Taking Back The Web With Haketilo

Presenter: Nicholas Johnson

Date: 19 March 2022

Event: LibrePlanet 2022 – Living Liberation

Project Developers
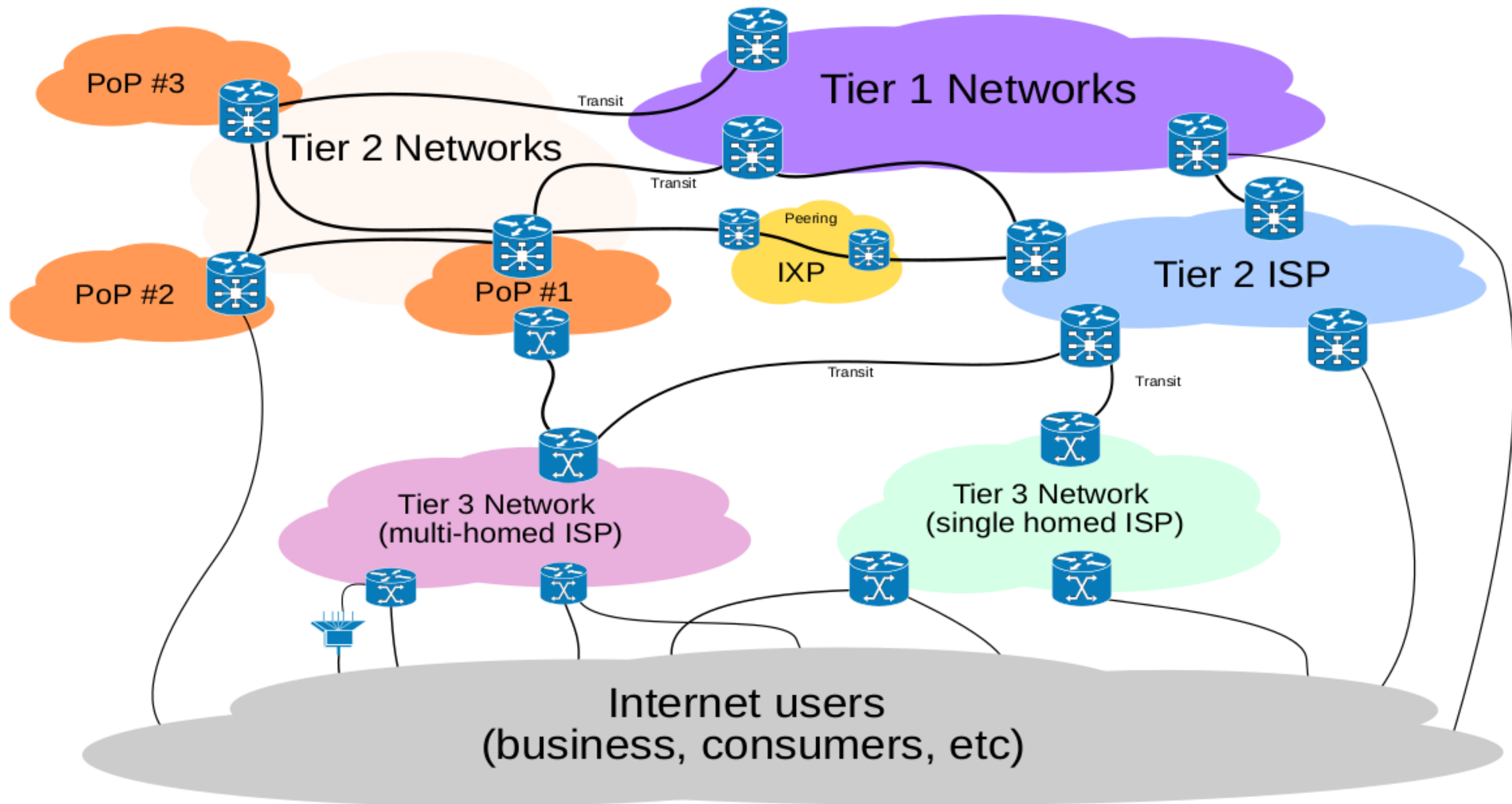
Name	Email
• Jahoti	jahoti@envs.net
• Nicholas Johnson	nick@nicksphere.ch
• Wojciech Kosior (main developer)	koszko@koszko.org

Internet Versus Web

What is the internet?

The global network of interconnected computers using the internet protocol (IPv4 or IPv6) to communicate.

PoP #3

Tier 1 Networks

Tier 2 Networks

Transit

Transit

PoP #2

PoP #1

Peering

IXP

Tier 2 ISP

Transit

Transit

Tier 3 Network
(multi-homed ISP)

Tier 3 Network
(single homed ISP)

Internet users
(business, consumers, etc)

Internet Versus Web

What is the Web?

A collection of billions of interlinked websites that can be accessed through a web browser.

What is a Web browser?

A computer program that implements the web standards as specified by the main international standards organization, the W3C.

Here's a non-exhaustive list of Web browsers: Firefox, Tor Browser, GNU Icecat, Brave, Lynx, Google Chrome, Microsoft Edge, Safari

The browsers in red don't respect user freedom. Do not use them.

The Beginning of The Web

The original web protocol (HTTP 0.9) was defined in 1991 by Tim Berners Lee. It was very simple.

Web browsers sent an HTTP GET request with one parameter. Web servers replied with the requested HTML document. HTML supported only 18 different tags.

HTTP Request:

```
GET /myawesomepage.html
```

HTTP Response:

```
<html>
This is my awesome webpage!
</html>
```

The Beginning of The Web

1993 was an important year for the web:

- Tim Berners Lee put the Web in the public domain.

- The multi-platform Mosaic Web browser was launched, gaining worldwide popularity and making the Web the most popular internet protocol.

- Hundreds of web servers came online.

In 1994 the World Wide Web Consortium was founded to develop Web standards.

Between 1991 and 1995, a few Web browsers began to take shape. New additions to the Web protocol were being tried. Incompatibility was rife.
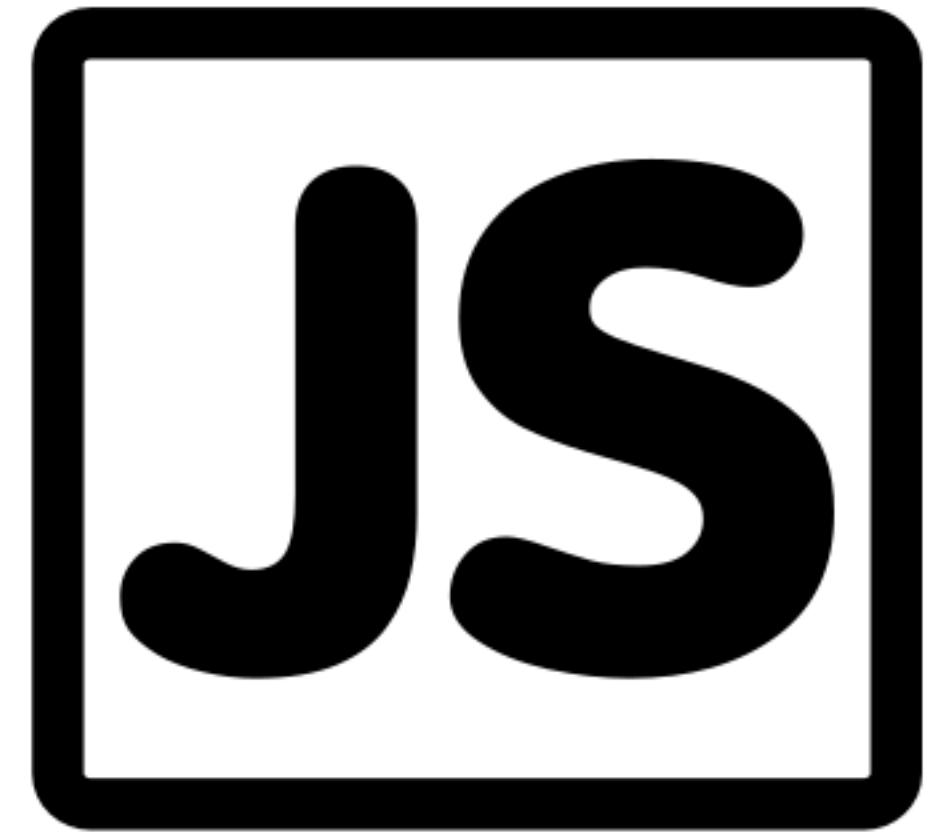
The Beginning of Javascript

In September 1995, Netscape added a scripting language to Navigator beta.

Mocha → Livescript → JavaScript

Navigator's competitor Internet Explorer 1.0 released JScript, its own implementation of JavaScript.

Originally used for form validation and minor presentation changes. Basically an extension of HTML and CSS.

JS

The Beginning of JavaScript

From 1997 to present day, JavaScript has grown into a full-fledged programming language suitable for developing large applications for the web.

Notable events:

- 1997: The first ECMAScript standard is published, marking the beginning of JavaScript as a general-purpose programming language for the web.

- 2009: Richard Stallman publishes "The JavaScript Trap", acknowledging the problem of non-free JavaScript programs.
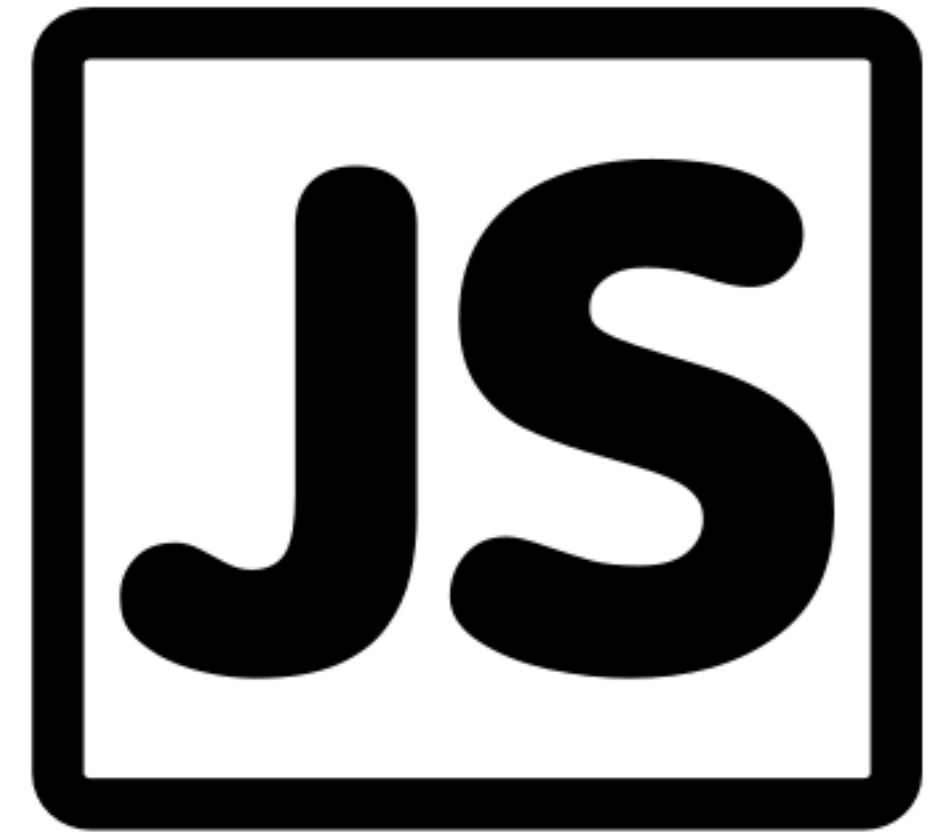
Modern JavaScript Usage

- Tracking scripts profile user activity and spy on them across webpages without the user's knowledge or consent.

- 97.8% of websites use JavaScript[1], often excluding users who have good reasons not to enable it. It also makes sites slow to load, bloated, and insecure.

- Most web application security vulnerabilities are caused by client-side JavaScript, not the server-side.

- Many sites don't work without non-free JavaScript, making the web hard to use in freedom.

References:
1: https://w3techs.com/technologies/details/cp-javascript
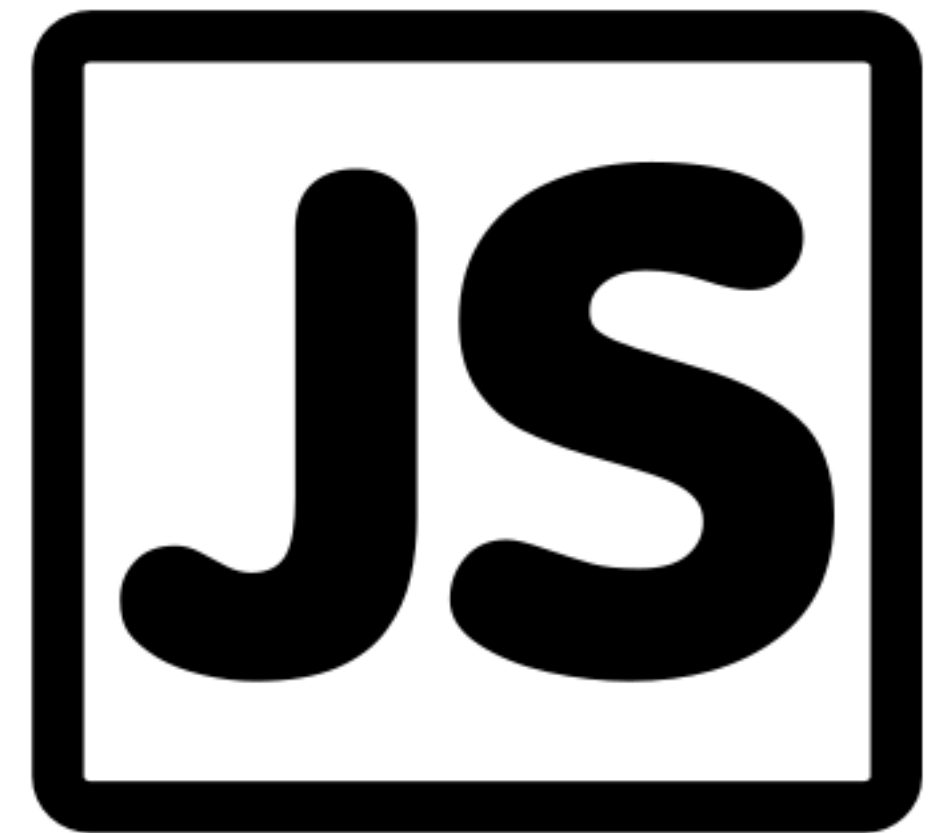
HTTP Cookies

- An HTTP cookie is data the server stores in the user's browser, useful for virtual shopping carts, etc.

- First implemented in 1994.

- In 1997, the Internet Engineering Task Force published RFC 2109 which specifies that third-party cookies should be disallowed or turned off by default.

- New security vulnerabilities like CSRF and XSS for cookie theft.

Hollyweb: DRM as a Standard

In 2017, the W3C sold out the web to Hollywood by making proprietary Digital restrictions management (DRM), known as EME, an official web standard. It is now supported by all major browsers.

The Electronic Frontier Foundation (EFF) resigned from the W3C the same day EME was made a web standard. In 2020, EME started being used by Reddit for browser fingerprinting.

All the widely used content decryption modules (CDMs) charge per-browser licensing fees to free (libre) browsers.

There are numerous ways to bypass EME.

See: https://www.defectivebydesign.org/drm-in-web-standards

The Web: Feature Creep

Since the very beginning of the web, browsers have competed over the number of features. Now the W3C has 1337 web specifications, totaling over 100 millions words and new features are still being added all the time.

Thus

- It's impossible to develop a new (non-fork) web browser.

- Google Chrome and Mozilla Firefox have a duopoly on the web and Mozilla strongly relies on Google for funding.

- Competing browser engines like Apple's WebKit fall behind in features.

- It's impossible to fork Firefox and keep up development pace.

The Web: A Security Nightmare

- The web is responsible for over 9,000 CVEs.[1]

- HTML brings CVEs, privacy invasion, and inaccessibility to email clients.[2]

- XSS, insecure DOR, CSRF, unvalidated redirects, etc.

References:
1: https://cve.mitre.org

The Web: A Privacy Nightmare

- URL query strings

- Browser caching

- The DOM property "window.name"

- CSS media queries

- Hidden form fields

- HTTP authentication

- ETags

- User-agent header

- Referer header

- Font fingerprinting

- WebGL

- Do not track

- Content filtering via webextensions

- Canvas fingerprinting

Protocols Similar to The Web

File   Edit   View   Search   Terminal   Help

```
        Welcome to Gopherproject.org!

     You are entering the gopherspace. All visitors are welcome to come here,
     settle down and become an inhabitant of the gopherspace.

     Here is a little introduction for you newcomers:
  [8]  Getting started with gopher. (floodgap)/

-->█ [10] Why is Gopher relevant?

     The fast links to get further in gopherspace:
  [13] The Gopher Lawn (categorized links)/
  [14] Search the gopherspace using Quarry <?>
  [15] Search the gopherspace using Veronica II <?>
  [16] Search using contrition (forthworks)/
  [17] Aggregator of phlog activity on gopherspace. (gopher.black)/

     Common centers of gopher activity:
  [20] sdf.org/
  [21] floodgap.com/
  [22] quux.org/
  [23] bitreich.org/

     Do you want to meet the community?
  [26] Join #gopherproject on libera.chat. <HTML>
  [27] Discuss with us at gopher-project@other.debian.org <HTML>

     Do you want to settle down in gopherspace?
  [30] Ask the nice people at sdf.org for a gopherhole.
     Or setup your own gopher server. Just ask the community for how to.
```

⟨⟩ gemini.circumlunar.space

Project Gemini

Overview

Gemini is a new internet protocol which:

- Is heavier than gopher
- Is lighter than the web
- Will not replace either
- Strives for maximum power to weight ratio
- Takes user privacy very seriously

Resources

➤ **Official Project Gemini news**

➤ Gemini documentation

➤ **Gemini software**

➤ **Known Gemini servers**

Browser Extensions

GNU LibreJS

- The only Webextension (except Haketilo) oriented towards software freedom.

- The first attempt to solve the problem laid out by rms in *the JavaScript trap*.

- Prevents non-free JavaScript from executing in the browser.

- Allows trivial JavaScript to run to reduce website breakage while retaining user freedom.

- Puts the onus on webmasters to identify their JS licenses.

- Supports Firefox-based browsers.

NoScript

- Security suite and content blocker.

- Protects against clickjacking, CSRF, XSS, and MITM.

- Supports Firefox and Chromium-based browsers.

uBlock Origin

- Selective content blocker. Fine-grained control over loading of page elements.

- Includes adware and spyware blacklists, making it suitable for ad blocking.

- Dynamic rule-based filtering. Highly flexible.

- Supports Firefox and Chromium-based browsers, MS edge, and Opera.

uMatrix

- Selective content blocker that shares part of uBlockOrigin's codebase.

- For advanced users.

- Unmaintained.

Greasemonkey

- Allows users to execute custom scripts on websites. Not a content blocker or security suite.

- Combined with a content blocker, Greasemonkey can replace sites' native JavaScript and CSS.

- There are hundreds of user scripts written for Greasemonkey and forks[1]. They can change websites to dark mode, remove ads, mod popular web games, aggregate search results, and more.

- Supports Firefox-based browsers.

Links:
1: https://greasyfork.org

Summary of: https://hydrillabugs.koszko.org/projects/haketilo/wiki/Comparison_with_other_extensions#Greasemonkey
Source license: CC-BY-SA 4.0 or GFDLv1.3+
Source authors: Jahoti, Wojciech Kosior

Violentmonkey

- Similar to and mostly compatible with Greasemonkey.

- Supports Chrome, Firefox 57+, MS Edge, Opera 15+, Vivaldi, QQBrowser, etc.

JShelter

- Increases privacy by wrapping Web APIs used for browser fingerprinting.

- Increases security by preventing microarchitectural vulnerabilities like Spectre/Meltdown.

- Lets users trade off between privacy and usability.

- Highest protection setting creates a homogeneous browser fingerprint like Tor Browser.

- Complementary to content blockers.

- Supports Firefox, Chrome, and Opera.

Haketilo

- Content blocker and user script manager.

- Replaces non-free JS with free, community-controlled scripts. Does not require webmaster cooperation.

- Community-controlled scripts are retrieved from an external centralized repository. The centralized repository software is called "Hydrilla".

- The main objective of Haketilo is software freedom.

- Another priority is promoting lightweight, ethical website design (some libre sites overuse JavaScript).

- Supports Firefox and Chrome-based browsers.

See: https://hydrillabugs.koszko.org/projects/haketilo/wiki

Haketilo Demo

Haketilo's Use Cases

- Block non-free JavaScript.

- Make sites usable without non-free JS.

- Substitute a site's user interface with a more preferred one.

- Provide independent translations of Web pages.

- Increase site accessibility.

- Give users control over the execution of free JavaScript.

- Make heavy sites usable on resource-constrained devices.

- Aggregate content from many websites into one place.

- Add features to webpages.

Haketilo's Social Goals

- Incentivize webmasters to design their sites ethically and redesign unethical websites.

- Reward volunteer contributions (scripts, translations, modifications) that prove useful.

- Give users ultimate control over their web browsing.

Haketilo's Challenges – Uncooperative Site Owners

Website owner insists on keeping non-free JS → custom script incoming.

Webmaster still refuses to remove non-free JS → block all the site's ads.

Webmasters spend time/money constantly redesigning their site to break our custom scripts → initiate cat-and-mouse game[1].

As a negotiation tactic, we can make default replacement JS served by Haketilo allow some free, non-tracking ads.

For users who strongly dislike ads, Haketilo supports scripts that block all ads. We believe users having control over their browser is more important than profits and mustn't be compromised.

1: See Youtube-dl

Haketilo's Challenges - Funding

User-provided translations, custom styling, and usability enhancements:

- NLnet

- Donations (multiple different interests)

- Users who want a particular custom script can put a bounty on it

- Still mainly volunteer-driven

Haketilo's Future Plans

- Site translations

- Port to Manifest v3

- Support for WebKit-based browsers

Haketilo Needs Your Help