Intro to Wireshark

Slides by Brian Johannesmeyer
Presented by Haoyang Fan
w/ some additions

Today’s agenda

- Go through assignment 5
- Introduce Wireshark
- Go through Cannon.py
- Q & A for assignment 5
- Q & A for assignment 4

Reminder

- Assignment 4 due tonight (05/24/2019) at 10pm, with total 8 challenges to solve
- Assignment 5 will likely to be released on tonight as well
- You will be provided with a new VM image for this assignment and this image will contains the starter code so that you don’t have to scp/sftp starter code into VM

Leading up to the Great Cannon of China

Great Firewall of China - monitors traffic entering and exiting China, then disrupts prohibited content and connections, i.e. with a denial of service (DoS)

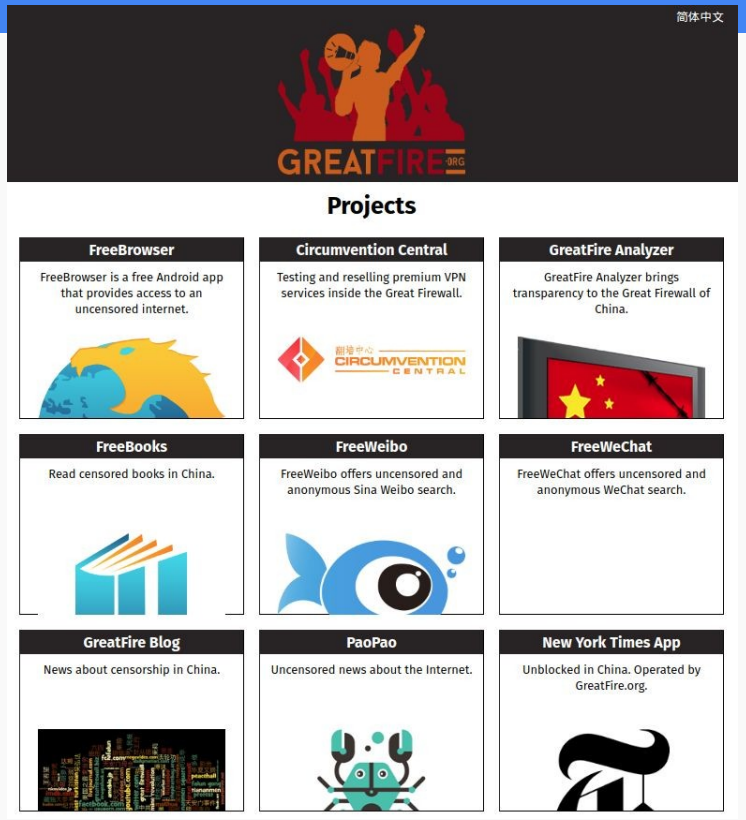
However, citizens started finding ways around the Great Firewall...

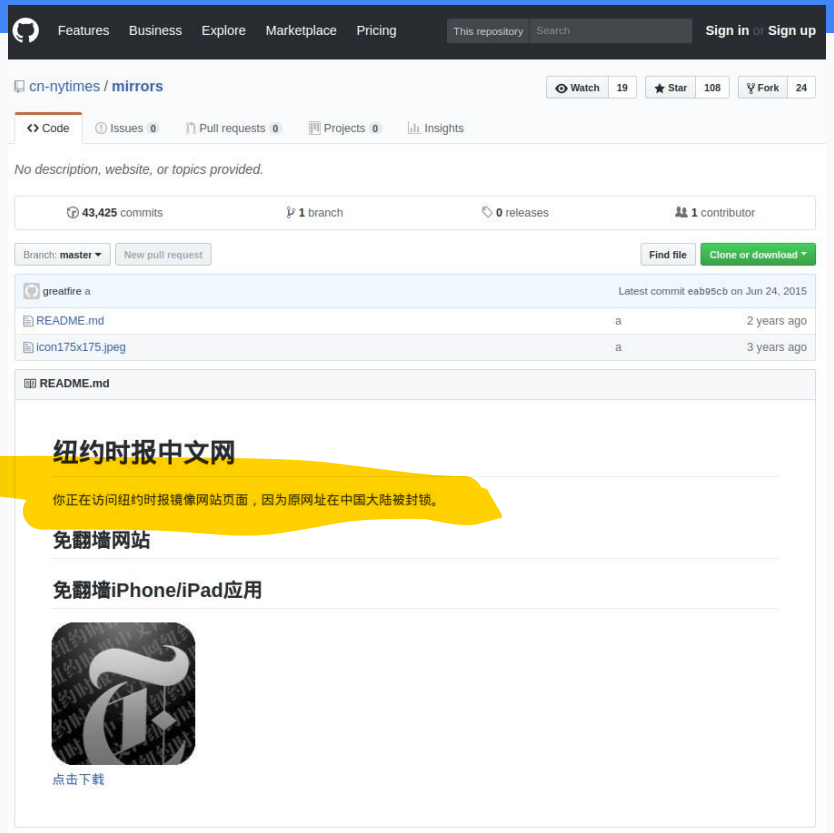
GreatFire.org - a non-profit organization that monitors the status of websites censored by the Great Firewall of China, and helps Chinese Internet users circumvent the censorship and blockage of websites

However, the Chinese government wanted to stop GreatFire.org...

Great Cannon of China - an attack tool that is used to launch distributed denial-of-service (DDoS) attacks on websites by intercepting massive amounts of web traffic and redirecting them to targeted websites

GreatFire.org





Great Cannon of China

In March 2015:

- The Great Cannon targeted a distributed denial-of-service attack on GreatFire.org's Amazon CloudFront services, which were used to make blocked websites accessible in China.
- Then, the Great Cannon targeted GitHub pages run by GreatFire.org.

How did it do this?

- By injecting malicious JavaScript into unencrypted HTTP connections
- E.g. an IFrame pointing to GreatCannon.org
  - An IFrame is an HTML document embedded inside another HTML document on a website.
  - IFrames are commonly used to embed advertisements into webpages.
- However, at this scale, GreatFire is hit with multiple requests from all sorts of IP addresses, causing a DoS attack

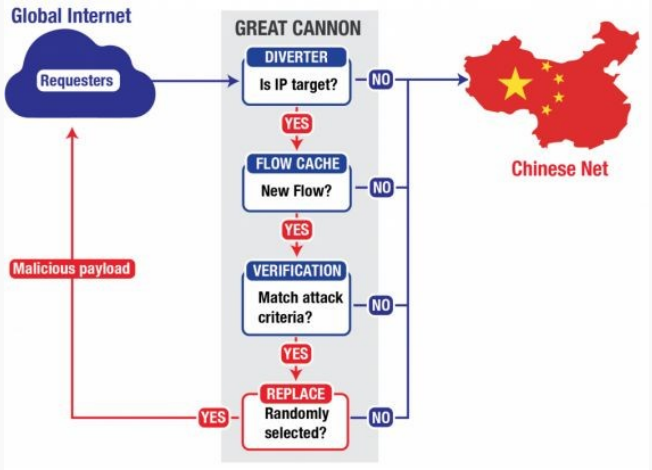
Assignment: Intro

You are to implement an in-path network attack similar to the Great Cannon.

Mininet: implements the hostand the switch

POX: implements the controller

```
[host] ------ [switch] ----- [Internet]
                    \
                     \_______ [controller]
```



The switch passes all traffic to the controller, which tells it what to do with it.

The `manipulate_packet` function in `/home/mininet/pox/pox/triton/cannon.py` implements this functionality. Meaning it decides if a packet should be allowed to pass through unmodified, modified, or dropped.

cannon.py

- Located at: /home/mininet/pox/pox/triton/cannon.py
- This should be the only code you need to write for this assignment
- Starter code is given, I will talk about a it later

Assignment: Getting started

In VBox's mininet terminal (not via SSH), run **`startx`** to start the graphical environment:

Some terminologies

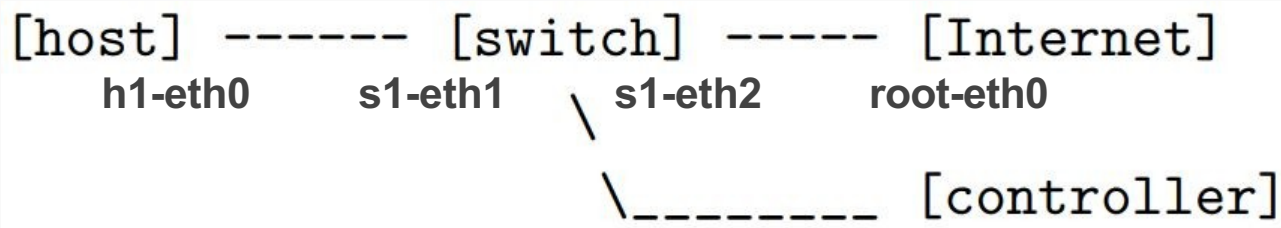
- Mininet: a realistic virtual network, running real kernel, switch and application code
  - http://mininet.org/
  - Recommended to learn: http://mininet.org/walkthrough/#interact-with-hosts-and-switches

- POX: implementation of OpenFlow protocol controller
  - open-source: https://github.com/noxrepo/pox
  - I personally find that spending some time taking a loot at source code of POX is very helpful for this assignment. Since you will need to write code on POX framework in cannon.py

- Wireshark: a popular network protocol analyzer
  - https://www.wireshark.org/

Assignment: Topology

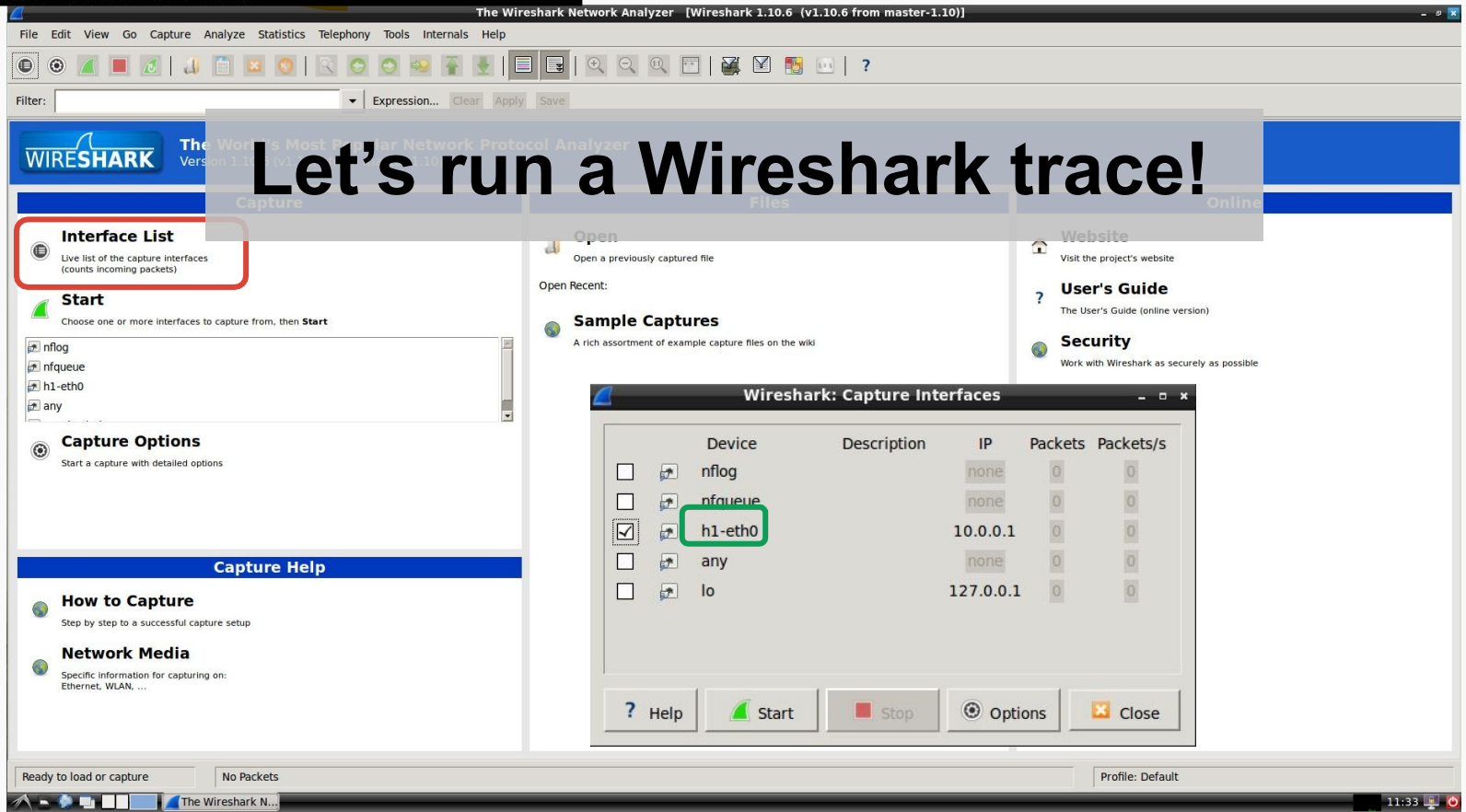
```
mininet@mininet-vm:~$ sudo /home/mininet/mininet/examples/cannon_network.py
*** Configuring hosts
h1
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Hosts are running and should have internet connectivity
*** Type 'exit' or control-D to shut down network
*** Starting CLI:
mininet> net
h1 h1-eth0:s1-eth1
s1 lo:  s1-eth1:h1-eth0 s1-eth2:root-eth0
c0
```



```
mininet> net
h1 h1-eth0:s1-eth1
s1 lo:  s1-eth1:h1-eth0 s1-eth2:root-eth0
c0
mininet> h1 wireshark &
```

```
[host] ------ [switch] ----- [Internet]
   h1-eth0     s1-eth1  \s1-eth2     root-eth0
                         \________ [controller]
```

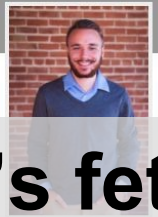
Let's run a Wireshark trace!



Let's fetch this document (index.html) and see what it looks like in Wireshark!

Brian Johannesmeyer's Home Page - Google Chrome

Not secure | www.sysnet.ucsd.edu/~bjohanne/



Brian Johannesmeyer

PhD Student
Computer Science and Engineering
University of California, San Diego
Office: EBU3B 3144

Email:
bjohanne@cs.ucsd.edu

Research

My research interests are primarily in syste

Publications

- S. Cauligi, G. Soeller, F. Brow, B. Johannesmeyer, Fac
  A Flexible Constant-Time Programming Language. IEEE SecDev 2017.
- Z. Yang, B. Johannesmeyer, A.T. Olesen, S. Lerner, and K. Levchenko. Dead Store Elimination (Still) Considered
- B. Yadegari, B. Johannesme
  Deobfuscation of Executable
- J. Qiu, B. Yadegari, B. Johan
  Checksumming Defenses in
- J. Qiu, B. Yadegari, B. Johan
  Dynamic Anti-Analysis Defen

Projects

Current

- Aerosec: The Aviation Cyber
- CESR: Center for Evidence-

Past

- Lynx: Analysis and Reverse

About me

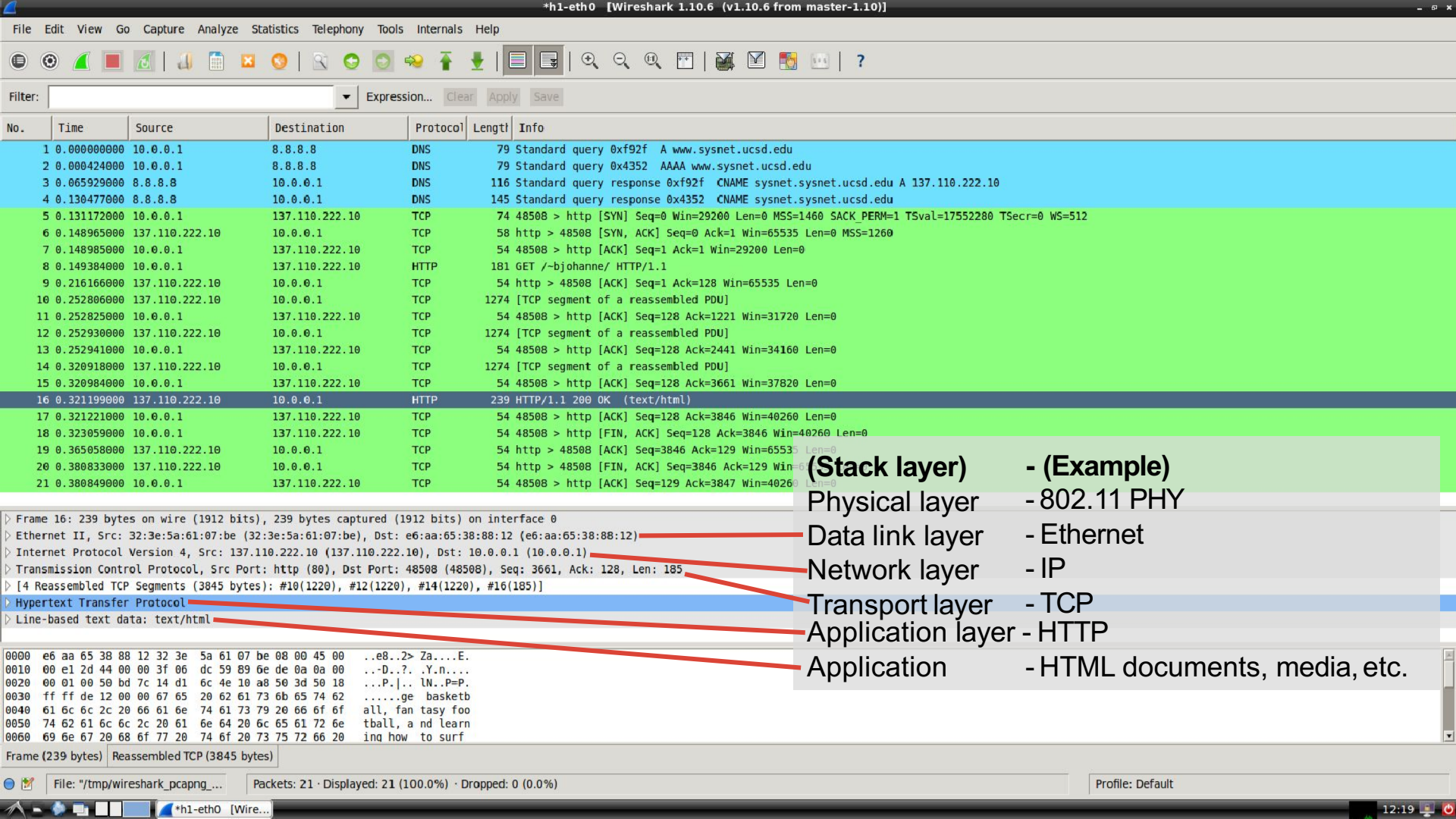
I am a Computer Science PhD student in the Systems and Networking Group and Security Group at the University of California, San Diego. I received undergrad degrees in Computer Science and Electrical and Computer Engineering from the University of Arizona. Besides Computer Science, I enjoy hiking, camping, photography, college basketball, fantasy football, and learning how to surf (now that I'm in San Diego).

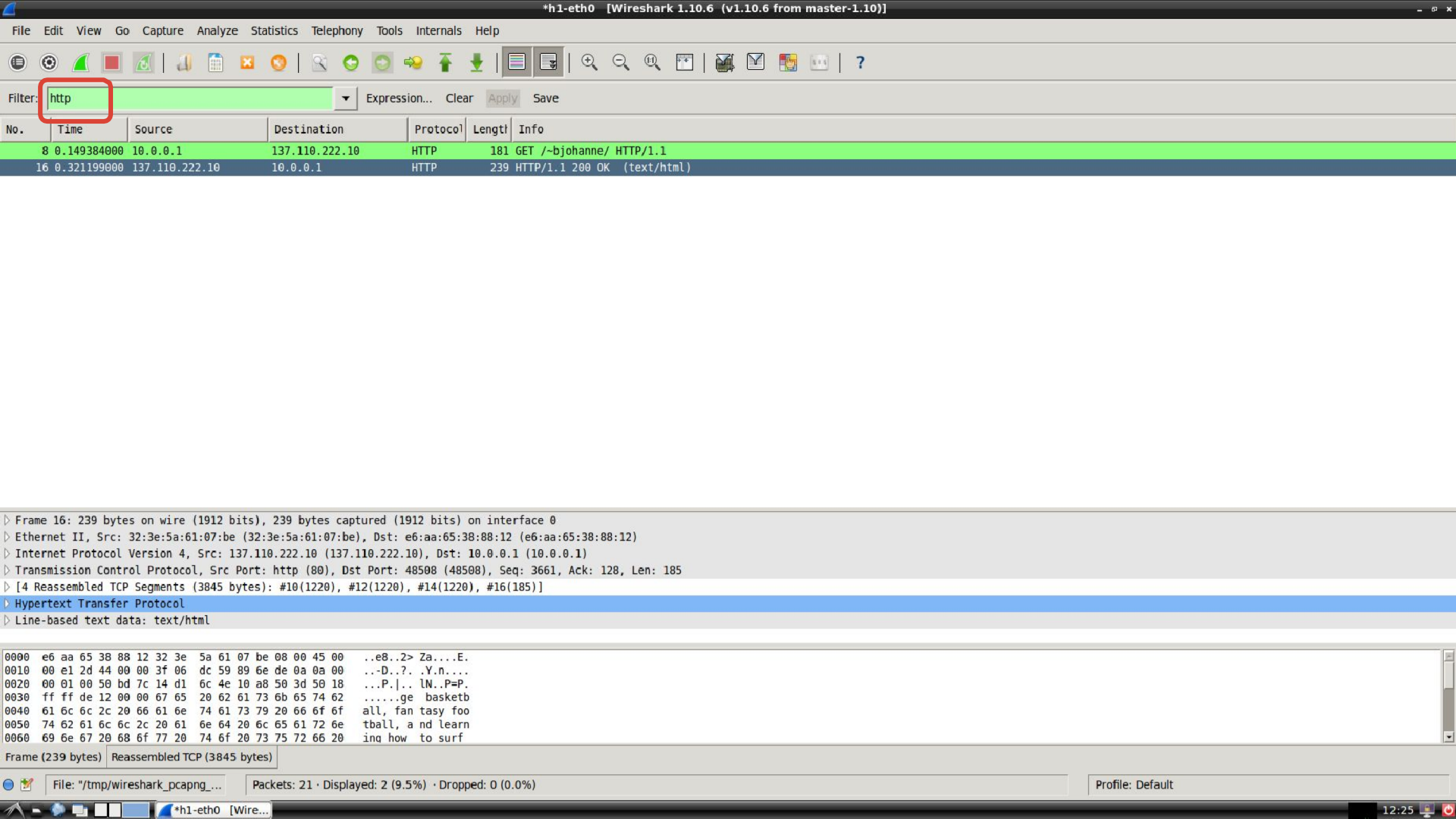
```
mininet> h1 su mininet -c "wget www.sysnet.ucsd.edu/~bjohanne/"
--2017-12-01 12:09:45--  http://www.sysnet.ucsd.edu/~bjohanne/
Resolving www.sysnet.ucsd.edu (www.sysnet.ucsd.edu)... 137.110.222.10
Connecting to www.sysnet.ucsd.edu (www.sysnet.ucsd.edu)|137.110.222.10|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3531 (3.4K) [text/html]
Saving to: 'index.html.12'

100%[======================================>] 3,531       --.-K/s   in 0.07s

2017-12-01 12:09:45 (46.3 KB/s) - 'index.html.12' saved [3531/3531]
```

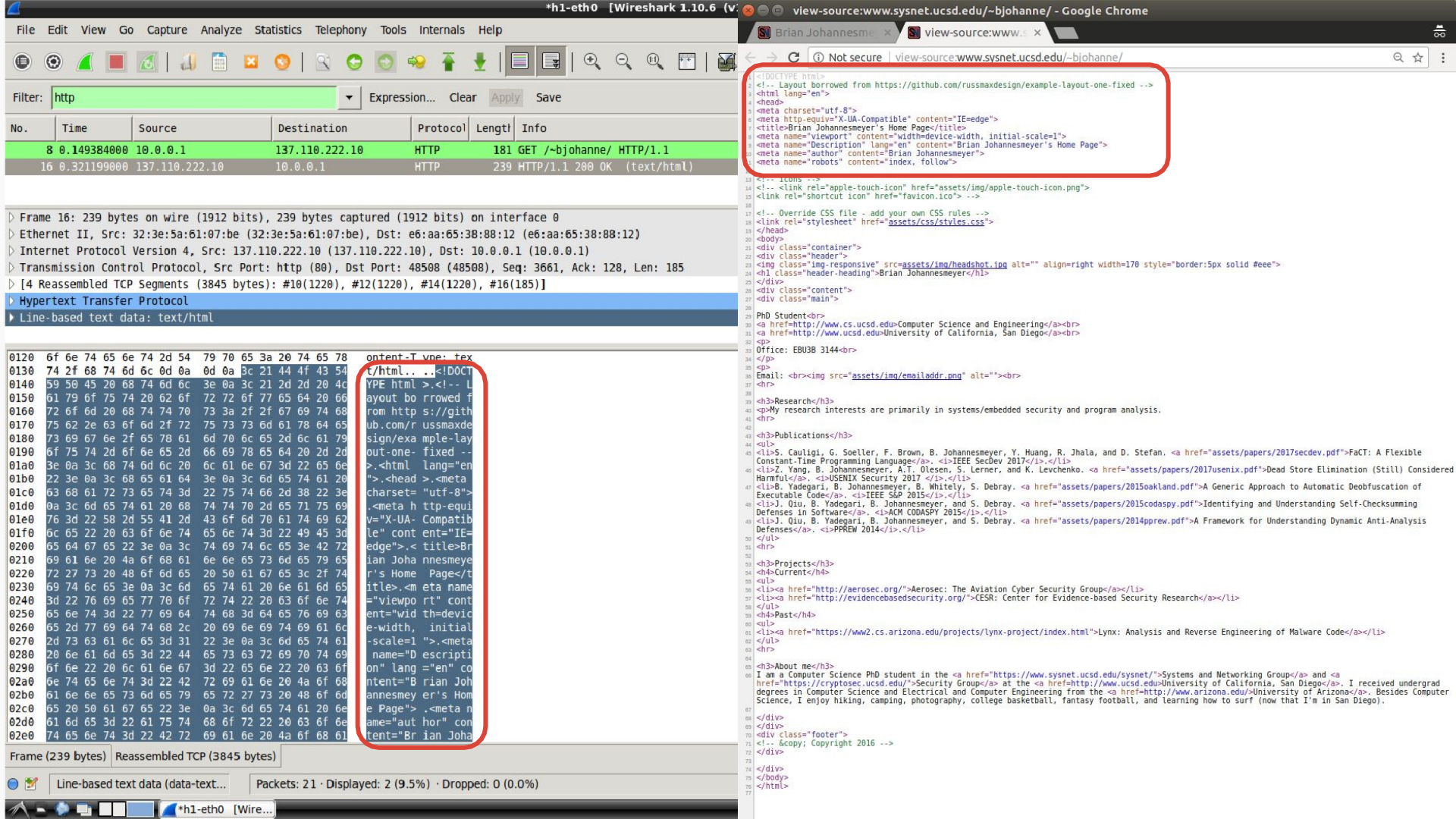
view-source:www.sysnet.ucsd.edu/~bjohanne/ - Google Chrome

Not secure | view-source:www.sysnet.ucsd.edu/~bjohanne/

```
<!DOCTYPE html>
<!-- Layout borrowed from https://github.com/russmaxdesign/example-layout-one-fixed -->
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<title>Brian Johannesmeyer's Home Page</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta name="Description" lang="en" content="Brian Johannesmeyer's Home Page">
<meta name="author" content="Brian Johannesmeyer">
<meta name="robots" content="index, follow">

<!-- icons -->
<!-- <link rel="apple-touch-icon" href="assets/img/apple-touch-icon.png">
<link rel="shortcut icon" href="favicon.ico"> -->

<!-- Bootstrap -->
<link rel="stylesheet" href="assets/bootstrap/css/bootstrap.min.css">
<link rel="stylesheet" href="assets/bootstrap/css/bootstrap-theme.min.css">
<link rel="stylesheet" href="assets/css/main.css">

<!-- Modernizr -->
<script src="assets/js/modernizr-2.6.2.min.js"></script>
</head>
<body>
<div class="container">
<div class="header">
<img class="img-responsive" src=assets/img/headshot.jpg alt="" align=right width=170 style="border:5px solid #eee">
<h1 class="header-heading">Brian Johannesmeyer</h1>
</div>
<br>
<a href=http://www.cs.ucsd.edu>Computer Science and Engineering</a><br>
<a href=http://www.ucsd.edu>University of California, San Diego</a><br>
<br>
Office: EBU3B 3144<br>
<br>
Email:<br>
<b>bjohanne@cs.ucsd.edu</b>
<br>
<br>
<h3>Research</h3>
<p>My research interests are primarily in systems/embedded security and program analysis.
<br>
<br>
<h3>Publications</h3>
<ul>
<li>S. Cauligi, G. Soeller, F. Brown, B. Johannesmeyer, Fabian Gorski, Ryan Kastner, Deian Stefan. <a href="assets/papers/2017secdev.pdf">FaCT: A Flexible Constant-Time Programming Language</a>. IEEE SecDev 2017.</li>
<li>Z. Yang, B. Johannesmeyer, A.T. Olesen, S. Lerner, and K. Levchenko. <a href="assets/papers/2017usenix.pdf">Dead Store Elimination (Still) Considered Harmful</a>. USENIX Security 2017.</li>
<li>B. Yadegari, B. Johannesmeyer, B. Whitely, and S. Debray. <a href="assets/papers/2015and.pdf">A Generic Approach to Automatic Deobfuscation of Executable Code</a>. IEEE S&P 2015.</li>
<li>J. Qiu, B. Yadegari, B. Johannesmeyer, S. Debray, and X. Su. <a href="assets/papers/2016spy.pdf">Identifying and Understanding Self-Checksumming Defenses in Software</a>. ACM CODASPY 2016.</li>
<li>J. Qiu, B. Yadegari, B. Johannesmeyer, S. Debray, and X. Su. <a href="assets/papers/2016dyn.pdf">A Framework for Understanding Dynamic Anti-Analysis Defenses</a>. DIMVA 2016.</li>
</ul>
<br>
<h3>Projects</h3>
Current
<ul>
<li><a href="http://aerosec.ucsd.edu">Aerosec: The Aviation Cyber Research</a></li>
<li><a href="http://cesr.ucsd.edu">CESR: Center for Evidence-based Security Research</a></li>
</ul>
Past
<ul>
<li><a href="http://lynx.ucsd.edu">Lynx: Analysis and Reverse Engineering of Malware Code</a></li>
</ul>
<br>
<hr>

<h3>About me</h3>
I am a Computer Science PhD student in the <a href="https://www.sysnet.ucsd.edu/sysnet/">Systems and Networking Group</a> and <a href="https://cryptosec.ucsd.edu/">Security Group</a> at the <a href="http://www.ucsd.edu">University of California, San Diego</a>. I received undergrad degrees in Computer Science and Electrical and Computer Engineering from the <a href="http://www.arizona.edu/">University of Arizona</a>. Besides Computer Science, I enjoy hiking, camping, photography, college basketball, fantasy football, and learning how to surf (now that I'm in San Diego).

</div>
</div>
<div class="footer">
<!-- &copy; Copyright 2016 -->
</div>

</div>
</body>
</html>
```


*h1-eth0 [Wireshark 1.10.6 (v1.10.6 from master-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: Expression... Clear Apply Save
No. Time Source Destination Protocol Length Info
1 0.000000000 10.0.0.1 8.8.8.8 DNS 79 Standard query 0xf92f A www.sysnet.ucsd.edu
2 0.000424000 10.0.0.1 8.8.8.8 DNS 79 Standard query 0x4352 AAAA www.sysnet.ucsd.edu
3 0.065929000 8.8.8.8 10.0.0.1 DNS 116 Standard query response 0xf92f CNAME sysnet.sysnet.ucsd.edu A 137.110.222.10
4 0.130477000 8.8.8.8 10.0.0.1 DNS 145 Standard query response 0x4352 CNAME sysnet.sysnet.ucsd.edu
5 0.131172000 10.0.0.1 137.110.222.10 TCP 74 48508 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=17552280 TSecr=0 WS=512
6 0.148965000 137.110.222.10 10.0.0.1 TCP 58 http > 48508 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1260
7 0.148985000 10.0.0.1 137.110.222.10 TCP 54 48508 > http [ACK] Seq=1 Ack=1 Win=29200 Len=0
8 0.149384000 10.0.0.1 137.110.222.10 HTTP 181 GET /~bjohanne/ HTTP/1.1
9 0.216166000 137.110.222.10 10.0.0.1 TCP 54 http > 48508 [ACK] Seq=1 Ack=128 Win=65535 Len=0
10 0.252806000 137.110.222.10 10.0.0.1 TCP 1274 [TCP segment of a reassembled PDU]
11 0.252825000 10.0.0.1 137.110.222.10 TCP 54 48508 > http [ACK] Seq=128 Ack=1221 Win=31720 Len=0
12 0.252930000 137.110.222.10 10.0.0.1 TCP 1274 [TCP segment of a reassembled PDU]
13 0.252941000 10.0.0.1 137.110.222.10 TCP 54 48508 > http [ACK] Seq=128 Ack=2441 Win=34160 Len=0
14 0.320918000 137.110.222.10 10.0.0.1 TCP 1274 [TCP segment of a reassembled PDU]
15 0.320984000 10.0.0.1 137.110.222.10 TCP 54 48508 > http [ACK] Seq=128 Ack=3661 Win=37820 Len=0
16 0.321199000 137.110.222.10 10.0.0.1 HTTP 239 HTTP/1.1 200 OK (text/html)
17 0.321221000 10.0.0.1 137.110.222.10 TCP 54 48508 > http [ACK] Seq=128 Ack=3846 Win=40260 Len=0
18 0.323059000 10.0.0.1 137.110.222.10 TCP 54 48508 > http [FIN, ACK] Seq=128 Ack=3846 Win=40260 Len=0
19 0.365058000 137.110.222.10 10.0.0.1 TCP 54 http > 48508 [ACK] Seq=3846 Ack=129 Win=65535 Len=0
20 0.380833000 137.110.222.10 10.0.0.1 TCP 54 http > 48508 [FIN, ACK] Seq=3846 Ack=129 Win=65535 Len=0
21 0.380849000 10.0.0.1 137.110.222.10 TCP 54 48508 > http [ACK] Seq=129 Ack=3847 Win=40260 Len=0
Frame 16: 239 bytes on wire (1912 bits), 239 bytes captured (1912 bits) on interface 0
Ethernet II, Src: 32:3e:5a:61:07:be (32:3e:5a:61:07:be), Dst: e6:aa:65:38:88:12 (e6:aa:65:38:88:12)
Internet Protocol Version 4, Src: 137.110.222.10 (137.110.222.10), Dst: 10.0.0.1 (10.0.0.1)
Transmission Control Protocol, Src Port: http (80), Dst Port: 48508 (48508), Seq: 3661, Ack: 128, Len: 185
[4 Reassembled TCP Segments (3845 bytes): #10(1220), #12(1220), #14(1220), #16(185)]
Hypertext Transfer Protocol
Line-based text data: text/html
0000 e6 aa 65 38 88 12 32 3e 5a 61 07 be 08 00 45 00 ..e8..2> Za....E.
0010 00 e1 2d 44 00 00 3f 06 dc 59 89 6e de 0a 0a 00 ..-D..?. .Y.n....
0020 00 01 00 50 bd 7c 14 d1 6c 4e 10 a8 50 3d 50 18 ...P.|.. lN..P=P.
0030 ff ff de 12 00 00 67 65 20 62 61 73 6b 65 74 62 ......ge basketb
0040 61 6c 6c 2c 20 66 61 6e 74 61 73 79 20 66 6f 6f all, fan tasy foo
0050 74 62 61 6c 6c 2c 20 61 6e 64 20 6c 65 61 72 6e tball, a nd learn
0060 69 6e 67 20 68 6f 77 20 74 6f 20 73 75 72 66 20 ing how to surf
Frame (239 bytes) Reassembled TCP (3845 bytes)
File: "/tmp/wireshark_pcapng_... Packets: 21 · Displayed: 21 (100.0%) · Dropped: 0 (0.0%)
Profile: Default
*h1-eth0 [Wire...
12:19
(Stack layer) - (Example)
Physical layer - 802.11 PHY
Data link layer - Ethernet
Network layer - IP
Transport layer - TCP
Application layer - HTTP
Application - HTML documents, media, etc.


*h1-eth0 [Wireshark 1.10.6 (v1.10.6 from master-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: http
Expression... Clear Apply Save
No. Time Source Destination Protocol Lengt Info
8 0.149384000 10.0.0.1 137.110.222.10 HTTP 181 GET /~bjohanne/ HTTP/1.1
16 0.321199000 137.110.222.10 10.0.0.1 HTTP 239 HTTP/1.1 200 OK (text/html)
Frame 16: 239 bytes on wire (1912 bits), 239 bytes captured (1912 bits) on interface 0
Ethernet II, Src: 32:3e:5a:61:07:be (32:3e:5a:61:07:be), Dst: e6:aa:65:38:88:12 (e6:aa:65:38:88:12)
Internet Protocol Version 4, Src: 137.110.222.10 (137.110.222.10), Dst: 10.0.0.1 (10.0.0.1)
Transmission Control Protocol, Src Port: http (80), Dst Port: 48508 (48508), Seq: 3661, Ack: 128, Len: 185
[4 Reassembled TCP Segments (3845 bytes): #10(1220), #12(1220), #14(1220), #16(185)]
Hypertext Transfer Protocol
Line-based text data: text/html
0000 e6 aa 65 38 88 12 32 3e 5a 61 07 be 08 00 45 00 ..e8..2> Za....E.
0010 00 e1 2d 44 00 00 3f 06 dc 59 89 6e de 0a 0a 00 ..-D..?. .Y.n....
0020 00 01 00 50 bd 7c 14 d1 6c 4e 10 a8 50 3d 50 18 ...P.|.. lN..P=P.
0030 ff ff de 12 00 00 67 65 20 62 61 73 6b 65 74 62 ......ge basketb
0040 61 6c 6c 2c 20 66 61 6e 74 61 73 79 20 66 6f 6f all, fan tasy foo
0050 74 62 61 6c 6c 2c 20 61 6e 64 20 6c 65 61 72 6e tball, a nd learn
0060 69 6e 67 20 68 6f 77 20 74 6f 20 73 75 72 66 20 ing how to surf
Frame (239 bytes) Reassembled TCP (3845 bytes)
File: "/tmp/wireshark_pcapng_...
Packets: 21 · Displayed: 2 (9.5%) · Dropped: 0 (0.0%)
Profile: Default
*h1-eth0 [Wire...
12:25


*h1-eth0 [Wireshark 1.10.6 (v
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: http
Expression... Clear Apply Save
No. Time Source Destination Protocol Lengtl Info
8 0.149384000 10.0.0.1 137.110.222.10 HTTP 181 GET /~bjohanne/ HTTP/1.1
16 0.321199000 137.110.222.10 10.0.0.1 HTTP 239 HTTP/1.1 200 OK (text/html)
Frame 16: 239 bytes on wire (1912 bits), 239 bytes captured (1912 bits) on interface 0
Ethernet II, Src: 32:3e:5a:61:07:be (32:3e:5a:61:07:be), Dst: e6:aa:65:38:88:12 (e6:aa:65:38:88:12)
Internet Protocol Version 4, Src: 137.110.222.10 (137.110.222.10), Dst: 10.0.0.1 (10.0.0.1)
Transmission Control Protocol, Src Port: http (80), Dst Port: 48508 (48508), Seq: 3661, Ack: 128, Len: 185
[4 Reassembled TCP Segments (3845 bytes): #10(1220), #12(1220), #14(1220), #16(185)]
Hypertext Transfer Protocol
Line-based text data: text/html
0120 6f 6e 74 65 6e 74 2d 54 79 70 65 3a 20 74 65 78 ontent-T ype: tex
0130 74 2f 68 74 6d 6c 0d 0a 0d 0a 3c 21 44 4f 43 54 t/html.. ..<!DOCT
0140 59 50 45 20 68 74 6d 6c 3e 0a 3c 21 2d 2d 20 4c YPE html >.<!-- L
0150 61 79 6f 75 74 20 62 6f 72 72 6f 77 65 64 20 66 ayout bo rrowed f
0160 72 6f 6d 20 68 74 74 70 73 3a 2f 2f 67 69 74 68 rom http s://gith
0170 75 62 2e 63 6f 6d 2f 72 75 73 73 6d 61 78 64 65 ub.com/r ussmaxde
0180 73 69 67 6e 2f 65 78 61 6d 70 6c 65 2d 6c 61 79 sign/exa mple-lay
0190 6f 75 74 2d 6f 6e 65 2d 66 69 78 65 64 20 2d 2d out-one- fixed --
01a0 3e 0a 3c 68 74 6d 6c 20 6c 61 6e 67 3d 22 65 6e >.<html lang="en
01b0 22 3e 0a 3c 68 65 61 64 3e 0a 3c 6d 65 74 61 20 ">.<head >.<meta
01c0 63 68 61 72 73 65 74 3d 22 75 74 66 2d 38 22 3e charset= "utf-8">
01d0 0a 3c 6d 65 74 61 20 68 74 74 70 2d 65 71 75 69 .<meta h ttp-equi
01e0 76 3d 22 58 2d 55 41 2d 43 6f 6d 70 61 74 69 62 v="X-UA- Compatib
01f0 6c 65 22 20 63 6f 6e 74 65 6e 74 3d 22 49 45 3d le" cont ent="IE=
0200 65 64 67 65 22 3e 0a 3c 74 69 74 6c 65 3e 42 72 edge">.< title>Br
0210 69 61 6e 20 4a 6f 68 61 6e 6e 65 73 6d 65 79 65 ian Joha nnesmeye
0220 72 27 73 20 48 6f 6d 65 20 50 61 67 65 3c 2f 74 r's Home Page</t
0230 69 74 6c 65 3e 0a 3c 6d 65 74 61 20 6e 61 6d 65 itle>.<m eta name
0240 3d 22 76 69 65 77 70 6f 72 74 22 20 63 6f 6e 74 ="viewpo rt" cont
0250 65 6e 74 3d 22 77 69 64 74 68 3d 64 65 76 69 63 ent="wid th=devic
0260 65 2d 77 69 64 74 68 2c 20 69 6e 69 74 69 61 6c e-width, initial
0270 2d 73 63 61 6c 65 3d 31 22 3e 0a 3c 6d 65 74 61 -scale=1 ">.<meta
0280 20 6e 61 6d 65 3d 22 44 65 73 63 72 69 70 74 69 name="D escripti
0290 6f 6e 22 20 6c 61 6e 67 3d 22 65 6e 22 20 63 6f on" lang ="en" co
02a0 6e 74 65 6e 74 3d 22 42 72 69 61 6e 20 4a 6f 68 ntent="B rian Joh
02b0 61 6e 6e 65 73 6d 65 79 65 72 27 73 20 48 6f 6d annesmey er's Hom
02c0 65 20 50 61 67 65 22 3e 0a 3c 6d 65 74 61 20 6e e Page"> .<meta n
02d0 61 6d 65 3d 22 61 75 74 68 6f 72 22 20 63 6f 6e ame="aut hor" con
02e0 74 65 6e 74 3d 22 42 72 69 61 6e 20 4a 6f 68 61 tent="Br ian Joha
Frame (239 bytes) Reassembled TCP (3845 bytes)
Line-based text data (data-text...
Packets: 21 · Displayed: 2 (9.5%) · Dropped: 0 (0.0%)
*h1-eth0 [Wire...
view-source:www.sysnet.ucsd.edu/~bjohanne/ - Google Chrome
Brian Johannesme
view-source:www.s
Not secure view-source:www.sysnet.ucsd.edu/~bjohanne/
<!DOCTYPE html>
<!-- Layout borrowed from https://github.com/russmaxdesign/example-layout-one-fixed -->
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<title>Brian Johannesmeyer's Home Page</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta name="Description" lang="en" content="Brian Johannesmeyer's Home Page">
<meta name="author" content="Brian Johannesmeyer">
<meta name="robots" content="index, follow">
<!-- Icons -->
<!-- <link rel="apple-touch-icon" href="assets/img/apple-touch-icon.png">
<link rel="shortcut icon" href="favicon.ico"> -->
<!-- Override CSS file - add your own CSS rules -->
<link rel="stylesheet" href="assets/css/styles.css">
</head>
<body>

<img class="img-responsive" src=assets/img/headshot.jpg alt="" align=right width=170 style="border:5px solid #eee">
<h1 class="header-heading">Brian Johannesmeyer</h1>

PhD Student
Computer Science and Engineering
University of California, San Diego

Office: EBU3B 3144

Email:
<img src="assets/img/emailaddr.png" alt="">
<hr>
<h3>Research</h3>
My research interests are primarily in systems/embedded security and program analysis.
<hr>
<h3>Publications</h3>

S. Cauligi, G. Soeller, F. Brown, B. Johannesmeyer, Y. Huang, R. Jhala, and D. Stefan. FaCT: A Flexible Constant-Time Programming Language. IEEE SecDev 2017.
Z. Yang, B. Johannesmeyer, A.T. Olesen, S. Lerner, and K. Levchenko. Dead Store Elimination (Still) Considered Harmful. USENIX Security 2017 .
B. Yadegari, B. Johannesmeyer, B. Whitely, S. Debray. A Generic Approach to Automatic Deobfuscation of Executable Code. IEEE S&P 2015.
J. Qiu, B. Yadegari, B. Johannesmeyer, and S. Debray. Identifying and Understanding Self-Checksumming Defenses in Software. ACM CODASPY 2015.
J. Qiu, B. Yadegari, B. Johannesmeyer, and S. Debray. A Framework for Understanding Dynamic Anti-Analysis Defenses. PPREW 2014.

<hr>
<h3>Projects</h3>
<h4>Current</h4>

Aerosec: The Aviation Cyber Security Group
CESR: Center for Evidence-based Security Research

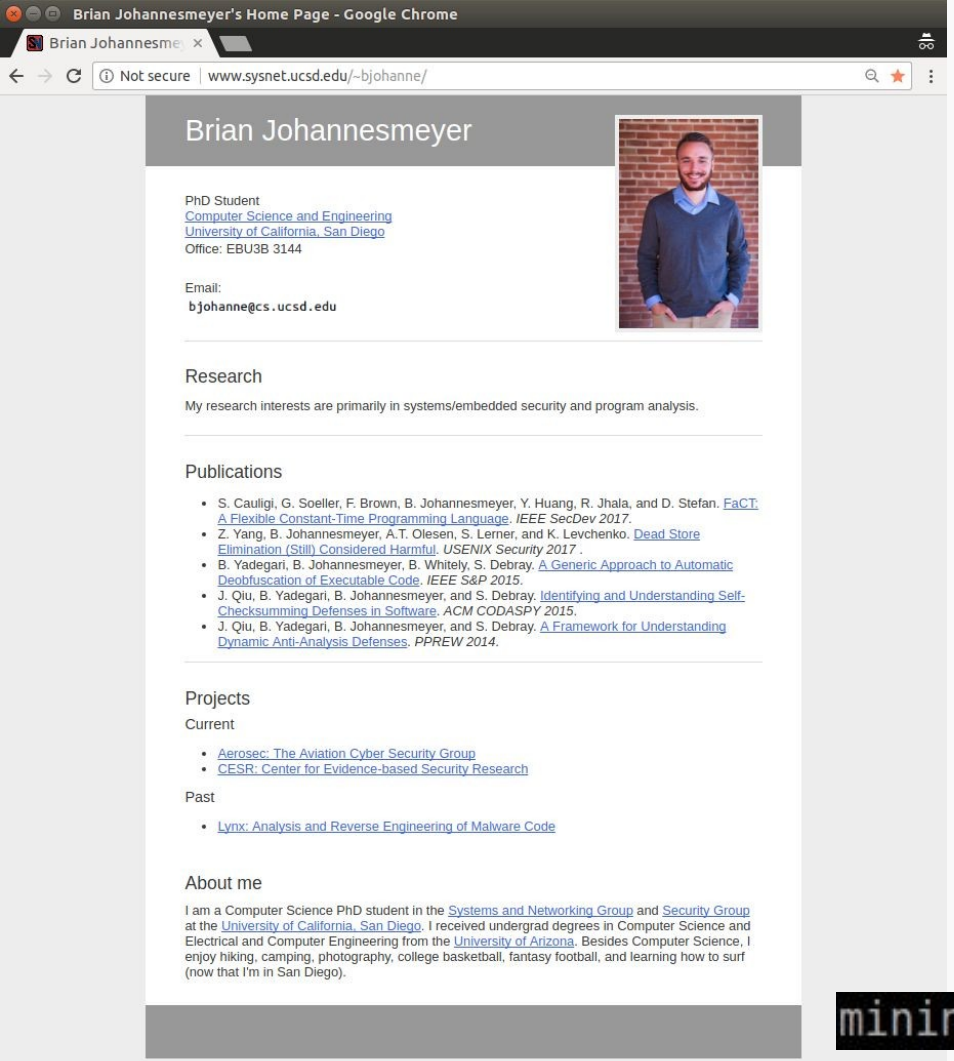
<h4>Past</h4>

Lynx: Analysis and Reverse Engineering of Malware Code

<hr>
<h3>About me</h3>
I am a Computer Science PhD student in the Systems and Networking Group and Security Group at the University of California, San Diego. I received undergrad degrees in Computer Science and Electrical and Computer Engineering from the University of Arizona. Besides Computer Science, I enjoy hiking, camping, photography, college basketball, fantasy football, and learning how to surf (now that I'm in San Diego).

<!-- © Copyright 2016 -->

</body>
</html>

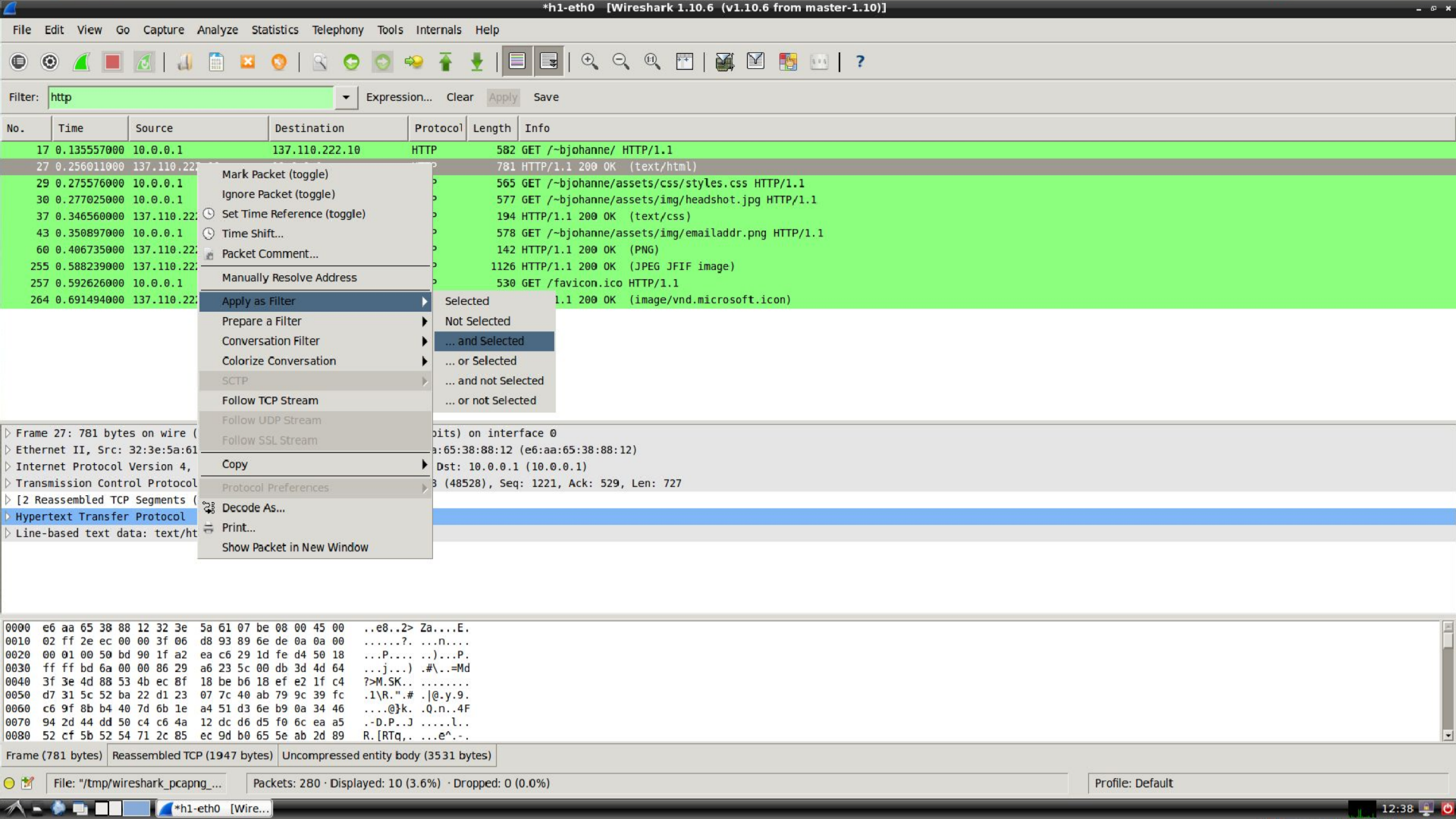


Great! We've loaded this webpage and started examining that in Wireshark!

However, we only loaded `index.html`, not any of the media associated with it, i.e. CSS, images, etc.

To do that, we can simply take another Wireshark trace, but this time load it in Chromium.

```
mininet> h1 su mininet -c chromium-browser
```


*h1-eth0 [Wireshark 1.10.6 (v1.10.6 from master-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: http
Expression... Clear Apply Save
No. Time Source Destination Protocol Length Info
17 0.135557000 10.0.0.1 137.110.222.10 HTTP 582 GET /~bjohanne/ HTTP/1.1
27 0.256011000 137.110.222 781 HTTP/1.1 200 OK (text/html)
29 0.275576000 10.0.0.1 565 GET /~bjohanne/assets/css/styles.css HTTP/1.1
30 0.277025000 10.0.0.1 577 GET /~bjohanne/assets/img/headshot.jpg HTTP/1.1
37 0.346560000 137.110.222 194 HTTP/1.1 200 OK (text/css)
43 0.350897000 10.0.0.1 578 GET /~bjohanne/assets/img/emailaddr.png HTTP/1.1
60 0.406735000 137.110.222 142 HTTP/1.1 200 OK (PNG)
255 0.588239000 137.110.222 1126 HTTP/1.1 200 OK (JPEG JFIF image)
257 0.592626000 10.0.0.1 530 GET /favicon.ico HTTP/1.1
264 0.691494000 137.110.222 1.1 200 OK (image/vnd.microsoft.icon)
Mark Packet (toggle)
Ignore Packet (toggle)
Set Time Reference (toggle)
Time Shift...
Packet Comment...
Manually Resolve Address
Apply as Filter
Prepare a Filter
Conversation Filter
Colorize Conversation
SCTP
Follow TCP Stream
Follow UDP Stream
Follow SSL Stream
Copy
Protocol Preferences
Decode As...
Print...
Show Packet in New Window
Selected
Not Selected
... and Selected
... or Selected
... and not Selected
... or not Selected
Frame 27: 781 bytes on wire (
Ethernet II, Src: 32:3e:5a:61
Internet Protocol Version 4,
Transmission Control Protocol
[2 Reassembled TCP Segments (
Hypertext Transfer Protocol
Line-based text data: text/ht
bits) on interface 0
a:65:38:88:12 (e6:aa:65:38:88:12)
Dst: 10.0.0.1 (10.0.0.1)
3 (48528), Seq: 1221, Ack: 529, Len: 727
0000 e6 aa 65 38 88 12 32 3e 5a 61 07 be 08 00 45 00 ..e8..2> Za....E.
0010 02 ff 2e ec 00 00 3f 06 d8 93 89 6e de 0a 0a 00 ......?. ...n....
0020 00 01 00 50 bd 90 1f a2 ea c6 29 1d fe d4 50 18 ...P.... ..)...P.
0030 ff ff bd 6a 00 00 86 29 a6 23 5c 00 db 3d 4d 64 ...j...) .#\..=Md
0040 3f 3e 4d 88 53 4b ec 8f 18 be b6 18 ef e2 1f c4 ?>M.SK.. ....
0050 d7 31 5c 52 ba 22 d1 23 07 7c 40 ab 79 9c 39 fc .1\R."..# .|@.y.9.
0060 c6 9f 8b b4 40 7d 6b 1e a4 51 d3 6e b9 0a 34 46 ....@}k. .Q.n..4F
0070 94 2d 44 dd 50 c4 c6 4a 12 dc d6 d5 f0 6c ea a5 .-D.P..J .....l..
0080 52 cf 5b 52 54 71 2c 85 ec 9d b0 65 5e ab 2d 89 R.[RTq,. ...e^.-.
Frame (781 bytes) Reassembled TCP (1947 bytes) Uncompressed entity body (3531 bytes)
File: "/tmp/wireshark_pcapng_...
Packets: 280 · Displayed: 10 (3.6%) · Dropped: 0 (0.0%)
Profile: Default
*h1-eth0 [Wire...
12:38

*h1-eth0 [Wireshark 1.10.6 (v1.10.6 from master-1.10)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: (http) && (ip.src == 137.110.222.10) Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
27	0.256011000	137.110.222.10	10.0.0.1	HTTP	781	HTTP/1.1 200 OK (text/html)
37	0.346560000	137.110.222.10	10.0.0.1	HTTP	194	HTTP/1.1 200 OK (text/css)
60	0.406735000	137.110.222.10	10.0.0.1	HTTP	142	HTTP/1.1 200 OK (PNG)
255	0.588239000	137.110.222.10	10.0.0.1	HTTP	1126	HTTP/1.1 200 OK (JPEG JFIF image)
264	0.691494000	137.110.222.10	10.0.0.1	HTTP	552	HTTP/1.1 200 OK (image/vnd.microsoft.icon)

- Frame 27: 781 bytes on wire (6248 bits), 781 bytes captured (6248 bits) on interface 0
- Ethernet II, Src: 32:3e:5a:61:07:be (32:3e:5a:61:07:be), Dst: e6:aa:65:38:88:12 (e6:aa:65:38:88:12)
- Internet Protocol Version 4, Src: 137.110.222.10 (137.110.222.10), Dst: 10.0.0.1 (10.0.0.1)
- Transmission Control Protocol, Src Port: http (80), Dst Port: 48528 (48528), Seq: 1221, Ack: 529, Len: 727
- [2 Reassembled TCP Segments (1947 bytes): #25(1220), #27(727)]
- Hypertext Transfer Protocol
- Line-based text data: text/html

```
0000  e6 aa 65 38 88 12 32 3e  5a 61 07 be 08 00 45 00   ..e8..2> Za....E.
0010  02 ff 2e ec 00 00 3f 06  d8 93 89 6e de 0a 0a 00   ......?. ...n....
0020  00 01 00 50 bd 90 1f a2  ea c6 29 1d fe d4 50 18   ...P.... ..)...P.
0030  ff ff bd 6a 00 00 86 29  a6 23 5c 00 db 3d 4d 64   ...j...) .#\..=Md
0040  3f 3e 4d 88 53 4b ec 8f  18 be b6 18 ef e2 1f c4   ?>M.SK.. ........
0050  d7 31 5c 52 ba 22 d1 23  07 7c 40 ab 79 9c 39 fc   .1\R."."# .|@.y.9.
0060  c6 9f 8b b4 40 7d 6b 1e  a4 51 d3 6e b9 0a 34 46   ....@}k. .Q.n..4F
0070  94 2d 44 dd 50 c4 c6 4a  12 dc d6 d5 f0 6c ea a5   .-D.P..J .....l..
0080  52 cf 5b 52 54 71 2c 85  ec 9d b0 65 5e ab 2d 89   R.[RTq,. ...e^.-.
```

Frame (781 bytes) | Reassembled TCP (1947 bytes) | Uncompressed entity body (3531 bytes)

File: "/tmp/wireshark_pcapng_... | Packets: 280 · Displayed: 5 (1.8%) · Dropped: 0 (0.0%)

*h1-eth0 [Wire...

Brian Johannesmeyer's Home Page - Google Chrome

Brian Johannesme

Not secure | www.sysnet.ucsd.edu/~bjohanne/

Brian Johannesmeyer

PhD Student
Computer Science and Engineering
University of California, San Diego
Office: EBU3B 3144

Email:
bjohanne@cs.ucsd.edu

Research

My research interests are primarily in systems/embedded security and program analysis.

Publications

- S. Cauligi, G. Soeller, F. Brown, B. Johannesmeyer, Y. Huang, R. Jhala, and D. Stefan. FaCT: A Flexible Constant-Time Programming Language. *IEEE SecDev 2017*.
- Z. Yang, B. Johannesmeyer, A.T. Olesen, S. Lerner, and K. Levchenko. Dead Store Elimination (Still) Considered Harmful. *USENIX Security 2017* .
- B. Yadegari, B. Johannesmeyer, B. Whitely, S. Debray. A Generic Approach to Automatic Deobfuscation of Executable Code. *IEEE S&P 2015*.
- J. Qiu, B. Yadegari, B. Johannesmeyer, and S. Debray. Identifying and Understanding Self-Checksumming Defenses in Software. *ACM CODASPY 2015*.
- J. Qiu, B. Yadegari, B. Johannesmeyer, and S. Debray. A Framework for Understanding Dynamic Anti-Analysis Defenses. *PPREW 2014*.

Projects

Current

- Aerosec: The Aviation Cyber Security Group
- CESR: Center for Evidence-based Security Research

Past

- Lynx: Analysis and Reverse Engineering of Malware Code

About me

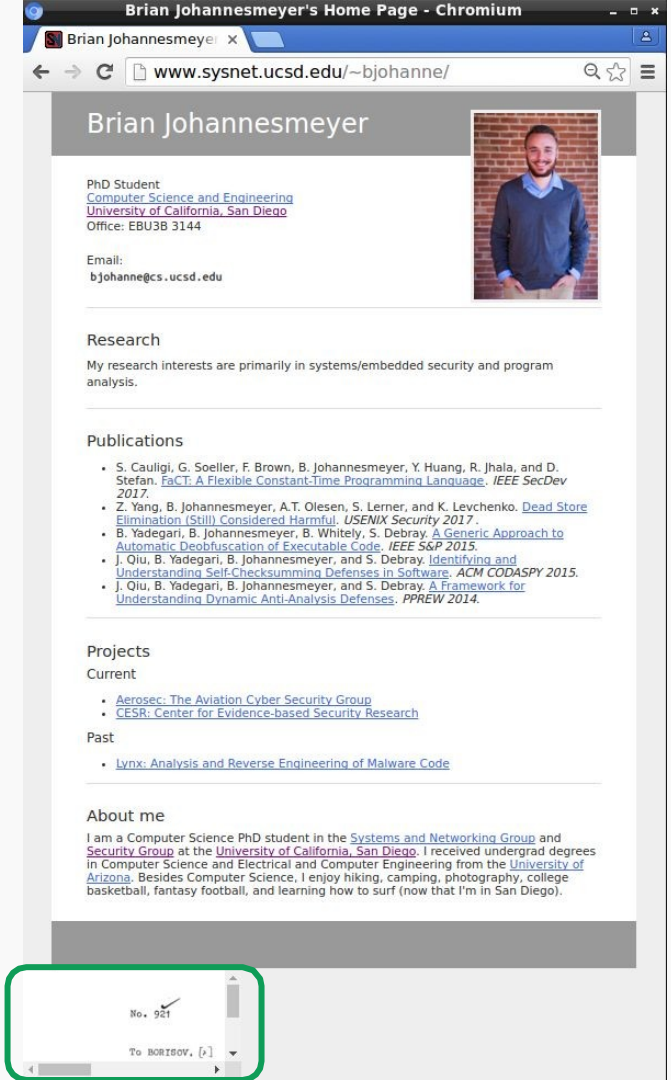
I am a Computer Science PhD student in the Systems and Networking Group and Security Group at the University of California, San Diego. I received undergrad degrees in Computer Science and Electrical and Computer Engineering from the University of Arizona. Besides Computer Science, I enjoy hiking, camping, photography, college basketball, fantasy football, and learning how to surf (now that I'm in San Diego).

Great! We've loaded the webpage in Chrome and examined it in Wireshark!

However, what should our network attack actually do?

Let's modify our target domain/url in ~/pox/pox/forwarding/dummy.py:

```
#TARGET_DOMAIN_RE = re.compile(r'^blink.ucsd.edu$', re.I)
TARGET_DOMAIN_RE = re.compile(r'^www.sysnet.ucsd.edu$', re.I)
#URL_PATH_RE = re.compile(r'^/technlogy/security/', re.I)
URL_PATH_RE = re.compile(r'^/~bjohanne/', re.I)
IFRAME_URL = 'http://cryptosec.ucsd.edu'
```

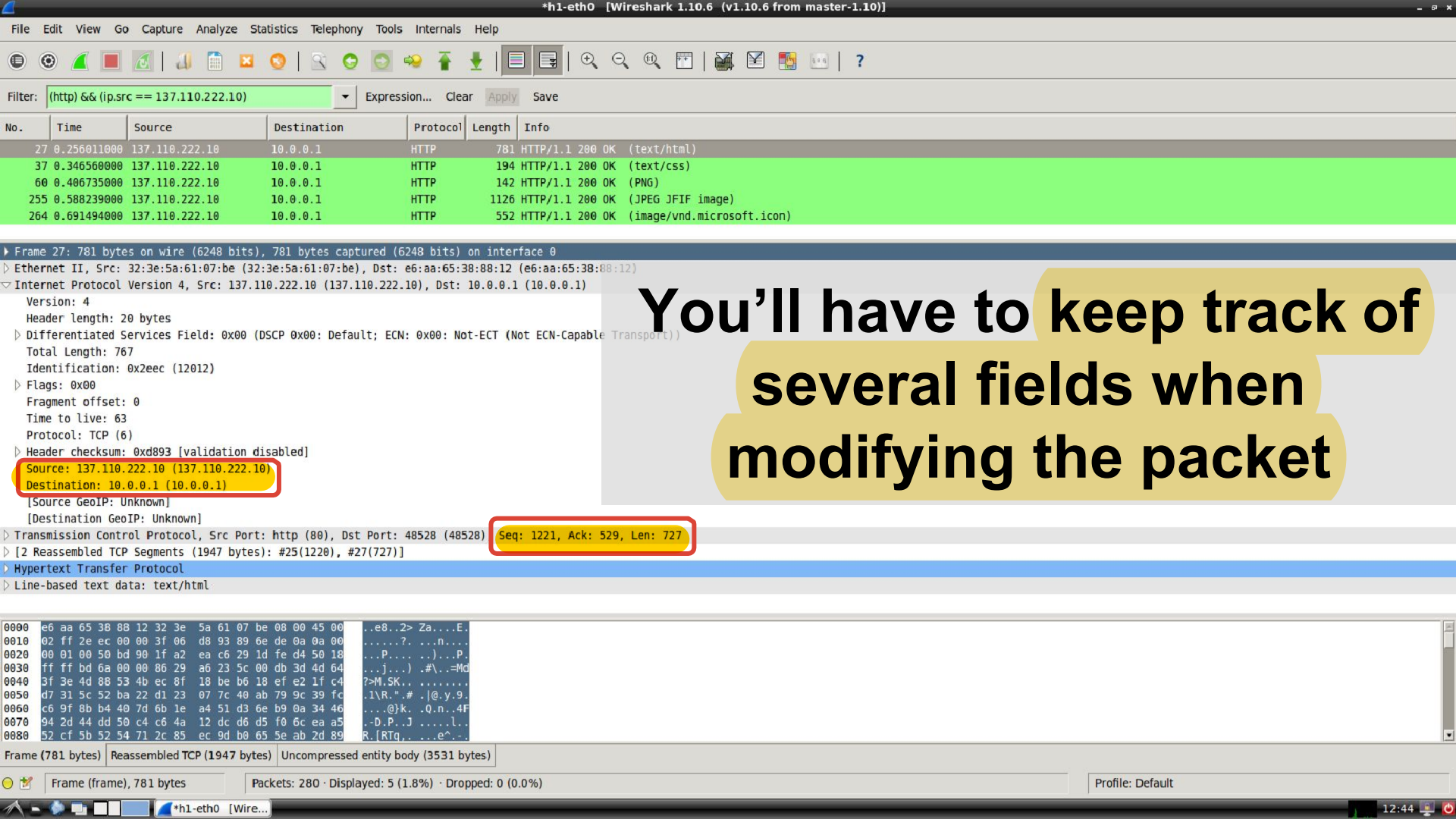


Challenges

- Targeting TCP packets from the right domain/url
  - Look at Python's `re` module
- Forcing the server not to compress responses
  - Modify the HTTP `Accept-Encoding` request header to list only `identity`
- Injecting iframes into the response
  - Modify the HTTP `Content-Length` header appropriately
- Handling longer packet streams
  - Make sure source and destination IPs/ports are the same as a previous packet
- Ensuring correct TCP functionality
  - Manipulate SEQ and ACK numbers
- Using the POX API
  - There's plenty available to use, e.g. ip_packet.find(), *.payload, *.ack, *.seq, etc.
- Testing webpages
  - You can use a server I've set up to test your code. The server has compression and chunked encoding turned off: saguaro.ucsd.edu

Targeting the right domain and URL

```
import re
domain_re = re.compile(r'^saguaro\.ucsd\.edu$', re.I)

# Matches -- the hostname matches
re.search(domain_re, 'saguaro.ucsd.edu')

# Does not match -- the \. ensures that the following hostname does not match
re.search(domain_re, 'saguaroXucsdXedu')

# Does not match -- the ^ and $ ensure that the following hostname does not match
re.search(domain_re, 'Xsaguaro.ucsd.eduX')

# Does not match -- this is an example HTTP header; however, we should first parse out the hostname
re.search(domain_re, 'GET / HTTP/1.1\r\nUser-Agent: Wget/1.15 (linux-gnu)\r\nAccept: */*\r\nHost:
saguaro.ucsd.edu\r\nConnection: Keep-Alive\r\n\r\n)'

# Let's change the regex so that all subdomains of saguaro.ucsd.edu are targeted
domain_re = re.compile(r'saguaro\.ucsd\.edu$', re.I)
# Matches -- the hostname matches
re.search(domain_re, 'foo.saguaro.ucsd.edu')
```



You'll have to keep track of several fields when modifying the packet

TCP's SEQ and ACK numbers

- **ACK** = num of bytes received
- **SEQ** = num of bytes sent (i.e. num of bytes known that the other end has received)

1. Connection establishment (3-way handshake)

Client HTTP request + Server ACK

2. Connection (packet/acknowledgement pairs)

Server HTTP responses + Client ACKs

3. Connection termination (3-way handshake)