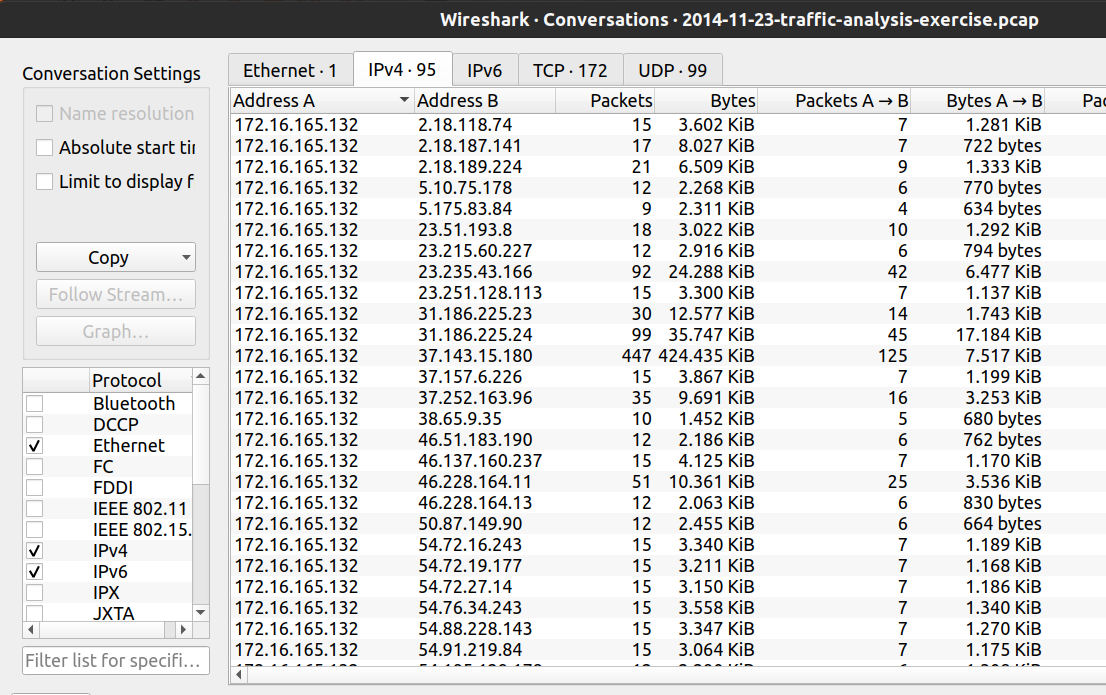
**2014-11-16 pcap Analysis**

1. Analysis performed in Ubuntu VM since malware is Windows based.
2. Exercise and pcap file located at:
   * <https://www.malware-traffic-analysis.net/2014/11/23/index.html>
3. What is the IP address of the Windows VM that gets infected?

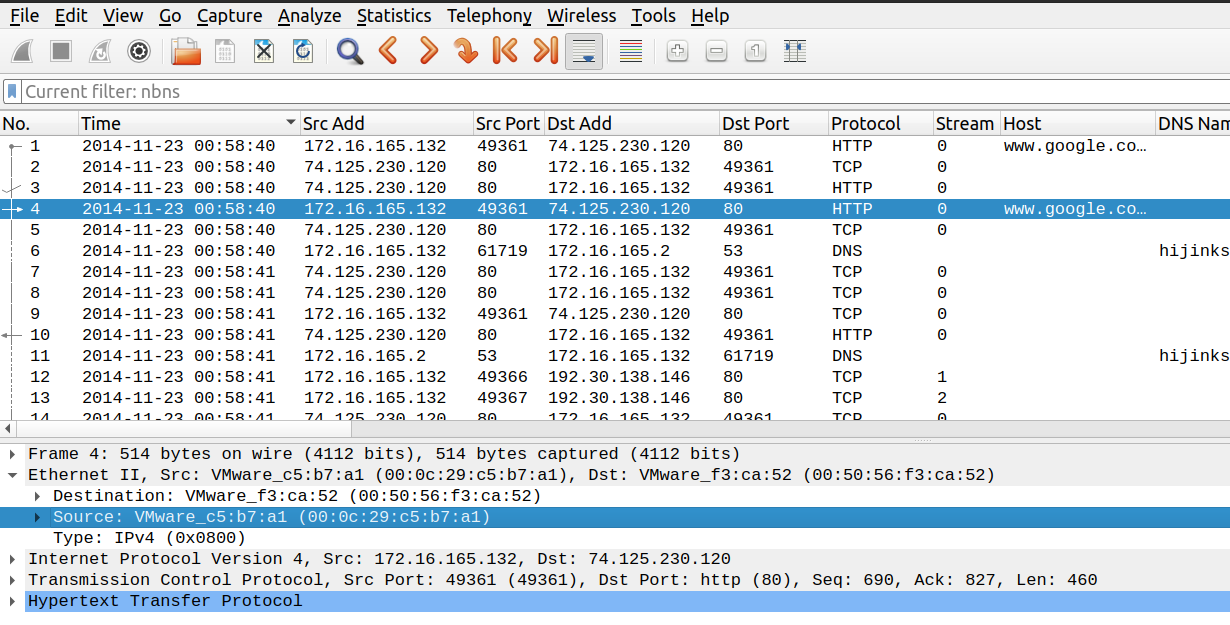
**Statistics > Conversations**



**Answer: - 172.16.165.132**

* This is the only private IP address communicating with other hosts in Conversations
* This is also the top talker having the most bytes and packets
* View all other tabs to verify this is the only IP address

1. What is the MAC address of the infected VM?
   * 1. **Select any packet with source IP 172.16.165.132**
     2. **Packet Details Pane > Expand Ethernet II > Source**



**2**

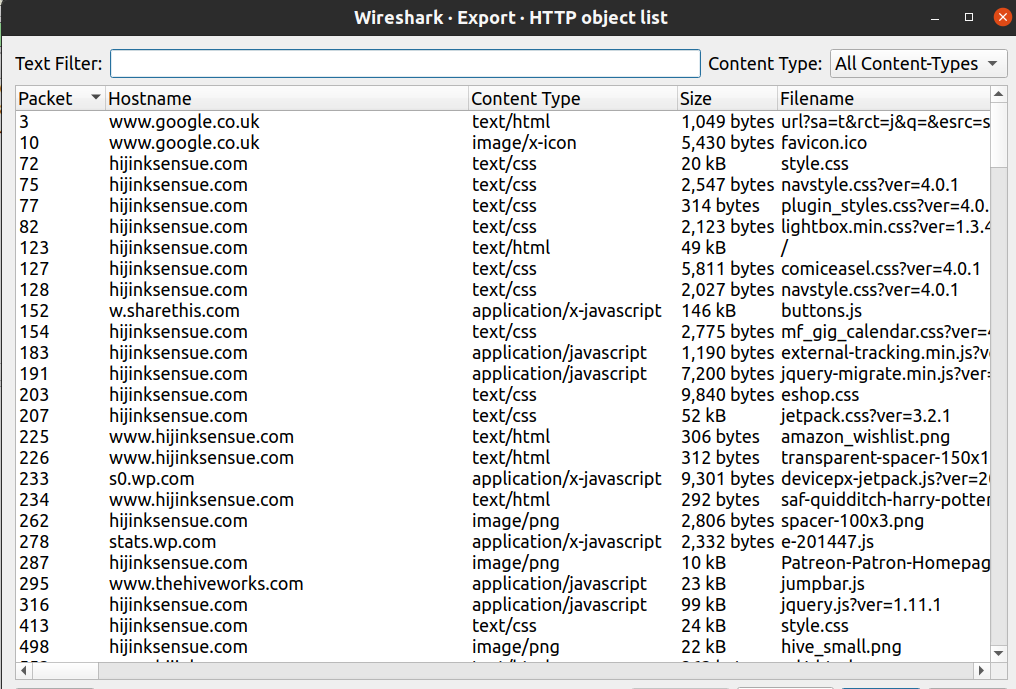
**1**

**Answer: - 00:0c:29:c5:b7:a1 (VMWare)**

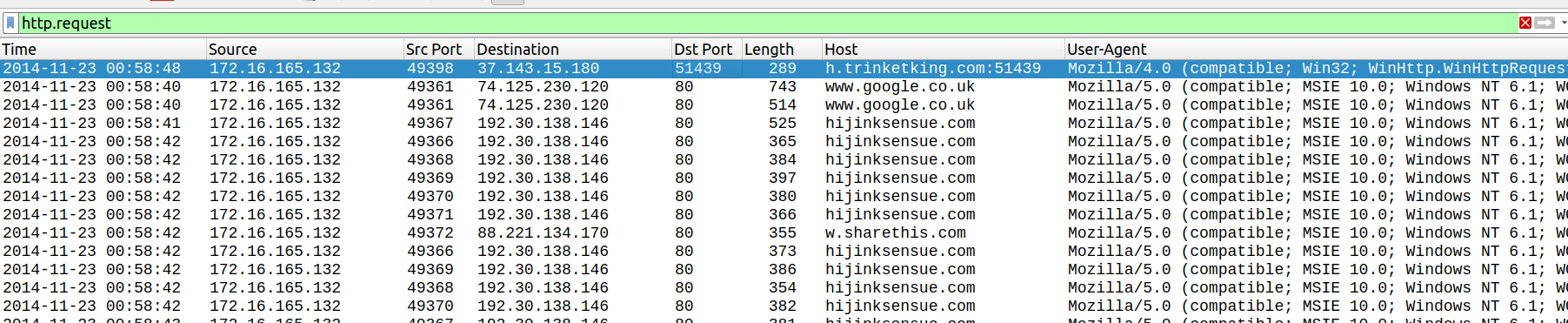
* + Any packet with the source or destination IP address of the infected device will have this information
  + This is OSI Layer II data

1. What is the IP address and domain name that delivered the exploit kit and malware?

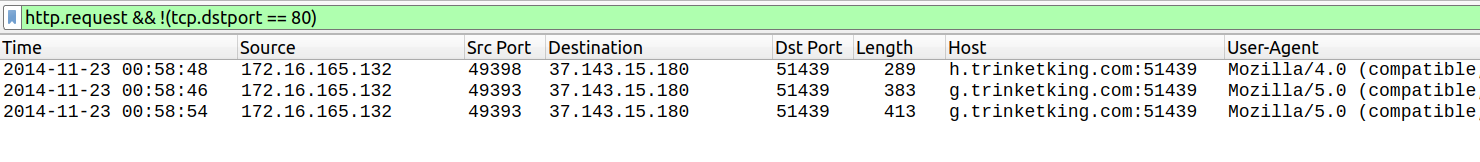
* We can find all files downloaded in this pcap by going to
  + **File > Export Objects > HTTP**



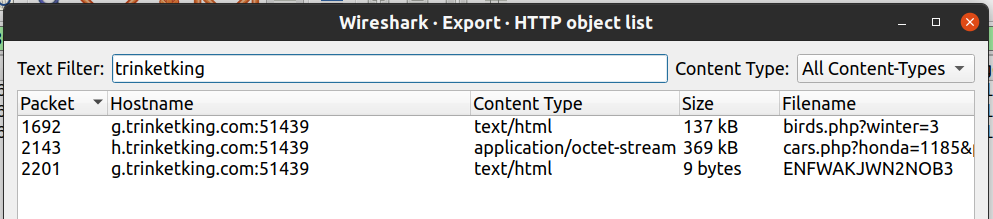
* If there were hundreds of files, it would not be efficient to submit all to Virustotal.
* Let’s narrow down this list by looking for http requests out of the ordinary
  + Look for suspicious User-Agents
    1. Grab an http GET request packet and set User-Agent as a column
    2. **Display filter > http.request > sort User-Agent column**
    3. By scrolling all requests have the same User-Agent except the top one. Take note of this.



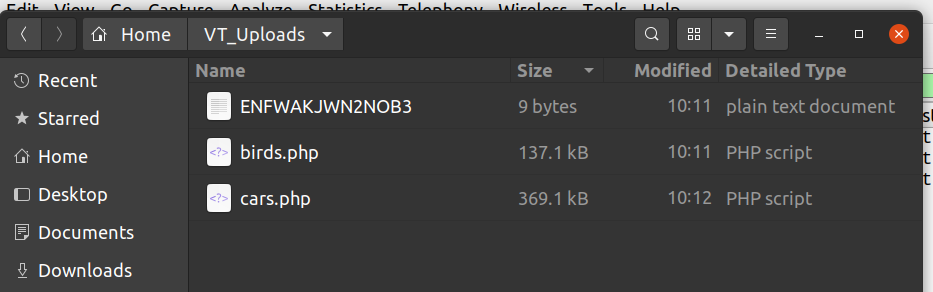
* + Look for suspicious port numbers
    1. Sort by destination port or filter out port 80
    2. Sorting out port 80 makes other ports stand out
    3. **http.request && !(tcp.dstport == 80)**
    4. These ports are not communicating on port 80 (suspicious)



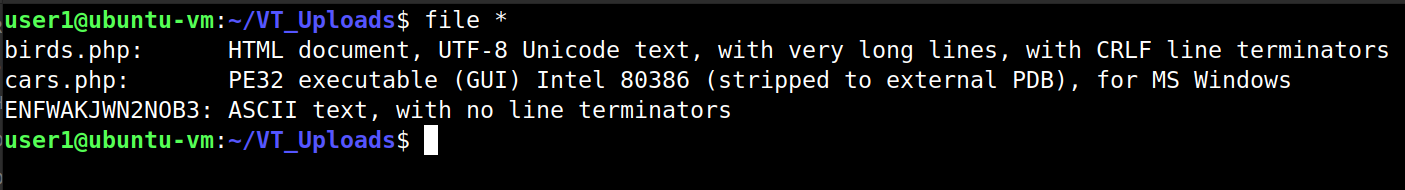
* Take note of the host name
* Back to **File > Export Objects > HTTP**
* Filter for “trinketking” and save these 3 files



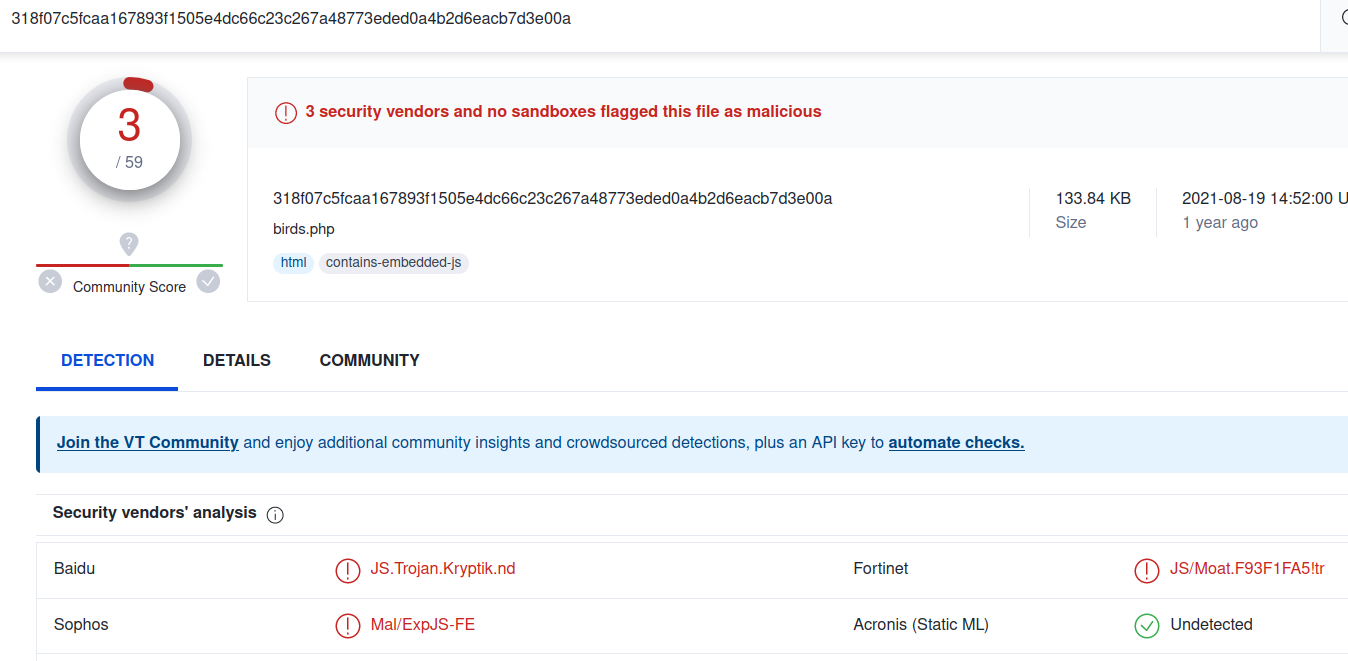
* Saved files



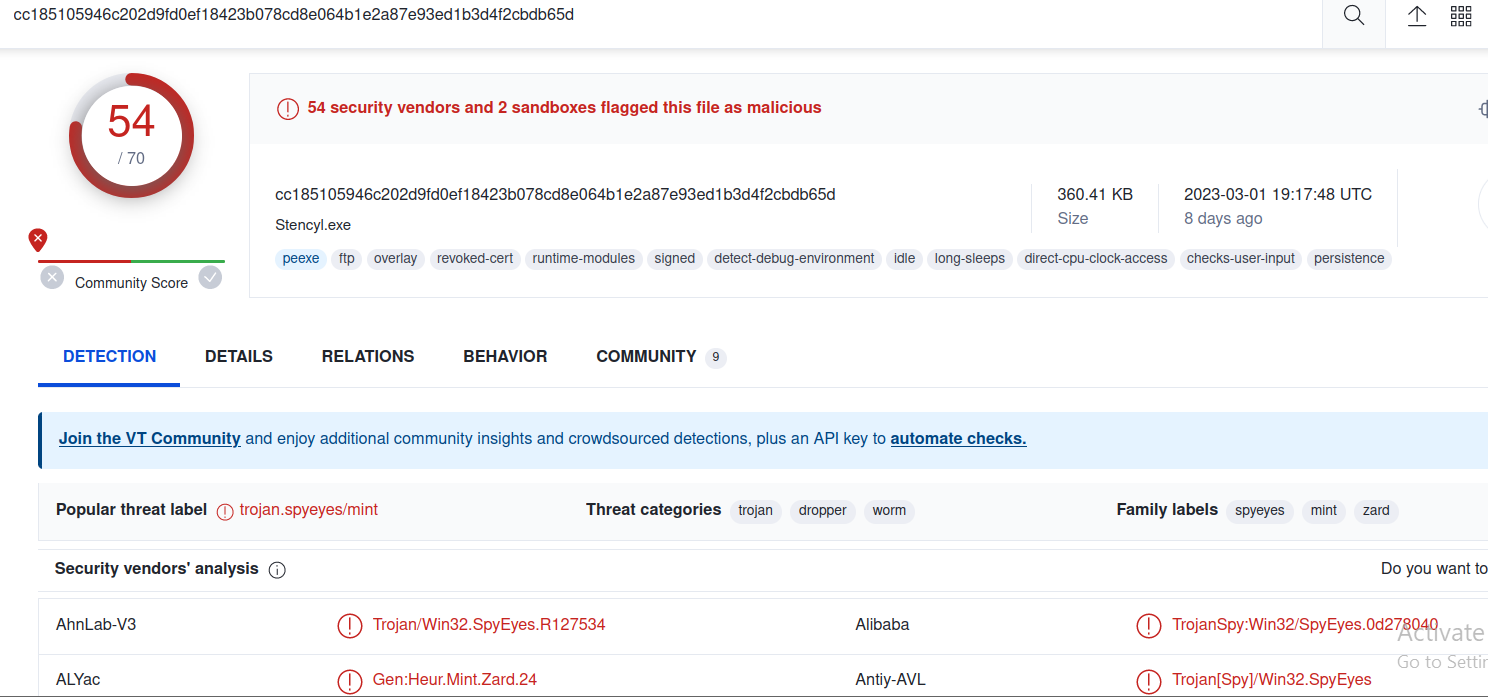
* Perform a quick analysis to check the file type
* **Open a terminal > cd <malware folder> > use command: file \***
* Are these files what they claim to be?



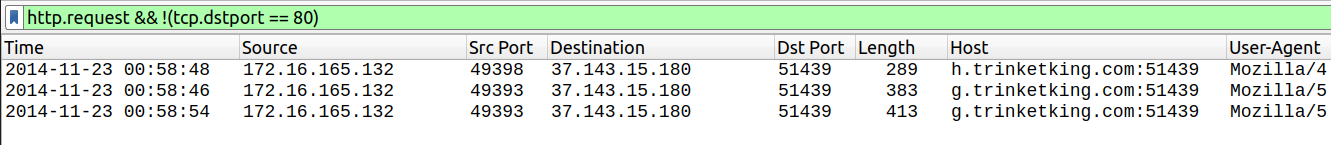
* Cars.php is definitely suspicious since this is not a php file, but a PE32 executable for MS Windows
* Use the **cat** **<filename>** command to view the other non-PE32 files
  + If you scroll through birds.php, you will see this does not look like normal html code; this is in fact, obfuscated JavaScript code
* Next submit these files or their hashes to Virustotal:
* Birds.php came back as malicious



* cars.php came back as malicious

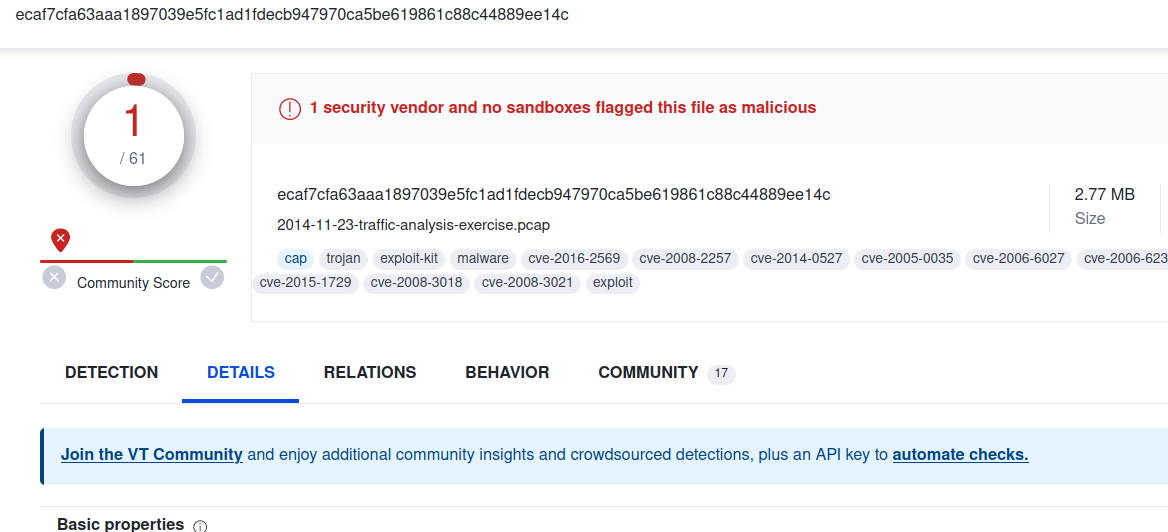


* We can conclude that two hosts are malicious:
  + **g.trinketking.com** delivered the exploit kit (birds.php)
  + **h.tinketking.com** delivered the malware (cars.php)
  + These hosts use one IP address of **37.143.15.180** over port **51439**

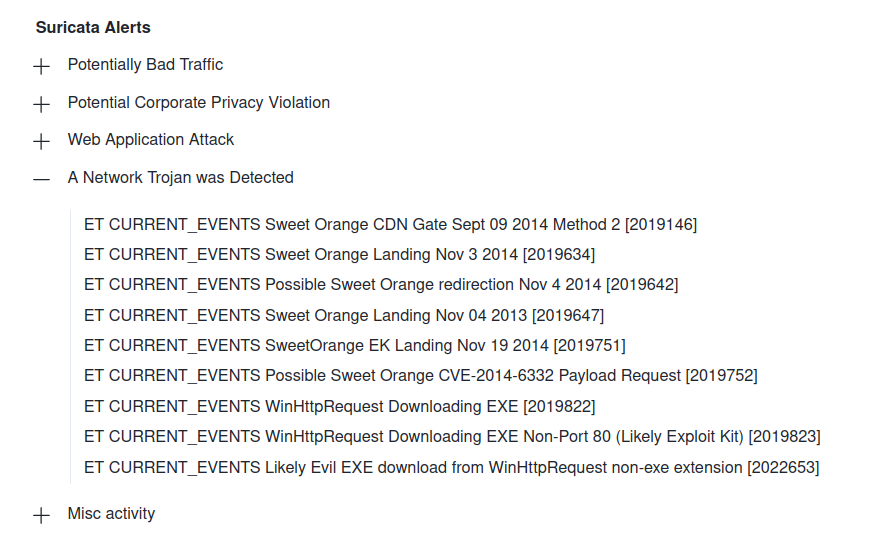


1. What is the exploit kit (EK) that delivers the malware?

* Upload the pcap file to Virustotal > Select **Details**

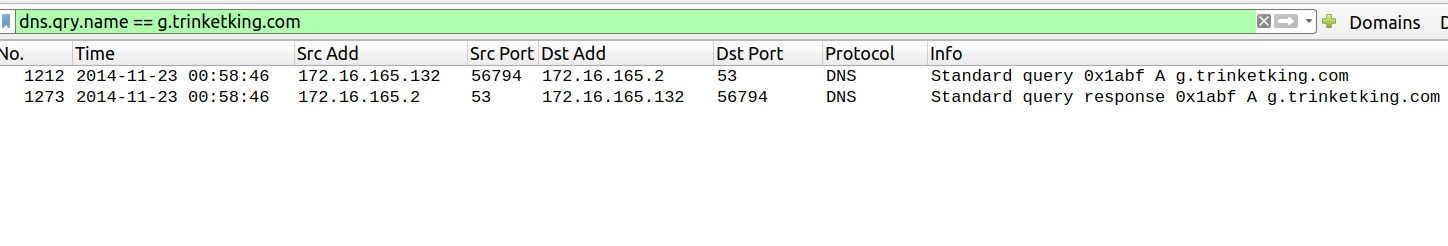


* Scroll down to Suricata or Snort Alerts
* Expand “A Network Trojan was Detected”:

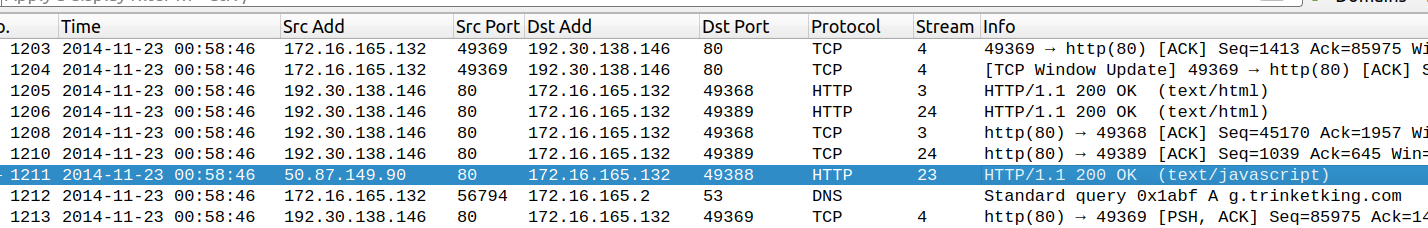


1. What is the redirect URL and IP address that points to the exploit kit (EK) landing page?

* From Question 2 above, use the malicious URL as a pivot
* **Display filter > dns.qry.name == g.trinketking.com**



* Let’s select packet 1212, then clear the Display Filter
* What happened just before this DNS query?
* There must have been some sort of instruction over http that requested the malicious URL
* Let’s have a look at the http packet just before packet 1212



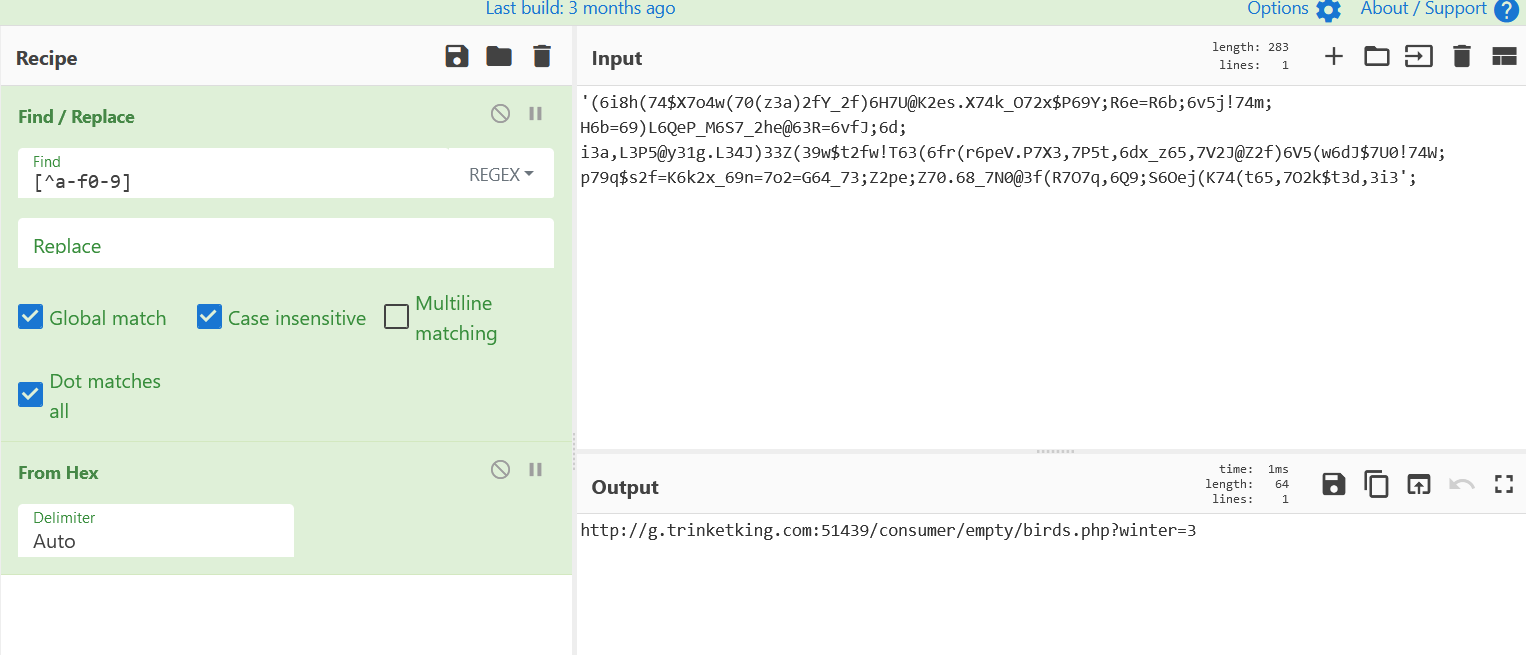
* Follow the HTTP stream
* First thing to notice is the strange GET request file
* Take note of the Host



* Scroll down to see the response
* Notice the obfuscated code



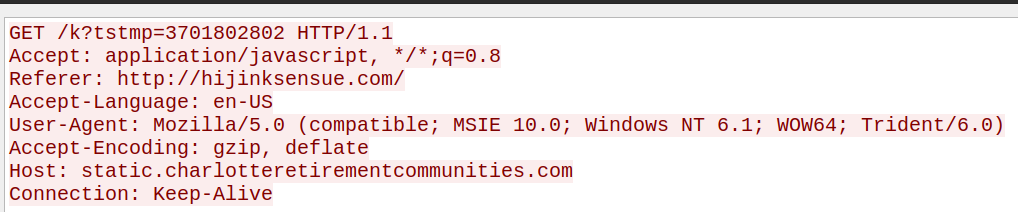
* A common encoding method is to encode the real part of the code in hexadecimal and obfuscate with junk characters in between
* Using CyberChef, let’s remove hexadecimal characters 0-9, a-f
* Then convert the characters back to Ascii



* The output shows the familiar malicious URL
* The redirect URL and IP address:
  + **static.charlotteretirementcommunities.com/k?tstmp=3701802802**
  + **50.87.149.90**

1. What is the IP address of the compromised web site?

* Let’s work back to that last HTTP stream and find the Referrer



1. What is the domain name of the compromised web site?

