



#MeetupGo  
#GoSec #Batch

@dadideo



1

# La Découverte de GoSec

David Aparicio

Meetup Go  
Mardi 3 Décembre 2024, 19h45



@dadideo

# David Aparicio

15/ DD INSA de Lyon / UNICAMP (Brésil)

Facebook Open Academy / MIT AppInventor

17/ Dev(Sec)Ops @ AMADEUS (Nice, 2 ans)

19/ Data(Sec)Ops @ OVHcloud (Lyon, 5 ans)

24/ Senior Dev(Sec)Ops @ Sopht (Lyon, 1 mois)





# Une solution pensée pour un pilotage centralisé de votre écosystème IT

## Collecte automatisée

Sur toute la chaîne de valeur d'un écosystème IT

## Observabilité à 360°

sur les scopes 1, 2 et 3 avec des données granulaires et dynamiques

## Actionnabilité guidée

Au moyen de simulations et recommandations contextualisées

## Suivi d'impact

Avec l'incidence de vos actions sur votre trajectoire en surveillant l'impact CO2/€

The screenshot displays the sopht platform's user interface across three main sections:

- Devices Monitoring:** Shows YTD Emissions (102 kgCO2eq), Range Emissions (53 kgCO2eq), and YTD Energy Cons (41 kgCO2eq) with year-over-year comparisons.
- Devices Usage:** A scatter plot showing device usage data for various Linux versions (e.g., Ubuntu 20.04, Ubuntu 22.04, etc.) across different regions (RU, BR, IT). A modal window prompts the user to "Select a device in the list below".
- Manufacturing Emission:** A chart showing Average Input (182) over time, with a callout highlighting a 45% reduction from 350 to 160.

On the left, a sidebar navigation includes Overview, Trajectories, Plans, Modules (Cloud & Infra, Workplace), and Asset Optimization. Favorites are listed under the Workplace section, including Multicloud YTD, Devices YTD, and Cloud Steering.



# Observabilité

- Mesure complète de l'écosystème IT
- Couverture des scopes 1, 2 et 3
- Facteurs d'émission multicritères
- Exploration dynamique des données
- Projection et prévision d'impact
- Filtrage des données par entité





## Nos principales références



BNP PARIBAS  
La banque d'un monde qui change



AIRBUS



Adecco

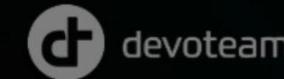
SEQENS

CED

soitec



## Nos partenaires



accenture

## Intégrations aux plateformes ESG



planA



SWEET



“

Évaluer et comprendre nos principaux facteurs d'émission est évidemment un enjeu en soi, mais les piloter de manière dynamique avec des plans d'actions concrets, c'est ce que nous permet la plateforme Sopht.



**Marie Ait-Daoud**  
Green IT manager  
Vinci Groupe



# Agenda

Introduction

Demo

Outils

Conseils

Conclusion

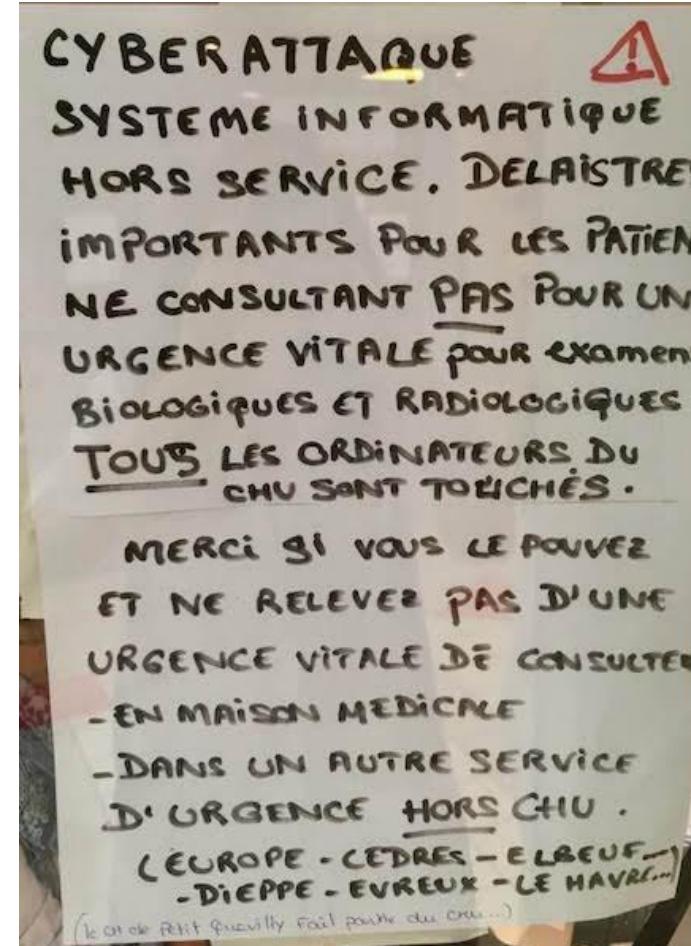


# Introduction





# Pourquoi ce talk ?



[Thread @zigazou](#)



# Agir.. Avant la PROD!

## Y a-t-il un pilote à jour dans l'avion ?

En 2015, les autorités états-uniennes de l'aviation alertaient les compagnies aériennes: le Boeing 787 Dreamliner devait être redémarré tous les 248 jours pour contourner un bogue pouvant entraîner une coupure de courant généralisée dont on peut imaginer les conséquences en vol. Cette fois, elles ont

annoncé qu'il faut éteindre et rallumer ces mêmes avions tous les 51 jours pour éviter des problèmes informatiques catastrophiques en raison d'une mémoire saturée de données sinon. Mesdames et Messieurs, veuillez regagner vos places et attacher vos ceintures de sécurité, nous allons bientôt rebouter!





# Pourquoi ?

2013	2017 (new, * from the community)	2021 (new, * from the survey)
A1 - Injection	A1 - <b>Injection</b>	A1 - Broken Access Control
A2 - Broken Authentication & Session Management	A2 - <b>Broken Authentication</b>	A2 - Cryptographic Failures
A3 - Cross-Site Scripting (XSS)	A3 - <b>Sensitive Data Exposure</b>	A3 - <b>Injection</b>
<b>A4 - Insecure Direct Object References</b>	A4 - XML External Entities (XXE)	<b>A4 - Insecure Design</b>
A5 - Security Misconfiguration	A5 - <b>Broken Access Control [MERGED A4+A7]</b>	A5 - Security Misconfiguration
A6 - Sensitive Data Exposure	A6 - <b>Security Misconfiguration</b>	A6 - Vulnerable and Outdated Components
<b>A7 - Missing Function Level Access Control</b>	A7 - Cross-Site Scripting (XSS)	A7 - Identification and Authentication Failures
A8 - Cross-Site Request Forgery (CSRF)	<b>A8 - Insecure Deserialization *</b>	<b>A8 - Software and Data Integrity Failures</b>
A9 - Using Components with Known Vulnerabilities	A9 - <b>Using Components with Known Vulnerabilities</b>	A9 - Security Logging and Monitoring Failures *
A10 - Unvalidated Redirects and Forwards	<b>A10 - Insufficient Logging &amp; Monitoring *</b>	<b>A10 - Server-Side Request Forgery (SSRF) *</b>

OWASP TOP 10



OWASP Top 10



Bonjour ChatGPT! Peux-tu me donner des actions concrètes pour sécuriser mon code ?



Bonjour! Voici quelques actions concrètes pour sécuriser votre code :

1. Effectuer des tests d'intrusion réguliers pour détecter les vulnérabilités dans votre code.
2. Utiliser des mots de passe forts et un stockage sécurisé des mots de passe.
3. Valider toutes les entrées utilisateur pour éviter les attaques par injection SQL ou d'autres types d'injection.
4. Utiliser des autorisations d'accès appropriées pour limiter l'accès aux fonctionnalités sensibles de votre application.
5. Utiliser des bibliothèques et des frameworks de sécurité réputés pour ajouter des couches supplémentaires de sécurité à votre code.
6. Assurez-vous que vos dépendances et bibliothèques utilisées sont à jour et sécurisées.
7. Utilisez des outils pour détecter les vulnérabilités de sécurité automatiquement.
8. Utilisez des mécanismes de chiffrement pour protéger les données sensibles.
9. Utilisez des certificats SSL pour sécuriser les communications réseau.
10. Utilisez une stratégie de gestion des vulnérabilités pour gérer efficacement les vulnérabilités de sécurité détectées.



---

# gosec - Go Security Checker

Version 1.0.0 en Juillet 2018

Par Cosmin Cojocar (Google 🇫🇷)

Intégrable à plusieurs CI/CD/Linters

Déetecte certaines attaques/failles **bien connues**

Dernier commit 30 Novembre 2024 (4 jours)

Assez populaire dans la communauté Go





# DAST (Gitlab)

Language (package managers) / framework	Scan tool
.NET Core	<a href="#">Security Code Scan</a> ↗
C/C++	<a href="#">Flawfinder</a> ↗
Go	<a href="#">Gosec</a> ↗
Helm Charts	<a href="#">Kubesec</a> ↗
Java ( <a href="#">Ant</a> ↗, <a href="#">Gradle</a> ↗, <a href="#">Maven</a> ↗, <a href="#">SBT</a> ↗)	<a href="#">SpotBugs</a> ↗ with <a href="#">find-sec-bugs</a> ↗
Java / Kotlin (Android)	<a href="#">MobSF (beta)</a> ↗
JavaScript	<a href="#">ESLint security plugin</a> ↗
Kubernetes manifests	<a href="#">Kubesec</a> ↗
Node.js	<a href="#">NodeJsScan</a> ↗
PHP	<a href="#">phpcs-security-audit</a> ↗
Python ( <a href="#">pip</a> ↗)	<a href="#">bandit</a> ↗

## Available rules

- G101: Look for hard coded credentials
- G102: Bind to all interfaces
- G103: Audit the use of unsafe block
- G104: Audit errors not checked
- G106: Audit the use of ssh.InsecureIgnoreHostKey
- G107: Url provided to HTTP request as taint input
- G108: Profiling endpoint automatically exposed on /debug/pprof
- G109: Potential Integer overflow made by strconv.Atoi result conversion to int16/32
- G110: Potential DoS vulnerability via decompression bomb
- G201: SQL query construction using format string
- G202: SQL query construction using string concatenation
- G203: Use of unescaped data in HTML templates
- G204: Audit use of command execution
- G301: Poor file permissions used when creating a directory
- G302: Poor file permissions used with chmod
- G303: Creating tempfile using a predictable path
- G304: File path provided as taint input
- G305: File traversal when extracting zip/tar archive
- G306: Poor file permissions used when writing to a new file
- G307: Deferring a method which returns an error
- G401: Detect the usage of DES, RC4, MD5 or SHA1
- G402: Look for bad TLS connection settings
- G403: Ensure minimum RSA key length of 2048 bits
- G404: Insecure random number source (rand)
- G501: Import blocklist: crypto/md5
- G502: Import blocklist: crypto/des
- G503: Import blocklist: crypto/rc4
- G504: Import blocklist: net/http/cgi
- G505: Import blocklist: crypto/sha1
- G601: Implicit memory aliasing of items from a range statement

## Retired rules

- G105: Audit the use of math/big.Int.Exp - CVE is fixed



# DAST (GitHub)

https://github.com/marketplace/actions/gosec-security-checker

Marketplace / Actions / Gosec Security Checker

GitHub Action

 Gosec Security Checker

v2.21.4 [Latest version](#)

Use latest version

gosec - Go Security Checker

Inspects source code for security problems by scanning the Go AST and SSA code representation.

Stars

Star 7.9k

Contributors

Links

- [securego/gosec](#)
- Open issues 9
- Pull requests 0
- Report abuse

Gosec Security Checker is not certified by GitHub. It is provided by a third-party and is governed by separate terms of service, privacy policy, and support documentation.

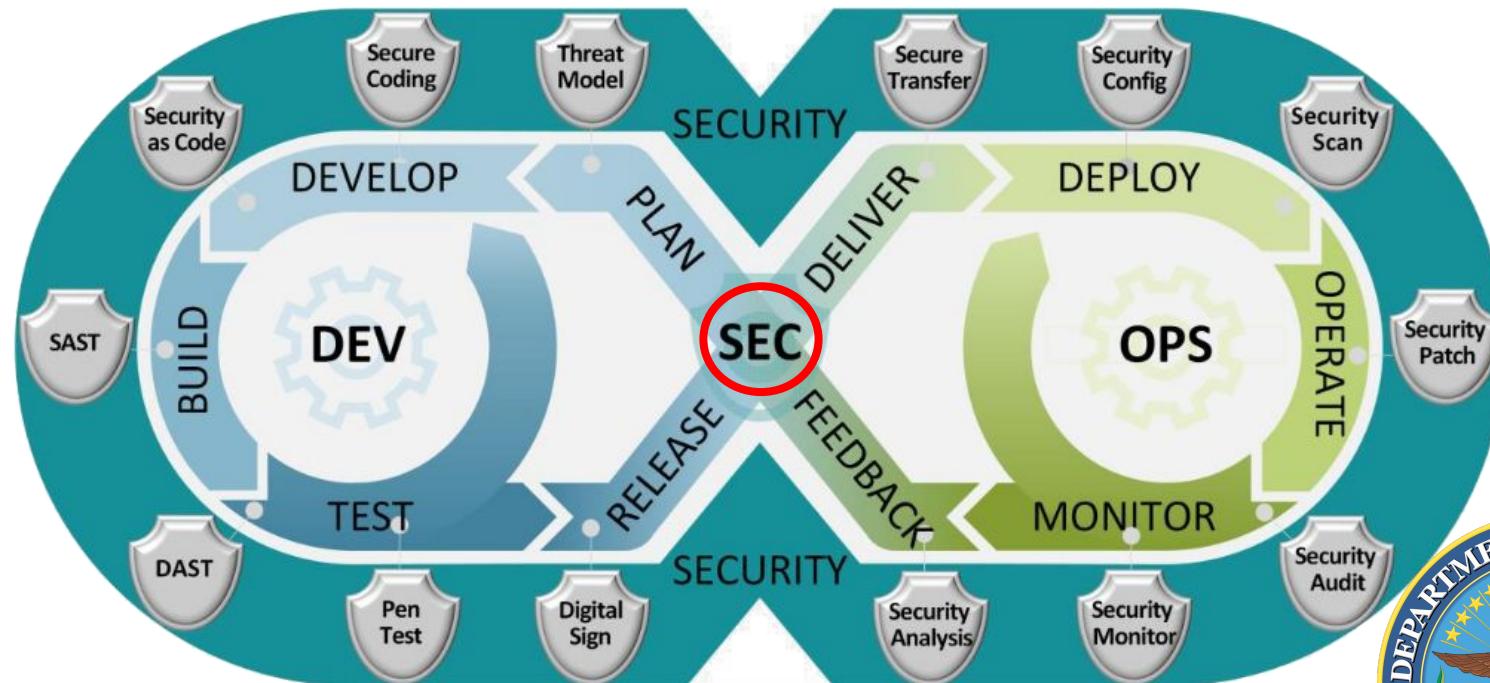
## License

Licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance with the License. You may obtain a copy of the License [here](#).





# Shift-left Security



[dodcio.defense.gov](http://dodcio.defense.gov)





# Demo





#MeetupGo  
#GoSec #Batch

@dadideo



18



# DDoS

Liran Tal on X: "⚠️ Beware of DoS attacks in Node.js! Learn how to protect your applications with rate limiting, timeouts, and asynchronous operations

#nodejs #security #webdev

4:07 PM · Dec 2, 2024 · 1 View

Post your reply

Reply

Discover more

Sourced from across X

Relevant people

Liran Tal @liran\_tal Following

Node.js Secure Coding: nodejs-security.com 🌟 Awarded @GitHub Star 🏆 Awarded @OpenJS Pathfinder award for Security 🥅 DevRel @snyksec

What's happening

News · Trending ALERTE INFO 6,661 posts

Twitter @Liran Tal



# DDoS

github-advanced-security (bot) found potential problems on Oct 23, 2022

cmd/server/main.go

```
61      -      }
62      -      }
63      -      if err := srv.ListenAndServe(); err != nil {
52 +      if err := http.ListenAndServe(":8080", nil); err != nil {
```

✖ Check failure

Code scanning / gosec

Use of net/http serve function that has no support for setting timeouts

Use of net/http serve function that has no support for setting timeouts

Show more details

Reply...

sonarcloud (bot) commented on Dec 31, 2022

Kudos, SonarCloud Quality Gate passed! Passed

- 0 Bugs
- 0 Vulnerabilities
- 0 Security Hotspots
- 0 Code Smells

No Coverage information

0.0% Duplication

...





#MeetupGo  
#GoSec #Batch

@dadideo



20



David Aparicio > gokvs > Pull Requests > 🔥 81 – 🔥 rem: Go Meetup 3 December 2024 (Batch, Lyon) 🎉 ✅

Summary Issues Security Hotspots Measures Code

## PR Summary

4 New Lines • 📈 meetup\_3Dec24 → ⚙ main

Quality Gate ?



# Passed

Last analysis 44 minutes ago • 🏷 dc7b9770

### New Issues

0

No conditions set

### Accepted Issues

0

Valid issues that were not fixed

### Coverage

There is not enough lines to compute coverage

### Duplications

0.0%

Required ≤ 3.0%  
on 4 New Lines

0.0% Estimated after merge

### Security Hotspots

0

No conditions set



```
[main.go:12] - G304: Potential file inclusion via variable
(Confidence: HIGH, Severity: MEDIUM)

11: func readFile(filename string) {
> 12:     data, err := ioutil.ReadFile(filename)
13:     if err != nil {
```



```
package main

import (
    "fmt"
    "io/ioutil"
)

func main() {
    userInput := "/etc/passwd" // Entrée contrôlée par l'attaquant
    content, err := ioutil.ReadFile(userInput)
    if err != nil {
        panic(err)
    }
    fmt.Println(string(content))
}
```



```
package main

import (
    "fmt"
    "io/ioutil"
    "path/filepath"
    "strings"
)

func main() {
    userInput := "/etc/passwd" // Entrée contrôlée par l'attaquant
    safebasePath := "/safe/path"

    cleanPath := filepath.Clean(userInput)
    if !strings.HasPrefix(cleanPath, safebasePath) {
        panic("Chemin non autorisé")
    }

    content, err := ioutil.ReadFile(cleanPath)
    if err != nil {
        panic(err)
    }
    fmt.Println(string(content))
}
```



# AI-inside

## Auto fixing vulnerabilities

gosec can suggest fixes based on AI recommendation. It will call an AI API to receive a suggestion for a security finding.

You can enable this feature by providing the following command line arguments:

- `ai-api-provider` : the name of the AI API provider, currently only `gemini` is supported.
- `ai-api-key` or set the environment variable `GOSEC_AI_API_KEY` : the key to access the AI API, For gemini, you can create an API key following [these instructions](#).
- `ai-endpoint` : the endpoint of the AI provider, this is optional argument.

```
gosec -ai-api-provider="gemini" -ai-api-key="your_key" ./...
```





# Outils





# Linters

## Go

Un linter est un outil d'analyse statique de code source. Il sert à détecter : des erreurs (très utile sur des langages interprétés comme JavaScript qui n'ont pas de phase de compilation) ; des problèmes de syntaxe et de non-respect de style (tabulation vs espaces, indentation, etc.)

```
linters:  
  disable-all: true  
  enable:  
    - bodyclose  
    - deadcode  
    - depguard  
    - dogsled  
    - dupl  
    - errcheck  
    - funlen  
    - goconst  
    - gocritic  
    - gocyclo  
    - gofmt  
    - goimports  
    - golint  
    - gomnd  
    - goprintffuncname  
    - gosec  
    - gosimple  
    - govet  
    - ineffassign  
    - interfacer  
    - misspell  
    - nakedret  
    - rowerrcheck  
    - scopelint  
    - staticcheck  
# - ...
```

## STATIC LINTS WITH GOLANG-CI



Customize: linters list, values...

In few situations you can bypass the linters with nolint directive.

//nolint



"Common mistakes" en Go, Aurélie Vache  
(Async 2021)



# CI/CD

Pipeline Jobs 5



[Philippe Charrière \(Twitter\)](#)



# Autre outil

## Govulncheck v1.0.0 is released!

*Julie Qiu, for the Go security team*

13 July 2023

We are excited to announce that govulncheck v1.0.0 has been released, along with v1.0.0 of the API for integrating scanning into other tools!

Go's support for vulnerability management was [first announced](#) last September. We have made several changes since then, culminating in today's release.

This post describes Go's updated vulnerability tooling, and how to get started using it. We also recently published a [security best practices guide](#) to help you prioritize security in your Go projects.

### Govulncheck

[Govulncheck](#) is a command-line tool that helps Go users find known vulnerabilities in their project dependencies. The tool can analyze both codebases and binaries, and it reduces noise by prioritizing vulnerabilities in functions that your code is actually calling.

You can install the latest version of govulncheck using [go install](#):

```
go install golang.org/x/vuln/cmd/govulncheck@latest
```

Then, run govulncheck inside your module:

```
govulncheck ./...
```

See the [govulncheck tutorial](#) for additional information on how to get started with using the tool.

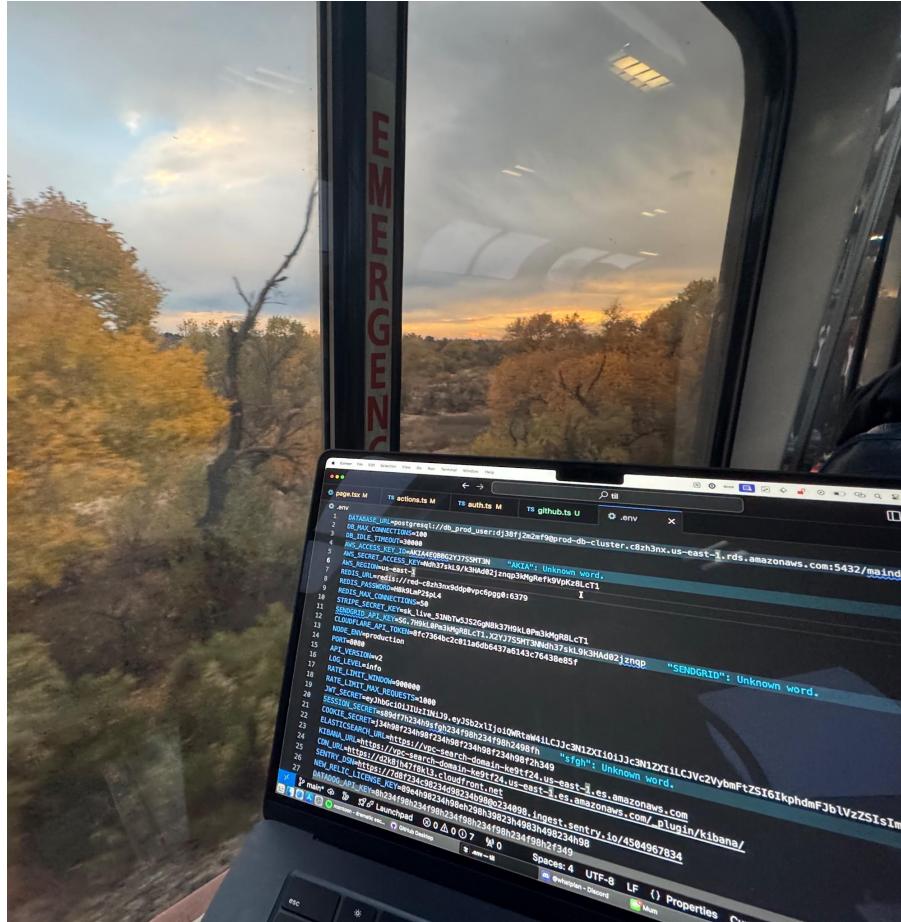
As of this release, there is now a stable API available, which is described at [golang.org/x/vuln/scan](#). This API provides the same functionality as the govulncheck command, enabling developers to integrate security scanners and other tools with govulncheck. As an example, see the [osv-scanner integration with govulncheck](#).

### Database

Govulncheck is powered by the Go vulnerability database, <https://vuln.go.dev>, which provides a comprehensive source of information about known vulnerabilities in public Go modules. You can browse the entries in the database at [pkgo.dev/vuln](http://pkgo.dev/vuln).



# Scan Every- where





# Conseils





Pas copier-coller depuis StackOverFlow

98% snippets sécu/crypto sont insecures



Fisher et al., 2017; Nadi et al., 2016; Das et al., 2014, Prevent cryptographic pitfalls by design



## GitHub Copilot Security Study: 'Developers Should Remain Awake' in View of 40% Bad Code Rate

By David Ramel 08/26/2021

Researchers published a scholarly paper looking into security implications of GitHub Copilot, an advanced AI system now being used for code completion in Visual Studio Code and possibly headed for Visual Studio after its current preview period ends.

In multiple scenario testing, some 40 percent of tested projects were found to include security vulnerabilities.

**GitHub Copilot** is described as an "[AI pair programmer](#)" whose advanced AI



[40% of Code Produced by GitHub Copilot Vulnerable to Threats](#)



# Bonnes pratiques ANSSI

Se documenter, se former

Lire les guides de l'ANSSI

Comparer les technologies, les langages de programmation

Effectuer l'analyse des risques

Identifier le modèle de l'attaquant pour ce produit en particulier

Préparer des spécifications / des ateliers

Participer à des conférences Sécurité

Choix du système hôte ([OS hardening](#))

Veille technologique ([Feedly/RSS](#))



ANSSI

Agence nationale de la sécurité des  
systèmes d'information



RECOMMANDATIONS RELATIVES À L'INTERCONNEXION D'UN SYSTÈME D'INFORMATION  
Réseaux

19/06/2020

architecture interconnexion Internet messagerie passerelle

RÈGLES DE PROGRAMMATION POUR LE DÉVELOPPEMENT D'APPLICATIONS SÉCURISÉES  
09/06/2020

application sécurisée bonne pratique développement sécurisé langage de règle

RECOMMANDATIONS DE SÉCURITÉ RELATIVES À TLS  
Cryptographie Réseaux

26/03/2020

chiffrement HTTPS TLS

RECOMMANDATIONS SUR LA SÉCURISATION DES SYSTÈMES DE CONTRÔLE D'ACCÈS PHYSIQUE ET DE VIDÉOPROTECTION



[Bonnes pratiques de sécurité numérique \(ANSSI\)](#)



# Conclusion





# TL;DR - The state of open source security 2019 report, at a glance



## Open source adoption

- ▷ Growth in indexed packages, 2017 to 2018
  - ❖ Maven Central - 102%
  - ❖ PyPI - 40%
  - ❖ npm - 37%
  - ❖ NuGet - 26%
  - ❖ RubyGems - 5.6%
- ▷ npm reported 304 billion downloads for 2018
- ▷ 78% of vulnerabilities are found in indirect dependencies



## Known vulnerabilities

- ▷ 88% growth in application vulnerabilities over two years
- ▷ In 2018, vulnerabilities for npm grew by 47%. Maven Central and PHP Packagist disclosures grew by 27% and 56% respectively
- ▷ In 2018, we tracked over 4 times more vulnerabilities found in RHEL, Debian and Ubuntu as compared to 2017



## Known vulnerabilities in docker images

- ▷ Each of the top ten most popular default docker images contains at least 30 vulnerable system libraries
- ▷ 44% of scanned docker images can fix known vulnerabilities by updating their base image tag



## Snyk stats

- ▷ In the second half of 2018 alone, Snyk opened more than 70,000 Pull Requests for its users to remediate vulnerabilities in their projects
- ▷ CVE/NVD and public vulnerability databases miss many vulnerabilities, only accounting for 60% of the vulnerabilities Snyk tracks
- ▷ In 2018 alone, 500 vulnerabilities were disclosed by Snyk's proprietary dedicated research team



## Vulnerability identification

- ▷ 37% of open source developers don't implement any sort of security testing during CI and 54% of developers don't do any docker image security testings
- ▷ The median time from when a vulnerability was added to an open source package until it was fixed was over 2 years



## Who's responsible for open source security?

- ▷ 81% of users feel developers are responsible for open source security
- ▷ 68% of users feel that developers should own the security responsibility of their docker container images
- ▷ Only three in ten open source maintainers consider themselves to have high security knowledge



The state of open  
source security - 2019



---

# Analogie

« Nul n'est censé ignorer la loi »





---

# Ma devise

« Nul développeur n'est censé ignorer la sécurité »





## 🎯 Pour aller plus loin

- [ANSSI \(Sécurité Agile\)](#), Applications sécurisés en [Rust](#), Déploiement de conteneurs [Docker](#)
- [10 leçons sur les 10 plus grosses fuites de données](#), de Adrien Pessu (JSC 2020)
- [La Cryptographie en 55' chrono](#) de m4dz (SnowCamp2020)
- [Sécurité du Cloud](#), de Eric Briand (RemoteClazz 2020)
- [La nuit tous les hackers sont gris](#) (Fiction écrite par Vincent Hazard, 2019)



#MeetupGo  
#GoSec #Batch

https://www.sstic.org

7 au 9 juin 2017

SSTIC 2017

38:39 / 1:16:25

## Traumatisme

- Ce genre d'incident de sécurité a plusieurs conséquences
  - Conditions de travail très dures : horaires importants, vacances annulées, pression croissante...
  - Traumatisme lié à l'attaque qui perdure et qui est difficile à percevoir lorsque l'ANSSI intervient
  - La crainte que l'attaquant revienne est permanente

Retour technique de  
l'incident de TV5Monde

ANSSI



# Pour aller plus loin



#MeetupGo  
#GoSec #Batch

@dadideo



40



[La Sécurité dès la conception \(Secure by design\)](#), Programmez! (Hors-série 8 Septembre/Octobre 2022)



# Rappelez-vous: Les hackers n'en ont rien à "faire"

- À propos du scope de votre projet
- Il est géré par une tierce partie / sous-traitant
- C'est un système ancien (Legacy)
- TPCM / " Touche pas ! C'est magique "
- C'est "trop critique pour être réparé"
- A propos de vos périodes de maintenance
- A propos de votre budget
- Vous l'avez toujours fait de cette façon
- À propos de votre date de mise en service
- Il s'agit seulement d'un pilote/PoC
- À propos des accords de non-divulgation
- Ce n'était pas une exigence dans le contrat
- C'est un système interne
- Il est vraiment difficile de modifier / changer
- Vous n'êtes pas sûr de savoir comment y remédier
- Il doit être remplacé
- C'est géré dans le Cloud
- À propos de votre inscription au registre des risques
- L'éditeur ne prend pas en charge cette configuration
- C'est une solution provisoire
- Il est conforme à [insérer la norme ici]
- Il est crypté sur disque
- Le rapport coût-bénéfice ne scale pas
- "Personne d'autre ne pouvait le comprendre"
- Vous ne pouvez pas expliquer le risque au "Business"
- Vous avez d'autres priorités
- Sur votre foi dans la compétence de vos utilisateurs internes
- Vous n'avez pas de justification commerciale
- Vous ne pouvez pas montrer le retour sur investissement
- Vous avez sous-traité ce risque
- C'était à la mode [insérer la technologie hype ici].
- De vos certifications





---

Merci beaucoup aux  
organisateurs & à Batch  
pour nous accueillir ce soir





# Merci pour votre attention !

 N'oubliez pas de me donner votre avis sur cette session: <https://tinyurl.com/meetgo24>

 Lien des slides dans les commentaires 



Faites-vous confiance aux QR Codes ?