

漏洞扫描之AWVS

漏洞扫描

AWVS简介

AWVS安装

Docker安装(推荐)

Win&Linux安装

AWVS使用

添加目标

批量脚本

漏洞扫描之XRAY

XRAY简介

XRAY安装

XRAY破解

XRAY使用

爬虫模式

被动扫描

AWVS联动XRAY

BurpSuite联动XRAY

Rad联动XRAY

XRAY脚本编写

漏洞扫描之AWVS

#1课时

<https://www.acunetix.com/support/>

漏洞扫描

漏洞扫描是指基于漏洞数据库，通过扫描等手段对指定的远程或者本地计算机系统的安全脆弱性进行检测，发现可利用漏洞的一种安全检测（渗透攻击）行为。

- ✓ 针对某类漏洞的：sql注入（sqlmap）、weblogic（weblogicscan）
- ✓ 针对某类CMS的：wordpress（wpscan）、dedecms（dedecmscan）
- ✓ 针对系统应用层：nessus
- ✓ 针对某类框架的：Struts2（Struts2漏洞检查工具）、springboot（SBActuator）
- ✓ 针对web服务的：burpsuite、xray、awvs

AWVS简介

Acunetix web vulnerability scanner（简称AWVS）是一款知名的网络漏洞扫描工具，它通过网络爬虫测试你的网站安全，检测流行安全漏洞。从 11.0 版本开始，AWVS 就变成了使用浏览器端打开的形式，使用安装时自定义的端口来访问

AWVS安装

Docker安装(推荐)

1. 下载镜像

```
docker pull xiaomimi8/awvs14-log4j-2022
```

```
→ ~ → docker pull xiaomimi8/awvs14-log4j-2022
Using default tag: latest
latest: Pulling from xiaomimi8/awvs14-log4j-2022
7b1a6ab2e44d: Pull complete
b92844b7ec15: Pull complete
Digest: sha256:a5f07afb3d17fece450c1b5c1a2b7ddf39636b0500626704c9b48b742e71d196
Status: Downloaded newer image for xiaomimi8/awvs14-log4j-2022:latest
docker.io/xiaomimi8/awvs14-log4j-2022:latest
→ ~ → docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
xiaomimi8/awvs14-log4j-2022	latest	5ca608c9a84f	2 months ago	1.11GB

2. 启动容器

```
docker run -it -d -p 13443:3443 xiaomimi8/awvs14-log4j-2022
```

```
→ ~ → docker run -it -d -p 13443:3443 xiaomimi8/awvs14-log4j-2022
488aaae1ac3c844ea6ab279daa58a3972aa57e10850e0b88336f256ead767bc1
→ ~ → docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
488aaae1ac3c	xiaomimi8/awvs14-log4j-2022	"/bin/sh -c 'echo 12...'"	8 seconds ago	Up 6 seconds	0.0.0.0:13443->3443/tcp

3. 登录AWVS

URL地址: <https://ip:13443>

用户名: admin@admin.com

密码: Admin123

Win&Linux安装

AWVS14.6.220117111破解

链接: <https://pan.baidu.com/s/1To8wL96JeA1L1EkvybLRqg>

提取码: uodm

AWVS使用

添加目标

- Add a Target

☰

仪表盘

目标 >

漏洞

扫描

报告

Discovery >

用户 >

扫描配置文件

网络扫描程序

问题跟踪程序

WAFs

电子邮件设置

引擎

排除时间

代理设置

常规设置

关于

No Targets Found [Add a Target](#)



☰

仪表盘

目标 ▾

多个目标

目标组

漏洞

扫描

报告

Discovery >

用户 >

扫描配置文件

网络扫描程序

问题跟踪程序

WAFs

电子邮件设置

引擎

排除时间

代理设置

常规设置

多个目标

保存 导入 CSV 取消

☐ 仅网络扫描

地址

描述

添加另一目标





仪表盘

目标

多个目标

目标组

漏洞

扫描

报告

Discovery

用户

扫描配置文件

网络扫描程序

问题跟踪程序

WAFs

电子邮件设置

引擎

排除时间

代理设置

常规设置

目标设置

http://testphp.vulnweb.com/

扫描

保存

目标信息

描述

vulnweb

业务关键性

正常

默认扫描配置文件

Full Scan

扫描速度

10 个并发请求

No throttling

较慢 缓慢 适度 快速

网站登录



业务逻辑记录器

业务逻辑记录器可用于配置扫描程序扫描站点时中遵循的业务逻辑或各步骤清单

新建 BLR

导入 BLR

目标已成功创建

X

- 新建扫描



- 等待扫描

Acunetix

by Invicti

Administrator

扫描

Full Scan - http://120.27.6...

停止扫描

暂停扫描

生成报告

Export to

扫描信息

漏洞

网站结构

Scan Statistics

事件

N/A

Acunetix 威胁等级 0

威胁级别尚不可用。

活动

正在进行

总体进度

0%

Scanning of 120.27.61.239:8007 started

Mar 21, 2022, 7:12:51 PM

Antivirus not found

Mar 21, 2022, 7:12:51 PM

扫描持续时间

35s

请求

406

平均响应时间

51ms

路径已确认

18

目标信息

地址

http://120.27.61.239:8007/

服务器

Apache/2.4.7 (Ubuntu)

操作系统

Unix

已识别技术

响应式

Yes

最新警报

0001

批量脚本

<https://github.com/test502git/awvs14-scan>

漏洞扫描之XRAY

#1课时

<https://github.com/chaitin/xray>

XRAY简介

xray 是一款功能强大的安全评估工具，由多名经验丰富的一线安全从业者呕心打造而成，主要特性有：

- 检测速度快：发包速度快；漏洞检测算法高效。
- 支持范围广：大至 OWASP Top 10 通用漏洞检测，小至各种 CMS 框架 POC，均可以支持。
- 代码质量高：编写代码的人员素质高，通过 Code Review、单元测试、集成测试等多层验证来提高代码可靠性。
- 高级可定制：通过配置文件暴露了引擎的各种参数，通过修改配置文件可以极大的定制化功能。
- 安全无威胁：xray 定位为一款安全辅助评估工具，而不是攻击工具，内置的所有 payload 和 poc 均为无害化检查。

XRAY安装

- github项目地址：
<https://github.com/chaitin/xray>

- Releases:
<https://github.com/chaitin/xray/releases>
- 官方文档:
<https://docs.xray.cool/#/>

XRAY破解

使用二进制编辑器打开xray程序，修改如下值（随意修改一个字符）即可

```
43 4F 4D 4D 55 4E 49 54 59
COMMUNITY

41 4F 4D 4D 55 4E 49 54 59
AOMMUNITY

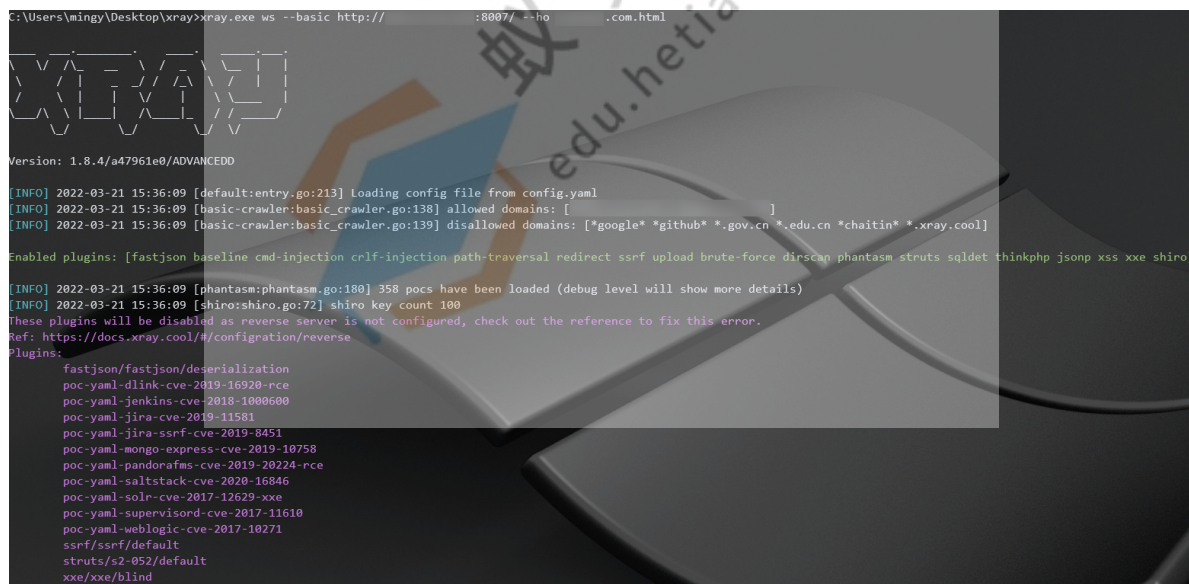
41 44 56 41 4E 43 45 44 44
ADVANCEDD
```

XRAY使用

爬虫模式

```
xray.exe webscan --basic-crawler http://xxx.com/ --html-output xray-xxx.html

xray.exe ws --basic http://xxx.com/ --ho xray-xxx.html
```



```
C:\Users\mingy\Desktop\xray>xray.exe ws --basic http://:8007/ --ho .com.html

XRAY

Version: 1.8.4/a47961e0/ADVANCEDD

[INFO] 2022-03-21 15:36:09 [default:entry.go:213] Loading config file from config.yaml
[INFO] 2022-03-21 15:36:09 [basic-crawler:basic_crawler.go:138] allowed domains: [ ]
[INFO] 2022-03-21 15:36:09 [basic-crawler:basic_crawler.go:139] disallowed domains: ["google" "github" *.gov.cn *.edu.cn "chaitin" *.xray.cool]

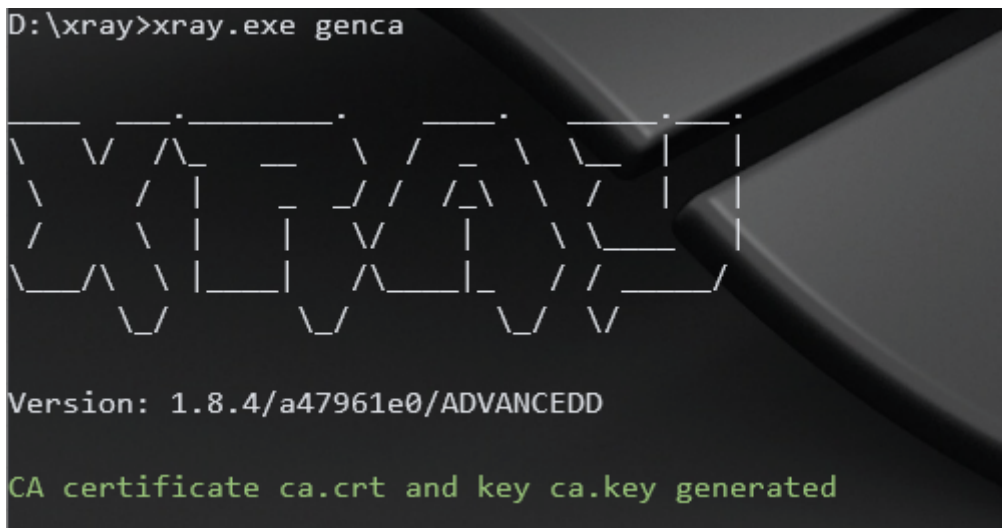
Enabled plugins: [fastjson baseline cmd-injection crlf-injection path-traversal redirect ssrf upload brute-force dirscan phantasm struts sqldet thinkphp jsonp xss xxe shiro]

[INFO] 2022-03-21 15:36:09 [phantasm:phantasm.go:180] 358 pocs have been loaded (debug level will show more details)
[INFO] 2022-03-21 15:36:09 [shiro:shiro.go:72] shiro key count 100
These plugins will be disabled as reverse server is not configured, check out the reference to fix this error.
Ref: https://docs.xray.cool/#/configuration/reverse
Plugins:
fastjson/fastjson/deserialization
poc-yaml-dlink-cve-2019-16920-rce
poc-yaml-jenkins-cve-2018-1000600
poc-yaml-jira-cve-2019-11581
poc-yaml-jira-ssrf-cve-2019-8451
poc-yaml-mongo-express-cve-2019-10758
poc-yaml-pandorafms-cve-2019-20224-rce
poc-yaml-saltstack-cve-2020-16846
poc-yaml-solr-cve-2017-12629-xxe
poc-yaml-supervisord-cve-2017-11610
poc-yaml-weblogic-cve-2017-10271
ssrf/ssrf/default
struts/s2-052/default
xxe/xxe/blind
```

被动扫描

1. 生成 `ca` 证书

```
xray.exe genca
```

2. 开启监听

完整: `xray.exe webscan --listen 127.0.0.1:7777 --html-output testphp.html`

简化: `xray.exe ws --listen 127.0.0.1:7777 --ho testphp.html`

3. 浏览器设置代理

 情景模式: xray

代理服务器

网址协议	代理协议	代理服务器	代理端口	
(默认)	HTTP 	127.0.0.1	7777	
 显示高级设置				

4. 浏览器访问待测试站点, 开启扫描

```
[INFO] 2022-03-23 10:13:15 [collector:mitm.go:215] loading cert from ./ca.crt and ./ca.key
[INFO] 2022-03-23 10:13:15 [collector:mitm.go:270] starting mitm server at 127.0.0.1:7777
[INFO] 2022-03-23 10:13:53 [default:dispatcher.go:433] processing GET http://120.27.61.239:8007/
[*] scanned: 0, pending: 1, requestSent: 535, latency: 42.68ms, failedRatio: 0.00%
[Vuln: dirscan]
Target      "http://120.27.61.239:8007/config/config.inc"
VulnType    "config/web"
Payload     "/config/config.inc"

[INFO] 2022-03-23 10:13:56 [default:dispatcher.go:433] processing GET http://120.27.61.239:8007/source/index.php?id=1
[Vuln: baseline]
Target      "http://120.27.61.239:8007/source/index.php?id=%24%7B822910330%2B879759905%7D"
VulnType    "sensitive/server-error"

[Vuln: xss]
Target      "http://120.27.61.239:8007/source/index.php?id=1"
VulnType    "reflected/default"
Payload     "<ScRiPt>alert(1)</sCrIpT>"
Position    "query"
ParamKey    "id"
ParamValue  "lkbyfxqewauklhbnumqt"

[Vuln: sqldet]
Target      "http://120.27.61.239:8007/source/index.php?id=1"
VulnType    "error-based/default"
Payload     "extractvalue(1,concat(char(126),md5(1754579187)))"
Position    "query"
ParamKey    "id"
ParamValue  "extractvalue(1,concat(char(126),md5(1754579187)))"
type        "heuristic"
title       "extractvalue function error based case"
```

AWVS联动XRAY

1. VPS启动Xray

```
./xray_linux_amd64 ws --listen 0.0.0.0:7777 --ho proxy.html
```

2. AWVS添加扫描目标

在添加扫描目标是配置代理服务器为VPS上Xray的监听端口

HTTP

HTTP 身份验证

客户端证书

代理服务器

协议

HTTP

地址

47.101.214.85

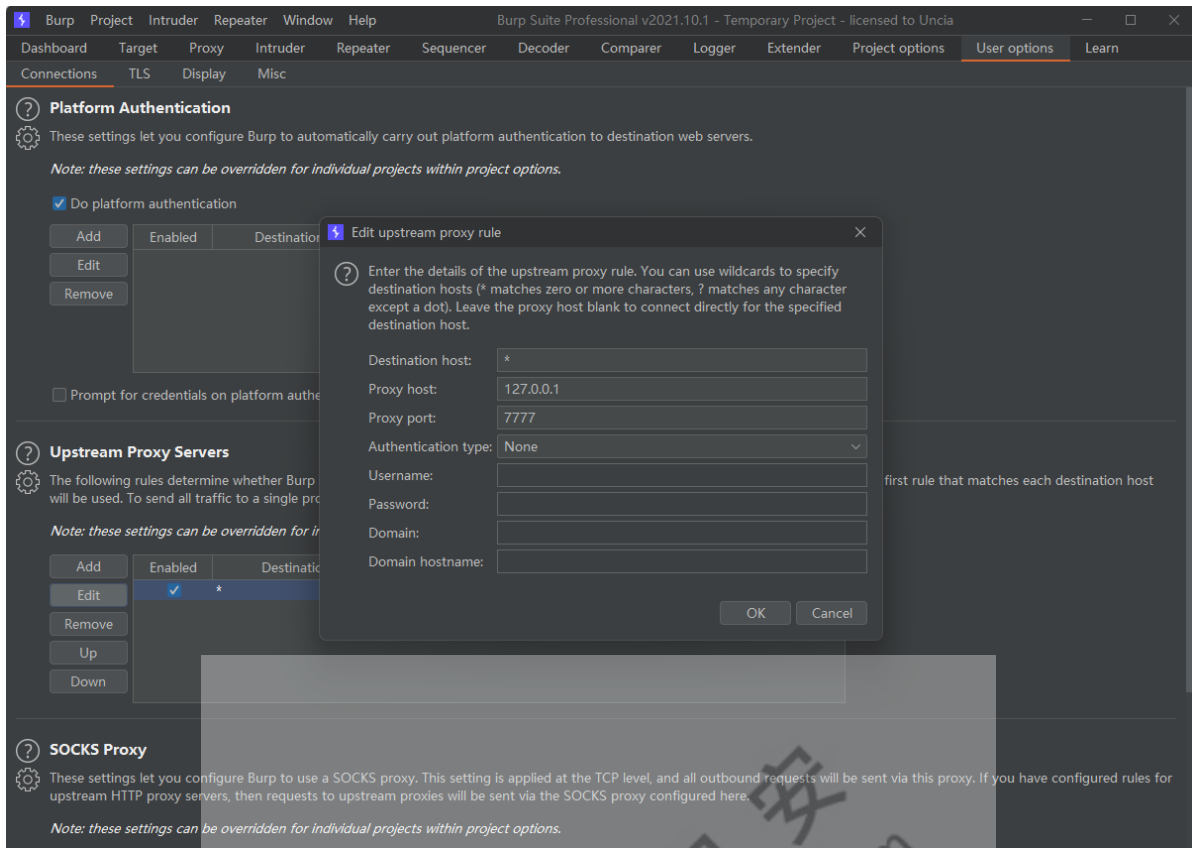
端口

7777

☐ 此代理服务器需要身份验证

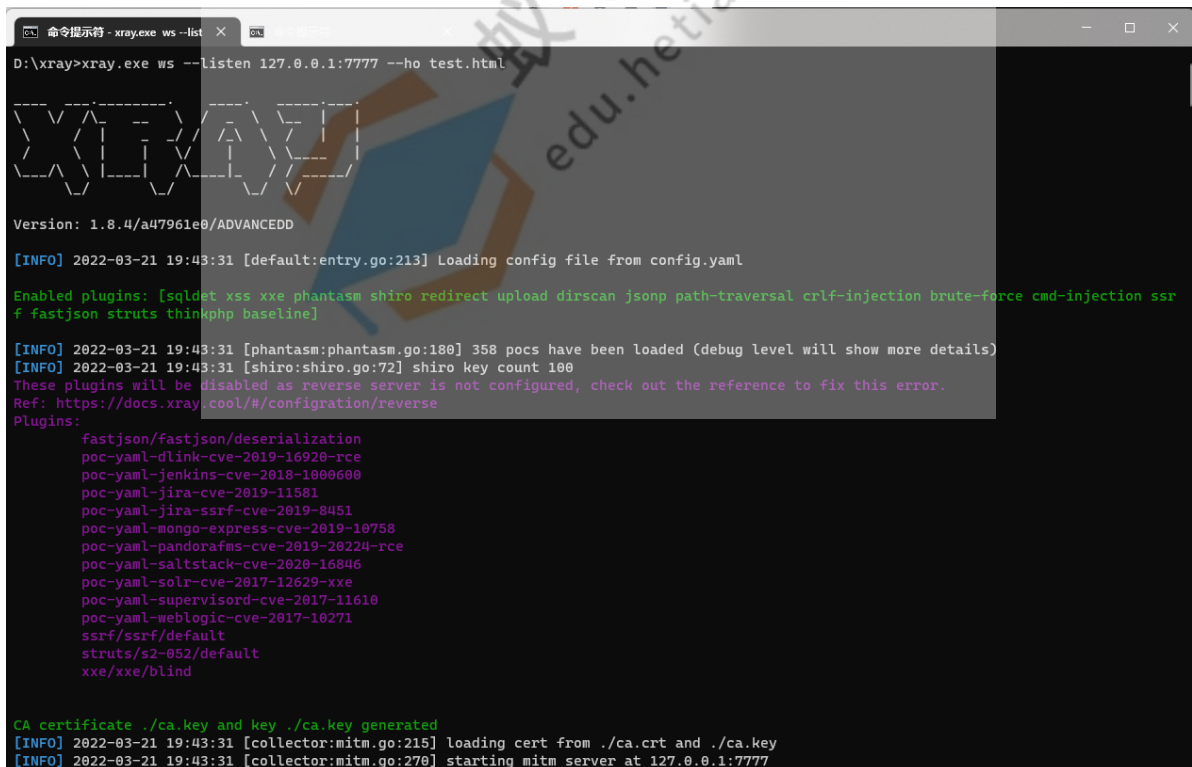
BurpSuite联动XRAY

1. User options -> Upstream Proxy Servers -> Add



2. 启动Xray监听

```
xray.exe ws --listen 127.0.0.1:7777 --ho test.html
```



Rad联动XRAY

1. 下载Rad

<https://github.com/chaitin/rad/releases>

2. Rad基本使用

- 基本使用

```
rad -t https://www.hetianlab.com/
```

- 手动登录

```
rad -t https://www.hetianlab.com/ -wait-login
```

执行以上命令会自动禁用无头浏览模式，开启一个浏览器供手动登录。
在登录完毕后在命令行界面点击回车键继续爬取。

- 将爬取基本结果导出为文件

```
rad -t https://www.hetianlab.com/ -text-output result.txt
```

3. Rad与Xray联动

```
xray.exe ws --listen 127.0.0.1:7777 --ho proxy.html
```

```
rad -t http://120.27.61.239:8007 -http-proxy 127.0.0.1:7777
```

4. 高级版Xray融合了Rad爬虫

```
xray ws --browser-crawler http://120.27.61.239:8007 --ho vuln.html
```

XRAY脚本编写

1. Xray POC编写辅助工具

<https://phith0n.github.io/xray-poc-generation/>

2. poc-nacos-unauth.yml

```
name: poc-nacos-unauth
groups:
  one:
    - method: GET
      path: /nacos/v1/auth/users?pageNo=1&pageSize=9
      follow_redirects: true
      expression: |
        response.status == 200 && r'"username":'.+?','.bmatches(response.body)
  two:
    - method: GET
      path: /v1/auth/users?pageNo=1&pageSize=9
      follow_redirects: true
      expression: |
        response.status == 200 && r'"username":'.+?','.bmatches(response.body)
detail:
  author: mingy
```

```
fofa: title="Nacos" && country!="CN"
```

3. 漏洞检测

```
xray ws -p mypocs/poc-nacos-unauth.yml -uf url.txt --ho nacos.html
```

4. 漏洞复现

<https://cloud.tencent.com/developer/article/1784279>

