

BurpSuite插件 - Hackvortor 编码解码的应用(payload编码 XOR破解)

[arr0w1](#) / 2019-01-18 17:26:00 / 浏览数 3436 [安全工具](#) [工具](#) [顶\(1\)](#) [踩\(0\)](#)

翻译自 PortSwigger Web Security Blog - [Bypassing WAFs and cracking XOR with Hackvortor | Blog](#)

原作者：Gareth Heyes [u2028u2029 \(@garethheyes\) | Twitter](#)

简介

BurpSuite插件[Hackvortor](#)是个基于标签的转换工具。

您可以通过Burp Extender工具中的BApp Store直接安装。

特点

- 支持各种转义和编码，包括：HTML5实体（HTML5 entities），十六进制，八进制，unicode，url编码等。
- 使用类似XML的标签来指定使用的编码/转换类型。
- 可以使用嵌套多个标签来进行编码转换。
- 标签也可以有参数，像函数一样运行。
- 它具有自动解码(auto decode)功能，可以猜测所需的转换类型并自动执行多次解码，返回最终结果。
- 可开启多个tab窗口(像repeater一样可开启多个tab)
- 字符集转换

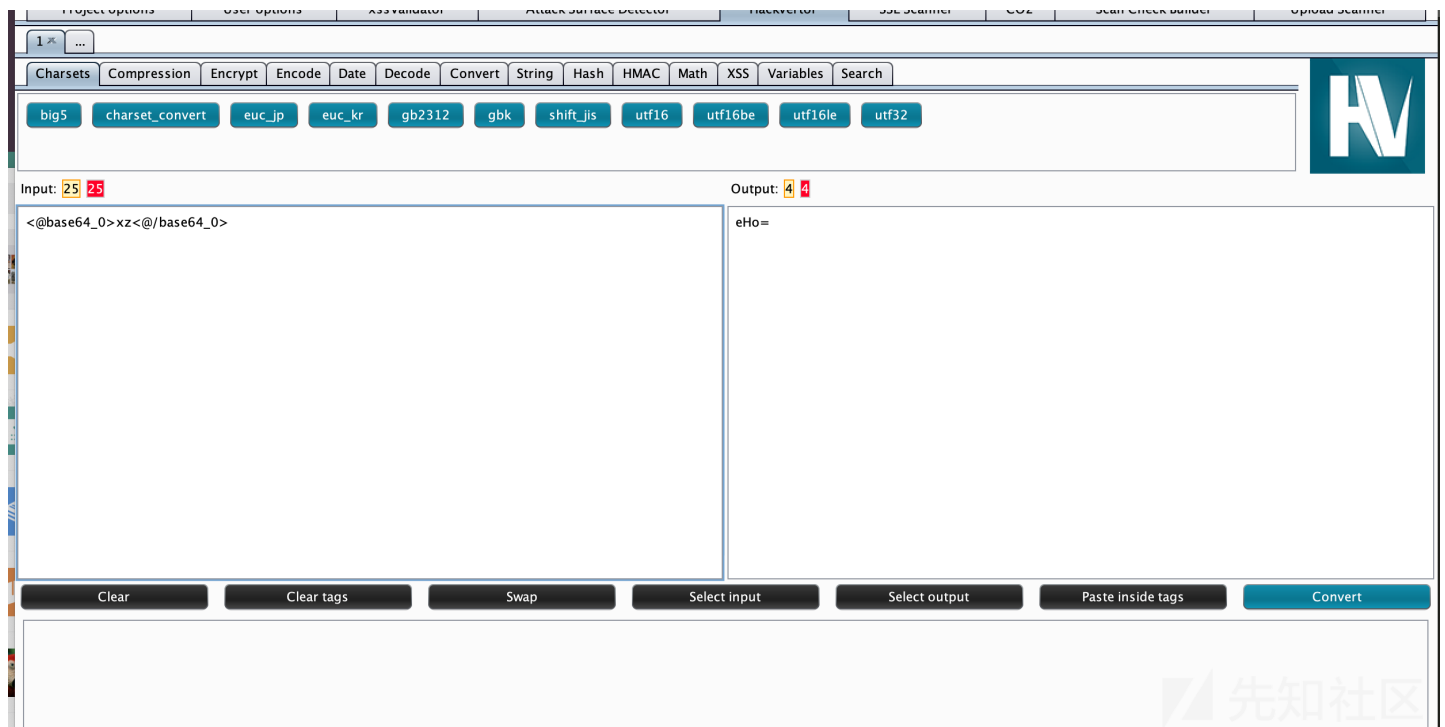
基本功能 - 编码转换

我最近一直努力开发的Hackvortor，具有基于标签的转换功能，这样的设计比Burp的内置的Decoder功能强大得多。

因为基于标记的转换，可以实现多层转换：内部的标签先完成第一次转换，并将结果作为输入，交给外部标签做第二次转换，以此类推。

例1 进行base64编码：

`<@base64_0>xz<@/base64_0>`

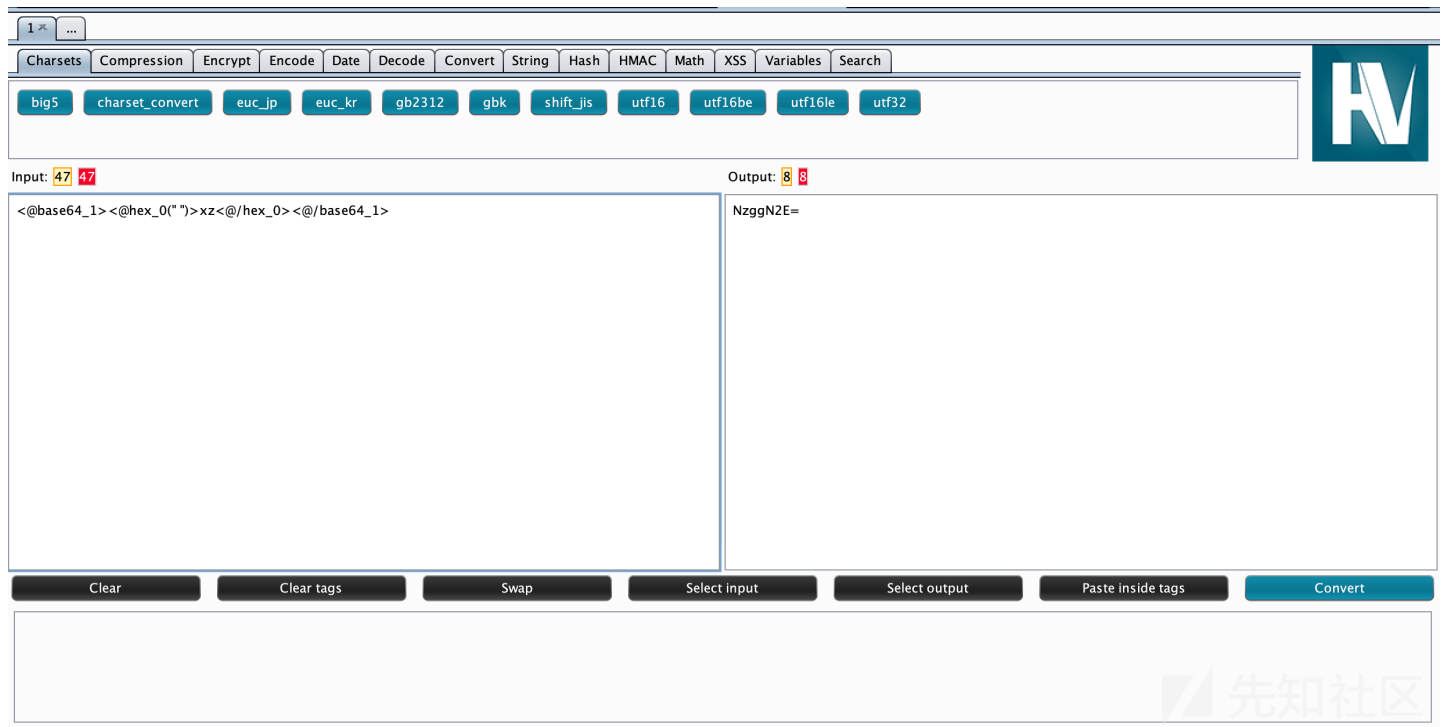


例2 进行多级编码(multiple levels of encoding)：

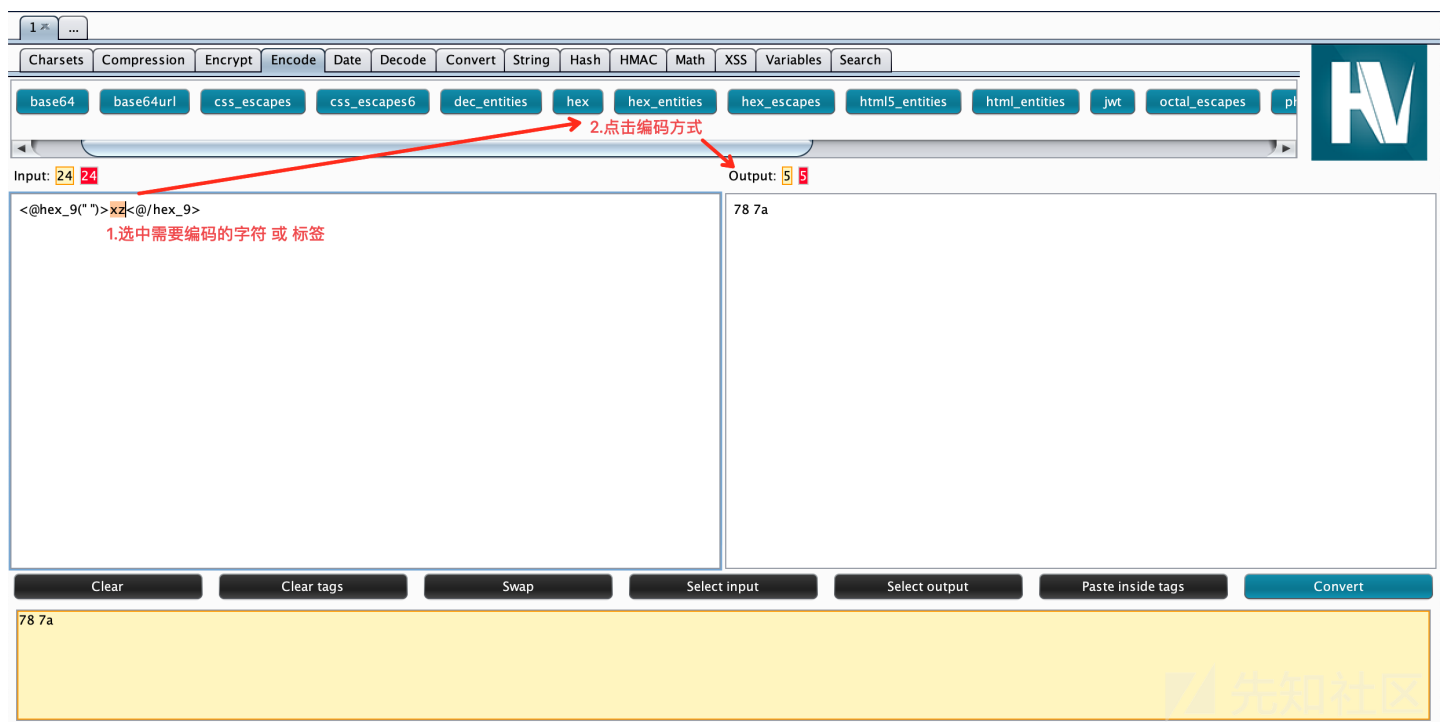
`<@base64_1><@hex_0(" ")>xz<@/hex_0><@/base64_1>`

先使用hex标签将字符串转换为十六进制，然后使用base64标签对其进行base64编码。

注意hex标签有一个分隔符参数，此处使用空格分隔每个十六进制字符串(hex string)。



例3 选中文本后快捷操作：
先选中需要编码的字符，再点击编码方式如hex



应用 - 编码payload

1.在Repeater/Intruder启动功能

Hackvector

Allow tags in Proxy

- ✓ Allow tags in Intruder
- ✓ Allow tags in Repeater
- ✓ Allow tags in Scanner
- ✓ Auto update content length

2.在Repeater中

如自己写的XSSpayload为<img/src/onerror=alert(1)>

选中alert(1)并进行如图设置：

右键Hackvector - XSS - throw_eval

在Burp中看到请求中的payload变为<img/src/onerror=<@throw_eval_1>alert(1)<@/throw_eval_1>>

此时发出的请求中的payload是经过编码的。

The screenshot shows the Burp Suite interface with the Hackvector context menu open. The menu options include 'Send to Hackvector', 'Copy URL', 'Convert tags', 'Auto decode & Convert', 'Charsets', 'Compression', 'Encrypt', 'Encode', 'Date', 'Decode', 'Convert', 'String', 'Hash', 'HMAC', 'Math', 'XSS', and 'Variables'. The 'XSS' option is selected, and a sub-menu is visible showing 'throw_eval' and 'uppercase_script'. The 'throw_eval' option is highlighted, and the resulting payload 'throw_eval(String str)' is shown in a yellow box.

The 'Request' tab on the left shows the raw request: `GET /<script>alert(1)</script> HTTP/1.1`. The 'Response' tab on the right shows the raw response: `HTTP/1.1 200 OK`.

可从服务器端看到实际发送的请求。

GET /<img/src/onerror=window.onerror=eval;throw'=alert\x281\x29'> HTTP/1.1

可看到payload为

```
<img/src/onerror=window.onerror=eval;throw'=alert\x281\x29'>
```

原本的alert(1)已经被编码成了alert\x281\x29

更方便的办法是使用Hackvector的 [Copy](#) [URL](#)等3个按钮都可看到。

Scan Send to Intruder ⌘+^+I Send to Repeater ⌘+^+R Send to Sequencer Send to Comparer Send to Decoder Show response in browser Request in browser ▶		<h2>Response</h2> <div> <div>Raw Headers Hex</div> <div> HTTP/1.1 404 Not Found Date: Fri, 18 Jan 2018 12:12:12 GMT Server: nginx Content-Type: text/html X-Via: 1.1 PShnlDD PSbjzwdx5io12:1 (C) Connection: close Content-Length: 16 </div> </div> <pre> <html> <head><title>404 Not Found</title> <body bgcolor="white"> </pre>
^-^ Copy this cookie ^-^ Get latest cookie ^-^ Update cookie ^-^ Add host to scope ^-^ U2C(unicode to chinese)		
Hackvertor ▶		<div> <div>Send to Hackvertor</div> <div> <div>Copy URL</div> <div>Convert tags</div> <div>Auto decode & Convert</div> </div> <div> <div>Charsets ▶</div> <div>Compression ▶</div> <div>Encrypt ▶</div> <div>Encode ▶</div> <div>Date ▶</div> <div>Decode ▶</div> <div>Convert ▶</div> <div>String ▶</div> <div>Hash ▶</div> <div>HMAC ▶</div> <div>Math ▶</div> <div>XSS ▶</div> <div>Variables ▶</div> </div> </div>
Send URL to SSL Scanner Hack Bar ▶ Send to SQLMapper Send to CeWler Send to Laudanum Send to Upload Scanner		
Engagement tools ▶		
Change request method Change body encoding Copy URL Copy as curl command Copy to file Paste from file Save item		
Save entire history Paste URL as request Add to site map		
Convert selection ▶ URL-encode as you type		
Cut ⌘+^+X Copy ⌘+^+C Paste ⌘+^+V		
Message editor documentation Burp Repeater documentation		

[RSS](#) [关于社区](#) [友情链接](#) [社区小黑板](#)