

什么是"源代码安全审计 (白盒扫描)" ?

- > 由于开发人员的技术水平和安全意识各不相同, 导致可能开发出一些存在安全漏洞的代码。
- > 攻击者可以通过渗透测试来找到这些漏洞, 从而导致应用被攻击、服务器被入侵、数据被下载、业务受到影响等问题。
- > "源代码安全审计"是指通过审计发现源代码中的安全隐患和漏洞, 而Cobra可将这个流程自动化。

Cobra为什么能从源代码中扫描到漏洞?

- > 对于一些特征较为明显的可以使用正则规则来直接进行匹配出, 比如硬编码密码、错误的配置等。
- > 对于OWASP Top 10的漏洞, Cobra通过预先梳理能造成危害的函数, 并定位代码中所有出现该危害函数的地方, 继而基于Lex(Lexical Analyzer Generator, 词法分析生成器)和Yacc(Yet Another Compiler-Compiler, 编译器代码生成器)将对应源代码解析为AST(Abstract Syntax Tree, 抽象语法树), 分析危害函数的入参是否可控来判断是否存在漏洞 (目前仅接入了PHP-AST, 其它语言AST接入中)。

Cobra和其它源代码审计系统有什么区别或优势?

- > Cobra定位是自动化发现源代码中大部分显著的安全问题, 对于一些隐藏较深或特有的问题建议人工审计。
- 开发源代码 (基于开放的MIT License, 可更改源码)
- 支持开发语言多 (支持十多种开发语言和文件类型)
- 支持漏洞类型多 (支持数十种漏洞类型)
- 支持各种场景集成 (提供API也可以命令行使用)
- 专业支持, 持续维护 (由白帽子、开发工程师和安全工程师一起持续维护更新, 并在多家企业内部使用)

Cobra支持哪些开发语言?

- > 目前Cobra主要支持PHP、Java等主要开发语言及其它数十种文件类型, 并持续更新规则和引擎以支持更多开发语言, 具体见支持的[开发语言和文件类型](#)。

Cobra能发现哪些漏洞?

- > 覆盖大部分Web端常见漏洞和一些移动端 (Android、iOS) 通用漏洞, 具体见支持的[漏洞类型](#)。

| ID | Label | Description(EN) | Description(CN) |
|-----|-------|------------------------------------|----------------------------------|
| 110 | MS | Misconfiguration | 错误的配置 |
| 120 | SSRF | Server-Side Forge | 服务端伪造 |
| 130 | HCP | Hard-coded Password | 硬编码密码 |
| 140 | XSS | Cross-Site Script | 跨站脚本 |
| 150 | CSRF | Cross-Site Request Forge | 跨站请求伪造 |
| 160 | SQLI | SQL Injection | SQL注入 |
| 163 | XI | Xpath Injection | Xpath注入 |
| 165 | LI | LDAP Injection | LDAP注入 |
| 167 | XEI | XML External Entity Injection | XML实体注入 |
| 170 | FI | Local/Remote File Inclusion | 文件包含漏洞 |
| 180 | CI | Code Injection | 代码注入 |
| 181 | CI | Command Injection | 命令注入 |
| 190 | IE | Information Exposure | 信息泄露 |
| 200 | PPG | Predictable Pseudorandom Generator | 可预测的伪随机数生成器 |
| 210 | UR | Unvalidated Redirect | 未经验证的任意链接跳转 |
| 220 | HRS | HTTP Response Splitting | HTTP响应拆分 |
| 230 | SF | Session Fixation | SESSION固定 |
| 260 | US | unSerialize | 反序列化漏洞 |
| 280 | DF | Deprecated Function | 废弃的函数 |
| 290 | LB | Logic Bug | 逻辑错误 |
| 320 | VO | Variables Override | 变量覆盖漏洞 |
| 350 | WF | Weak Function | 不安全的函数 |
| 355 | WE | Weak Encryption | 不安全的加密 |
| 970 | AV | Android Vulnerabilities | Android漏洞 |
| 980 | IV | iOS Vulnerabilities | iOS漏洞 |
| 999 | IC | Insecure Components | 引用了存在漏洞的三方组件(Maven/Pods/PIP/NPM) |

Cobra能应用在哪些场景?

1. 【漏洞出现前】通过内置的扫描规则对公司项目进行日常扫描, 并推进解决发现的漏洞。
2. 【漏洞出现后】当出现一种新漏洞, 可以立刻编写一条Cobra扫描规则对公司全部项目进行扫描来判断受影响的项目。

Cobra是什么类型应用?

- > Cobra提供Web服务的同时也提供了命令行服务。

1. 【CLI】通过命令行扫描本地源代码, 发现其中安全问题。
2. 【API&GUI】以Web Server形式部署在服务器上, 供内部人员通过GUI的形式访问使用, 并可以通过API集成到CI或发布系统中。

如何参与Cobra开发?

- > Cobra发展离不开开源社区的贡献, Cobra欢迎有Python开发经验且对代码审计感兴趣的人加入到我们的开源社区开发团队 (QQ群: 578732936) 共同参与贡献。

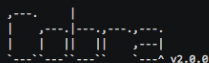
互联网相关企业如何接入Cobra?

> Cobra欢迎任何互联网相关企业免费接入使用，并提供技术支持，可以联系我们（QQ：3504599069，加之前请备注企业名称）。

Cobra文档

- 安装
 - [Cobra安装](#)
- 基础使用
 - [CLI模式使用方法](#)
 - [API模式使用方法](#)
- 进阶使用
 - [高级功能配置](#)
 - [升级框架和规则源](#)
- 规则开发规范
 - [规则模板](#)
 - [规则样例](#)
 - [规则文件命名规范](#)
 - [规则开发流程](#)
- 框架引擎
 - [开发语言和文件类型定义](#)
 - [漏洞类型定义](#)
 - [危害等级定义](#)
 - [程序目录结构](#)
- 贡献代码
 - [单元测试](#)
 - [贡献者](#)

```
usage: cobra [-h] [-v] [-t <target>] [-f <format>] [-o <output>]
            [-r <rule_id>] [-d] [-sid SID] [-H <host>] [-P <port>]
```



GitHub: <https://github.com/wuifeifei/cobra>

Cobra is a static code analysis system that automates the detecting vulnerabilities and security issue.

optional arguments:

```
-h, --help            show this help message and exit
-v, --version          show program's version number and exit
```

Scan:

```
-t <target>, --target <target>
                        file, folder, compress, or repository address
-f <format>, --format <format>
                        vulnerability output format (formats: html, json, csv,
                        xml)
-o <output>, --output <output>
                        vulnerability output STREAM, FILE, HTTP API URL, MAIL
-r <rule_id>, --rule <rule_id>
                        specifies rules e.g: CVI-100001,cvi-190001
-d, --debug            open debug mode
-sid SID, --sid SID    scan id(API)
```

RESTful:

```
-H <host>, --host <host>
                        REST-JSON API Service Host
-P <port>, --port <port>
                        REST-JSON API Service Port
```

Usage:

```
./cobra.py -t tests/vulnerabilities
./cobra.py -t tests/vulnerabilities -r cvi-190001,cvi-190002
./cobra.py -t tests/vulnerabilities -f json -o /tmp/report.json
./cobra.py -t https://github.com/wuifeifei/vc.git -f json -o feei@feei.cn
./cobra.py -t https://github.com/wuifeifei/vc.git -f json -o http://push.to.com/api
./cobra.py -H 127.0.0.1 -P 80
+ cobra git:(master) ./cobra.py -t tests/vulnerabilities
[18:30:16] [WARNING] Dependency analysis cannot be done without finding dependency files
[18:30:16] [INFO] Unknown Framework
[18:30:16] [INFO] Static analysis
[18:30:16] [INFO] > Target: folder, Output: stream
[18:30:16] [INFO] > /Users/wuifeifei/Projects/cobra/tests/vulnerabilities/
[18:30:16] [INFO] > Language: php, Framework: Unknown Framework
[18:30:16] [INFO] > Files: 5, Extensions:5, Consume: 0.00018
[18:30:16] [INFO] [PUSH] 25 Rules
[18:30:16] [INFO] [CVI-rule.x] [STATUS] OFF, CONTINUE...
```



Cobra

Information Targets Vulnerabilities

Welcome to Cobra!

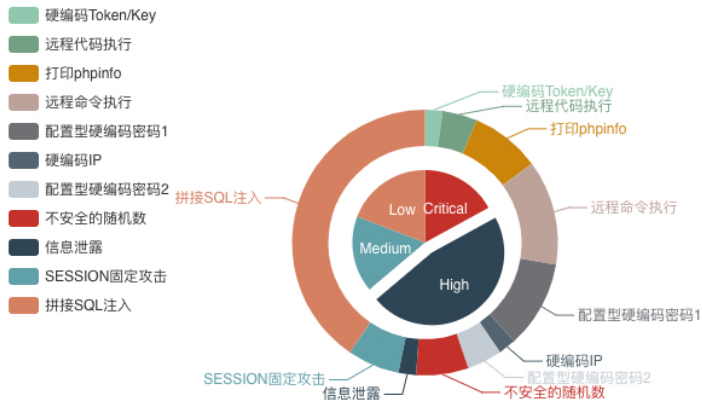
Information summary

| Item | Value |
|-------------------------|---------------------|
| Report time | 2017-08-29 13:06:07 |
| Total number of targets | 1 |

Number of vulnerabilities

| Level | Number |
|----------|--------|
| Critical | 8 |
| High | 22 |
| Medium | 8 |
| Low | 9 |
| Total | 47 |

Vulnerability distribution statistics



Vulnerabilities

MVE-36

vulnerabilities/sqli_blind/cookie-input.php:12

=> 2015-09-18 10:13:06

MVE-37

vulnerabilities/weak_id/source/high.php:11

=> 2017-04-17 22:57:23

MVE-38

vulnerabilities/weak_id/source/impossible.php:12

=> 2017-04-17 22:57:23

MVE-39

vulnerabilities/exec/source/low.php:10

=> 2015-09-27 19:15:44

MVE-40

vulnerabilities/exec/source/low.php:14

=> 2015-09-27 19:15:44

MVE-41

vulnerabilities/exec/source/high.php:26

=> 2015-09-27 19:15:44

MVE-42

vulnerabilities/exec/source/high.php:30

=> 2015-09-27 19:15:44

MVE-43

vulnerabilities/exec/source/low.php:10

```
1 <?php
2
3 if( isset( $_POST[ 'Submit' ] ) ) {
4     // Get input
5     $ip = $_REQUEST[ 'ip' ];
6
7     // Determine OS and execute the ping command.
8     if( striistr( php_uname( 's' ), 'Windows NT' ) ) {
9         // Windows
10         $cmd = shell_exec( 'ping ' . $ip );
11
12     }
13     else {
14         // *nix
15         $cmd = shell_exec( 'ping -c 4 ' . $ip );
16     }
17
18     // Feedback for the end user
19     $html .= "<pre>{$cmd}</pre>";
20 }
21
22 php
```

Cobra项目：<https://github.com/wufeifei/cobra>

点击收藏 | 3 关注 | 1

[上一篇：无弹窗渗透测试实验](#) [下一篇：企业安全工作要点思维导图](#)

1. 3 条回复



[chengable](#) 2017-09-05 05:30:35

没找到changelog，大致看了下代码，是主要增加了变量追踪吗

0 回复Ta



[xianzhi](#) 2017-09-05 06:38:26

主要简化了安装和使用成本；增加命令行调用模式；增加了AST；

0 回复Ta



[chengable](#) 2017-09-05 09:49:07

嗯嗯，更关注AST，期待之后的更新

0 回复Ta

[登录](#) 后跟帖

先知社区

[现在登录](#)

热门节点

[技术文章](#)

[社区小黑板](#)

目录

[RSS](#) [关于社区](#) [友情链接](#) [社区小黑板](#)