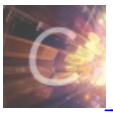
基于 Swoole 构建的 CTF AWD 比赛环境搭建与实践

原创

Rytia 于 2019-05-08 23:47:39 发布 1134 收藏 2

分类专栏： 网络安全 文章标签： CTF PHP Swoole

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。
本文链接：https://blog.csdn.net/winyuan789/article/details/89981514
版权



网络安全 专栏收录该内容
1 篇文章 0 订阅
订阅专栏

> Author: Rytia
> Date: 20190427
> Blog: www.zzfly.net
> 本人才学疏浅，望君不吝赐教

背景

受学校老师邀请，为学弟学妹举办分享会介绍 AWD 相关经验，本人一时头脑风暴采用 PHP 的 `Swoole` 扩展搭建了比赛的环境，将分享会变成了友谊赛。

出题思路

本次题目来自于我的一个外包项目实践。项目里面大致有这么一个需求：客户登录系统后，由外部设备触发一个 websocket 发送操作（例如嵌入式中常遇见的“打开门禁”、“滴卡”、“按下开关”等），该请求接收方为某个已经登陆的某个用户（通常靠 user id 或用户名绑定）。本人在初次开发这类应用时，将用户唯一身份标识的 user id 作为了这个 websocket 通道的名字，如此一来带来的后果便是无论这个用户在哪台电脑登录，无论用户登陆多少次（小项目无重复登陆+挤下线判断），只要成功登录并打开相应网页便会收到这个 websocket 请求，带来了某些非预期的信息泄露。在实际的项目中解决方法有很多，例如重复登陆的判断以及 websocket “匿名通道”建立。本次出题，便以此为基础展开

AWD 运作原理

CTF-AWD 是线下赛中常见的比赛类型，通常因为攻守兼备而广受选手喜爱。这里我主要实现了 AWD 两大关键功能：回合制 、存活检测 与 动态 flag，至于 flag 提交等前端部分内容，则交由我校 GOCTF 平台处理。

- 回合制：比赛以 10 分钟为一回合。在中心服务器方面，事先使用脚本，根据已开启的靶机数量以及 ip 生成 flag 并保存为文件，并于比赛开始时记录开始时间。每次中心服务器接收到选手靶机的请求时，根据靶机的 ip，以及距离比赛开始的时间，计算出当前处于第几个回合，并返回 flag 数组中相应的值。而在 pusher （一个第三方 websocket 工具，可以为 cli 运行的脚本语言提供发送 websocket 的能力，本次出题依赖 pusher 进行），同样以 10 分钟为一回合，主动向选手端推送含有比赛关键内容的消息
- 存活检测：本次平台存货监控设置的比较简单，主要为 web 服务（靶机 80 端口）的监控。每隔一段时间向靶机发送 http 请求，下载靶机上某个保存着 pusher 密钥的 js 文件，若该文件大小符合预期，则判定靶机存活。选手在比赛过程中需要盗取到对手的密钥，以窃听他方 websocket 内容，并修改己方 pusher 密钥以放泄露
- 动态 flag：该部分同样由中心服务器与 pusher 完成。中心服务器在被请求时根据时间不同（回合不同）向选手返回不同 flag。pusher 根据时间不同（回合不同）主动推送不同的 flag 到选手的页面上。

AWD 缺陷总结

本次比赛完成之后，发现这套平台想要真正用于日常选手的训练还有几个问题需要克服

- 由于学校 GOCTF 平台截止比赛时仍未能很好的支持动态 flag 功能，因此根据每一回合中心服务器主动通知CTFF平台 flag 变更的功能无法实现，只能暂时让选手记录自己的 flag 233333….TAT
  时钟难以统一。因为 pusher 和 AWD 中心服务器不一定运行在同一台服务器上，且中心服务器为被- 动接收数据，pusher 为主动推送，加之启动时间先后有别，因此每个回合难以做到完全统一。往后如果需要再次制作 AWD 比赛平台可以加一个定时心跳包以保证时钟统一

Writeup

介绍完比赛平台的基本运作思路，下面简单讲解一下这道 AWD 题目的做法。

攻

访问 80 端口，注册多个用户，登陆系统后，发现 User ID 为1的用户已经注册，且系统提示一定要以 User ID 为 2 的用户登录
登录到 User ID 为 2 的用户后，发现页面启动了 websocket，隧道名称为 user.2。切换到其他用户后均发现隧道名称为 user.*（*为当前用户登录 id） 的 websocket 链接
每一回合，发现被主动以弹窗形式提醒了“比赛消息”，且“比赛消息”中包含 flag1，但是弹窗出现后2秒内强制跳转到了用户注销页面（暗示含有XSS）
登录服务器后台，修改视图文件(home.blade.php)，将底部 JS 部分中当前用户 ID 的输出 user.{{Auth::user()->id}} 修改为 user.1（表示强制接收 user.1 隧道的消息）
下一回合推送，接收到了 user.1 的消息，得到 flag2，且 flag2 以 = 结束，像 base64 编码，解码后得出 （保密） 账户的密码的 rot13 值，重新进行 rot13 旋转后得到正确密码
以 （保密） 管理员身份登录后，发现有头像上传的模块，且仅做了前端校验，burpsuit 修改请求中的文件名即可上传PHP“菜刀”，得到系统 shell，且用户为 root
了解整个过程后，发现 pusher.js 中含有 pusher 账户的密钥，且这个文件可以在80端口轻松下载，因此每台靶机都存在泄露问题。得到对手的 pusher 密钥后，修改到自己的服务器上即可接收到对手的 user.1 隧道中的消息，获取到对手的 （保密） 管理员账号密码，从而利用文件上传漏洞 get shell。

守

除了常规的备份、上监控、源码审计外，主要有以下几点解题思路。

主线

/root 目录下存在 readme.txt，提示从历史命令记录里寻找入口点
发现以下两条命令可疑：curl 172.17.0.1 和 php artisan tinker
使用 `ip addr 命令发现与前条命令处于同一网段，猜测是向中心服务器发起请求
进行 curl 测试发现无此命令，且 apt、yum 等工具均为无效，使用 cat /etc/issue 查看到当前系统为 alpine linux，因此使用 apk add curl 命令安装 curl 工具，正常请求后得到 flag3
根据比赛规则，flag3 为动态 flag，每回合变动一次，且不能提交自己靶机上的，因此可以按照这个思路去配合负责进攻部分的队友完成对方靶机的渗透
php artisan tinker 命令为赛题所用 Laravel 框架对 psy shell 的封装，可以直接进入 psy shell 通过 ORM 操作修改管理员账户的密码，从而获取自己靶机的管理员权限，与上文所述修改 user id 以窃听 websocket 的效果相似，但是难度比较大，也不太符合预期关于 websocket 的考点

副线

每个回合接收到弹窗，都会被强制跳转到“注销”页面，推测有 XSS 漏洞，进行源码审计后，将 resources/views/home.blade.php 文件中 $(“”).html(data.message) 修改为 $(“”).text(data.message) 即可不解析推送内容以放推送 XSS 投毒（仰天大笑~）
源码审计发现存在 upload() 方法用于处理 /upload 这个 url 下的文件上传操作，且不对扩展名、mine type 等进行判断， 可以给对手的服务器尝试强制上传文件。但是 Laravel 框架默认开启 csrf 拦截，需要在对手服务器上注册一个账号以获取 csrf token

Github 项目地址

https://github.com/zzfly256/CTF-AWD-demo

运行环境

AWD 平台部分：PHP 7 + Swoole 4
靶机部分：任意版本 docker

项目文件介绍

- server：AWD 中心服务器，运行于 docker 母机，负责根据提供每个回合的 flag：flag3
  - getFlag 文件为 php 可执行文件，根据 `docker ps` 命令中的启动的容器的 ip，为容器生成不同的 flag 并保存为 `flags.json`
  - server 文件为 php 可执行文件，监听 `80` 端口，根据不同回合返回 `flags.json` 文件中的相应值
  - flags.json 保存生成的 flag
- pusher：`Pusher.js` 的服务端，运行于 `docker` 母鸡或任何一台电脑。关于 pusher 的介绍可移步官网：www.pusher.com
  - pusher-admin 文件为 php 可执行文件，为赛题中管理员用户推送消息（供选手窃听 `websocket`，包含一个 flag ：flag2）
  - pusher-user 文件为 php 可执行文件，为赛题中(ID为 2)的普通注册用户推送消息（包含一个 flag ： flag1）
  - monitor 文件为 php 可执行文件，用于监控选手靶机是否存活（监控 web 服务/ `pusher.js` 文件大小）
  - pusher-server 文件为 php 可执行文件，按照回合（时间）推送消息以及存活检测
  - getFlag 文件为 php 可执行文件，生成管理员消息（flag2）所用
  - pusher-key.json 保存各个靶机的 `pusher` 密钥，以及 flag2 的值（flag1的值为固定值，每个选手一样）
- web：比赛赛题企业网站部分， 是为 `laravel 5.8` 框架，采用 `sqlite` 数据库
  - 业务逻辑主要在 `/app/Http/Controllers/HomeController.php`
  - 视图文件在 `/resources/views` 目录下
- docker：靶机 docker 镜像

更多

更多关于本项目的介绍，可以移步：http://www.zzfly.net/build-a-ctf-awd-platform-by-swoole/