

Privacy Protections in Brave Browser


Pete Snyder
pes@brave.com



Outline

- Quickies (i.e., everything that's not below)
- Fingerprinting Protections
- Storage Policy
- Longer Term Projects


Outline

- Quickies (i.e., everything thats not below) 
- Fingerprinting Protections
- Storage Policy
- Longer Term Projects

Shields

Shields **UP** for this site

If a site appears broken, try shields down

 github.com

3 Items blocked

✓ 3

Cross-site trackers blocked

✓ 0

Connections upgraded to HTTPS

✓ 0

Scripts blocked

Cross-site cookies blocked

✓ 0

Cross-site fingerprinting blocked

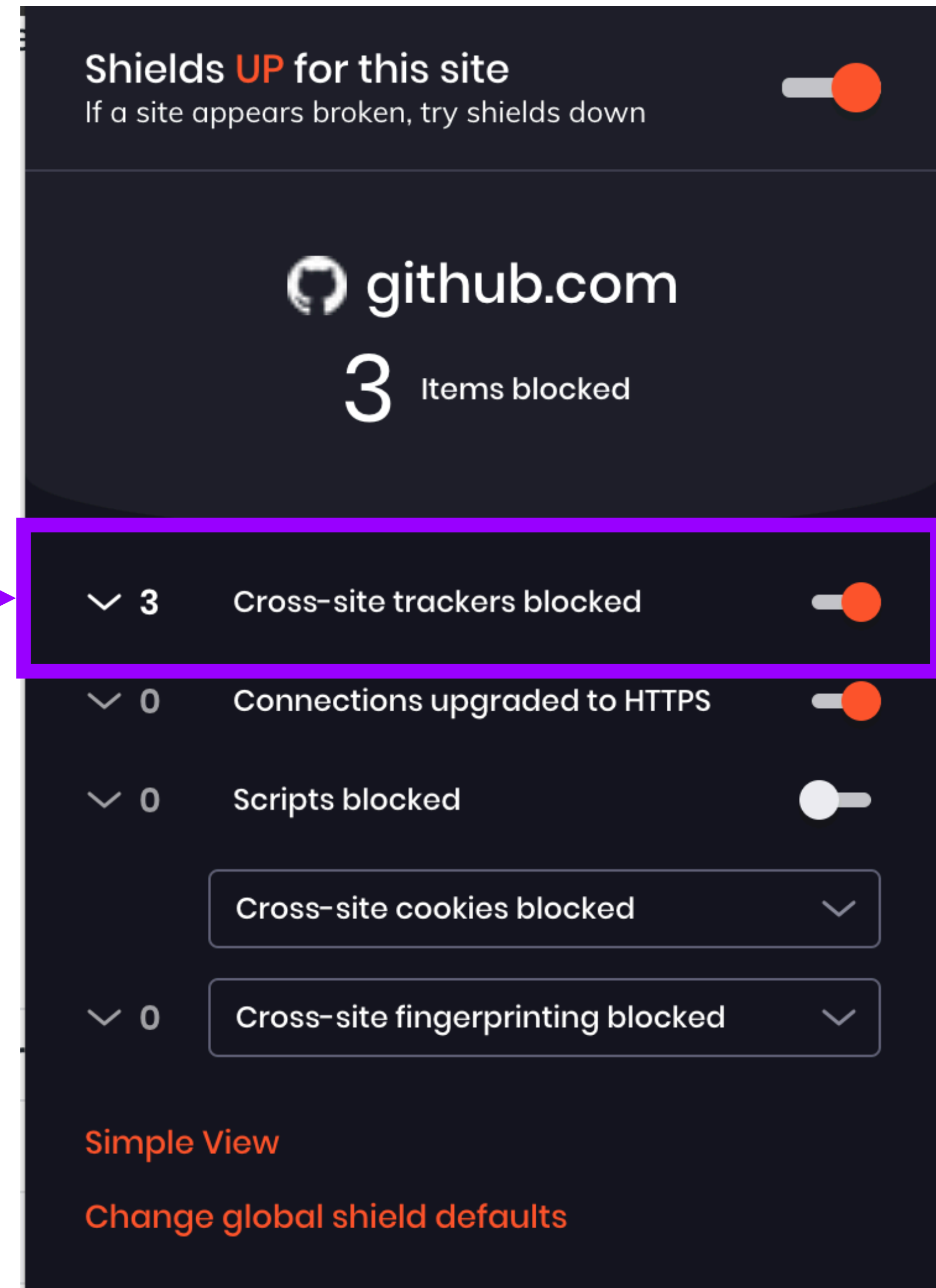
Simple View

Change global shield defaults

Shields

Filter Lists

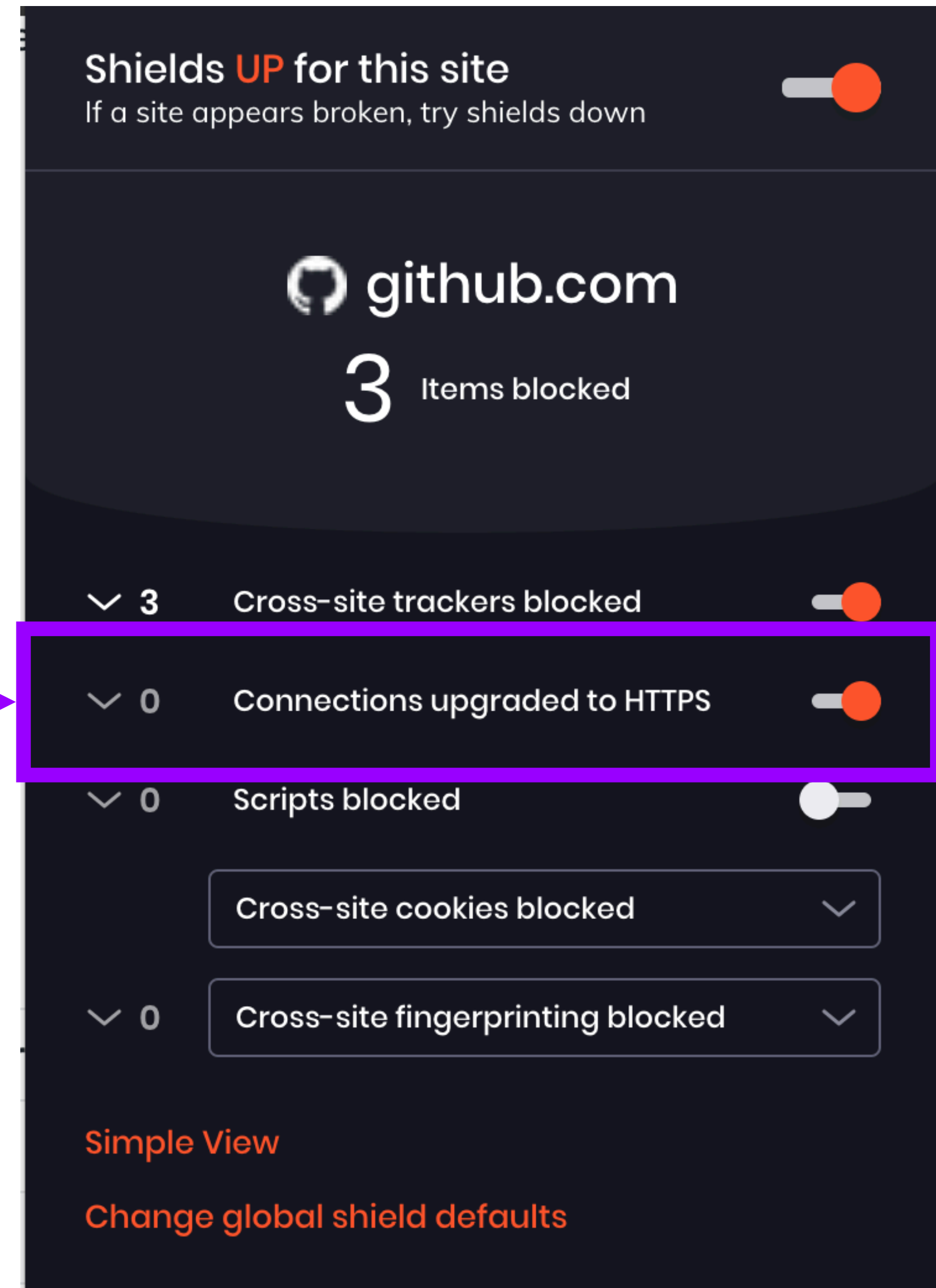
- EasyList / EasyPrivacy
- Disconnect
- uBlock Origin (w/ replacements)
- Brave originals
- Fanboy Notifications
- [adblock-rust \(rust and node\)](#)
- [SlimList for iOS](#)



Shields

HTTPS Upgrades

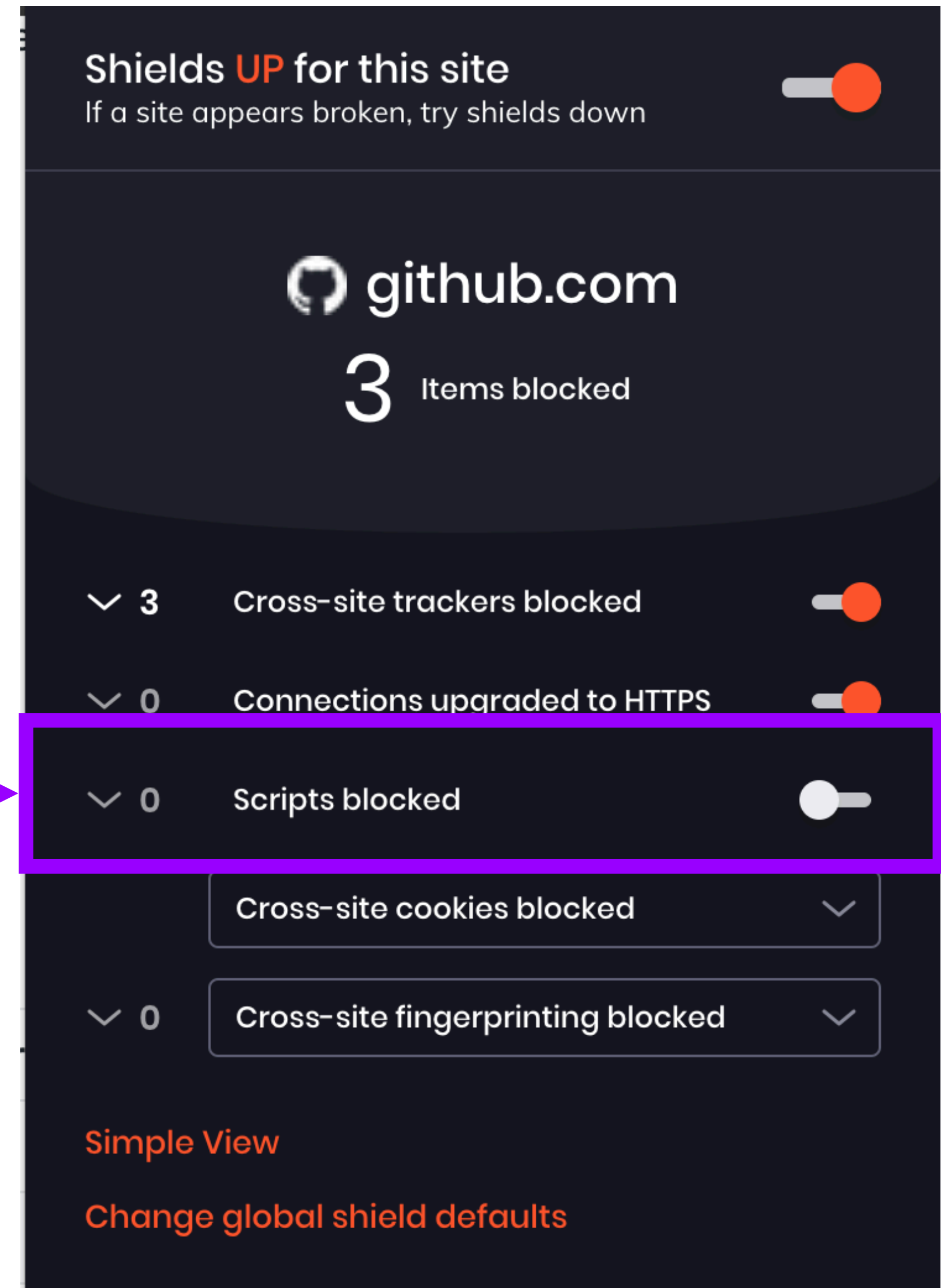
- HTTPSEverywhere
- <https://www.eff.org/https-everywhere>



Shields

Script Blocking

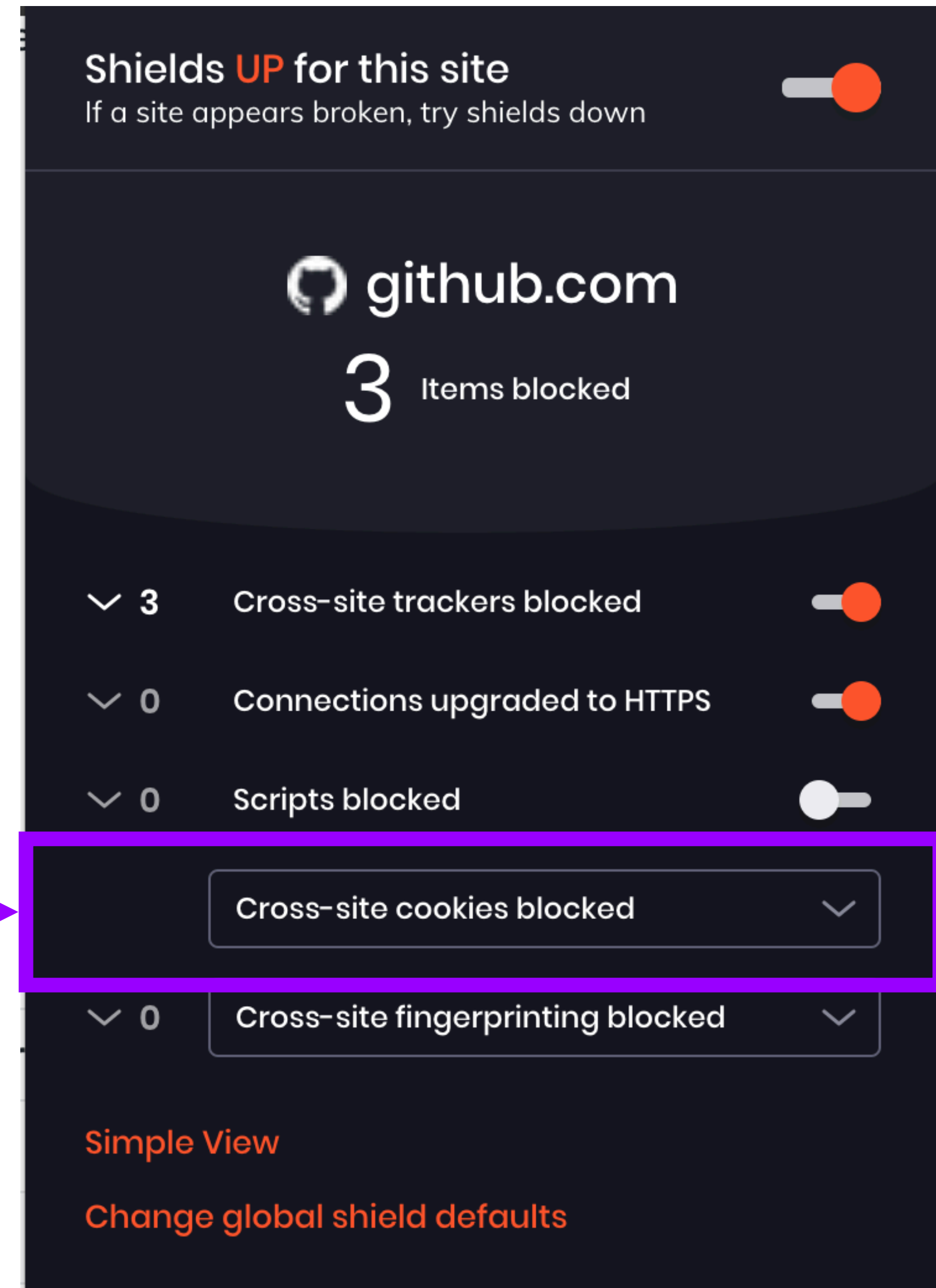
- Disable all script
- Default off



Shields

Cookie / Storage

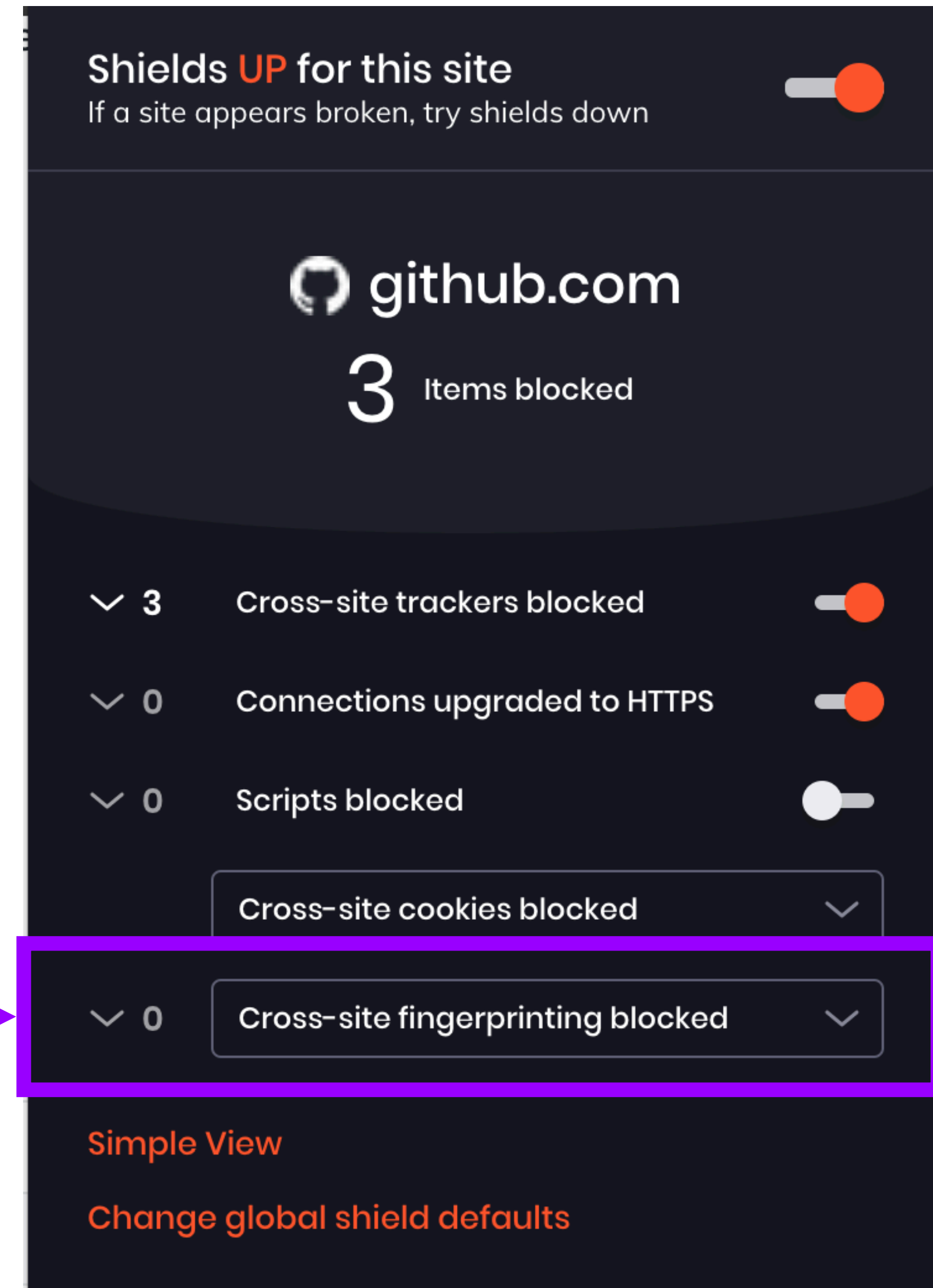
- Default: no 3p storage
- Network: don't send cookies
- JS Storage: no script storage
- Relative to eTLD+1 of top frame
- JS cookies: max 7 day lifetime



Shields

Fingerprinting protections

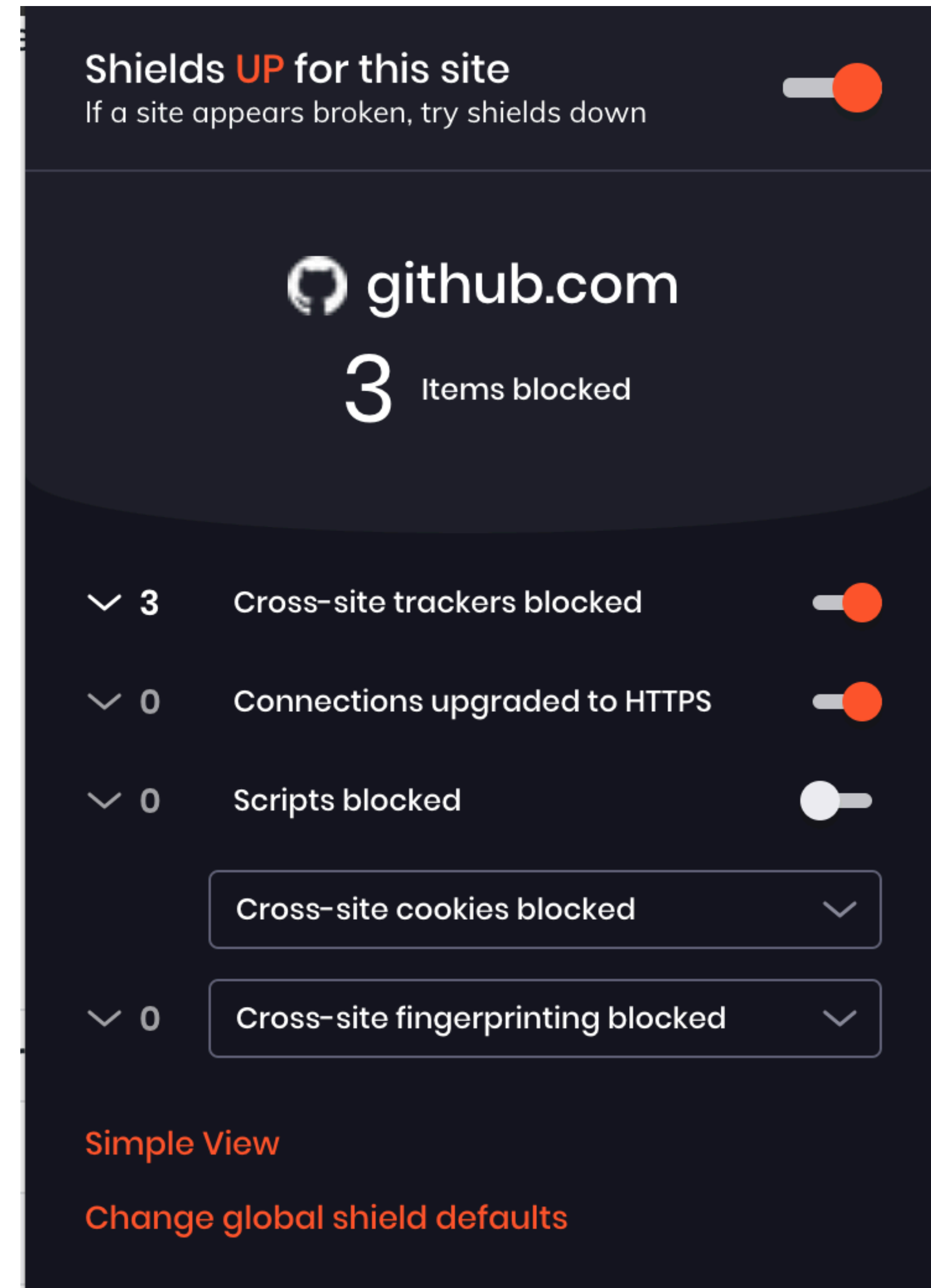
- More on that in a second...



Shields

Referrer policy

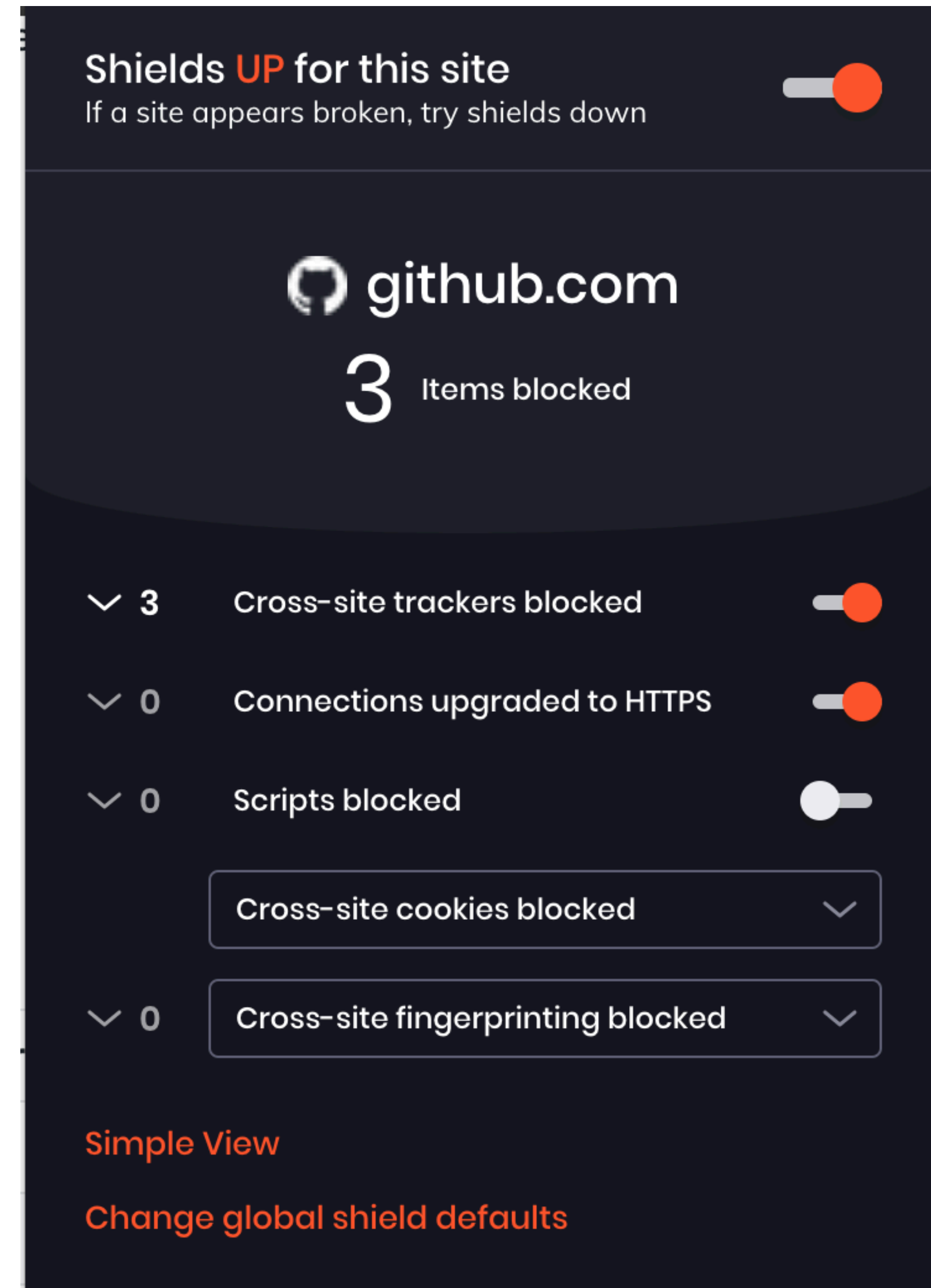
- Cross-site request:
Report requested origin
- Cross-site navigation:
Always omitted
- Same-site request:
Do what Chromium does




Shields

Grab bag

- Query param filtering
- Exception lists
 - Cookies
 - Referrer
 - Fingerprinting
 - Filter list exceptions
 - `https://github.com/brave/*`



Outline

- Quickies (i.e., everything that's not below)
- Fingerprinting Protections 
- Storage Policy
- Longer Term Projects

Fingerprinting Protections

Current policy

- Default: 3p frames only
- Disable parts of the following APIs
 - Canvas
 - WebAudio
 - WebRTC
 - WebGL
 - SVG

Fingerprinting Protections

Privacy through randomization

- **Farbling:** minor randomization to confuse fingerprinters
- “...anonymization through generalization does not sufficiently protect anonymity”
- Different fingerprint, for every site, for every session

Browsing Unicity: On the Limits of Anonymizing Web Tracking Data

Clemens Deußner
Chair of Privacy and Security
TU Dresden, Germany
Email: clemens.deusser@tu-dresden.de

Steffen Passmann
INFOnline GmbH
Berlin, Germany
Email: SPassmann@infonline.de

Thorsten Strufe
Karlsruhe Institute of Technology
Centre for Tactile Internet, TU Dresden
Email: strufe@kit.edu

Abstract—Cross domain tracking has become the rule, rather than the exception, and scripts that collect behavioral data from visitors across sites have become ubiquitous on the Web. The collections form comprehensive profiles of browsing patterns and contain personal, sensitive information. This data can easily be linked back to the tracked individuals, most of whom are likely unaware of this information’s mere existence, let alone its perpetual storage and processing. As public pressure has increased, tracking companies like Google, Facebook, or Baidu now claim to anonymize their datasets, thus limiting or eliminating the possibility of linking it back to data subjects.

In cooperation with Europe’s largest audience measurement association we use access to a comprehensive tracking dataset to assess both identifiability and the possibility of convincingly anonymizing browsing data. Our results show that anonymization through generalization does not sufficiently protect anonymity. Reducing unicity of browsing data to negligible levels would necessitate removal of all client and web domain information as well as click timings. In tangible adversary scenarios, supposedly anonymized datasets are highly vulnerable to dataset enrichment and shoulder surfing adversaries, with almost half of all browsing sessions being identified by just two observations. We conclude that while it may be possible to store single coarsened clicks anonymously, any collection of higher complexity will contain large amounts of pseudonymous data.

I. INTRODUCTION

Tracking has become pervasive on the Web. More than four out of five sites employ behavioral tracking, some on a large scale, with dozens of different scripts tracking their users at the same time [1], [2]. The average page access on the Web is tracked by eight scripts, today¹. Some sites employ local tracking to optimize their user experience, others use legitimate scripts to perform reliable audience and reach measurements. The majority of trackers, however, is used to

and store entire browsing profiles or sequences of observed visits as so called click traces in vast tracking databases [5].

The usual reflex to inquiry is the statement that this data was anonymized, usually through generalization (truncation, or “coarsening”) of stored attributes, such as IP addresses [6] or through differential privacy techniques. Differential privacy is a powerful tool which delivers provable privacy guarantees. In this paper we will not examine practical implementations of differential privacy, but in the past they have often been either misused (eg. through lack of a properly enforced privacy budget) or have lead to severely restricted utility [7], [8]. Instead we will focus on examining generalization techniques.

Whether and how generalized data can be de-anonymized has been extensively researched by Narayanan et al. and others in the past [9], [10], [11]. Nevertheless, anonymization through generalization techniques not only continue to be used, but the industry in which they are applied plays an increasingly ubiquitous role in modern society. Their position is that these results are not universally valid and do not apply to other methods of generalization on different kinds of data. In this work we will attempt to close that gap as it relates to web tracking data. More specifically, both structural information, such as position in a social graph, as well as pseudonyms in general have been shown to be highly identifying. In recognition of this fact, modern privacy regulations like the European GDPR specifically enforce restrictions such as obtaining informed consent before allowing collection and processing of pseudonymous data. Storing a client browsing session as a sequence of website visits with very general page and client information, as audience measurement providers often do, appears to avoid these restrictions.

Fingerprinting Protections

Farbling implementation

- Generate random seed on session start
- $\text{HMAC256}(\text{session seed} \parallel (\text{eTLD}+1)) \rightarrow \text{eTLD}+1 \text{ session seed}$
- Derive all farbled values from eTLD+1 seed
- Fames use top level eTLD+1 seed
- Sites get consistent results during each session

Fingerprinting Protections

Farbling levels

- **Off**
No modifications; do Chromium
- **Default**
 - Mix noise with underlying values
 - Main goal: make recovery of “true” values difficult
 - Secondary goal: make fingerprinting look very different from benign use
- **Maximum**
 - No “true” values
 - Only farbled values

Fingerprinting Protections

Farbling canvas

	Default	Maximum
Serialization / Readback <ul style="list-style-type: none">- CanvasRenderingContext2d.getImageData- HTMLCanvasElement.toDataURL- HTMLCanvasElement.toBlob- OffscreenCanvas.convertToBlob	Flip some low order bits, offsets from seed & canvas contents	Random values derived from seed
Content Querying <ul style="list-style-type: none">- CanvasRenderingContext2d.measureText- CanvasRenderingContext2d.isPointInPath- CanvasRenderingContext2d.isPointInStroke	-	Return bottom values

Fingerprinting Protections

Farbling web audio


	Default	Maximum
Serialization / Readback <ul style="list-style-type: none">- AnalyserNode.getTimeDomainData- AnalyserNode.getFloatTimeDomainData- AnalyserNode.getBytesFrequencyData- AnalyserNode.getFloatFrequencyData- AudioBuffer.getChannelData	Change output volume by [0,-.01] under eTLD+1 session seed	Randomly generate low amplitude white noise under eTLD+1

Fingerprinting Protections

Other first round farbed features

- `MediaDevices.enumerateDevices`
- `WebGL2RenderingContextBase.getParameter`
- `WebGLRenderingContext.get*`
- `WEBGL_debug_renderer_info`
- `NavigatorPlugins.plugins`
- `NavigatorID.userAgent`
- `XRSystem.isSessionSupported`
- Issue [#8787](#)

Outline

- Quickies (i.e., everything that's not below)
- Fingerprinting Protections
- Storage Policy 
- Longer Term Projects

Storage Policy

Current policy


- Limit JS cookie lifetime (7 days)
- No storage in 3p frames (eTLD+1 frames get storage)
- No cookies for 3p requests
- Short exception list

Storage Policy

Incoming policy

- **Short term:** Fix Chromium quirks (e.g., {local,session}Storage throws)
- **Medium term:** 3p frames get frame lifetime storage
- **Long term:** Storage Access API escape valve
- **Exploring:**
 - Garbage collecting JS storage
 - 1p cookie jar restrictions ([PageGraph](#) + taint crawls)

Outline

- Quickies (i.e., everything that's not below)
- Fingerprinting Protections
- Storage Policy
- Longer Term Projects 

Longer Term Projects

Grab bag / Summer plans

- **Link decoration protections**
PageGraph + taint crawls to build query filters
- **Programatic generation of resource replacements**
PageGraph + static JS analysis to build uBO style replacements
- **Farble additional endpoints**
Based on reactions to v1
- **Font fingerprinting protections**
Crawls + standards work

The End / Summary

- Farbling fingerprinting protections
- Ephemeral frame storage
- Filter list improvements
- Bounce and link decoration protections

