# 全文工具箱设计文档

1. 功能说明

增加“全文工具箱”SearchToolkit模块，计划包含目前已知的涉赌，涉黄，涉枪，飞机场4种依赖全文的线索模型。当前先实现涉赌模型，其他模型预留接口。

SearchToolkit模块依赖内网全文集群，现实中有**直连**和**堡垒机跳转**两种连接方式。目前只考虑**堡垒机跳转**模式。用户通过鼠标配置，执行模型，并可以下载结果到本机。业务过程类似侦察兵模块。

1. 业务逻辑
2. 用户选择涉赌模型后，提供堡垒机用户名，密码，堡垒机IP，全文机IP，C2通过SSH协议自动实现堡垒机登陆跳转到全文机，在默认文件夹上传涉赌脚本并远程后台执行。
3. 每个选中模型的每次执行，以“任务”的形式抽象表示，包括“+创建任务”，“任务详情”，“任务删除”。每个“任务”用一个唯一ID表示，暂定用pid+时间的组合。
4. “任务”在创建后，对应的模型脚本是在服务器上异步执行，每次点击‘任务详情’

C2去服务器获取任务状态后展示相关信息。

1. 持久化信息，用户名->全文工具箱->任务ID\_模型名称\_创建时间.bcp，内含堡垒机各种配置信息和任务相关情况，按\t分割。一个任务对应一个BCP
2. 查看“任务详情”，通过ssh协议用linux命令查看进程和结果文件信息来确定任务状态。
3. 默认的linux服务器上跑全文脚本的目录为/tmp/iao/search\_toolkit/{模型类型}，tmp每7天Linux自动清理一次。
4. 采用现有的胶水赌博脚本，将自定义的加密功能去掉
5. 交互设计
6. 左侧主面板增加一项“全文工具箱”，位于“侦察兵”下方。
7. 左侧针对“全文工具箱”增加一个二级面板，用来承载创建任务，选择模型，配置参数，查看任务详情等用户实际操作。
8. 针对每个任务，增加任务详情子窗口，包括堡垒机用户名，堡垒机密码，堡垒机IP，全文机IP，模型选择，结果下载，服务器上临时存储结果的目录等元素。
9. 每个‘任务’一个元素项，右键菜单：任务详情，任务删除
10. tooltip待设计
11. 说明
12. 整体类似‘侦察兵’的交互和逻辑，尤其是“异步的，被动的，客户触发的状态更新”这种设计取舍在其中有描述。
13. 待上传到服务器的脚本，用zip压缩，作为资源嵌入到项目中（待研究）
14. 业务逻辑代码放置于Business\SearchToolkit目录中
15. 界面代码按传统放置到对应的目录中
16. 对应的ssh工具包，约简单约小越好
17. 跳转下载，跳转上传功能尚未预研，如果理论上不可实现，就到时候去掉。
18. 实现
19. 将每次执行全文脚本的行为抽象为一次“任务”，用TaskInfo信息类来描述所用到的所有配置信息，比如堡垒机用户名，密码，IP，任务状态，任务创建时间（暂定）等；TaskInfo同时负责自我的读写持久化
20. 用TaskManager类来管理所有任务，比如新建，删除，生成任务ID，一些条件检查，Linux服务器上执行各种操作等。
21. SearchToolkitForm用来创建任务时的参数配置和查看任务详情时展示。两种状态下，界面有稍微不同，比如，查看任务详情时，会有任务ID和任务状态。还有一些参数检查，方便用户的界面操作等功能。
22. SearchToolkitControl用来展示所有任务，创建任务等，内含一个TaskManager用来实现所有业务逻辑。
23. SSH类提供ssh协议的所有基础功能，比如，登录，跳转，下载，执行命令。
24. BastionAPI类在SSH的基础上封装所有相关的业务逻辑操作，如：通过堡垒机跳转全文机，执行模型脚本，创建模型目录，下载模型结果，查看模型运行状态，PID等
25. 全文远程任务管理和BastionAPI类设计
26. Linux服务器上目录 / {RemoteWorkspace}/{PID}\_{TaskCreateTime}/result 存放某次任务的结果
27. Linux服务器上目录 / {RemoteWorkspace}/{PID}\_{TaskCreateTime}/script 存放某次任务的模型脚本
28. 获取结果时，根据C2上存储的PID和TaskCreateTime，查看目录是否有文件，进程PID是否已结束
29. 任务状态根据PID, 任务时间，结果文件来判断：

|  |  |
| --- | --- |
| 判断条件 | 结果 |
| PID不存在，结果文件存在 | 运行成功 |
| PID不存在，没有结果文件 | 运行失败 |
| PID存在，没有结果文件，未超时(72小时) | 正在运行 |
| PID存在，有结果文件 | 运行成功 |

1. 执行新任务时，发现重名文件，删除之；
2. 任务执行完毕后，服务器上的结果文件不删，等待7天后系统自动删除，或者使用者自己手工删除。
3. BastionAPI利用Linux命令完成以上功能