

CA Filter 使用手册

目录

1. 背景	1
2. 简介	1
3. 使用方法	2
3.1 根据标准 CA 告警文件过滤	2
3.2 根据保存的结果过滤	3
3.3 忽略特定文件的 CA 告警	4
3.4 使用 MCAnalyzer 进行过滤	4
4. 工作原理	5

1. 背景

由于我们的 SVTI 项目原来的代码有大量的 CA 告警，对代码 CA 的时候我们自己的 CA 告警就淹没其中不太好找；另一方面，由于持续集成环境的需要每次对整个项目的代码进行 CA，海量的 CA 告警会让上传者找不到自己产生的 CA 告警在哪里。于是写了这个小工具，如果有什么 BUG 及时反馈哦。

2. 简介

CA_Filter 是一个 CA 内容过滤工具。CA_Filter 工作前要对一个标准的 CA 告警文件进行学习，然后对要过滤的 CA 告警文件进行过滤，如果某条 CA 告警是之前在学习阶段遇到的，就把这条 CA 告警过滤掉。

CA_Filter 可以嵌入到 Source Insight 中，CA_Filter.exe 的参数和 CAnalyzer.bat 完全一样，其他设置也一样。CA_Filter 可以保存和读入学习结果避免每次 CA 的时候都要进行重复的学习。CA_Filter 也可以进行 MCA。

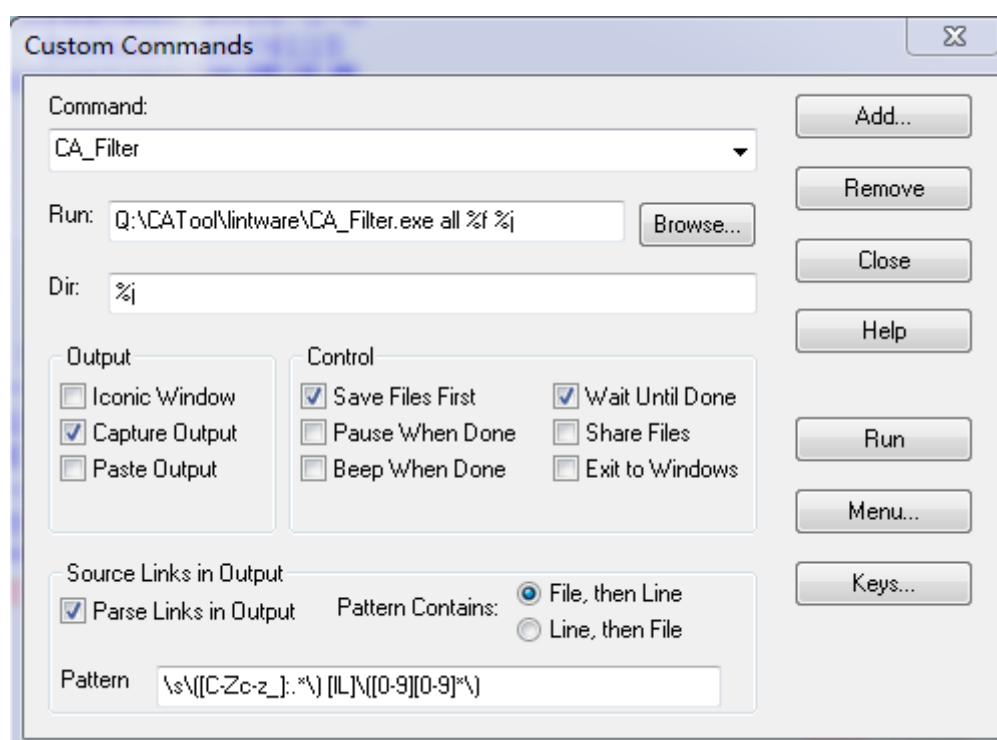


图 2.1 Source 中嵌入 CA Filter

3. 使用方法

3.1 根据标准 CA 告警文件过滤

操作步骤：

- (1) 拷贝 CA_Filter.exe 到 CATool\lintware 目录下
- (2) 在 CATool\lintware 目录下新建 CA_Filter_Study.txt，用记事本打开，直接复制某次 CA 的输出，或者把希望被过滤的 CA 告警复制到其中。
- (3) CA 过滤时，在 Source 的“Option”---“Custom Commands”中参照图 2.1 的设置，然后点“Run”即可。

在编辑 CA_Filter_Study.txt 时要注意以下几点：

- (1) 在编辑时最好使用 Windows 自带的记事本而不是 Notepad++编辑。这是因为未过滤的原 CA 告警文件采用的是“windows 的”ANSI 编码。
- (2) 复制 CA 告警到这个文件时，确保一个字符都不能少或多（空行除外），这是因为CA_Filter会计算告警信息的MD5，如果发生变化则认为是不同的。
- (3) 每个 CA 告警之间至少要有一个空行，每个 CA 告警和其他信息之间也最好用空行隔开。注意某些行看似空行但其实有一个下划线“_”。
- (4) 第一行不要存放 CA 告警。建议第一行类似普通 CA 结果一样：

```
--- 10:55:47.01 ALL CAnalyzer X:\XXX\XXXX\XXXXXX
```

完成过滤之后，如果希望查看原版的未经过滤的 CA 告警，打开 CATool\lintware\CA_Filter_Result_Origin.txt 即可。

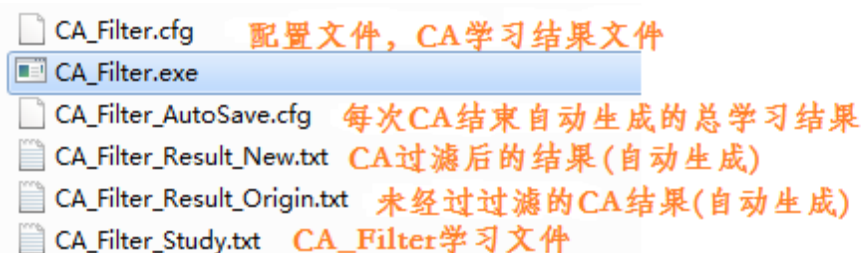


图 3.1 CA Filter 目录结构

3.2 根据保存的结果过滤

CA_Filter 每次 CA 完都会将本次学习的结果保存到 CA_Filter_AutoSave.cfg(包括读取的 CA_Filter.cfg)。但进行过滤工作时并不使用这个文件，而是 CA_Filter.cfg(该文件不存在则不读取)。把 CA_Filter_AutoSave.cfg 重命名为 CA_Filter.cfg，这样 CA 过滤的时候就会根据保存的结果进行过滤。

CA_Filter.cfg 的内容可以直接编辑，由两部分组成，第一部分是要忽略 CA 的项目文件（见 3.3 节），第二部分是每条 CA 告警的 MD5。

CA_Filter 工作流程如图 3.2 所示，CA_Filter 在每次过滤前，会先读取 CA_Filter.cfg(如果不存在则跳过步骤①)，然后对 CA_Filter_Study.txt 进行学习(如果不存在则跳过步骤②)。CA_Filter_Study.txt 是标准的 CA 告警文件，用户可以往里面添加要过滤掉的 CA 告警内容。



图 3.2 CA Filter 的工作流程

举个 CA_Filter_Study.txt + CA_Filter.cfg 结合使用的例子。其中 CA_Filter.cfg 存放项目原来存在的大量 CA 告警的学习结果，CA_Filter_Study 用来存放我们自己产生的又要忽略的 CA 告警（方便经常性地修改）：

- （1）新建 CA_Filter_Study.txt，将项目原来存在的大量 CA 告警拷贝到其中。删除 CA_Filter.cfg，运行 CA_Filter.exe，得到 CA_Filter_AutoSave.cfg
- （2）将 CA_Filter_AutoSave.cfg 重命名为 CA_Filter.cfg
- （3）删除然后新建 CA_Filter_Study.txt，添加我们自己产生的 CA 告警到里面

3.3 忽略特定文件的 CA 告警

有时候我们想过滤某个文件的所有 CA 告警，则可以在 CA_Filter.cfg 中添加希望要忽略 CA 告警的文件名（不带路径名），每个文件名写一行，且以“--”开头，如图 3.3 所示：

```
# The Files with CA error to be ignore:
--ipsec_main.c
--tunnel_xml_tunnels.c
--ipsec_data_issucheck.c

# The MD5 of CA error:
0005C640AA9F01D31E1FD385DBEB43D0
001D7ACEC98A2094133D3F55909A262E
00242C48FF08BC9932DCEF53FDCCE6B
0033126C732342CC1466E26A9C24CDA0
003D384F4817CEB48B8809E78407EA29
00483F9548BE8E1CCF602ED5B0DB6C89
0057835AB43C43A61B85AEB3490DEBAA
0057F9F8D7DEF80C61DF0F462A92B85A
```

图 3.3 CA_Filter.cfg 内容

3.4 使用 MCAAnalyzer 进行过滤

将 CA_Filter.exe 复制一份重命名为 MCA_Filter.exe 即可用来进行 MCA，Source Insight 中的设置与 MCAAnalyzer 一致，如图 3.4 所示。

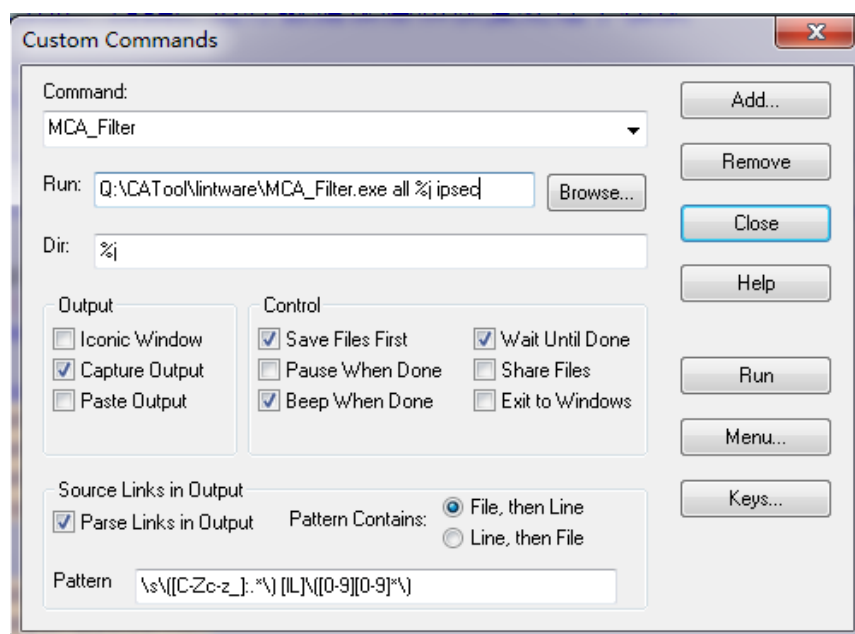


图 3.4 MCA 在 Source Insight 中的配置

4. 工作原理

CA_Filter 识别某条 CA 告警是通过对 CA 告警的特征进行 MD5 运算来实现的，每条 CA 告警用一个 MD5 字符串来唯一标识。若两个 CA 告警的特征相同，则他们的 MD5 也相同，反之亦然。

而一个 CA 告警的特征由以下四部分组成：

- (1) 引起 CA 告警的那一行代码
- (2) 引起 CA 告警的文件名
- (3) CA 告警类型和告警 ID
- (4) CA 告警错误消息

```
STATIC VOID _SetControllerTypeByPara(IN INT iArgc, IN CHAR * const *pcArgv)  
E:\SVTI\Source\VPN\src\sbin\ipsec\app\ipsec_main.c L1025 Info 3700  
violates name guidelines, 'pcArgv' should be 'ppcXXXX'
```

图 4.1 CA 告警的特征提取

如果两个 CA 告警的以上四点完全一致（每个字符都相同），那么 CA_Filter 则认为他们是两个相同的 CA 告警。

CA_Filter 简要工作流程如下：

- (1) 尝试读取配置文件（里面含有学习后的 MD5 列表）
- (2) 尝试读取一个标准的 CA 告警文件学习，里面存放希望被过滤的 CA 告警：
 - a. 把每条告警按段分隔开来
 - b. 计算每条 CA 告警的特征的 MD5（若告警本身含有 MD5 则用本身的）
 - c. 将所有的 MD5 保存成“MD5 列表”到内存中
- (3) 调用 CAnalyzer.bat 脚本进行 CA，得到原 CA 告警信息，并保存成文件。
- (4) 读取原 CA 告警信息的文件，查询保存的 MD5 列表进行过滤：
 - a. 把每条告警按段分隔开来
 - b. 读取一段 CA 告警，计算其特征的 MD5（若告警本身含有 MD5 则用本身的）

- c. 如果这段 CA 告警的 MD5 在“MD5 列表”中找不到，则说明这条告警是新产生的，则把这段告警写到新的文件中。如果找到了则不写入。
- d. 不断地重复步骤 b 和 c 知道原 CA 告警文件读取结束