Secure By Defaults with Semgrep

Agenda

1. Understanding Secure by Defaults
2. Introduction to Semgrep
3. Semgrep Implementation Process
4. Semgrep Enforcement Process
5. Outcomes and Performance

Secure by Defaults

- A holistic approach to solving security problems at the root cause
- Acts as a scale to reduce the overall harm to a system or type of component
- Secure by Default covers the long-term technical effort to ensure that the right security primitives are built into software and hardware
- It mainly covers the basic principles outlined by OWASP like HTTPS flag, not using weak hashing functions etc.,
- Takes an average of 30 hours to fix a critical vulnerability when not secured by default

OWASP TOP 10	Secure By Default Rules
Injection	• Potential Command injections • Potential XPath injection • Potential SQL injection • Potential code injection • Potential template injection
Broken Authentication	• Missing Anti-CSRF protection
Sensitive Data Exposure	• Leaking of hardcoded credentials
XML External Entities (XXE)	• Potential XXE attack
Broken Access Control	• URL redirects • Weak SSL Contexts
Security Misconfiguration	• Persistent Cookie usage • Potential CORS attack • HTTP secure flag not set • Cookie without HTTPOnly flag
Cross-Site Scripting (XSS)	• Weak XSS protection • Potential XSS attack
Insecure Deserialization	• Struts and Spring files disclosure • HTTP parameters pollution • Potential HTTP Response splitting • Potential deserialization vulnerabilities
Using Components with Known Vulnerabilities	• Weak hash functions • Weak encryptions
Insufficient Logging & Monitoring	• Improper error handling

Semgrep

https://semgrep.dev

Semgrep is a lightweight, offline, open-source, static analysis tool

It supports multiple languages like Go, Java, Python, Ruby and many more

Takes lesser time to scan and reduces false positives

You can use community provided rules or build your custom rules to find vulnerabilities.

Semgrep is also recommended by OWASP

Semgrep vs SAST

- Custom rulesets based on regular expression
- Fewer false positives
- Faster scans
- Mostly up to date

What are the Benefits?

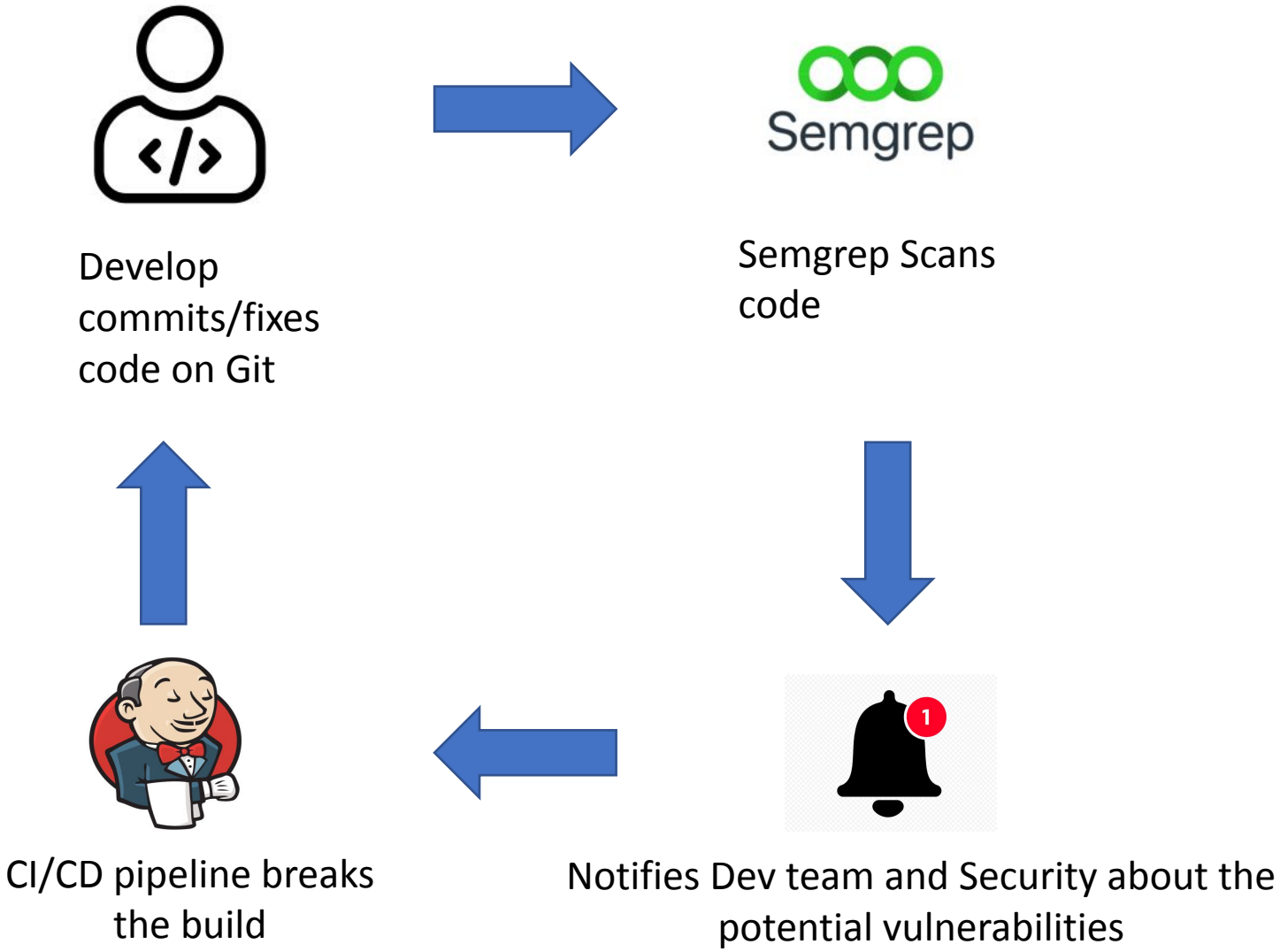
Helps developers reduce the repetitive errors

Custom rulesets which can be tailored to the languages

Implemented on the new code which reduces burden for developers in the future

Focus on remediating vulnerability classes

High level Workflow



IMPLEMENTATION PROCESS

Script to Manage Commits

- Trigger for every commit on GitHub
- Commit Data sent using webhook created per repo for automated response
- Using the Git APIs to get the content of the file/files that was modified
- Formatting the contents to get the code that was modified
- Create a temp file to run Semgrep
- Capture the output for notifying developers
- Format the captured output to send alerts through email and add comments on the PR

Custom Rulesets

- Community rulesets are available based on the language
- Rulesets can be tailored based on the requirements and goals
- Rulesets are written in YAML
- Primarily uses regular expression to detect patterns
- Here is an example of the custom ruleset
- Link to the community rules https://semgrep.dev/r

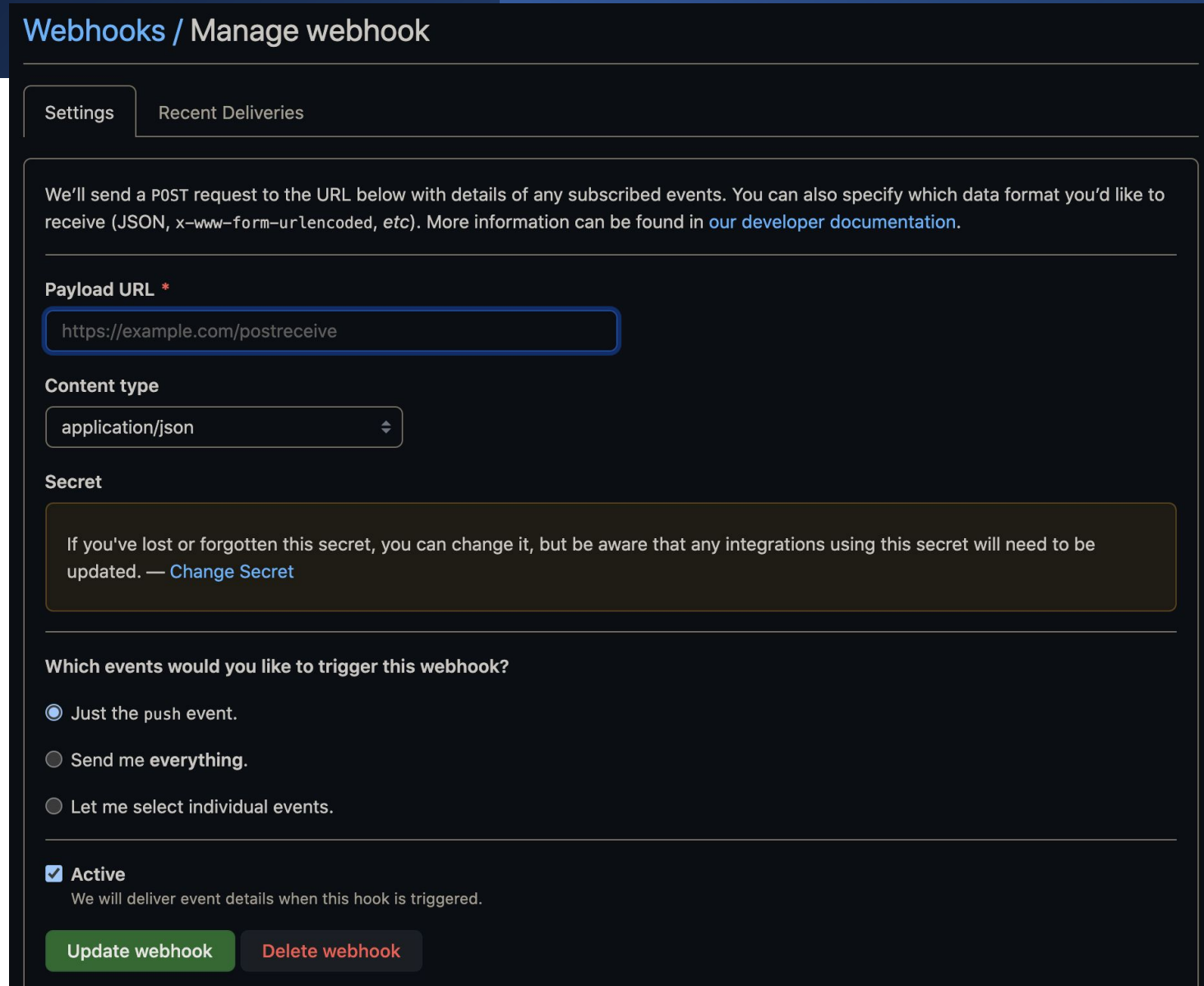
```yaml
- id: missing_httponly_flag
  pattern: $COOKIE.setHttpOnly(false);
  languages: [java]
  message: Missing HTTP only flag at this line
  fix: $COOKIE.setHttpOnly(true);
  severity: WARNING
```

Containerizing the Script

- Scaling up to manage the number outgoing commits every minute
- Makes debugging easier with logging
- Helps in modifying the script to add functionality
- Load balancing is efficient since the script needs to be up and listening
- Isolated environments

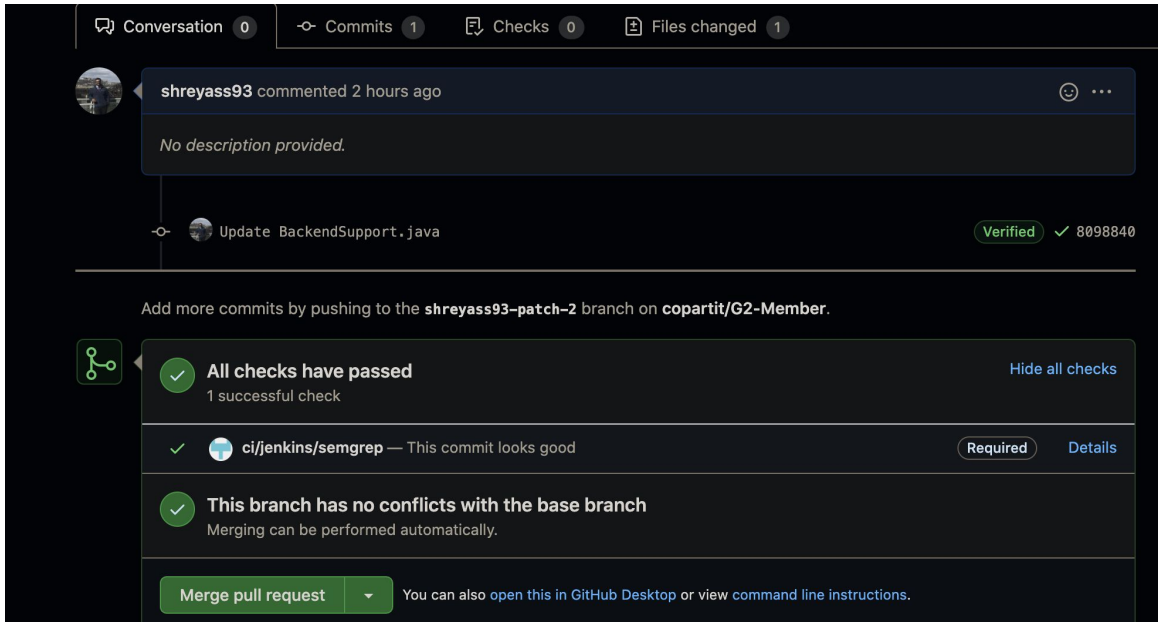
Webhook Creation

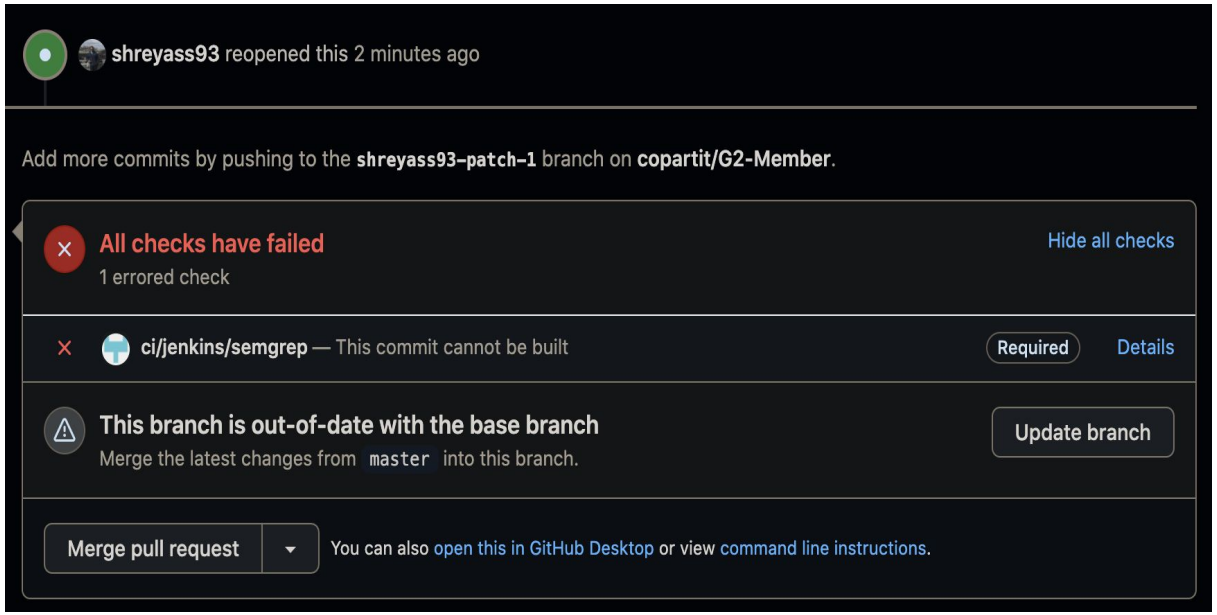
- Using URL from the deployed script to create webhooks
- Webhooks can be created based on the language
- They in turn trigger the script which continues the cycle
- This must be added to every repo to get the alerts



CI/CD Integration

- Semgrep offers CLI support which can be directly integrated with the pipeline
- Add the semgrep command as part of requirements within pipeline automation
- Semgrep can be added as status check for the Pull Requests
- Vulnerable code won’t move forward in pipeline unless they are fixed





Snippet of our Implementation

Search or jump to... / Pull requests Issues Marketplace Explore

copartit / Github-guardian Private

Watch 2 Star 0 Fork 1

Code Issues Pull requests 2 Actions Projects Wiki Security Insights Settings

master Github-guardian / alertTest Go to file

shreyass93 Update alertTest Latest commit 5a72db2 2 hours ago History

1 contributor

272 lines (234 sloc) 8.91 KB Raw Blame

```
package com.copart.g2.seller.authentication;



import java.util.List;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import com.copart.g2.seller.login.service.IABPoolService;
import com.copart.g2.seller.preferences.service.IUserPreferenceService;
import com.copart.g2.seller.preferences.util.JSONMapper;
import com.copart.g2.seller.preferences.vo.PreferenceQuery;
import com.copart.g2.seller.properties.EnvironmentProp;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.authentication.BadCredentialsException;
import org.springframework.security.authentication.UsernamePasswordAuthenticationToken;
import org.springframework.security.core.Authentication;
import org.springframework.security.core.AuthenticationException;
import org.springframework.security.core.context.SecurityContextHolder;
import org.springframework.stereotype.Component;
import org.springframework.web.context.request.RequestContextHolder;
```

Enforcing Secure by Default

- Semgrep is integrated to our Build tool (jenkins)
- Semgrep scans the code as part of prebuild process

Jenkins	Update semgrep.groovy
src	Merge branch 'master' into mds-develop-release-3.26
README.md	Update README.md
owner.md	Create owner.md
pom.xml	Purge CSS plugin

```
            }
        steps {
          script {
                sh "cd ${app_workspace}"
                def target_commit = sh(returnStdout: true, script: "git rev-parse origin/${env.CHANGE_TARGET}").trim()
                // print files changed
                sh "git diff --diff-filter=ACM --name-only ${target_commit} HEAD"
                sh "git diff --diff-filter=ACM --name-only ${target_commit} HEAD | xargs docker run -v ${env.workspace}:/src returntocorp/semgrep:latest --config https://semgrep.dev/s/shreyass93:copartjavarules --error --lang
java --verbose"          }
        }
    }
```

Pull Request: PR-10631 | 1m 4s | No changes

Commit: aa899ef | 19 hours ago | Pull request #10631 opened

Start — Checkout — semgrep — End

semgrep - 19s

- Maven 3.3.3 — Use a tool from a predefined Tool Installation <1s
- Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step. <1s
- OpenJDK 11.0.6 — Use a tool from a predefined Tool Installation <1s
- Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step. <1s
- cd . — Shell Script <1s
- git rev-parse origin/mds-develop-release-3.26 — Shell Script <1s
- git diff --diff-filter=ACM --name-only b9e556403f2de89f69fcd652c238df06c237d89f HEAD — Shell Script <1s
- ✕ git diff --diff-filter=ACM --name-only b9e556403f2de89f69fcd652c238df06c237d89f HEAD | xargs docker run -v /opt/copart/node/workspace/ecops-semgrep_G2-Member_PR-10631:/src returntocorp/semgrep:latest --config https://semgre... — Shell Script 18s
- Delete workspace when build is done <1s
- Recursively delete the current directory from the workspace <1s

```
[2021-09-29T20:08:46.652Z] severity:warning rule:improper_error_handling: Error handling is also important from an intrusion detection perspective. Certain attacks against your application may trigger errors whi
can help detect attacks in progress
[2021-09-29T20:08:46.652Z]
[2021-09-29T20:08:46.652Z] autofix: log.error(msg, exception);
[2021-09-29T20:08:46.652Z] 123:                e.printStackTrace();
[2021-09-29T20:08:46.652Z]
[2021-09-29T20:08:46.652Z] src/main/java/com/copart/g2/member/exception/handlers/webservice/CreatePasswordWSExceptionHandler.java
[2021-09-29T20:08:46.652Z] severity:warning rule:improper_error_handling: Error handling is also important from an intrusion detection perspective. Certain attacks against your application may trigger errors whi
can help detect attacks in progress
[2021-09-29T20:08:46.652Z]
[2021-09-29T20:08:46.652Z] autofix: log.error(msg, exception);
[2021-09-29T20:08:46.652Z] 48:                e.printStackTrace();
[2021-09-29T20:08:46.652Z]
[2021-09-29T20:08:46.652Z] src/main/java/com/copart/g2/member/exception/handlers/webservice/SecurityQuestionsWSExceptionHandler.java
[2021-09-29T20:08:46.653Z] severity:warning rule:improper_error_handling: Error handling is also important from an intrusion detection perspective. Certain attacks against your application may trigger errors whi
can help detect attacks in progress
[2021-09-29T20:08:46.653Z]
[2021-09-29T20:08:46.653Z] autofix: log.error(msg, exception);
[2021-09-29T20:08:46.653Z] 49:                e.printStackTrace();
[2021-09-29T20:08:46.653Z]
```

Outcomes and Performance

- 45% reduction in tickets within past 6 months
- About 60 potential vulnerabilities blocked in the past 6 months
- Early detection and rectification
- Better awareness among the new developers
- Secure coding practices
- Processes close to 1800 lines of code per second

What are the Challenges?

CONTEXT OF THE COMMITTED CODE

MANAGING RULESETS OF MULTIPLE LANGUAGES

CREATING AND ADDING WEBHOOKS

CREATING AWARENESS AMONG DEV TEAMS

What are the Gaps?

- Supports only the major languages
- Business logic flaws cannot be handled
- Affects the pipeline when Semgrep is down
- Pipeline automation script must be added/updated on every repo

Next steps

Adding Rulesets

- Replace HTTP URLs with HTTPS on new commits
- Find unauthenticated routes on new commits
- Disallow URLs with usernames and passwords
- Extension to other languages

Questions

How is it Implemented?

- Semgrep is initially implemented with alert mode

Search or jump to... / Pull requests Issues Marketplace Explore

copartit / Github-guardian Private

Watch 2 Star 0 Fork 1

Code Issues Pull requests 2 Actions Projects Wiki Security Insights Settings

master Github-guardian / alertTest Go to file

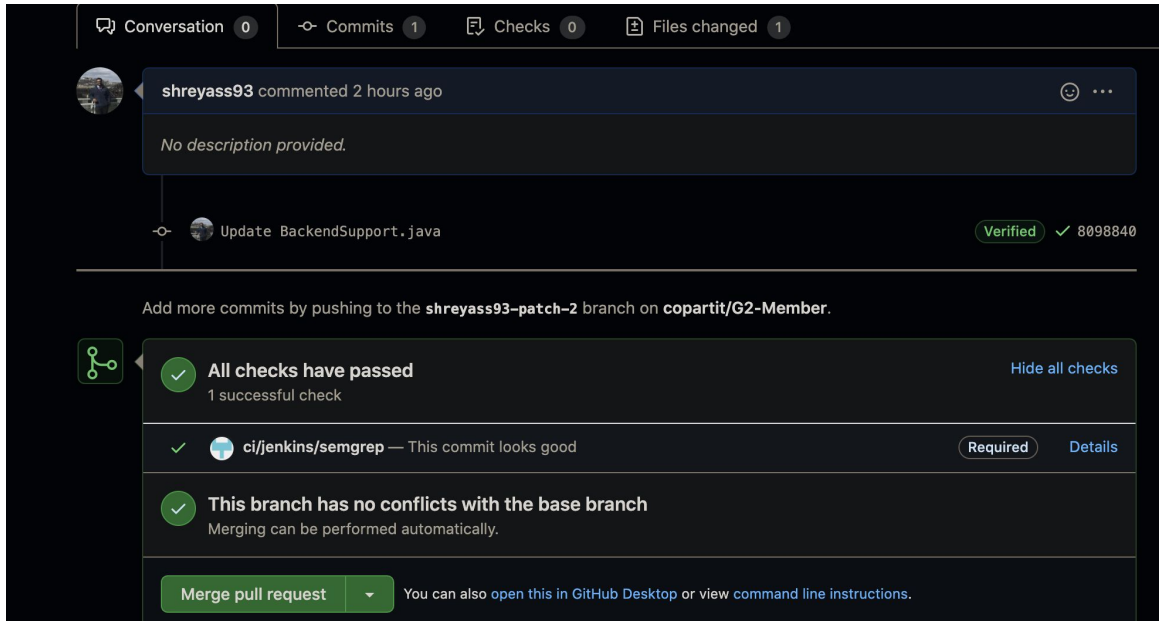
shreyass93 Update alertTest Latest commit 5a72db2 2 hours ago History

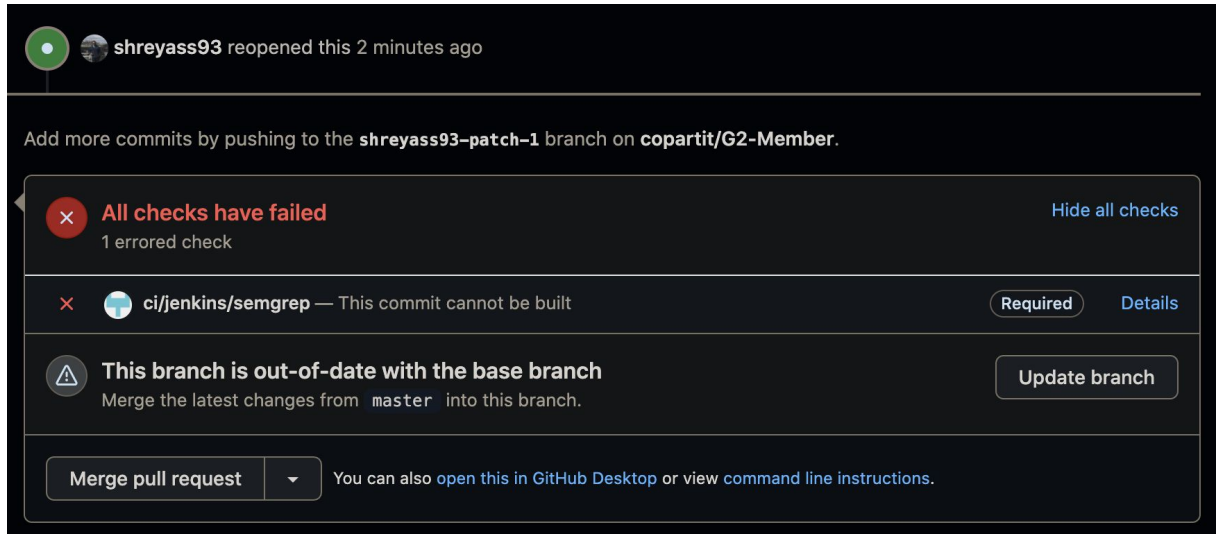
1 contributor

272 lines (234 sloc) 8.91 KB Raw Blame

```
package com.copart.g2.seller.authentication;



import java.util.List;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import com.copart.g2.seller.login.service.IABPoolService;
import com.copart.g2.seller.preferences.service.IUserPreferenceService;
import com.copart.g2.seller.preferences.util.JSONMapper;
import com.copart.g2.seller.preferences.vo.PreferenceQuery;
import com.copart.g2.seller.properties.EnvironmentProp;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.authentication.BadCredentialsException;
import org.springframework.security.authentication.UsernamePasswordAuthenticationToken;
import org.springframework.security.core.Authentication;
import org.springframework.security.core.AuthenticationException;
import org.springframework.security.core.context.SecurityContextHolder;
import org.springframework.stereotype.Component;
import org.springframework.web.context.request.RequestContextHolder;
```

- Semgrep is enforced as a PR check which stops the code from merging if there are any errors
- It is only done on master branches for now





Why do we need Semgrep?

- Semgrep helps developers reduce the repetitive errors by alerting them early in the pipeline
- Custom rulesets which are tailor made to Copart repos and are also based on previous tickets found within the repo
- Semgrep is only implemented on the new outgoing code which means it reduces burden for developers in the future
- Therefore, Semgrep is way efficient and faster than it’s counterparts like Checkmarx and Sonarqube and helps mitigate vulnerability classes

Semgrep is a lightweight, offline, open-source, static analysis tool

It supports multiple languages like Go, Java, Python, Ruby and many more

Takes lesser time to scan and reduces false positives

You can use community provided rules or build your custom rules to find vulnerabilities.