Defending Computer Networks
Lecture 14: HTTP & NIDS

Stuart Staniford

Adjunct Professor of Computer Science

Logistics

- No class Tuesday
- HW3.
  - Up on website later today
  - Due Weds 10/22/14 11:59pm
- Project options
  - Up on website later today
  - Will sketch today

Assigned Reading

- Rain Forest Puppy. A Look at Whisker's Anti-IDS Tactics. http://www.ussrback.com/docs/papers/IDS/whiskerids.html

Obama Had Security Fears on JPMorgan Data Breach

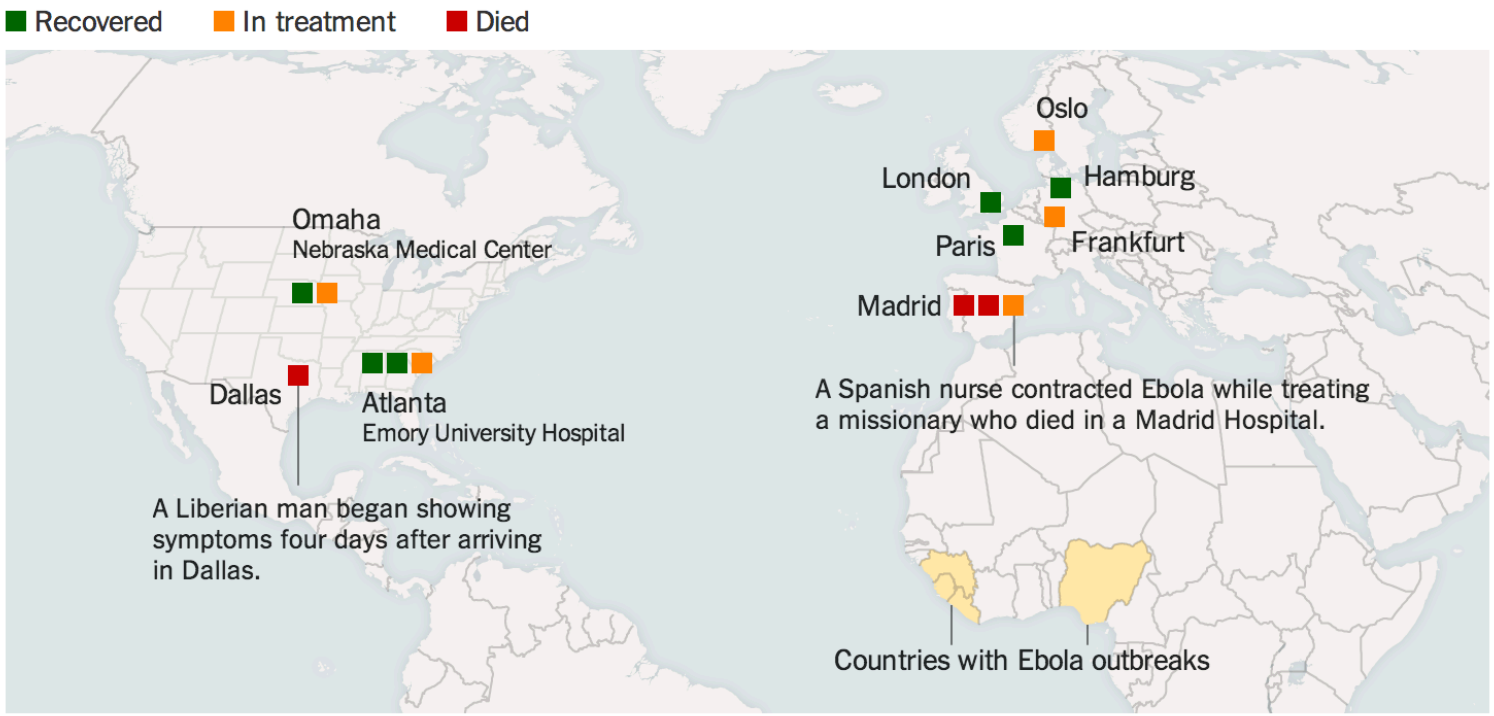
By MICHAEL CORKERY, JESSICA SILVER-GREENBERG and DAVID E. SANGER
OCTOBER 8, 2014 2:08 PM 106 Comments

Updated, 9:57 p.m. | President Obama and his top national security advisers began receiving periodic briefings on the huge cyberattack at JPMorgan Chase and other financial institutions this summer, part of a new effort to keep security officials as updated on major cyberattacks as they are on Russian incursions into Ukraine or attacks by the Islamic State.

But in the JPMorgan case, according to administration officials familiar with the briefings, who would not speak on the record about intelligence matters, no one could tell the president what he most wanted to know: What was the motive of the attack? "The question kept coming back, 'Is this plain old theft, or is Putin retaliating?' " one senior official said, referring to the American-led sanctions on Russia. "And the answer was: 'We don't know for sure.' "

More than three months after the first attacks were discovered, the source is still unclear and there is no evidence any money was taken from any institution.

http://dealbook.nytimes.com/2014/10/08/cyberattack-on-jpmorgan-raises-alarms-at-white-house-and-on-wall-street/



Cases of Ebola Outside of West Africa (as of Oct. 8, 2014)

United States	Arrival date	
Aid worker	Aug. 2	Recovered
Missionary	Aug. 2	Recovered
Doctor	Sept. 5	Recovered
Doctor	Sept. 9	In treatment
Visitor	Sept. 30*	Died
NBC cameraman	Oct. 6	In treatment
France		
Nurse	Sept. 19	Recovered
Britain		
Nurse	Aug. 24	Recovered

Spain	Arrival date	
Missionary	Aug. 7	Died
Priest	Sept. 22	Died
Nurse	Oct. 6*	In treatment
Germany		
Doctor	Aug. 27	Recovered
Doctor	Oct. 3	In treatment
Norway		
Aid worker	Oct. 6	In treatment

*Date of Ebola diagnosis.

Main Goals for Today

- Introduce Projects
- Introduce basics of HTTP
- NIDS in Context of HTTP Server Attacks

Projects

40% class project. You will build a non-trivial piece of C code from scratch to do an interesting task in network security. You will write a document describing its algorithms and architecture (10% of total grade) and demonstrate how well it works at an interim milestone (10%) and towards the end of the course (20%).

Projects will be solo.

C vs other languages

Project Description 1

Develop a working remote exploit for a previously unknown vulnerability in a widespread piece of software that works on a current 64 bit operating system with all defenses in place. Note that attacking software across the Internet (or Cornell's network) is generally illegal, so you should attack a piece of software for which you have local access. You should then notify the software vendor/development team of the vulnerability and provide them with your proof of concept exploit (keeping it secret in the meantime).

Intermediate milestone is to have selected your vulnerable application/OS, and demonstrate that you can crash it with malicious input.

The document should explain the nature of the vulnerability, how you worked around the various OS/compiler defenses, and what your shellcode/ROP chain/etc does.

Note that this project has hard-to-estimate risks of failure if you pick something that turns out not to be exploitable by you in the available time. But if you succeed, we know for sure that you are 31337.

Project Description 2

Build a simple network firewall from scratch in C. Your firewall should have the ability to handle transferring packets between multiple network interfaces (eg wireless and wired interfaces on your laptop), and also the ability to transfer packets between pcap files for testing purposes. Your firewall should be stateful, with the ability to keep track of TCP, UDP, and ICMP conversations going on in the network. You should implement a text-based rules language of your own design that includes the ability to block/pass/reject network conversations based on source/destination address ranges and port numbers. You should obtain multi-gigabyte pcap files online for testing, and be able to demonstrate that your algorithms do not crash or blow-up in time/space demands on large files.

Intermediate milestone is to demonstrate that you can pass packets between multiple interfaces and files, with a single rule of some kind.

The document should describe and give the rationale for: 1) your rules language, 2) the data structures/algorithms used in your code, and 3) your test plan and the results of your tests.

Project Description 3

Build a web-exploit scanner from scratch in C. Your code should be able to take a list of malicious domains, reach out to them via HTTP, replay the content in a virtual machine and perform some simple steps to determine if bad things have happened in the virtual machine (eg look for browser crashes or memory explosions). You should include code to obtain secondary downloads that the virtual machine asks for. Note that you must implement your own HTTP client/proxy that can handle all three major methods of length delineation. It's acceptable to use libraries to handle gzip decoding of content. You should demonstrate that your code can stay up on a list of at least hundreds of bad domains, and you should demonstrate that you can detect at least some malicious websites.

Intermediate milestone is to be able to get an HTML file off disk started in a virtual machine browser, and demonstrate the intercept and parsing of outbound HTTP requests from the VM.

The document should describe and give the rationale for: 1) how your code interacts with the VM/browser, 2) the data structures/algorithms used in your code, and 3) your test plan and the results of your tests, including the malicious domains you detected.

Project Description 4

Project Description X

- Other. If you have a burning desire to do something else of similar scope to the above,
  - Put together a one paragraph description
  - Let me know.

Project Deadlines

- Interim milestone demonstration:
  - Friday November 7th
- Document Due
  - Tuesday December 2nd
- Final implementation due:
  - Friday December 5th

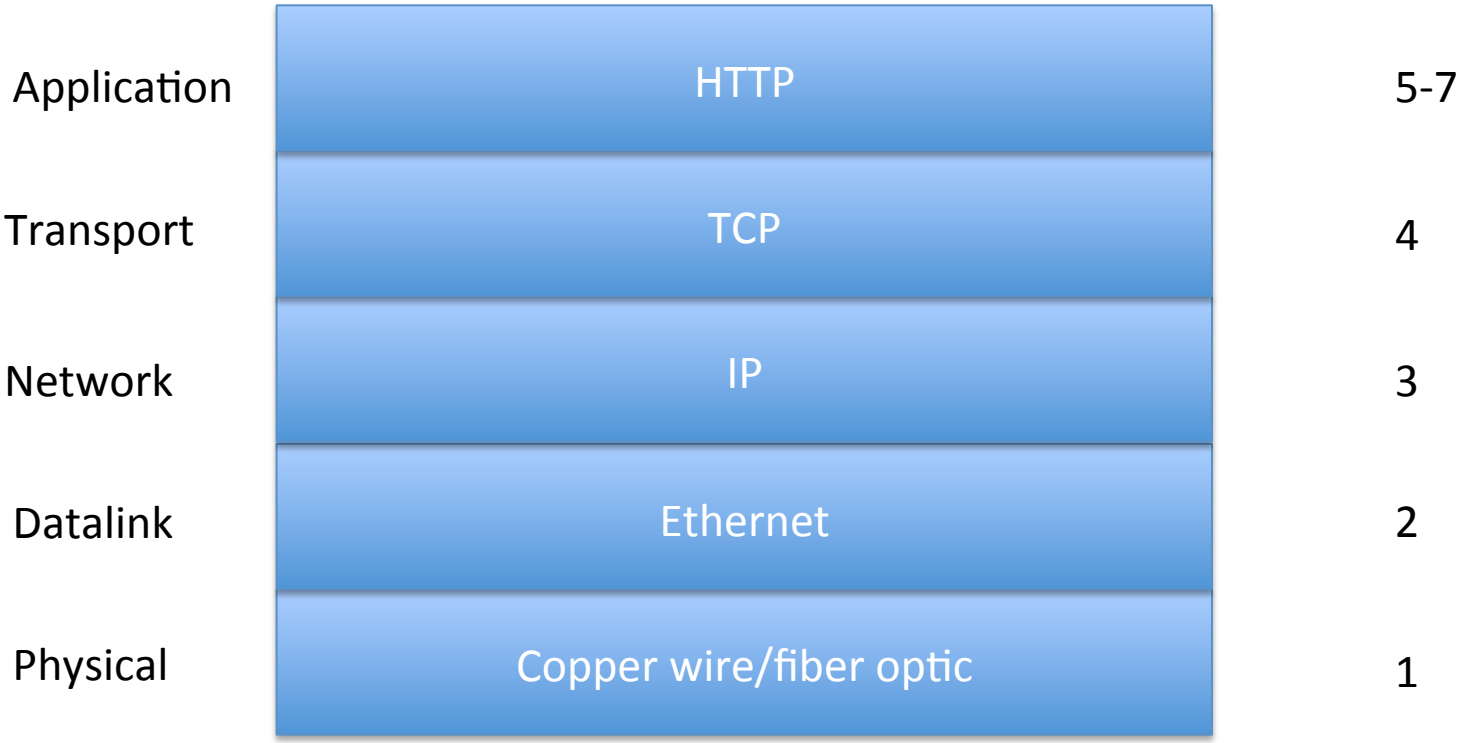
HTTP 1.1

- Main protocol that web runs over
- By far most important application protocol on Internet
- RFC 2616 (1999)
  - Obsoletes RFC 1945 for HTTP 1.0
  - HTTP originally dates back to Tim Berners Lee/CERT in 1991 (v 0.9)
- Text based request/response protocol
  - Originally primarily to identify/download files
  - Also provides for web applications

Protocol Layering

- HTTP runs over TCP
- In HTTP 1.1, one TCP connection can have a series of HTTP requests
  - Reverse direction carries responses
- NB: connection structure is not that meaningful to HTTP
  - Different browser may use one or multiple TCP connections
    - And spread requests between them differently.
  - Proxies can rearrange requests/responses to different connections.

Protocol Layering Names/Numbers



HTTP Request

GET /dumprequest HTTP/1.1\r\n
Host: djce.org.uk\r\n
Connection: keep-alive\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_5) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/30.0.1599.101 Safari/537.36\r\n
DNT: 1\r\n
Referer: https://www.google.com/url?
sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=0CD4QFjAC&url=http%3A%2F
%2Fdjce.org.uk
%2Fdumprequest&ei=835lUpjEM5Xb4APEgIGoDA&usg=AFQjCNEeAn5wSZMp_y_oTmOK
onq482sS9A&sig2=pSajtDK-YYIvE4HFDqmRfA&bvm=bv.54934254,d.dmg\r\n
Accept-Language: en-US,en;q=0.8\r\n
\r\n

Try it at http://www.procato.com/my+headers/

HTTP Header Basics

- Text lines separated by \r\n
  - Servers often accept "\n" only, but protocol is "\r\n"
- Header is terminated by a blank line (\r\n\r\n)
- Initial request line
  - GET /dumprequest HTTP/1.1\r\n
    - Other methods include POST, CONNECT, HEAD, DELETE, etc.
    - Focus on GET for now
- Followed by headers of form
  - Header: Value…\r\n
  - No request headers are actually required

Let’s try it

- Telnet to www.google.com 80 and try a manually entered request for nosuchpage.html

A Few Popular Request Headers

- Host:
  - Used to specify domain (server might have several).
- User-Agent:
  - Gives browser specifics (allows server to customize responses to browser)
- Referer:
  - What page (etc) sent us here
- Accept-Language:
  - We speak English, or…
- Accept:
  - media formats we accept (eg text/html)