Restore Online Freedom!

Mike Gerwitz

20 March, LibrePlanet 2016

Who am I?

- Long-time free software hacker and activist
- GNU volunteer, software evaluator, and part of maintainers team
- Author of GNU ease.js
- 10+ years of web development

Who am I?

- Long-time free software hacker and activist
- GNU volunteer, software evaluator, and part of maintainers team
- Author of GNU ease.js
- 10+ years of web development
  - The Web has changed a lot

Who am I?

- Long-time free software hacker and activist
- GNU volunteer, software evaluator, and part of maintainers team
- Author of GNU ease.js
- 10+ years of web development
  - The Web has **worsened** a lot

“But the Web is great!”

“But the Web is great!”

- The Web has improved *technically*
  - Modern software on the web rivals traditional desktop software

"But the Web is great!"

- The Web has improved *technically*
  - Modern software on the web rivals traditional desktop software
- The truth is: the Web has become alarmingly hostile and freedom-denying toward users

“But the Web is great!”

- The Web has improved *technically*
  - Modern software on the web rivals traditional desktop software
- The truth is: the Web has become alarmingly hostile and freedom-denying toward users
- Freedom trumps all.

Interoperability

- Fundamental building blocks are standardized (W3C)
  - HTML—Describes the document
  - CSS—Describes the document style (appearance)
  - HTTP—Communication protocol

Interoperability

- Fundamental building blocks are standardized (W3C)
  - HTML—Describes the document
  - CSS—Describes the document style (appearance)
  - HTTP—Communication protocol
  - JavaScript—Embed programs in web browser

Interoperability

- Fundamental building blocks are standardized (W3C)
  - HTML—Describes the document
  - CSS—Describes the document style (appearance)
  - HTTP—Communication protocol
  - JavaScript—Embed programs in web browser
- Any software can interact freely
- Threatened in the past: Java applets, ActiveX, Flash, Silverlight, and other proprietary/non-standard software
- Threatened today: Encrypted Media Extensions (EME) Silverlight, and other proprietary/non-standard software

Interoperability

- Fundamental building blocks are standardized (W3C)
  - HTML—Describes the document
  - CSS—Describes the document style (appearance)
  - HTTP—Communication protocol
  - JavaScript—Embed programs in web browser
- Any software can interact freely
- Threatened in the past: Java applets, ActiveX, Flash, Silverlight, and other proprietary/non-standard software
- Threatened today: Encrypted Media Extensions (EME) Silverlight, and other proprietary/non-standard software
- Now the greatest threat is JavaScript

What is JavaScript?

- ECMAScript (Standard: ECMA-262; ISO/IEC 16262)
- Traditionally: Browser-based, client-side scripting language

What is JavaScript?

- ECMAScript (Standard: ECMA-262; ISO/IEC 16262)
- Traditionally: Browser-based, client-side scripting language
- Today: General-purpose programming language
  - Heavily popularized by Node.js

What is JavaScript?

- ECMAScript (Standard: ECMA-262; ISO/IEC 16262)
- Traditionally: Browser-based, client-side scripting language
- Today: General-purpose programming language
  - Heavily popularized by Node.js
- “Assembly language” of the web
  - Subset: asm.js
  - Soon: WebAssembly (WebASM)

The Illusion of Remote Execution

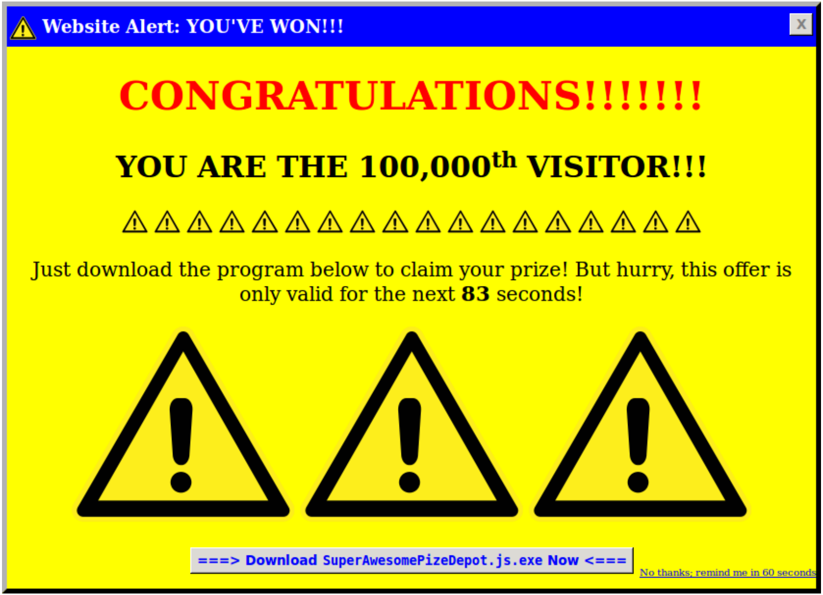
- Looks like the web page is manipulating itself
  - Many consider the web page as a remote resource, not a local copy of that resource

The Illusion of Remote Execution

- Looks like the web page is manipulating itself
  - Many consider the web page as a remote resource, not a local copy of that resource
- JavaScript programs run *on the client*

The Illusion of Remote Execution

- Looks like the web page is manipulating itself
  - Many consider the web page as a remote resource, not a local copy of that resource
- JavaScript programs run *on the client*
- Illusion inhibits consideration of freedoms



⚠ Website Alert: YOU'VE WON!!!

X

CONGRATULATIONS!!!!!!!

YOU ARE THE 100,000th VISITOR!!!

⚠ ⚠ ⚠ ⚠ ⚠ ⚠ ⚠ ⚠ ⚠ ⚠ ⚠ ⚠ ⚠ ⚠ ⚠ ⚠ ⚠ ⚠

Just download the program below to claim your prize! But hurry, this offer is only valid for the next **83** seconds!

===> Download SuperAwesomePizeDepot.js.exe Now <===

No thanks; remind me in 60 seconds

Would you run some random website's program on your computer?

Well, you do.

Covert Ephemeral Software

- Browser downloads and executes arbitrary, often non-free software
  - (Automatically clicks the download button for you!)

Covert Ephemeral Software

- Browser downloads and executes arbitrary, often non-free software
  - (Automatically clicks the download button for you!)
- Most users have no idea this is happening
  - How would they?
  - Most who *do* know don't care.

Covert Ephemeral Software

- Browser downloads and executes arbitrary, often non-free software
  - (Automatically clicks the download button for you!)
- Most users have no idea this is happening
  - How would they?
  - Most who *do* know don't care.
- Ephemeral software

Just Say “Yes!”

Just Say “Yes!”

Advertisers

Just Say “Yes!”

Crackers & Script Kiddies

Just Say “Yes!”

Governments (also Crackers & Script Kiddies)

Just Say “Yes!”

ISPs (Like Comcast)

Just Say “Yes!”

People who want to show off their cool stuff

Just Say "Yes!"...and "Yes!" and "Yes!" and "Yes!"...

- Everyone expects their code to run, always, or there's something wrong with your browser

Just Say "Yes!"…and "Yes!" and "Yes!" and "Yes!"…

- Everyone expects their code to run, always, or there's something wrong with your browser
- But most browsers today give you a binary choice:
  1. No—disable JavaScript
  2. Yes—run everything!

Just Say "Yes!"...and "Yes!" and "Yes!" and "Yes!"...

- Everyone expects their code to run, always, or there's something wrong with your browser
- But most browsers today give you a binary choice:
  1. No—disable JavaScript
  2. Yes—run everything!
- You aren't just running that site's JavaScript

How Do We Install Software (Usually)?

How Do We Install Software (Usually)?

- Explicitly.

How Do We Install Software (Usually)?

- Explicitly.
  - GNU/Linux—Usually through a package manager
  - Source distribution (tarballs/etc)
  - Binaries (especially on proprietary operating systems)

How Do We Install Software (Usually)?

- Explicitly.
  - GNU/Linux—Usually through a package manager
  - Source distribution (tarballs/etc)
  - Binaries (especially on proprietary operating systems)
- There is a conscious effort made by the user

Software Signing

- Package manager (should!) verify signatures of package maintainers
- Many projects distribute detached signatures for manual verification

Software Signing

- Package manager (should!) verify signatures of package maintainers
- Many projects distribute detached signatures for manual verification
- No such thing exists for the Web

Software Signing

- Package manager (should!) verify signatures of package maintainers
- Many projects distribute detached signatures for manual verification
- No such thing exists for the Web
- I wish I had time to discuss this

Software Signing

- Package manager (should!) verify signatures of package maintainers
- Many projects distribute detached signatures for manual verification
- No such thing exists for the Web
- I wish I had time to discuss this
  - What not to do: Firefox refuses to install/run addons that are not signed by Mozilla
  - No "walled gardens"

Everyone is flocking to the web

Effortless Distribution

- Software is downloaded *automatically*—guaranteed!
- It's so easy for you—just click here!

Effortless Distribution

- Software is downloaded *automatically*—guaranteed!
- It's so easy for you—just click here!
- Cross-{device,platform}!

Effortless Distribution

- Software is downloaded *automatically*—guaranteed!
- It's so easy for you—just click here!
- Cross-{device,platform}!
- Automatic updates! We know best! (Ephemeral software)
  - Your browser is effectively a backdoor

Effortless Distribution

- Software is downloaded *automatically*—guaranteed!
- It's so easy for you—just click here!
- Cross-{device,platform}!
- Automatic updates! We know best! (Ephemeral software)
    - Your browser is effectively a backdoor
- We control your data and computing (SaaSS)!

Effortless Distribution

- Software is downloaded *automatically*—guaranteed!
- It's so easy for you—just click here!
- Cross-{device,platform}!
- Automatic updates! We know best! (Ephemeral software)
    - Your browser is effectively a backdoor
- We control your data and computing (SaaSS)!
- So easy to spy on you!

We can finally get rid of that pesky little bit of control you had left over your physical computing!

The Web Is Easy

- Low Barrier To Entry
- Huge number of libraries and tools for web development
- Even server/desktop software [substitutes] using web libraries

This is great for software freedom.

But we have a problem.

Effortless *Proprietary* Software

- We have a licensing problem.

Effortless *Proprietary* Software

- We have a licensing problem.
- Permissive licensing *enables* proprietary software

The Web Scene

- Startups (e.g. YCombinator-funded)
- Large corporations
- New programmers, new culture

The Web Scene

- Startups (e.g. YCombinator-funded)
- Large corporations
- New programmers, new culture
  - “Open Source”
  - “Copyleft” is a dirty word

The Web Scene

- Startups (e.g. YCombinator-funded)
- Large corporations
- New programmers, new culture
  - “Open Source”
  - “Copyleft” is a dirty word
- Peer pressure

Partial freedom isn't freedom!

No Such Thing As Neutral On Freedom

- Permissive (pushover) licenses are sometimes used in an attempt to be neutral

No Such Thing As Neutral On Freedom

- Permissive (pushover) licenses are sometimes used in an attempt to be neutral
- But there's no such thing as neutrality!
- Neutral on freedom $\equiv$ allowing erosion of freedoms

No Such Thing As Neutral On Freedom

- Permissive (pushover) licenses are sometimes used in an attempt to be neutral
- But there's no such thing as neutrality!
- Neutral on freedom $\equiv$ allowing erosion of freedoms
- Copyleft is *essential* for a free Web
  - If you use pushover licenses, you're helping to push over everyone else (but perhaps not intentionally)

If you value freedom, talk about Free Software!

If you value freedom, choose copyleft!

Freedom is only as good as your ability to exercise it

Maybe free...?

- Almost always no license information or link to source code
- Might also load non-free code as a separate program (e.g. Google Analytics spyware)

Corresponding Source Code

- Where can I find the corresponding source?

Corresponding Source Code

- Where can I find the corresponding source?
- Minified code is not source code!
- “The ’source code’ for a work means the preferred form of the work for making modifications to it. ’Object code’ means any non-source form of a work.”

```
add:function(a,b,c,d,e){var
f,g,h,i,j,k,l,m,o,p,q,r=N.get(a);if(r){c.handler&&(f=c,c=f.handler,
e=f.selector),c.guid||(c.guid=n.guid++),(i=r.events)||(i=r.events={}),
(g=r.handle)||(g=r.handle=function(b){return"undefined"!=typeof
n&&n.event.triggered!==b.type?n.event.dispatch.apply(a,arguments):void
0}),b=(b||"").match(G)||[""],j=b.length;while(j--)h=fa.exec(b[j])||[],
o=q=h[1],p=(h[2]||"").split(".").sort(),o&&(l=n.event.special[o]||{},
o=(e?l.delegateType:l.bindType)||o,l=n.event.special[o]||{},k=n.extend(
{type:o,origType:q,data:d,handler:c,guid:c.guid,selector:e,needsContext:
e&&n.expr.match.needsContext.test(e),namespace:p.join(".")},f),(m=i[o])
||(m=i[o]=[],m.delegateCount=0,l.setup&&l.setup.call(a,d,p,g)!==!1||
a.addEventListener&&a.addEventListener(o,g)),l.add&&(l.add.call(a,k),
k.handler.guid||(k.handler.guid=c.guid)),e?m.splice(m.delegateCount++,0,k)
:m.push(k),n.event.global[o]=!0)}}
```

https://code.jquery.com/jquery-2.2.1.min.js

LibreJS

Free JavaScript in your browser

LibreJS

- Blocks execution of non-free JavaScript
- JavaScript should contain `@license` tags
- Hashes of common libraries recognized as free
- Web Labels map scripts to corresponding source code

.

Submit Patches To Projects!

- Patch your own projects!
- Submit patches to add headers to projects
- Patches for minifiers (e.g. UglifyJS)
- Patches for programs that combine sources (e.g. Browserify)
- For Node.js programs: can pull license right out of `package.json`

Replacing Programs

- How do you replace a program (free or non-) on your computer?

Replacing Programs

- How do you replace a program (free or non-) on your computer?
  - You just do.

Replacing Programs

- How do you replace a program (free or non-) on your computer?
  - You just do.
- How do you replace a program (free or non-) on the Web?

Is that a trick question?

Replacing Programs—The Best Case

1. The program is entirely packaged in its own JavaScript file(s)
2. All dependencies are packaged with or alongside it

Replacing Programs—The Best Case

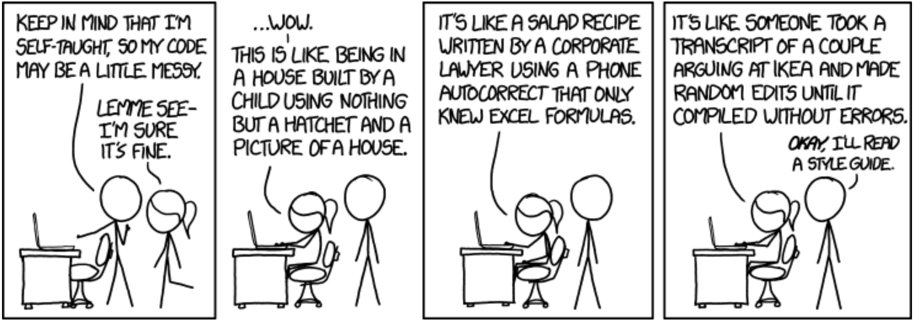
1. The program is entirely packaged in its own JavaScript file(s)
2. All dependencies are packaged with or alongside it

Solution:

1. Block program scripts
2. Block undesirable dependencies
3. Inject replacement program

Replacing Programs—The Realistic Case

1. The program is actually a number of independent programs (scripts) doing different things
2. Some scripts are packaged in one or more JavaScript files
3. Some are inline `<script>` tags scattered throughout
4. Some are inline JavaScript fragments in HTML attributes


KEEP IN MIND THAT I'M SELF-TAUGHT, SO MY CODE MAY BE A LITTLE MESSY.
LEMME SEE– I'M SURE IT'S FINE.
...WOW.
THIS IS LIKE BEING IN A HOUSE BUILT BY A CHILD USING NOTHING BUT A HATCHET AND A PICTURE OF A HOUSE.
IT'S LIKE A SALAD RECIPE WRITTEN BY A CORPORATE LAWYER USING A PHONE AUTOCORRECT THAT ONLY KNEW EXCEL FORMULAS.
IT'S LIKE SOMEONE TOOK A TRANSCRIPT OF A COUPLE ARGUING AT IKEA AND MADE RANDOM EDITS UNTIL IT COMPILED WITHOUT ERRORS.
OKAY, I'LL READ A STYLE GUIDE.

"Code Quality" by Randall Munroe (http://xkcd.com/1513/) CC BY-NC 2.5

Solution?

Write a presentation complaining about it.

Blocking Scripts—Entirely

- Can disable JavaScript entirely
- But then you can't run replacements

Blocking Scripts—NoScript

- NoScript is effective at blocking scripts per (sub-)domain
- Careful—some domains serve lots of scripts (e.g. CDNs)
- Some sites require you to trust many different domains

Blocking Scripts—LibreJS

- Blocks non-free JS with a high level of granularity
- Recognizes JavaScript files (or parts of), script tags, inline scripts in attributes, etc.
- But doesn't block scripts recognized as free
    - So that we can replace specific programs
    - Free software can do bad things too

Injecting Scripts—Greasemonkey

- Expat / Modified BSD / MPL 1.1
- Inject user-defined scripts into web pages
- `http://openuserjs.org`, `http://greasyfork.org`, `http://userscripts-mirror.org`

Injecting Scripts/Styles—dotjs

- MPL 1.1
- Inject user-defined scripts from `~/.js`
- Inject user-defined styles from `~/.css`

Replacing Scripts—What We Need

- Something your grandparents can use!

Replacing Scripts—What We Need

- Something your grandparents can use!
- Package changes:
  - Install replacement software like you'd install an addon
  - Automatically block scripts with granularity of LibreJS
  - Automatically inject scripts
  - Verify package signatures
  - Update mechanism
- Ideally: add that support to LibreJS

But all this is just a kluge.

Service as a Software Substitute (SaaSS)

- Also called “Software as a Service” (SaaS)
- “Who Does That Server Really Serve?” essay by rms

Service as a Software Substitute (SaaSS)

- Also called “Software as a Service” (SaaS)
- “Who Does That Server Really Serve?” essay by rms
- You simply cannot modify a program running on someone else’s server

Service as a Software Substitute (SaaSS)

- Also called “Software as a Service” (SaaS)
- “Who Does That Server Really Serve?” essay by rms
- You simply cannot modify a program running on someone else’s server
- SaaS—Shackles as a Service

There Is No Cloud

- Talking about the “cloud” leads to cloudy judgment

There Is No Cloud

- Talking about the “cloud” leads to cloudy judgment
- It’s just someone else’s computer—lots of them
- Difference: elasticity—servers/resources brought online only when needed

There Is No Cloud

- Talking about the “cloud” leads to cloudy judgment
- It’s just someone else’s computer—lots of them
- Difference: elasticity—servers/resources brought online only when needed
- We can’t talk about freedoms using terms that cloud understanding

There Is No Cloud

- Talking about the “cloud” leads to cloudy judgment
- It’s just someone else’s computer—lots of them
- Difference: elasticity—servers/resources brought online only when needed
- We can’t talk about freedoms using terms that cloud understanding
  - May as well call it the “fog”

{Desktop,Web} Software

- To regain control, must host it yourself
- This means running the entire stack (server and client) on your own computer

{Desktop,Web} Software

- To regain control, must host it yourself
- This means running the entire stack (server and client) on your own computer
- Unweb

Unwebbing

- Cannot unweb a partially (or non-) free stack

Unwebbing

- Cannot unweb a partially (or non-) free stack
  - Webserver
  - Programs/scripts that it runs to generate output
  - Script dependencies
  - Client

The Affero GPL Doesn't Solve All Problems

- If users can access a *modified* version of the program over a network, that they must have access to the source code
- But even if you have the source code, you can't modify it on the server that's using it

The Affero GPL Doesn't Solve All Problems

- If users can access a *modified* version of the program over a network, that they must have access to the source code
- But even if you have the source code, you can't modify it on the server that's using it
- AGPL'd programs can be used in a pipeline with proprietary programs
- Doesn't cover works that the AGPL'd program might invoke externally

Truly Unwebbed

1. Download unwebbed package

Truly Unwebbed

1. Download unwebbed package
2. Invoke unwebbed program `./foo`

Truly Unwebbed

1. Download unwebbed package
2. Invoke unwebbed program `./foo`
3. Package creates an isolated environment (possibly with no network access)
4. Package mounts persistent storage
5. Package starts all necessary services
6. Package invokes client (e.g. web browser, possibly in “app” mode / SSB)

Truly Unwebbed

1. Download unwebbed package
2. Invoke unwebbed program `./foo`
3. Package creates an isolated environment (possibly with no network access)
4. Package mounts persistent storage
5. Package starts all necessary services
6. Package invokes client (e.g. web browser, possibly in "app" mode / SSB)
7. Program terminates
8. All services are stopped
9. Environment is destroyed (but not persistent storage)

Web software works just as desktop software

End Goal: Just Run `./foo`

Mike Gerwitz
mtg@gnu.org

Interested in submitting your project to GNU?
https://www.gnu.org/help/evaluation.html

https://gitlab.com/mikegerwitz/online-freedom