The story so far

~~Introduction~~

~~Software security~~

~~Host security~~

~~Network security~~

Web security

Today

Web model

Web history

Key Web security concepts

What's this Web thing?

Documents: text, images, etc.

***Hypertext:* documents+edges**

Some examples: SGML, GNU `info`, Mundaneum, Project Xanadu...

... and HTML!

Source: Zinneke via Wikimedia

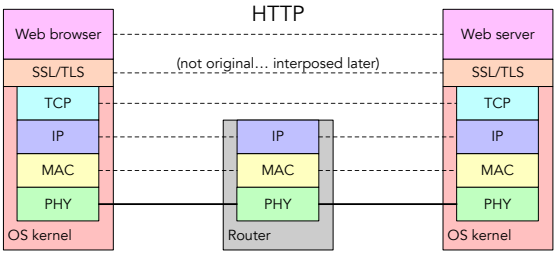
This image shows people working in the Universal Bibliographic Repertory, which would eventually become the Mundaneum. This system of cross-referenced index cards was meant to catalogue all of the world's information, which is now pretty close the Google's mission statement. Project Xanadu is particularly wild to read about!

Web formats and protocols

HTTP

HTML

... other stuff



The *hypertext transfer protocol* (HTTP, see RFC 2616) describes how we transfer Web content from servers to clients. We can use `telnet`, `nc` (netcat), etc., to talk to an HTTP server and act as our own Web browser:

```
GET / HTTP/1.0

HTTP/1.1 301 Moved Permanently
Date: Mon, 25 Jul 2022 14:08:02 GMT
Server: Apache
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
Content-Security-Policy: frame-ancestors 'self';
...
```

Or, even better:

```
GET / HTTP/1.1
Host: www.engr.mun.ca

HTTP/1.1 301 Moved Permanently
Date: Mon, 25 Jul 2022 14:08:02 GMT
Server: Apache
...
```

This "other stuff" includes static-ish content like images, declarative content like CSS and dynamic software in languages like JavaScript and WebAssembly!

Web history

1980s

Tim Berners-Lee @ CERN
http://info.cern.ch/

1990s

NCSA Mosaic
Netscape
Everything else

The first Web server: a NEXT computer at CERN

Now *Sir* Tim Berners-Lee, I suppose!

Much like the ARPAnet was developed to address a real-world problem, the Web was originally created to address a problem: making it easier for scientists at CERN to share information with each other without having to all be running exactly the same software. One lesson that we should take from this history is that the most interesting and impactful projects come from _____ _____ _____ _____. Another good lesson is probably that engineering is never done for its own sake: we build things _____ _____ _____!

In the early days, one could visit websites like info.cern.ch (the first Web server) using tools like the line mode browser, a very bare-bones command-line tool that could run on almost any operating system with a network stack.

Tim Berners-Lee initially created a graphical browser for the NeXT computer, but they weren't able to port it to other operating systems. Instead, the graphical browser that really helped the Web become popular was Mosaic from the NCSA (National Center for Supercomputing Applications). Mosaic only lived for 4–5 years: it was developed in 1992, released in 1993 and discontinued in 1997.

Mosaic was rapidly overtaken by Netscape, a browser started by one of Mosaic's original authors.

Netscape was itself rapidly overtaken by Internet Explorer, partially due to what a US judge would later find to be an abuse of its OS monopoly power (the whole thing makes for a great read... did you know that the judge originally ordered Microsoft to be broken up into two companies?).

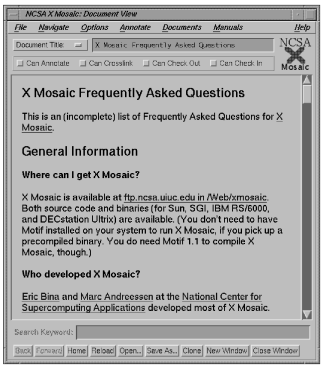
The original Web

Scientists sharing data

Collaboration without (onerus) standardization

A few lowest-common-denominator formats: HTML, GIF, JPEG, MPEG...

Trust assumptions?



*NCSA Mosaic browser**

* Andreessen, "NCSA Mosaic Technical Summary", *National Center for Supercomputing Applications*, 20 Feb 1993. Available: http://web.archive.org/web/19991009182307/http://cbl.leeds.ac.uk/WWW/ps/ghindex.html

This "lowest common denominator" approach allowed collaboration via text and images, which could be produced from almost any software and which could end up in scientific publications. It also allowed videos, which are essential for communicating some scientific results.

Original HTML was an elegant approach for a more civilized age (as, if you don't mind quite a lot of profanity, this website expresses).

Once again, the assumptions of a system's designers have a huge effect on how that system is used in perpetuity. What assumptions might the scientists building this data-sharing system have had, and what might the implications for the Web's TCB have been?

Commercialization

Netscape, then Internet Explorer

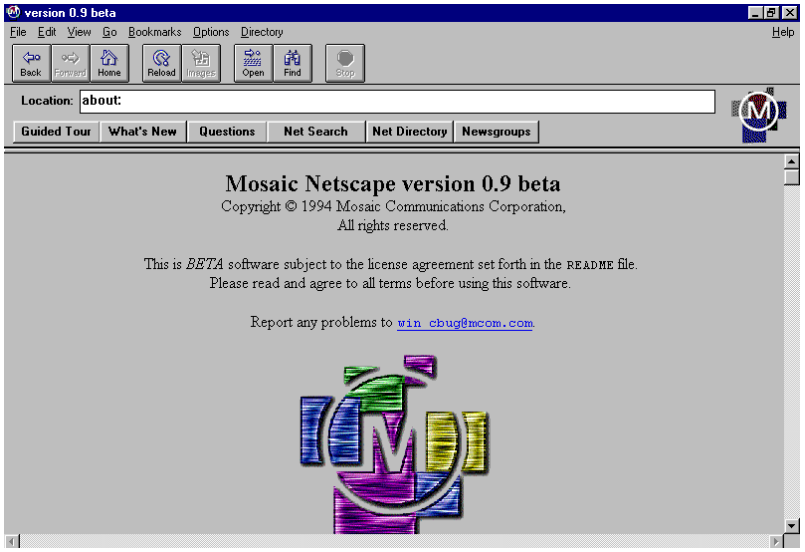
Dramatic acceleration through competition

Security... what's that?

Netscape started from a dominant position in user share, but Microsoft used code from NCSA Mosaic and the dominance of its operating system to challenge (and eventually overtake) Netscape.

Every new major version of Netscape and IE saw major new features added as proprietary HTML extensions which would then need to be quickly copied by the other. Fonts, background colours, frames, tables... it was all a bit chaotic, with different browsers supporting different features and Web developers (now starting to become "a thing") needing to support many versions of unevenly-extended HTML.

In the race to implement the Next Big Feature, security wasn't a first-class consideration.



Netscape 0.9 (beta), back when it was still called Mosaic Netscape. The owners of the Mosaic trademark weren't pleased about that; subsequent versions of Netscape dropped the "Mosaic" part.

Cookies

Origin

Kinds

Session cookies, persistent cookies, "secure" cookies, third-party cookies...

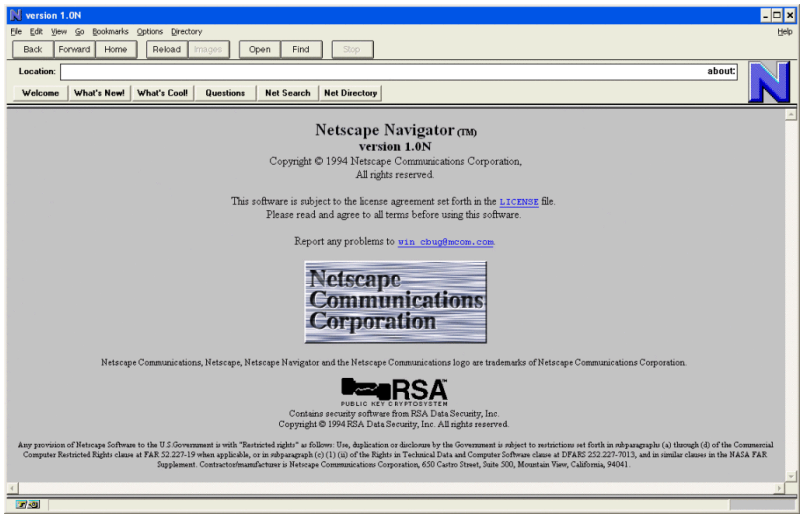
Privacy

What's that? (user tracking, FTC hearings...)

This might be the first time that the US Federal Trade Commission had to discuss the word "cookies" so publically! Since companies didn't seem very interested in user privacy, it became a matter of regulation, with reports to the US Congress on online privacy and ever-increasing awareness of online privacy issues by regulators around the world.

The FTC is still watching how companies use cookies, in some cases settling for tens of millions with Web giants. Also today, the FTC has a very long webpage explaining how *they* use cookies on their site... it's a pretty good example of real transparency on cookies.

Now, however, the EU has taken the lead on regulating cookies via the General Data Protection Regulation (GDPR). However, forcing people to click on "yes, find, whatever" isn't exactly the user-respecting new world we hoped it would be...


version 1.0N
File Edit View Go Bookmarks Options Directory
Help
Back Forward Home Reload Images Open Find Stop
Location:
about:
Welcome What's New! What's Cool! Questions Net Search Net Directory
Netscape Navigator (TM)
version 1.0N
Copyright © 1994 Netscape Communications Corporation,
All rights reserved.
This software is subject to the license agreement set forth in the LICENSE file.
Please read and agree to all terms before using this software.
Report any problems to win cbug@mcom.com
Netscape Communications Corporation
Netscape Communications, Netscape, Netscape Navigator and the Netscape Communications logo are trademarks of Netscape Communications Corporation.
RSA
PUBLIC KEY CRYPTOSYSTEM
Contains security software from RSA Data Security, Inc.
Copyright © 1994 RSA Data Security, Inc. All rights reserved.
Any provision of Netscape Software to the U.S.Government is with "Restricted rights" as follow: Use, duplication or disclosure by the Government is subject to restrictions set forth in subparagraphs (a) through (d) of the Commercial Computer Restricted Rights clause at FAR 52.227-19 when applicable, or in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, and in similar clauses in the NASA FAR Supplement. Contractor/manufacturer is Netscape Communications Corporation, 650 Castro Street, Suite 500, Mountain View, California, 94041.

Netscape 1.0!

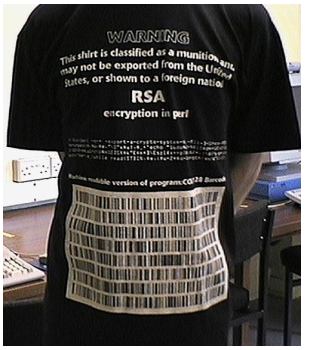
SSL

Secure sockets layer

(later *Transport Layer Security*)

- Diffie-Hellman key exchange
- RSA, DES, MD5...

Export controls

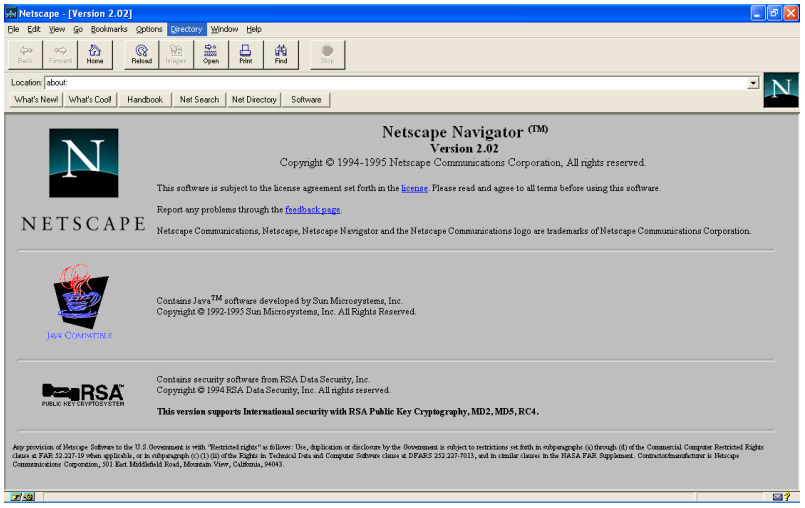


A protest T-shirt

We've looked at TLS in the lab, so we've seen most of these things before. One difference is that instead of AES and SHA-384, we see older algorithms like DES and MD5. Another key difference is that these versions of SSL didn't have the same level of formal-methods rigour applied to them, which led to lots of vulnerabilities at the protocol level:

- BEAST
- FREAK
- Logjam
- POODLE
- CRIME
- ... and many, many more

Another key problem with early SSL was the "crypto wars" that we've previously discussed. Cryptography was export-controlled, so only cripped algorithms with small keys could be used in software exported from the US to other countries. After much public brouhaha, the Clinton administration eventually relented and allowed "proper" crypto to be exported. It's since paid off pretty well in terms of economic activity.



This image isn't quite Netscape 2.0: it's actually v2.02. That's pretty close, though.

Netscape Navigator 2.0

Java!

- Web pages could run code via *applets*
- applets could be stitched into Web pages using...

JavaScript!

JavaScript was originally meant to be a side show language, something with just enough functionality to tie "real" programs into HTML pages. Now, Java applet support is deprecated and JavaScript (now ECMAScript) continues to develop as a first-class programming language... though unfortunately its lack of original "real language" intent sometimes shines through (see https://www.youtube.com/watch?v=et8xNAc2ic8).

The ability to use JavaScript led to the ability to *load* JavaScript from other servers, which led to...

Same Origin Policy

- code should only have access to things from the "same origin"
- not really a *policy*... organically-grown *convention**
- not always applied consistently!†

---

* Barth, "The Web Origin Concept", RFC 6454, 2011.

† Schwenk, Niemietz and Mainka, "Same-Origin Policy: Evaluation in Modern Browsers", in *Proceedings of the 26th USENIX Security Symposium*, 2017. Available: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/schwenk

We'll spend next class talking about some of the implications of the Same Origin Policy. Specifically, we'll talk about _____ and what we can do about it.

Summary

Web model

Web history

Key Web security concepts