Recommendations for Software to Mitigate Long-term Abuse on Wikimedia

Trevor Bolliger, WMF Product Manager, March 1, 2019

Background

Long-term abuse

A small handful of malicious actors have made harassing or manipulating Wikipedia and its users their hobby. These users sometimes take pride in making themselves known and generally are patient enough to wait for blocks to expire. They take pleasure in gaming the system and the majority of their abuse occurs while logged-in: pushing their agenda in article pages, creating attack usernames, leaving inappropriate messages, stalking, etc.[1]

Blocking

Currently, MediaWiki software allows permissioned users to stop a person from performing actions (e.g. editing, creating pages, messaging other users) by blocking their username, IP address, or IP range.[2] Block evasions are simple on Wikimedia wikis: User accounts only requirement a unique username, password, and solving a captcha. IP addresses and IP ranges can also easily be changed or spoofed. These are low hurdles.

Block evasions and sockpuppetry

Competent malicious individuals commonly evade blocks to continue disrupting wikis with vandalism, harassment, spam, and other forms of misconduct. There are workflows and tools used to detect block evasions (often called “sockpuppetry”) but improved software could mitigate the damage and deter abusers.

[1] https://en.wikipedia.org/wiki/Wikipedia:Long-term_abuse

[2] https://www.mediawiki.org/wiki/Manual:Block_and_unblock

This document

A common anti-abuse tactic is to identify the exact device a malicious individual uses and prevent them from further participating from that device, even if they change their IP address. This document includes notes from internal and external discussions, industry readings, technical investigations, and a legal evaluation about potential software interventions, primarily focused around device blocking. At the end of the document is our recommendations for next steps.

What is the problem we are trying to solve?

Being a significant presence on the internet, Wikipedia and other Wikimedia wikis have become the target of disruption, notably vandalism, spam, disinformation, and harassment. There are many systems in place to mitigate easy-to-identify content and user manipulation, including edit rollback, page protection and user blocking.[3]

Unfortunately, some malicious individuals operate on a larger and longer scale than average and are colloquially known as "long-term abusers." They are the small handful who have made a hobby of gaming the system to harass or manipulate Wikipedia. Their recurring and unrelenting presence is a detractor from volunteer and staff Wikimedian productivity, can lead to burnout,[4] and in the case of harassment can result in serious offline harm.

Long-term abusers can be generally sorted into three categories of motivation:

Vandals
: Disrupt content for the thrill and humor

Agenda pushers
: Manipulate content for financial, ideological, or political reasons.

Harassers
: Have a vendetta to terrorize and abuse specific individuals.

[3] There is always a need for additional tools in the never-ending 'arms race' against spam and vandalism. This document is specifically about mitigating long-term abuse.

[4] https://en.wikipedia.org/wiki/Wikipedia:Admin_burnout_and_meltdown

Long-term abusers may perform their manipulation while logged-out, but a majority of this abuse occurs while logged-in as a means of flying under the radar to bypass the typical anti-abuse safeguards. They know the rules enough to abide by them enough to slip through the cracks.[5]

While agenda pushers never want to be noticed, vandals' and harassers' goal is to be noticed. The problem is not to identify them from amongst the good-faith contributors, rather to prevent them from returning to cause their harm. Long-term abusers are sophisticated or patient enough to evade or wait out a block.

What does success look like?

In brief, we want fewer long-term abusers to cause harm on our wikis. Preventative measures are preferred but a comprehensive suite of tools will likely be required. Some potential measurements for success for this project — devoid of any baselines or feasibility of measurement — include:

- Incidents of long-term abuse are decreased
    - Fewer WMF + volunteer hours are spent addressing long-term abuse cases
    - *N%* of known abusers stop returning
    - Decrease in sockpuppet blocks
    - Decrease in time between edit + sockpuppet block
    - Decrease in attack usernames suppressed
    - etc.
- Targets of long-term abusers feel safer
    - Users who interact with long-term abusers are retained
    - Users who utilize the Mute features are retained
    - Potentially a qualitative measurement
    - etc.

[5] Based on discussion with WMF Trust & Safety, January 2019

Device blocking

Preface

There will never be a silver bullet to prevent all highly motivated abusers from evading our technical intervents. We still we owe it to our volunteers and our staff to find some solutions that can reduce the harm and damage that occurs from these nefarious individuals. 'Perfect' is the enemy of the good.

Conception

The concept "block by device" has been suggested in the Wikimedia ecosystem over the years.[6] On a high level, device blocking sounds like a promising suggestion: capture enough data about the user's device and allow admins[7] to set blocks that only prevent edits coming from that exact device. It's a common tactic for other major web platforms.

The data captured would be encrypted and deleted in accordance to the privacy policy and would never be surfaced in CheckUser or to any other volunteer or staff position. Device blocks would more-or-less work how existing username or IP blocks work in terms of configuration, logging, and user experience.

Despite the successful adoption of the iOS and Android apps, the vast majority of edits come from web, most notably desktop. After discussing with other technology leaders within the Wikimedia Foundation and external anti-abuse experts, we pursued "fingerprinting" as a general approach.

Technical research on Fingerprinting Libraries and Services

Alex Ezell, January 2019

First, this is the requisite definition of "device fingerprinting" which is also known as "browser fingerprinting" though there are some differences. In short, the idea is to use characteristics of

[6] https://phabricator.wikimedia.org/T100070

[7] It would be possible to build a type of block that would only be available to Stewards, CheckUsers, and/or WMF staff. For the purposes of this document, implementation details are unessential and intentionally omitted.

the software, network, and user to identify them as unique among all visitors. Most understanding of this practice is that it is passive and generally invisible to the user as contrasted against something like a browser cookie which a user can inspect and manage. This makes it a definite privacy issue. The EFF takes a position as to how GDPR will impact this kind of fingerprinting. In short, regardless of how one hashes or encrypts a fingerprint to hide the PII that is used to calculate the fingerprint, the resulting fingerprint is itself PII and is subject to all considerations we might give to email addresses or other PII.

Fingerprinting works because of a mathematical concept called *entropy*. The EFF lays out a lot of details on their website. Simply put, entropy is a measure of how many "bits" of information are required to uniquely identify a given device/browser. To be completely unique across the global population, a system would need around 33 bits of entropy. Most of the fingerprinting systems provide something between 12 and 18 bits of entropy. Given the limited amount of the population that has access to the Internet, 18 bits is considered to be pretty close. However, it's not enough to be mathematically certain. When two devices/browsers are considered as the same by any fingerprinting system, this is called a collision.

Therefore, the holy grail of fingerprinting is a lack of collision. That is, what algorithmic combination of factors can uniquely identify a specific browser/user? Tools like AmIUnique? help to demonstrate and explain the uniqueness of fingerprints. Most of the open source libraries publish no data about their efficacy with regards to uniqueness. Mozilla has published some data from the EFF about uniqueness. Without these kinds of measurements of entropy or uniqueness, we cannot be deterministic about how good our fingerprinting is.

To minimize these collisions, it might be possible to combine an idea like collision-resistant IDs with a fingerprint into a hash that is more unique than a fingerprint alone. There would need to be some investment and research into truly globally-unique IDs to support the breadth of users visiting the wiki projects.

As far as implementation, we would likely choose one of the open-source Javascript libraries readily available. This means that the fingerprint would be calculated on the client-side and sent with each request to our servers. We would then need to compare the fingerprint against

our database of existing fingerprints to determine how we should handle that request. It's possible we could limit this to only the login path or some other subset of features like saving an edit.

Given that the fingerprinting code would be client-side, it would be trivial to prevent its execution by disabling Javascript. As our services are generally written to support usage without Javascript enabled, this would be an easy workaround. A more sophisticated attacker could also modify the code in their client to fake a fingerprint or mimic some other fingerprint.

Persistent cookies

An alternative to browser fingerprinting could be to set a persistent cookie with an indefinite expiration. We would then track actions (edits, etc.) by the id of that persistent cookie and users with the Check User privilege would be able to issue blocks by that cookie id. This method has the benefit of not having any collateral damage, as a single cookie can only be associated with one device, but the mechanism is also easy to evade (clearing your cookies would generate a new id for you).

Estimated development time

For either fingerprinting or cookie blocks, the team estimates 12-18 months of continuous software development for a 3-4 person team. This project would essentially require an entire rewrite of MediaWiki's blocking work. At the time of this estimate, the Anti-Harassment Tools team is intimately familiar with this domain, having recently spent six months working on partial blocks.

Wikimedia Foundation Legal Department consultation

Aeryn Palmer, February 2019

The Anti-Harassment Tools team consulted with the Legal department while they researched various potential avenues and developed this report. We will remain in touch with them throughout design, development, and deployment of their chosen solution.

Recommendation

Avoid device blocking

In short, device bocking poses a high development cost and low likelihood for success to achieve our goals.

Device blocks are an alluring tactic as they, in theory, can be decoupled from a user's IP address and target unchangeable data about a user's device. In reality, while a large swath of data is capturable via fingerprinting these configurations are as easily changed as IP addresses. In addition to device blocks being ineffective due to their ease of evasion, they face challenges to implement: high technical costs and a vocal privacy advocate community of our Wikimedia volunteers and colleagues.

Furthermore, time is not on our side. Technology advancements are working against us, making it easier for consumers to clear cookies or alter their fingerprints. Long-term abusers will always be motivated to evade blocks, and device blocks are a low hurdle.

Pursue alternate mitigations

The real cost of device blocks would be the opportunity cost of Wikimedia software developers working on more feasible features that would have a biggest impact on achieving the goals of abating long-term abuse, in addition to other Community Health objectives.

- User reporting system
    - Harassment will occur, and the users who receive abusive messages should have a respectful avenue to seek resolution[8]
- Improved sockpuppet hunting tools (CheckUser)
    - Productivity improvements to existing functionality. Many are already documented on Phab tag #checkuser
    - CheckUser could proactively suggest users who may be sockpuppets, rather than just reactively show raw data based on a user query.

[8] https://meta.wikimedia.org/wiki/Community_health_initiative/User_reporting_system_consultation_2019

    - CheckUser could obfuscate the data, meaning more privacy conscious users (e.g. German Wikipedia) would use it more.
- Improved username blacklist
    - Some harassers will create usernames that are designed to inflict damage in themselves[9]
    - Improvements to Extension:TitleBlacklist for more sophisticated prevention of creating undesirable usernames.
- Improve the functionality of existing cookie blocks
    - Cookie blocks do not currently work on the VisualEditor — T196575
    - Cookie blocks expire after 24 hours because they can propagate to other users via IP autoblocks. We could extend this value slightly longer, acknowledging the risk of potential collateral damage.
- Tools that give individual users better control over who can communicate with them
    - Email and notification blacklists exist, but could be expanded to support user groups (e.g. newer users.)
    - Mute lists should be global (i.e. cross-wiki)
    - Other enhancements include: T164542
- Redesign the blocked-user experience to discourage block evasions
    - Currently, when a user is blocked we say “you are blocked.” This is so they can appeal it if it is made in error.
    - We could have some blocks be “shadow blocks” which do not allow the user to edit but also do not show them that they are blocked.
    - Consider how much LTA documentation is out in the open. Consider putting it behind
    - Promote the use of role accounts (e.g. the blocks and messages come from User:WikipediaAdministrators instead of User:Bob123)

[9] https://meta.wikimedia.org/wiki/Community_health_initiative/Mechanisms_to_prohibit_blatant_attack_usernames

Perform in-depth research

This report is the syntheses of discussions, competitive analysis, reading industry publications and technical investigations that occured over the first two months of 2019. We feel confident in our broad, initial recommendation to avoid device blocks but in no way do we believe this report is comprehensive.

If mitigating long-term abuse is a priority of the Wikimedia Board of Trustees and/or the Wikimedia Foundation, we recommend additional research but performed in this realm before any large-scale development projects are begun.

Potential research questions include: How broad is the problem of long-term abuse? How technically competent are these abusers? What are the profiles of the most nefarious long-term abusers? How do they evade blocks? What AbuseFilter solutions are effective at combating long-term abuse?

Better define success metrics

We know that long-term abuse is problematic but we do not have hard data on the actual extent of its disruption and harm. While this document above suggests some possible metrics to measure success they were drafted in a vacuum without baselines or further research. We recommend that data measurements be put into place to gain insight into the productivity and stress loads that are caused by abuse, both long-term and short-term. This data will better illuminate project prioritization.