

Progettazione e sviluppo di una piattaforma Web per la generazione visuale di query per tecnologie SIEM

Facoltà di Ingegneria dell'informazione, informatica e statistica presso UNITELMA
Corso di Laurea in Informatica

Alessio Giovannini

Responsabile: Angelo Monti

Co-responsabile: Alessio Dalla Piazza



SAPIENZA
UNIVERSITÀ DI ROMA



UnitelmaSapienza
Università degli Studi di Roma

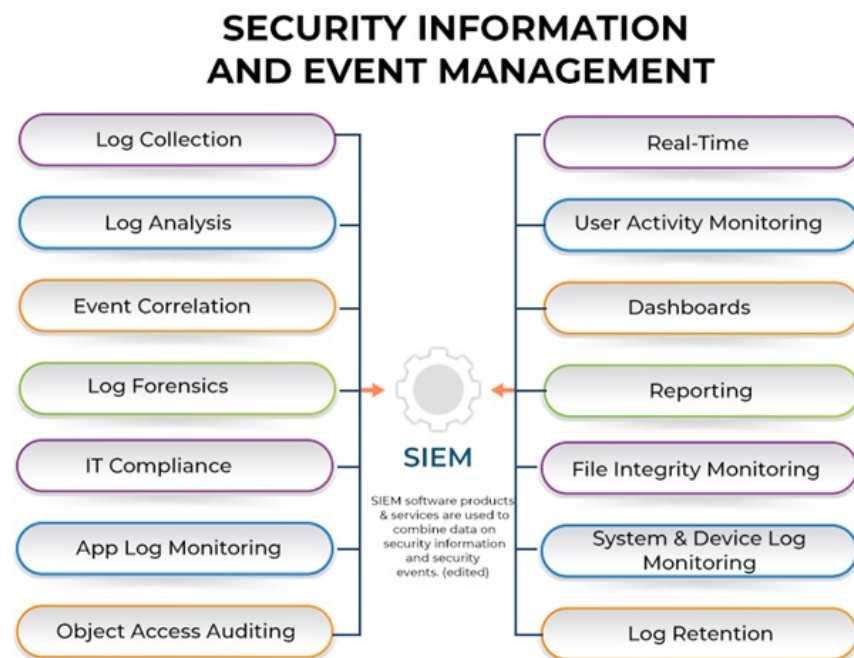
Scenario di riferimento

Scenario di riferimento: Le tecnologie SIEM e il loro ruolo nella sicurezza difensiva

Un **Security Information and Event Management (SIEM)** è una soluzione software di supporto per la rilevazione, analisi e risposta alle minacce alla sicurezza. Un SIEM è composto dalla combinazione di due soluzioni:

- Il **Security Information Management (SIM)**:
 - Raccolta;
 - Monitoraggio;
 - Analisi.
- Il **Security Event Management (SEM)**:
 - Registrazione eventi;
 - Valutazione eventi.

Le tecnologie SIEM vengono spesso adottate all'interno dei SOC, in singolo o combinati con altre tecnologie affini come EDR, IDS e molte altre.



Scenario di riferimento: Security Operation Center (SOC)

Un **Security Operation Center (SOC)** rappresenta un'unità che si occupa di tenere in sicurezza la rete dell'azienda presso cui opera, monitorando e rispondendo agli incidenti di Cybersecurity.

All'interno di un SOC sussistono diversi ruoli con diversi gradi di responsabilità:

- **Triage Specialist** (1° Livello);
- **Incident Responder** (2° Livello);
- **Threat Hunter** (3° Livello);
- **SOC Manager**.



Scenario di riferimento: necessità della piattaforma

- Lo scopo di rendere facile ed accessibile uno strumento come **sigma** attraverso un unico portale web:
 - Generando graficamente delle regole e generare query SIEM specifiche;
 - Testing di regole già pronte;
 - Aiuto automatico nella generazione delle regole.
- Esonerare l'utente finale da problematiche relative a:
 - Preparazione dell'ambiente di lavoro;
 - Gestione delle dipendenze imposte dal software originale;
 - Apprendimento nell'uso dello strumento originale disponibile solo da riga di comando (CLI).

Analisi dei requisiti

Analisi dei requisiti: requisiti della piattaforma

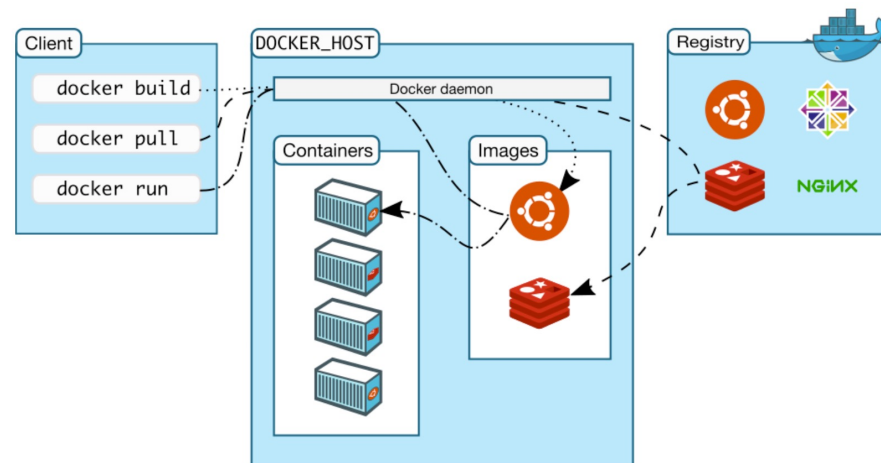
- Creazione della regola;
- Caricamento di una regola;
- Salvataggio della regola;
- Controllo della validità sintattica della regola;
- Generazione della query a partire dalla regola.

Funzionalità essenziali per permettere la corretta generazione di una query per tecnologia SIEM specifica con relativa configurazione

Analisi dei requisiti: raffinamento e funzionalità aggiuntive

Approccio basato sui container Docker

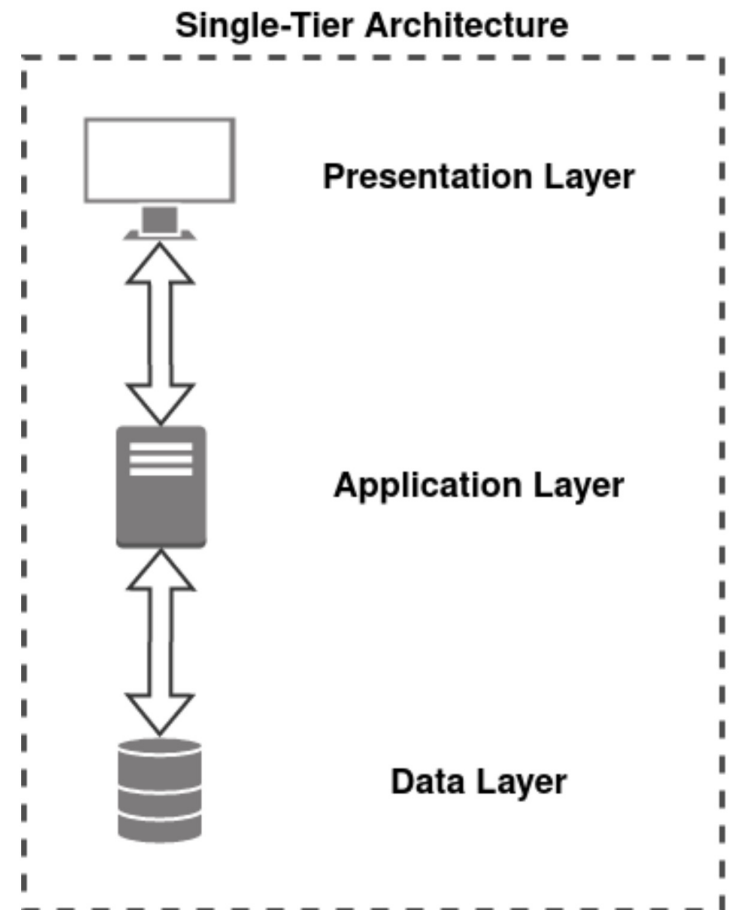
- Isolamento delle risorse e maggiore sicurezza;
- Dipendenze software soddisfatte all'interno del container;
- Portabilità dell'applicativo verso qualsiasi piattaforma che supporti Docker.



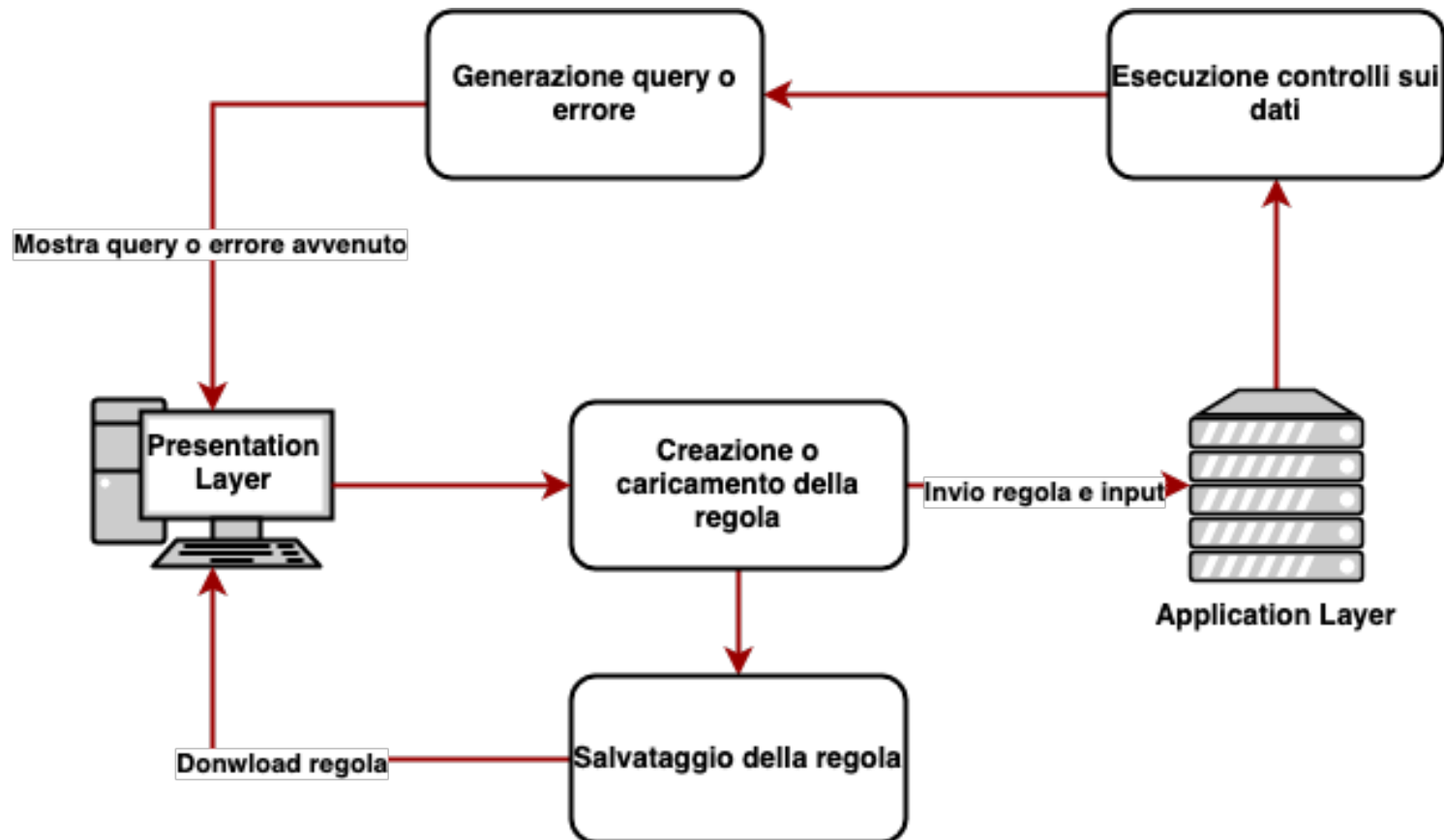
Progettazione

Progettazione: architettura

- **Presentation Layer (User Interface).** Si occupa di organizzare e mostrare i risultati forniti dall'applicativo tramite un'interfaccia che l'utente che utilizza l'applicativo può comprendere;
- **Application Layer (Backend Login).** Si occupa della logica dell'applicativo e ha lo scopo di processare i dati tra il Presentation Layer e il Data Layer;
- **Data Layer (Database).** Si occupa della conservazione delle informazioni e si occupa di mandare quest'ultima all' Application Layer.

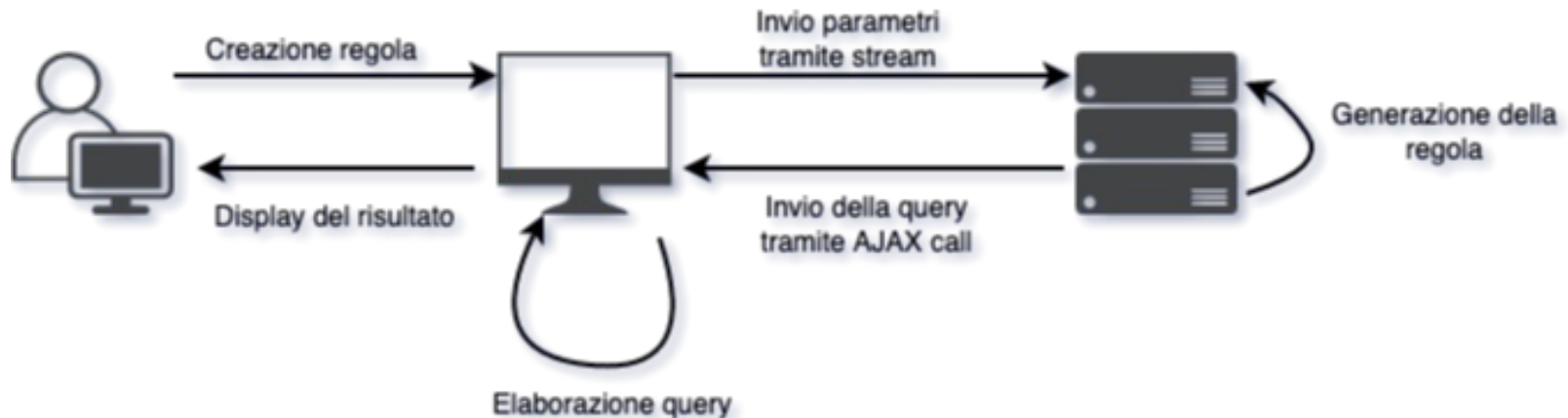


Progettazione: architettura



Progettazione: Approccio basato su stream

- Azzeramento delle operazioni I/O su disco;
- Maggiore velocità di esecuzione;
- Richieste non bloccanti.



Implementazione

Implementazione: Application Layer

- **Parte logica ed applicativa della piattaforma:** controllo sui dati forniti dall'utente, vincoli derivanti dai requisiti e implementazione delle funzionalità;
- Trasmissione dei dati mediante l'utilizzo degli **stream**;
- **Core di sigma modificato** per interfacciare l'applicativo al **server web**;



- **Flask & moduli python custom**, framework che permette la creazione e la gestione di un server web out of the box.



Implementazione: Application Layer

- Definizione delle **route** per definire i percorsi raggiungibili via richieste HTTP/S;
- Definizione del metodo di interfaccia tra il server web e il **core di sigma modificato**.

```
1 from flask import Flask, render_template, url_for, request
2 import custom_sigmac

4 # The name of Flask Application
5 sigma_app = Flask(__name__)

7 #
8 -----

9 # ROUTES FOR REACHABLE PAGES

11 # Homepage
12 @sigma_app.route("/", methods=["GET"])
13 @sigma_app.route("/homepage", methods=["GET"])
14 @sigma_app.route("/index", methods=["GET"])
15 def homepage():
16     return render_template("index.html")

18 # Documentation page
19 @sigma_app.route("/doc", methods=["GET"])
20 @sigma_app.route("/documentation", methods=["GET"])
21 def doc():
22     return render_template("documentation.html")

24 #
25 -----

26 # ROUTES FOR ERROR HANDLING

28 # 404 error handler
29 @sigma_app.errorhandler(404)
30 def error_handler(error):
31     return render_template("404.html")

33 #
34 -----

35 # ROUTES FOR BACKEND FUNCTIONALITIES (NO REACHABLE HTML PAGES)
36 @sigma_app.route("/convert", methods=["POST"])
37 def sigmaconverter():

39     srule = str(request.form.get('srule'))           # the sigma
40     target = request.form.get('target')             # this field go
41     with '-t' option                                # this field go
42     config = request.form.get('config')             # this field go
43     with '-c' option
```

Implementazione: Application Layer

```
43     # Check for missing parameters that are required for sigma
basic usage
44     if(srule==None or target==None or config==None):
45         missing_params = []
46         error = "<h1>The following parameter/s missing:\n"
47         if srule==None:
48             error += "Sigma Rule\n"
49         if target==None:
50             error += "Target\n"
51         if config==None:
52             error += "Configuration\n"
53         error += "</h1>"
54
55         return error
56
57     result = custom_sigmac.CustomSigmac([str(srule)], target, [
config+".yaml"]).convert()
58     print("RESULT: ",result)
59     return result #DEBUG
60
61 #
62 -----
63
64 # Use this for DEVELOPMENT DEPLOYMENT
65 if __name__ == "__main__":
66     sigma_app.run(debug=True) # Use this command for DEVELOPMENT
DEPLOYMENT
67
68 # Use the next commands for PRODUCTION DEPLOYMENT
69 #     from waitress import serve
70 #     serve(sigma_app, host="0.0.0.0", port=8080)
```


Implementazione: Application Layer

- Implementazione del modulo **io.StringIO** di python per la creazione e gestione degli stream.

```
1...
2 f = io.StringIO(str(sigmafile))
3 parser = SigmaCollectionParser(f, sigmaconfigs, rulefilter, "
    sigmafile")
4 backend.setYmlFileName(str(sigmafile))
5 results = parser.generate(backend)
6...
```

Implementazione: Presentation Layer

- **Interfaccia utente**, pagine web che consentono la comunicazione tra l'utente finale e le funzionalità dei precedenti moduli;
- **Javascript**, linguaggio di scripting orientato agli oggetti e agli eventi che viene utilizzato nella programmazione web lato client per creare controlli sulle pagine o effetti dinamici;
- **BULMA**, framework CSS per design responsive e javascript-free;
- **SASS**, estensione di CSS che aggiunge variabili, funzioni e una migliore struttura.



Implementazione: Presentation Layer

Editor

title: Generic Title
description: Generic Description
logsource:
detection:
selection:
condition: selection
level: informational

Add List Node

Add Inline Node

Remove Selected Node

Suggestions

Your query will appear here

Select SIEM solution

Select configuration

Check Rule

Generate query

Clear

Download

Browse...

No file selected.

Upload

SIEMqg by CYS4 srl. The source code is licensed MIT. The website content is licensed CC BY NC SA 4.0.

Implementazione: Presentation Layer

SIEMqg Documentation

How to use the Editor

Create your own rule

SIEMqg thanks to the visual editor allows you to compose custom YAML rule for sigma, with the buttons present at the bottom of the editor it is possible to insert, delete and modify nodes. Moreover, the suggestions section helps you in populate the tree without effort

Download generated rule

With the use of **Download** button you can download the rule you have generated thanks to the editor and save it locally on your machine.

Upload your rule

With the use of **Upload** button it is possible upload a well written YAML rule for sigma directly into the editor.

Rule Creation

Sigma is a very flexible standard with many optional fields. This guide will help you create a Sigma rule that aligns with the other community rules in our repository.

Rule Template

The best way is to use an existing rule that gets close to what you plan like to write. Make sure that the following fields are set in a rule that you would like to push to our public repository:

```
title: a short capitalised title with less than 50 characters
id: generate one here https://www.uuidgenerator.net/version4
status: experimental
description: A description of what your rule is meant to detect
references:
  - A list of all references that can help a reader or analyst understand the meaning of a triggered rule
tags:
  - attack.execution # example MITRE ATT&CK category
  - attack.t1059 # example MITRE ATT&CK technique id
  - car.2014-04-003 # example CAR id
author: Michael Haag, Florian Roth, Markus Nels # example, a list of authors
date: 2018/04/06 # Rule date
logsource: # important for the field mapping in predefined or your additional config
files
  category: process_creation # In this example we choose the category 'process_creation'
  product: windows # the respective product
detection:
  selection:
    fieldName: 'StringValue'
    fieldName: IntegerValue
    fieldName|modifier: 'Value'
  condition: selection
fields:
  - fields in the log source that are important to investigate further
# example rule
```

Implementazione: Presentation Layer

CYS4

SIEMqg Documentation

Editor

title: Generic Title

description: Generic Description

logsource:

product: windows

▼ Select SIEM solution

sqlite

es-qs

elastalert

crowdstrike

athena

humio

splunk

qualys

limacharlie

kibana-ndjson

es-dsl

datadog-logs

fortisiem

streamalert

qradar

es-rule-eql

es-qs-lr

logiq

sql

Select SIEM solution ▼

cmd.exe

Add Inline Node

Remove Selected Node

Suggestions

product: custom value

service : custom value

category: custom value

Generate query

Clear

Download

Check Rule

Browse... No file selected.

Upload

SIEMqg by CYS4 srl. The source code is licensed MIT. The website content is licensed CC BY NC SA 4.0.

Implementazione: Presentation Layer

CYS4

SIEMqg Documentation

Editor

title: Generic Title
description: Generic Description
logsource:
product: windows
detection:
selection:
Image|endsWith: cmd.exe
condition: selection
level: informational

Add List Node

Add Inline Node

Remove Selected Node

Suggestions

product: custom value
service : custom value
category: custom value

Your query will appear here

splunk

Generate query

Clear

Select configuration
elk-defaultindex
elk-defaultindex-filebeat
elk-defaultindex-logstash
elk-linux
elk-windows
elk-winlogbeat
elk-winlogbeat-sp
✓ powershell
splunk-windows
splunk-windows-index
splunk-zeek
sysmon
windows-audit
windows-services

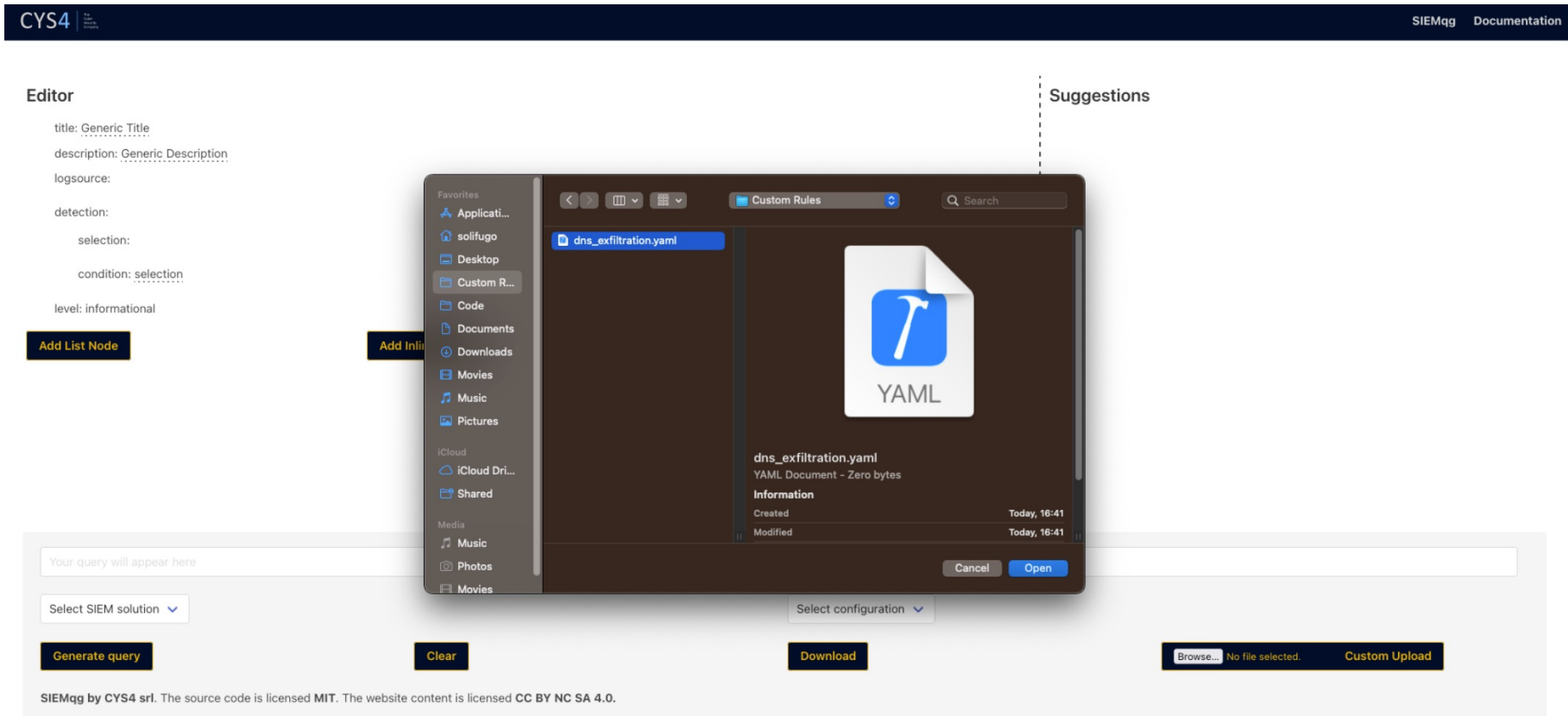
Download

Check Rule

Browse... No file selected. Upload

SIEMqg by CYS4 srl. The source code is licensed MIT. The website content is

Implementazione: Presentation Layer



Implementazione: Presentation Layer

CYS4

siemmqg_rule_download.yml
Completed — 181 bytes
Show all downloads

Editor

title: Generic Title

description: Generic Description

logsource:
product: windows

detection:
selection:
ImageEndswith: cmd.exe

condition: selection

level: informational

Add List Node

Add Inline Node

Remove Selected Node

Suggestions

product: custom value

service : custom value

category: custom value

Your query will appear here

Select SIEM solution

Select configuration

Generate query

Clear

Download

Upload

SIEMqg by CYS4 srl. The source code is licensed MIT. The website content is licensed CC BY NC SA 4.0.

Implementazione: Presentation Layer

CYS4

SIEMqg Documentation

Editor

title: Telegram Bot API Request

description: Detects suspicious DNS queries to api.telegram.org used by Telegram Bots of any kind

logsource:

category: dns

detection:

selection:

query: 'api.telegram.org'

condition: selection

level: medium

Add List Node

Add Inline Node

Remove Selected Node

Suggestions

product: custom value

service : custom value

category: custom value

query="api.telegram.org"

splunk

sysmon

Check Rule

Generate query

Clear

Download

Browse... No file selected. Upload

SIEMqg by CYS4 srl. The source code is licensed MIT. The website content is licensed CC BY NC SA 4.0.

Implementazione: Presentation Layer

```
1 //Send the proper header information along with the request
2 http.setRequestHeader('Content-type', 'application/x-www-form-
    urlencoded');

4 http.onreadystatechange = function() { //Call a function when the
    state changes.
5     if(http.readyState == 4    http.status == 200) {
6         document.getElementById("qresult").value = http.responseText;
7     }
8     else if(http.readyState == 4    http.status == 500){
9         alert("The configuration selected is not currently implemented
            correctly in sigma. Please use different configuration...");
10    }
11 }
12 http.send(params);
13 }
```

Implementazione: Dockerfile

```
1 FROM python:3.8.15-bullseye

3 COPY . /sigma_app
4 RUN git clone https://github.com/SigmaHQ/sigma.git
5 RUN mv /sigma/tools/sigma /usr/local/lib/python3.8/site-packages/
6 RUN rm -rf sigma
7 RUN pip install coverage==5.0 yamllint==1.21 elasticsearch==7.6
    elasticsearch-async==6.2 pytest==5.4 colorama setuptools stix2
    attackcti pipfile requests==2.25 urllib3==1.26 progressbar2==3
    .47 pymisp==2.4.123 PyYAML==5.1 ruamel.yaml flask waitress

9 EXPOSE 5954

11 CMD ["python", "/sigma_app/src/sigma_server.py"]
```

Sviluppi futuri

Sviluppi futuri

- Integrazione con YARA;
- Autocompletamento delle regole in fase di creazione
- Autocorrezione semantica delle regole create
- Supporto alle API di differenti tecnologie SIEM
- Valutazione di tecnologie server web alternative

Progettazione e sviluppo di una piattaforma Web per la generazione visuale di query per tecnologie SIEM

Facoltà di Ingegneria dell'informazione, informatica e statistica presso UNITELMA
Corso di Laurea in Informatica

Alessio Giovannini

Responsabile: Angelo Monti

Co-responsabile: Alessio Dalla Piazza



SAPIENZA
UNIVERSITÀ DI ROMA



UnitelmaSapienza
Università degli Studi di Roma