

Laboratoire 2: Tests de pénétration et Outils pour Intégration et Déploiement continus



LOG8100: DEVSECOPS

2323809 – Zhibo Zhang

2412685 – Mathéo Mercier

2410029 – Hoang Thuan Pham

2085581 – Maxence Lefebvre

Date de Remise : 24 Octobre 2024

Automne 2024

Polytechnique Montréal

GitHub Pages : https://mlevrais.github.io/LOG8100_TP2/

1. Liens

- docker hub : https://hub.docker.com/repository/docker/jesusles/log8100_tp1
- github : https://github.com/mlevrais/LOG8100_TP2
- github pages : https://mlevrais.github.io/LOG8100_TP2/
-

2. Introduction

Ce compte-rendu a pour objectif de présenter les travaux effectués par notre équipe pour répondre au sujet du TP2 du cours LOG8100.

L'application analysée est Damn Vulnerable NodeJS Application (DVNA). Cette application est développée avec Node.js pour le front-end et PostgreSQL pour la base de données. Elle utilise des bibliothèques fréquemment utilisées dans l'industrie comme Express, Passport, et Sequelize.

Les utilisateurs peuvent s'inscrire et se connecter au site pour lire l'introduction brève de chaque vulnérabilité dans OWASP TOP 10 (2017). Nous avons effectué un scan des vulnérabilités contre cette application avec OWASP ZAP, un scanneur web open-source.

Pour démarrer l'application, on doit configurer le docker et définir les variables d'environnement. Premièrement, on a changé la base de données à postgresQL, les variables d'environnement sont aussi données (le nom d'utilisateur, le port, le mot de passe). Voici le fichier *docker-compose.yml*:

```
services:
  app:
    build:
      context: ./
      dockerfile: Dockerfile
    ports:
      -
    volumes:
      - ./app
    depends_on:
      - db
    environment:
      POSTGRES_USER: db           # Assigning user directly
      POSTGRES_PASSWORD: db      # Assigning password directly
      POSTGRES_DB: db            # Name of the database
```

```

        POSTGRES_PORT: 5432          # Port on which PostgreSQL
listens

    db:
        image: postgres:13
        ports:
            -
        volumes:
            - pgdata:/var/lib/postgresql/data
        environment:
            POSTGRES_USER: db          # Assigning user directly
            POSTGRES_PASSWORD: db     # Assigning password directly
            POSTGRES_DB: db           # Name of the database
            POSTGRES_PORT: 5432       # Port on which PostgreSQL
listens

        volumes:
            pgdata:

```

En plus, pour que la version de node et le système Ubuntu soient compatible avec les modules utilisé dans l'application (bcrypt et libxmljs). On a composé un docker avec node.js 8 basé sur une image Ubuntu 20. Voici le fichier *Dockerfile*:

```

# Damn Vulnerable NodeJS Application
# https://github.com/appsecco/dvna

```

```

FROM          :

# Install dependencies
RUN           && \
            \
            \
            &&

# Install nvm (Node Version Manager)
ENV
RUN

# Load nvm and install Node.js 8.x
RUN          $

# Add Node.js and npm to the PATH

```

```

ENV          =                               $

# Verify Node.js and npm versions
RUN          &&                               &&

# Set working directory
WORKDIR

# Set up PATH for nvm and Node.js
ENV          $

CMD

```

On a aussi modifier les deux fichier .sh pour qu'ils soient compatibles avec les deux docker container. Voici le fichier *entrypoint.sh*:

```

#!/bin/bash

chmod

/bin/bash -t 300 --

Voici le fichier
#!/bin/bash

npm
npm

```

À la fin, on a aussi mis un fichier db.js qui indique à l'application les informations de db. Voici le fichier *db.js*:

```

.          = {
    .      .POSTGRES_USER,
    .      .POSTGRES_PASSWORD,
    .      .POSTGRES_DB,
    .      .POSTGRES_HOST || ,
    .      .POSTGRES_PORT || 5432,
}

```

3. Analyse de sécurité

D'après le scan de OWASP ZAP, le rapport de ZAP est attaché dans l'annexe, 8 vulnérabilités sont trouvées. Elles sont résumées dans la table suivante:

#	Type	Gravité	Localisation	OWASP TOP10
1	CSP: Wildcard Directive	Medium	./robots.txt ./sitemap.xml	A1:2017-Injection A6:2017-Security Misconfiguration A7:2017-Cross-Site Scripting (XSS)
2	Content Security Policy (CSP) Header Not Set	Medium	./forgotpw ./learn ./learn/vulnerability/a1_injection ./learn/vulnerability/a2_broken_auth ./learn/vulnerability/a3_sensitive_data ./learn/vulnerability/a4_xxe ./learn/vulnerability/a5_broken_access_control ./login ./register	A1:2017-Injection A6:2017-Security Misconfiguration A7:2017-Cross-Site Scripting (XSS)
3	Missing Anti-clickjacking Header	Medium	./forgotpw ./learn ./learn/vulnerability/a1_injection ./learn/vulnerability/a2_broken_auth ./learn/vulnerability/a3_sensitive_data ./learn/vulnerability/a4_xxe ./learn/vulnerability/a5_broken_access_control ./login ./register	A6:2017-Security Misconfiguration A7:2017-Cross-Site Scripting (XSS)
4	Vulnerable JS Library	Medium	./assets/jquery-3.2.1.min.js	CVEs: A9:2017 - Using Components with Known Vulnerabilities CVE-2020-11023 and CVE-2020-11022: A7:2017-Cross-Site Scripting (XSS) CVE-2019-11358: A6:2017-Security Misconfiguration and A5:2017-Broken Access Control
5	Cookie without SameSite Attribute	Low	./login ./robots.txt ./sitemap.xml	A5:2017 - Broken Access Control A6:2017 - Security Misconfiguration
6	Cross-Domain JavaScript Source File Inclusion	Low	./forgotpw ./learn ./learn/vulnerability/a1_injection ./learn/vulnerability/a2_broken_auth	A1:2017-Injection A7:2017-Cross-Site Scripting (XSS)

			auth ./learn/vulnerability/a3_sensitive_data ./learn/vulnerability/a4_xxe ./learn/vulnerability/a5_broken_access_control ./login ./register	
7	Server Leaks Information via "X-Powered-By" HTTP Response	Low	./ ./assets/fa/css/font-awesome.min.css ./assets/jquery-3.2.1.min.js ./assets/showdown.min.js ./forgotpw ./learn ./learn/vulnerability/a10_logging ./learn/vulnerability/a1_injection ./learn/vulnerability/a2_broken_auth ./learn/vulnerability/a3_sensitive_data ./learn/vulnerability/a4_xxe ./learn/vulnerability/a5_broken_access_control ./learn/vulnerability/a6_sec_misconf ./learn/vulnerability/a7_xss ./learn/vulnerability/a8_ids ./learn/vulnerability/a9_vuln_component ./learn/vulnerability/ax_csrf ./learn/vulnerability/ax_redirect ./login ./logout ./register ./robots.txt ./sitemap.xml ./forgotpw ./login ./register	A3:2017-Sensitive Data Exposure
8	X-Content-Type-Options Header Missing	Low	./assets/fa/css/font-awesome.min.css ./assets/jquery-3.2.1.min.js ./assets/showdown.min.js ./forgotpw ./learn ./learn/vulnerability/a1_injection ./learn/vulnerability/a2_broken_auth ./learn/vulnerability/a3_sensitive_data ./learn/vulnerability/a4_xxe ./learn/vulnerability/a5_broken_access_control ./login ./register	A6:2017 - Security Misconfiguration

3.1. Description des vulnérabilités

#1(Medium). CSP: Wildcard Directive:

wildcard

#2(Medium). Content Security Policy (CSP) Header Not Set: le site web n'a pas de CSP via les en-têtes HTTP. Cela signifie que la CSP n'est pas activée sur le site web. La conséquence est la même que la vulnérabilité #1.

#3(Medium). Missing Anti-clickjacking Header: Le site web ne met pas en place de protections contre les attaques de clickjacking. Dans une attaque de clickjacking, l'attaquant incite un utilisateur à cliquer sur un élément malveillant de page Web qui est invisible ou déguisé en un autre élément.

#4(Medium). Vulnerable JS Library: Le .js script [/assets/jquery-3.2.1.min.js](#) a utilisé la librairie vulnérable, library jquery de version 3.2.1. Il contient 3 CVEs: CVE-2020-11023, CVE-2020-11022, et CVE-2019-11358. La CVE-2020-11023 et la CVE-2020-11022 permettent de passer du code HTML contenant des éléments `<option>` même après les avoir nettoyés. Du code non fiable peut être injecté à l'une des méthodes de manipulation DOM de jQuery (e.g., `.html()`, `.append()`, etc.) pour exécuter. La CVE-2019-11358 permet à tout utilisateur de prolonger le `Object.prototype` si un objet source non assaini contient une propriété `__proto__`. Donc un attaquant peut injecter du code malveillant dans cet objet prolongé.

#5(Low). Cookie without SameSite Attribute: Les cookies n'ont pas d'attribut `SameSite` ce qui veut dire qu'ils peuvent être envoyés lors de cross-site requests ce qui permet à un attaquant d'exécuter des attaques CSRF, XSS ou des attaques temporelles.

#6(Low). Cross-Domain JavaScript source file inclusion: Le site web utilise un script qui provient d'un domaine tiers. Cela veut dire que si un attaquant compromet ce script, l'exécution de code malveillant sur le navigateur d'une victime ainsi que des fuites de données sont possibles.

#7(Low). Server Leaks Information via "X-Powered-By" HTTP Response: Lors de réponse par HTTP, le serveur révèle des informations sur lui-même. Que ce soit des logiciels ou des composants qu'il utilise. Cela peut ainsi augmenter la surface d'attaque.

#8(Low). X-Content-Type-Options Header Missing: Avec l'en-tête manquante, des anciens navigateurs Internet Explorer et Chrome vont tenter de deviner en sniffant la réponse ce qui peut finir par un mauvais affichage du contenu et révéler des informations sensibles ou bien exécuter du code malicieux.

3.2. Stratégies de mitigation

#1(Medium). CSP: Wildcard Directive: 1. Utiliser des sources spécifiques et de confiance au lieu de * dans la configuration CSP. 2. Il faut aussi vérifier le contenu dans le paquet venant de l'utilisateur.

#2(Medium). Content Security Policy (CSP) Header Not Set: 1. Les propriétaires de sites web doivent définir des en-têtes CSP appropriés. En plus, comme indiqué pour la vulnérabilité #1, il ne faut pas donner une CSP très large. 2. Il faut aussi vérifier le contenu dans le paquet venant de l'utilisateur.

#3(Medium). Missing Anti-clickjacking Header: Comme le nom a indiqué, s'il manque en-tête anti-clickjacking, il faut en mettre. Par exemple, 1. on peut mettre en place une CSP comme: *Content-Security-Policy: frame-ancestors 'self'*; Elle permet uniquement le contenu du même domaine donc les contenus malveillants dans d'autres domaines sont exclus. 2. Ou on peut aussi avoir la même effet avec l'en-tête anti-clickjacking: *X-Frame-Options: SAMEORIGIN*.

#4(Medium). Vulnerable JS Library: Il faut mettre à jour les librairies.

#5(Low). Cookie without SameSite Attribute: Il faut ajouter un attribut SameSite aux cookies. Il est préférable que l'attribut soit mis à "strict" pour empêcher les attaques lorsque possibles et sinon à "lax".

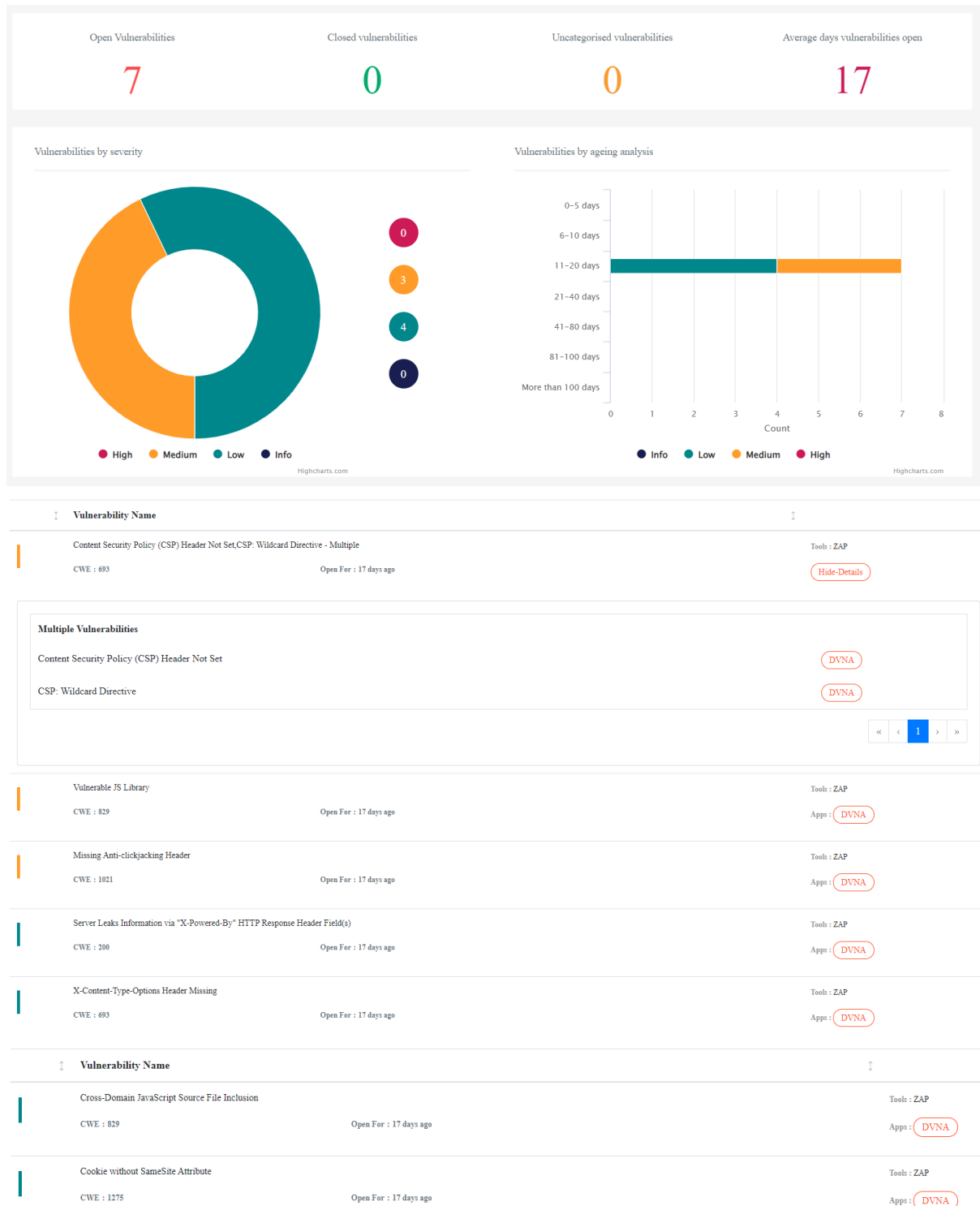
#6(Low). Cross-Domain JavaScript source file inclusion: Implémentez une Content Security Policy ne permettant l'utilisation de script uniquement de domaine fiables, l'utilisation de check sur l'intégrité du script qui est utilisé avec une comparaison à un hash ou bien l'arrêt d'utilisation de script provenant de sources tierce sont des stratégies pour régler la vulnérabilité.

#7(Low). Server Leaks Information via "X-Powered-By" HTTP Response: Trouver et modifier les fichiers de configuration pour enlever ou modifier l'en-tête "X-Powered-By". Les fichiers se trouvent à différents endroits dépendamment du serveur utilisé.

#8(Low). X-Content-Type-Options Header Missing: Il suffit de rajouter l'en-tête avec la propriété "nosniff".

3.3. Rapport de Test avec Orchestron

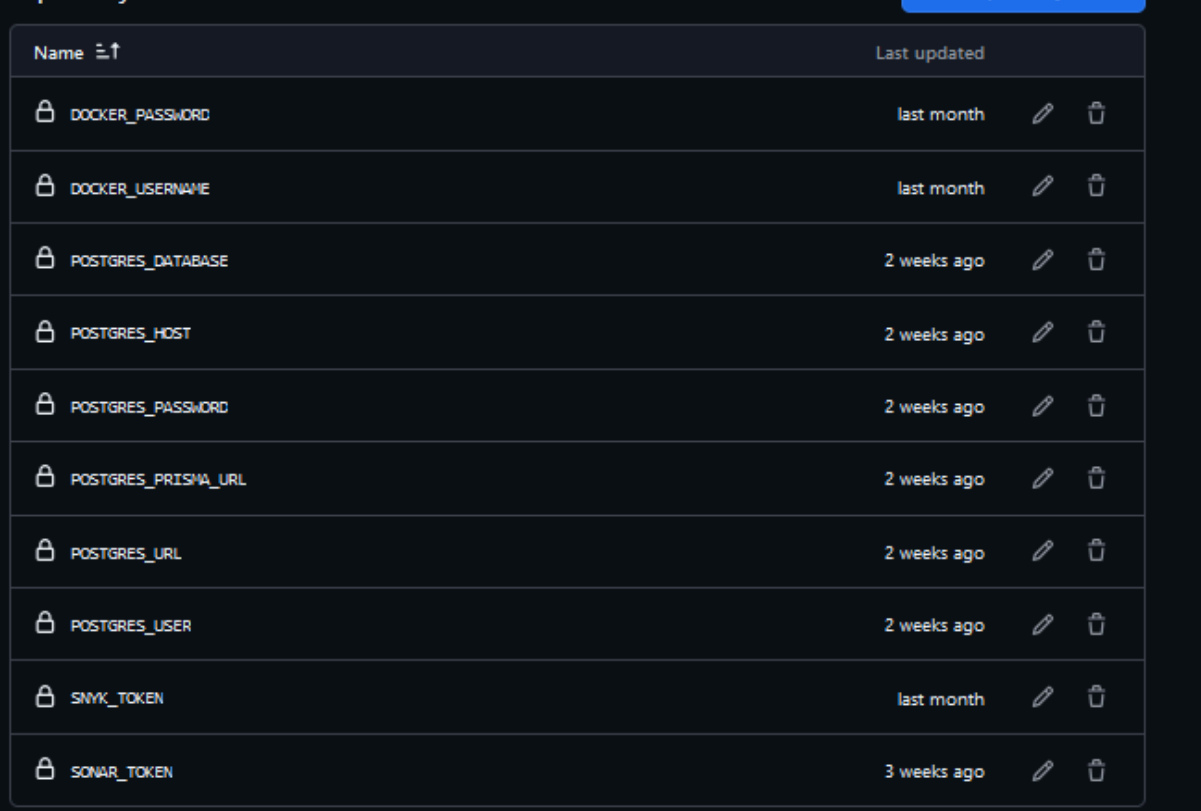
Orchestron permet de visualiser les rapports obtenus par différentes applications telles que ZAP, BurpSuite ou Bandit. Nous avons donc envoyé le fichier xml du rapport créé par ZAP à des fins de visualisation. Nous pouvons ainsi voir que les vulnérabilités rapportées par ZAP sont au nombre de 7 avec une qui à un tag multiple faisant références à deux vulnérabilités.



4. Implémentation d'une interface de CI/CD

Configuration des secrets

Nous configurons manuellement les différents secrets nécessaires pour ne pas les écrire directement dans le code. A ce jour voici la liste des secrets intégrés dans les github secrets:

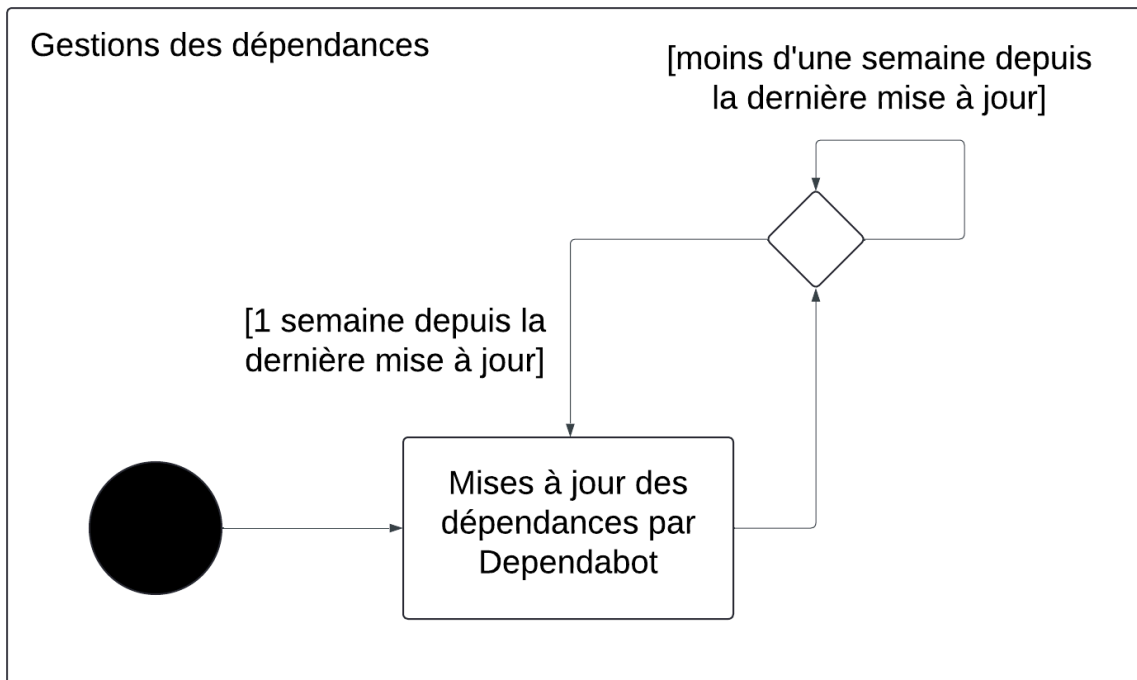
A screenshot of the GitHub Secrets interface. It shows a table with two columns: 'Name' and 'Last updated'. There are 10 rows of secrets listed. Each row has a lock icon, the secret name, the last updated time, and edit/delete icons.

Name	Last updated
DOCKER_PASSWORD	last month
DOCKER_USERNAME	last month
POSTGRES_DATABASE	2 weeks ago
POSTGRES_HOST	2 weeks ago
POSTGRES_PASSWORD	2 weeks ago
POSTGRES_PRISMA_URL	2 weeks ago
POSTGRES_URL	2 weeks ago
POSTGRES_USER	2 weeks ago
SNYK_TOKEN	last month
SONAR_TOKEN	3 weeks ago

Description du pipeline d'intégration continue (CI)

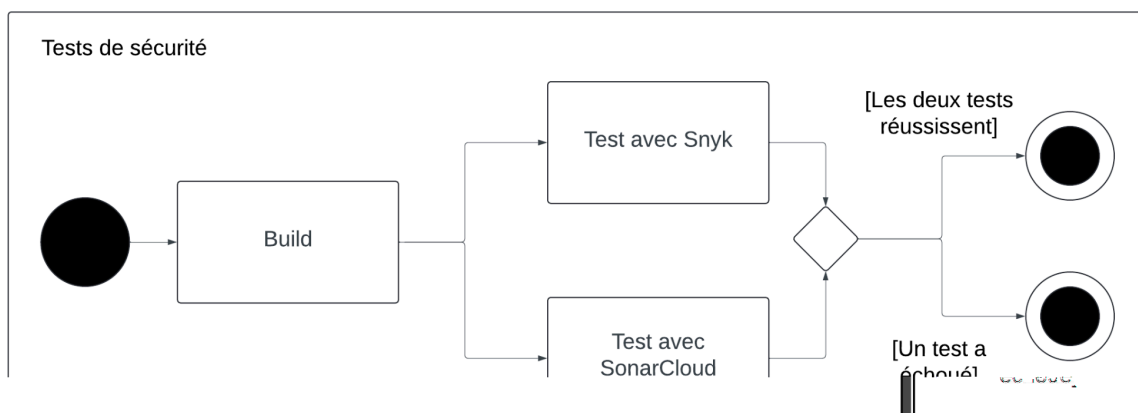
Notre pipeline de CI est composé de 2 parties: Un pipeline pour les gestions des dépendances et un autre pour les tests de sécurité.

Pour le pipeline de gestion des dépendances, on a utilisé Dependabot. C'est un logiciel pour détecter et résoudre les vulnérabilités dans les dépendances des projets. Chaque semaine, Dependabot cherche des mises à jour pour les dépendances du projet et crée des pull requests pour les mises à jour automatiquement.



Pour le pipeline de tests de sécurité, on a utilisé Snyk et SonarCloud. On a choisi SonarCloud à la place de SonarQube car ils offrent les mêmes fonctionnalités, mais il est plus facile d'intégrer SonarCloud dans le pipeline parce que le service est toujours disponible dans le Cloud.

Premièrement, on installe les packages dépendantes avec npm (l'étape **Build**). C'est requis par Snyk parce que certaines dépendances ne sont résolues qu'après que le projet est installé. Puis, on fait les analyses avec Snyk et SonarCloud en parallèle (les étapes **Test**). Snyk échouera s'il trouve des vulnérabilités dans le code. Pour SonarCloud, le job passera toujours. On devrait voir le rapport dans le dashboard SonarCloud pour voir l'analyse complète.



Créations d'image vers le docker hub

Pour faire cela, on s'intéresse à nouveau aux github actions. On crée une github action qui s'exécute après chaque push dans la branche releases. Pour cela on doit d'abord se connecter au github hub, on utilise donc les secrets qu'on a configurés auparavant. Ensuite on build & push l'image avec une action. Et éventuellement on ajoute les métadonnées de l'image pour l'action.

```
name:

on:
  push:
    branches:

jobs:
  push_to_registry:
    name:
    runs-on:

    steps:
      - name:
        uses:

      - name:
        uses:
        with:
          username:
          password:

      - name:
        id:
        uses:

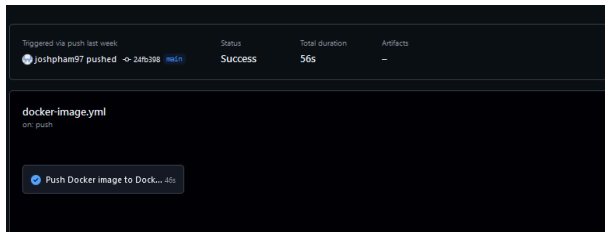
        with:
          images:

      - name:
        id:
        env:
          POSTGRES_USER:
          POSTGRES_HOST:

          POSTGRES_PASSWORD:
          POSTGRES_DATABASE:
        uses:
```

```
with:
  context:
  file:
  push: true
  tags:
  labels:
```

On vérifie que tout s'exécute correctement :



Publication automatique vers github pages

La publication du github pages est assez simple. On pourrait choisir d'implémenter une github actions, mais nous avons choisi de publier directement de publier à partir de branche main. Donc la page sera à jour uniquement lors d'un push sur la branche main. En réalité il y a bien une github action qui s'exécute simplement on a pas de code .yml. Pour le compte rendu, nous avons créé un pdf dans le dossier /docs/CR qui est affiché par le fichier html du github pages.

Pipeline CD

Pour le pipeline CD nous avons 2 choses :

Premièrement, nous avons choisis de déployer la base de données postgresQL en utilisant Vercel. Au bout de 15 jours nous sommes encore loin de la limite gratuite :

Status	Region	Endpoint	Storage Size	Compute Time
Available	Washington, D.C., USA iad1	ep-lively-shape-a4xbenvr-pooler	31 MB/256 MB	0.30 hrs/60 hrs
Quickstart				

L'application postgre est toujours disponible, l'application web s'y connecte à chaque déploiement en utilisant notamment des github secrets.

Deuxièmement, nous n'avons pas trouvé d'hébergeur gratuit sans carte bleue pour héberger l'application. L'application est donc a build en local ..

5. Conclusion

Pour sécuriser l'application, la Content Security Policy (CSP) doit être bien configurée. Cela aide à bloquer les attaques comme l'injection de code et les attaques XSS. D'un autre côté, il faut toujours mettre à jour les bibliothèques JavaScript, comme jQuery, pour éviter que des pirates exploitent des failles connues. Ces actions renforcent la sécurité du site en réduisant les risques d'attaques.

Les github actions permettent vraiment de sécuriser et d'automatiser de déploiement continu/intégration continue. Elles sont simples à utiliser et bien documentées et il y a des actions préparées pour de nombreuses solutions.

Annexe:

TP2 ZAP Scanning Report

Site: <http://localhost:3000>

Generated on Sat, 5 Oct 2024 20:16:09

ZAP Version: 2.15.0

ZAP is supported by the [Crash Override Open Source Fellowship](#)

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	4
Low	4
Informational	4

Alerts

Name	Risk Level	Number of Instances
CSP: Wildcard Directive	Medium	2
Content Security Policy (CSP) Header Not Set	Medium	9
Missing Anti-Flickering Header	Medium	9
Vulnerable JS Library	Medium	1
Cookie without SameSite Attribute	Low	3
Cross-Domain Authentication Source File Inclusion	Low	27
Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	Low	26
X-Content-Type-Options Header Missing	Low	12
Authentication Request Identified	Informational	1
Information Disclosure - Suspicious Comments	Informational	3
Logout - Logout Cookie	Informational	4
Session Management Response Identified	Informational	6

Alert Detail

Medium	CSP: With Directive
Description	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including (but not limited to) Cross Site Scripting (XSS), and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.
URL	http://localhost:3000/index.js
Method	GET
Parameter	Content-Security-Policy
Attack	
Evidence	default-src 'none'
Other Info	The following directives either allow wildcard sources (or ancestors), are not defined, or are overly broadly defined: frame-ancestors, form-action The directive(s): frame-ancestors, form-action are among the directives that do not fallback to default-src, missing/including them is the same as allowing anything.
URL	http://localhost:3000/index.js?rand
Method	GET
Parameter	Content-Security-Policy
Attack	
Evidence	default-src 'none'
Other Info	The following directives either allow wildcard sources (or ancestors), are not defined, or are overly broadly defined: frame-ancestors, form-action The directive(s): frame-ancestors, form-action are among the directives that do not fallback to default-src, missing/including them is the same as allowing anything.
Instances	2
Solution	Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header. https://www.w3.com/TSV/SPW https://hackerone.com/intercepting-content-security-policy https://content-security-policy.com/ https://github.com/mozilla/content-security https://devdocs.io/html/headers/content-security-policy and/or to a wide variety of resources
Reference	
CWE ID	595
WASC ID	15
Plugin ID	10255
Medium	Content Security Policy (CSP) Header Not Set
Description	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.
URL	http://localhost:3000/index.js
Method	GET
Parameter	
Attack	
Evidence	
Other Info	
URL	http://localhost:3000/team
Method	GET
Parameter	
Attack	
Evidence	
Other Info	
URL	http://localhost:3000/team/vulnability?id1_injection
Method	GET
Parameter	
Attack	
Evidence	
Other Info	
URL	http://localhost:3000/team/vulnability?id2_broken_auth
Method	GET
Parameter	
Attack	
Evidence	
Other Info	
URL	http://localhost:3000/team/vulnability?id3_sensitive_data
Method	GET
Parameter	

Attack	
Evidence	
Other Info	
URL	http://localhost:3000/team/vulnerability/cv1_xss
Method	GET
Parameter	
Attack	
Evidence	
Other Info	
URL	http://localhost:3000/team/vulnerability/cv5_broken_access_control
Method	GET
Parameter	
Attack	
Evidence	
Other Info	
URL	http://localhost:3000/login
Method	GET
Parameter	
Attack	
Evidence	
Other Info	
URL	http://localhost:3000/vulnerable
Method	GET
Parameter	
Attack	
Evidence	
Other Info	
Instances	9
Solution	<p>Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.</p> <p>https://developer.mozilla.org/en-US/docs/Web/Security/CSP/Introducing_Content_Security_Policy</p> <p>https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html</p> <p>https://www.cisploit.org/</p> <p>https://hackerone.com/bug-bounty/csp</p> <p>https://web.archive.org/web/20190901000000/http://www.vulnerability-lab.com/vulnerability-lab.php</p> <p>https://www.vulnerability-lab.com/vulnerability-lab.php</p> <p>https://www.vulnerability-lab.com/vulnerability-lab.php</p>
Reference	
CWE ID	603
WASC ID	15
Plugin ID	10038
Medium	Mixing Anti-clickjacking Header
Description	The response does not protect against 'Clickjacking' attacks. It should include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options.
URL	http://localhost:3000/home
Method	GET
Parameter	x-frame-options
Attack	
Evidence	
Other Info	
URL	http://localhost:3000/team
Method	GET
Parameter	x-frame-options
Attack	
Evidence	
Other Info	
URL	http://localhost:3000/team/vulnerability/cv1_injection
Method	GET
Parameter	x-frame-options
Attack	
Evidence	
Other Info	
URL	http://localhost:3000/team/vulnerability/cv2_broken_auth
Method	GET
Parameter	x-frame-options
Attack	
Evidence	
Other Info	
URL	http://localhost:3000/team/vulnerability/cv3_sensitive_data
Method	GET
Parameter	x-frame-options
Attack	
Evidence	
Other Info	
URL	http://localhost:3000/team/vulnerability/cv4_xss
Method	GET
Parameter	x-frame-options
Attack	
Evidence	
Other Info	
URL	http://localhost:3000/team/vulnerability/cv5_broken_access_control
Method	GET
Parameter	x-frame-options
Attack	
Evidence	
Other Info	
URL	http://localhost:3000/login
Method	GET
Parameter	x-frame-options
Attack	
Evidence	
Other Info	
URL	http://localhost:3000/vulnerable
Method	GET
Parameter	x-frame-options

Compte-rendu LOG8100 - TP2 - Groupe 8 - 2024/09/26
page.17

Attack	<script type="text/javascript" src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
Evidence	
Other Info	
URL	http://localhost:3000/team/vulnecab/thc01_injection
Method	GET
Parameter	http://html5shim.googlecode.com/svn/trunk/html5.js
Attack	
Evidence	<script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>
Other Info	
URL	http://localhost:3000/team/vulnecab/thc01_injection
Method	GET
Parameter	https://cdnjs.cloudflare.com/ajax/libs/jquery.bootstrapvalidator/0.5.3/js/bootstrapValidator.js
Attack	
Evidence	<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/jquery.bootstrapvalidator/0.5.3/js/bootstrapValidator.js"></script>
Other Info	
URL	http://localhost:3000/team/vulnecab/thc01_injection
Method	GET
Parameter	https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js
Attack	
Evidence	<script type="text/javascript" src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
Other Info	
URL	http://localhost:3000/team/vulnecab/thc02_broken_auth
Method	GET
Parameter	http://html5shim.googlecode.com/svn/trunk/html5.js
Attack	
Evidence	<script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>
Other Info	
URL	http://localhost:3000/team/vulnecab/thc02_broken_auth
Method	GET
Parameter	https://cdnjs.cloudflare.com/ajax/libs/jquery.bootstrapvalidator/0.5.3/js/bootstrapValidator.js
Attack	
Evidence	<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/jquery.bootstrapvalidator/0.5.3/js/bootstrapValidator.js"></script>
Other Info	
URL	http://localhost:3000/team/vulnecab/thc02_broken_auth
Method	GET
Parameter	https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js
Attack	
Evidence	<script type="text/javascript" src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
Other Info	
URL	http://localhost:3000/team/vulnecab/thc03_sensitive_data
Method	GET
Parameter	http://html5shim.googlecode.com/svn/trunk/html5.js
Attack	
Evidence	<script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>
Other Info	
URL	http://localhost:3000/team/vulnecab/thc03_sensitive_data
Method	GET
Parameter	https://cdnjs.cloudflare.com/ajax/libs/jquery.bootstrapvalidator/0.5.3/js/bootstrapValidator.js
Attack	
Evidence	<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/jquery.bootstrapvalidator/0.5.3/js/bootstrapValidator.js"></script>
Other Info	
URL	http://localhost:3000/team/vulnecab/thc03_sensitive_data
Method	GET
Parameter	https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js
Attack	
Evidence	<script type="text/javascript" src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
Other Info	
URL	http://localhost:3000/team/vulnecab/thc04_xss
Method	GET
Parameter	http://html5shim.googlecode.com/svn/trunk/html5.js
Attack	
Evidence	<script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>
Other Info	
URL	http://localhost:3000/team/vulnecab/thc04_xss
Method	GET
Parameter	https://cdnjs.cloudflare.com/ajax/libs/jquery.bootstrapvalidator/0.5.3/js/bootstrapValidator.js
Attack	
Evidence	<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/jquery.bootstrapvalidator/0.5.3/js/bootstrapValidator.js"></script>
Other Info	
URL	http://localhost:3000/team/vulnecab/thc04_xss
Method	GET
Parameter	https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js
Attack	
Evidence	<script type="text/javascript" src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
Other Info	
URL	http://localhost:3000/team/vulnecab/thc05_broken_access_control
Method	GET
Parameter	http://html5shim.googlecode.com/svn/trunk/html5.js
Attack	
Evidence	<script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>
Other Info	
URL	http://localhost:3000/team/vulnecab/thc05_broken_access_control
Method	GET
Parameter	https://cdnjs.cloudflare.com/ajax/libs/jquery.bootstrapvalidator/0.5.3/js/bootstrapValidator.js
Attack	
Evidence	<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/jquery.bootstrapvalidator/0.5.3/js/bootstrapValidator.js"></script>
Other Info	
URL	http://localhost:3000/team/vulnecab/thc05_broken_access_control
Method	GET

Parameter	https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js
Attack	
Evidence	<script type="text/javascript" src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
Other Info	
URL	http://localhost:3000/login
Method	GET
Parameter	http://html5shim.googlecode.com/svn/trunk/html5.js
Attack	
Evidence	<script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>
Other Info	
URL	http://localhost:3000/login
Method	GET
Parameter	https://cdnjs.cloudflare.com/ajax/libs/jquery.bootstrapvalidator/0.5.3/js/bootstrapValidator.js
Attack	
Evidence	<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/jquery.bootstrapvalidator/0.5.3/js/bootstrapValidator.js"></script>
Other Info	
URL	http://localhost:3000/login
Method	GET
Parameter	https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js
Attack	
Evidence	<script type="text/javascript" src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
Other Info	
URL	http://localhost:3000/register
Method	GET
Parameter	http://html5shim.googlecode.com/svn/trunk/html5.js
Attack	
Evidence	<script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>
Other Info	
URL	http://localhost:3000/register
Method	GET
Parameter	https://cdnjs.cloudflare.com/ajax/libs/jquery.bootstrapvalidator/0.5.3/js/bootstrapValidator.js
Attack	
Evidence	<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/jquery.bootstrapvalidator/0.5.3/js/bootstrapValidator.js"></script>
Other Info	
URL	http://localhost:3000/verify
Method	GET
Parameter	https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js
Attack	
Evidence	<script type="text/javascript" src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
Other Info	
Instances	27
Solution	Ensure JavaScript source files are loaded from only trusted sources, and the sources can't be controlled by end users of the application.
Reference	
CWE ID	829
WASC ID	15
Plugin ID	10007
Low	Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)
Description	The web/application server is leaking information via one or more "X-Powered-By" HTTP response headers. Access to such information may facilitate attackers identifying other frameworks/components your web application is reliant upon and the vulnerabilities such components may be subject to.
URL	http://localhost:3000/
Method	GET
Parameter	
Attack	
Evidence	X-Powered-By: Express
Other Info	
URL	http://localhost:3000/assets/css/font-awesome.min.css
Method	GET
Parameter	
Attack	
Evidence	X-Powered-By: Express
Other Info	
URL	http://localhost:3000/assets/jquery-3.2.1.min.js
Method	GET
Parameter	
Attack	
Evidence	X-Powered-By: Express
Other Info	
URL	http://localhost:3000/assets/showdown.min.js
Method	GET
Parameter	
Attack	
Evidence	X-Powered-By: Express
Other Info	
URL	http://localhost:3000/vendor/jquery
Method	GET
Parameter	
Attack	
Evidence	X-Powered-By: Express
Other Info	
URL	http://localhost:3000/team
Method	GET
Parameter	
Attack	
Evidence	X-Powered-By: Express
Other Info	
URL	http://localhost:3000/team/dfscaibthca10_login
Method	GET
Parameter	
Attack	
Evidence	X-Powered-By: Express

Other Info	
URL	http://localhost:3000/team/vulnerability/cv2_injection
Method	GET
Parameter	
Attack	
Evidence	X-Powered-By: Express
Other Info	
URL	http://localhost:3000/team/vulnerability/cv2_broken_auth
Method	GET
Parameter	
Attack	
Evidence	X-Powered-By: Express
Other Info	
URL	http://localhost:3000/team/vulnerability/cv3_sensitive_data
Method	GET
Parameter	
Attack	
Evidence	X-Powered-By: Express
Other Info	
URL	http://localhost:3000/team/vulnerability/cv4_xss
Method	GET
Parameter	
Attack	
Evidence	X-Powered-By: Express
Other Info	
URL	http://localhost:3000/team/vulnerability/cv5_broken_access_control
Method	GET
Parameter	
Attack	
Evidence	X-Powered-By: Express
Other Info	
URL	http://localhost:3000/team/vulnerability/cv6_sec_misconf
Method	GET
Parameter	
Attack	
Evidence	X-Powered-By: Express
Other Info	
URL	http://localhost:3000/team/vulnerability/cv7_ess
Method	GET
Parameter	
Attack	
Evidence	X-Powered-By: Express
Other Info	
URL	http://localhost:3000/team/vulnerability/cv8_404s
Method	GET
Parameter	
Attack	
Evidence	X-Powered-By: Express
Other Info	
URL	http://localhost:3000/team/vulnerability/cv9_path_traversal
Method	GET
Parameter	
Attack	
Evidence	X-Powered-By: Express
Other Info	
URL	http://localhost:3000/team/vulnerability/cv10_curl
Method	GET
Parameter	
Attack	
Evidence	X-Powered-By: Express
Other Info	
URL	http://localhost:3000/team/vulnerability/cv11_reflected
Method	GET
Parameter	
Attack	
Evidence	X-Powered-By: Express
Other Info	
URL	http://localhost:3000/login
Method	GET
Parameter	
Attack	
Evidence	X-Powered-By: Express
Other Info	
URL	http://localhost:3000/logout
Method	GET
Parameter	
Attack	
Evidence	X-Powered-By: Express
Other Info	
URL	http://localhost:3000/register
Method	GET
Parameter	
Attack	
Evidence	X-Powered-By: Express
Other Info	
URL	http://localhost:3000/robots.txt
Method	GET
Parameter	
Attack	

Attack	
Evidence	
Other Info	The origin domain used for comparison was: localhost connect.sid=s%3ABYMO_3EHoN7k0baeRoFMdpN0S24Xx.%2BgRITCqOT9f%2FgAJuz0M6fAo4a4HccEga28Hy%2BN0k
URL	http://localhost:3000/validateurl
Method	GET
Parameter	
Attack	
Evidence	
Other Info	The origin domain used for comparison was: localhost connect.sid=s%3AFIAf1geEO7gtAWHcMSh8XPyXfGEidvS%3A%2gAK%2Fqts6%2FKQbcaf73Y
URL	http://localhost:3000/validateurl
Method	GET
Parameter	
Attack	
Evidence	
Other Info	The origin domain used for comparison was: localhost connect.sid=s%3AwbHm_LHTS_m0k2geNw8fCRX2URo4zuQlSoHdBSBDEgR5Rm8tE1SKRXQ2ZnRfgoAJmml
Instances	4
Solution	Always scope cookies to a FQDN (Fully Qualified Domain Name).
Reference	https://nodejs.org/en/blog/2018/03/26/cookies-4 https://owasp.org/www-project-web-security-testing-guide/v114.20th/Work_Application_Security_Testing/06-Session_Management_Testing/02-Testing_for_cookies_Attributes.html https://nodejs.org/en/blog/2018/03/26/cookies-4#the-secure-property-for-cookies
CWE ID	265
WASC ID	15
Plugin ID	99035
Informational	Session Management Response Identified
Description	The given response has been identified as containing a session management token. The 'Other Info' field contains a set of header tokens that can be used in the Header Based Session Management Method. If the request is in a context which has a Session Management Method set to "Auto-Detect" then this rule will change the session management to use the tokens identified.
URL	http://localhost:3000/login
Method	GET
Parameter	connect.sid
Attack	
Evidence	s%3ABYMO_3EHoN7k0baeRoFMdpN0S24Xx.%2BgRITCqOT9f%2FgAJuz0M6fAo4a4HccEga28Hy%2BN0k
Other Info	cookie:connect.sid
URL	http://localhost:3000/login
Method	GET
Parameter	connect.sid
Attack	
Evidence	s%3ABYVY4f77L3WwLc03K3pFP_fm6SDwBcGR8Op82p4NFTFkYx1U4eg76Qc9OKQidsJh1SV8v4
Other Info	cookie:connect.sid
URL	http://localhost:3000/validateurl
Method	GET
Parameter	connect.sid
Attack	
Evidence	s%3AFIAf1geEO7gtAWHcMSh8XPyXfGEidvS%3A%2gAK%2Fqts6%2FKQbcaf73Y
Other Info	cookie:connect.sid
URL	http://localhost:3000/validateurl
Method	GET
Parameter	connect.sid
Attack	
Evidence	s%3AwbHm_LHTS_m0k2geNw8fCRX2URo4zuQlSoHdBSBDEgR5Rm8tE1SKRXQ2ZnRfgoAJmml
Other Info	cookie:connect.sid
URL	http://localhost:3000/validateurl
Method	GET
Parameter	connect.sid
Attack	
Evidence	s%3AFIAf1geEO7gtAWHcMSh8XPyXfGEidvS%3A%2gAK%2Fqts6%2FKQbcaf73Y
Other Info	cookie:connect.sid
URL	http://localhost:3000/validateurl
Method	POST
Parameter	connect.sid
Attack	
Evidence	s%3AFIAf1geEO7gtAWHcMSh8XPyXfGEidvS%3A%2gAK%2Fqts6%2FKQbcaf73Y
Other Info	cookie:connect.sid
Instances	6
Solution	This is an informational alert rather than a vulnerability and so there is nothing to fix.
Reference	https://www.warrior-guy.com/2018/03/26/cookies-4-the-secure-property-for-cookies/
CWE ID	
WASC ID	
Plugin ID	10111

