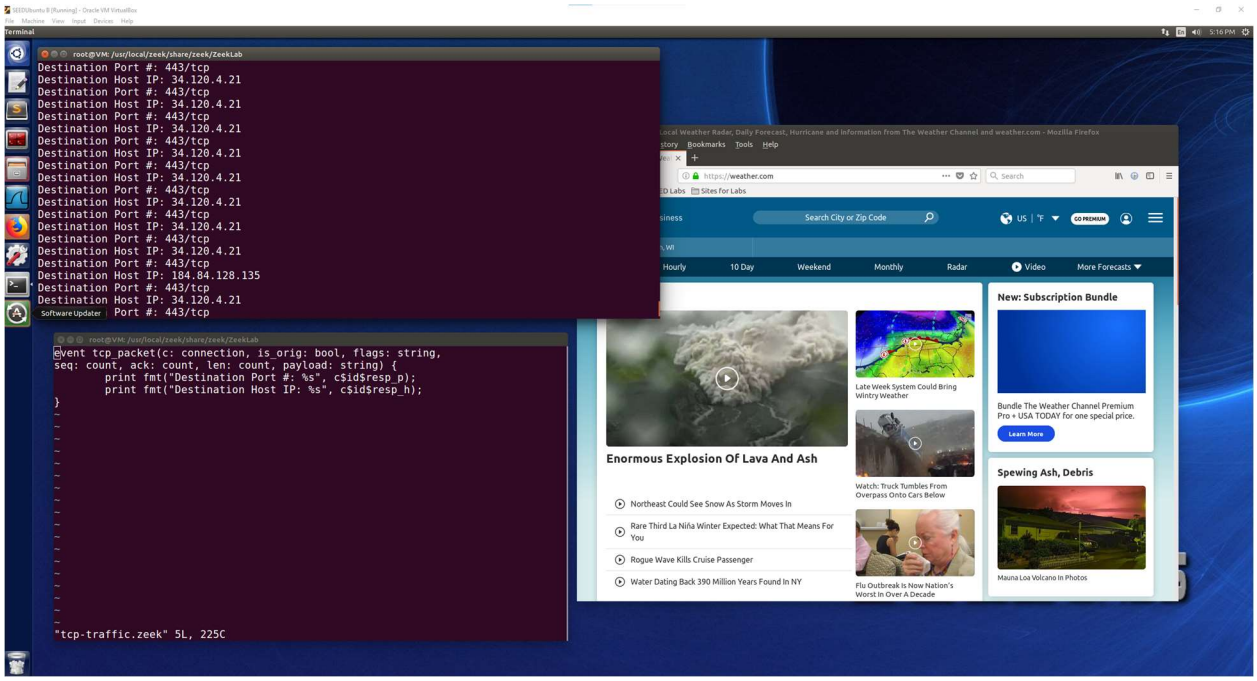
Lab 5
Gunnar Yonker

Executing a TCP Zeek Script



The website I was browsing was https://weather.com and the Zeek window(top terminal window) is showing that the destination IP address is that of weather.com. It shows that when browsing there are two different IP addresses being navigated to and that the port being used on this website for this connection is port 443.

In the modified script, the first print fmt line shows that the Destination Port number being printed out through the variable cidresp_p and this is seen in the Zeek printout window as 443/tcp. This connection is on port 443 and is a tcp pkt connection. The next print line shows Destination Host IP and is printed out using the variable cidresp_h which is the host IP address. This is seen in the Zeek window as printing out Destination Host IP: 34.120.4.21. By using resp_p and resp_h we can gather the destination port and IP information. If orig_p and orig_h was used, then it would be the information of the machine making the request not the destination information.

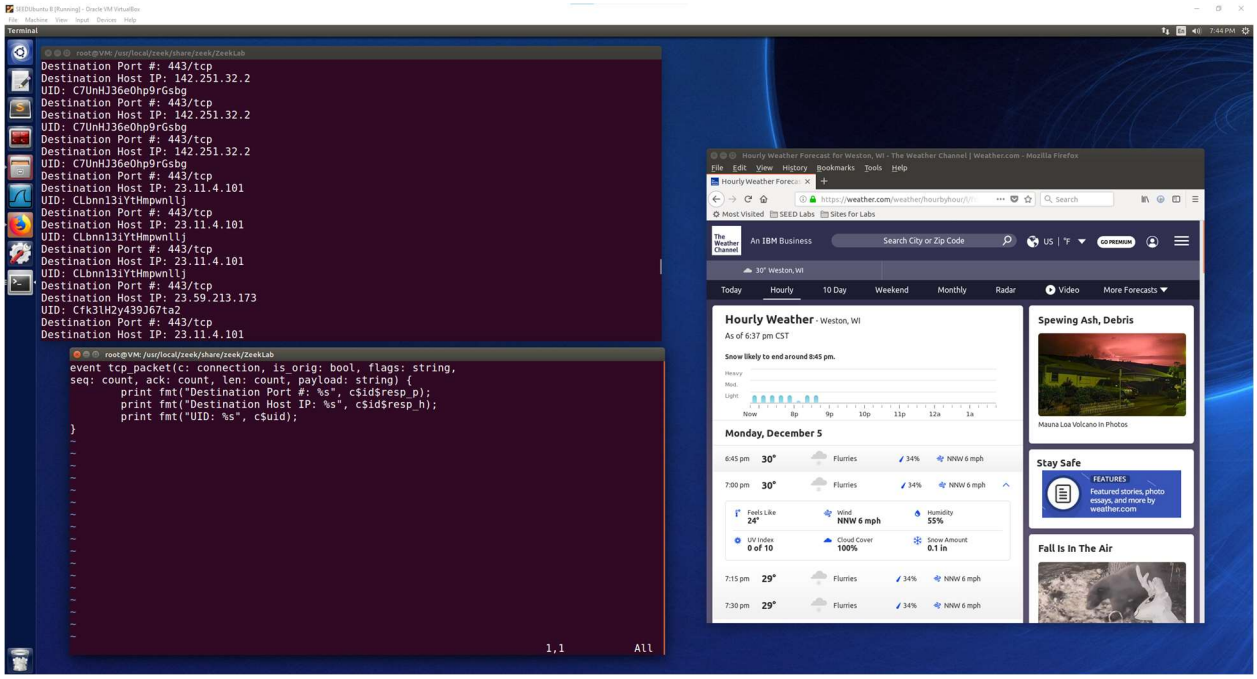
I attempted to determine what script adjustments I would need to also print out the sequence number but in the Zeek guide I was only able to find get_resp_seq which I could not figure out how to use it when entering it into my script without producing an error. If cidresp_h with %d to produce an integer printout, it would become a formatting error due to that variable being a string representation. I was unable to find any guidelines on where to find information on this script modification in the lecture slides unless I overlooked that portion of the slides. I tried to add in the get_resp_seq at various location but couldn't seem to find the right location for it that would not produce a segmenting error when running the zeek script. Below are examples of what I did try when encountering these errors:

print fmt("Seq: %d", get_resp_seq);

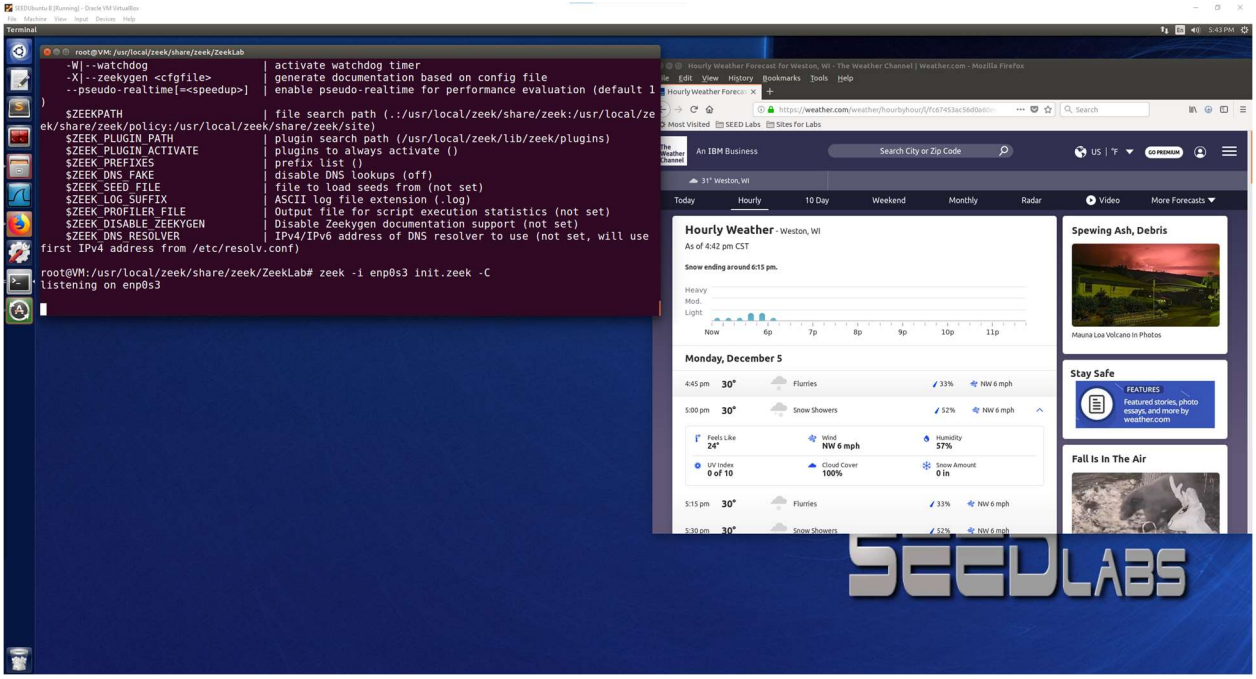
print fmt("Seq: %d", cidget_resp_seq);

The next method I tried was to add in the uid of the connection so that the connection could still be traced. I am not sure if this is what the intention of this task was asking for, but it is a solution that I was able to get working to demonstrate that the uid could be added into the printout of information when the zeek script was running. This is shown below where the destination IP, port, and uid are shown.

This shows that the uid of the connection can additionally be added to the output of the zeek script during the web browsing with no errors encountered.

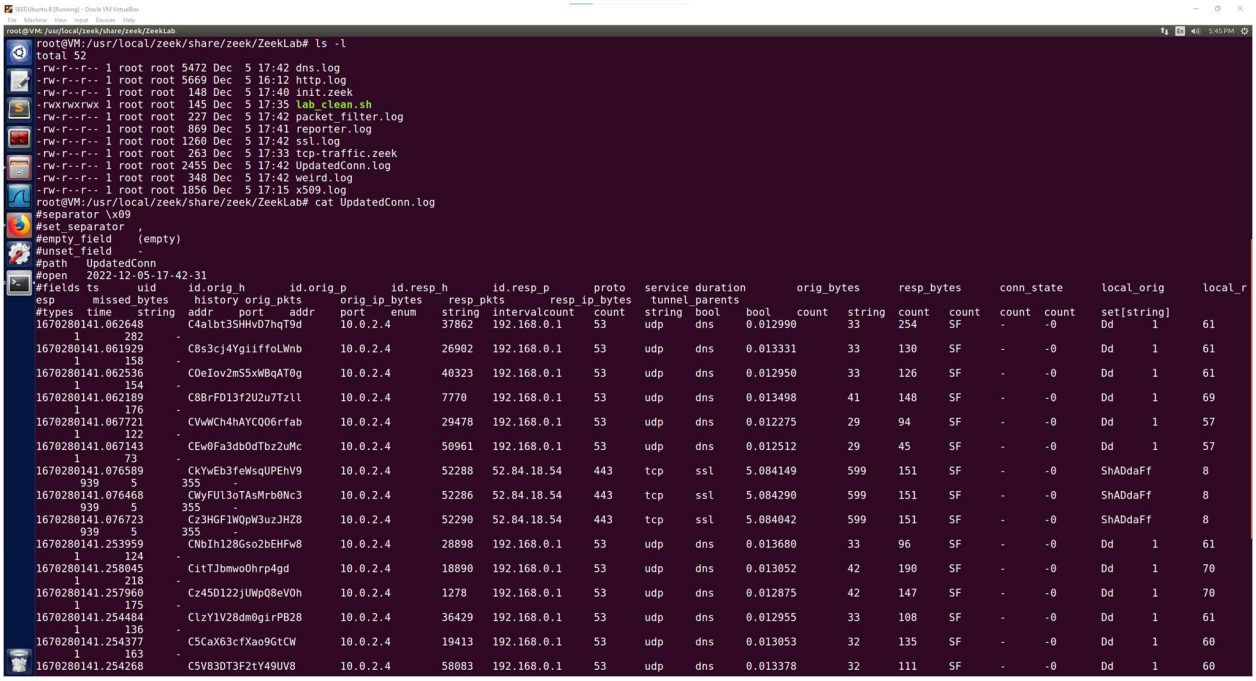


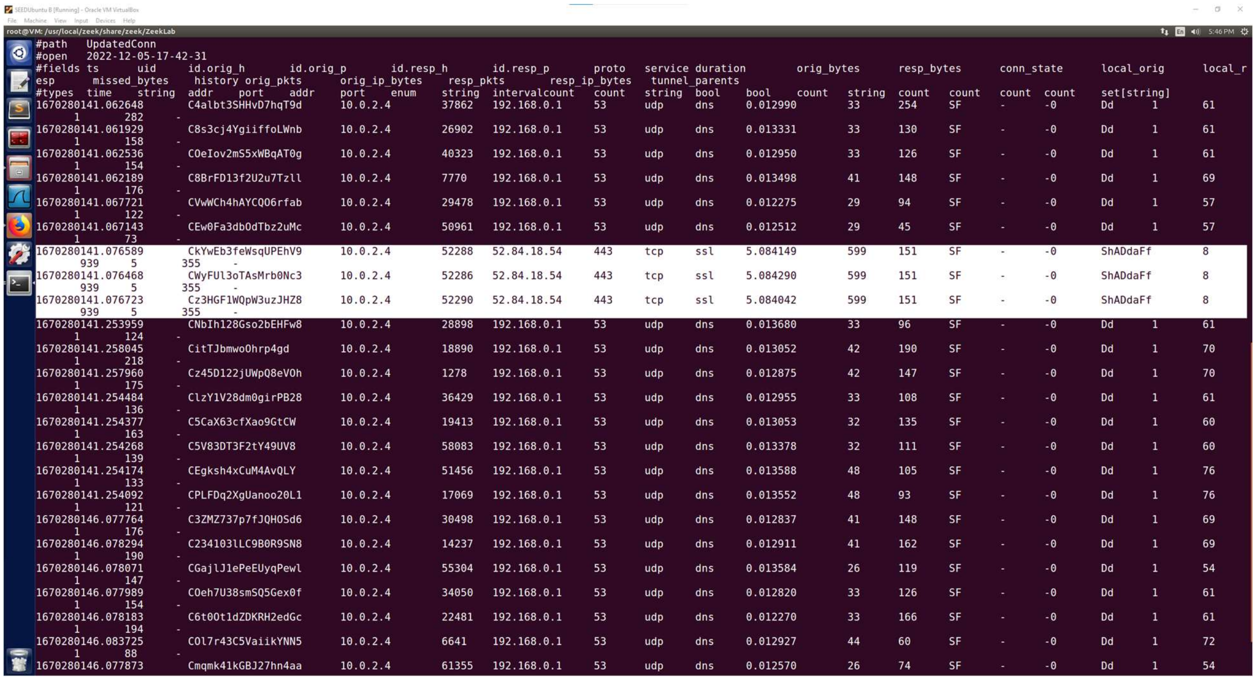
Modifying Zeek log streams



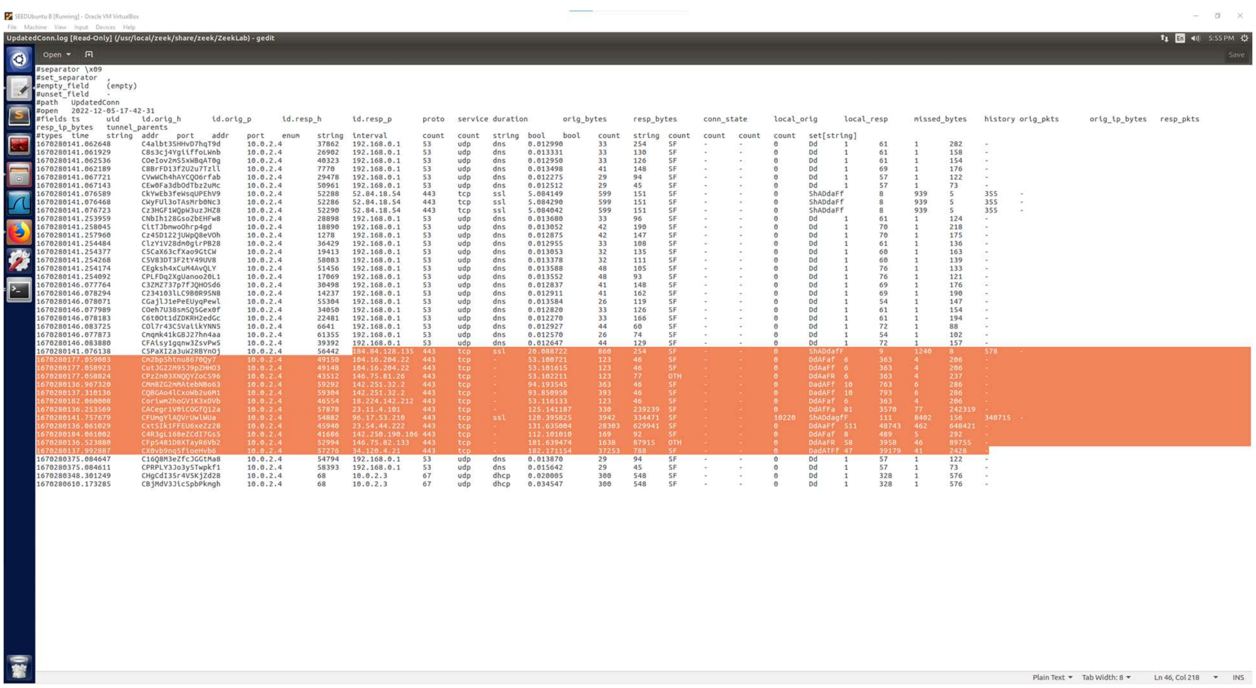
Browsing https://weather.com again with the init.zeek script running.

Showing the contents of the UpdatedConn.log file located in the ZeekLab folder:





This screenshot above shows the connections to the weather.com website that I was browsing, when the website was connected to it shows the destination IP address of 52.84.18.54 and that port 443 was connected to. Also shown is ssl which means that there is a secure connection the website, this can further be confirmed because the website is an https connection, not http which means there is an authenticated certificate. These connections also are attributed with SF which means that this connection is considered a “normal establishment and termination”. This just means that my connection was normal and that the termination of the connection was also normal, there were no abnormalities about how the system was connected/terminated to this destination host.

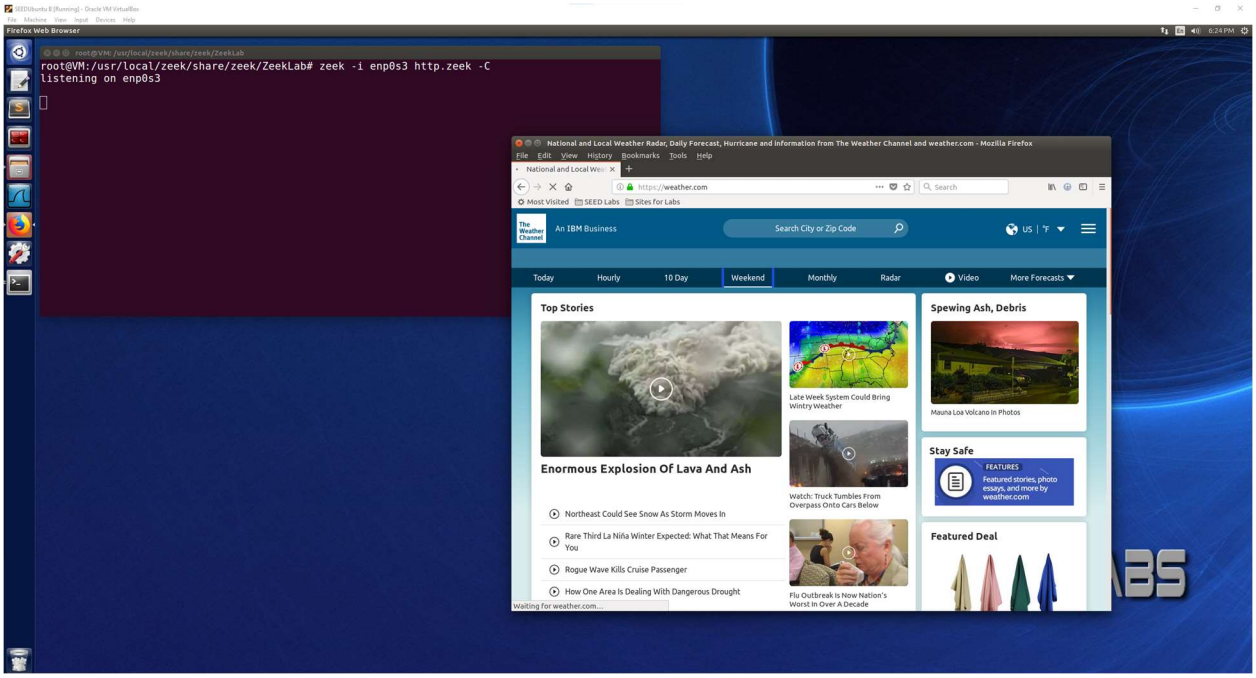


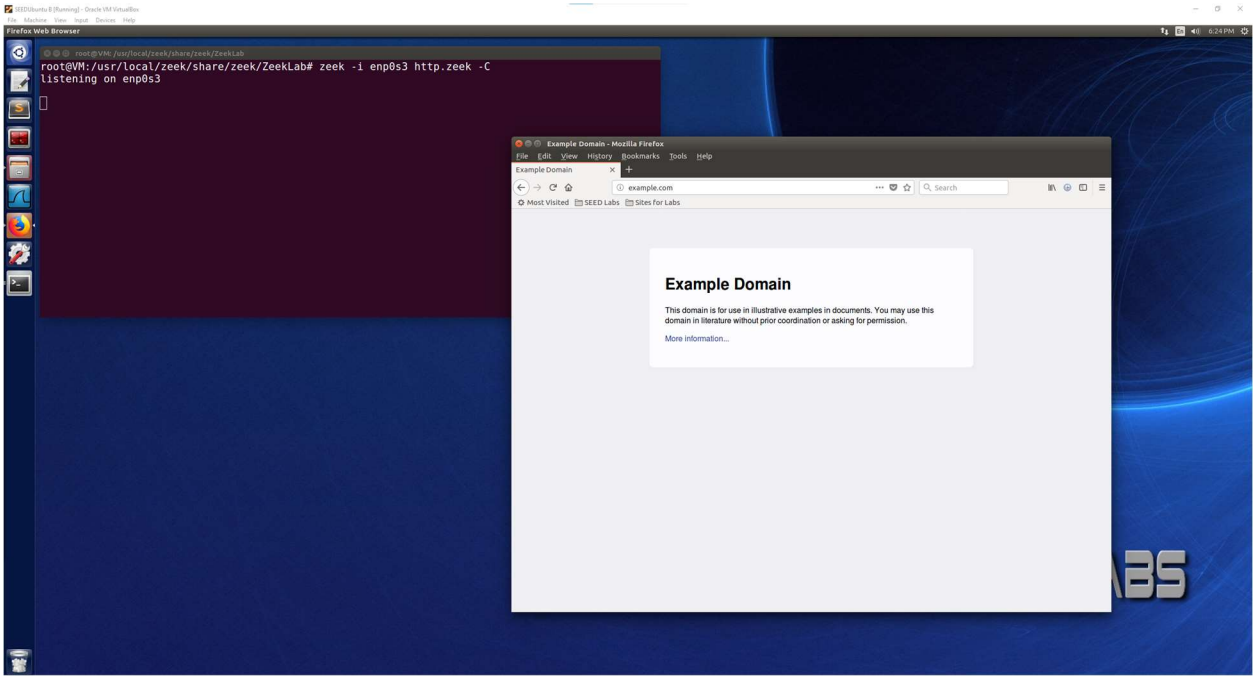
Similarly looking further down the file, other browsing activity was also collected through tcp connections also using port 443. OTH means that there was no SYN seen and that it is just midstream traffic which can be attributed to a partial connection.

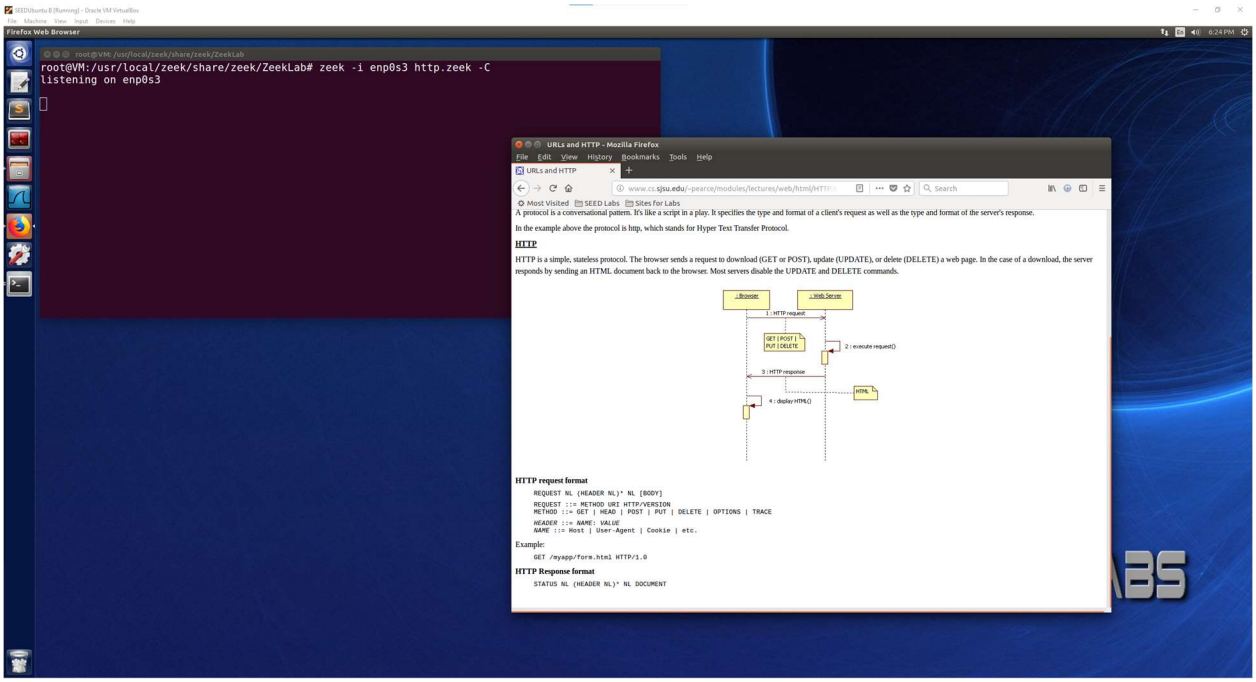
Additional with this information the UID is given for each connection which can help to trace the connection across different logs that are collected by Zeek. The originating port is also given for these connections coming from the IP address of my VM. The timestamp(ts) could also be useful information for when the connection was established but is represented as a very large number. The orig bytes and resp bytes also show how many bytes were sent out in the packet and how many bytes were received by the host. That is the information from this log attributed to my web browsing that I can understand from this log, there is more information than just that included in this log, but most importantly I am able to see when my connections took place, the ports used, the destination IP address, protocol, and if ssl was used along with what the connection state is logged as.

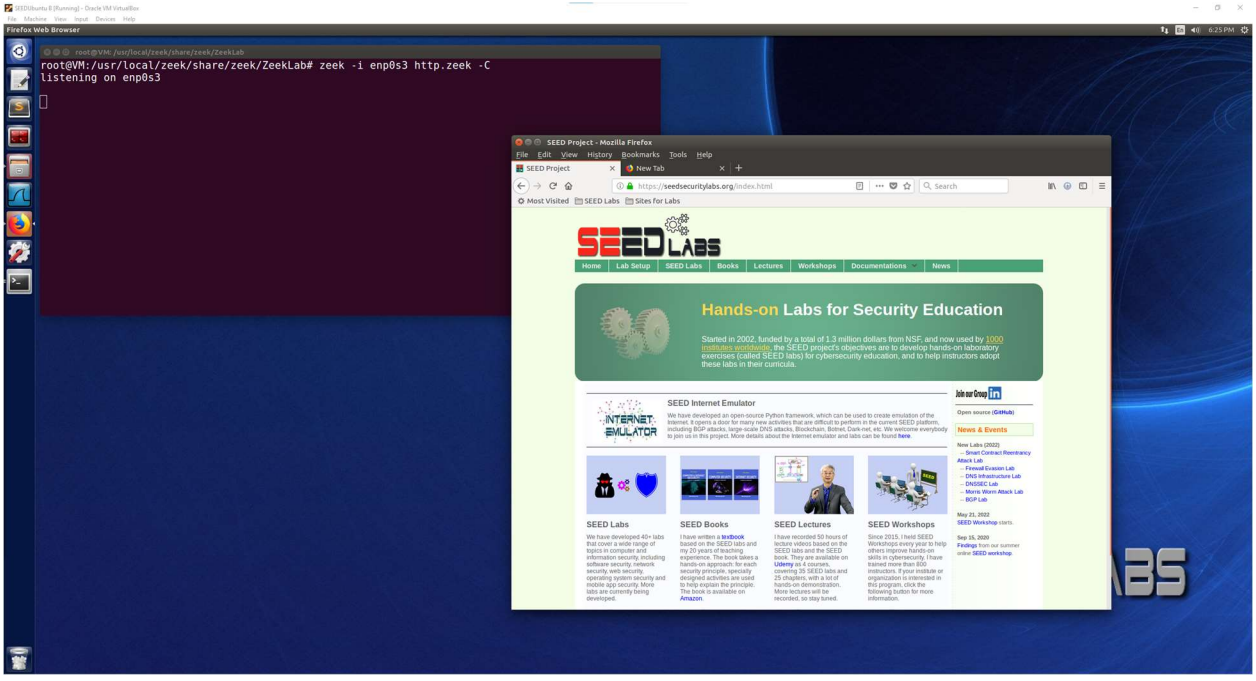
Updating the conn.log stream

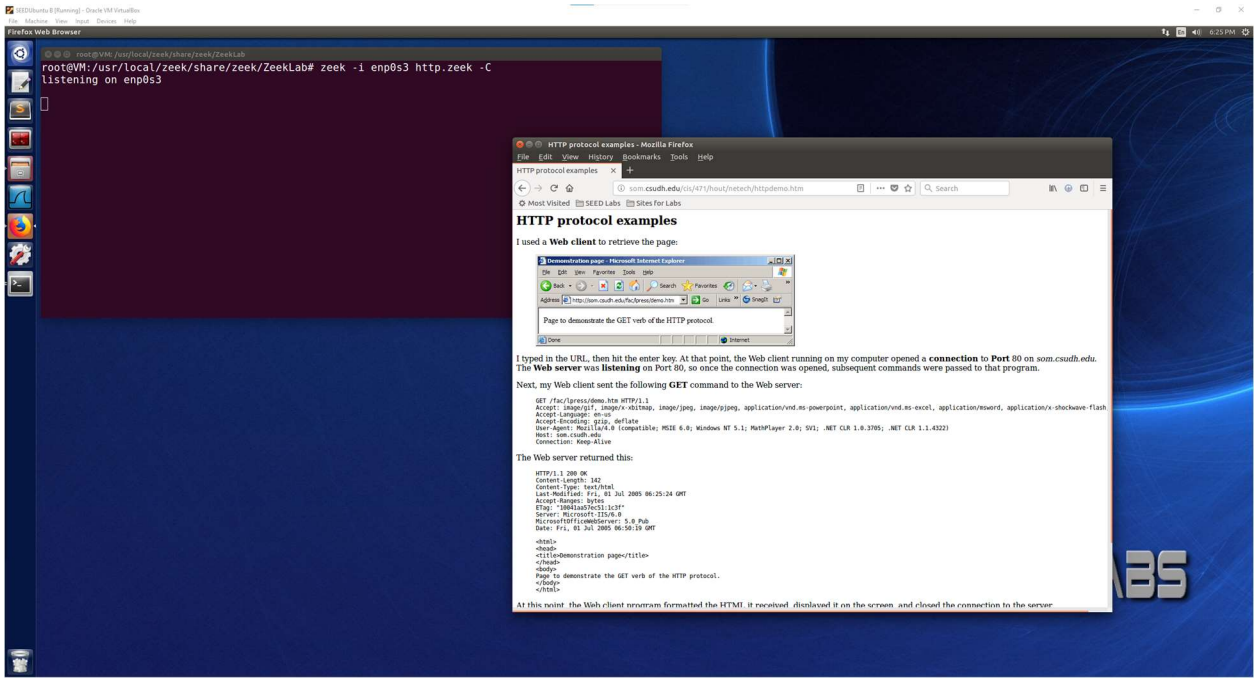
Browsed websites:

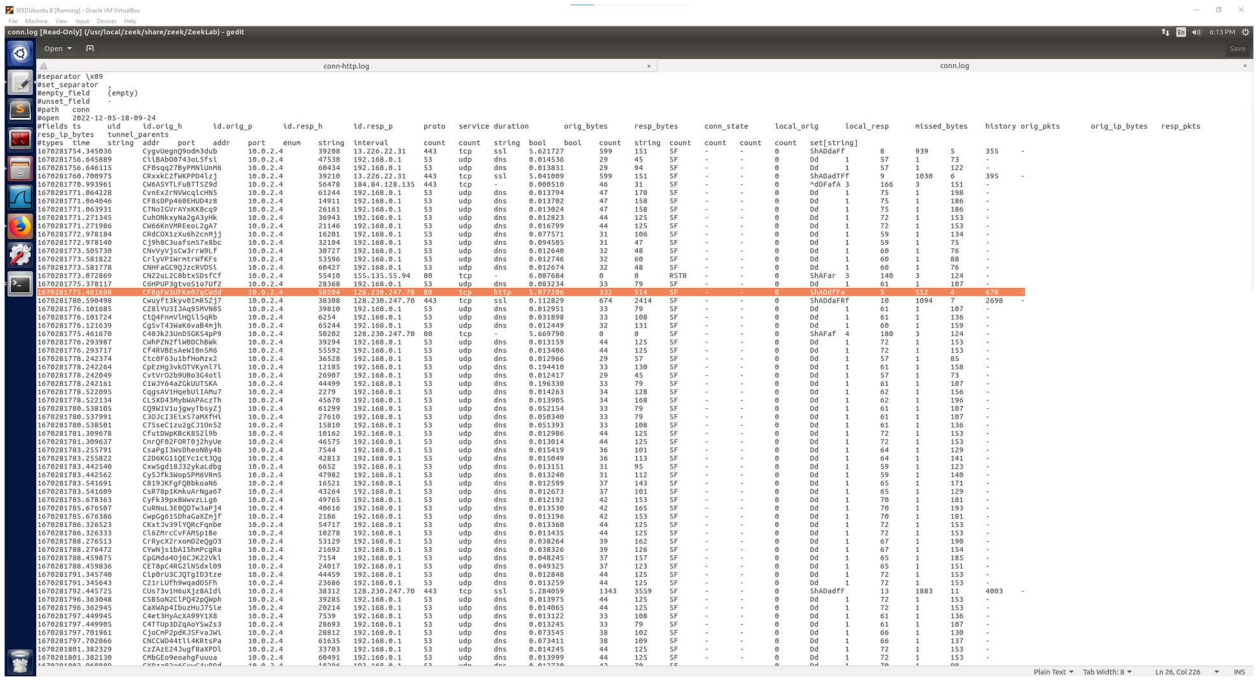




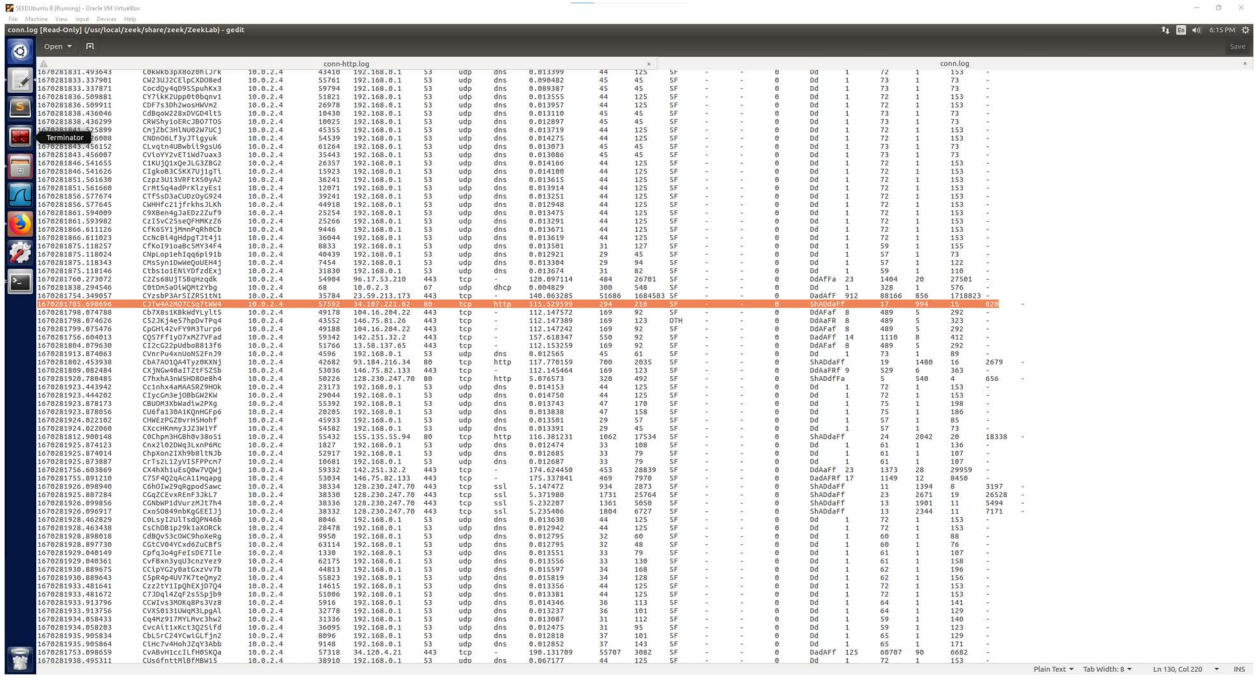

Firefox Web Browser
root@VM:/usr/local/zeek/share/zeek/ZeekLab
root@VM:/usr/local/zeek/share/zeek/ZeekLab# zeek -i enp0s3 http.zeek -C
listening on enp0s3
URLs and HTTP - Mozilla Firefox
File Edit View History Bookmarks Tools Help
URLs and HTTP
www.cs.sjsu.edu/~pearce/modules/lectures/web/html/HTTP.htm
Most Visited SEED Labs Sites for Labs
A protocol is a conversational pattern. It's like a script in a play. It specifies the type and format of a client's request as well as the type and format of the server's response.
In the example above the protocol is http, which stands for Hyper Text Transfer Protocol.
HTTP
HTTP is a simple, stateless protocol. The browser sends a request to download (GET or POST), update (UPDATE), or delete (DELETE) a web page. In the case of a download, the server responds by sending an HTML document back to the browser. Most servers disable the UPDATE and DELETE commands.
: Browser
: Web Server
1 : HTTP request
GET | POST | PUT | DELETE
2 : execute request()
3 : HTTP response
HTML
4 : display HTML()
HTTP request format
REQUEST NL (HEADER NL)* NL [BODY]
REQUEST ::= METHOD URI HTTP/VERSION
METHOD ::= GET | HEAD | POST | PUT | DELETE | OPTIONS | TRACE
HEADER ::= NAME: VALUE
NAME ::= Host | User-Agent | Cookie | etc.
Example:
GET /myapp/form.html HTTP/1.0
HTTP Response format
STATUS NL (HEADER NL)* NL DOCUMENT

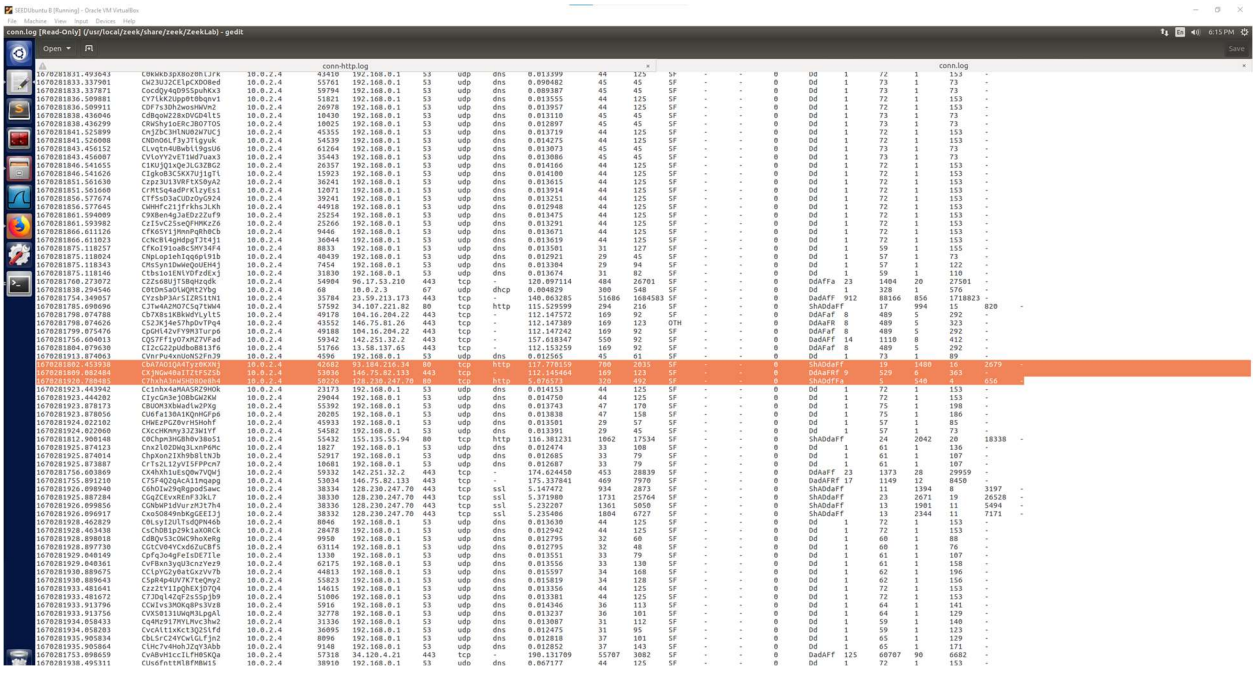

Firefox Web Browser
root@VM:/usr/local/zeek/share/zeek/ZeekLab
root@VM:/usr/local/zeek/share/zeek/ZeekLab# zeek -i enp0s3 http.zeek -C
listening on enp0s3
SEED Project - Mozilla Firefox
File Edit View History Bookmarks Tools Help
SEED Project
New Tab
https://seedsecuritylabs.org/index.html
Most Visited SEED Labs Sites for Labs
SEED LABS
Home Lab Setup SEED Labs Books Lectures Workshops Documentations News
Hands-on Labs for Security Education
Started in 2002, funded by a total of 1.3 million dollars from NSF, and now used by 1000 institutes worldwide, the SEED project's objectives are to develop hands-on laboratory exercises (called SEED labs) for cybersecurity education, and to help instructors adopt these labs in their curricula.
SEED Internet Emulator
We have developed an open-source Python framework, which can be used to create emulation of the Internet. It opens a door for many new activities that are difficult to perform in the current SEED platform, including BGP attacks, large-scale DNS attacks, Blockchain, Botnet, Dark-net, etc. We welcome everybody to join us in this project. More details about the Internet emulator and labs can be found here.
Join our Group
Open source (GitHub)
News & Events
New Labs (2022)
– Smart Contract Reentrancy Attack Lab
– Firewall Evasion Lab
– DNS Infrastructure Lab
– DNSSEC Lab
– Morris Worm Attack Lab
– BGP Lab
May 21, 2022
SEED Workshop starts.
Sep 15, 2020
Findings from our summer online SEED workshop.
SEED Labs
We have developed 40+ labs that cover a wide range of topics in computer and information security, including software security, network security, web security, operating system security and mobile app security. More labs are currently being developed.
SEED Books
I have written a textbook based on the SEED labs and my 20 years of teaching experience. The book takes a hands-on approach; for each security principle, specially designed activities are used to help explain the principle. The book is available on Amazon.
SEED Lectures
I have recorded 50 hours of lecture videos based on the SEED labs and the SEED book. They are available on Udemy as 4 courses, covering 35 SEED labs and 25 chapters, with a lot of hands-on demonstration. More lectures will be recorded, so stay tuned.
SEED Workshops
Since 2015, I held SEED Workshops every year to help others improve hands-on skills in cybersecurity. I have trained more than 800 instructors. If your institute or organization is interested in this program, click the following button for more information.

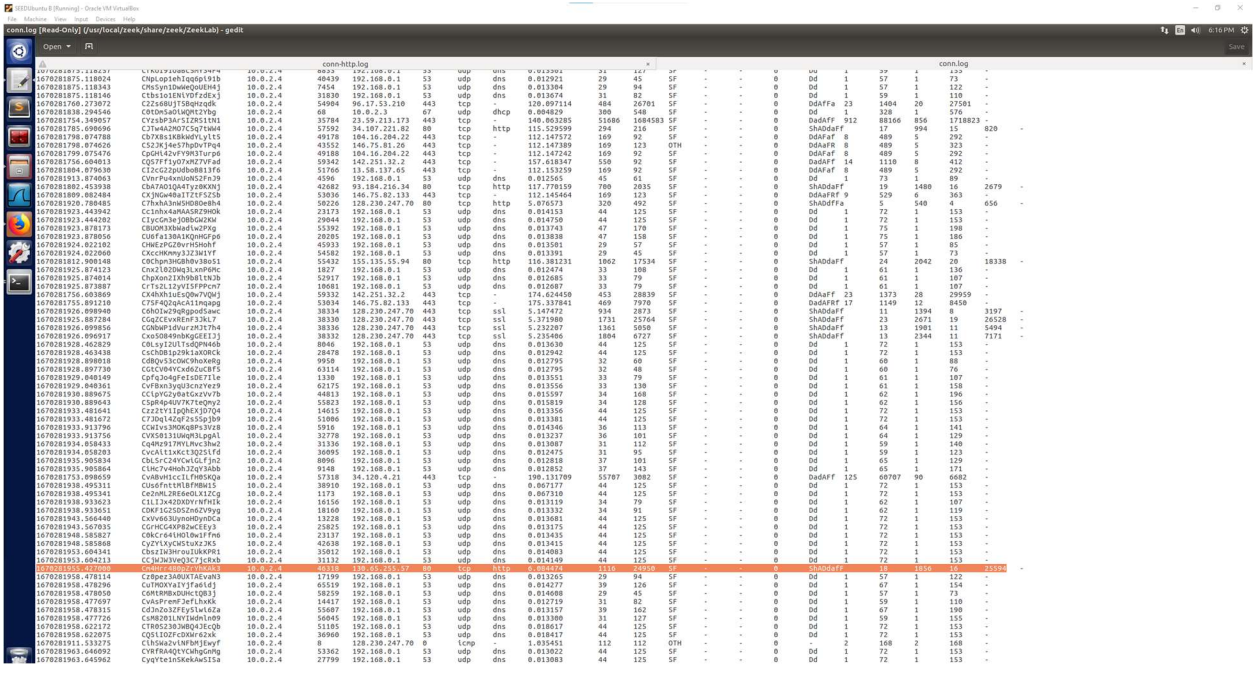

SEEDUbuntu 8 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Firefox Web Browser
6:25 PM
root@VM: /usr/local/zeek/share/zeek/ZeekLab
root@VM:/usr/local/zeek/share/zeek/ZeekLab# zeek -i enp0s3 http.zeek -C
listening on enp0s3
HTTP protocol examples - Mozilla Firefox
File Edit View History Bookmarks Tools Help
HTTP protocol examples
som.csudh.edu/cis/471/hout/netech/httpdemo.htm
Search
Most Visited SEED Labs Sites for Labs
HTTP protocol examples
I used a Web client to retrieve the page:
Demonstration page - Microsoft Internet Explorer
Page to demonstrate the GET verb of the HTTP protocol.
I typed in the URL, then hit the enter key. At that point, the Web client running on my computer opened a connection to Port 80 on som.csudh.edu. The Web server was listening on Port 80, so once the connection was opened, subsequent commands were passed to that program.
Next, my Web client sent the following GET command to the Web server:
GET /fac/lpress/demo.htm HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-powerpoint, application/vnd.ms-excel, application/msword, application/x-shockwave-flash
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; MathPlayer 2.0; SV1; .NET CLR 1.0.3705; .NET CLR 1.1.4322)
Host: som.csudh.edu
Connection: Keep-Alive
The Web server returned this:
HTTP/1.1 200 OK
Content-Length: 142
Content-Type: text/html
Last-Modified: Fri, 01 Jul 2005 06:25:24 GMT
Accept-Ranges: bytes
ETag: "10041ad57ec51:1c3f"
Server: Microsoft-IIS/6.0
MicrosoftOfficeWebServer: 5.0_Pub
Date: Fri, 01 Jul 2005 06:50:19 GMT
<html>
<head>
<title>Demonstration page</title>
</head>
<body>
Page to demonstrate the GET verb of the HTTP protocol.
</body>
</html>
At this point, the Web client program formatted the HTML it received, displayed it on the screen, and closed the connection to the server.



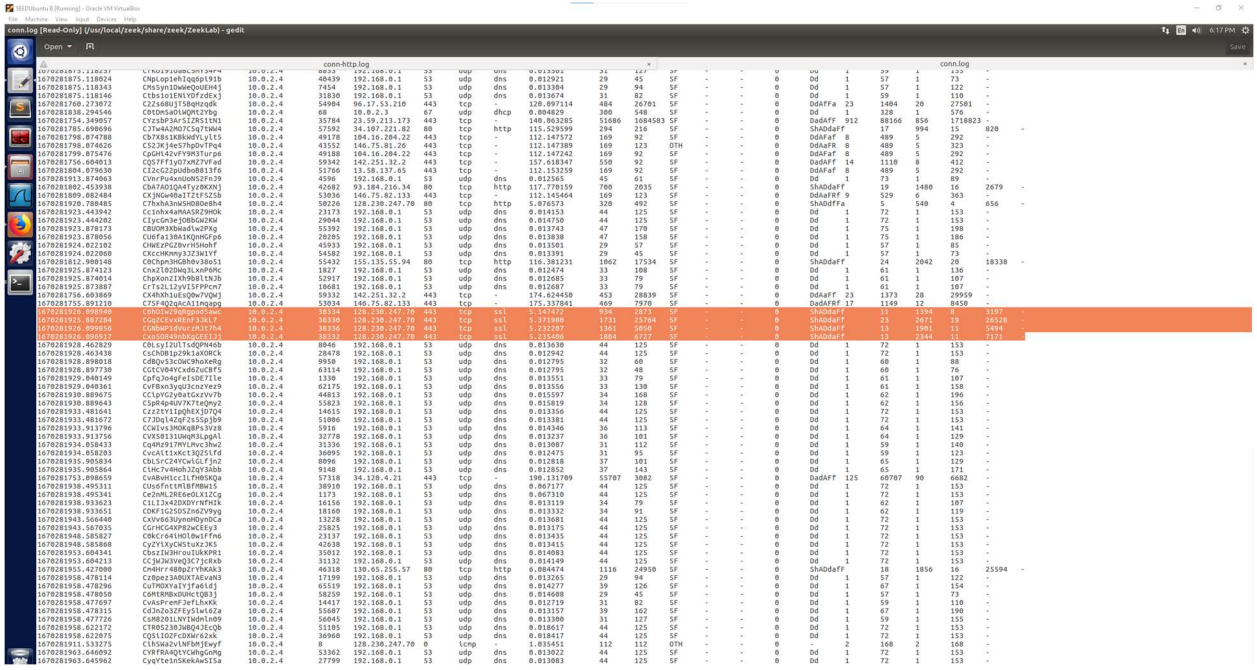
As pictured above, in the conn.log, the highlighted connection is a http connection, the following screenshots also show more http connections that were logged during the website browsing.

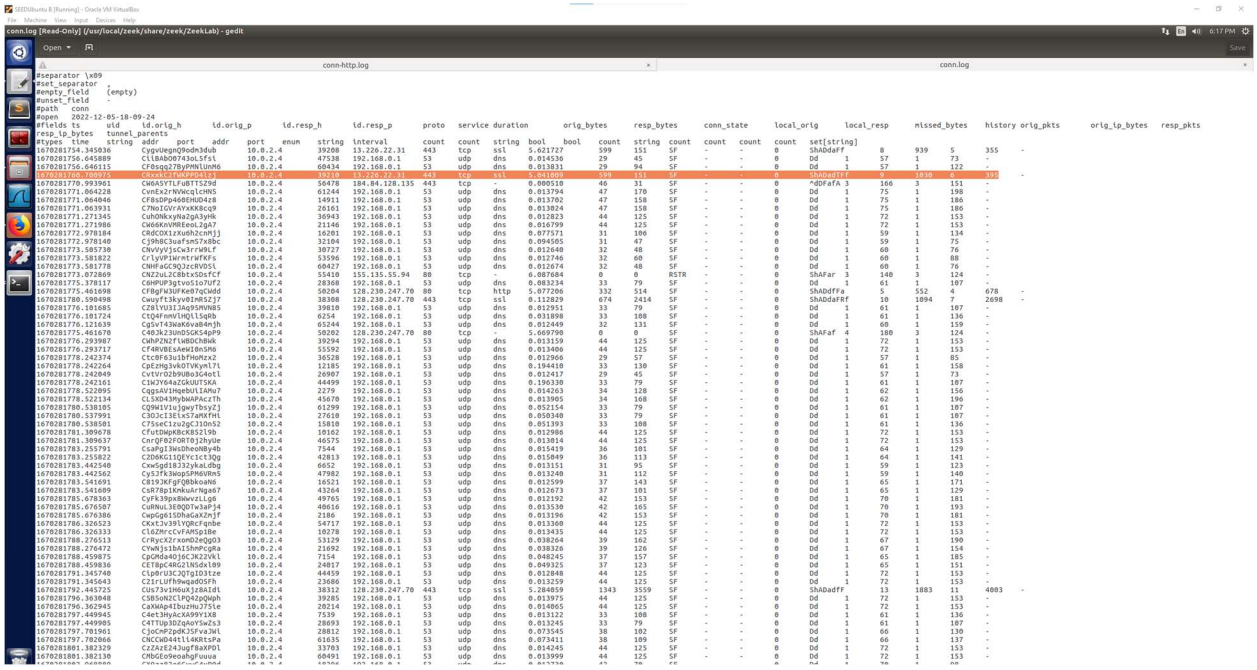



conn.log [Read-Only] (/usr/local/zeek/share/zeek/ZeekLab) - gedit
conn-http.log
conn.log

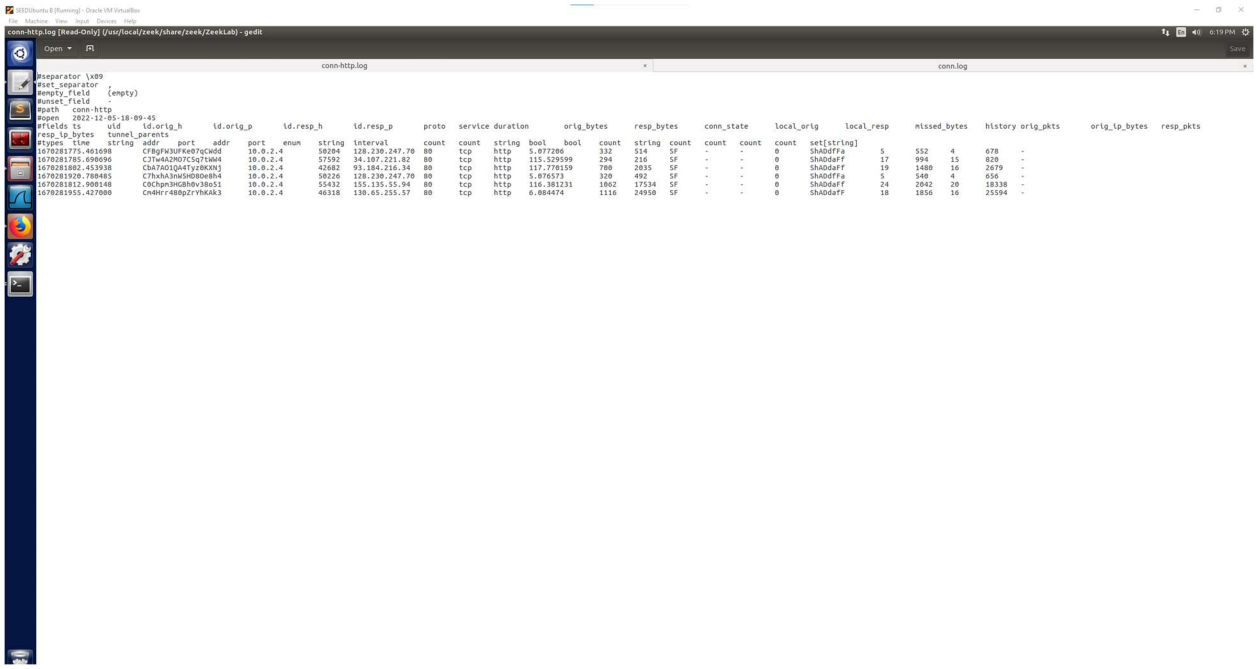

conn.log [Read-Only] (/usr/local/zeek/share/zeek/ZeekLab) - gedit
conn-http.log
conn.log

There are also some ssl connections logged which would again be connections to https websites. As pictured below:





These connections that are highlighted all provide the information previously discussed in the lab task with the UpdatedConn.log, this time there is a distinction shown between the http connections and https connections. When browsing through the conn-http.log, only the http connections are logged and shown.

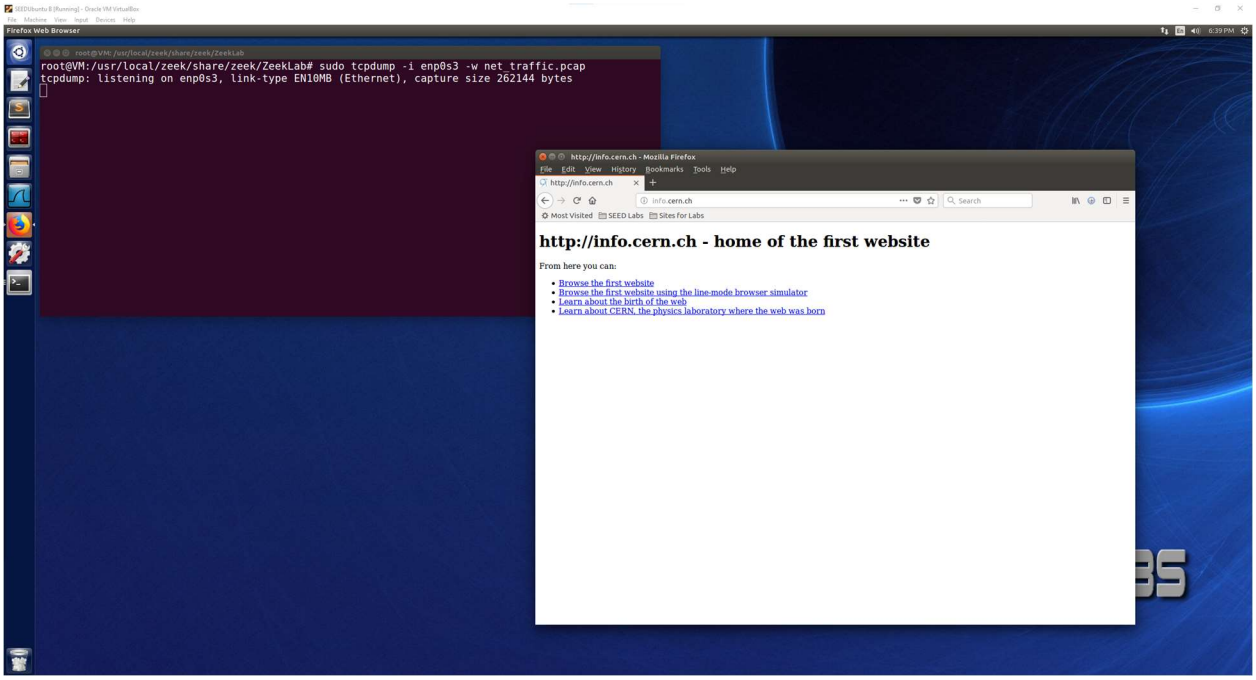


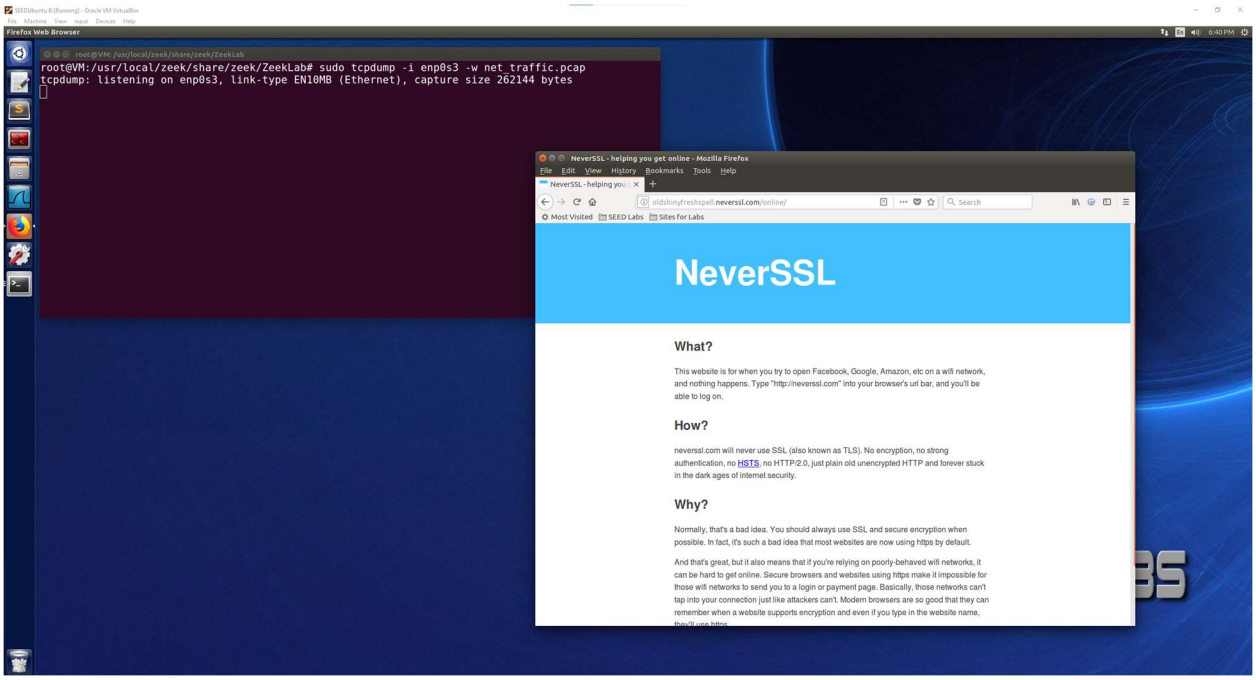
All of these connections shown are http connections from my browsing and show the IP addresses of the websites that were browsed with an http connection. This is an important example of using a Zeek script to gather only the intended connecitons that you want to analyze. The conn.log shows all connections and would be much more work to find specific connections if the script was running for a long duration of time. Whereas, you could run a script to collect only the connections you are curious about in a much cleaner log file. These connections show the source and destination ports as well as the IP addresses for both systems involved, all these connections were using the source port 80. These connections were also labeled as SF which means the were normally established and terminated.

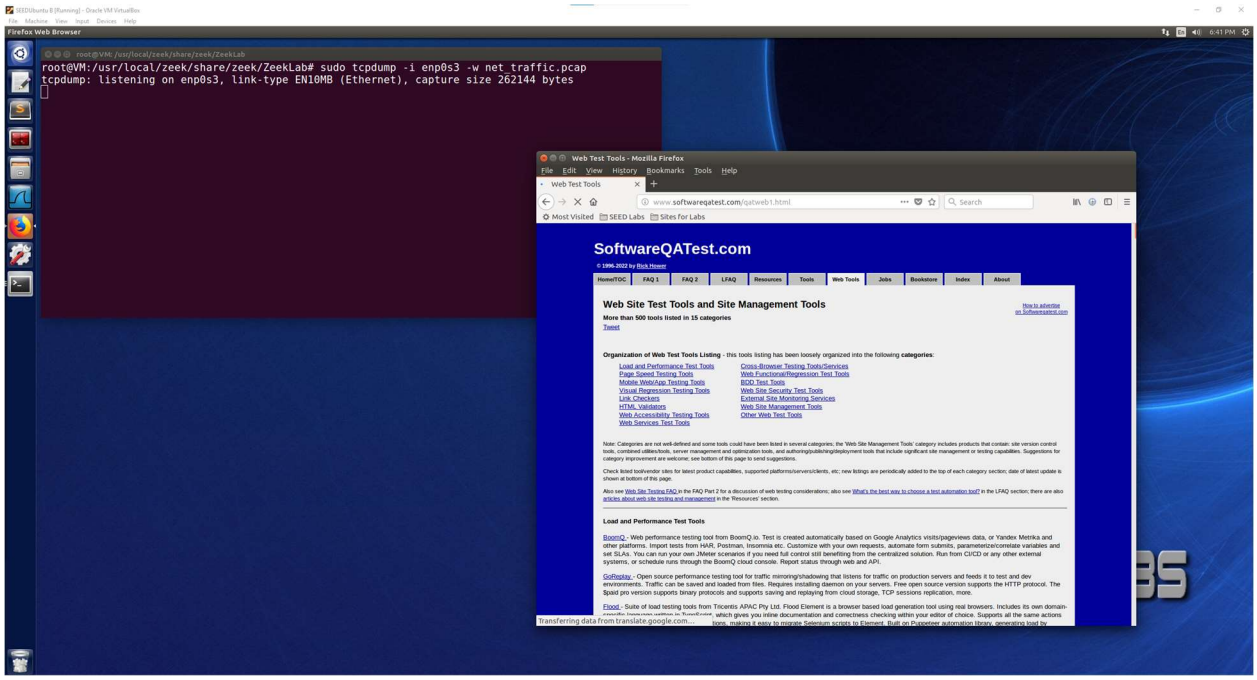
By using the different logs we were able to separate out the http traffic into a cleaner log file rather than searching through all of the connections to find a few http connections. This would save time when looking for a specific connection type like http vs https to see what IP addresses were browsed to that were http.

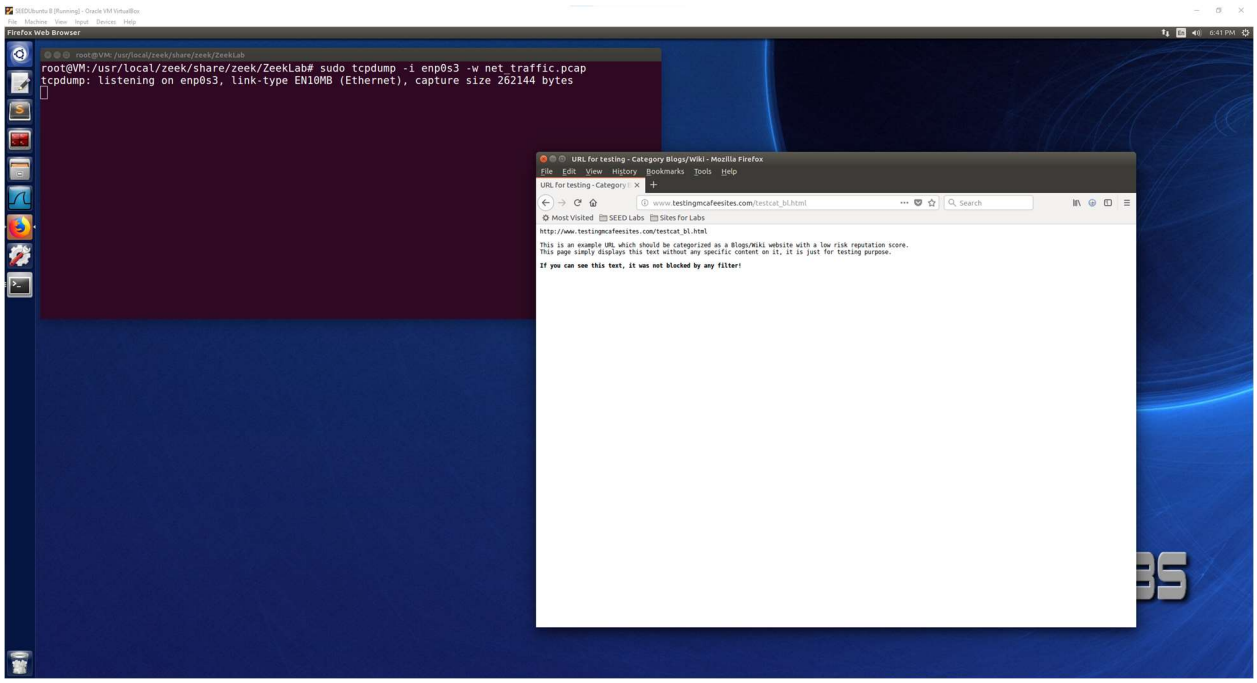
Zeek Signatures

http websites browsed:

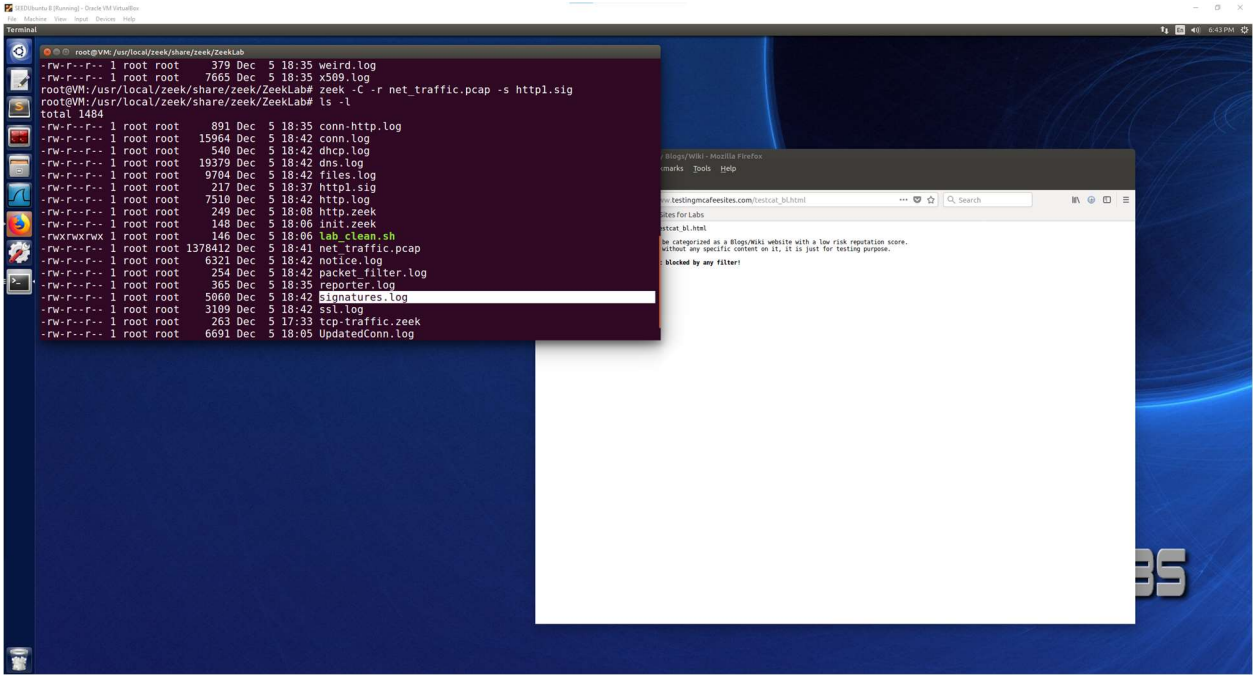





SEEDUbuntu 8 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Firefox Web Browser
6:41 PM
root@VM: /usr/local/zeek/share/zeek/ZeekLab
root@VM:/usr/local/zeek/share/zeek/ZeekLab# sudo tcpdump -i enp0s3 -w net_traffic.pcap
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
Web Test Tools - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Web Test Tools
www.softwareqatest.com/qatweb1.html
Search
Most Visited SEED Labs Sites for Labs
SoftwareQATest.com
© 1996-2022 by Rick Hower
HomeTOC FAQ 1 FAQ 2 LFAQ Resources Tools Web Tools Jobs Bookstore Index About
Web Site Test Tools and Site Management Tools
More than 500 tools listed in 15 categories
Tweet
How to advertise on Software QA/Test .com
Organization of Web Test Tools Listing - this tools listing has been loosely organized into the following categories:
Load and Performance Test Tools
Page Speed Testing Tools
Mobile Web/App Testing Tools
Visual Regression Testing Tools
Link Checkers
HTML Validators
Web Accessibility Testing Tools
Web Services Test Tools
Cross-Browser Testing Tools/Services
Web Functional/Regression Test Tools
BDD Test Tools
Web Site Security Test Tools
External Site Monitoring Services
Web Site Management Tools
Other Web Test Tools
Note: Categories are not well-defined and some tools could have been listed in several categories; the 'Web Site Management Tools' category includes products that contain: site version control tools, combined utilities/tools, server management and optimization tools, and authoring/publishing/deployment tools that include significant site management or testing capabilities. Suggestions for category improvement are welcome; see bottom of this page to send suggestions.
Check listed tool/vendor sites for latest product capabilities, supported platforms/servers/clients, etc; new listings are periodically added to the top of each category section; date of latest update is shown at bottom of this page.
Also see Web Site Testing FAQ in the FAQ Part 2 for a discussion of web testing considerations; also see What's the best way to choose a test automation tool? in the LFAQ section; there are also articles about web site testing and management in the 'Resources' section.
Load and Performance Test Tools
BoomQ - Web performance testing tool from BoomQ.io. Test is created automatically based on Google Analytics visits/pageviews data, or Yandex Metrika and other platforms. Import tests from HAR, Postman, Insomnia etc. Customize with your own requests, automate form submits, parameterize/correlate variables and set SLAs. You can run your own JMeter scenarios if you need full control still benefiting from the centralized solution. Run from CI/CD or any other external systems, or schedule runs through the BoomQ cloud console. Report status through web and API.
GoReplay - Open source performance testing tool for traffic mirroring/shadowing that listens for traffic on production servers and feeds it to test and dev environments. Traffic can be saved and loaded from files. Requires installing daemon on your servers. Free open source version supports the HTTP protocol. The Pro and pro version supports binary protocols and supports saving and replaying from cloud storage, TCP sessions replication, more.
Flood - Suite of load testing tools from Tricentis APAC Pty Ltd. Flood Element is a browser based load generation tool using real browsers. Includes its own domain-specific language written in TypeScript, which gives you inline documentation and correctness checking within your editor of choice. Supports all the same actions tions, making it easy to migrate Selenium scripts to Element. Built on Puppeteer automation library, generating load by
Transferring data from translate.google.com...


SEEDUbuntu 8 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Firefox Web Browser
6:41 PM
root@VM: /usr/local/zeek/share/zeek/ZeekLab
root@VM:/usr/local/zeek/share/zeek/ZeekLab# sudo tcpdump -i enp0s3 -w net_traffic.pcap
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
URL for testing - Category Blogs/Wiki - Mozilla Firefox
File Edit View History Bookmarks Tools Help
URL for testing - Category
www.testingmcafeesites.com/testcat_bl.html
Search
Most Visited SEED Labs Sites for Labs
http://www.testingmcafeesites.com/testcat_bl.html
This is an example URL which should be categorized as a Blogs/Wiki website with a low risk reputation score.
This page simply displays this text without any specific content on it, it is just for testing purpose.
If you can see this text, it was not blocked by any filter!

Showing signatures.log file was generated using the command given:

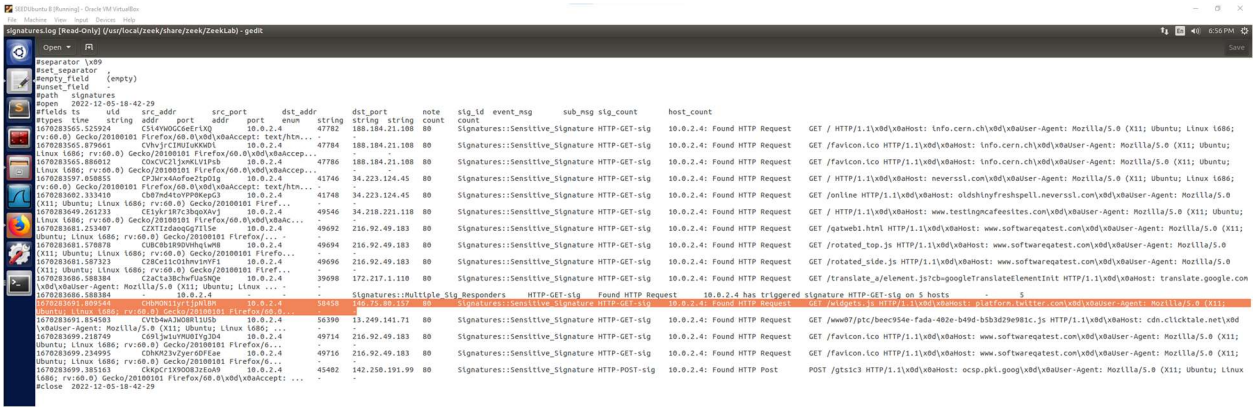


Contents of signatures.log file:

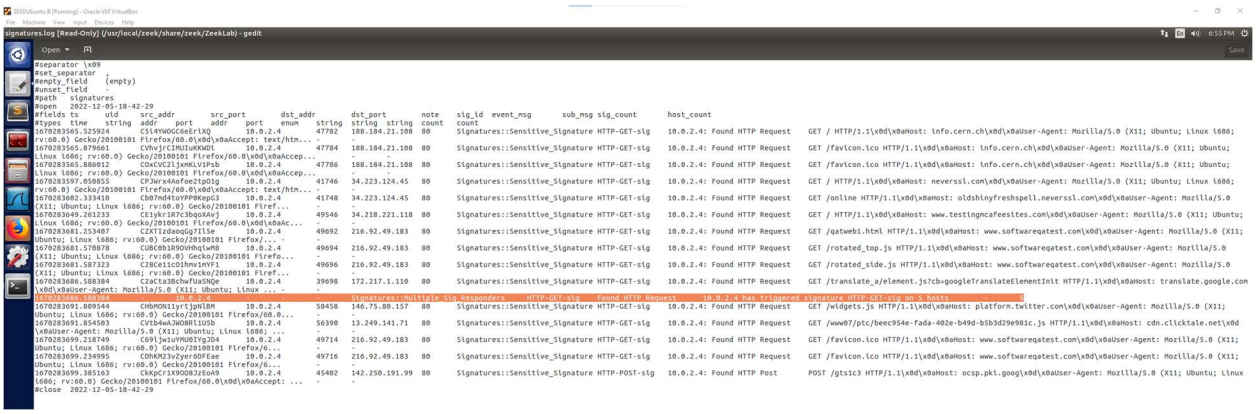
The screenshot shows the logged information about the http websites that were visited in the browser while the zeek script was running. The logged information fits the http1.sig script that was also ran where we wanted to filter out port 80, as seen in the screenshot above all of the connections logged have port 80. The event logged from our script is also seen in the log as “Found HTTP Request” meaning that an http tcp port 80 event was found and then logged.

The connections logged fit with the websites that I had browsed to when looking for http connections. Such as when browsing to softwareqatest.com, which is shown in the log to have an IP address of 216.92.49.183 and is connected to through port 80.

An interesting observation that I noticed in the log was attributed to a twitter widget on one of the pages which produced its own line in the log file despite not being directly navigated to as the webpage that it was located on.



There also was a line that is highlighted below that I thought was interesting because it was the only one that did not present the same information as the other http websites that were navigated to. This one logged as having multiple sig responders and trigged the HTTP-GET event on 5 hosts. I am not sure if that means that it was a printout on the status of logging and that the following 5 logged connections are attributed to that line.



The UID are also still shown with these connections so I believe the user would be able to look in the other log files searching for the UID of one of these connections to find any other additional information on that connection.

The signatures.log file shows the websites that were visited such as softwareqatest.com, testingmcafeesites.com, info.cern.ch, and oldshinyfreshpell.neverssl.com. All of these visited websites were logged amongst other traffic, and then using the signature file we created we were able to process the net_traffic.pcap file to create the sigantures.log file. This log file now contain the http connections that used port 80 and prints out the event of Found HTTP Request. This log contains the source port, destination port(80), source IP address, and destination IP address. By using this signature file we were able to separate out the specific http signature requests from the other traffic that was logged by Zeek.

One last interesting aspect that I noted about the signatures.log file was that the last logged line had Found HTTP Post as the even rather than the request and the IP address belongs to https://google.com.

This is different than the other logged lines being the only post that was logged and since google is an https I would not have expected it to be logged in with the http connections.