithoutProtection, XMLDoS, YAMLParsing
```

```
== Warnings ==

Confidence: High
Category: Redirect
Check: Redirect
Message: Possible unprotected redirect
Code: redirect_to(DocMethod.find(params[:id]).to_github)
File: app/controllers/doc_methods_controller.rb
Line: 45

Confidence: High
Category: Redirect
```

```
Check: Redirect
Message: Possible unprotected redirect
Code: redirect_to(IssueAssignment.find(params[:id]).issue.html_url)
File: app/controllers/issue_assignments_controller.rb
Line: 17
```

Warnings Detected

- Redirect

Learnings

Perform Static Code Analysis using the Brakeman tool.

References:

- Autolab (https://github.com/autolab/Autolab.git)
- Ciao (https://github.com/brotandgames/ciao.git)
- Code Triage (https://github.com/codetriage/CodeTriage.git)