Documentacion de MISP

Para realizar la instalacion de misp tendremos que bajarnos un script con la instalacion de misp en un servidor y meterlo en la carpeta tmp

```
wget -O /tmp/INSTALL.sh https://raw.githubusercontent.com/MISP/MISP/2.4/INSTALL/INSTALL.sh
```

Una vez lo tengamos descargado, ejecutaremos el script

```
bash /tmp/INSTALL.sh
```

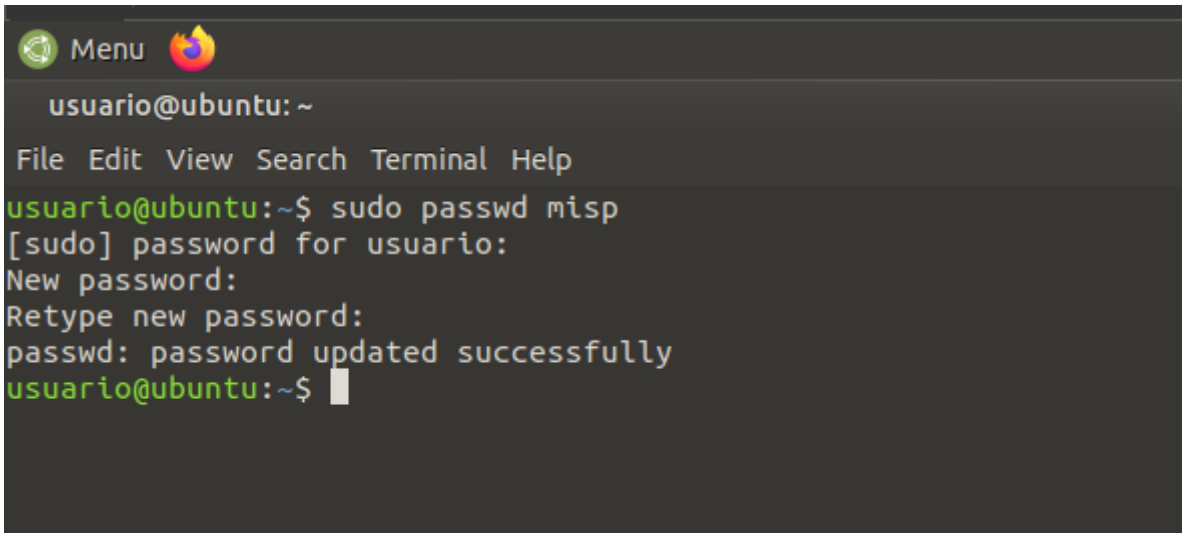
Nada mas ejecutarlo nos pedira, que modo de instalación de misp queremos realizar

Con la opcion -A nos realizara una instalacion completa y desatendida

```
bash /tmp/INSTALL.sh -A
```

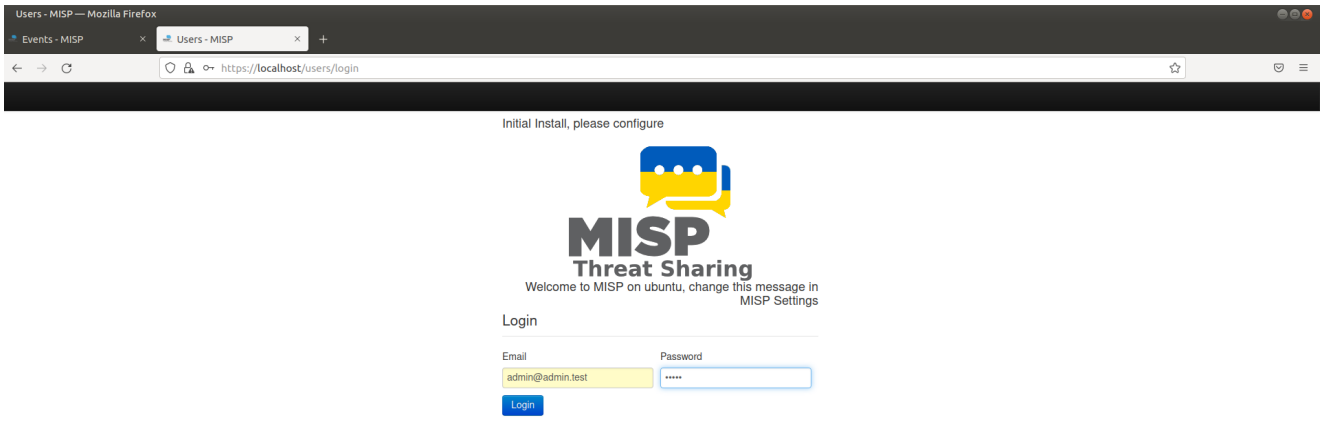
Tardara un tiempo en instalar pero una vez instalado nos saldra un resumen de todas las contraseñas generadas de los distintos servicios o cuentas.

tambien tener en cuenta que se nos generará un usuario de sistema con uid(misp) el cual su contraseña es bastante larga y seria recomendable cambiarla desde dicha cuenta o con sudo passwd misp



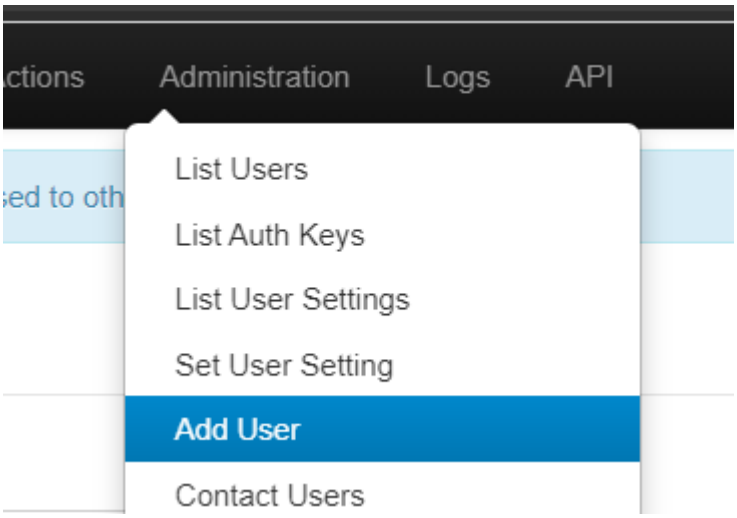
Para acceder a MISP una vez instalado, con ip/localhost en un navegador

--- Credenciales: ---
user: admin@admin.test
passowrd: admin

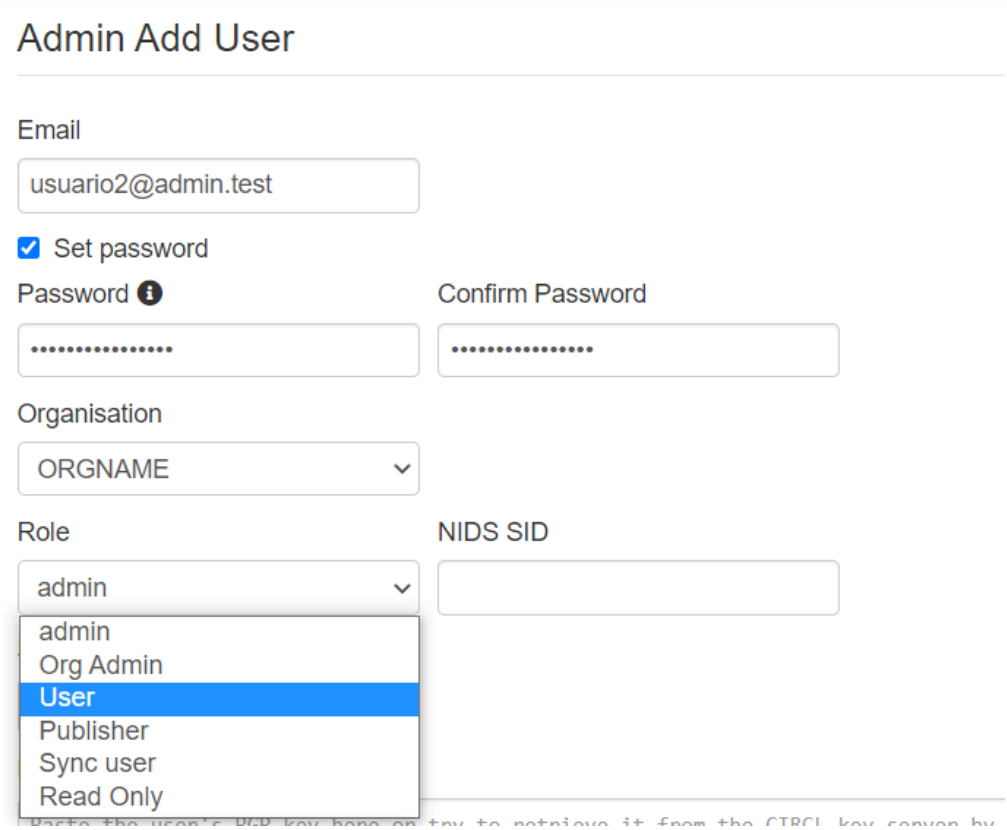


Creacion de usuarios en MISP

Para crear un usuario en MISP sería Administration⇒ Add User

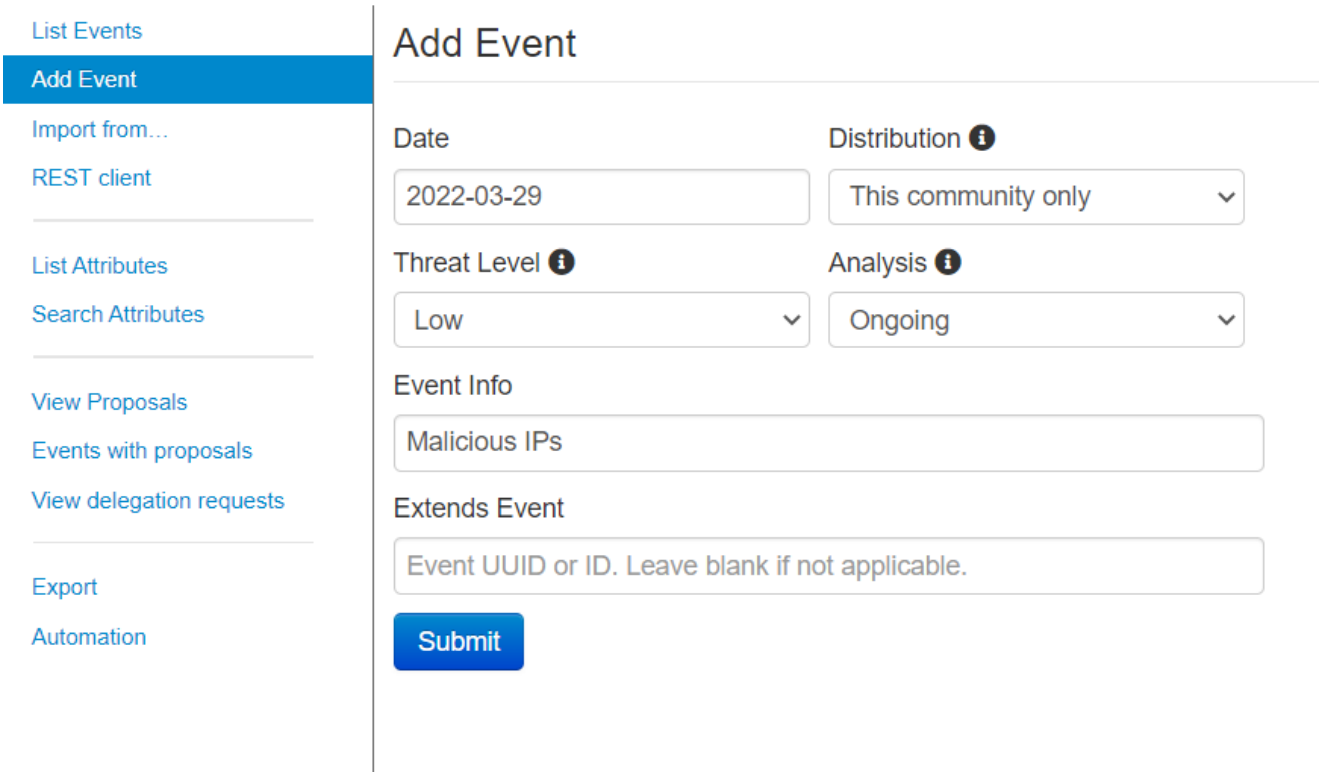


Donde podremos asignar a que tipo de organizacion, email, contraseña y rol va a tener dentro de MISP

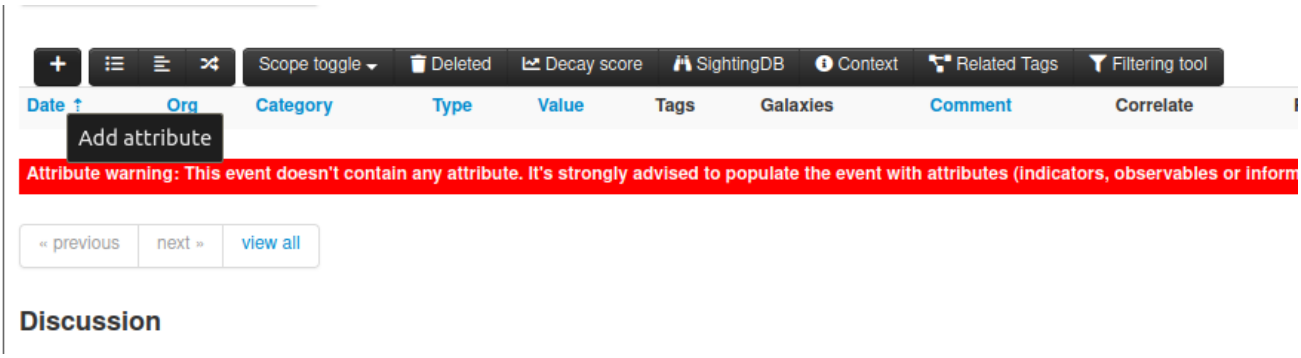


Para añadir un evento MISP

Tendremos que ir a home ⇒ add event



Dentro de añadir un evento, especificando la **distribucion del evento, el nivel de amenaza, y el análisis de la amenaza**

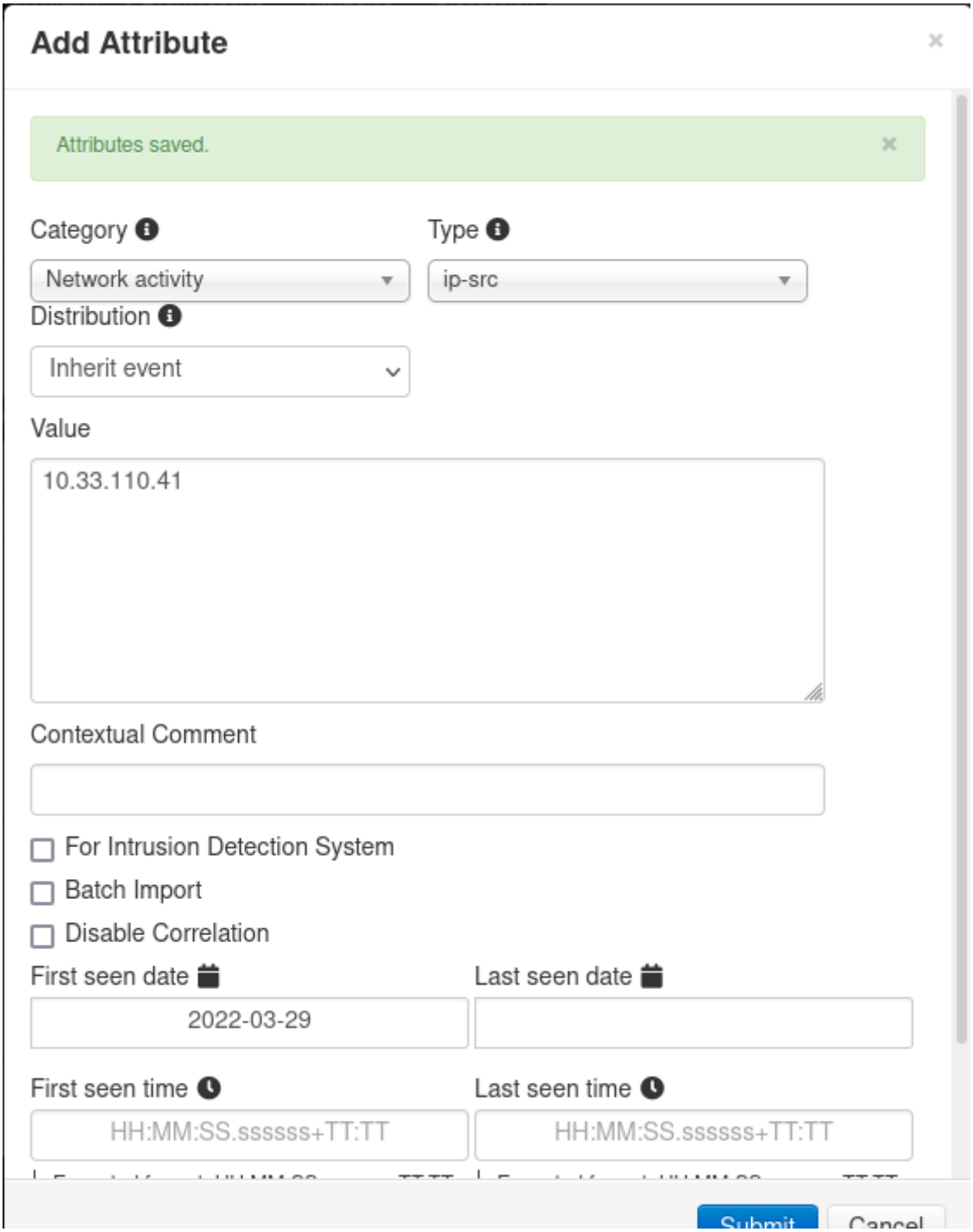


Para añadir informacion de la amenaza, daremos a "Add attribute"

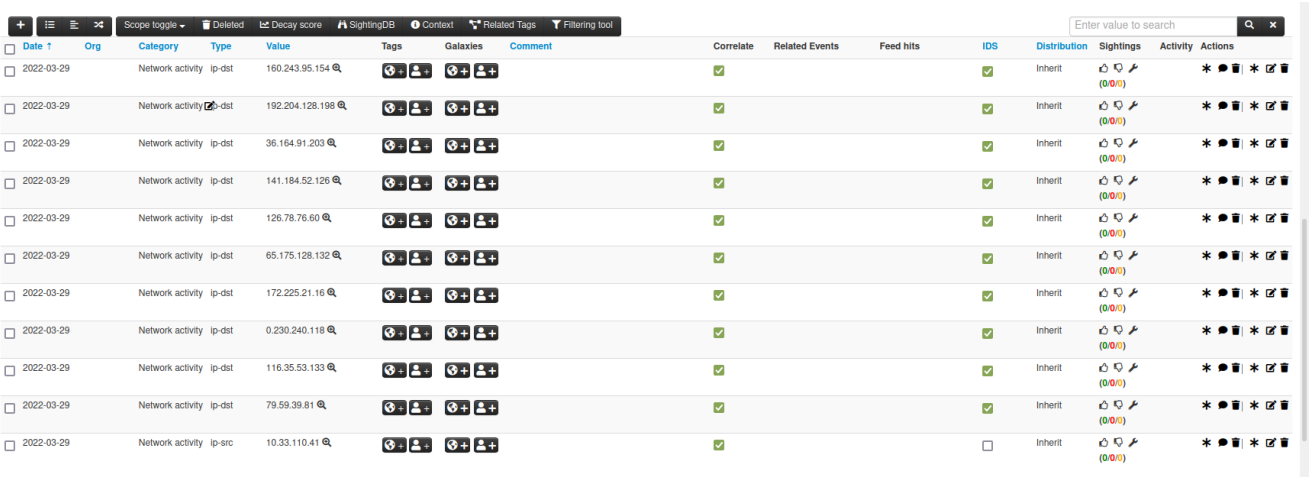
Dentro de El Atributo, añadiremos la categoría de la informacion y el tipo, (El nuestro al ser IPs maliciosas)

El tipo de categoria seía **network activity**
y el Tipo, **ip-src**

Tambien podemos añadir, en que fecha se han divisado dicha amenaza



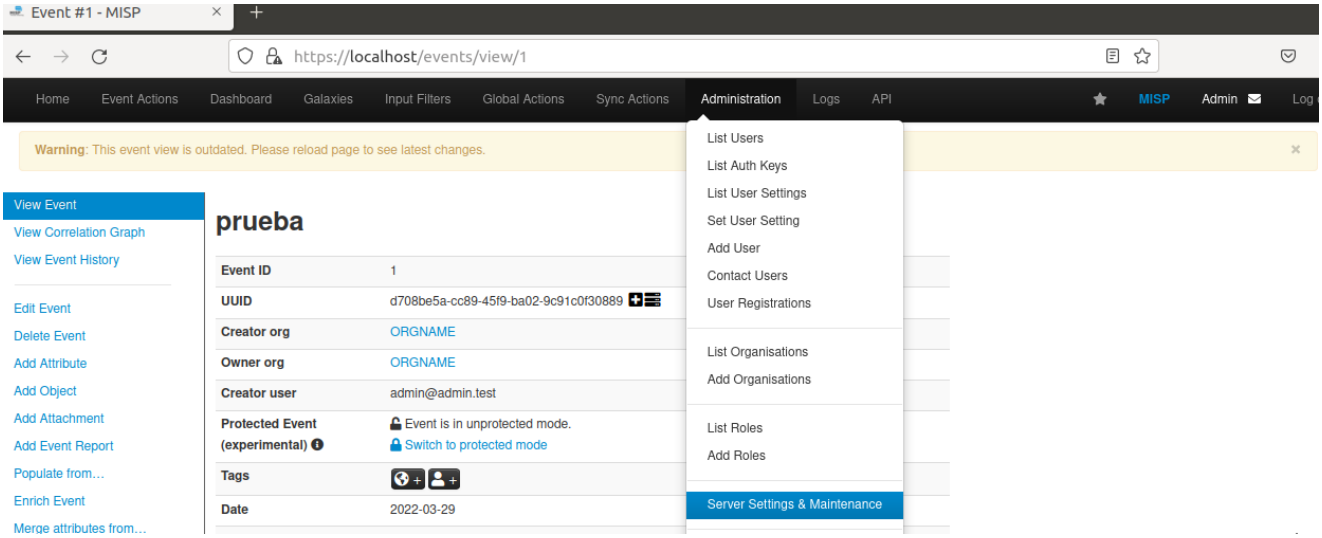
Y como vemos se añadirá.



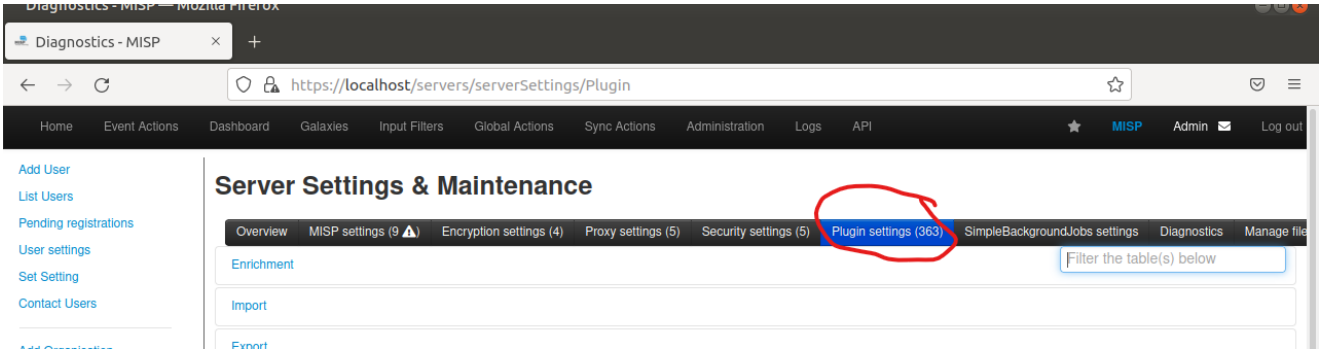
instalacion de pluggins

es interesante recalcar que podemos meter plugins a la plataforma lo cual nos permite hacer desde cosas especificas como relacionar eventos o sus atributos a otros como poder hacer que los eventos se retro-alimenten añadiendo cuando sea necesario nuevos atributos("amenazas"), y se conseguiria de la siguiente manera

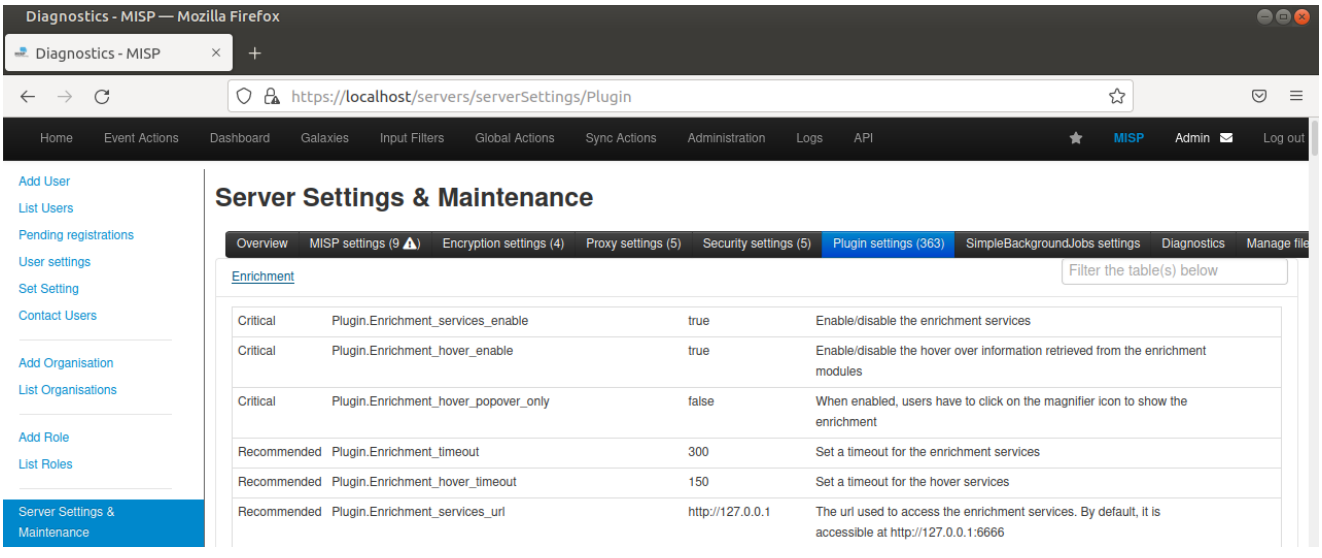
primero nos dirigimos a ajustes de servidor y mantenimiento



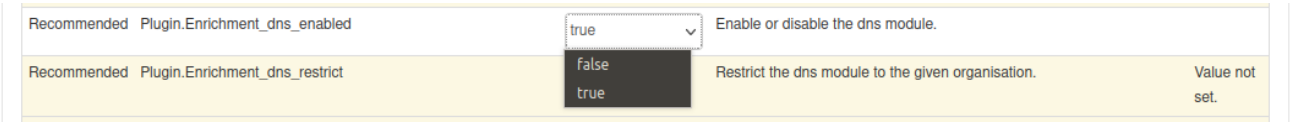
luego le damos a ajustes de pluggins

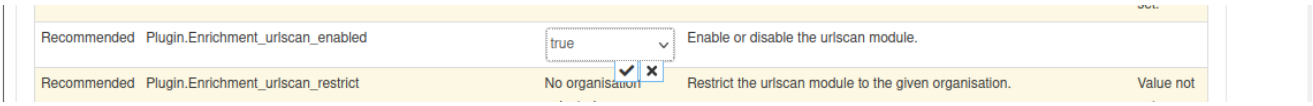


despues clicamos en la primera opcion (enrichment) y nos saldra una lista de las cuales podemos activar las que necesitemos y no sean de pago



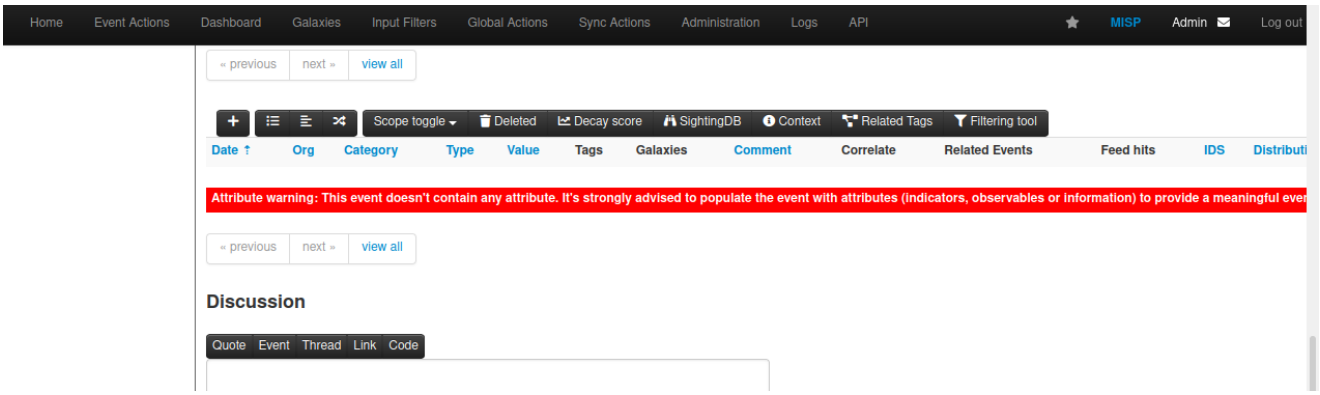
en este caso se activaran 2 pluggins los cuales seran:



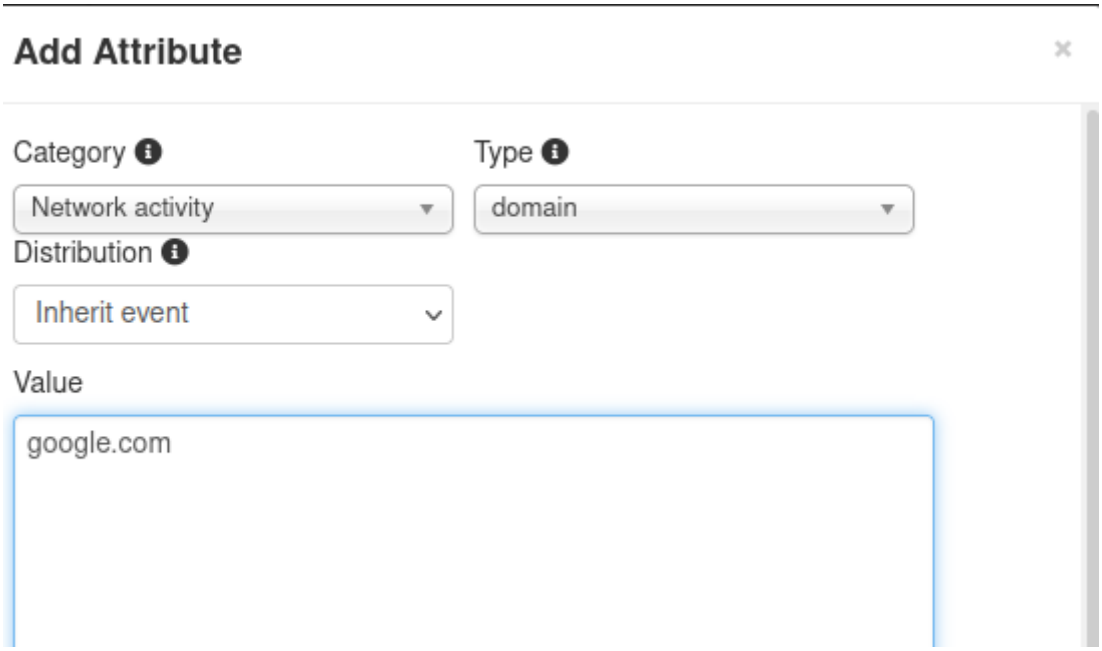


estos plugins nos permitiran por ejemplo en el caso de un evento, añadir un nombre de dominio como atributo y que su ip se añada automaticamente o en el caso del segundo pluggin que cuando lo utilicemos nos avisaria de todas las url relacionadas con dicho atributo. asi en seleccion del usuario se podria abarcar un amplio abanico de posiblidades.

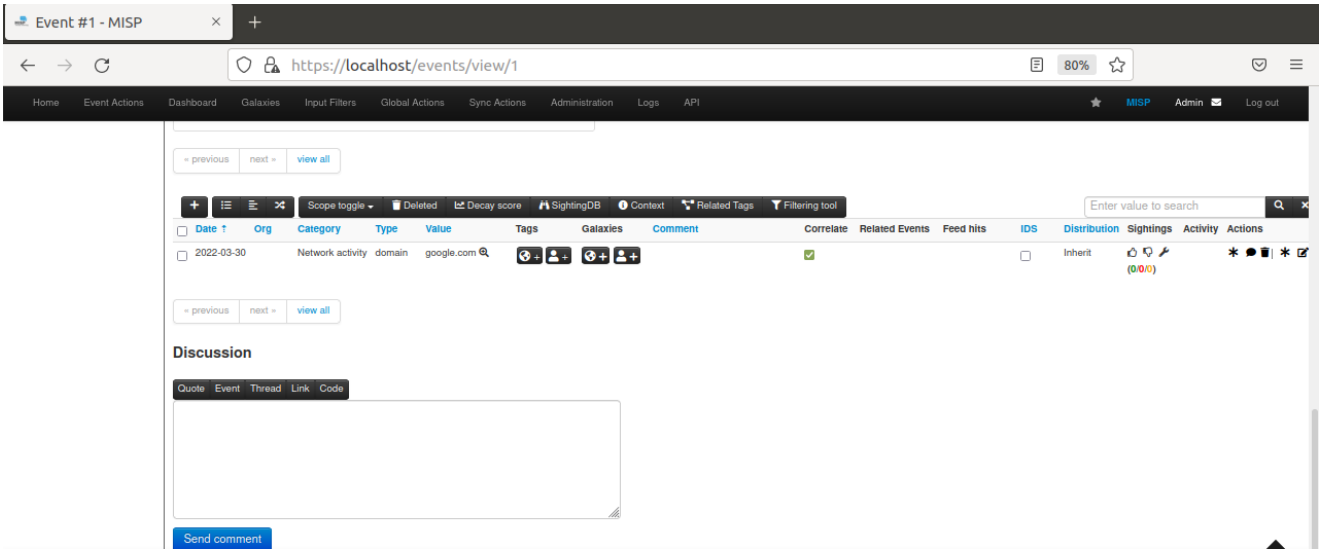
un ejemplo de esto seria:



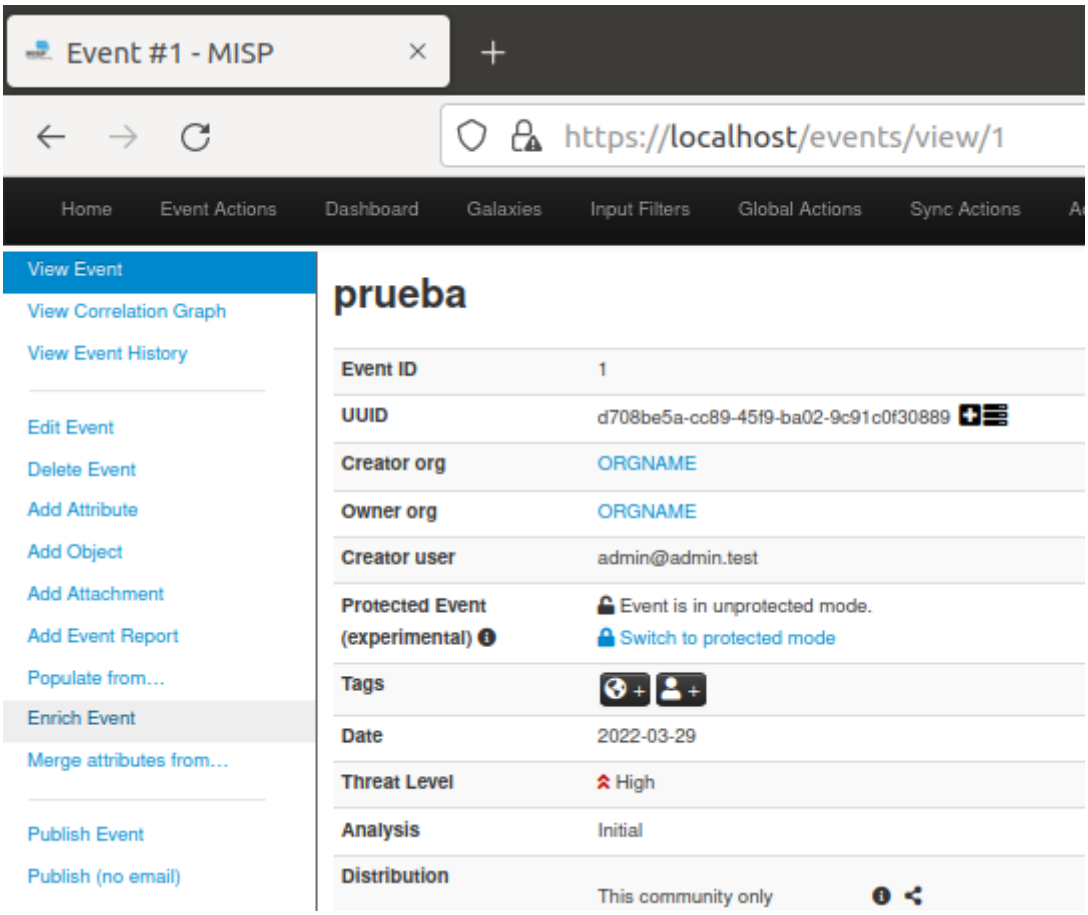
añadiría ahora el dns de google.com



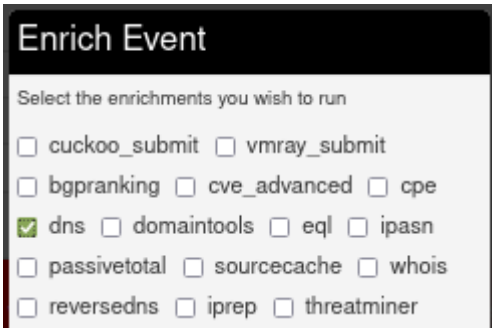
una vez añadido se mostraria solo en dns



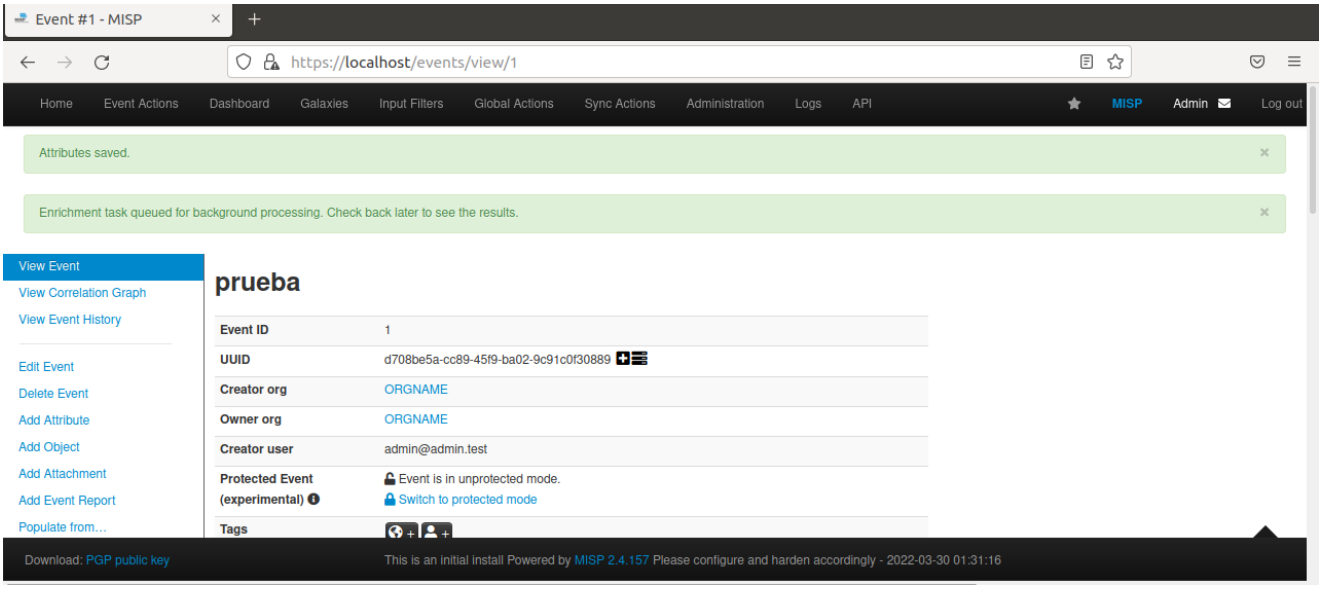
iriamos a la parte de arriba del evento y clicariamos a la izquierda en enrich event

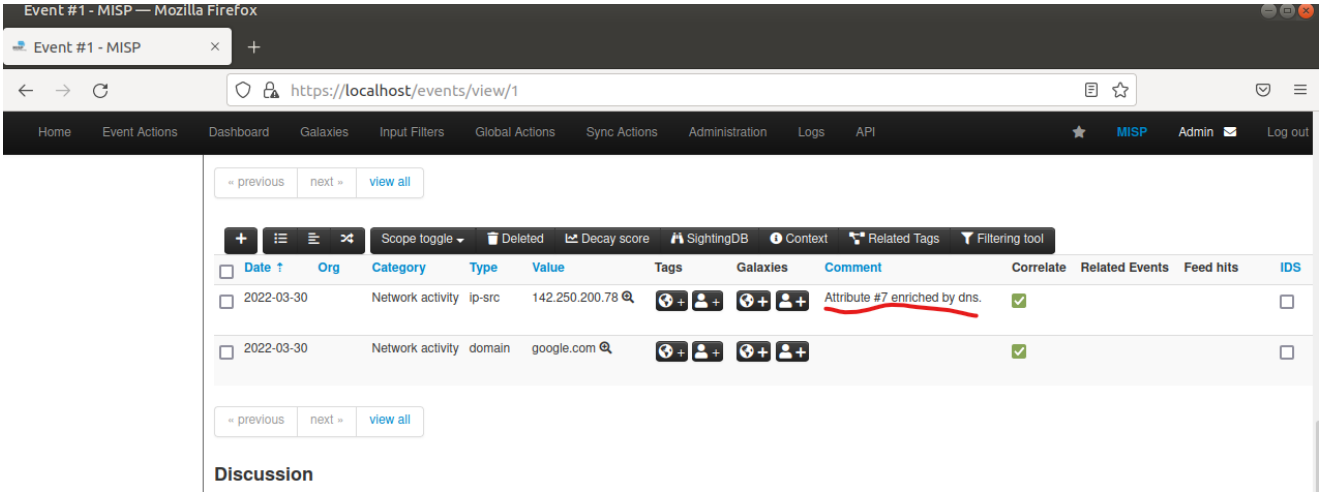


entonces saltaria una ventana la cual nos muestra todos los pluggins instalados en este caso hay mas de los necesarios pero solo daremos clic a dns:

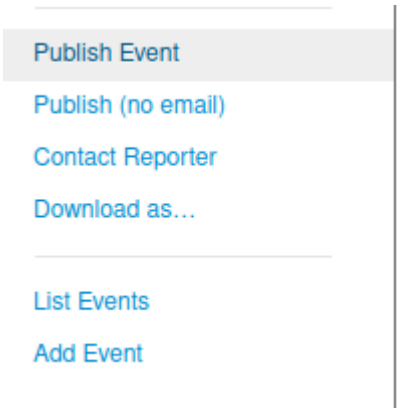


despues de darle y aceptar refrescamos la pagina y se puede apreciar la aparicion de otro atributo el cual es una ip y pertenece al dominio agregado anteriormente





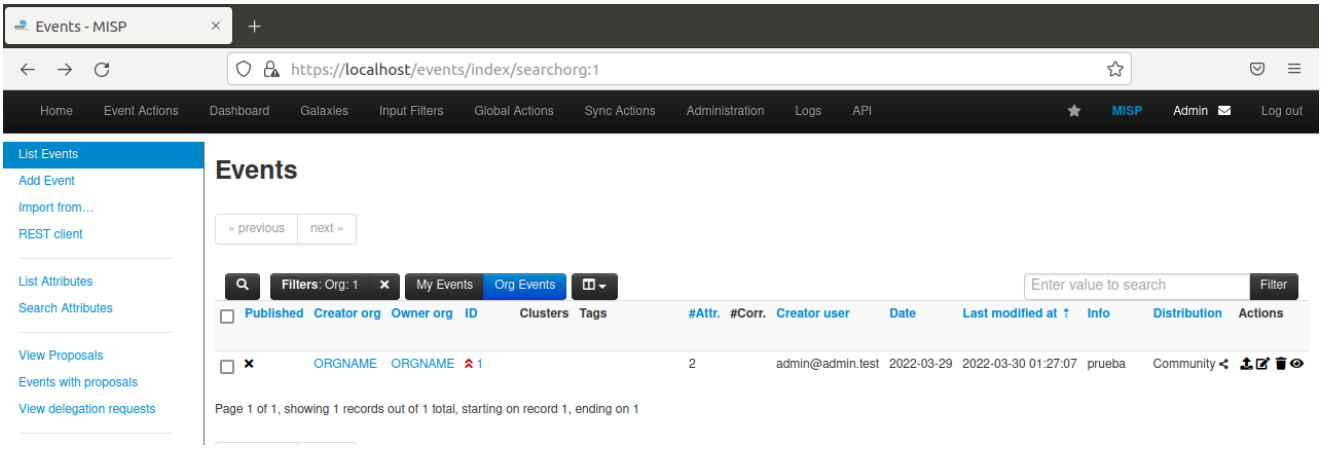
Una vez tengamos un lista de ips podremos guardar el evento y publicarlo



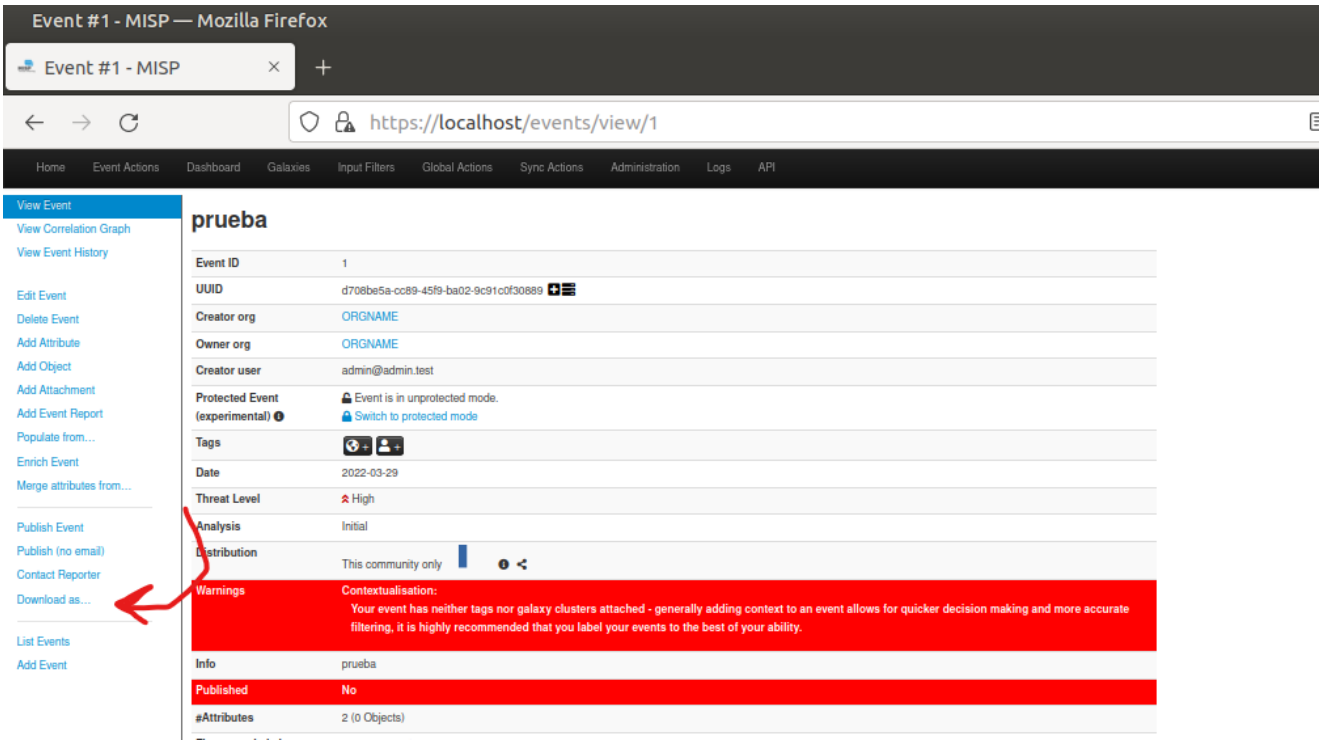
Y si nos vamos a la home, tendremos nuestro evento publicado

Exportar / Importar eventos

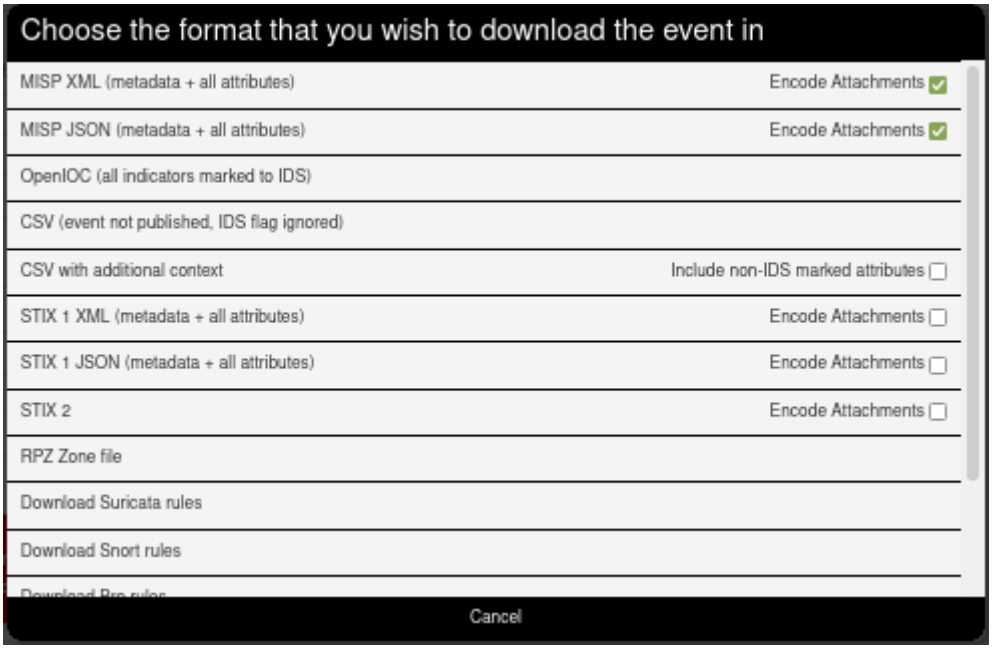
para exportar eventos lo recomendble es descargar el evento que queremos exportar en este caso uno de prueba que contiene 4 atributos simples



al entrar al evento podemos apreciar un menu a la izquierda el cual tiene la opcion de descargar el evento ("download as....")

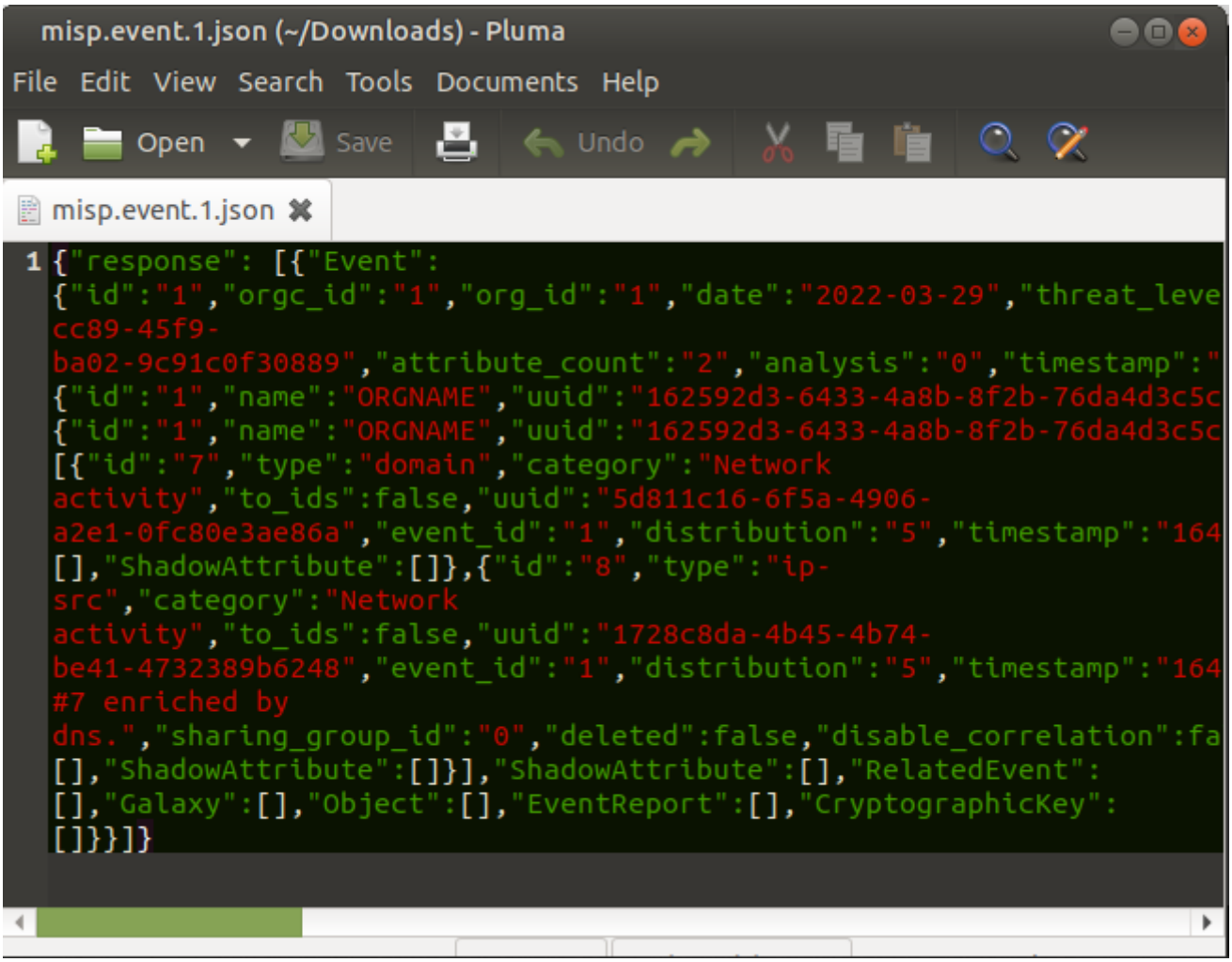


si clicamos en el el nos mostrara varias opciones



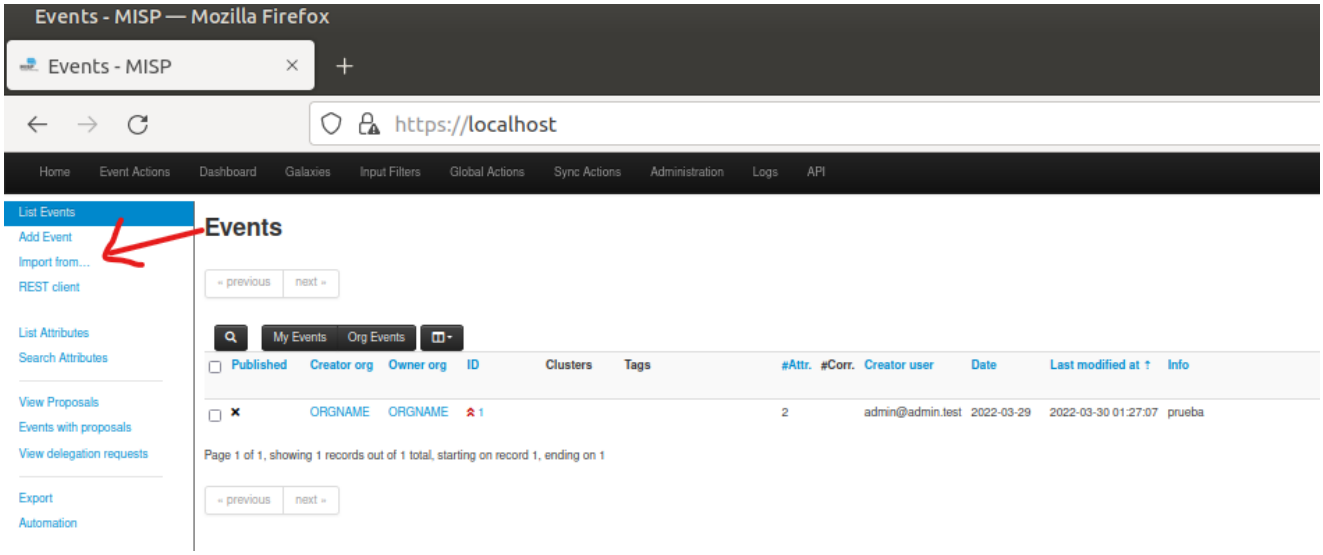
la que nos interesaria para poder pasar eventos desde un modo practico seria en formato JSON. Aunque tambien se puede usar otros formatos como csv que se podrian implementar con otros servicios o plataformas como ELK que veremos mas adelante.....

una vez seleccionada la opcion se descargaria con el formato seleccionado por ejemplo en el caso de JSON:

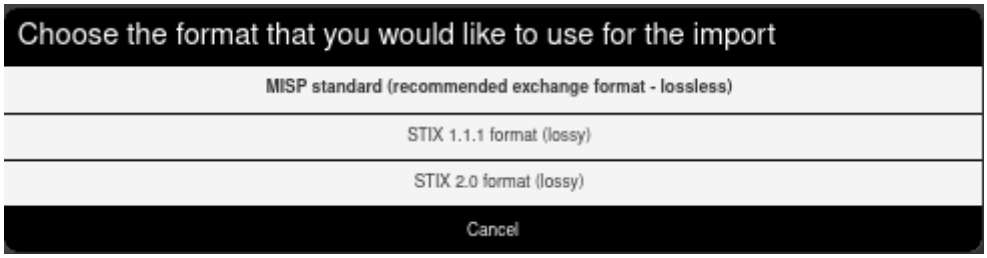


para importar eventos tenemos que tener en cuenta una cosa y es que los archivos/eventos a importar tienen que tener un formato xml o json. y se realizaria de la siguiente manera:

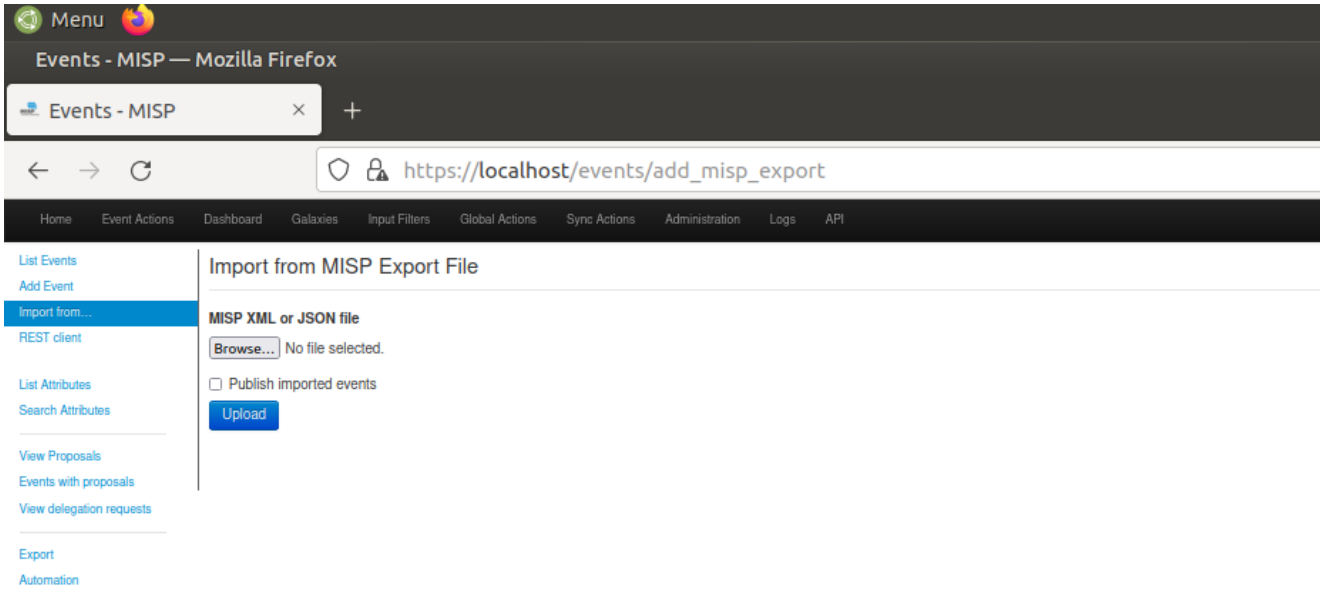
desde el home de misp veremos la opcion de import from("importar desde")



cuando le demos nos pedira un formato

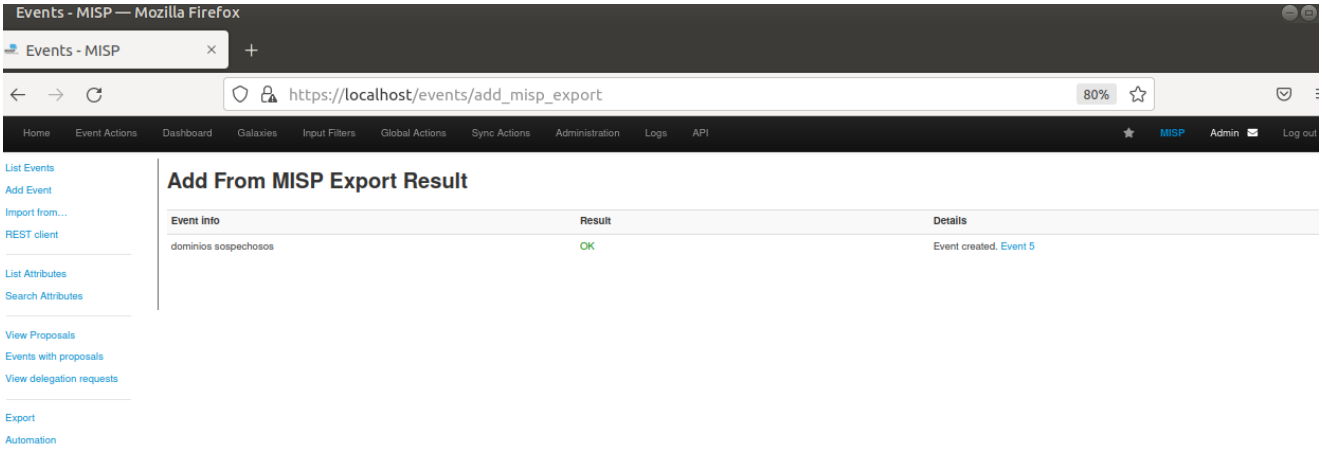


en este caso vamos a probar a seleccionar el formato MISP que se basa en xml/json

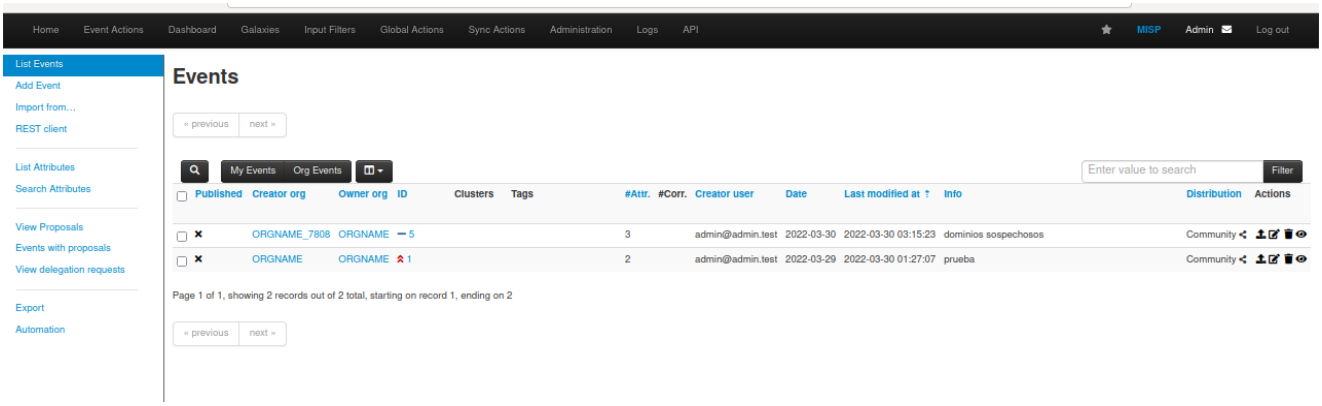


se puede ver que tambien se puede publicar a la hora de importar

una vez le demos a upload nos mostrara los eventos importados



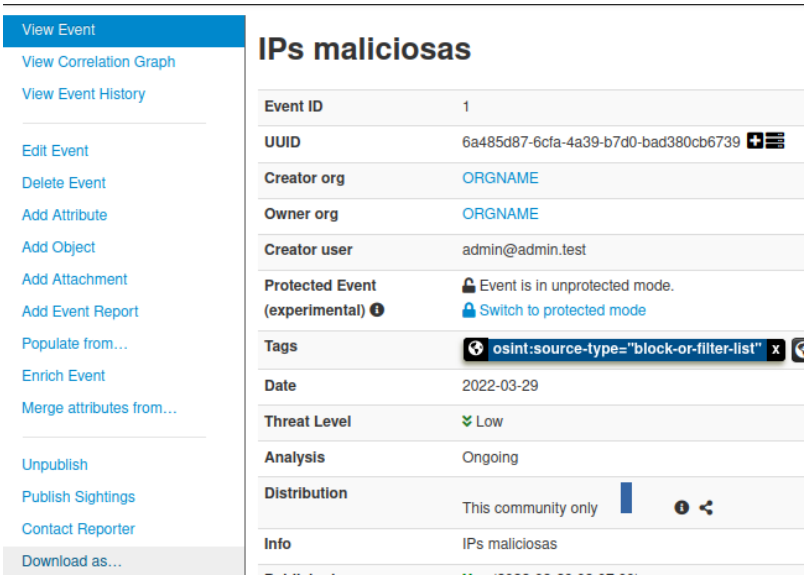
luego al dar a home veremos que se han creado correctamente



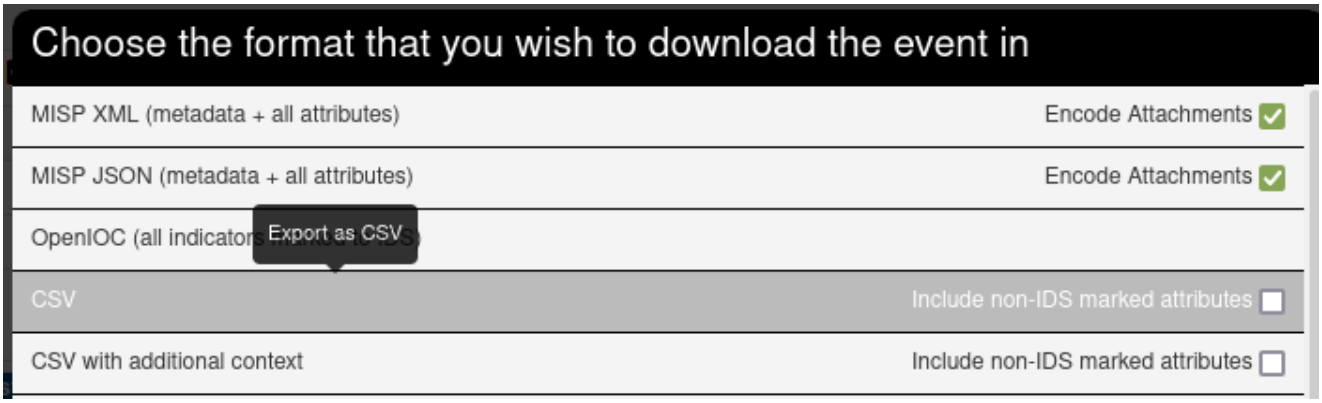
Implementar MISP a ELK

Para poder implementar MISP a ELK tendremos que descargarnos un evento en forma de CSV y aplicarlo con logstash

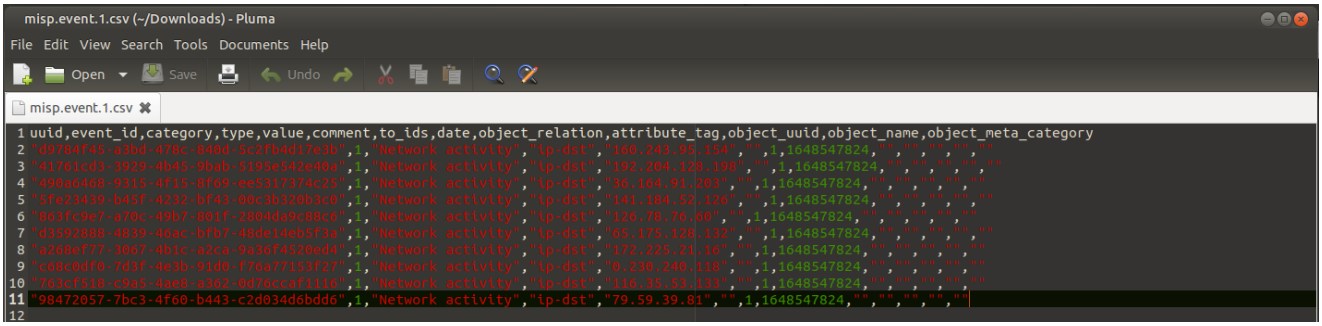
Para ello, **dentro de un evento** tendremos que irnos a la parte de "download as"



Eligiremos el csv



Como vemos se nos descarga en formato de CSV



Para implementarlo a ELK tendremos que tener una index mapeado creado

```
POST /ipsmaliciosas/_doc
{
        "mappings": { "properties": {
        "valor": { "type": "long" },
        "valor": { "type": "text" },
        "valor": { "type": "long" },
        "valor": { "type": "long" },
        "valor": { "type": "long" },
        "valor": { "type": "long" },
        "valor": { "type": "long" },
        "valor": { "type": "text" }
                }
        }
}
```

Y crearnos un pipeline para meter los datos de CSV

```
input {
        file {
        path => "ruta al fichero de csv"
        start_position => beginning
        sincedb_path => "/dev/null"
        mode => read
        exit_after_read => true
        }
}
        filter {
        csv {
        columns => ["columnas creadas en la platilla del index"]
        }
        mutate {
        remove_field => ["columnas a eliminar"]
        }
}
        output {
        stdout { }
        elasticsearch {
        index => "nombre del index creado(ipsmaliciosas en este caso)"
        }
}
```

Y relizar un logstash

```
sudo /usr/share/logstash/bin/logstash -f '/home/usuario/Desktop/pipeline.conf'
```

Posibles usos de misp

visto todo lo anterior misp podria tener diferentes implementaciones gracias a su versatilidad y su adaptacion al uso que le queremos dar. Existen usos como la implementacion de eventos a un firewall en el cual se irian añadiendo bloqueos a ips no deseadas de forma automatica. por ejemplo, tambien existe la posiblidad de un dns automatizado con la implementacion de un evento autosuficiente o se podria usar como una herramienta de aprendizaje donde los usuarios o equipo de seguridad se informen de nuevas amenazas, debilidades y fortalezas.