CyDefSIG: Cyber Defence Signature Sharing Platform, version: 1.1.1

Table of Contents

1. Layout and features

Main page:

The main page lists the events stored on the site. See data structure section for further details.

The **site PGP public key** and **log-out button** are at the bottom of the page and will be accessible in any page of the site.

Left Menu

The left menu allows the user navigating to the different features/pages of the site:

- ***New Event:***
  Allow user to create a new event. See How to share a malware signatures section for further details.
- ***List Events:***
  List all events and allows users to
  - display the details of the events
  - contact the publishing party of an even by clicking **Contact Reporter** button in the Event page.
  - Modify or delete an event and attributes you have imported.
- ***List Attributes:***
  Lists all attributes cross events.
- ***Search Attribute:***
  You can search for attributes based on key words and apply a filtering based on the category and or attribute type.
- ***Export:***
  Different format are supported: XML (all or per event), text (all or per attribute type), and IDS format. Note that only the attributes that have been selected to be in the part of IDS will be included in this latter.
- ***News:***
  Provide the latest news regarding the site like last changes.
- ***My Profile:***
  Allows to setup the user profile:
  - email address to which new events will be sent,
  - the AuthKey used to automate the export of events/attributes from the application (see Export),
  - NIDS starting SID,
  - PGP public key used to encrypt the events sent by email
- ***Member List***
  Provide statstics about the site.
- ***User Guide***
  Displays this document.
- ***Terms & Conditions***
  Defines terms of use of this platform.
- ***List Servers***
  Displays a list of servers that the user synchronizes his account to.

2. How to share a malware/attack attributes

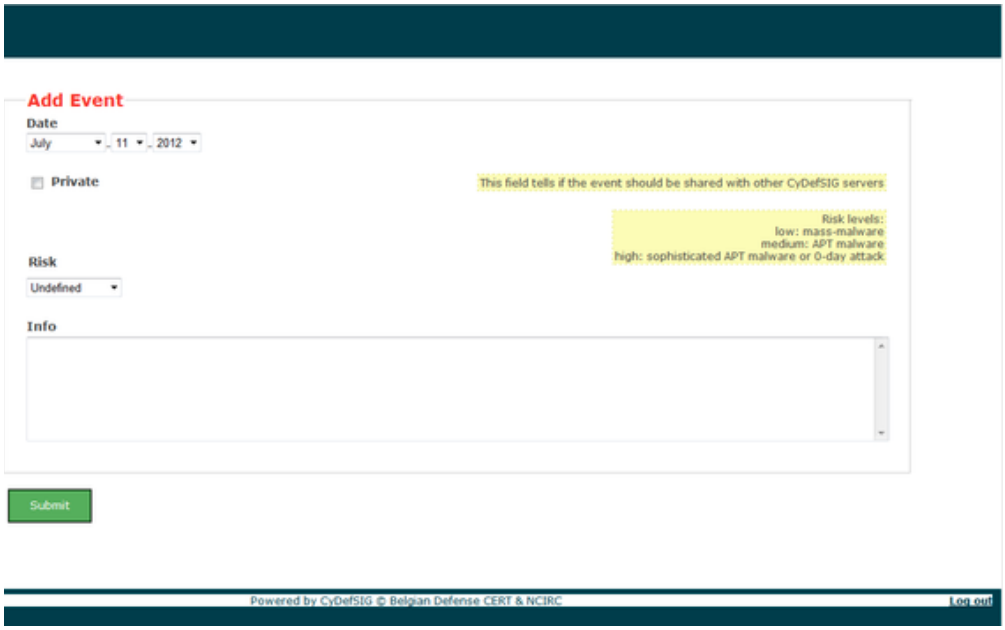
Data structure

The following diagram depicts the data structure to store malware signatures.

- An ***Event*** is a containers that hosts one or more ***attributes*** of a malware. This is the main data structure that host the signatures of a malware. An event is identified by a unique id number automatically assigned by the system.
- An ***Attribute*** is a characteristic of malware that can be used as a descriptor. Attributes are categorised and always linked to an Event via the Event id.

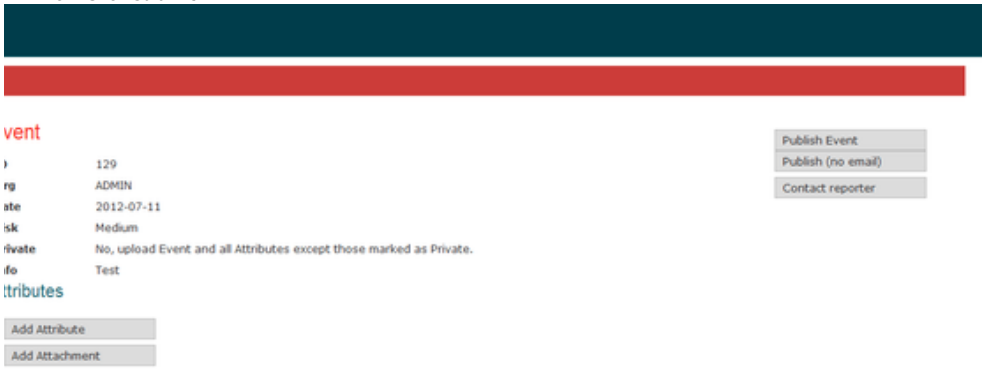
Note that it may happen that different events are related to a same malware or variants as the data may be imported by different groups. The application creates automatically links between events with same attributes.

Sharing malware/attack information steps by steps



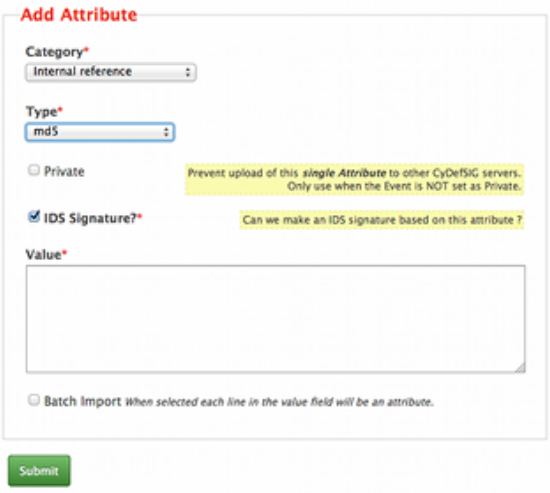
Mandatory fields are marked with *

1. Click on ***New Event*** (left menu)
2. Fill-in the form:
   - ***Date**:* date of the malware was discovered
   - ***Private**:* is the event sharable with other servers. (only in sync-mode)
   - ***Risk**:* estimated risk level related to the malware.
     Guideline for risk level:
     - Undefined (default)
     - Low - TBD
     - Med - Advanced Persistent Threat
     - High - Very sophisticated APT (e.g. including 0-day)
   - ***Info**:* High level information that can help to understand the malware/attack, like title and high level behavior.
     This field should remain as short as possible (recommended max 50 words). The full description of the malware behavior and its artifacts must be defined as an attribute (other).
3. Click ***Submit***

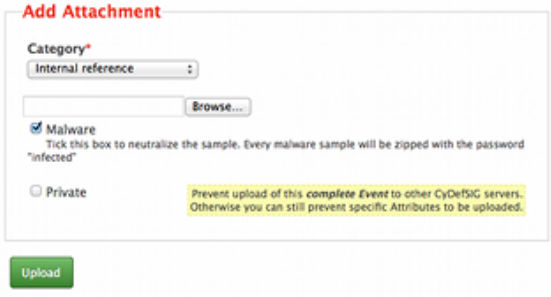


Note that at this stage, the information is shared on the site but no notification is sent to the other parties yet.

4. Click ***Add Attribute*** or ***Add Attachment***
5. For Attribute:
   - ***Category****: see Category section below
   - ***Type****:* see Type section below
   - ***Private****:* prevent upload of this specific Attribute to other servers. (only in sync-mode)
   - ***IDS Signature?***: Check this box if you want the attribute to be part of the IDS signature generated by the site. Make sure that the information in value is usable in an IDS signature, do not check if it is free text, Vulnerability.
   - ***Value:*** enter the attribute value. Note that the value format will be validated for some types like hash and IP addresses.
   - ***Batch Import:*** check this box to import data in batch. Enter an attribute value per line, each entry will be assigned the selected Category and Type.
   - Click ***Submit***



6. For Attachment:
   - ***Category:*** see Category section below
   - Select the file to upload
   - ***Malware:*** Check this box if the file to upload is harmful. The system will then encrypt with zip before storing the file with the default password, ***"infected"***. This will protect other systems against accidental infection. Note that a hash will be automatically computed and added to the event as an attribute.
   - Click ***Upload***



7. Redo steps 5-6 as many time as attributes you need to upload.
8. Click ***Publish Event*** once all attributes are uploaded.
   The application will then send the event with all uploaded information to all users of the site.
   In sync-mode the event will also be uploaded to other servers users have configured in their profile.

   You can modify, delete or add new attributes after publishing. In that case, any change will be accessible by other users via the GUI and only released by email to all users once you re-Publish the event.

3. Attribute Categories and Types

Attribute Categories vs Types

Category	Internal reference	Antivirus detection	Payload delivery	Artifacts dropped	Payload installation	Persistence mechanism	Network activity	Payload type	Attribution	External analysis	Other
md5			X	X	X					X	
sha1			X	X	X					X	
filename			X	X	X	X				X	

filename|md5			X	X	X					X	
filename|sha1			X	X	X					X	
ip-src			X				X			X	
ip-dst			X				X			X	
hostname			X				X			X	
domain			X				X			X	
email-src			X								
email-dst			X				X				
email-subject			X								
email-attachment			X								
url			X				X			X	
user-agent			X				X			X	
regkey				X		X				X	
regkey|value				X		X				X	
AS			X				X			X	
snort							X			X	
pattern-in-file			X	X	X		X			X	
pattern-in-traffic			X		X		X			X	
pattern-in-memory				X	X					X	
yara			X	X	X						
vulnerability			X		X					X	
attachment		X	X	X	X		X			X	
malware-sample			X	X	X					X	
link	X	X	X							X	
comment	X	X	X	X	X	X	X	X	X	X	X
text	X	X	X	X	X	X	X	X	X	X	X
other	X	X	X	X	X	X	X	X	X	X	X
Category	**Internal reference**	**Antivirus detection**	**Payload delivery**	**Artifacts dropped**	**Payload installation**	**Persistence mechanism**	**Network activity**	**Payload type**	**Attribution**	**External analysis**	**Other**

Categories

Category	Description
Internal reference	Reference used by the publishing party (e.g. ticket number)
Antivirus detection	List of anti-virus vendors detecting the malware or information on detection performance (e.g. 13/43 or 67%). Attachment with list of detection or link to VirusTotal could be placed here as well.
Payload delivery	Information about the way the malware payload is initially delivered, for example information about the email or web-site, vulnerability used, originating IP etc. Malware sample itself should be attached here.

Artifacts dropped	Any artifact (files, registry keys etc.) dropped by the malware or other modifications to the system
Payload installation	Location where the payload was placed in the system and the way it was installed. For example, a filename|md5 type attribute can be added here like this: c:\windows\system32\malicious.exe|41d8cd98f00b204e9800998ecf8427e.
Persistence mechanism	Mechanisms used by the malware to start at boot. This could be a registry key, legitimate driver modification, LNK file in startup
Network activity	Information about network traffic generated by the malware
Payload type	Information about the final payload(s). Can contain a function of the payload, e.g. keylogger, RAT, or a name if identified, such as Poison Ivy.
Attribution	Identification of the group, organisation, or coountry behind the attack
External analysis	Any other result from additional analysis of the malware like tools output Examples: pdf-parser output, automated sandbox analysis, reverse engineering report.
Other	Attributes that are not part of any other category

Types

Type	Description
md5	You are encouraged to use filename|md5 instead. A checksum in md5 format, only use this if you don't know the correct filename
sha1	You are encouraged to use filename|sha1 instead. A checksum in sha1 format, only use this if you don't know the correct filename
filename	Filename
filename|md5	A filename and an md5 hash separated by a | (no spaces)
filename|sha1	A filename and an sha1 hash separated by a | (no spaces)
ip-src	A source IP address of the attacker
ip-dst	A destination IP address of the attacker or C&C server. Also set the IDS flag on when this IP is hardcoded in malware
hostname	A full host/dnsname of an attacker. Also set the IDS flag on when this hostname is hardcoded in malware
domain	A domain name used in the malware. Use this instead of hostname when the upper domain is important or can be used to create links between events.
email-src	The email address (or domainname) used to send the malware.
email-dst	A recipient email address that is not related to your constituency.
email-subject	The subject of the email
email-attachment	File name of the email attachment.
url	url
user-agent	The user-agent used by the malware in the HTTP request.
regkey	Registry key or value
regkey|value	Registry value + data separated by |
AS	Autonomous system
snort	An IDS rule in Snort rule-format. This rule will be automatically rewritten in the NIDS exports.
pattern-in-file	Pattern in file that identifies the malware
pattern-in-traffic	Pattern in network traffic that identifies the malware

pattern-in-memory	Pattern in memory dump that identifies the malware
yara	Yara signature
vulnerability	A reference to the vulnerability used in the exploit
attachment	Please upload files using the ***Upload Attachment*** button.
malware-sample	Please upload files using the ***Upload Attachment*** button.
link	Link to an external information
comment	Comment or description in a human language. This will not be correlated with other attributes (NOT IMPLEMENTED YET)
text	Name, ID or a reference
other	Other attribute

4. Export and Import

The platform has full support for automated data export and import.

IDS and script export

First of all you can export data in formats that are suitable for NIDS or scripts (text, xml,...).
All details about this export can be found on the **Export** page.

REST API

The platform is also **RESTfull**, so this means you can use structured format (XML) to access Events data.

Requests

Use any HTTP compliant library to perform requests. However to make clear you are doing a REST request you need to either specify the `Accept` type to `application/xml`, or append `.xml` to the url.

The following table shows the relation of the request type and the resulting action:

HTTP format	URL	Controller action invoked
GET	/events	EventsController::index() (1)
GET	/events/123	EventsController::view(123) (2)
POST	/events	EventsController::add()
PUT	/events/123	EventsController::edit(123)
DELETE	/events/123	EventsController::delete(123)
POST	/events/123	EventsController::edit(123)

(1) Warning, there's a limit on the number of results when you call `index`.
(2) Attachments are included using base64 encoding below the `data` tag.

Authentication

REST being stateless you need to authenticate your request by using your **authkey/apikey**. Simply set the `Authorization` HTTP header.

Example - Get single Event

In this example we fetch the details of a single Event (and thus also his Attributes).
The request should be:

```
GET http://localhost:8888/events/123
```

And with the HTTP Headers:

```
Accept: application/xml
Authorization: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

The response you're going to get is the following data:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
        <Event>
                <id>123</id>
                <date>2012-04-06</date>
                <risk>Undefined</risk>
```

```
                <info>TEST</info>
                <published>0</published>
                <uuid>4f7eff11-4e98-47b7-ae96-6a7fff32448e</uuid>
                <private>0</private>
                <Attribute>
                        <id>9577</id>
                        <event_id>123</event_id>
                        <category>Artifacts dropped</category>
                        <type>other</type>
                        <value>test other</value>
                        <to_ids>1</to_ids>
                        <uuid>4f7fe870-e5a4-4b9e-a89c-a45bff32448e</uuid>
                        <revision>1</revision>
                        <private>0</private>
                </Attribute>
                <Attribute>
                        <id>9576</id>
                        <event_id>123</event_id>
                        <category>Payload delivery</category>
                        <type>filename</type>
                        <value>test attribute</value>
                        <to_ids>1</to_ids>
                        <uuid>4f7fe85b-0f78-4e40-91f3-a45aff32448e</uuid>
                        <revision>1</revision>
                        <private>0</private>
                </Attribute>
        </Event>
</response>
```

Example - Add new Event

In this example we want to add a single Event.
The request should be:

```
POST http://localhost:8888/events
Accept: application/xml
Authorization: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

And the request body:

```
<Event>
        <date>2012-05-06</date>
        <risk>Undefined</risk>
        <info>TEST REST</info>
        <published>0</published>
        <private>0</private>
        <attribute/>
</Event>
```