Michael Saia – Cloudflare Workers for Gaming

Cloudflare is well-positioned to add immense value to gaming development studios across the globe. Between a CDN with best in-class latency for UDP (User Datagram Protocol) and DDoS protection with specific strength against attacks to layers 3, 4, and 7, Cloudflare already addresses the two key concerns facing gaming studios, speed and security. The only problem: existing competition with similar solutions already in place for gaming[1]. Apt additions to existing Wrangler solutions in Cloudflare Workers for Gaming can differentiate Cloudflare in the space, making it developers' CDN, DDoS security, and serverless computing platform of choice.

Analysis of 8 of Cloudflare's closest competitors reveals that, only two, Amazon Cloudfront and Akamai, have developed nuanced, gaming specific solutions[2]. As it pertains to Cloudflare Workers for Gaming, Amazon's subsidiary GameSparks appears to be the only solution offered by immediate competitors to directly address problems facing video game developers. Their solutions center around templates for game development, similar to current industry-specific template offerings by Cloudflare Workers, like multiplayer match-making, real-time leaderboards, and virtual/meta currency templates. It's my belief that as Cloudflare develops its Workers solution for gaming these templates will be reproduced with ease as user adoption expands. However, to gain new work and differentiate our product I propose the use of *Run-Time Verification* to improve in-game security, the game development process, and user experience.

Run-time verification is "the process of observing a sequence of events generated by a running system and comparing it to some formal specification for potential violations." Researchers at the University of Quebec have tested its use automating the process of finding "bugs" during game development and have achieved favorable results with games across genres and scales[3]. Bug tweaking is a major pain point for game developers for two key reasons. First, bugs pose serious security threats and enable "bad actors" opportunities that may be denied elsewhere (like DDoS). Second, developers are under a tremendous amount of pressure to ship games as quickly as possible for their studios, often preventing sufficient analysis for bugs toward the end of the game development lifetime. Runtime verification can detect and automatically report bugs as they are experienced by players in real time.

Bugs and credential stuffing are the two predominant reasons why security within the gaming is susceptible to attack. Ease of user authentication to avoid customer attrition creates the opportunity for credential stuffing and bugs are inherent to the development all programs. Based on market research done by Limelight Networks, 53.9% of gamers would not continue to purchase from or play on an online gaming website that had been hacked or been subject to a data breach, creating a serious predicament for gaming studios everywhere[4]. Credential stuffing is typically enacted by botnets (and often-times known botnets) and Cloudflare already has an existing solution for preventing anomalous traffic from breaching security. On the other hand, the only current solution for debugging is simply that, manual developer debugging. Run-time verification for automating the debugging process within the gaming community when proposed in March 2014 was to the knowledge of the researchers novel, creating a unique differentiator that can help Cloudflare win market share in the game development ecosystem.

For Cloudflare Workers for Gaming's go-to-market strategy, I suggest deploying our product for indie-game developers to start. This exercise will propel us along our customer-discovery and market validation paths in gaming. Major indie-game publishing platform Steam saw 9,050 games published in 2018[5]. This number dwarfs the number of games produced by large studios by nearly an order of

magnitude. Getting our product in front of as many developers as possible to start and then collecting feedback will offer us key insights into developer problems as quickly as can be hoped. Prior to this first launch I suggest including templates for publishing to key indie platforms like Steam, Itch.io, and the iOS App Store and making our initial offering free to accelerate user adoption. This will allow us a playground to test exploratory concepts, *like run-time verification for bug-fixing*, and collect feedback on a large sample size of developers and improve our product prior to approaching larger studios.

To effectively measure success, Cloudflare Workers for Gaming needs to rely on the customer's experience. User adoption goals, market share and formal feedback scores are all fine metrics to measure the traction we have with developers, however the latter and an implementation of a sophisticated feedback system for truly understanding the problems faced by our end user is paramount when entering a new industry. Because of the qualitative nature of feedback, implementing a scoring system to assess the capabilities of all features of the product will pay dividends when versioning and perfecting our product by allowing us to set quantitative goals. Setting a benchmark score for our product based on this feedback system and incentivizing users to provide us feedback for temporary cost-free use of additional Cloudflare products creates an experimental set-up that allows our team to tailor our product to pain points that are only felt by those on the front lines of video game development.

Despite the favorable positioning of Cloudflare Workers for entry into this space, direct competition, specifically with Amazon Cloudfront, does pose several threats. Amazon's ecosystem can make switching costs for current AWS customers difficult to overcome. Amazon's products including Lumberyard (a free gaming engine), Amazon GameLift (dedicated game server hosting) and GameSparks, already have major studio clients like Bethesda, Ubisoft, and Gearbox Software. In addition, introducing concepts like run-time verification require sufficient customer education and awareness so that developers can fully understand what they stand to gain by making the choice to choose Cloudflare. If Cloudflare is to succeed in this space, we will need to make our product visible, easy to understand via documentation, and cost-effective to offset potential switching costs.

In summary, for Cloudflare's family of products (CDN, DDoS Protection, and Workers) to succeed in the gaming industry we will need differentiation, developer feedback, and cost effectiveness. We can differentiate our product by solving problems specific to the industry like credential stuffing (anomalous traffic prevention) and debugging (run-time verification). We can iteratively improve our product by first launching to a large base of indie developers and using incentivized feedback systems to create needed solutions that are not apparent at the surface. We can avoid competitive risks by investing in user education that will allow developers to see our value proposition while offering a low enough price-point to justify experimentally adopting our new gaming solution. If Cloudflare Workers for Gaming is able to capitalize in these key areas, industry traction should follow.

Citations:

[1] See “Why Amazon Cloudfront?” at https://aws.amazon.com/gametech/cdn/

[2] The competitors list I used is from “Datanyze” at datanyze.com

[3] See “Automated Bug Finding in Video Games: A Case Study for Runtime Monitoring” by Varvaressos, Lavoie, Massé, Gaboury and Hallé

[4] See https://www.limelight.com/resources/white-paper/state-of-online-gaming-2019/#security

[5] See https://www.statista.com/statistics/552623/number-games-released-steam/