

恶意代码扫描平台用户手册

一、系统概述

(一) 平台简介

本平台是基于Vue+MySQL的恶意代码在线扫描平台，我们在后端集成了Yara引擎与支持Sigma规则的Zircolite引擎，您可以自行选择基于恶意软件的静态扫描或基于Sysmon捕获日志的动态扫描。此外，您可以上传自定义的Yara规则与Sigma规则，通过选择某些特定规则，来进行有针对性的扫描工作。

(二) 主要特性

1. 用户在部署之后可以自行选择在网页进行测试或在本地进行测试，具体测试方法详见[详细功能指南](#)部分
2. 本平台可以支持规则单独上传，例如.yar .yara .yml；也可支持压缩包形式的上传，例如.zip形式上传。
3. 本平台可以支持恶意样本的批量处理，您可以通过上传压缩包来一次性扫描多个恶意程序。
4. 本平台可以提供简易的扫描报告，如果您选择前端扫描测试，可以直观看到匹配的规则结果；如果您选择后端扫描测试，我们可以在您指定的位置输出简易的扫描报告（我们使用的json文件形式，您可能需要安装相应的插件或程序，例如python以便能够看到详细的结果输出）
5. 本平台在后端已经完成了规则的预编译，可以一定程度上缓解您上传规则数量较多的问题，但请不要上传过量的规则，以防止出现卡顿的现象。

(三) 系统支持

本网站仅能够支持Windows系统，针对MacOS系统与Linux系统，我们并没有进行配置。

我们推荐使用Windows10及以后的操作系统，且已经完成了MySQL的安装与配置，建议您使用Edge浏览器进行访问。

二、快速入门

(一) 本地配置

在本地部署之后，您可以通过下面的命令实现环境的搭建：

```
1 cd vuln_backend
2 python -m src.run
3 cd ..
4 cd dist
5 python -m http.server 8081
```

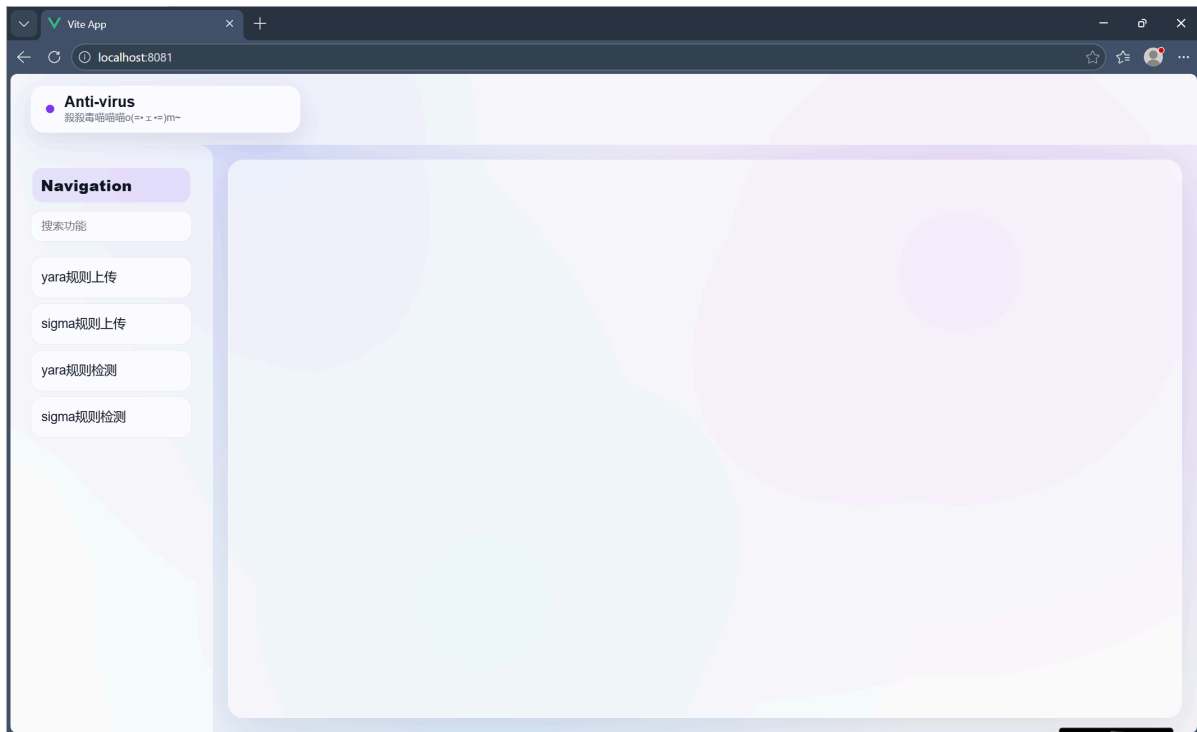
之后您就可以通过访问下面这个URL访问到我们的网页了：

```
1 http://127.0.0.1:8081/index.html
```

如果您不想使用这个上面的IP+端口+制定文件的形势，您可以使用下面这个方法也可以快速访问到指定的页面：

1 | localhost:8081

如果您能够看到下面这个页面：



即可证明您已经完成了我们环境的配置了，接下来您就可以进行规则上传、文件扫描、规则匹配、检测报告的获取了。

（二）功能测试

在完成部署之后，您可以简单上传某个规则以及某个恶意样本进行功能性测试，我们将给您提供示例：

样本上传

您可以在Yara规则上传或Sigma规则上传页面点击**选择文件**，然后就可以打开本地文件，选择您想要上传的测试文件，如果上传之后出现下面这个结果：

YARA 规则导入

支持上传 .yar / .yara / .zip (zip 内只允许包含 yar/yara 文件)。

manual-upload

选择文件

选择文件 已选择: Lab03_01.yar (303 B)

已选择: Lab03_01.yar (303 B)

清空

上传成功 (单文件)。

后端响应

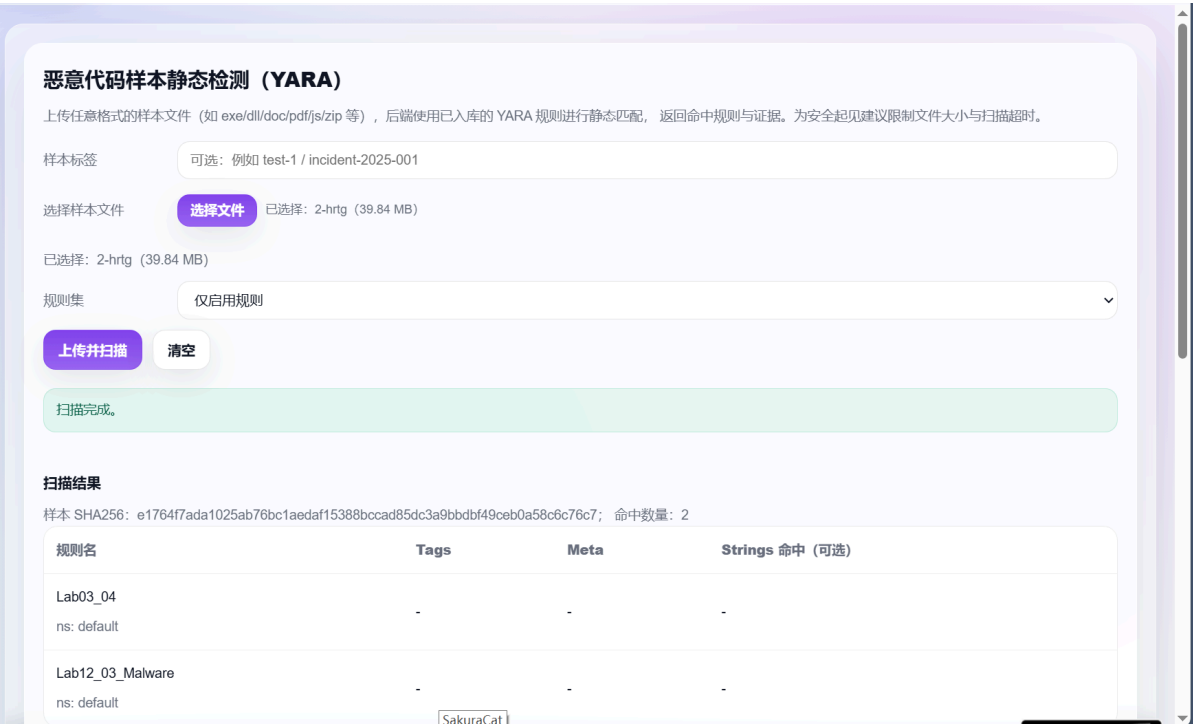
```
{
  "created_at": "2025-12-26T16:44:20",
  "file_sha256": "86f1e0eec27b20629a66a58e49d42e052c8c4db1621d06ff6f38667204612ffc6",
  "filename": "Lab03_01.yar",
  "kind": "single",
  "ok": true,
  "rule_names": [],
  "skipped_count": 1,
  "source_name": "manual-upload",
  "stored_count": 0
}
```

说明您已经成功上传了指定文件了，您可以在数据库表中找到对应的规则，如下所示：

[illegible]

恶意样本测试

您可以选择**Yara规则检测模块**，点击选择文件，然后选择一个测试的恶意样本，上传后点击测试，如果看到下面的结果，说明您的环境已经配置完毕了，基础的测试功能也已经实现了。



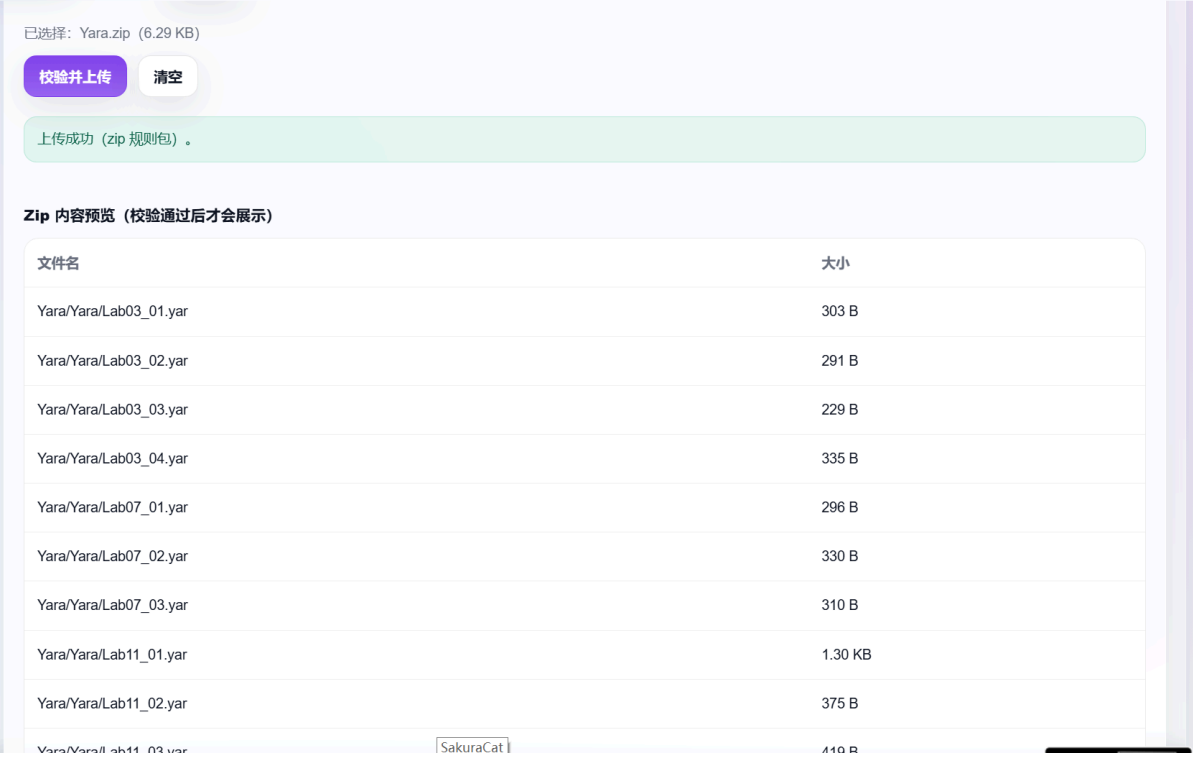
三、详细功能指南

如果您已经完成了上面所说的环境配置与基本功能测试，接下来我们就可以进行详细功能的展示了。

(一) 批量规则上传与检测

批量上传

与普通规则上传一致，您只需要将您需要上传的规则变为压缩包，就能完成一次性的成批上传了。需要注意的是，处于安全考虑，我们不允许其他样式后缀名规则的上传，因此在您的压缩包中，如果存在我们指定的格式（.yml .yara .yar）之外的后缀名，会产生错误提示。为了能够正常使用，我们建议您在批量上传时检查一下文件的后缀名：



批量检测

与普通上传一致，您只需要将您需要上传的恶意程序变为**压缩包**，就能完成一次性的成批上传检测了。

恶意代码样本静态检测 (YARA)

上传任意格式的样本文件（如 exe/dll/doc/pdf/js/zip 等），后端使用已入库的 YARA 规则进行静态匹配，返回命中规则与证据。为安全起见建议限制文件大小与扫描超时。

样本标签

可选：例如 test-1 / incident-2025-001

选择样本文件

选择文件

已选择：Chapter_3L.zip (37.58 KB)

已选择：Chapter_3L.zip (37.58 KB)

规则集

仅启用规则

上传并扫描

清空

扫描完成。

扫描结果

样本 SHA256：ef609561158dda293de8b6375f6cba18a5e52e929f3b636f1dbf79fe0c61aa6a；命中数量：1

规则名	Tags	Meta	Strings 命中 (可选)
Lab03_03	-	-	-
ns: default			

后端原始响应

(二) 后端扫描并输出日志

如果您不满足于仅能分辨出样本是否存在恶意行为，还想要得到相应的扫描日志，我们推荐您绕过前端直接使用后端进行测试，在这里，您不仅可以扫描单个文件，还可以扫描整个文件夹。您可以使用下面的指令来进行操作：

```
1 python yara_folder_scan_client.py "您的待扫描程序目录" --api "http://127.0.0.1:3000/scansamplewithYara" --rule-set enabled
```

这样您就会在您的扫描目录下面找到后缀为**您的待扫描程序目录_testres**的文件夹，这里就是对应的扫描结果，您可以查看相应的扫描报告。我们已经提前进行过后端的扫描测试了，下面是一个扫描的结果，可供您进行参考：

```
1 {
2   "ok": true,
3   "code": null,
4   "message": null,
5   "http_status": 200,
6   "sample_filename": "5-BossDaMajor",
7   "sample_sha256_from_server":
8     "730a41a7656f606a22e9f0d68782612d6e00ab8cfe1260160b9e0b00bc2e442a",
9   "rule_set": "enabled",
10  "hit_rule_count": 1,
11  "hit_rule_names": [
12    "Lab03_04"
13  ],
14  "matches": [
15    {
16      "meta": {},
17      "namespace": "default",
```

```

17     "rule": "Lab03_04",
18     "strings": [],
19     "tags": []
20   }
21 ],
22 "engine_stderr_tail": "",
23 "engine_stdout_tail": "",
24 "rel_path": "5-BossDaMajor",
25 "abs_path": "C:\\Users\\Malware_test_yzx\\Desktop\\恶意代码教学样本-在沙盒中修
改文件名\\恶意代码教学样本-在沙盒中修改文件名\\5-BossDaMajor",
26 "size": 2014208,
27 "local_sha256":
"730a41a7656f606a22e9f0d68782612d6e00ab8cfe1260160b9e0b00bc2e442a",
28 "timestamp": "2025-12-22 08:25:58"
29 }

```

这里是一个文件的扫描结果，您可以找到您需要的内容，包括**匹配到的规则**，**文件目录**、**文件大小**、**扫描时间**等信息。

```

1  {
2    "root_dir": "C:\\Users\\Malware_test_yzx\\Desktop\\恶意代码教学样本-在沙盒中修
改文件名\\恶意代码教学样本-在沙盒中修改文件名",
3    "api": "http://127.0.0.1:3000/scansamplewithYara",
4    "rule_set": "enabled",
5    "started_at": "2025-12-22 08:25:56",
6    "total_files_seen": 4,
7    "total_scanned": 4,
8    "total_ok": 4,
9    "total_hit_files": 3,
10   "total_skipped_too_large": 0,
11   "total_errors": 0,
12   "hit_rule_counter": {
13     "Lab03_04": 2,
14     "Lab12_03_Malware": 2
15   },
16   "finished_at": "2025-12-22 08:25:58"
17 }

```

如果您扫描了一整个文件夹，那么还可以找到一个文件夹的扫描结果。这里我们会向您展示**这个文件夹所在的目录**，**文件夹中文件的数量**，**扫描到的恶意文件**，**匹配到的对应规则**，以及**扫描完成的时间**。在这里，您可以**计算扫描准确率**、**漏报率**等信息。

(三) 扫描测试

四、支持文件格式

- 对于**Yara规则**和**Sigma规则**的上传，我们支持使用**.yara .yar .yml**的未压缩形式，以及**.zip**的压缩包形式，您可以自行选择需要的方式进行上传。
- 对于**日志文件**的上传，我们支持日志的**普通形式**与**.zip**的形式进行上传。
- 对于**恶意程序**的上传，我们支持使用**单个样本**的形式与**压缩包**的形式进行上传。

五、部署与使用

您可以查看我们项目中的README.md进行项目的部署配置，详细的过程我们再次便不再赘述了，如果还存在部署配置的问题，请联系我们。

六、常见问题解答

(一) 数据库配置问题

您在本地克隆了我们的项目之后，需要对数据库配置进行修改，下面我们会详细给您介绍需要修改哪些部分：

vuln_backend/src/config.py

```
1 class Config:
2     SECRET_KEY = os.environ.get('SECRET_KEY') or 'your-secret-key'
3     #连接串 下面这一行要根据您的mysql密码来填写: SQLALCHEMY_DATABASE_URI =
4     'mysql+pymysql://root:您的数据库密码@localhost:3306/nvd_database'
5     SQLALCHEMY_DATABASE_URI =
6     SQLALCHEMY_TRACK_MODIFICATIONS = False
7     PORT = 3000
```

这个文件是我们的数据库配置文件，在这里需要配置连接串，您需要根据您的数据库配置进行自行的修改，我们在注释中已经给出了一个示例说明。

vuln_backend/src/apps/services/CnvdDataInfoImpl.py

```
1 def delVulnCnvd(vuln_id):
2     db = pymysql.connect(host="localhost", user="root", password="您的数据库密
3     码", database="nvd_database")
4     cursor = db.cursor()
5
6 def updateVulnCnvd(id, data):
7     db = pymysql.connect(host="localhost", user="root", password="你的数据库密
8     码", database="nvd_database")
9     cursor=db.cursor()
```

这个文件中您也需要将您的数据库密码进行输入，我们需要连接您部署在本地的数据库，这样才能够复现我们的功能。

在完成这两个文件中的配置之后，数据库问题一般就得到解决了，如果存在问题，请联系我们。

七、联系我们

下面是我们的邮箱，您可以根据邮箱来联系我们：

2312796@mail.nankai.edu.cn

2313781@mail.nankai.edu.cn

2313508@mail.nankai.edu.cn

2312323@mail.nankai.edu.cn