WebCrack：網站後台弱口令批量檢測工具

Web開發 2022-09-09 10:32 發表於福建

> 來自：先知社區
> 鏈接：https://xz.aliyun.com/t/63

在做安全測試的時候，隨著資產的增多，經常會遇到需要快速檢測大量網站後台弱口令的問題。

然而市面上並沒有一個比較好的解決方案，能夠支持對各種網站後台的通用檢測。

所以WebCrack就應運而生。

工具簡介

WebCrack是一款web後台弱口令/萬能密碼批量爆破、檢測工具。

不僅支持如discuz，織夢，phpmyadmin等主流CMS

並且對於絕大多數小眾CMS甚至個人開發網站後台都有效果。

在工具中導入後台地址即可進行自動化檢測。

項目地址

https://github.com/yzddmr6/WebCrack

實現思路

大家想一下自己平常是怎麼用burpsuite的intruder模塊來爆破指定目標後台的

```
抓包 -> send to intruder -> 标注出要爆破的参数 -> 发送payload爆破 -> 查看返回结果
```

找出返回包長度大小不同的那一個，基本上就是所需要的答案。

那麼WebCrack就是模擬這個過程

但是就要解決兩個問題

- 如何自動識別出要爆破的參數
- 如何自動判斷是否登錄成功

識別爆破參數

對於這個問題採用了web_pwd_common_crack的解決辦法

就是根據提取表單中user pass 等關鍵字，來判斷用戶名跟密碼參數的位置
但是在測試中還發現，

有些前端程序員用拼音甚至拼音縮寫來給變量命名

什麼yonghu，zhanghao，yhm(用戶名), mima 等

雖然看起來很捉急，但是也只能把它們全部加進關鍵字判斷名單裡。

如何判斷登錄成功

這個可以說是最頭疼的問題

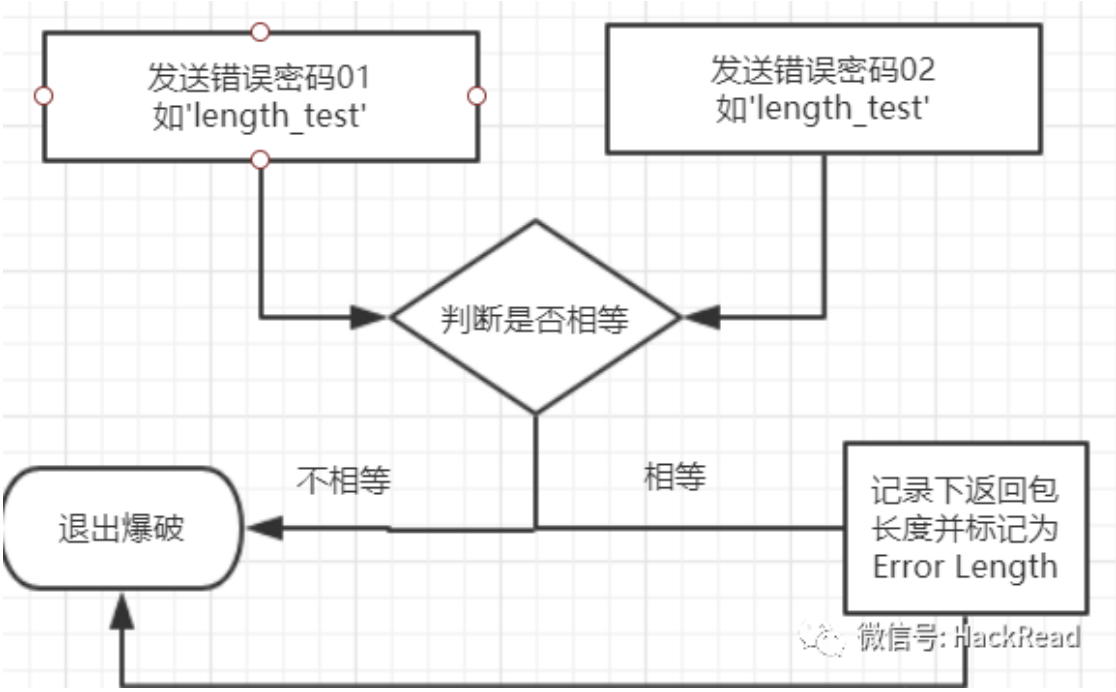
如果對於一種管理系統還好說，只要找到規律，判斷是否存在登錄成功的特徵就可以

但是作為通用爆破腳本來說，世界上的網站各種各樣，不可能去一個個找特徵，也不可能一個個去正則匹配。

經過借鑒web_pwd_common_crack的思路，與大量測試

總結出來了以下一套比較有效的判斷方式。

判斷是否動態返回值並獲取Error Length



先發送兩次肯定錯誤的密碼如`length_test`

獲取兩次返回值並比較

如果兩次的值不同，則說明此管理系統面對相同的數據包返回卻返回不同的長度，此時腳本無法判斷，退出爆破。

如果相同，則記錄下此值，作為判斷的基準。

然而實際中會先請求一次，因為發現有些管理系統在第一次登錄時會在響應頭部增加標記。如果去掉此項可能會導致判斷失誤。

判斷用戶名跟密碼的鍵名是否存在在跳轉後的頁面中

這個不用過多解釋，如果存在的話說明沒登錄成功又退回到登錄頁面了。

有人會問為什麼不直接判斷兩個頁面是否相等呢

因為測試中發現有些CMS會給你在登錄頁面彈個登錄失敗的框，所以直接判斷是否相等並不准確。

還有一種計算頁面哈希的辦法，然後判斷兩者的相似程度。

但是覺得併沒有那個必要，因為有不同的系統難以用統一的閾值來判斷，故捨棄。

關鍵字黑名單檢測

本來還設置了白名單檢測機制

就是如果有“登錄成功”的字樣出現肯定就是爆破成功

但是後來發現並沒有黑名單來的必要。

因為首先不可能把所有CMS的登錄成功的正則樣本都放進去

其次在測試的過程中，發現在其他檢測機制的加持後，白名單的判斷變得尤其雞肋，故捨棄。

並且黑名單的設置對於萬能密碼爆破模塊很有好處，具體往下看吧。

Recheck環節

為了提高準確度，防止誤報。

借鑒了web_pwd_common_crack的思路增加recheck環節。

就是再次把crack出的賬號密碼給發包一次，並且與重新發送的error_length作比對
如果不同則為正確密碼。

在這裡沒有沿用上一個error_length，是因為在實際測試中發現由於waf或者其他因素會導致返回包長度值變化。

框架拓展

用上面幾種辦法組合起來已經可以做到基本的判斷算法了

但是為了使WebCrack更加強大，我又添加了以下三個模塊

動態字典

這個不用過多解釋，很多爆破工具上都已經集成了。

假如沒有域名，正則檢測到遇到IP的話就會返回一個空列表。

假如域名是
`test.webcrack.com`

那麼就會生成以下動態字典列表

```
test.webcrack.com
webcrack.com
webcrack
webcrack123
webcrack888
```

```
webcrack666
webcrack123456
```

後綴可以自己在腳本中定義。

萬能密碼檢測

後台的漏洞除了弱口令還有一大部分是出在萬能密碼上

在WebCrack中也添加了一些常用的payload

```
admin' or 'a'='a
'or'='or'
admin' or '1'='1' or 1=1
')or('a'='a
'or 1=1--
```

可以自行在腳本里添加更多payload。

但是同時帶來個問題會被各大WAF攔截

這時候黑名單就派上用場啦

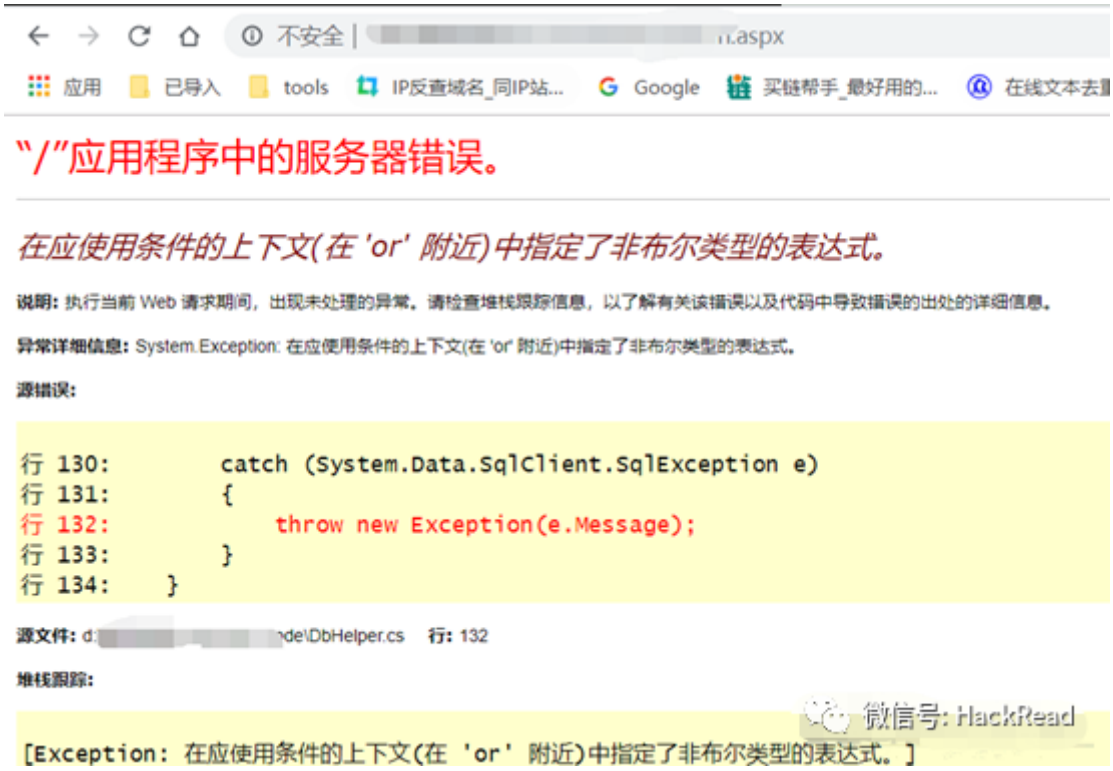
可以把WAF攔截的關鍵字寫到檢測黑名單裡，從而大大減少誤報。

小插曲

用webcrack檢測目標資產進入到了recheck環節

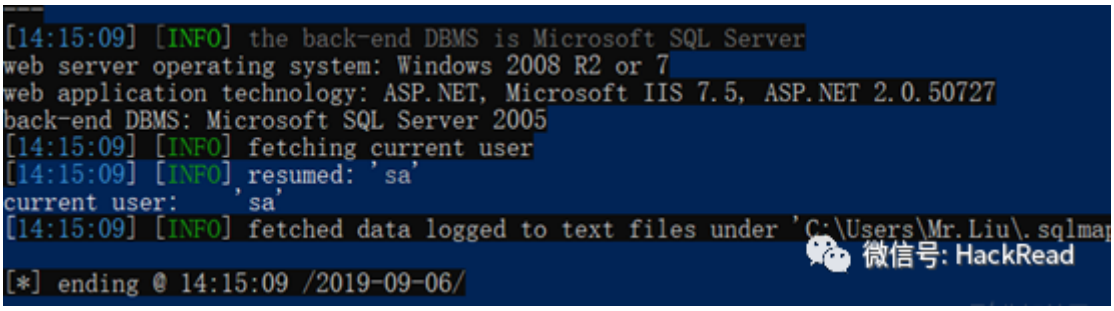
但是webcrack卻提示爆破失敗。

手工測試了一下檢測出的萬能密碼



發現出現了sql錯誤信息

意識到可能存在後台post注入



發現了sa注入點

這也反應了對於後台sql注入，webcrack的正則匹配還做的不夠完善，下一個版本改一下。

自定義爆破規則

有了上面這些機制已經可以爆破大部分網站後台了

然而還是有一些特(sha)殊(diao)網站，並不符合上面的一套檢測算法

於是webcrack就可以讓大家自定義爆破規則。

自定義規則的配置文件放在同目錄cms.json文件裡

參數說明

```
[
    {
        "name":"这里是cms名称",
        "keywords":"这里是cms后台页面的关键字,是识别cms的关键",
        "captcha":"1为后台有验证码，0为没有。因为此版本并没有处理验证码，所以为1则退出爆破",
        "exp_able":"是否启用万能密码模块爆破",
        "success_flag":"登录成功后的页面的关键字",
        "fail_flag":"请谨慎填写此项。如果填写此项，遇到里面的关键字就会退出爆破，用于dz等对爆破次数有限制的cms",
        "alert":"若为1则会打印下面note的内容",
        "note":"请保证本文件是UTF-8格式，并且请勿删除此说明"
    }
]
```

舉個例子

```
{
        "name":"discuz",
        "keywords":"admin_questionid",
        "captcha":0,
        "exp_able":0,
        "success_flag":"admin.php?action=logout",
        "fail_flag":"密码错误次数过多",
        "alert":0,
        "note":"discuz论坛测试"
    }
```

其實對於dz,dedecms,phpmyadmin等框架本身的邏輯已經可以處理

添加配置文件只是因為程序默認會開啟萬能密碼爆破模塊

然而萬能密碼檢測會引起大多數WAF封你的IP

對於dz，dedecms這種不存在萬能密碼的管理系統如果開啟的話不僅會影響效率，並且會被封IP

所以配置文件裡提供了各種自定義參數，方便用戶自己設置。

關於驗證碼

驗證碼識別算是個大難題吧

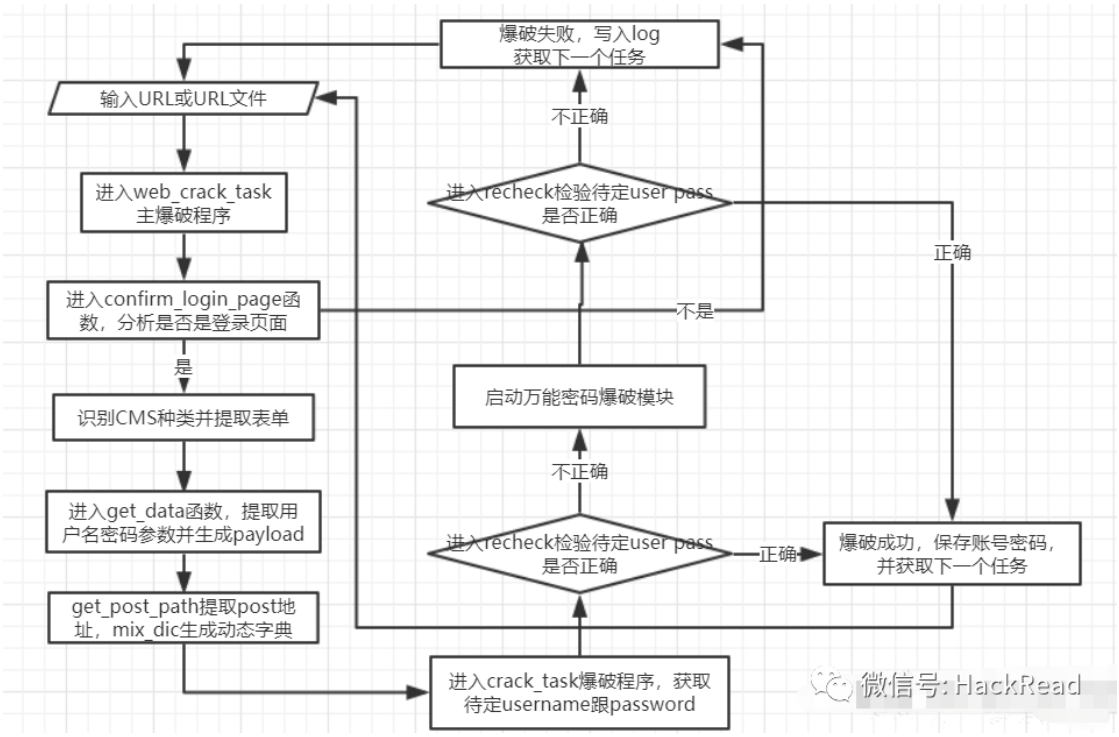
自己也寫過一個帶有驗證碼的demo，但是效果並不理想

簡單的驗證碼雖然能夠識別一些，但是遇到復雜的驗證碼就效率極低，拖慢爆破速度
並且你識別出來也不一定就有弱口令。。。

所以就去掉了這個功能

總流程圖

一套流程下來大概是長這個亞子



對比測試
找了一批樣本測試，跟tidesec的版本比較了一下

- web_pwd_common_crack 跑出來11個

  > 其中7個可以登錄。4個是邏輯上的誤報，跟waf攔截後的誤報。

- webcrack 跑出來19個

  > 其中16個可以登錄。2個是ecshop的誤報，1個是小眾cms邏輯的誤報。

- webcrack比web_pwd_common_crack多探測出來的9個中

> 有5個是萬能密碼漏洞，2個是發現的web_pwd_common_crack的漏報，2個是動態字典探測出來的弱口令。

最後

這個項目斷斷續續寫了半年吧

主要是世界上奇奇怪怪的網站太多了，後台登錄的樣式五花八門。

有些是登錄後給你重定向302到後台

有些是給你重定向到登錄失敗頁面

有些是給你返回個登錄成功，然後你要手動去點跳轉後台

有些直接返回空數據包。。。

更神奇的是ecshop(不知道是不是所有版本都是這樣)

假如說密碼是yzddmr6

但是你輸入admin888 與其他錯誤密碼後的返回頁面居然不一樣。。。

因為加入了萬能密碼模塊後經常有WAF攔截，需要測試各個WAF對各個系統的攔截特徵以及關鍵字。

總的半年下來抓包抓了上萬個都有了。。。。。。

因為通用型爆破，可能無法做到百分百準確，可以自己修改配置文件來讓webcrack更符合你的需求。

--- EOF ---

閱讀原文

喜歡此內容的人還喜歡

《ASP.NET Core 6框架揭秘》實例演示[22]：如何承載你的後台服務[補充]

大內老A

從Multirepo到Monorepo 袋鼠雲數棧前端研發效率提升探索之路

數棧研習社

超全面的前端工程化配置指南！

前端下午茶