Anomaly Detection User Manual

SPHINX

A Universal Cyber Security Toolkit for Health-Care Industry

Table of contents

1 Introduction 3
2 Installation/Deployment 3
2.1 Prerequisites and hardware 3
2.2 Deployment with Docker 3
2.3 Deployment with Kubernetes 3
3 Operation and maintenance 3
3.1 Basic Case Examples 3
3.2 Links with other Components 5
3.3 Outcomes 5
4 Application UI presentation 6

Table of figures

Figure 3.1 Main test menu 4
Figure 3.2 Simulation screen 4
Figure 3.3 Sample message in ad-alert topic 5
Figure 4.1 AD - The main screen 6
Figure 4.2 Algorithm configuration tab 6
Figure 4.3 General Tab 7
Figure 4.4 Configuring the k-means algorithm 8
Figure 4.5 Configuration of the sflow-based algorithm 9
Figure 4.6 C&C BotNets configuration 10

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

1 Introduction

Anomaly Detection is a SPHINX component that raises alerts when anomalous or suspicious activities are detected. AD does not use the raw network data. AD uses as input the logs generated by Data Traffic Monitoring component.

Anomaly detection uses the following types of algorithms:

- k-means-clustering algorithm for analysing HTTP and DNS traffic.
- Statistical algorithms for identifying the following type of issues: SMTP talker identified, Alien accessing too much hosts, UDP amplifier (DDoS), P2P communication, Abused SMTP Server, Media streaming client, DNS Tunnel, ICMP Tunnel, C&C BotNet communication, etc.

2 Installation/Deployment

The installation is based on docker images for deploying AD.

2.1 Prerequisites and hardware

Preconditions:

1. Kafka:
   - optional: kafdrop (for browser based interaction with kafka)
2. Docker image for PostgreSQL
3. Docker image for Sphinx Component ID-UI
4. Docker image for HBase (version: 2.1.3)

Hardware:

1. CPU: CPU like Intel I7
2. RAM: 32GB (of RAM allocated to the Java heap)
3. GPU: Not needed
4. SPACE: 3TB (of raw disk capacity per RegionServer (HBase))

2.2 Deployment with Docker

1. #docker login https://sphinx-repo.intracom-telecom.com/
2. #docker pull registry.sphinx-repo.intracom-telecom.com/sphinx-project/anomaly-detection/ad□deployment:latest
3. #docker run

2.3 Deployment with Kubernetes

AD can be deployed on a K8S cluster using .yml files.

3 Operation and maintenance

Algorithms testing is performed in the "Simulation" (Figure **Figure 3.1**) component. Because this component was created to facilitate algorithm testing, it will not be visible to users.

3.1 Basic Case Examples

Case: Run P2P communication algorithm

Objective:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

The user wants to run the P2P communication algorithm.

Steps:

1. In the Algorithms component, in the tab with the same name, the P2P communication algorithm must be checked. Access the Algorithm option: https://sphinx-kubernetes.intracom-telecom.com/id-ui/ad/algorithms
2. Simulation component contains a list of csv files, which represent input data for algorithms. Access the Simulation option: https://sphinx-kubernetes.intracom-telecom.com/id-ui/ad/simulation

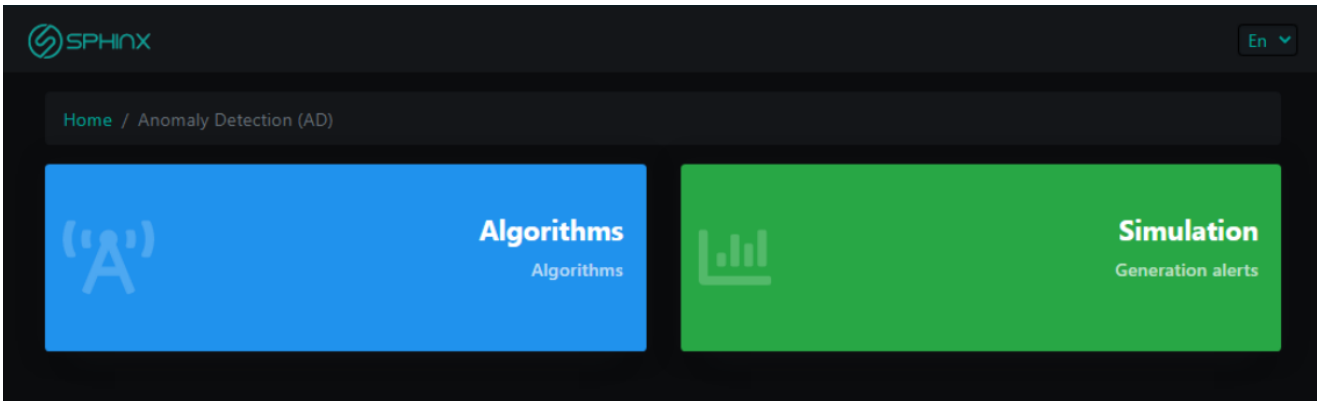

Figure 3.1 Main test menu

3. For the csv file "test_P2PComunication_v2" click the "Simulate" button. When you press this button, the data in the "adml_sflow" table in hbase is deleted, put the new data from the CSV file and run the algorithm (Figure **Figure 3.2**)
4. At the end of the execution the user receives a message with the number of alerts detected.

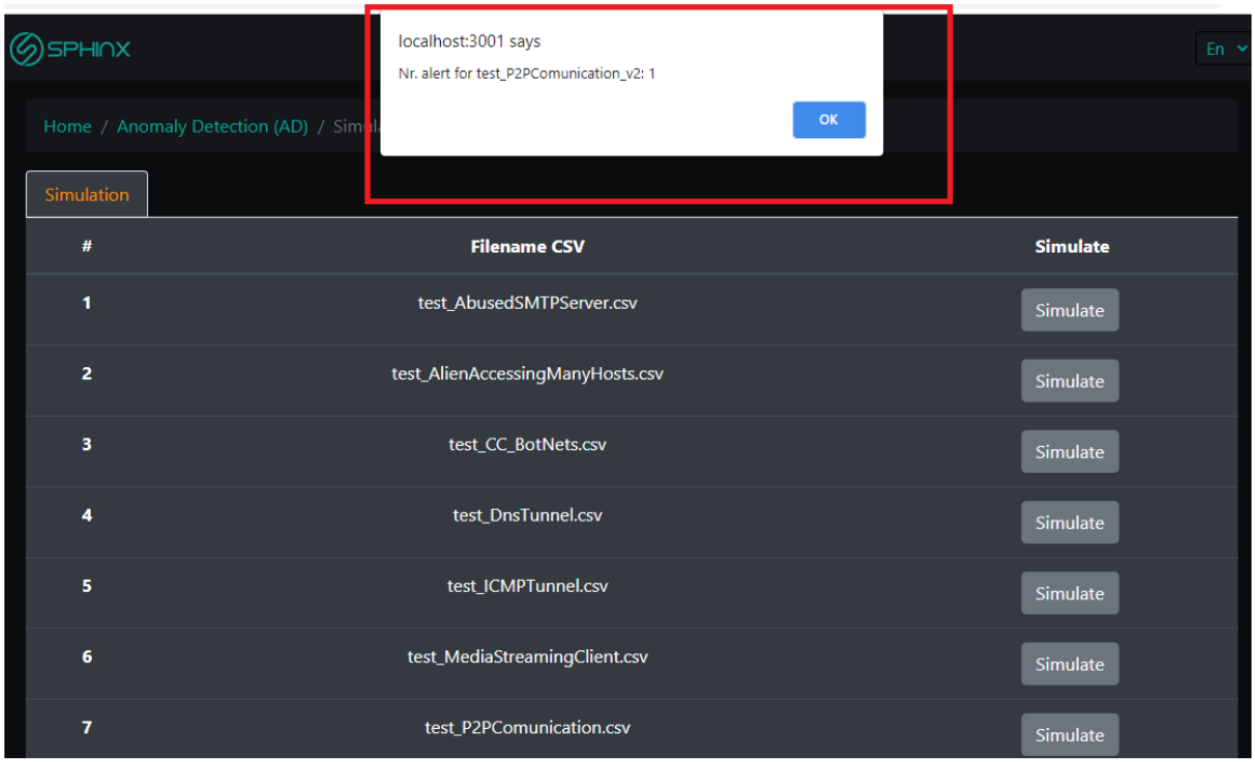

Figure 3.2 Simulation screen

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

3.2 Links with other Components

The AD component publishes messages to Kafka message service:

- ad-alert topic

3.3 Outcomes

Upon completion of these test cases, alerts will be thrown that may be visible in the ad-alert topic (Figure **Figure 3.3**).

Offset 562 Key empty Timestamp 2021-04-26 12:55:56.488 Headers _TypeId_: ro.simavi.sphinx.ad.model.StixAnomalyDetectionAlert
{
"type": "bundle",
"id": "bundle--513fbddd-afd1-48c7-b0ac-5e1a1da34c13",
"objects": [
{
"type": "identity",
"id": "identity--1d9d75dc-16ed-46ae-825a-0fc8aecb3488",
"spec_version": "2.1",
"created": "2021-04-26 12:55:56",
"modified": "2021-04-26 12:55:56",
"name": "AD"
},
{
"type": "x-sphinx-ad-alert",
"id": "x-sphinx-ad-alert--8841c6dd-22a3-498c-b265-65b9396c6d51",
"spec_version": "2.1",
"created": "2021-04-26 12:55:56",
"modified": "2021-04-26 12:55:56",
"details": {
"totalFlows": 0,
"protocolFlow": {
"id": null,
"detectedProtocol": null,
"lowerPort": 0,
"upperPort": 0,
"upperIp": "255.255.255.255",
"lowerIp": "195.82.138.10",
"ipProtocol": 0,
"FlowDuration": 0,
"bytes": 0,
"packets": 0,
"packetsWithoutPayload": 0,
"avgPacketSize": 0,
"minPacketSize": 0,
"maxPacketSize": 0,
"avgInterTime": 0,
"packetSize0": 0,
"interTime0": 0,
"packetSize1": 0,
"interTime1": 0,
"packetSize2": 0,
"interTime2": 0,
"packetSize3": 0,
"interTime3": 0,
"packetSize4": 0,
"interTime4": 0,
"hostname": null,
"dnaType": null,
"timeStamp2": null,
"malware": false,
"Flags": 0
},
"text": "This IP was detected by Hogzilla performing an abnormal activity. In what follows, you can see more information.\nAbnormal beh
"title": "H2: P2P communication",
"flowId": "161944175299",
"coords": null,
"username": null,
"timestamp": "2021-55-26 03:55:56",
"algorithm": {
"type": "P2PCommunication_sFlow",
"numberOfPairs": "7",
"myIP": "195.82.138.10",
"bytesUp": "0",
"bytesDown": "759970752",
"numberPkts": "16",
"stringFlows": "\n195.82.138.10:11200 <?> 192.168.1.5:11200 (TCP, L-to-R: 0 B, R-to-L: 190.7MB,2 pkts, duration: 215s, sampling: 1/
}
}
},
{
"type": "relationship",
"id": "relationship--f31f0c92-17c5-4a1b-9bb4-9ab36b167e1d",
"spec_version": "2.1",
"created": "2021-04-26 12:55:56",
"modified": "2021-04-26 12:55:56"
}
]
}

Figure 3.3 Sample message in ad-alert topic

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

4 Application UI presentation

Figure **Figure 4.1** shows the main screen in AD that contains the Algorithms button.

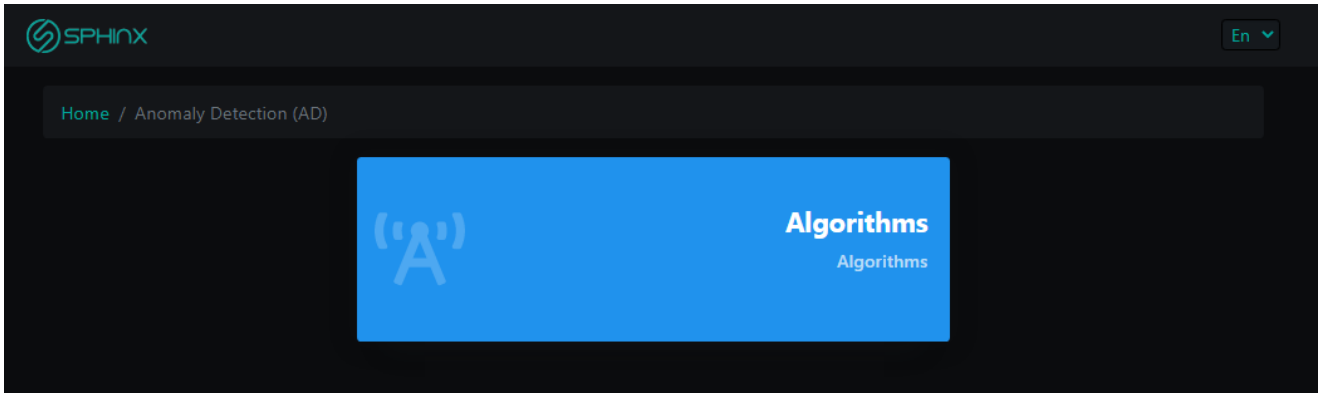

Figure 4.1 AD - The main screen

Algorithm button allows access to the configuration sections for each algorithm used in order to detect anomalies in network traffic. This screen is divided into several tabs. The first tab is used to enable or disable the desired algorithms. The rest of the tabs are used to configure the algorithms listed in the first tab. If one of the algorithm names is pressed, the corresponding tab will open (**Error! Reference source not found.**).

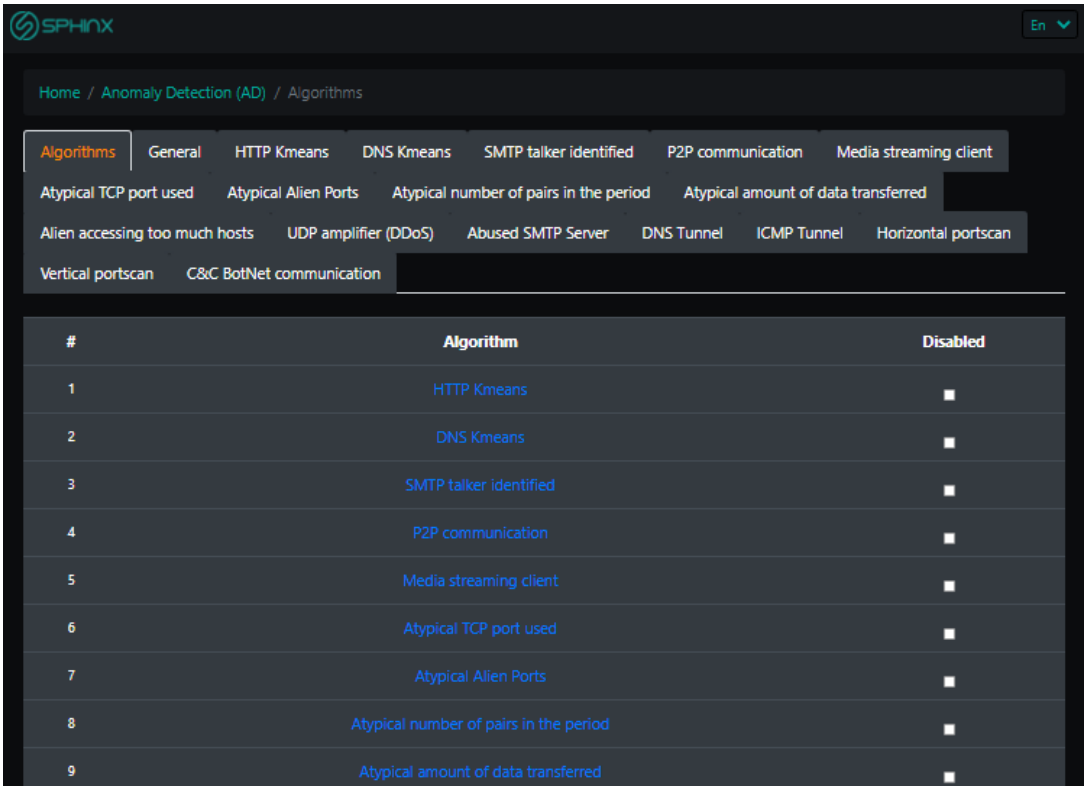

Figure 4.2 Algorithm configuration tab

General Tab is a way to manage general configuration for all algorithms. For a parameter that contains a list of values, each value must be separated by a comma. Optionally, after a value you can add a comment using the (#) sharp symbol, as in the following example: 10.1.1.1#SMTP Server (**Error! Reference source not found.**).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

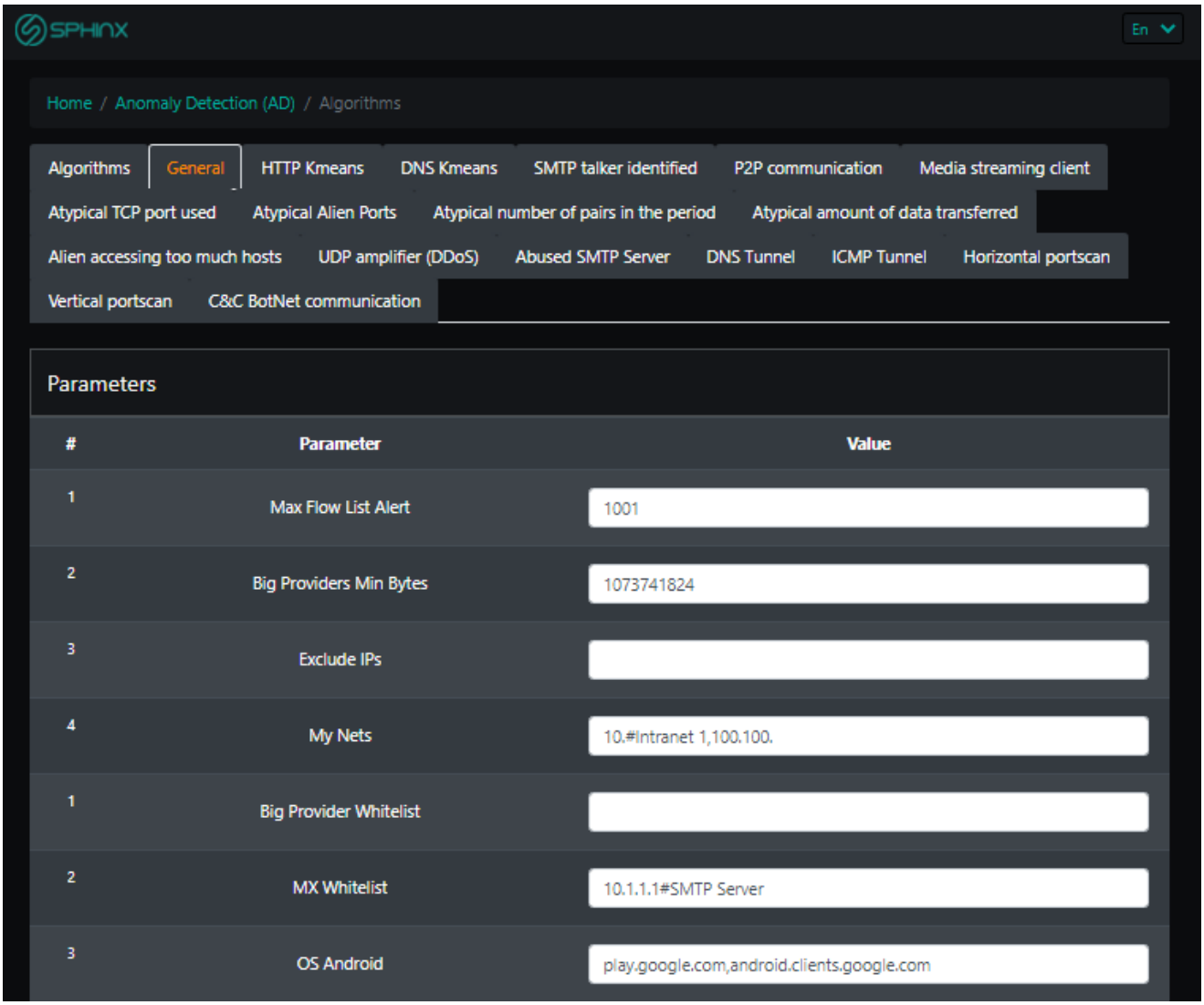

Figure 4.3 General Tab

The algorithms based on machine learning (ML) are managed by:

- HTTP Kmeans-clustering
- DNS Kmeans-clustering

For these algorithms you can configure what parameters and features are used by K-Means algorithm (**Figure 4.4**).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

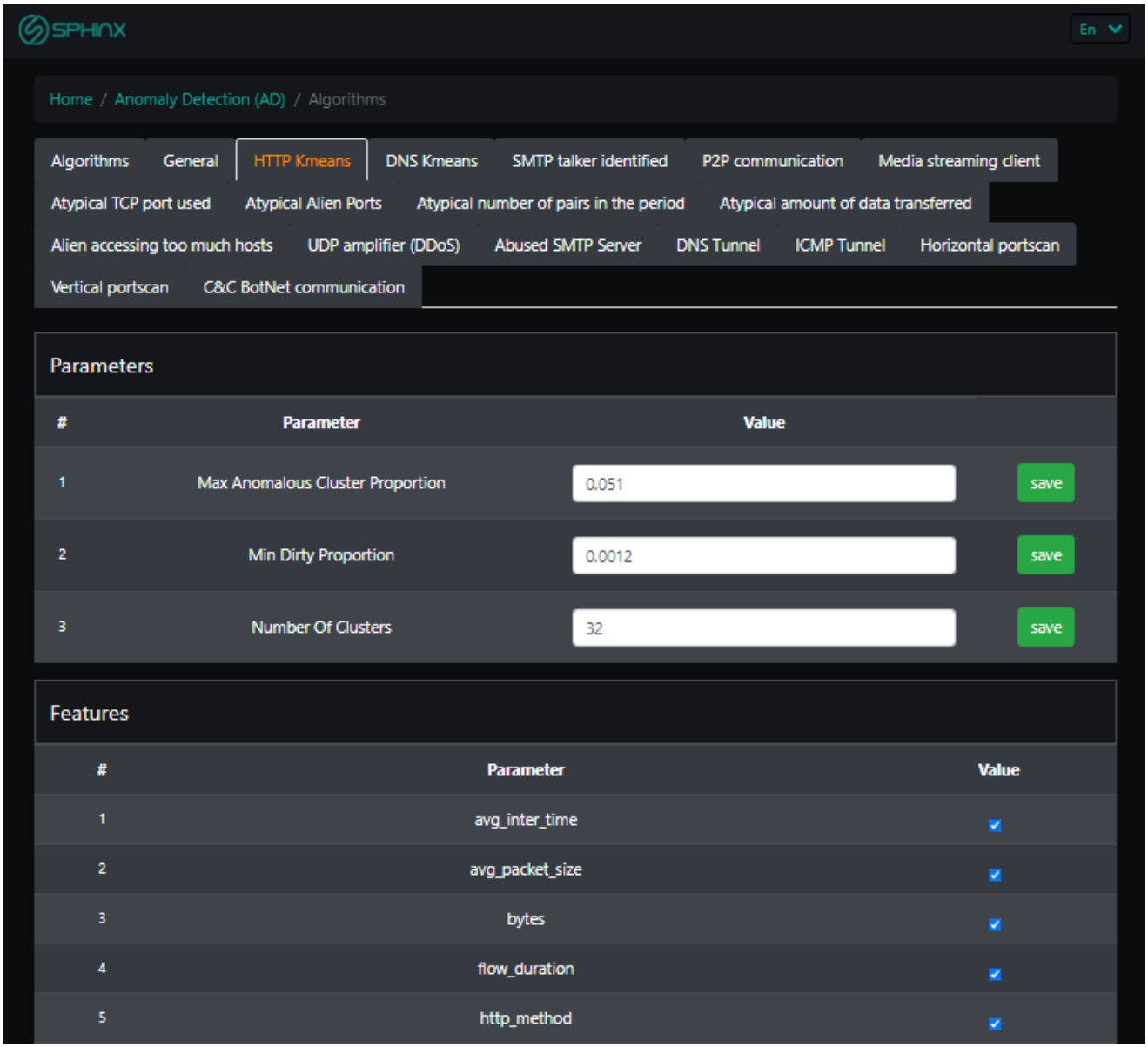

Figure 4.4 Configuring the k-means algorithm

The statistics algorithms are managed by (**Figure 4.5**):

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

1. SMTP talker identified
2. P2P communication
3. Media streaming client
4. Atypical TCP port used
5. Atypical Alien Ports
6. Atypical number of pairs in the period
7. Atypical amount of data transferred
8. Alien accessing too much hosts
9. UDP amplifier (DDoS)
10. Abused SMTP Server
11. DNS Tunnel
12. ICMP Tunnel
13. Horizontal portscan
14. Vertical portscan
15. C&C BotNet communication

All flows used by these algorithms are filtered by protocols:

- TCP
- UDP
- ICMP
- ICMPv6

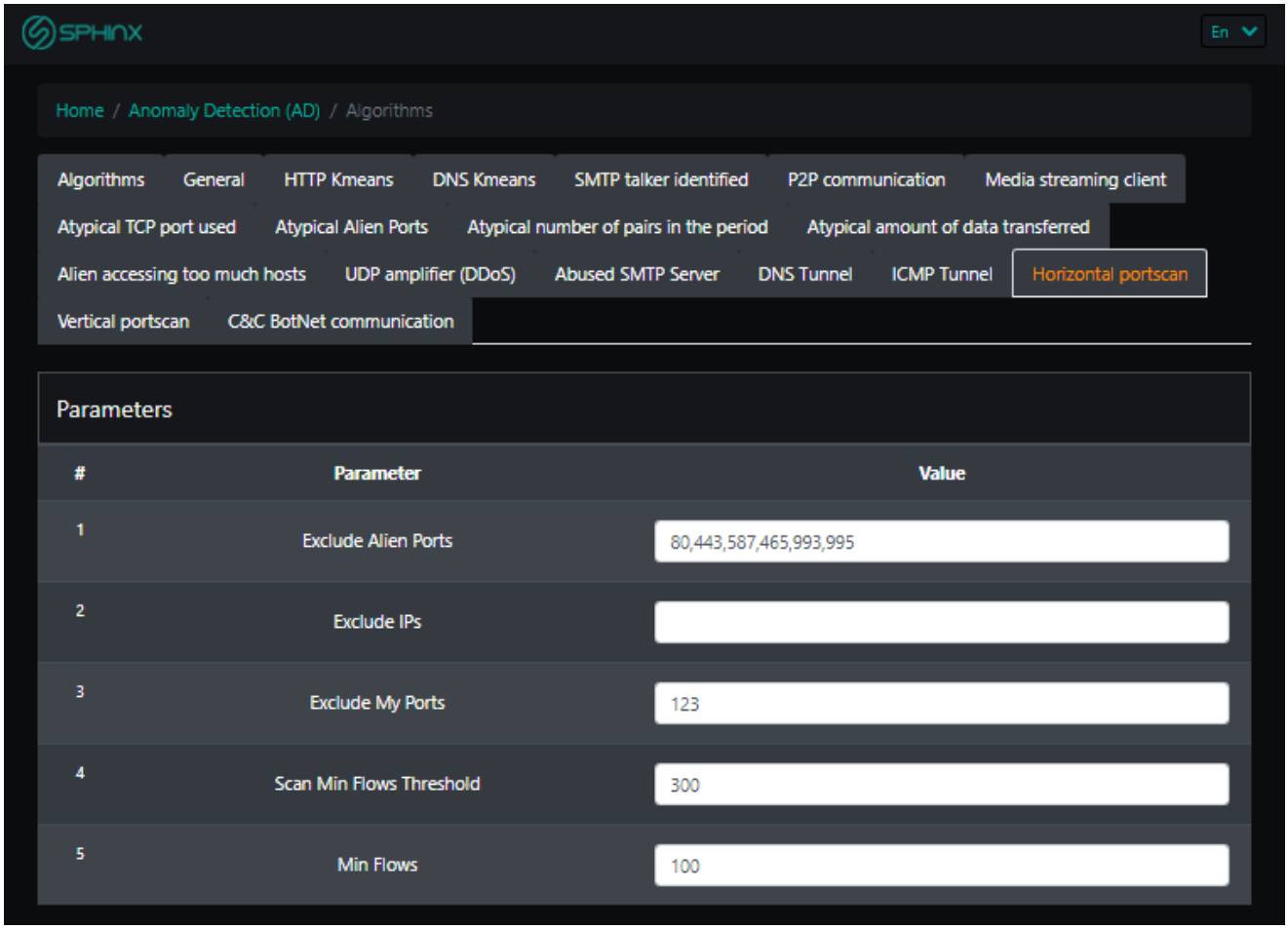

Figure 4.5 Configuration of the sflow-based algorithm

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

C&C BotNets (**Figure 4.6**), for example, alert you if:

- the source port is larger than 1023
- the number of packages is higher than the Min Packets Per Flow parameter (default is 20)
- source ip is not among the excluded IPs (Excluded IPs parameter)
- destination ip is not among the excluded IPs (Excluded IPs parameter)
- destination ip is found in the list of IPs that are found at a certain URL (set via the URL parameter; by default this URL is: https://rules.emergingthreats.net/blockrules/emerging-botcc.rules)

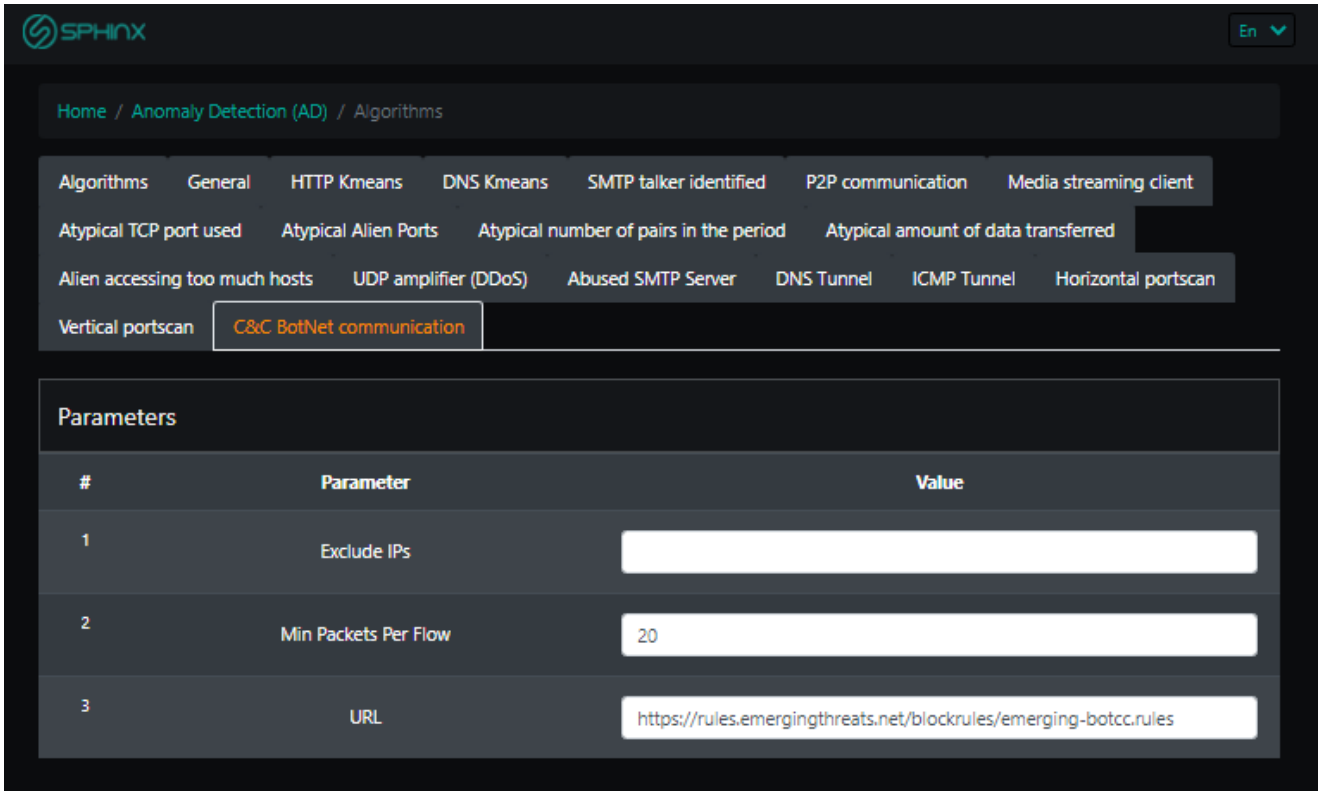

Figure 4.6 C&C BotNets configuration

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.