

# 安全工具可视化编排

星阑科技 汤青松

文档仅限技术交流，严禁商用，违者必究

## 个人简介

- u 汤青松
- u 网络安全从业8年经验、5年甲方安全体系建设落地经验
- u 实体书《PHP Web安全开发实战》作者
- u 开源项目 QingScan、蜻蜓、XssPlatform 作者

文档仅限技术交流，切勿商用，违者必究

一、工作场景

二、解决方案

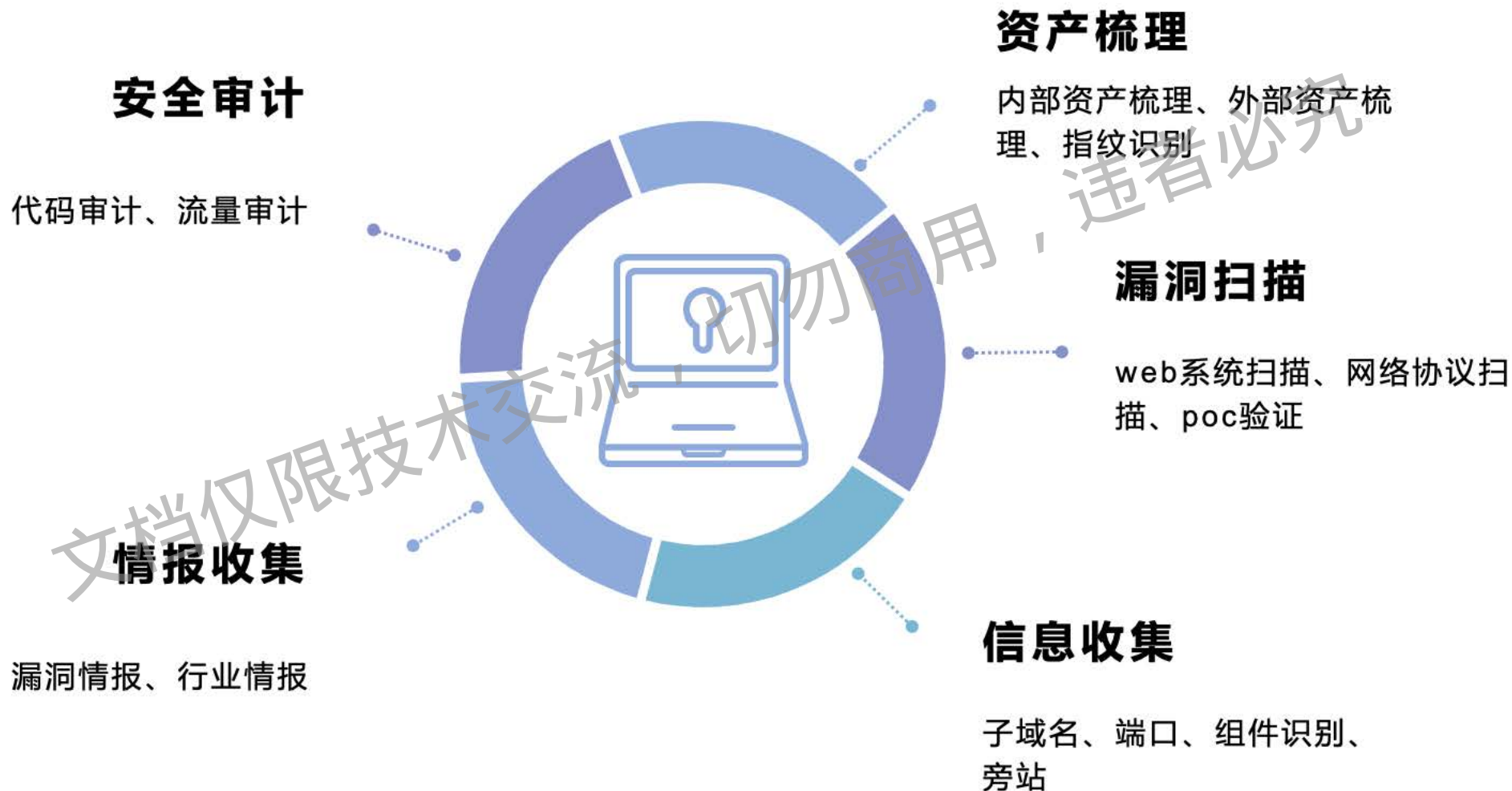
三、真实案例

文档仅限技术交流，切勿商用，违者必究

# 一、工作场景

文档仅限技术交流，切勿商用，违者必究

# 安全工作场景



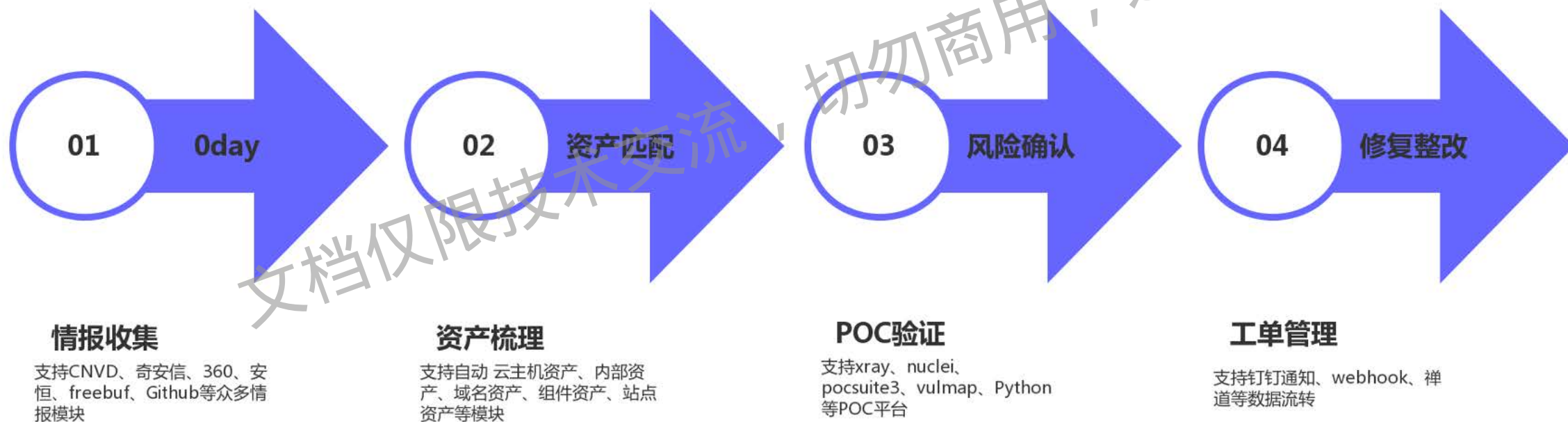
## 分享背景

安全工程师日常会用到大量工具和接口，不同的场景所使用的工具不同，安装方法不同、数据格式不同；

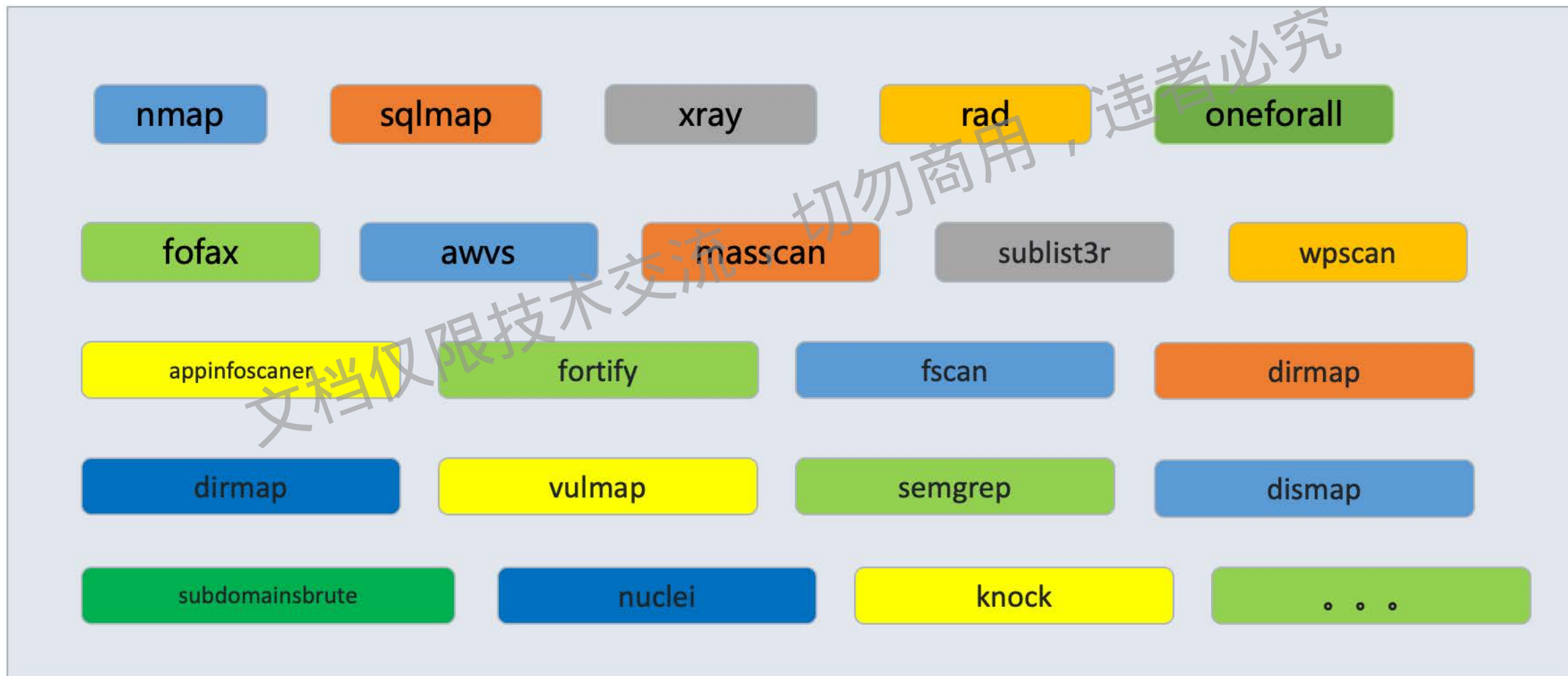
熟悉这些工具或接口需要花费不少精力，而且工具之间数据不一致，通常需要人工将数据加工才能进行流转，浪费大量精力在重复工作中

文档仅限技术交流，切勿商用，违者必究

## 例如：应急响应流程



# 常用安全工具





# 常用API接口

阿里云API	AWS API	安全组	域名API	fofa
shodan	ZoomEye	cnvd	微步在线	零零信安
当前IP	ICP备案	同主体域名	工商信息查询	SRC域名检索
Zabbix 告警	jumpser	飞书通知	钉钉通知	微信通知
禅道API	邮件发送	DNSPod	蓝信消息通知	...

工具多，使用方法复杂

安装复杂，环境依赖

一件事情，多个工具

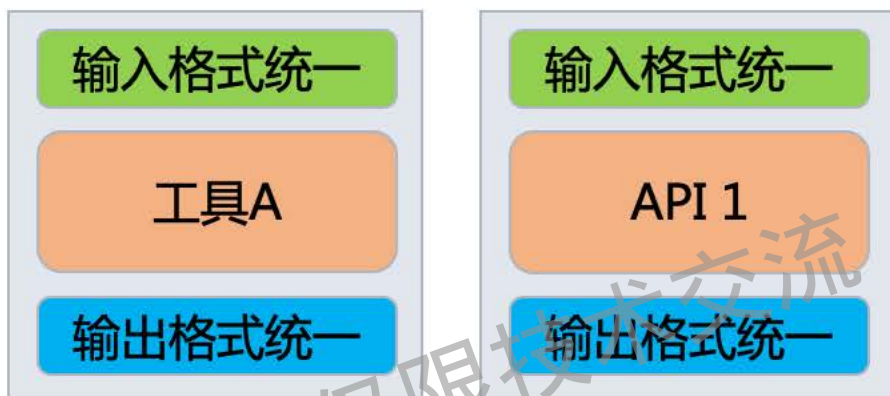
数据碎片化，数据不联动

文档仅限技术交流，切勿商用，违者必究

## 二、解决方案

文档仅限技术交流，切勿商用，违者必究

## 解决思路



- 工具二次封装接口
- 统一接口调用参数
- 产生的数据能流转

# 自动化编排系统

安全编排

## 剧本管理

渗透测试

代码审计

风险排查

...

## 流程编排引擎

安全接口

API 1

API 2

API 3

API 4

API 5

API 6

安全人员

安全监控

安全分析

响应处置

安全知识库

事件1

事件2

事件3

处置建议

黑白名单

...

安全工具

情报收集

WAF

代码审计

扫描器

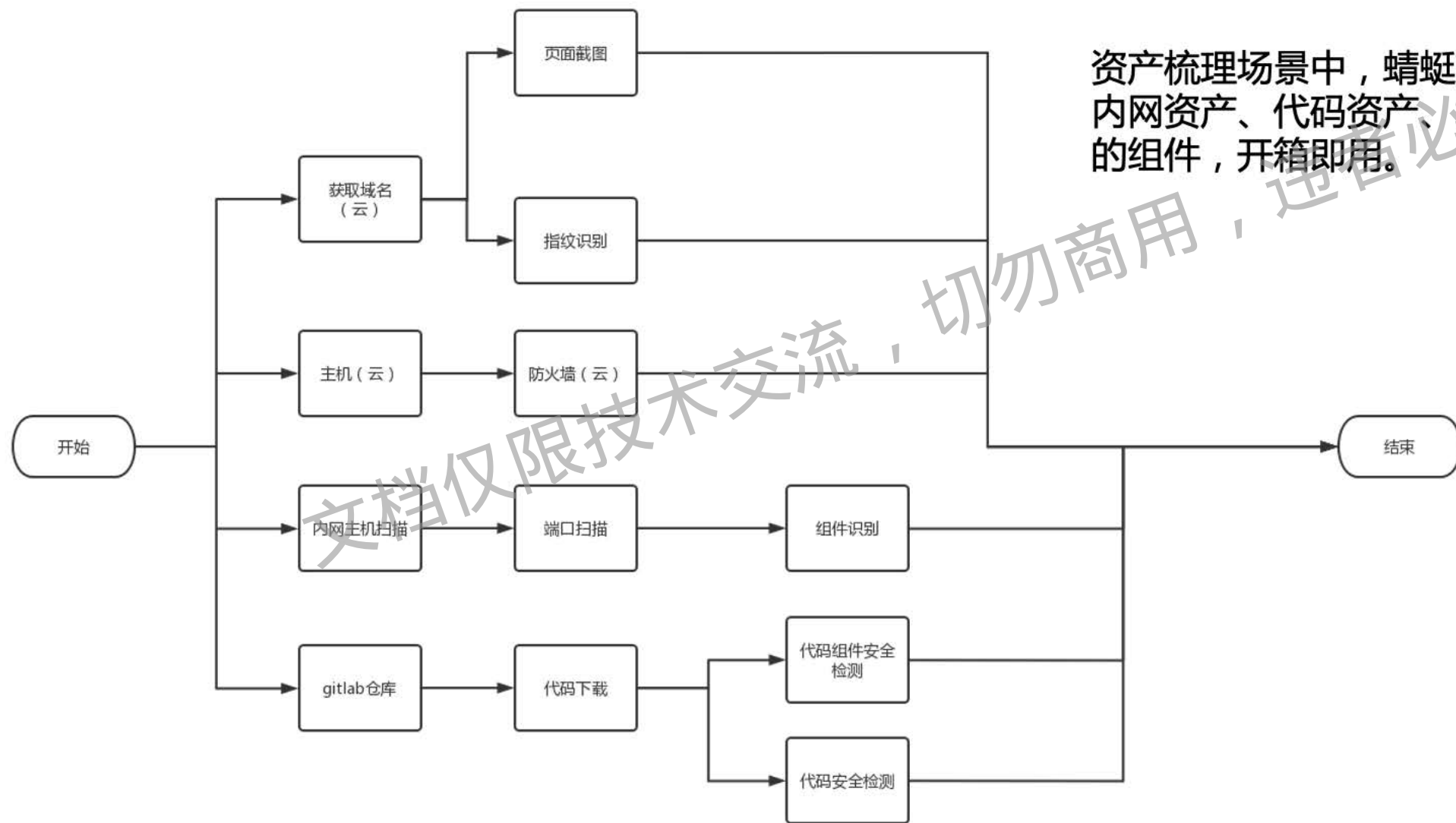
邮件网关

端口扫描

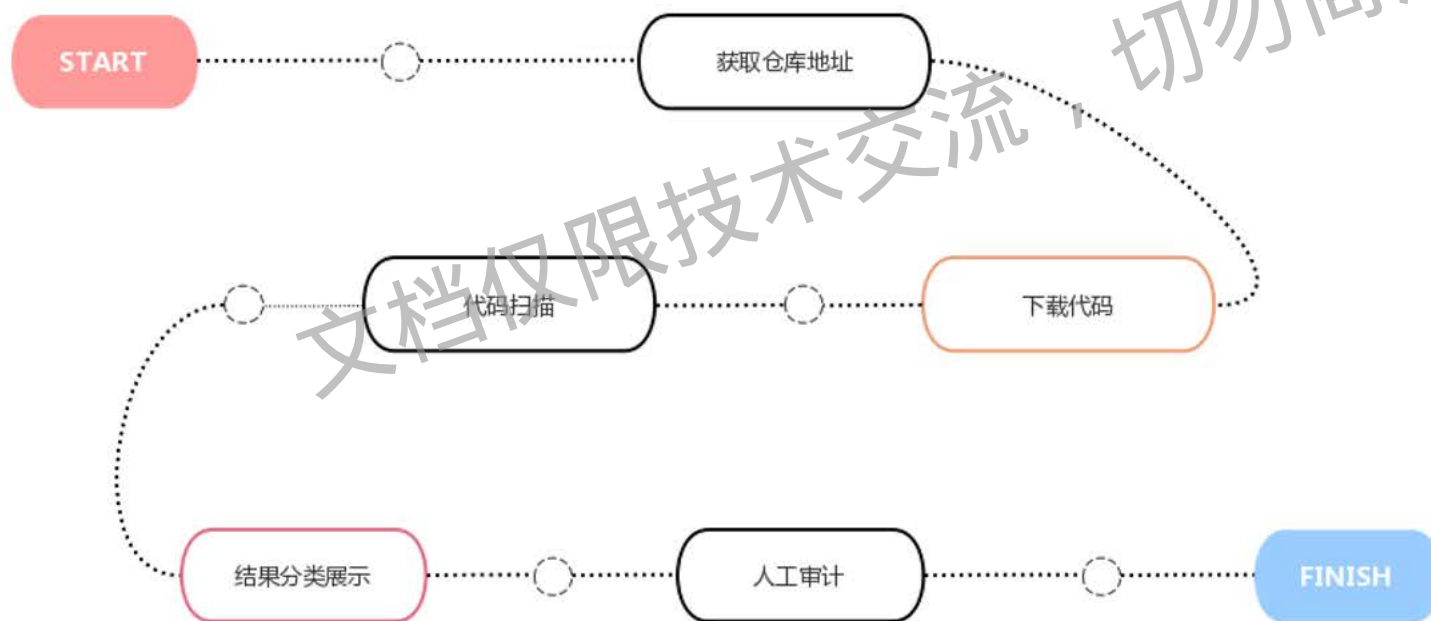
威胁情报

堡垒机

# 资产梳理

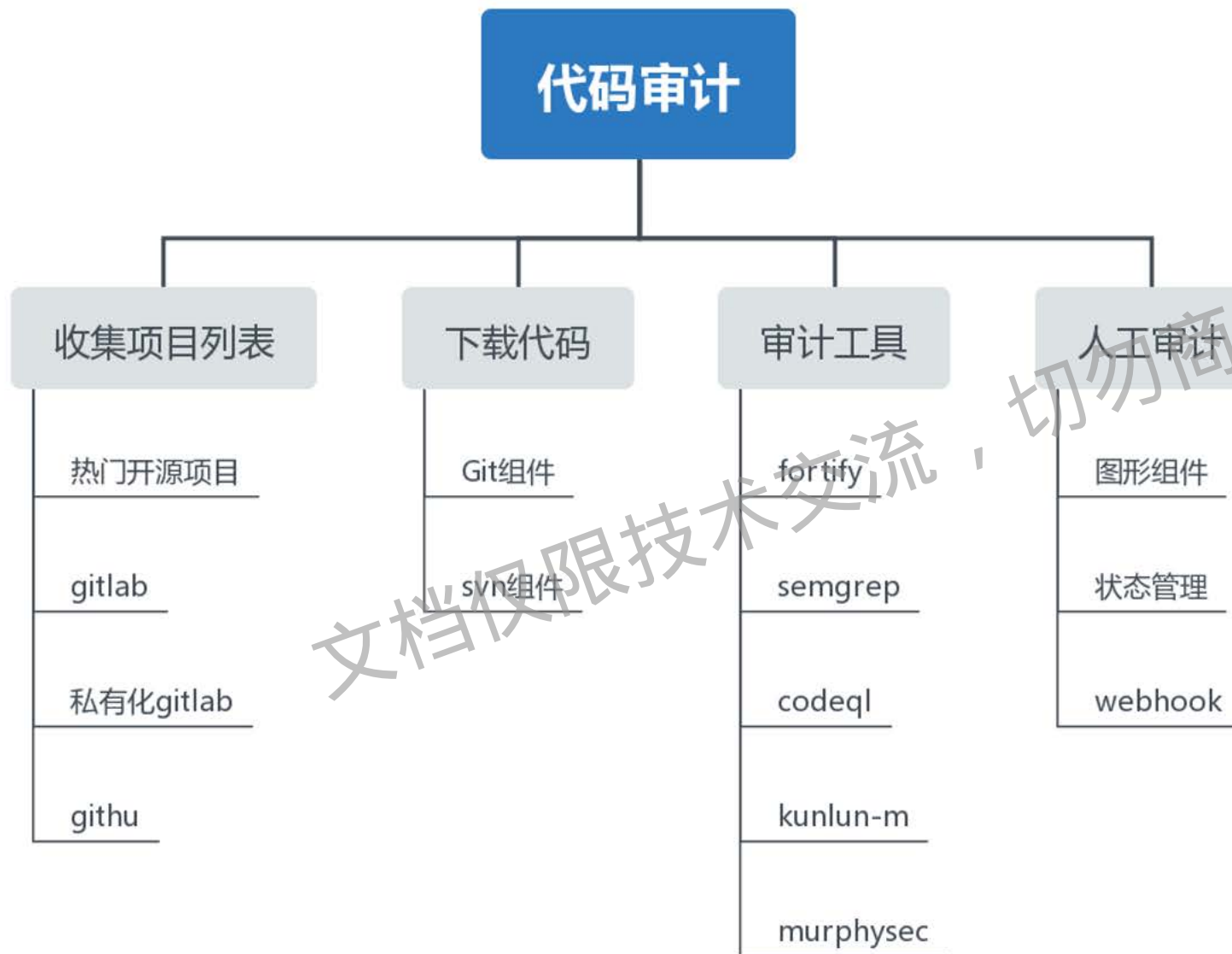


# 自动化代码审计



你只需要将场景流程构思出来，再使用蜻蜓的组件进行填充，即可快速编排出你的 workflow。



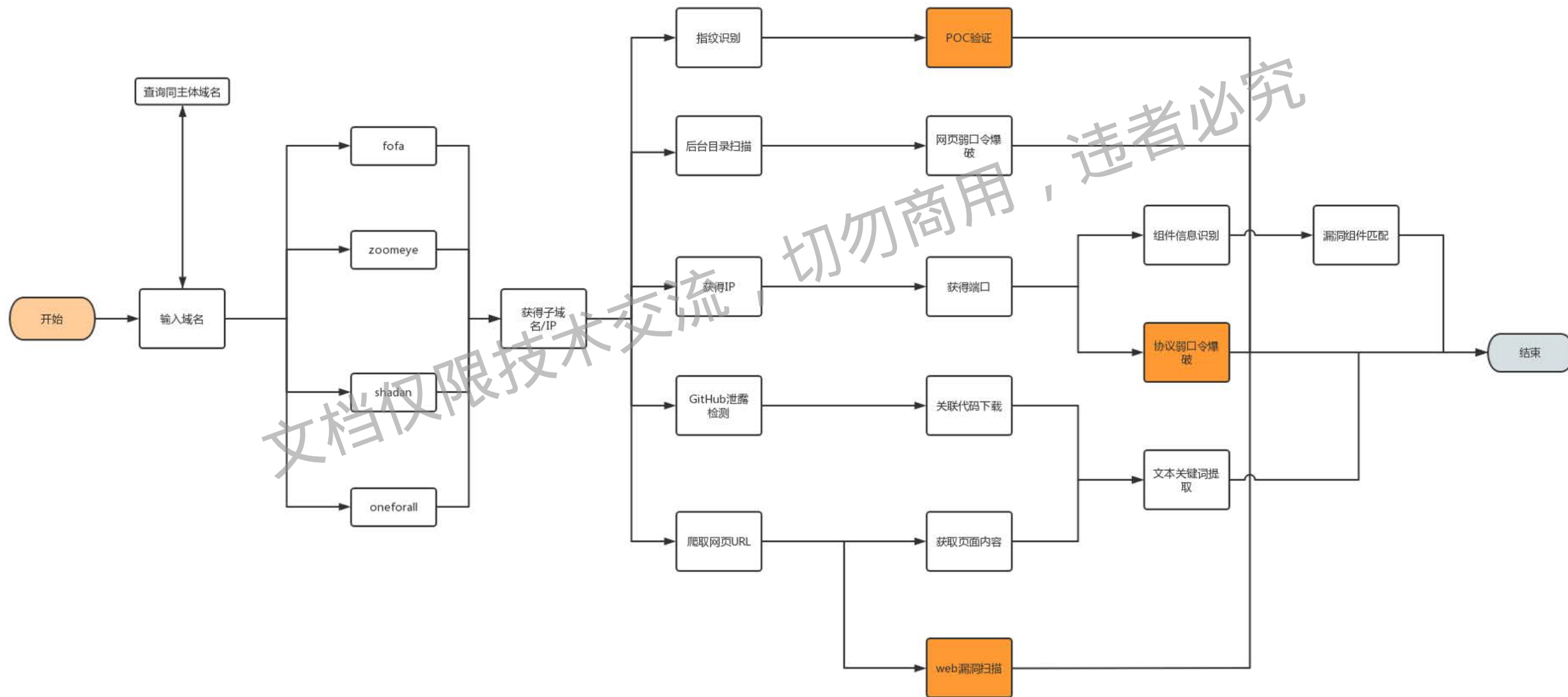


以代码审计的常用节点为例，蜻蜓内置了丰富的组件，基本可以满足90%的需求，另外蜻蜓的组件还在快速增长。

如果组件还不够满足你的需求，你也可以在自定义组件中将脚本代码置入~



# 信息收集



### 三、真实案例

文档仅限技术交流，切勿商用，违者必究

## 场景1：漏洞检测

检测一批URL是否存在0day漏洞，并将检测的漏洞信息钉钉通知到群里。



# 漏洞检测

01100111 11111100 01111101 01111101 11011001 11001010 11101000 10011110 11101111 10100000 10010111 00100001 00010111 01000011 00011100  
11100010 10011100 01100010 01011111 11010011 10001100 10001101 01110101 10010000 01011011 01110000 10111110 10110010 10110101 10011001  
10011011 01001001 11100010 01011011 11000101 10001011 01000011 01000011 00011100 01000010 11100100 00011010 10010010 10000010  
00100011 00011111 00001000 01011011 01001111 10100101 01111001 0101010 01000110110111 10010011 10000000 01100110  
10111001 01010010 01100011 00000100 00001011 10011100 01010010 10001100 11010100 10001101 01100111 00000101  
00100011 11010010 10000111 00011000 10011101 01111001 10010111 11000011 10100100 01001000 00110111 00100001  
01000101 01010001 00100101 11110001 00110101 00110101 00010000 01001000 00110101 11111100 01111101 01111101 11011001  
10010101 11010000 10011110 11101111 10100000 11111100 01111101 01111101 01000101 11000101 11000101 11000101 11000101  
10001100 10001101 01110101 10010000 01011011 11111100 01111101 01111101 11000101 11000101 11000101 11000101  
10001011 01000011 01000111 00011110 01011011 11111100 01111101 01011011 11000101 11000101 11000101 11000101  
10100101 01111001 00111001 11001101 11001101 11001101 11001101 11001101 11001101 11001101 11001101 11001101  
10011100 00101000 00010110 00111010 10011011 10011011 10011011 10011011 10011011 10011011 10011011 10011011  
01111011 01011010 10001100 10101010 11011011 11011011 11011011 11011011 11011011 11011011 11011011 11011011  
00100100 11010110 10101100 01011111 11011011 11011011 11011011 11011011 11011011 11011011 11011011 11011011  
10010111 00100001 00010111 01000011 00011100 01110001 10011100 11011011 11011011 11011011 11011011 11011011  
01110000 10111110 10110010 10110101 10011001 11001011 11001011 11001011 11001011 11001011 11001011 11001011  
00000010 11100100 00011010 10010010 10000010 00100011 01010010 01100011 00011101 01011101 01011101 01011101  
0100011011101111 10010011 10000000 01100110 10111001 01010010 01100011 10011100 00101000 00010110 00111010 10000100  
10011001 11010100 10001101 01100111 00000101 00100011 11010010 10001101 01111011 01011010 10001100 10101010 11101100  
11100001 10100100 01001000 00100011 00010011 01010001 00100101 11110001 00111010 10101100 01011111 00111001  
01100111 11111100 01111101 01111101 11011011 10100000 01011011 11011011 11011011 11011011 11011011 11011011  
11100010 10011100 01100010 01011011 10011011 10011011 10011011 10011011 10011011 10011011 10011011 10011011  
11001011 01001001 11100010 01011011 10011011 10011011 10011011 10011011 10011011 10011011 10011011 10011011  
00100011 00011111 00001000 01011011 10011011 10011011 10011011 10011011 10011011 10011011 10011011 10011011  
10111001 01010010 01100011 00001000 01011011 10011011 10011011 10011011 10011011 10011011 10011011 10011011  
00100011 11010010 10000111 00001000 01011011 10011011 10011011 10011011 10011011 10011011 10011011 10011011  
01000101 01010001 00100011 11111001 11111001 11111001 11111001 11111001 11111001 11111001 11111001 11111001  
11001010 11101000 10011110 11101101 11101101 11101101 11101101 11101101 11101101 11101101 11101101 11101101  
10001100 10001101 01110101 10011000 10011000 10011000 10011000 10011000 10011000 10011000 10011000 10011000  
10001011 01000011 01000111 00011100 01110001 10011100 11011011 11011011 11011011 11011011 11011011 11011011  
10100101 01111001 00111001 10011011 10011011 10011011 10011011 10011011 10011011 10011011 10011011 10011011  
10011100 00010100 00010110 00111010 10011011 10011011 10011011 10011011 10011011 10011011 10011011 10011011  
01111011 01011010 10001100 10101010 11101101 11101101 11101101 11101101 11101101 11101101 11101101 11101101  
00100100 11010110 10101100 01011111 00111001 11011011 11011011 11011011 11011011 11011011 11011011 11011011  
10010111 00100001 00010111 01000011 00011100 01110001 10011100 11011011 11011011 11011011 11011011 11011011  
01110000 10111110 10110010 10110101 10011001 11001011 11001011 11001011 11001011 11001011 11001011 11001011  
00000010 11100100 00011010 10010010 10000010 00100011 01010010 01100011 00011101 01011101 01011101 01011101  
0100011011101111 10010011 10000000 01100110 10111001 01010010 01100011 10011100 00101000 00010110 00111010 10000100  
10011001 11010100 10001101 01100111 00000101 00100011 11010010 10000111 00011000 10011101 01111011 01011010 10001100 10101010 11101100  
11100001 10100100 01001000 00110111 00100001 01000101 01010001 00100101 11110001 00110111 00100100 11010110 10101100 01011101 00111001

标准流程：

- ① 获取目标列表
- ② 处理数据格式
- ③ 调用工具检测
- ④ 结果推送钉钉

流程简单，传统实现至少半天。

蜻蜓安全平台可以在五分钟内实现一个流程编排，从获得URL到将结果推送到钉钉



工作台

市场

节点管理

HTTP发包

脚本

读数据库

过滤器

读取文件

文本导入

容器

我的组件

组件市场

NIKTO扫描器

钟馗之眼

POC脚本

扫描器

containers

containers

钉钉通知

containers

containers

▶

发布

查看结果

HTTP发包  
Clone代码目标

脚本  
过滤部分目标

容器  
扫描器

脚本  
WebHook

脚本

配置

描述

上游数据变量引用

ID	扫描器
----	-----

脚本语言

Select

编写代码

代码内容

保存

发布

## 漏洞检测

在蜻蜓安全平台中只需要拖动几个组件按钮，将必要的参数往上面填写即可；

流程：`获取URL内容`->`对数据做过滤`->`扫描器扫描`->`钉钉通知`；

前后可能不会超过五分钟时间，就可以把需求做完。而且会发现这个图中，不需要代码却可以让打造适合自己的安全工具；



## 账号设置

个人信息

我的资产

## 漏洞上报

漏洞上报

批量上报漏洞

漏洞规则报送

事件型漏洞批量上报

## 我的任务

我的任务

## 我的关注

资产关联漏洞

漏洞收藏

## 我的荣誉

我的证书

原创漏洞积分

## 我的消息



您好, 尊敬的c , 欢迎回来!

手机: | 邮箱: |

荣誉值: | 用户积分: |

立即上传漏洞

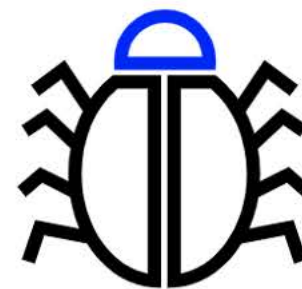
您有( 8 )条消息未读, 请前往系统消息查看!

编号	漏洞标题	状态	上报时间	评论/关注	操作
CNVD-C-2022-08	有限公司存在SQL注入漏洞	已归档	2022-06-26	0/0	<a href="#">跟踪</a>
CNVD-C-2022-05	有限公司存在SQL注入漏洞	已作废	2022-06-26	0/0	<a href="#">跟踪</a>
CNVD-C-2022-04	有限公司存在SQL注入漏洞	已归档	2022-06-26	0/0	<a href="#">跟踪</a>
CNVD-C-2022-303	有限公司存在SQL注入漏洞	已作废	2022-06-26	0/0	<a href="#">跟踪</a>
CNVD-C-2022-301	有限公司存在SQL注入漏洞	已归档	2022-06-26	0/0	<a href="#">跟踪</a>
CNVD-C-2022-300	有限公司存在SQL注入漏洞	已作废	2022-06-26	0/0	<a href="#">跟踪</a>
CNVD-C-2022-797	有限公司存在SQL注入漏洞	已归档	2022-06-26	0/0	<a href="#">跟踪</a>
CNVD-C-2022-796	有限公司存在SQL注入漏洞	已归档	2022-06-26	0/0	<a href="#">跟踪</a>
CNVD-C-2022-794	有限公司存在SQL注入漏洞	已作废	2022-06-26	0/0	<a href="#">跟踪</a>
CNVD-C-2022-792	有限公司存在SQL注入漏洞	已归档	2022-06-26	0/0	<a href="#">跟踪</a>

1 2 3 4 5 下页 共 44 条

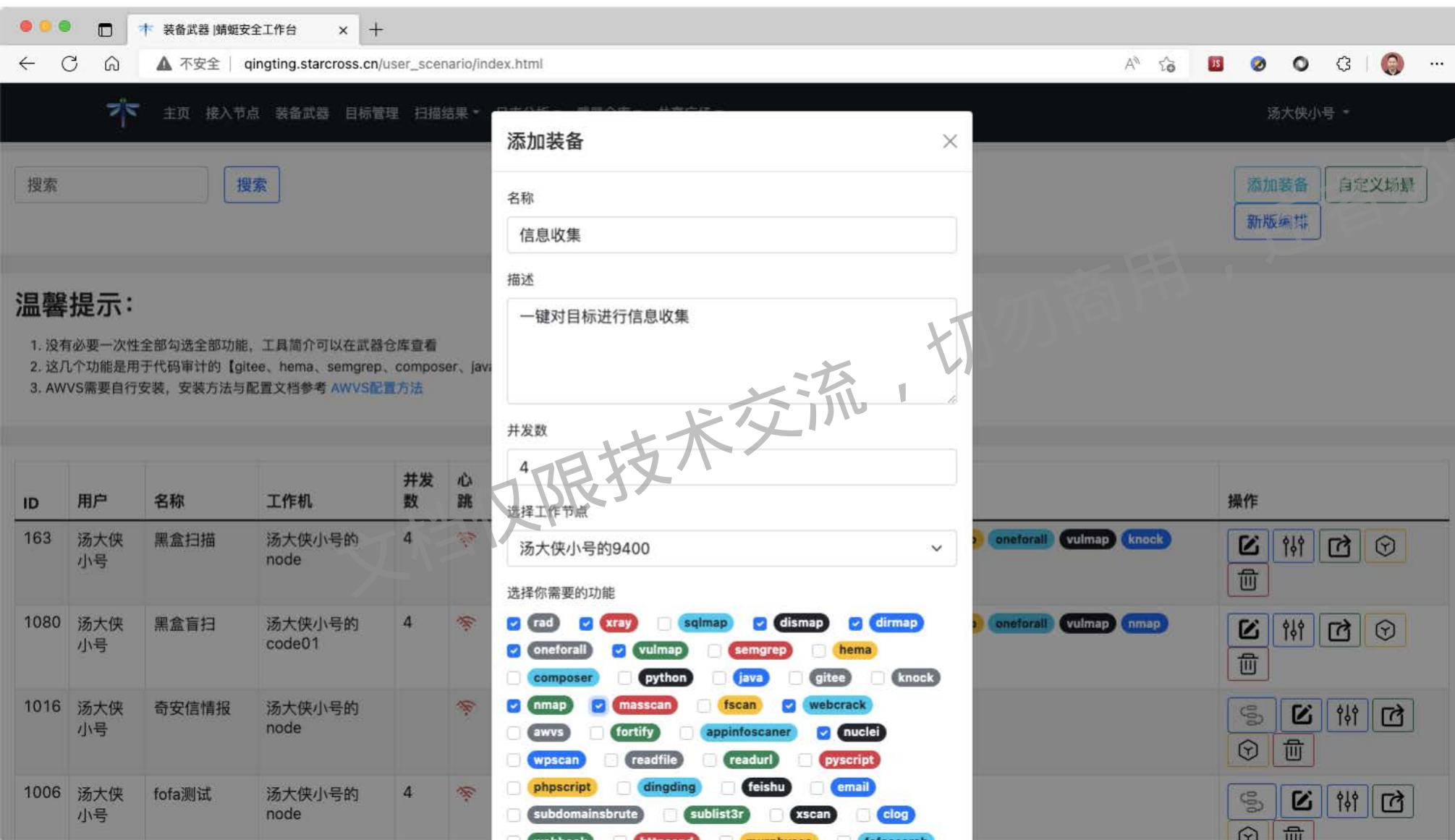
## 场景2：信息收集

添加URL地址，调用多个工具扫描，并将结果展现。



文档仅限技术交流，切勿商用，违者必究





一些场景中，你仅需要收集信息，并不会将工具A数据传递到工具B，你可以使用简易的工具包模式，将你所需的工具勾选；

然后使用此工具包对目标进行扫描，蜻蜓会自动调用你所选的工具对目标扫描，极大地提高你的工作效率~

基本信息

id: 65091

状态: 扫描中

名称: 未命名

URL: <http://txy8g.songboy.site:8888/>

创建: 2022-07-06 11:51:05

工具【dirmap】结果

重扫

ID	检测目标	URL	添加时间
1117292	未命名(txysongboy.site)	http://txy8g.songboy.site:8888/.git/FETCH_HEAD	2022-07-06 11:51:30
1117293	未命名(txysongboy.site)	http://txy8g.songboy.site:8888/.git/HEAD	2022-07-06 11:51:30
1117290	未命名(txysongboy.site)	http://txy8g.songboy.site:8888/.git/config	2022-07-06 11:51:30
1117291	未命名(txysongboy.site)	http://txy8g.songboy.site:8888/.git/description	2022-07-06 11:51:30
1117304	未命名(txysongboy.site)	http://txy8g.songboy.site:8888/.git/index	2022-07-06 11:51:30
1117294	未命名(txysongboy.site)	http://txy8g.songboy.site:8888/.git/info/exclude	2022-07-06 11:51:30
1117296	未命名(txysongboy.site)	http://txy8g.songboy.site:8888/.git/logs/HEAD	2022-07-06 11:51:30
1117295	未命名(txysongboy.site)	http://txy8g.songboy.site:8888/.git/logs/refs/heads/master	2022-07-06 11:51:30
1117297	未命名(txysongboy.site)	http://txy8g.songboy.site:8888/.git/logs/refs/remotes/origin/HEAD	2022-07-06 11:51:30
1117298	未命名(txysongboy.site)	http://txy8g.songboy.site:8888/.git/logs/refs/remotes/origin/master	2022-07-06 11:51:30


工具【dismap】结果

重扫

ID	检测目标	组件名称	技术栈	添加时间
5390	未命名(txysongboy.site)	None	[302] [Bootstrap] [Nginx]	2022-07-06 11:52:39

工具【fscan】结果

重扫

标准输出	文件输出	创建时间
 start infoscans 已完成 0/0 [*] 扫描结束,耗时: 177.774μs fscan version: 1.7.1		2022-07-06 11:51:56

在蜻蜓工具包模式中，你可以勾选你需要的工具，添加目标后，你所勾选的工具会对目标进行扫描，并将结果进行展示。



### 工具【masscan】结果

[重扫](#)

标准输出	文件输出	创建时间
Discovered open port 8888/tcp on 49.233.184.147 Discovered open port 22/tcp on 49.233.184.147 Discovered open port 3389/tcp on 49.233.184.147 Discovered open port 3309/tcp on 49.233.184.147 Discovered open port 111/tcp on 49.233.184.147 Discovered open port 2376/tcp on 49.233.184.147 Discovered open port 80/tcp on 49.233.184.147 Discovered open port 8443/tcp on 49.233.184.147 Discovered open port 880/tcp on 49.233.184.147 Discovered open port 7000/tcp on 49.233.184.147 Discovered open port 443/tcp on 49.233.184.147 Discovered open port 7500/tcp on 49.233.184.147		2022-07-06 11:51:37

### 工具【nmap】结果

[重扫](#)

标准输出	文件输出	创建时间
8888/tcp open sun-answerbook		2022-07-06 11:54:31
22/tcp open ssh		2022-07-06 11:54:31
3389/tcp open ms-wbt-server		2022-07-06 11:54:31
3309/tcp open tns-adv		2022-07-06 11:54:31
111/tcp open rpcbind		2022-07-06 11:54:31
2376/tcp open docker		2022-07-06 11:54:31
80/tcp open http		2022-07-06 11:54:31
8443/tcp open https-alt		2022-07-06 11:54:32
880/tcp open unknown		2022-07-06 11:54:32
7000/tcp open afs3-fileserver		2022-07-06 11:54:32

## 场景3：代码批量扫描

给你一个Gitlab代码仓库地址，需要你对代码进行安全分析，并将结果推送到指定地址。







# 漏洞检测

### 标准流程：

- ① 调用gitlab的API获得仓库列表
- ② 将代码下载到本地
- ③ 使用代码审计工具扫描
- ④ 将结果推送到项目管理系统

流程很简单，实现需要花费不少时间。

集成大量组件，只需要将所需的组件拖放到画布中并连续，就可以将流程串联起来

工作台

市场

节点管理

HTTP发包

脚本

读数据库

过滤器

读取文件

文本导入

容器

我的组件

组件市场

NIKTO扫描器

钟馗之眼

POC脚本

扫描器

containers

containers

钉钉通知

containers

containers

containers

发布

查看结果

脚本

获取GitLab仓库

脚本

Clone代码

容器

墨菲代码扫描

脚本

WebHook

脚本

配置

描述

节点名字

WebHook

详细描述

保存

发布



用户筛选

目标筛选

漏洞类型

搜索

搜索

ID	用户	目标	文件	漏洞类型	扫描工具	添加时间	操作
29181	aasd1w32a2	pig	pig-auth/src/main/java/com/pig4cloud/pig/auth/s...	java.log4j.security.log4j-message-lookup-injection...	semgrep	2022-06-15 01:28:56	<a href="#">详情</a>
29180	aasd1w32a2	pig	pig-auth/src/main/java/com/pig4cloud/pig/auth/s...	java.log4j.security.log4j-message-lookup-injection...	semgrep	2022-06-15 01:28:56	<a href="#">详情</a>
29179	aasd1w32a2	pig	docker-compose.yml	yaml.docker-compose.security.writable-filesystem-...	semgrep	2022-06-15 01:28:56	<a href="#">详情</a>
29178	aasd1w32a2	pig	docker-compose.yml	yaml.docker-compose.security.writable-filesystem-...	semgrep	2022-06-15 01:28:56	<a href="#">详情</a>
29177	aasd1w32a2	pig	docker-compose.yml	yaml.docker-compose.security.writable-filesystem-...	semgrep	2022-06-15 01:28:56	<a href="#">详情</a>
29176	aasd1w32a2	pig	docker-compose.yml	yaml.docker-compose.security.writable-filesystem-...	semgrep	2022-06-15 01:28:56	<a href="#">详情</a>
29175	aasd1w32a2	pig	docker-compose.yml	yaml.docker-compose.security.writable-filesystem-...	semgrep	2022-06-15 01:28:56	<a href="#">详情</a>
29174	aasd1w32a2	pig	docker-compose.yml	yaml.docker-compose.security.writable-filesystem-...	semgrep	2022-06-15 01:28:56	<a href="#">详情</a>
29173	aasd1w32a2	pig	docker-compose.yml	yaml.docker-compose.security.writable-filesystem-...	semgrep	2022-06-15 01:28:56	<a href="#">详情</a>
29172	aasd1w32a2	pig	docker-compose.yml	yaml.docker-compose.security.writable-filesystem-...	semgrep	2022-06-15 01:28:56	<a href="#">详情</a>
29171	aasd1w32a2	pig	docker-compose.yml	yaml.docker-compose.security.writable-filesystem-...	semgrep	2022-06-15 01:28:56	<a href="#">详情</a>
29170	aasd1w32a2	pig	docker-compose.yml	yaml.docker-compose.security.writable-filesystem-...	semgrep	2022-06-15 01:28:56	<a href="#">详情</a>



←

↶

🏠

⚠ 不安全 | qingting.starcross.cn/audit/fortify\_detail.html?hash=afca05ebb3f844af02b56af4a4d29fb5

🔊

🔍

☆

JS

🌐

🔌

⚙

👤

...

id	640
create_time	2022-05-20 11:55:58
Category	SQL Injection
Folder	Critical
Kingdom	Input Validation and Representation
Abstract	Line 24 of mysql_func.php invokes a SQL query built using input coming from an untrusted source. This call could allow an attacker to modify the statement's meaning or to execute arbitrary SQL commands.
Priority	Critical
Primary	<div>--- FileName: mysql_func.php FilePath: core/mysql_func.php LineStart: "24" Snippet: "\n //6.\uFFFFD\uFFFFD\uFFFFD\uFFFFD\uFFFFD\uFFFFDsq\uFFFFD\uFFFFD\uFFFFD\uFFFFD\uFFFFD\uFFFFD\n \ \$res = mysqli_query(\$link, \$sql);\n\n" TargetFunction: mysqli_query() ...</div>
Source	<div>--- FileName: cate.php FilePath: admin/action/cate.php LineStart: "27" Snippet:  2- if (!empty(\$_POST['cname'])) { \$pid = \$_POST['pid']; \$cname = \$_POST['cname'];  \$sql = "insert into bbs_cate(pid,cname) values('\$pid','\$cname')"; ...</div>



蜻蜓安全：一个便捷、免费、开源的安全工具协作平台。

文档仅限技术交流，切勿商用，违者必究

## 联系微信

官网地址: <http://qingting.starcross.cn>

GitHub: <https://github.com/StarCrossPortal/QingTing>



**THANKS**

文档仅限技术交流，切勿商用，违者必究