THM_S3BOTS2 [200 series questions]

In this task, we'll attempt to tackle the 200 series questions from the BOTSv2 dataset.

Note: As noted in the previous task, this guide is not the only way to query Splunk for the answers to the questions below.

Question 1

What version of TOR Browser did Amber install to obfuscate her web browsing? Answer guidance: Numeric with one or more delimiter.

torbrowser-install-7.0.4_en-US.exe

Our first task is to identify the version of Tor that Amber installed. You can use a keyword search to get you started.

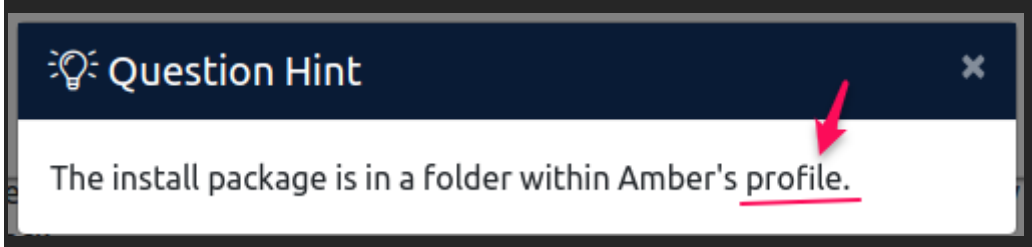
What are some good keywords? Definitely **Amber**. Another would be **Tor**. Give that a go.

Command: index="botsv2" amber tor
index="botsv2" amber Tor
Over 300 results are returned. You can reverse the order of results (hoping the 1st event is the TOR installation) and see if you can get the answer.

You should add another keyword to this search query. I'll leave that task to you.

Command: index="botsv2" amber tor ***KEYWORD***

Replace the **KEYWORD**with another search term to help narrow down the events to the answer.
Using the hint



index="botsv2" amber tor profile
| reverse

3rd entry

8/24/17
4:20:34.000 AM

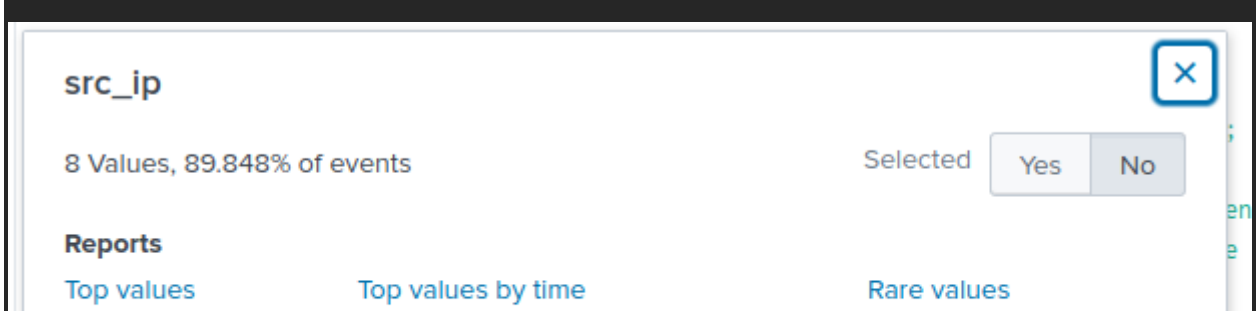

```
<Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><
System><Provider Name='Microsoft-Windows-Sysmon' Guid='{5770385F-C22A-
43E0-BF4C-06F5698FFBD9}'/><EventID>2</EventID><Version>4</Version><Lev
el>4</Level><Task>2</Task><Opcode>0</Opcode><Keywords>0x80000000000000
00</Keywords><TimeCreated SystemTime='2017-08-24T04:20:34.396973800Z'/
><EventRecordID>118470</EventRecordID><Correlation/><Execution Process
ID='900' ThreadID='1824'/><Channel>Microsoft-Windows-Sysmon/Operationa
l</Channel><Computer>wrk-aturing.frothly.local</Computer><Security Use
rID='S-1-5-18'/></System><EventData><Data Name='UtcTime'>2017-08-24 0
4:20:34.396</Data><Data Name='ProcessGuid'>{B2E0DF5E-9CDF-598C-0000-00
101AF3CB01}</Data><Data Name='ProcessId'>4536</Data><Data Name='Imag
e'>C:\Users\amber.turing\Downloads\torbrowser-install-7.0.4_en-US.exe
</Data><Data Name='TargetFilename'>C:\Users\amber.turing\Desktop\Tor B
rowser\Browser\TorBrowser\Data\Browser\profile.default\extensions\tor-
launcher@torproject.org.xpi</Data><Data Name='CreationUtcTime'>2000-01
-14 10:30:00.000</Data><Data Name='PreviousCreationUtcTime'>2017-08-24
04:20:34.365</Data></EventData></Event>
```


Questions 2 & 3

What is the public IPv4 address of the server running www.brewertalk.com?

52.40.10.231

index="botsv2" brewertalk sourcetype="stream:http"



Events with this field

Values	Count	%
45.77.65.211	8,966	90.047%
52.40.10.231	317	3.184%
172.31.10.10	303	3.043%
71.39.18.125	133	1.336%
174.209.13.154	125	1.255%
10.0.2.109	90	0.904%
136.0.2.138	17	0.171%
136.0.0.125	6	0.06%

Oops!
Let's see the hint

Question Hint

Public IP, not private.

dest_ip

5 Values, 89.875% of events

Selected Yes No

Reports

Top values

Events with this field

I'm not sure why it isn't the first entry; guess I need to learn more^*

Values	Count	%
172.31.4.249	9,562	96.004%
52.42.208.228	387	3.886%
172.31.10.10	5	0.05%
52.40.10.231	5	0.05%
45.77.65.211	1	0.01%

Provide the IP address of the system used to run a web vulnerability scan against www.brewertalk.com.

45.77.65.211

You need to determine the publicIP address for brewertalk.com and the IP address performing a web vulnerability scan against it.

You should be able to tackle this one on your own. Use the previous search queries as your guide.

Hmmm. I did something like this in the first Splunk room. I'll try to remember.
Lets start with

index="botsv2" brewertalk.com sourcetype="stream:http"

http_user_agent

10 Values, 89.74% of events

Selected Yes No

Reports

Top values Top values by time Rare values

Events with this field

Top 10 Values	Count	%
Mozilla/4.0 (compatible; MSIE 8.0; Windows NT	8,811	88.597%

6.1; Trident/4.0; w3af.org)		
Splunk Website Monitoring (+https://splunkbase.splunk.com/app/1493/)	620	6.234%
Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:55.0) Gecko/20100101 Firefox/55.0	145	1.458%
Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/30.0.1599.17 Safari/537.36	136	1.368%
Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1)	126	1.267%
Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.90 Safari/537.36	78	0.784%
Mozilla/5.0 (X11; U; Linux i686; ko-KP; rv: 19.1br) Gecko/20130508 Fedora/1.9.1-2.5.rs3.0 NaenaraBrowser/3.5b4	23	0.231%
() { :;}; echo "shellshock: check"	2	0.02%
() { test; }; ping -c 13 127.0.0.1	2	0.02%
() { test; }; sleep 12	2	0.02%

This looks very fishy!!!

index="botsv2" brewertalk.com sourcetype="stream:http" http_user_agent="() { :;}; echo \"shellshock: check\""

status: 200

src_ip

1 Value, 100% of events

Selected Yes No

Reports

Top values Top values by time Rare values

Events with this field

BINGO!

Values	Count	%
45.77.65.211	2	100%

Even though I got the right answer, I wonder if I did this correctly? Let me explore a bit

site

4 Values, 89.848% of events

Selected Yes No

Reports

Top values Top values by time Rare values

Events with this field

The location of the server?

Values	Count	%
www.brewertalk.com	9,860	99.026%
brewertalk.com	86	0.864%
ec2-52-40-10-231.us-west-2.compute.amazonaws.com:8088	10	0.1%
45.77.65.211:9999	1	0.01%

index="botsv2" brewertalk.com sourcetype="stream:http" status=404

src_ip

3 Values, 100% of events

Selected Yes No

Reports

more corroboration

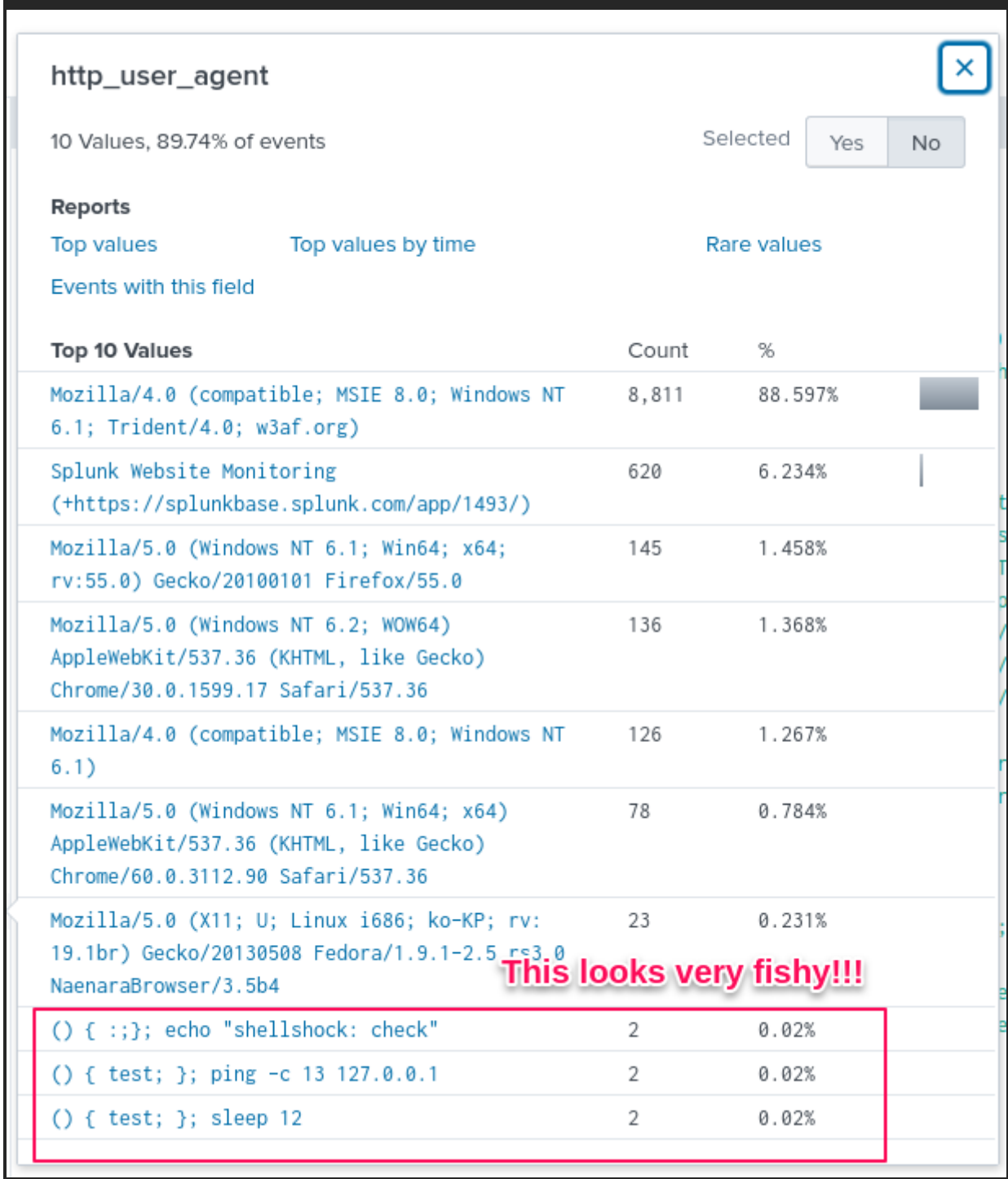
Top values Top values by time Rare values

Events with this field

Values	Count	%
45.77.65.211	8,050	99.888%

71.39.18.125	6	0.074%
10.0.2.109	3	0.037%

It seems looking up src_headers, in this case 100+, and user agents is how to do it. How to limit the src_headers properly is something i can't figure out.



Top 10 Values	Count	%
Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0; w3af.org)	8,811	88.597%
Splunk Website Monitoring (+https://splunkbase.splunk.com/app/1493/)	620	6.234%
Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:55.0) Gecko/20100101 Firefox/55.0	145	1.458%
Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/30.0.1599.17 Safari/537.36	136	1.368%
Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1)	126	1.267%
Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.90 Safari/537.36	78	0.784%
Mozilla/5.0 (X11; U; Linux i686; ko-KP; rv:19.1br) Gecko/20130508 Fedora/1.9.1-2.5.rs3.0 NaenaraBrowser/3.5b4	23	0.231%
() { :;}; echo "shellshock: check"	2	0.02%
() { test; }; ping -c 13 127.0.0.1	2	0.02%
() { test; }; sleep 12	2	0.02%

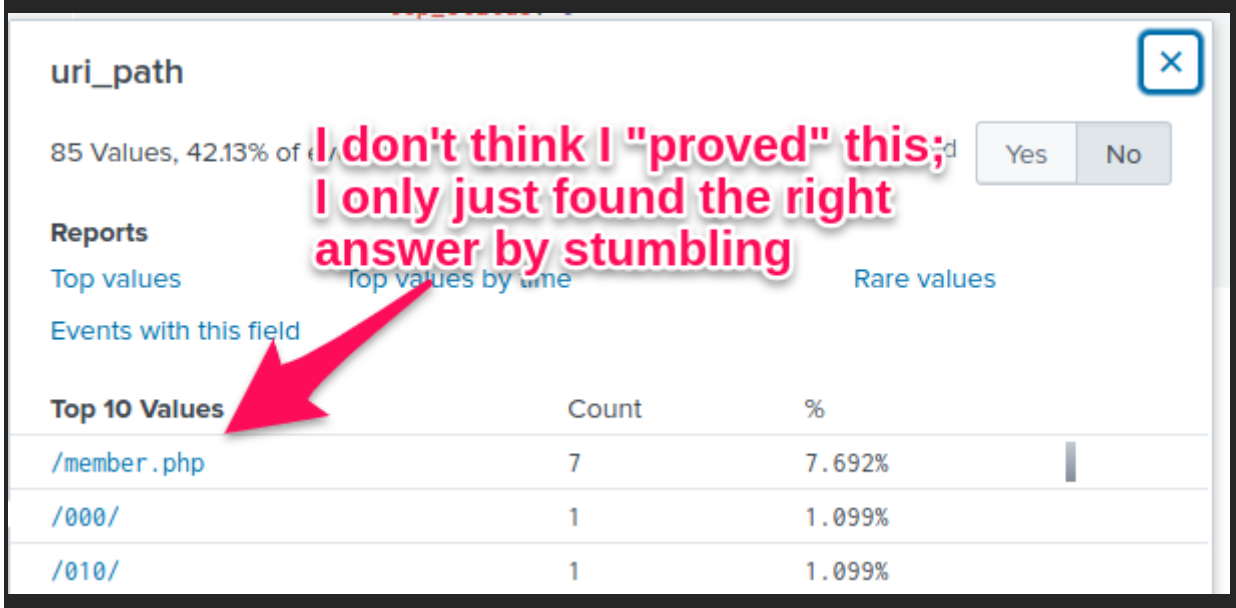
I do know that shellshock is a web server vulnerability and the sleep command is someone trying to figure out the SQL database type.

Questions 4 & 5

The IP address from Q#2 is also being used by a likely different piece of software to attack a URI path. What is the URI path? Answer guidance: Include the leading forward slash in your answer. Do not include the query string or other parts of the URI. Answer example: /phpinfo.php

/member.php

index="botsv2" src_ip=45.77.65.211



Top 10 Values	Count	%
/member.php	7	7.692%
/000/	1	1.099%
/010/	1	1.099%

Now that you have the attacker IP address, build your new search query with the attacker IP as the **source IP**.

Command: index="botsv2" src_ip="***ATTACKER_IP***"

Tip: Change the**Sampling**to **1:100**or your query will auto-cancel and throw errors.

Yikes! The number of events returned is over 18,000 .. but that is fine.

Use the **Interesting Fields**to help you identify what the URI path that is being attacked is.

Once the URI path has been identified, you can use it to expand the search query further to determine what SQL function is being abused.

Command: index="botsv2" src_ip="***ATTACKER_IP***" uri_path="***URI_PATH***"

You should have over 600 events to sift through but fret not; the answer is there.

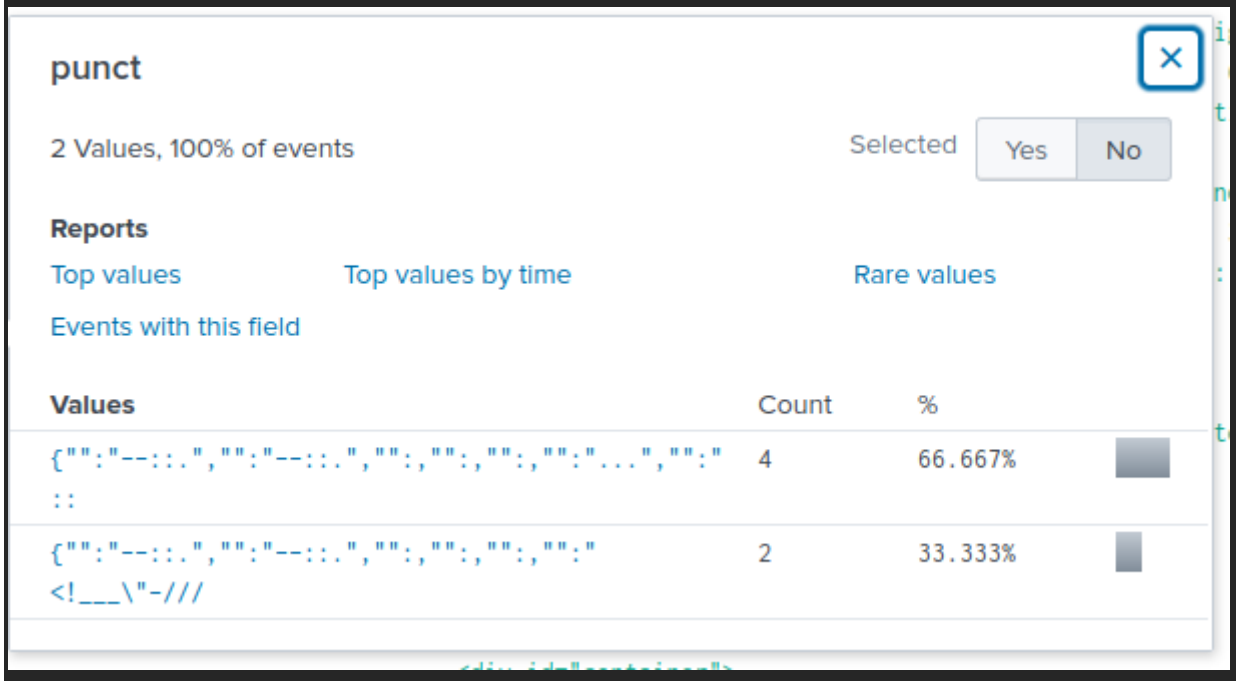
What SQL function is being abused on the URI path from the previous question?

{"":"--::.","":"--::.","":,"":,"":,"":"...","":"::

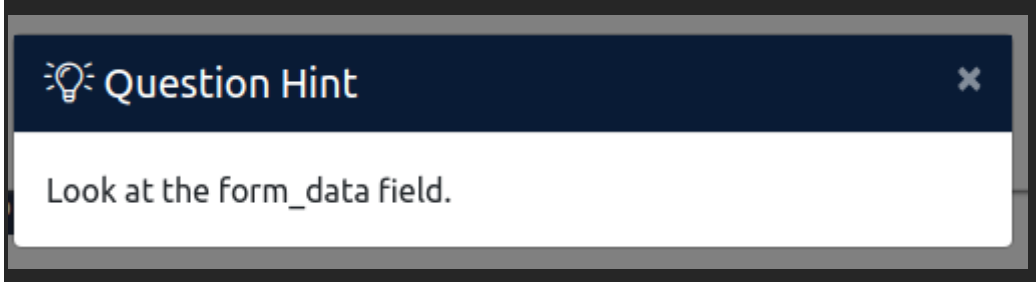
{"":"--::.","":"--::.","":,"":,"":,"":"<!___\"-///

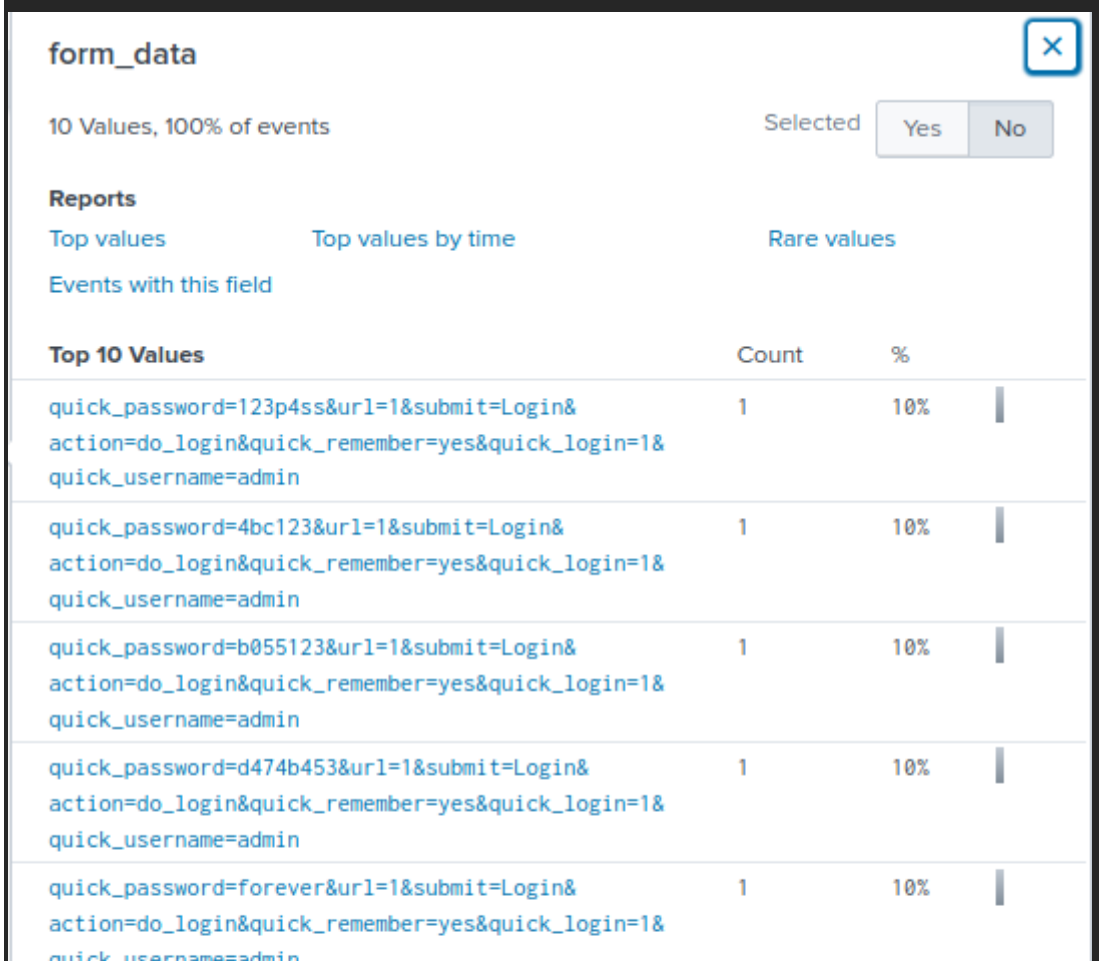
index="botsv2" src_ip=45.77.65.211 uri_path="/member.php"

```
index="botsv2" sourcetype="stream:http" "45.77.65.211" url=*
```



Nope the wrong track



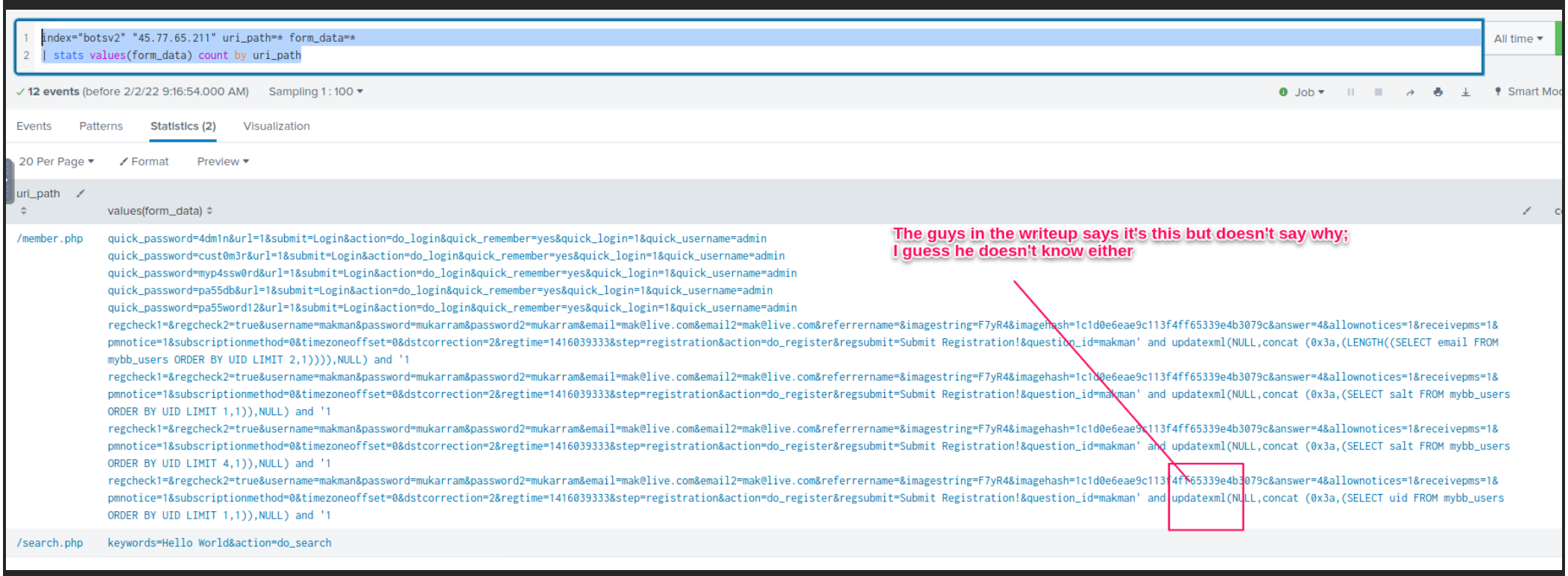


quick_password=internet&url=1&submit=Login&action=do_login&quick_remember=yes&quick_login=1&quick_username=admin	1	10%
quick_password=l0g1n&url=1&submit=Login&action=do_login&quick_remember=yes&quick_login=1&quick_username=admin	1	10%
quick_password=passwd&url=1&submit=Login&action=do_login&quick_remember=yes&quick_login=1&quick_username=admin	1	10%

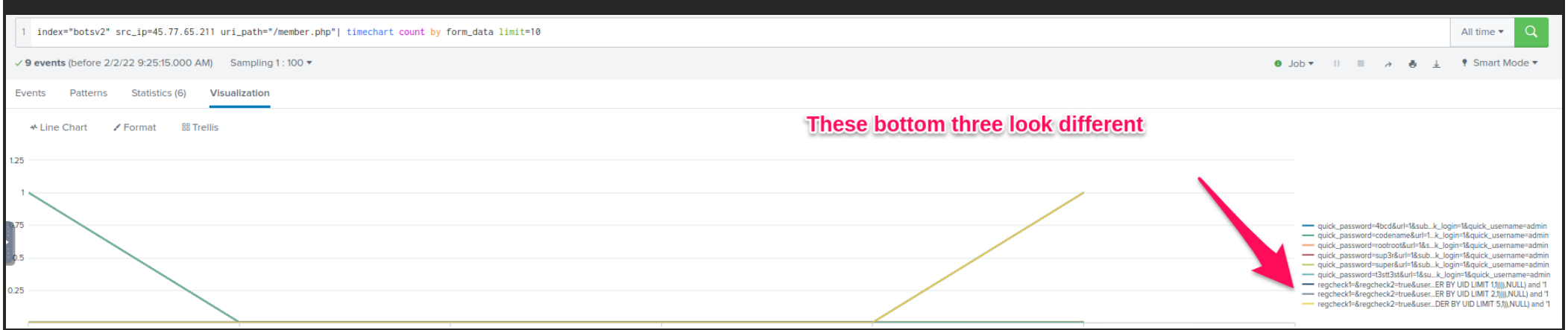
regcheck1=&regcheck2=true&username=makman&password=mukarram&password2=mukarram&email=mak@live.com&email2=mak@live.com&referrername=&imagestring=F7yR4&imagehash=1c1d0e6eae9c113f4ff65339e4b3079c&answer=4&allownotices=1&receivepms=1&pmnotice=1&subscriptionmethod=0&timezoneoffset=0&dstcorrection=2&regtime=1416039333&step=registration&action=do_register&regsubmit=Submit Registration!&question_id=makman' and updatexml(NULL,concat (0x3a,(LENGTH((SELECT salt FROM mybb_users ORDER BY UID LIMIT 1,1)))),NULL) and '1	1	10%
regcheck1=&regcheck2=true&username=makman&password=mukarram&password2=mukarram&email=mak@live.com&email2=mak@live.com&referrername=&imagestring=F7yR4&imagehash=1c1d0e6eae9c113f4ff65339e4b3079c&answer=4&allownotices=1&receivepms=1&pmnotice=1&subscriptionmethod=0&timezoneoffset=0&dstcorrection=2&regtime=1416039333&step=registration&action=do_register&regsubmit=Submit Registration!&question_id=makman' and updatexml(NULL,concat (0x3a,(LENGTH((SELECT uid FROM mybb_users ORDER BY UID LIMIT 3,1)))),NULL) and '1	1	10%

Hmm. I needed help so I found a walkthrough

index="botsv2" "45.77.65.211" uri_path=* form_data=*
| stats values(form_data) count by uri_path



Going back to form data, I did some digging




```
index="botsv2" src_ip=45.77.65.211 uri_path="/member.php" form_data="regcheck1=&regcheck2=true&username=makman&password=mukarram&password2=mukarram&email=mak@live.com&email2=mak@live.com&referrername=&imagestring=F7yR4&imagehash=1c1d0e6eae9c113f4ff65339e4b3079c&answer
    =4&allownotices=1&receivepms=1&pmnotice=1&subscriptionmethod=0&timezoneoffset=0&dstcorrection=2&regtime=1416039333&step=registration&action=do_register&regsubmit=Submit Registration!&question_id=makman' and updatexml(NULL,concat (0x3a,(SELECT username FROM
    mybb_users ORDER BY UID LIMIT 5,1)),NULL) and '1"
```

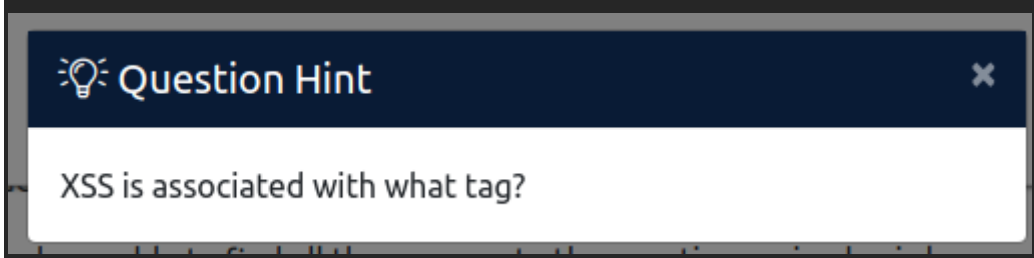
The answer was here but I wouldn't have recognized it since I don't know what it is exactly.

Questions 6 & 7

What was the value of the cookie that Kevin's browser transmitted to the malicious URL as part of an XSS attack? Answer guidance: All digits. Not the cookie name or symbols like an equal sign.

1502408189

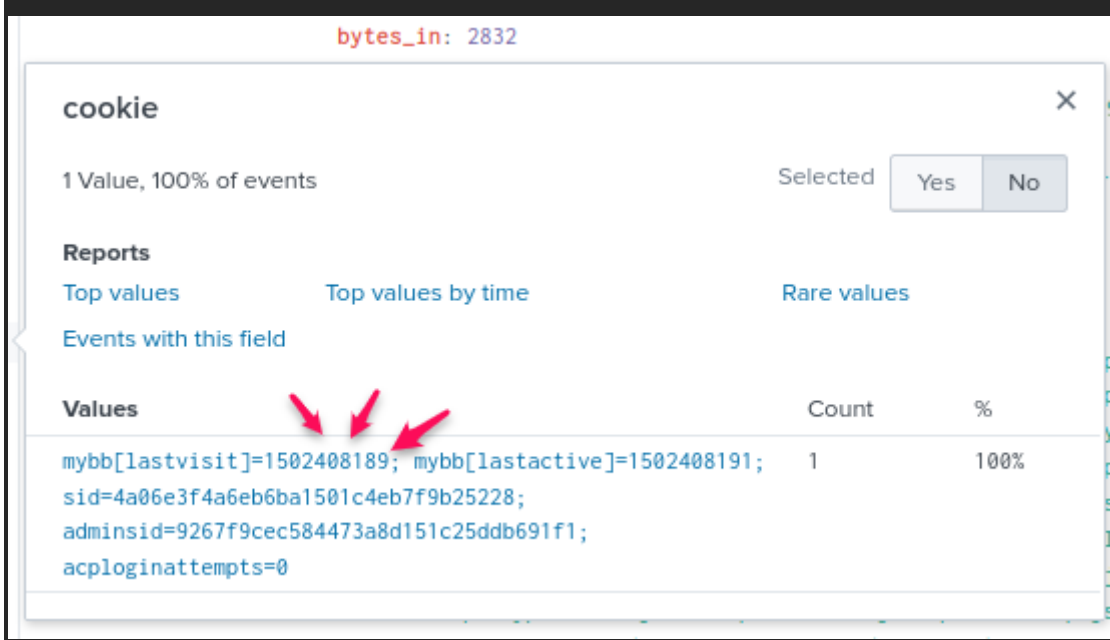
Who the HELL is Kevin? Ah, well.



ANSWER: The **<script> tag** is the most straightforward XSS payload. A script tag can reference external JavaScript code or you can embed the code within the script tag itself.

This game me one cookie entry

index="botsv2" Kevin "<script>"

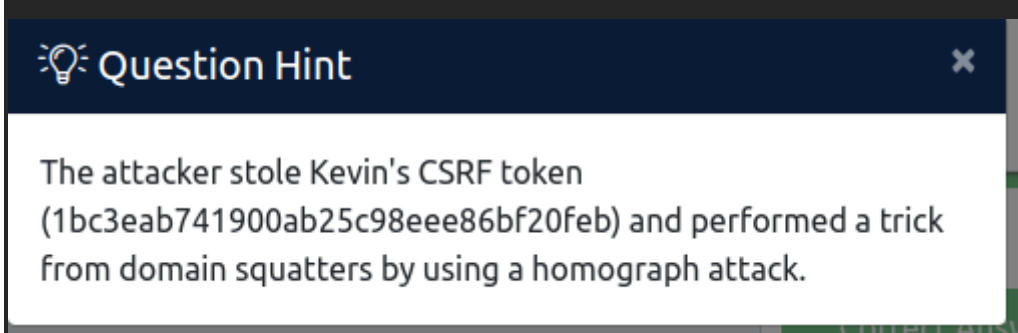


I didn't even use the guide below^^^^^^

What brewertalk.com username was maliciously created by a spear phishingattack?

klagerfield

Hmmm



Still looking at the cookie, I wanted to know where to find the CSRF token; it's under dest_content > my_post_key

```
//]]>
</script>
        <script type="text/javascript">
//<![CDATA[
var loading_text = 'Loading<br />Please wait...';
var cookieDomain = '.brewertalk.com';
var cookiePath = '/';
var cookiePrefix = '';
var imagepath = '../images';

lang.unknown_error = "An unknown error has occurred.";
lang.saved = "Saved";
//]]>
</script>
</head>
<body>
<div id="container">
        <div id="logo"><h1><span class="invisible">MyBB Admin CP</span></h1></div>
        <div id="welcome"><span class="logged_in_as">Logged in as <a href="index.php?module=user-users&amp;action=edit&amp;uid=17" class="username">kevin</a></span> | <a href="http://www.brewertalk.com" target="_blank"
class="forum">View Forum</a> | <a href="index.php?action=logout&amp;my_post_key=1bc3eab741900ab25c98eee86bf20feb" class="logout">Log Out</a></div>
<div id="menu">
<ul>
<li><a href="index.php">Home</a></li>
<li><a href="index.php?module=config">Configuration</a></li>
```

Now, on with the show!!!

index="botsv2" 1bc3eab741900ab25c98eee86bf20feb

index="botsv2" 1bc3eab741900ab25c98eee86bf20feb uri_query="module=user-users&action=add"


```
dest_ip: 172.31.4.249
dest_mac: 0A:42:7E:25:21:B4
dest_port: 80
endtime: 2017-08-16T15:19:18.185233Z
flow_id: 17a517b4-2f1f-4d6a-ab84-b02059e71241
form_data: my_post_key=1bc3eab741900ab25c98eee86bf20feb&username=kIagerfield&password=beer_lulz&confirm_password=beer_lulz&email=kIagerfield@froth.ly&usergroup=4&additionalgroups[]=4&displaygroup=4
http_comment: HTTP/1.1 302 Found
http_content_length: 0
```


WOW! I DID THIS ON MY OWN!!!!!!!!

Awesome, thus far, you have identified Amber downloaded Tor Browser (you even know the exact version). You identified what URI path and the SQL function attacked on brewertalk.com.

Your task now is to identify the cookie value that was transmitted as part of an XSS attack. The user has been identified as Kevin.

Before diving right in, get some details on Kevin. This is the first time you hear of him.

Command: index="botsv2" kevin

Ok, now you have Kevin's first and last name. Time to figure out the cookie value from the XSS attack.

As before, you can start with a simple keyword search.

You know that you're looking for events related to Kevin's HTTPtraffic with an XSS payload, and you're focused on the cookie value.

Honestly, you should be able to tackle this one on your own as well. Use the previous search queries as your guide.

After you executed the search query that yields the events with the answer, you can identify the username used for the spear phishingattack.

Based on the question hint, you can perform a keyword search query here as well.

Command: index="botsv2" ***KEYWORD***

As times before, replace **KEYWORD**with the actual keyword search term.

Great! You should have been able to find all the answers to the questions using basic keyword searches.

Answer the questions below