AnoMark

Détection d'Anomalies dans des lignes de commande à l'aide de Chaînes de Markov

Alexandre Junius

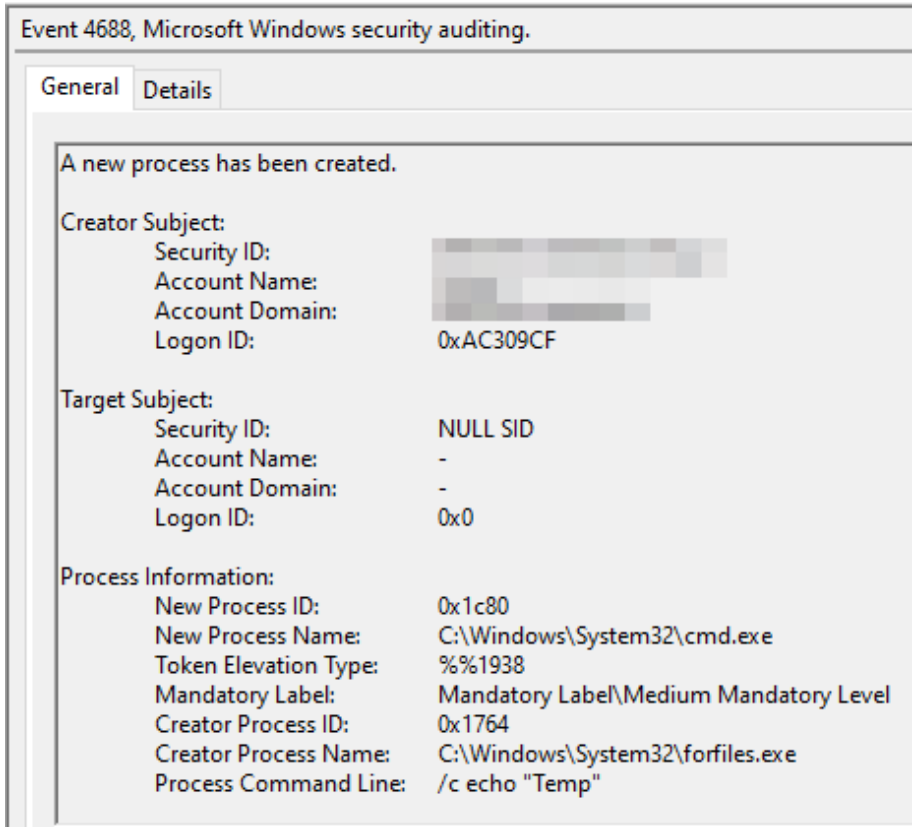
ANSSI

03/06/2022

Sommaire

Sommaire

Un journal d'événement système



Événement 4688 : Création de processus

Méthodes de la détection système

Communément la détection système repose sur :

- La recherche d'IOC (Indicateurs de compromission)
- La création de signatures pour des comportements connus

Méthodes de la détection système

Communément la détection système repose sur :

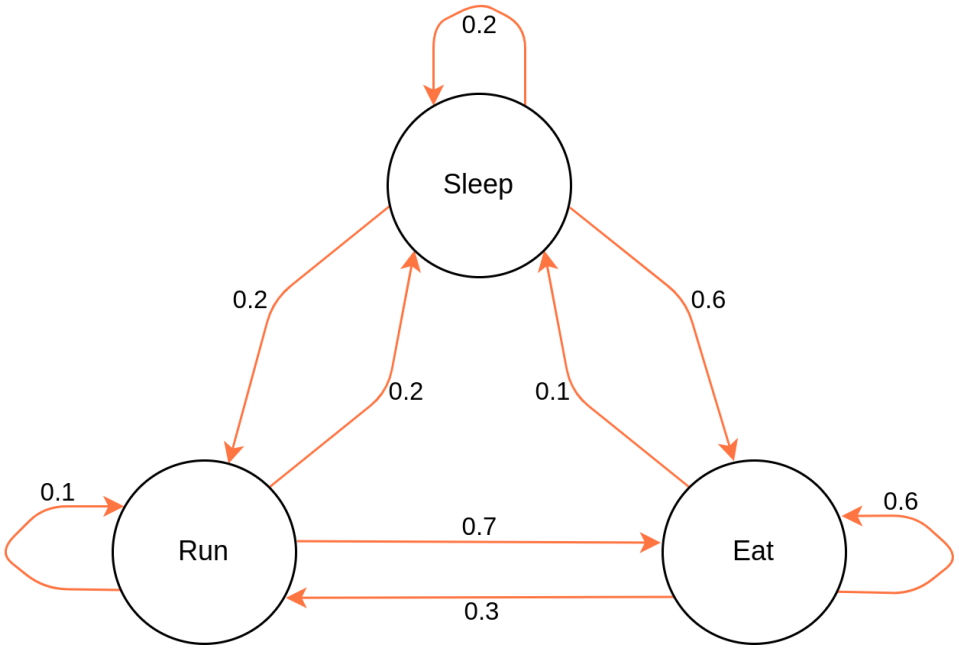
- La recherche d'IOC (Indicateurs de compromission)
- La création de signatures pour des comportements connus

Mais c'est aussi un formidable champ d'application des algorithmes d'apprentissage statistique, notamment en détection d'anomalies, qui permettent de s'orienter vers des comportements inconnus jusqu'alors.

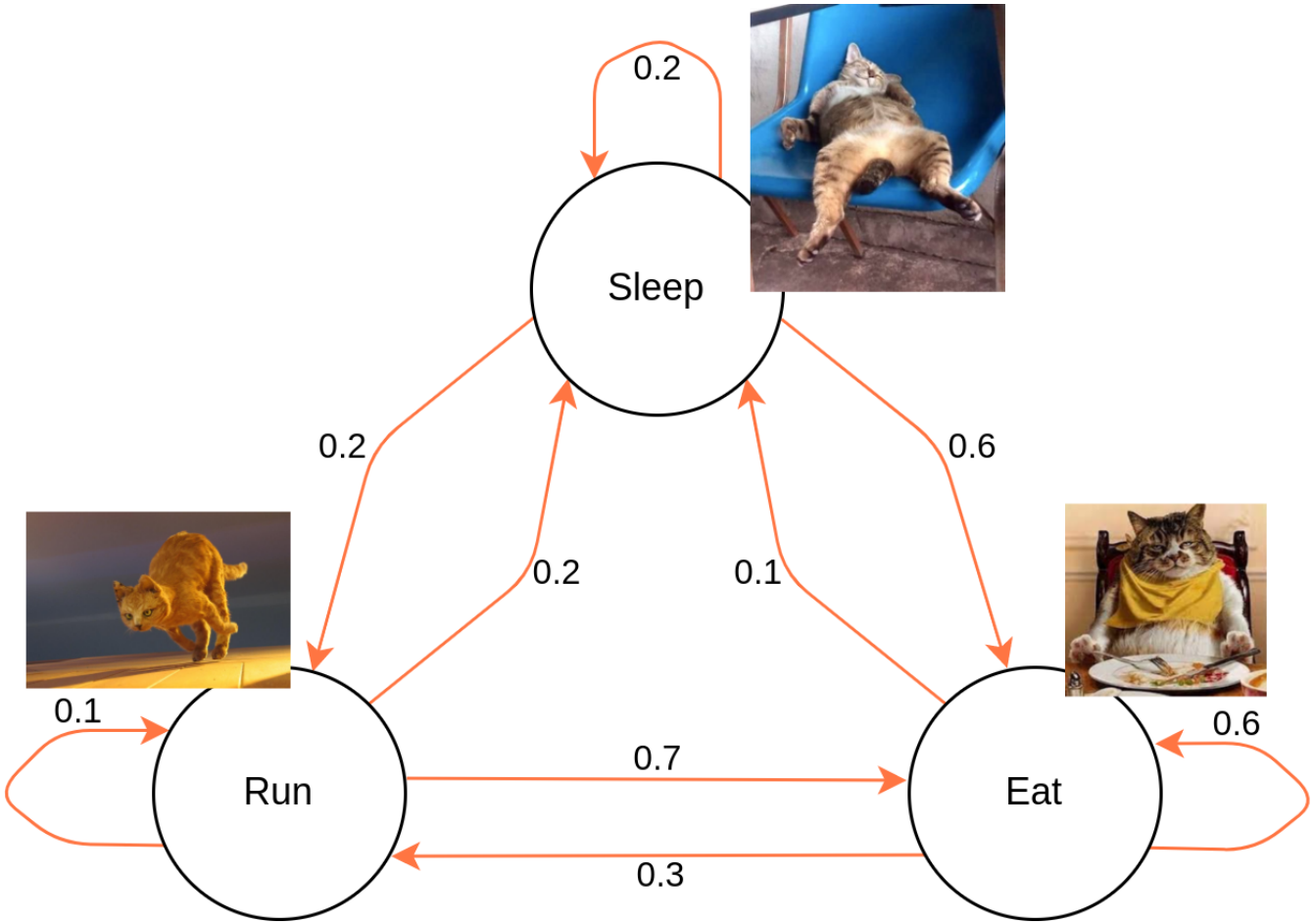
Sommaire

Les chaînes de Markov

L'expression *chaîne de Markov* fait référence à un concept mathématique permettant de **modéliser** les transitions entre états indépendamment du passé. C'est un processus stochastique dont la prédiction du futur à partir du présent n'est pas rendu plus précise par le passé.



Les chaînes de Markov



Les Ngrams de groupes de lettres

On appelle découpage en ngrams des lignes de commandes le fait de découper ces dernières en groupes de n lettres.

» `cmd.exe /c handle.exe`

Modèle :

```
{"cmd.":  {"e":  100%}}
```

Les Ngrams de groupes de lettres

On appelle découpage en ngrams des lignes de commandes le fait de découper ces dernières en groupes de n lettres.

```
» cmd.exe /c handle.exe
```

Modèle :

```
{"cmd.":  {"e":  100%},
"md.e":  {"x":  100%}}
```

Les Ngrams de groupes de lettres

On appelle découpage en ngrams des lignes de commandes le fait de découper ces dernières en groupes de n lettres.

```
» cmd.exe /c handle.exe
```

Modèle :

```
{"cmd.":  {"e":  100%},
"md.e":  {"x":  100%},
"d.ex":  {"e":  100%} }
```

Les Ngrams de groupes de lettres

etc.

Les Ngrams de groupes de lettres

On appelle découpage en ngrams des lignes de commandes le fait de découper ces dernières en groupes de n lettres.

```
» cmd.exe /c handle.exe
» cmd.jar /c something.exe
```

Modèle :

```
{"cmd.":  {"e":  50%, "j":  50%},
"md.e":  {"x":  100%},
"d.ex":  {"e":  100%},
...  }
```

Application

Étapes de l'application de l'algorithme :

- Entraînement modèle en *mode lettres*
- Parcours de lignes de commande avec une fenêtre glissante de taille n
- Produit des probabilités pour obtenir la vraisemblance de la commande en entier
- Classement de la moins vraisemblable à la plus vraisemblable, après liste blanche.

Exemples de lignes de commande détectées

On détecte par exemple :

- les lignes de commandes encodées :

```
» powershell -EncodedCommand Q29uY2VudHJlLnRvaS5zdXIubWEucHJlc2VudGF0aW9uIQ==
```

- les *ping* vers des domaines inhabituels :

```
» ping heeeeeeeey.com
```

- l'exécution de processus inconnus :

```
» iWillPawnYou.exe /user adminAccount
```

Exemples de lignes de commande détectées

Mais aussi :

- l'utilisation de *flags* inconnus :

» `legit.exe -newflag newdata`

- le changement de quelques lettres :

» `CmD.eXe -someflag -someparam`

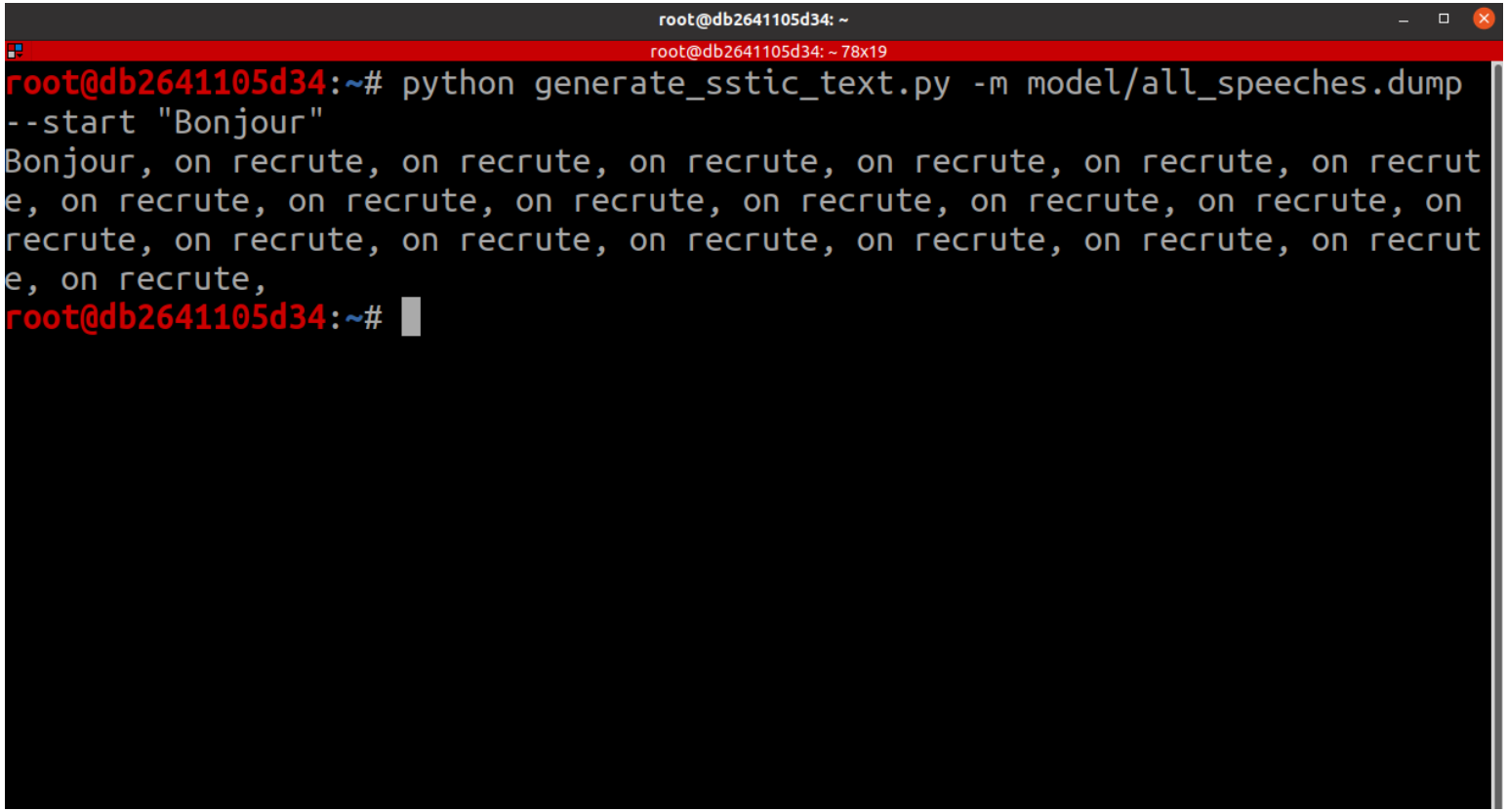
- l'exécution de processus connus depuis des chemins inconnus :

» `C:\newfolder\myproc.exe`

Démonstration

```
root@db2641105d34:/opt/anomark# python apply_model.py -d data/testing_data.csv -c CommandLine -m models/demo_model.dump --verbose --color -n 20
Applying model to dataframe
100%|| 348/348 [00:00<00:00, 4237.24it/s]

Displaying top 20

net view \\.

hostname

cmd /C "dir C:\"

powershell.exe  -NoP -NoL -sta -NonI -W Hidden -Exec Bypass -Enc JABQAHIAbwBnAHIAZQBzAHMAUAByAGUAZgBlAHIAZQBuAGMAZQA9ACIAUwBpAGwAZQBuAHQAbAB5AEMAbwBuAHQAaQBuAHUAZQAiADsA
UgBlAGcAaQBzAHQAZQByAC0AUwBjAGgAZQBkAHUAbABlAGQAVABhAHMAawAgACcAQQBkADAAYgBlAFUAcABkAGEAdABlACcAIAAtAEkAbgBwAHUAdABPAGIAagBlAGMAdAAgACgATgBlAHcALQBTAGMAaABlAGQAdQBsAGUAZ
ABUAGEAcwBrACAALQBBAGMAdABpAG8AbgAgACgATgBlAHcALQBTAGMAaABlAGQAdQBsAGUAZABUAGEAcwBrAEEAYwB0AGkAbwBuACAALQBFAHgAZQBjAHUAdABlACAAJwBwAG8AdwBlAHIAcwBoACAAbABsAC4AZQB4AGUAJw
AgAC0AQQByAGcAdQBtAGUAbgB0ACAAJwAtAE4AbwBuAEkAbgB0AGUAcgBhAGMAdABpAHYAZQAgAC0ATgBvAEwAbwBnAG8AIAAtAE4AbwBQAHIAbwBmAGkAbABlACAALQBGAGkAbABlACAAIgBDADoAXABQAHIAbwBnAHIAYQB
tAEQAYQB0AGEAXABzAHMAaABcAEEAZABvAGIAZQBVAHAAZABhAHQAZQAuAHAAcwAxACIAJwApACAALQBUAHIAaQBnAGcAZQByACAAKABOAGUAdwAtAFMAYwBoAGUAZAB1AGwAZQBkAFQAYQBzAGsAVABByAGkAZwBnAGUAcgAg
AC0ARABhAGkAbAB5ACAALQBBAHQAIAAzAGEAbQApACAALQBTAGUAdAB0AGkAbgBnAHMAIAAoAE4AZQB3AC0AUwBjAGgAZQBkAHUAbABlAGQAVABhAHMAawBTAGUAdAB0AGkAbgBnAHMAUwBlAHQAKQApAA==

cmd /C dir C:

whoami /all

cmd /C wmic logicaldisk get volumename;name

powershell.exe  -NoP -NoL -sta -NonI -W Hidden -Exec Bypass -Enc JABQAHIAbwBnAHIAZQBzAHMAUAByAGUAZgBlAHIAZQBuAGMAZQA9ACIAUwBpAGwAZQBuAHQAbAB5AEMAbwBuAHQAaQBuAHUAZQAiADsA
JABBAGMAdABpAG8AbgAgAD0AIABOAGUAdwAtAFMAYwBoAGUAZAB1AGwAZQBkAFQAYQBzAGsAQQBjAHQAaQBvAG4AIAAtAEUAeABlAGMAdQB0AGUAIAAnAHAAbwB3AGUAcgBzAGgAZQBsAGwALgBlAHgAZQAnACAALQBBAHIAZ
wB1AG0AZQBuAHQAIAAnAC0ATgBvAG4ASQBuAHQAZQByAGEAYwB0AGkAdgBlACAALQBOAG8ATABvAGcAbwAgAC0ATgBvAFAAcgBvAGYAaQBsAGUAIAAtAEYAaQBsAGUAIAAiAEMAOgBcAFAAcgBvAGcAcgBhAG0ARABhAHQAYQ
BcAHMAcwBoAFwAQQBkAG8AYgBlAFUAcABkAGEAdABlAC4AcABzADEAIgAnAA==

cmd.exe /Q /c powershell.exe -NoP -NoL -sta -NonI -W Hidden -Exec Bypass -Enc JABQAHIAbwBnAHIAZQBzAHMAUAByAGUAZgBlAHIAZQBuAGMAZQA9ACIAUwBpAGwAZQBuAHQAbAB5AEMAbwBuAHQAaQB
uAHUAZQAiADsAJABBAGMAdABpAG8AbgAgAD0AIABOAGUAdwAtAFMAYwBoAGUAZAB1AGwAZQBkAFQAYQBzAGsAQQBjAHQAaQBvAG4AIAAtAEUAeABlAGMAdQB0AGUAIAAnAHAAbwB3AGUAcgBzAGgAZQBsAGwALgBlAHgAZQAn
ACAALQBBAHIAZwB1AG0AZQBuAHQAIAAnAC0ATgBvAG4ASQBuAHQAZQByAGEAYwB0AGkAdgBlACAALQBOAG8ATABvAGcAbwAgAC0ATgBvAFAAcgBvAGYAaQBsAGUAIAAtAEYAaQBsAGUAIAAiAEMAOgBcAFAAcgBvAGcAcgBhA
G0ARABhAHQAYQBcAHMAcwBoAFwAQQBkAG8AYgBlAFUAcABkAGEAdABlAC4AcABzADEAIgAnAA== 1> \\127.0.0.1\ADMIN$\__1635524379.4070108 2>&1

whoami

whoami  /groups

cmd.exe /Q /c powershell.exe -NoP -NoL -sta -NonI -W Hidden -Exec Bypass -Enc JABQAHIAbwBnAHIAZQBzAHMAUAByAGUAZgBlAHIAZQBuAGMAZQA9ACIAUwBpAGwAZQBuAHQAbAB5AEMAbwBuAHQAaQB
uAHUAZQAiADsAJABUAGEAcwBrACAAPQAgAE4AZQB3AC0AUwBjAGgAZQBkAHUAbABlAGQAVABhAHMAawAgAC0AQQBjAHQAaQBvAG4AIAAkAEEAYwB0AGkAbwBuACAALQBUAHIAaQBnAGcAZQByACAAJABUAHIAaQBnAGcAZQBy
ACAALQBTAGUAdAB0AGkAbgBnAHMAIAAkAFMAZQB0AHQAaQBuAGcAcwA= 1> \\127.0.0.1\ADMIN$\__1635524379.4070108 2>&1

powershell.exe  -NoP -NoL -sta -NonI -W Hidden -Exec Bypass -Enc JABQAHIAbwBnAHIAZQBzAHMAUAByAGUAZgBlAHIAZQBuAGMAZQA9ACIAUwBpAGwAZQBuAHQAbAB5AEMAbwBuAHQAaQBuAHUAZQAiADsA
```

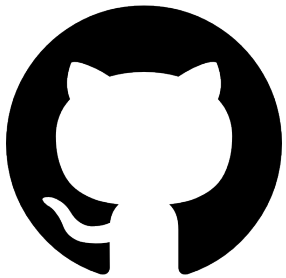
Sur-apprentissage



Sommaire

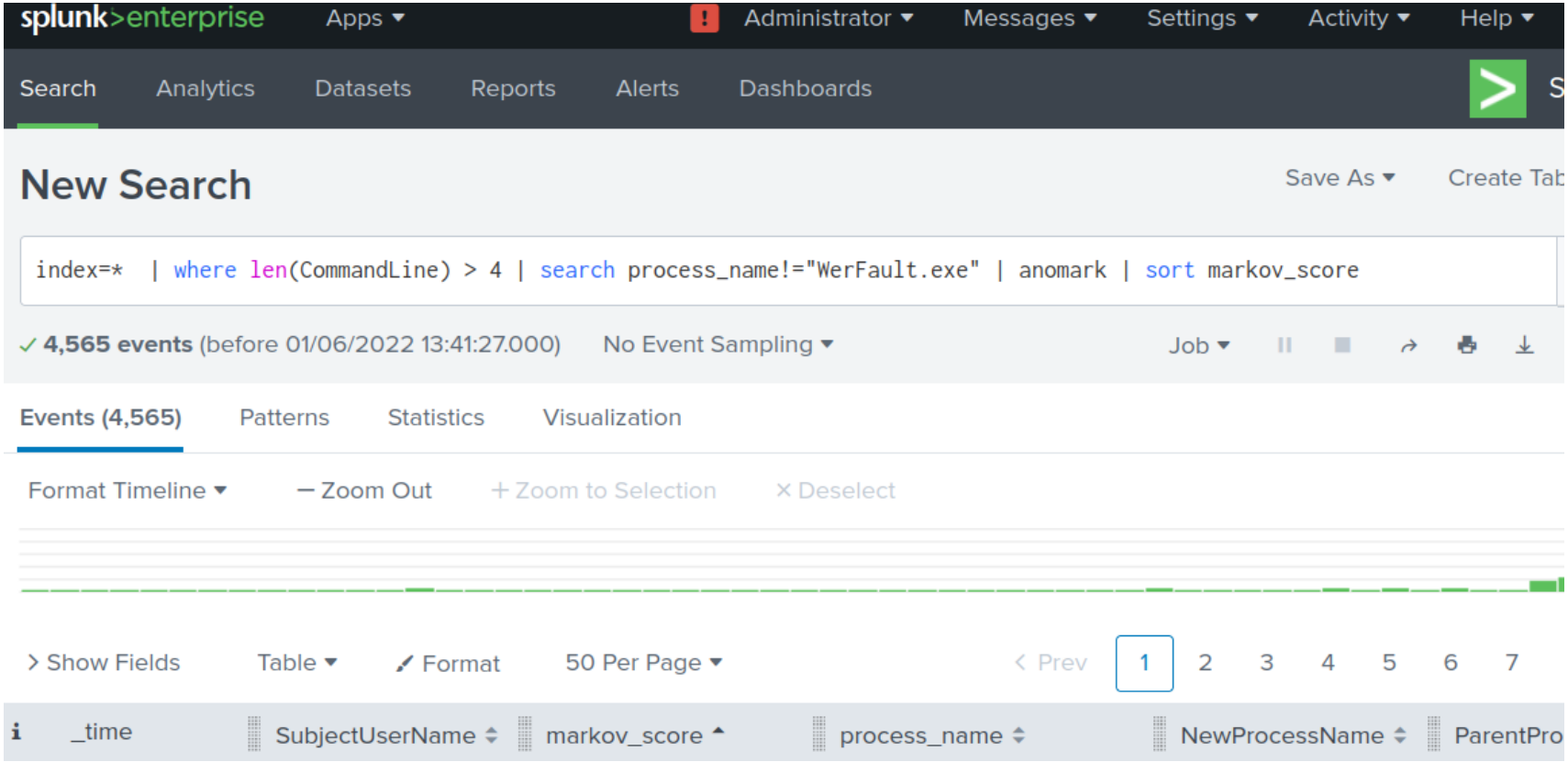
Outil à disposition

- Projet disponible sur le Github ANSSI
- Version écrite en python
- Version *custom command* dans Splunk



splunk>

Custom command Splunk



Conclusion

Cet algorithme nous prouve que l'apprentissage statistique est une source d'informations supplémentaires utile, et il ouvre la voie à d'autres algorithmes de détection d'anomalies, en traitement du langage ou pour d'autres exemples modélisables par des chaînes de Markov.

Merci pour votre écoute !