Bots That Pass as Current Browsers Evade a User-Agent Detection Rule: A 30-Day Examination of a Disability Community Site and Implications for Accessibility

Yoonjae Jeon
Ravenwood High School
Brentwood, TN, USA
jeonyoonjae465@gmail.com

Abstract

Bots crawl the internet without regard to how large a website is. In 2025, automated traffic made up more than 53% of all web traffic, more than the share that came from people. This study analyzed the 32,980 requests that reached a disability community site that I run, over 30 days. Ten targets that sent many requests were grouped by the paths they went after, and the Shy Guys User-Agent rule was applied to each. Six of the ten went after the site's internal files and four went after login-related paths. The rule caught five of the nine individual-IP targets; the separate multi-IP /wp-login.php group was not caught. Most of the targets it missed were passing as a current browser. This points to a weakness in detection rules that read only the surface information a bot writes about itself. One way to catch bots is to plant a hidden link as a trap. But a screen reader will announce a link that is hidden visually yet left in the page's accessibility tree, so on a disability community site this existing method needs to be checked for accessibility first; display:none is one way to hide the link from it.

Keywords: bot detection, web scanning, web security, screen reader, web accessibility

1. Introduction

Bots do not discriminate between large and small sites. In 2025, over 53% of web traffic was generated by automation, exceeding that of human traffic at 47% (Imperva, 2026). ABLE TIP (abletip.com), a disability advice exchange site that I built and manage, attracted around 33,000 requests within 30 days. Many of these were for resources that do not exist on the site, including files and pages such as /.env, /wp-login.php, and /backup2.zip.

In order for a disability community site to be accessible to visually impaired people, consideration must also be given to how assistive technologies such as screen readers interact with bot prevention techniques. How can 30 days of requests to a disability community site be categorized according to what the requesting bots are seeking? And of these, which categories of bot does the Shy Guys User-Agent rule detect?

2. Related Work

Many internet bots will scan all sites. Over a 16-month period, Durumeric et al. (2014) monitored unused internet address space and reported that in January 2014 alone around 1.76 million hosts undertook 10.8 million scans. Canali and Balzarotti (2013) ran 500 honeypot sites, and the first suspicious activity appeared 2 hours and 10 minutes after the sites went live. Even small

sites are therefore at genuine risk of unseen automated attack, and web security matters at this scale.

Other research has also directly studied small sites. Kopriva (2020), writing for the SANS Internet Storm Center, collected 12 months of log data from a site that runs no CMS. Of the 1,680 requests to the 10 most requested URLs, 1,140 (67.9%) were to /wp-login.php. That URL relates to the login page of the WordPress CMS, yet the site in question does not run WordPress. Kaplan (2026) analyzed around 3 years of 404 responses from a small personal site. Approximately 34,000 requests were for files such as /.env and around 15,000 for /wp-login.php. Neither of these studies, however, investigated which features of a bot cause it to be caught or missed by a detection rule.

There has also been research into bot detection. Shy Guys (Van Boxem et al., 2026) detects bots purely by analyzing server logs. It checks whether a client has requested the site's favicon, and whether the client's User-Agent appears normal. This caught 67.6% of bot traffic and wrongly labeled 3.0% of humans as bots. McKenna (2016) concealed links so that they were not visible to humans, and treated every client that followed one as a bot. Of the 21 bot campaigns that fell into this trap within the unrestricted folder, 11 were using a spoofed User-Agent. McKenna's trap requires links to be planted in advance, and the favicon check requires records for individual IP addresses, so neither could be applied to my free plan dashboard data. Only the Shy Guys User-Agent rule was therefore tested in this study.

In summary, previous work has shown that small sites are heavily targeted by bots and that bots can be detected from server logs. Yet none of these studies analyzed a small disability community site's real data in order to find out which categories of bot a detection rule catches and which it misses. This study therefore categorized 30 days of incoming bot requests according to the URLs they sought, and tested which categories the Shy Guys User-Agent rule caught.

3. Method

3-1. Data

Requests to my site, ABLE TIP (abletip.com), were analyzed for the 30 days from October 21 to November 20, 2026. Because that 30-day window begins and ends partway through a day, it spans 31 calendar dates, and the spreadsheet therefore holds 31 daily records. Data was obtained from the Cloudflare Analytics dashboard on the free plan (Figure 1). The free plan keeps records for only 30 days, so dashboard values were copied into a spreadsheet to retain them. Dashboard figures are rounded to the nearest 10. As my site has not been actively promoted, it has very few real users.

Where a client requests a URL that does not exist, my site responds with the home page. A 200 status code therefore does not indicate that data has been exposed. When /.env and /.git/config were opened directly, they also returned the home page and not any confidential information.



Figure 1. The top User-Agent panel of the dashboard (retrieved November 25, 2026; October 26, 18:30 to November 20, 23:59 PST). Counts here therefore cover a shorter window than Table 3, which covers all 30 days. This is the form of the data used for the grouping and the rule test. The panel shows only the top nine, and request counts are rounded to the nearest ten. Because of the free plan's retention limit, the window that can still be queried no longer covers the first five days of the study period.

3-2. Targets

Dashboard data is largely aggregated by URL, by User-Agent, and by country, so it is rarely possible to associate a specific IP with both a particular URL and a particular User-Agent. Consequently, 10 targets were analyzed. These comprised 9 IPs that either requested a high volume of resources or caused a spike in requests, and one group of /wp-login.php requests from IPs in several countries.

3-3. Grouping Rules

Targets were categorized according to the URLs they sought, using the following 4 rules:

1. If a URL includes one of { config, .env, .zip, .tgz, sql, backup } → the IP is seeking internal files.
2. If a URL includes one of { login, signin, signup, register } → the IP is seeking login-related paths.
3. If both rule 1 and rule 2 are met → overlap.
4. If neither rule 1 nor rule 2 is met → undetermined.

OWASP Automated Threat (OAT) numbers (OWASP, n.d.) were then allocated to each category. Internal files corresponds to OAT-014 (Vulnerability Scanning), and login-related corresponds to both OAT-007 (Credential Cracking) and OAT-019 (Account Creation).

3-4. How the UA Rule Was Applied

The Shy Guys User-Agent rule (Van Boxem et al., 2026) was applied to the categorized targets. Algorithm 1 of the original paper marks a client as a bot if its User-Agent includes a bot-related

word such as bot, crawler, or spider; if it matches an entry in a list of known bot User-Agents; if it does not start with Mozilla/5.0; or if it names a browser or operating system version that is no longer supported, meaning more than 2 years past the date its support ended. The known-bot list check was not applied in this study, so three of the four checks in Algorithm 1 were used. As the original paper does not specify the date from which the 2-year period should be calculated, the last day of the analyzed data, November 20, 2026, was used. Furthermore, as the dashboard does not show whether a particular IP requested the site's favicon, that part of the original method was not applied.

The Shy Guys rule judges individual User-Agents, but this study categorized data at the IP level. Therefore, where an IP used several User-Agents, the target was counted as caught if the requests arriving under User-Agents that matched the rule made up at least 50% of that IP's total requests. Requests arriving under User-Agents not listed on the dashboard were counted as not caught, which makes the result conservative. Request data for individual User-Agents was re-queried from the dashboard on November 25, 2026. Once more, because of the 30-day retention limit, only data from October 26 to November 20 could be obtained. For the three IPs decided by the majority rule, the totals returned for that window matched their 30-day totals in the spreadsheet (1,740, 980, and 750 requests), so the numerator and the denominator come from the same window.

3-5. Limitations

1. Only one site and 10 of its targets were analyzed. These 10 are not necessarily the 10 most highly requested targets. Consequently, the findings cannot easily be generalized to other sites.
2. The dashboard only provides aggregated data, and request counts for individual User-Agents could only be obtained from October 26 onwards. The figures may therefore not be wholly precise.
3. There is no ground truth that specifies whether or not a target is in fact a bot. The judgment has therefore been based upon the URLs requested. Should that judgment prove incorrect, so too would the "missed" result.
4. As noted above, the favicon check in the original paper could not be applied, and within the User-Agent rule the known-bot list check was not applied either. What was tested is therefore three of the four checks in Algorithm 1, not the whole Shy Guys method. A User-Agent on such a list that tripped none of the three checks used here would have been counted as not flagged.
5. This study selected November 20 as the point from which to calculate the 2-year period, as the original paper does not specify one. Had another date been selected, some verdicts would have differed.
6. The majority rule at the IP level and the treatment of User-Agents not listed on the dashboard are not part of the original paper. Were a different criterion adopted, for instance counting a target as caught if any one of its User-Agents matched, the results would differ. Indeed, target 7 was judged to be missed for this reason.
7. OAT-007 and OAT-019 were inferred, as the contents of the requests could not be seen. Consequently, the true nature of an attack may differ from the allocated OAT number.

8. Table 3 only covers the top 9 User-Agents displayed on the dashboard, which represent 57.3% of the 30 days of requests. The rule could not be applied to the remaining 42.7%.
9. The discussion of screen readers and concealed links is a design implication drawn from prior work, not a finding of this study. No screen reader user was observed and no concealed-link trap was deployed on the site, so whether such a trap would in fact misclassify screen reader users remains untested.

3-6. Ethics

Only incoming requests were observed and categorized. No IP used by a bot was connected to or scanned. Within this paper, only the leading digits of an IP are shown, for example 93.123.x.x. The data comes from a site I run myself, and no information that could identify an ordinary visitor is recorded here.

4. Results

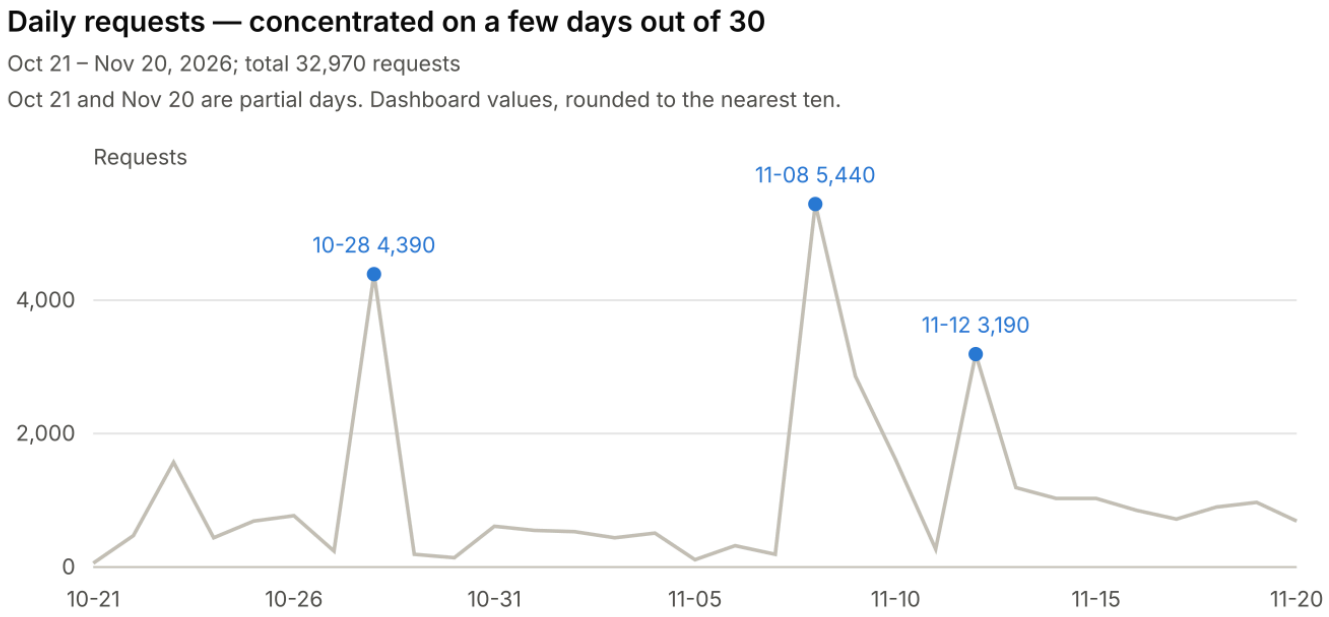

Figure 2. Daily requests, October 21 to November 20, 2026, from the 31 daily records kept in the spreadsheet. Each daily value is rounded to the nearest ten, so the daily values sum to 32,970 against the dashboard's 30-day total of 32,980.

Over the 30 days, requests were not received at a steady rate (Figure 2). A few hundred requests per day was usual. However, there were days when far more arrived, of particular note 5,440 on November 8, 4,390 on October 28, and 3,190 on November 12.

Following application of the categorization rules, 6 targets were categorized as seeking internal files and 4 as seeking login-related paths. By number of requests, the former accounted for 10,120 and the latter 3,880. No target fell into both categories or into neither (Table 1).

Table 1. Grouping results

Group	Targets	Requests	OAT
Internal files	6	10,120	OAT-014
Login-related	4	3,880	OAT-007, OAT-019
Overlap	0	0	—
Undetermined	0	0	—

Applying the Shy Guys User-Agent rule caught 3 of the 6 internal-files targets and 2 of the 4 login-related targets. Every internal-files target that was caught was caught by rule ③ (does not begin with Mozilla/5.0), and every login-related target that was caught was caught by rule ④ (a browser or operating system version whose support ended more than 2 years ago). Five targets were missed (Table 2). Because target 10 is a group of requests from IPs in several countries rather than a single IP, the ten targets are not a uniform unit: five of the nine individual-IP targets were caught, and the multi-IP group was not.

Table 2. User-Agent rule verdicts

Group	No.	IP (partial)	User-Agent	Rule matched	Verdict
Internal files	1	93.123.x.x	TLM-Audit-Scanner/2.0	③	Caught
Internal files	2	195.178.x.x	(none)	③	Caught
Internal files	4	185.177.x.x	curl/8.7.1	③	Caught

Group	No.	IP (partial)	User-Agent	Rule matched	Verdict
Internal files	6	45.138.x.x	Win Chrome/149	—	Missed
Internal files	8	34.148.x.x	Mac Chrome/131	—	Missed
Internal files	9	162.62.x.x	Mac Chrome/131	—	Missed
Login-related	3	195.178.x.x	8 or more UAs	④ (1,490/1,740)	Caught
Login-related	5	195.178.x.x	9 or more UAs	④ (720/980)	Caught
Login-related	7	93.123.x.x	9 or more UAs	④ (320/750)	Missed
Login-related	10	/wp-login.php group	Win Chrome/155 and others	— (0/410)	Missed

The User-Agent rule was also applied at the request level. Of the 32,980 total requests within the 30 days, the 9 highest-ranking User-Agents shown on the dashboard account for 18,890 requests (57.3%). Applying the same rule to those 9 flags 12,240 requests (64.8%) (Table 3). Because these nine User-Agents were not put through the path-based grouping, Table 3 reports only whether the rule flagged a request, not whether the request came from a bot.

Table 3. The rule applied to the nine top User-Agents (30 days)

User-Agent	Requests	Rule matched	Verdict
TLM-Audit-Scanner/2.0	4,630	③	Flagged
(none)	4,030	③	Flagged
Win Chrome/131	2,530	—	Not flagged

User-Agent	Requests	Rule matched	Verdict
curl/8.7.1	1,820	③	Flagged
Win Chrome/149	1,540	—	Not flagged
Mac Chrome/131	1,380	—	Not flagged
Linux Chrome/131	1,200	—	Not flagged
iPhone iOS 13.2.3 Safari 13.0.3	980	④	Flagged
Mac Chrome/121	780	④	Flagged
Total	**18,890**		**Flagged 12,240 (64.8%)**

All 6,650 requests that did not match came from User-Agents naming Chrome 131 or 149. Separately, during the 30 days the home page ("/") was requested 3,850 times while the favicon ("/favicon.ico") was requested 240 times, and the largest share of requests by country came from the Netherlands, at 14,980 (45%).

5. Discussion

The bots in this data fall into 3 types. Those that make no attempt to pass as a browser are the easiest to catch. Those that try to pass as an outdated browser are caught, because the Shy Guys rule checks for versions that are no longer supported. However, bots that pass as a current browser — meaning a browser version recent enough not to trip rule ④ — are not caught by the User-Agent rule. Target 7 is an unusual case, as it mixed old and current browsers (Figure 3).

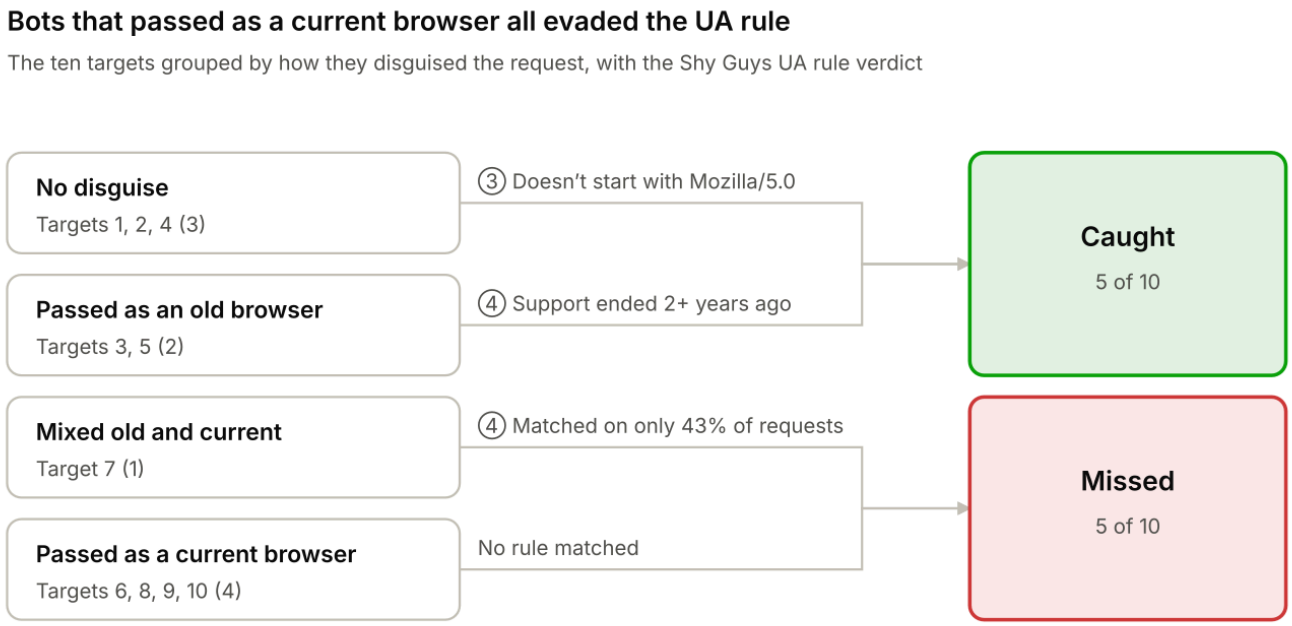

Figure 3. The ten targets — four ways of disguising a request and the resulting verdicts.While Shy Guys reports catching 67.6% of bot traffic, it does not address which types of bot it misses. Here, both the bots that made no attempt to pass as a browser and those that passed as outdated browsers were caught. The bots that passed as current browsers were not.

Furthermore, the 64.8% obtained at the request level cannot be compared directly with the 67.6% detection rate. The latter is the proportion of requests already known to be bots that the method caught, and this study has no ground truth for determining which requests come from bots. Nevertheless, my site has not been actively promoted and so has very few real users; during the 30 days the home page was requested 3,850 times while the favicon was requested only 240 times; and 45% of requests came from the Netherlands. A browser stores the favicon once it has fetched it, so the favicon count does not stand for a number of people. Taken together, though, these three figures suggest that most of the traffic was in fact from bots. The proportion of bots among the 6,650 unflagged requests could not be determined from the available data. The favicon check in the original paper has to be applied at the IP level and could not be used here, but the favicon count across all requests could still be read in this way.

To strengthen the User-Agent rule, a concealed link can be used as a trap (McKenna, 2016). As sighted users can only click links they can see, they have no occasion to follow one. Bots, by contrast, can read concealed links, so any client that follows one can be categorized as a bot. For screen reader users, however, this may not hold. A link that is hidden visually but left in the page's accessibility tree will be announced by a screen reader, so a screen reader user could follow the trap and be classified as a bot. Concealed-link traps therefore need to be evaluated for accessibility before being used on a disability community site. Using the CSS property display:none is one option, since it removes the link from the accessibility tree, but its effectiveness as a trap was not tested in this study. Alternatively, a further category could be

created for IPs with a screen reader enabled, but this approach would present technical and privacy problems and so further research is required.

6. Conclusion

This study categorized 30 days of bot traffic to a disability community site into 2 groups: bots seeking internal files and bots seeking login-related paths. The Shy Guys User-Agent rule caught the bots that made no attempt to pass as a browser and the bots that passed as outdated browsers. However, it failed to catch the bots that passed as current browsers.

This suggests that there are shortcomings in bot detection rules that analyze only the surface information a bot writes about itself, such as its User-Agent.

Further research is also required into bot detection that considers behavior, including favicon requests and concealed-link traps, and that is flexible enough to adapt to circumstances rather than reading the User-Agent alone.

AI Use Disclosure

Throughout this paper I used the AI tool Claude (Anthropic). I wrote the manuscript in Korean and had Claude translate it into English. I then rewrote that translation in my own words, and Claude edited my rewrite for accuracy and American usage. Claude also advised on sentence structures and drafted parts of the Introduction, Related Work, Method, Results, Discussion, and Conclusion, which I subsequently completed, edited, and approved. It queried request counts for individual IP addresses from the Cloudflare dashboard, checked figures cited from earlier work against the original sources (which prompted me to amend 2 figures in my text), and confirmed browser and operating system version dates against official release notes. Figures 2 and 3 were drawn by Claude from numbers and verdicts I supplied, and it cropped the dashboard screenshot used as Figure 1. It also produced a corrected copy of my spreadsheet recording those checks. Claude proposed the structure of the abstract and the form of the title, and wrote this disclosure statement.

I set the research question, wrote the categorization rules, judged all 10 targets against the User-Agent rule, chose the title and the content of the abstract, and wrote the interpretation, the discussion, and the whole of the Korean manuscript from which this paper is translated. During the categorization and the judging Claude only asked questions; every classification decision was mine. Claude is not an author of this paper.

References

1. Canali, D., & Balzarotti, D. (2013). Behind the scenes of online attacks: An analysis of exploitation behaviors on the web. NDSS 2013. https://www.ndss-symposium.org/wp-content/uploads/2017/09/12_4_0.pdf

2. Durumeric, Z., Bailey, M., & Halderman, J. A. (2014). An internet-wide view of internet-wide scanning. USENIX Security '14. https://faculty.cc.gatech.edu/~mbailey/publications/usesec14_scanning.pdf
3. Imperva. (2026). Bad bot report 2026: Bots in the agentic age. https://www.imperva.com/blog/bad-bot-report-2026-bots-agentic-age/
4. Kaplan, M. S. (2026). The attack behind the 404s. https://mertskaplan.com/en/blog/the-attack-behind-the-404s
5. Kopriva, J. (2020). What pages do bad bots look for? SANS Internet Storm Center. https://isc.sans.edu/diary/What+pages+do+bad+bots+look+for/26414
6. McKenna, S. F. (2016). Detection and classification of web robots with honeypots [Master's thesis, Naval Postgraduate School]. https://faculty.nps.edu/ncrowe/oldstudents/28Mar_McKenna_Sean_thesis.htm
7. OWASP. (n.d.). Automated threats to web applications (OAT-001–021). https://owasp.org/www-project-automated-threats-to-web-applications/
8. Van Boxem, R., Barbette, T., Pelsser, C., & Sadre, R. (2026). Shy Guys: A light-weight approach to detecting robots on websites. arXiv:2603.28546. https://arxiv.org/abs/2603.28546