sentryo

Cyber Security for the **Industrial Internet**

Situational Awareness + Anomaly Detection + Security Management

Sentryo HQ | 66 Bd Niels Bohr CS 52132 69603 Lyon-Villeurbanne - France
+33 970 469 694 | contact@sentryo.net | www.sentryo.net

sentryo

Golang dans une startup de cyber-sécurité

Sommaire

- Présentation de Sentryo et de l'ICS CyberVision
- Adoption de Golang
- Évolution avec Golang
- Conclusion

COMPANY PURPOSE / IDENTITY

sentryo

Protects

the Industrial Internet against cyber risks

- **Incorporated** in June 2014
- **Raised** 3 M€ funding
- **Headquarter:** Lyon
- Design, develop & sell cybersecurity solutions
- Funded and managed by serial entrepreneurs & cybersecurity veterans

DIGITAL IN-PULSE 2015

Award winner
Innovative Startup Contest,
Sponsorisé par BPI, Business France and Huawei Sept 2015

prix de l'innovation des assises 2015

Award winner
Prix de l'innovation des Assises de la sécurité July 2015

PrizeWinner
EIT ICT Labs IDEA CHALLENGE

Prize winner
European Institute of Technology
Idea Challenge Nov 2014

TREMPLIN ENTREPRISES 2015 SÉNAT

Award winner
Tremplin Entreprise
Sénat / Essec Feb 2015

Member: HEXATRUST CYBERSECURITY & DIGITAL TRUST — TECH'IN FRANCE

USE CASE : STEEL MILL IN GERMANY

WIRED – Jan 8th 2015

‘A cyberattack has caused confirmed physical damage’.

According to the German BSI, hackers had struck an unnamed steel mill in Germany.

They did so by manipulating and disrupting control systems to such a degree that a blast furnace could not be properly shut down, **resulting in ‘massive’ damage**.

SENTRYO ICS CYBERVISION BENEFITS

Sentryo ICS CyberVision delivers an operational security capacity to prevent detect and respond to cyber attacks targeting the Industrial Internet.

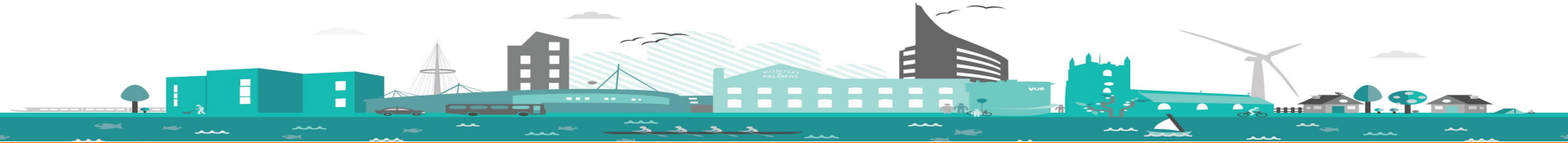
Prevent & Protect

- **Prevent:** Enable **Control Engineers** to reduce the attack surface
- **Report & comply:** Provide reports to the **Security Officer**

Detect & Respond

- **Detect :** Detect malicious behaviors and **weak signals** of cyber attacks
- **Respond :** Facilitate incident response by gathering all the informations required by **Cybersecurity Expert**

Streamline OT and IT collaboration



sentryo

Cyber Security for the **Industrial Internet**

Adoption d’un langage

LANGAGES DISPONIBLES

Python

Librairie standard fournie, beaucoup de développeurs, développement rapide, ...

C / C++

Performance, beaucoup de ressources disponibles, nombreux paradigmes, ...

C / C++

Java

VM éprouvée, langage connu, accessible et beaucoup de développeurs disponibles, ...

Pourquoi le choix a-t-il été porté sur **Golang** ?

2014

Proof-of-Concept fonctionnel en Go

Courant 2015

MVP (Minimum Viable Product) terminé, version 1.0 disponible pour les clients.

Aujourd'hui

Go toujours omni-présent, quelques scripts d'administrations / de configurations en Python et en Bash.

2 années de Golang pour Sentryo

sentryo

Cyber Security for the **Industrial Internet**

Utilisation d’un langage

GOLANG - PHILOSOPHIE

Lorsque Golang répond à un problème, il essaye d'y répondre d'une manière simple et claire **pour le développeur**.

Définition claire et syntaxe simple du langage
- Effective Go, ~ 70 pages pour l'intégralité
- Go est maintenant écrit en… Go

Gestion de la mémoire
- Garbage-collector performant

Déploiement / mise-en-production
- Binaire statique
- Cross-compilation très simple

Concurrence facilitée au coeur même du langage
- go routine, channels
- Package `sync`

Effective Go

GOLANG - VOUS AVEZ DIT SIMPLE ?

Exécution synchrone de A et B

```
package main

import (
    "fmt"
    "time"
)

func main() {
    A()
    B()

    time.Sleep(5 * time.Second)
}

func A() {
      fmt.Println("A")
}
func B() {
      fmt.Println("B")
}
```

GOLANG - VOUS AVEZ DIT SIMPLE ?

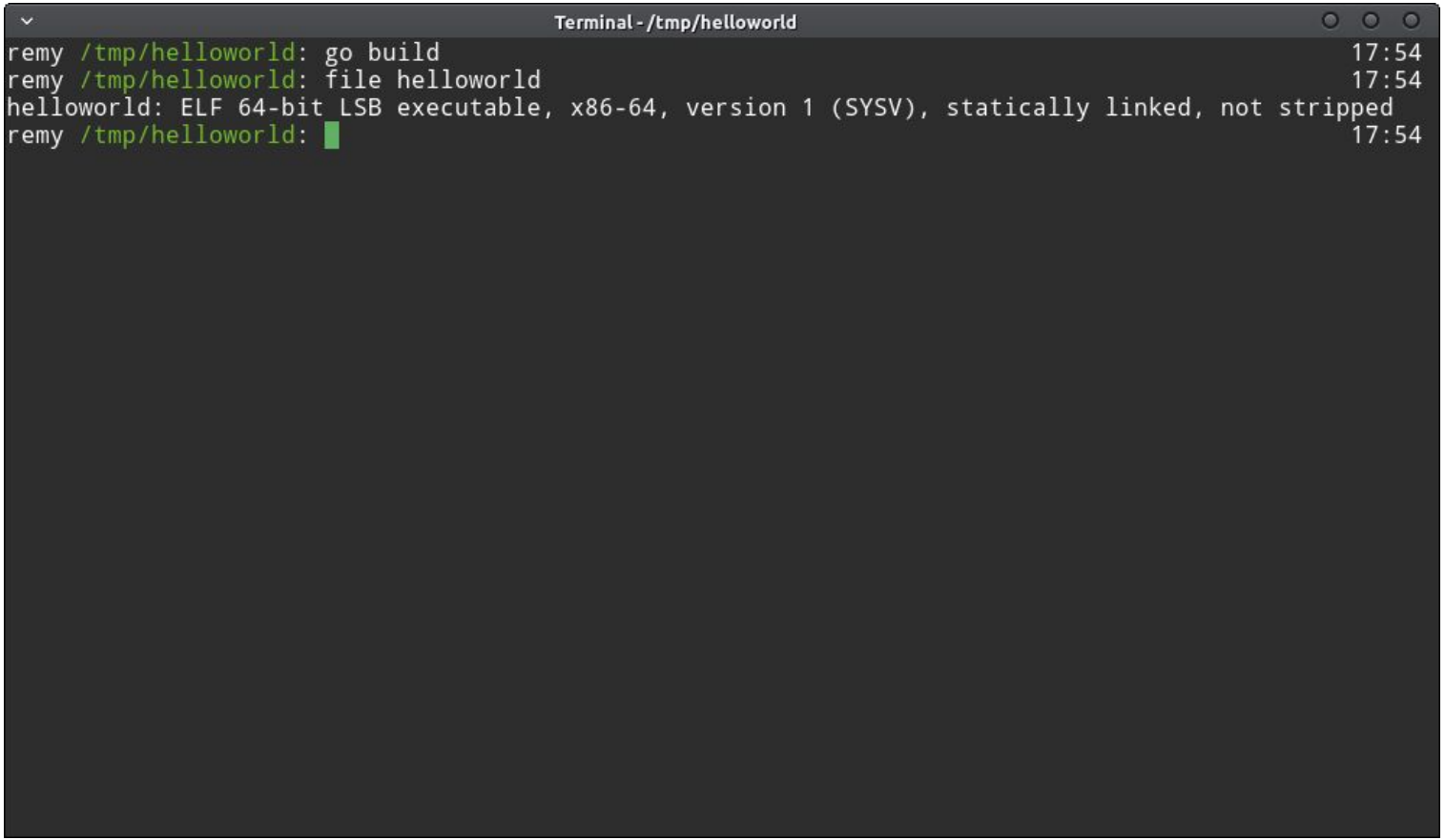
Exécution synchrone de A et B

```
package main

import (
    "fmt"
    "time"
)

func main() {
    A()
    B()

    time.Sleep(5 * time.Second)
}

func A() {
      fmt.Println("A")
}
func B() {
      fmt.Println("B")
}
```

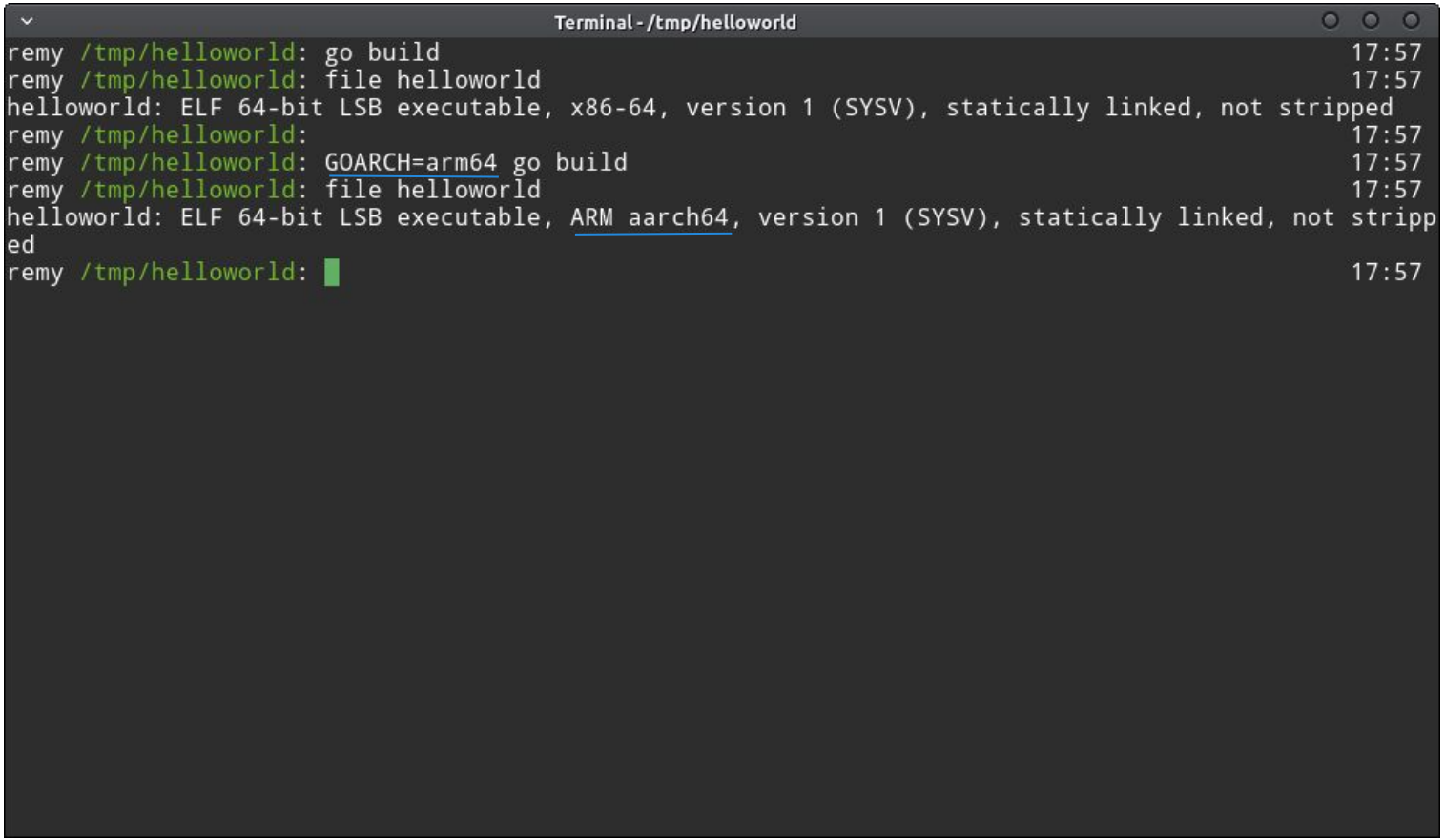
Exécution **concurrente** de A et B

```
package main

import (
    "fmt"
    "time"
)

func main() {
    go A()
    go B()

    time.Sleep(5 * time.Second)
}

func A() {
      fmt.Println("A")
}
func B() {
      fmt.Println("B")
}
```

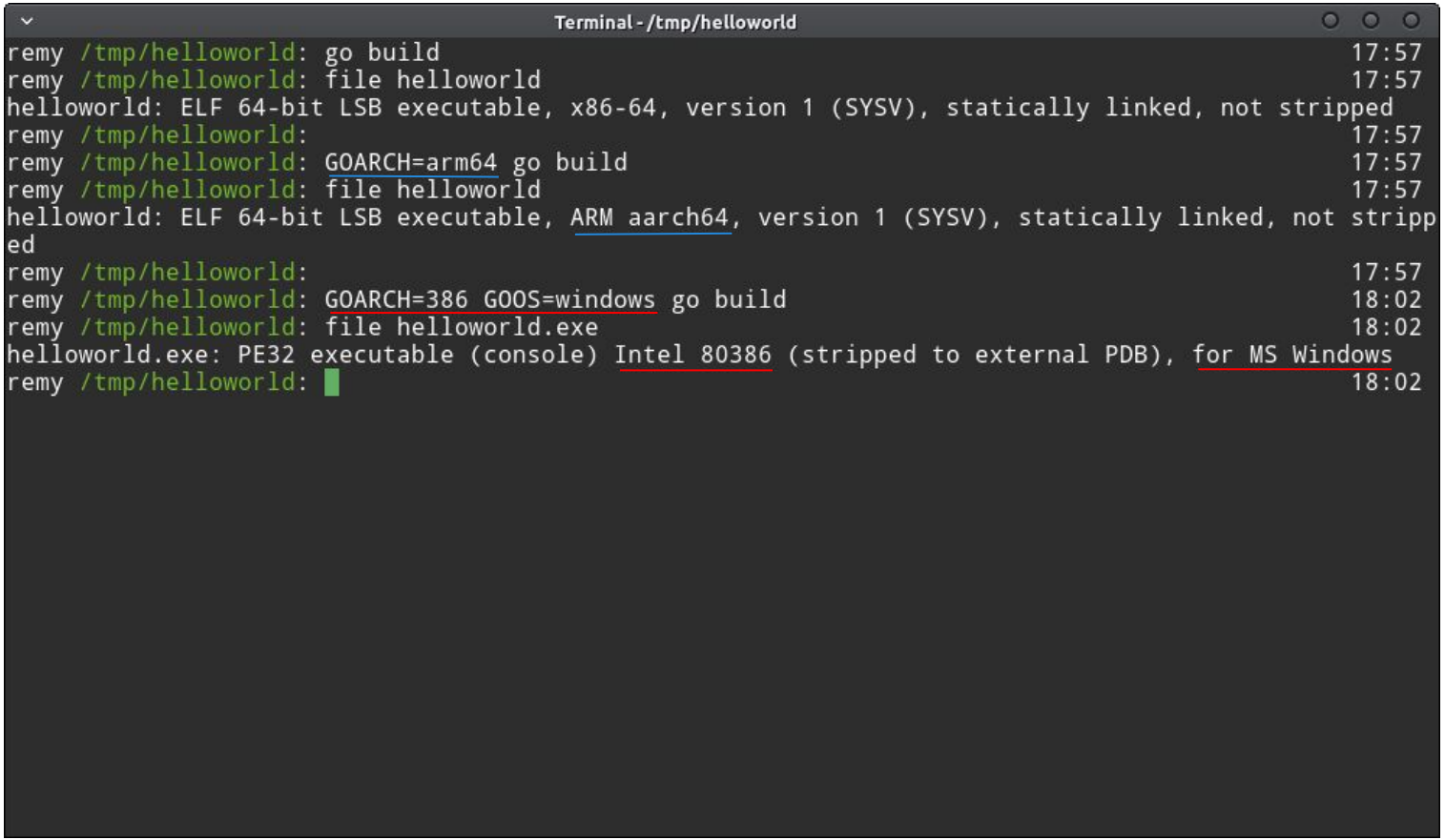
CROSS COMPILATION - VOUS AVEZ DIT SIMPLE ?



CROSS COMPILATION - VOUS AVEZ DIT SIMPLE ?



CROSS COMPILATION - VOUS AVEZ DIT SIMPLE ?



GOLANG - STDLIB

- Les packages standards sont utilisés en production dans de nombreuses entreprises
  - Éprouvés
  - Performants

- Sources disponibles, claires et entièrement documentées : les meilleurs exemples idiomatiques

Ce sont les packages standards, les outils et l'implémentation du langage (GC, etc.) qui changent entre les versions, le **langage lui n'a (quasiment) pas changé depuis Go 1.0 en 2009**

TOOLING - DEVELOPMENT

Chacun son éditeur de texte mais partage du **format du code** :

- go fmt

Outils de **tests unitaires** et de **benchmarks** intégrés au langage :

- go test [-bench]

Génération de la d**ocumentation** :

- go doc

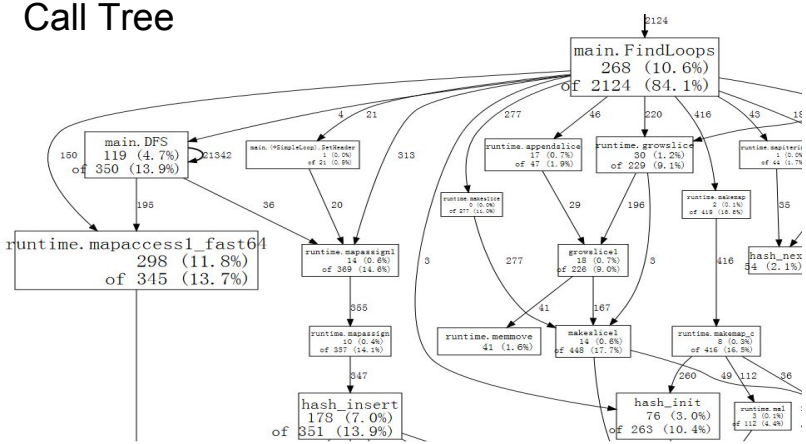
Compilation et création des binaires :

- go generate
- go build

Et gocode, guru, goimports, play.golang.org, go tool, ...

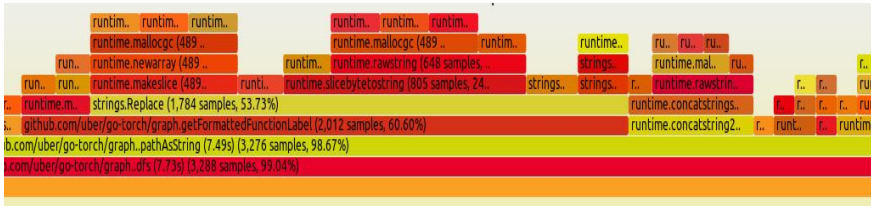
TOOLING - PROFILING

Call Tree



En plus des outils pour le développement, les outils d'analyse "en production" des applications.

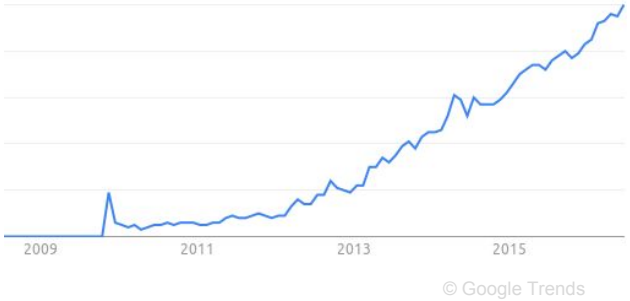
Flame graph (mis à disposition par Uber)



ENTREPRISE - RECRUTEMENT

- Langage attractif
- Encore peu présent dans les formations = souvent un investissement personnel
- Possibilité de rapidement former quelqu'un qui vient d'un autre langage / framework.

Recherche **Golang** sur Google



Malgré tout, il est encore difficile de recruter un développeur Go

GOLANG - LANGAGE PARFAIT ?

Non !

Encore des lacunes du côté du debuggage

- Des outils comme `delve` font leur apparition

Binaires statiques relativement gros

- 30% de gain avec Golang 1.7

```
remy /tmp/helloworld: go build && strip helloworld                                  17:04
remy /tmp/helloworld: gcc -o hello -O3 -Werror -Wall -static hello.c && strip hello 17:04
remy /tmp/helloworld: ls -lh                                                        17:04
total 2.3M
-rwxr-xr-x 1 remy remy 710K Jun 27 17:04 hello
-rw-r--r-- 1 remy remy   52 Jun 27 17:00 hello.c
-rwxr-xr-x 1 remy remy 1.6M Jun 27 17:04 helloworld
-rw-r--r-- 1 remy remy   71 Jun 27 16:59 helloworld.go
remy /tmp/helloworld:                                                               17:04
```

Garbage collector

- Ne convient évidemment pas à tous les use-cases

Si l'on compare à Java, l'adoption par les développeurs est encore en cours.

CONCLUSION

- Facilité de prise en main, d'utilisation, d'optimisation = efficacité.
- Une philosophie saine de développement.
- Documentation fournie, nombreuses librairies disponibles et ecosystème hyper actif.
- Performances irréprochables.

"Go: 90% Perfect, 100% of the time" - bradfitz, 2014

sentryo

Cyber Security for the **Industrial Internet**

Questions

Rémy Mathieu - remy.mathieu@sentryo.net - www.sentryo.net