An Arms Race On Broadway:

Bot-Based Ticket Scalping Hounds Blockbuster Musicals

Ben Nissan

December 13, 2017

Abstract

The recent surge in ultra-popular Broadway musicals such as *Hamilton*, *Dear Evan Hansen*, and *The Book of Mormon* has resulted in an equal surge in ticket scalpers, who buy tickets en masse in order to resell them at drastically inflated prices. In order to bypass theaters' customer authentication systems, these scalpers have developed sophisticated bots that spoof large numbers of unique customer identities. Theatre companies, in turn, have repeatedly responded with equally sophisticated authentication systems. The result is an ongoing security arms race between ticket sellers and ticket scalpers. This article will explore the causes, tools, and security ramifications of this conflict, as well as its relevance to the larger context of the ticket economy.

A major recent trend in the theatre industry is the rise of the ultra-popular Broadway musical, and with it, the ultra-rare Broadway ticket. Musicals such as *Hamilton*, *Dear Evan Hansen*, and *The Book of Mormon* are more than mere hits: they have become cultural mainstays, generating universal rave reviews and regularly selling out months in advance. An unfortunate side effect of this sheer popularity, however, is that such shows have become high-profile targets for ticket scalpers: resellers who purchase tickets en masse, only to flip them for drastically inflated prices.

With online ticket sales largely supplanting in-person and telephone sales in recent years, scalpers have developed new techniques to maximize tickets scalped, and correspondingly profits. In particular, scalping bots—programs capable of making rapid-fire purchases and circumventing ticket sale authentication methods—have become the center of new conflicts between ticket sellers and ticket scalpers. Such bots are capable of automating massive amounts of ticket purchases in a manner of seconds, substantially reducing the stock available to everyday theatergoers—and, in turn, angering theater owners trying to eliminate illegitimate sales. Ticket sellers have responded with ever more complex authentication methods, which scalpers, in turn, have attempted to circumvent. The result is an ongoing security arms race between ticket sellers and ticket scalpers, as both parties attempt to take control of the ticket economy—and, by extension, profits.

The particular phenomenon of widespread bot-based ticket scalping owes its genesis to a number of historical factors. Primarily, it draws from the simultaneous development of robust web-based ticketing systems and a surge in the number and frequency of hit Broadway shows in the 21st century. Ticketmaster, now the largest online ticket retailer, began selling tickets online in 1996,[1] while the Schubert Organization, one of New York's largest theatre-owning groups, opened its ticketing division, Telecharge, in the early 1980s,[2] quickly moving into the online sphere. The two companies have rapidly grown to dominate the online ticket

[1] "Our History", *Ticketmaster*, https://www.ticketmaster.com/about/our-history.html.
[2] "History", *The Schubert Organization*, http://www.shubert.nyc/about-us/history/.

sphere, with Telecharge alone selling two-thirds of all Broadway tickets.[3]

The growth of robust online ticketing infrastructure coincided with a number of productions that thrust Broadway into the national limelight. *Rent*, as the first musical to introduce then-controversial, highly discounted same-day rush tickets, garnered an enormous following among young audiences historically shut out of Broadway theaters.[4] *In The Heights* and *Next to Normal*, pushed the boundaries of musical genre and subject matter in musical theatre, respectively, garnering widespread audiences and across-the-board critical acclaim for it; both were nominated for the Pulitzer Prize in Drama, with *Next to Normal* winning the first for a musical since *Rent* itself.[5] Of course, no discussion of blockbuster musical theatre is complete without *Hamilton*; a Pulitzer prize winner as well,[6] it has continually sold out its Broadway production for the entirety of its nearly three-year run, setting the record for the highest-weekly-grossing show in Broadway history.[7] Finally, recent hit *Dear Evan Hansen* indicates that the Broadway ticket surge shows no sign of stopping, garnering upwards of $10 million in advance sales just after its opening.[8]

Above all, one thing is clear: Broadway tickets have become a blistering-hot commodity, largely propelled by the online ticketing industry. It is unsurprising, then, that they have become a major target for industrial-scale ticket scalpers. In a recent suit, Ticketmaster alleged that upwards of 30,000 *Hamilton* tickets—nearly 40% of the total stock—were fraudulently purchased for resale by Prestige Entertainment, one of the country's largest ticket scalping enterprises. Ticketmaster alleged that a number of malicious techniques were used to obtain the tickets, including the use of duplicate email accounts, IP address spoofing, and automated CAPTCHA solving. Furthermore, they leveraged large-scale distributed systems

[3] "SeatGeek and Telecharge Announce Ticketing Integration", *SeatGeek*, November 11, 2014, https://seatgeek.com/press/seatgeek-and-telecharge-announce-ticketing-integration.

[4] "How *Rent* Revolutionized Modern Musical Theatre", *The Learned Fangirl*, June 11, 2016, http://thelearnedfangirl.com/2016/06/how-rent-revolutionized-modern-musical-theatre/.

[5] "Drama", *The Pulitzer Prizes*, http://www.pulitzer.org/prize-winners-by-category/218

[6] Ibid.

[7] Michael Paulson, "'Hamilton' Hits a New High: The Most Money Grossed in a Week on Broadway", *The New York Times*, November 28, 2016.

[8] Olivia Clement, "*Dear Evan Hansen* Advance Sales Climb to $10 Million After Opening", *Playbill*, December 9, 2016.

to both spoof unique identities and gain an edge on high-speed bandwidth for purchases.[9]

Such professional-grade, widespread, and influential ticket scalping does not merely plague *Hamilton*, however; it has become endemic across the New York entertainment industry. In a landmark 2016 report, New York State Attorney General Eric T. Schneiderman identifies ticket bots as a primary tool in the city-wide scalping epidemic predicated by the 2007 repeal of New York's anti-scalping laws.[10] Schneiderman draws a clear line between the heavily regulated ticket market of the "Anti-Scalping Era", pre-2007, and the situation since: though still nominally regulated, significantly laxer than the hard-and-fast bans of the past. He notes that after the 2007 repeal, ticket bots were not explicitly banned by law until 2010.[11] Evidently, according to Schneiderman, this was too little, too late: in the interceding years, ticket scalpers developed a near-foolproof formula for ticket bots capable of automating the entire purchase process, from detection, to selection, to automation.

The bot-based purchase cycle can be split into four major phases, each or all of which can be encapsulated within a single bot or distributed across a larger system. First, there are bots solely devoted to scanning sites for available tickets and new releases. Schneiderman refers to these as "spinner", or "drop checker", bots, and notes that they are set up as constant-monitor systems;[12] Ticketmaster itself has estimated that up to 90% of its web traffic comes from such spinner bots.[13]

Next, there are bots focused on trawling ticket sites to make the initial ticket reservations. These bots leverage the several-minute grace period allotted to real users making online reservations; as soon as a spinner bot indicates a particular ticket is on sale, reservation bots place mass amounts of simultaneous reservations, taking those tickets out of the pool available for actual customers and buying the attacker time to filter out as few, or as

[9]Gene Maddaus, "Ticketmaster Says Bot Army Bought 30,000 'Hamilton' Tickets", *Variety*, October 2, 2017.

[10]Eric T. Schneiderman, "Obstructed View: What's Blocking New Yorkers from Getting Tickets", *New York State Office of the Attorney General*, January 28th, 2016, 3.

[11]Ibid, 8.

[12]Ibid, 15.

[13]Emily B. Hager, Channon Hodge, and Tim Nackashi, "Fair Ticketing: Fans Before Scalpers", *The New York Times*, May 27, 2013.

many, tickets as they ultimately want. Of this particular phase, Schneiderman points out:

> Some brokers use this interval to offer the temporarily reserved tickets for sale on secondary market platforms such as StubHub at a given markup, and only if the resale is quickly consummated do they then actually buy the reserved tickets from the primary ticket vendor. This ability to use temporary reserve to avoid risk greatly undermines a shibboleth repeated to NYAG during our investigation, that brokers benefit the ticket industry as a whole, including artists, promoters, and venues, by taking on financial risk through up-front purchases of lots of tickets which they may be unable to resell.[14]

Once tickets are reserved, and those designated for final purchase selected by the scalper, a third set of bots automates the actual ticket purchases. This step constitutes the majority of fraud in the ticket scalping process; customer names, addresses, and credit card numbers are frequently forged or stolen, and purchase locations and IP addresses are spoofed here to prevent duplicate purchases from registering as fraudulent. Up to dozens or hundreds of fraudulent customer identities can be used over the course of a single sweep of bot-based ticket purchases, providing a prime outlet for data obtained via identity theft.

Finally, a fourth group of bots is devoted to circumventing anti-bot security systems, primarily those based on CAPTCHA. It is here that the crux of the arms race lies: "Over the years, Ticketmaster has repeatedly refurbished its CAPTCHA program, using different versions of CAPTCHA created by third parties such as Google and Solve Media."[15] Schneiderman goes on to discuss at length the role of machine-learning-based CAPTCHA solvers in keeping up with new innovations in the authentication technology:

> In many cases, they collected thousands of the new CAPTCHAs and used them to "train" their software to "read" the new CAPTCHAs through improved optical character recognition. In other instances, the Bots transmit in real-time images of the CAPTCHAs they encounter on Ticketmaster and other sites to armies of typers, human workers in foreign countries where labor is less expensive. These typers employed by companies such as Death by CAPTCHA, Image Typerz, and DeCaptcher read the CAPTCHAs in real-time and type the security phrases into a text box for the Bot to use to bypass ticket vendors defenses and use their sites.

The race, so to speak, thus initially appears stacked heavily in the ticket scalpers' favor; each new CAPTCHA innovation quickly falls victim to OCR-based solver systems

[14]Schneiderman, 16.
[15]Ibid, 17.

capable of compounding their own progress, each iteration building both on the one before it and on new, real-world CAPTCHA data. Conscious of this new reality, however, ticket sellers have begun implementing holistic authentication methods outside of CAPTCHA focused on building more complete pictures of verifiable customer identities. Three Broadway shows—*Hamilton*, *Harry Potter and the Cursed Child*, and Bruce Springsteen's one-man show *Springsteen on Broadway*—have implemented Ticketmaster's Verified Fan system, "A technology," as described by The New York Times, "that scrutinizes the purchase histories of potential ticket buyers in an effort to eliminate bots and high-volume resellers."[16] In the West End, major producers including Cameron Mackintosh are attempting paperless ticketing: instead of e-tickets, online customers solely receive an email confirmation, which must be brought to will call on the day of the show (along with the credit card used for the purchase and a valid photo ID) to claim tickets.[17]

Unfortunately, significant barriers still stand before widespread adoption of these more involved authentication methods. While *Harry Potter* and *Springsteen* have already folded Verified Fan into their overall ticket purchase process, *Hamilton* is more wary, so far only using it for a small selection of single-day presales.[18] In a world where private citizens across the board, not just theatre customers, are ever-more wary of privacy violations, it is understandable that an authentication system based on collecting large amounts of customer information for each sale is a difficult sell itself—even if Ticketmaster already has the information on hand. Furthermore, New York still legally restricts paperless ticketing to the point that it is currently unfeasible to implement at scale on this side of the pond.[19]

As a result, the arms race between ticket sellers and ticket scalpers has hit a begrudging stalemate, with a slight advantage to the scalpers, for the time being. While scalpers have machine learning methods capable of progressively improving their CAPTCHA solu-

[16]Michael Paulson, "'Hamilton' Tries New Sales Method to Battle Bots and Scalpers", *The New York Times*, August 15, 2017.

[17]Michael Paulson and Ben Sisario, "'Hamilton' and 'Harry Potter' Productions Try to Outwit Scalpers", *The New York Times*, February 12, 2017.

[18]Paulson, "'Hamilton' Tries New Sale Method".

[19]Paulson and Sisario.

tions, such approaches inevitably produce diminishing returns against ever more complex CAPTCHAs: simple (though frustrating) for real customers, but beyond the scope of enisting OCR systems. Meanwhile, although ticket sellers have run aground of legal and social roadblocks to non-CAPTCHA authentication systems, successes outside of New York make them ultimately promising, and we will likely see more such approaches used in the United States in the coming years. While such involved security systems dedicated to fighting ticket bots currently lie somewhere next to normal in the public consciousness, it appears they are currently settling into the new normal. If the seemingly nonstop trend of the Broadway blockbuster continues to escalate as it has for the past decade, they will surely be necessary.

A Source Code Analysis

Publicly-available resources exist capable of accomplishing the first step in the ticket scalping process: spinning. We will analyze, on a high level, GitHub user Alistair Rutherford's `ticket-check-bot`,[20] a basic tool for creating spinner bots.

```java
/**
 * Main worker class.
 *
 */
@Singleton
public class TicketBot
{
        private static final Logger logger = LoggerFactory.getLogger(TicketBot.class);

        private List<BotModule> modules;

        private static WebClient webClient = null;

        @Inject
        private EmailService emailService;

        @Inject
        private TemplateService templateService;

        private VelocityContext context;

        private static final String TEMPLATE_VALID_LINKS = "links";

        private static final String TEMPATE_NAME = "availability";
```

The main structure of the ticket bot is outlined here, in file `TicketBot.java`. Each bot is composed of a series of bot modules, each corresponding to a spinner for a single website, and it is in these modules that the primary functionality of the bot is contained. The remainder of class `TicketBot` is used to organize and transfer obtained data via email, so we will devote our focus to analyzing the bot modules themselves.

[20] Alistair Rutherford, "ticket-check-bot", https://github.com/alistairrutherford/ticket-check-bot.

```java
/**
 * A bot module will return a list of valid links for the sites which match the
 * implementing module requirement as 'valid' i.e tickets available.
 *
 */
public interface BotModule
{
        /**
         * Run module.
         *
         */
        public void run();

        /**
         * Return list of valid links.
         *
         * @return links.
         */
        public List<String> getLinkStatus();
}
```

The interface for a `BotModule` is fairly simple: the only required methods are a `run()` method called by the enclosing `TicketBot`, and a `getLinkStatus()` method used to obtain the list of valid ticket URLs for the given purchase site. Once these URLs are obtained, they can be passed onto a reservation bot to execute the reservations, then a purchase bot to make the purchases, along with an anti-security bot to evade authentication barriers to the sale. That said, the `BotModule` interface itself is not particularly interesting, so we will go on to examine a specific implementation Rutherford includes for British booking website Gigs and Tours.

```java
/**
 * Module to scan "gigs and tours" web-site for available tickets.
 *
 */
public class ModuleGigsAndTours implements BotModule
{
        private static final Logger LOG = LoggerFactory.getLogger(ModuleGigsAndTours.class);

        private static final String URL_LIST = "http://www.gigsandtours.com/tour/prince";
        private static final String URL_LINK = "/event/prince";

        // private static final String URL_LIST =
        // "http://www.gigsandtours.com/tour/kate-bush";
        // private static final String URL_LINK = "/event/kate-bush";

        // private static final String URL_LIST =
        // "http://www.gigsandtours.com/tour/little-mix";
        // private static final String URL_LINK = "/event/little-mix";

        private static final String URL_HTTP = "http://";
        private static final String ID_BUTTON_BUY_TICKETS = "buyTickets";

        private String baseURL;
        private WebClient webClient;
        private List<String> links;
        private List<String> linkStatus;
```

Instance variables are added to the implementation to store a target subcategory of performances the user wants to search under, as well as metadata specific to the website that can be easily found by manually examining the purchase page. A web client is declared, as well as lists to store obtained links and their corresponding statuses.

```java
/**
 * Process target site.
 *
 */
@Override
public void run()
{
        try
        {
                // Switch off java-script.
                webClient.getOptions().setJavaScriptEnabled(false);

                HtmlPage page = webClient.getPage(URL_LIST);

                LOG.info("Fetch links");

                links = fetchLinks(page);

                for (String link : links)
                {
                        LOG.info("Link: " + link);
                }

                LOG.info("Examine availability");

                linkStatus = fetchLinkStatus(links);

                for (String element : linkStatus)
                {
                        LOG.info("Tickets available at: " + element);
                }
        }
        catch (FailingHttpStatusCodeException e)
        {
                LOG.error(e.getMessage());
        }
        catch (MalformedURLException e)
        {
                LOG.error(e.getMessage());
        }
        catch (IOException e)
        {
                LOG.error(e.getMessage());
        }
}
```

Things start to get interesting in the `run()` method. Aside from basic error logging, we see that the web client is instantiated with JavaScript disabled, likely for security purposes. The main page for the desired subcategory is sent to the web client, and a `fetchlinks()` method is called on it. This returns a list of links, on which the method `fetchLinkStatus()` is called. We will now examine these two methods, which do the real heavy lifting for the

spinner.

```java
/**
 * Fetch links.
 *
 * @param page
 *            The target page.
 *
 * @return List of links.
 */
private List<String> fetchLinks(HtmlPage page)
{
        List<String> validLinks = new LinkedList<>();

        try
        {
                // final List<?> links = page.getByXPath("//a");
                final List<?> links = page.getAnchors();

                for (Object element : links)
                {
                        HtmlAnchor anchor = (HtmlAnchor) element;

                        String path = anchor.getHrefAttribute();

                        if (path.contains(URL_LINK))
                        {
                                String url = URL_HTTP + baseURL + path;

                                validLinks.add(url);
                        }
                }
        }
        catch (Throwable t)
        {
                LOG.error(t.getMessage());
        }

        return validLinks;
}
```

`fetchLinks()` manually scrapes the main web page for the desired purchase subcategory for relevant purchases. By examining each HTML anchor element on the page, and comparing its href attribute to the desired link, the method assembles the set of all desired purchase links. By appending these to the base URL, it can assemble a full list of potentially available purchases.

```java
/**
 * Find valid links.
 *
 * @param pageLinks
 *
 * @return Map of URLS with target criteria.
 */
private List<String> fetchLinkStatus(List<String> pageLinks)
{
        List<String> validLinks = new LinkedList<>();

        // For each link check status.
        for (String link : pageLinks)
        {
                // Navigate to details page.
                try
                {
                        HtmlPage page = webClient.getPage(link);

                        // Look for 'buy tickets' button.
                        Object button = page.getElementById(ID_BUTTON_BUY_TICKETS);

                        if (button != null)
                        {
                                validLinks.add(link);
                        }

                }
                catch (FailingHttpStatusCodeException e)
                {
                        LOG.error(e.getMessage());
                }
                catch (MalformedURLException e)
                {
                        LOG.error(e.getMessage());
                }
                catch (IOException e)
                {
                        LOG.error(e.getMessage());
                }
        }

        return validLinks;
}
```

`fetchLinkStatus` then examines each of the obtained links for a working button to buy tickets, again by manually scraping each link. It checks each page for an element corresponding to the desired button, and verifies that the button is not null. If that is the case, the bot knows a purchase can be made for the given performance, and adds the link to a list of valid links, which it finally returns.

As we see, the process of creating a spinner bot essentially comes down to building persistent web scraping services customized to the format of each ticketing website. For sites like Ticketmaster, which follow consistent design site-wide, this can be used to obtain a wide range of potentially scalpable purchases. Although it requires a degree of manual work on the part of the user to figure out the relevant site's format for representing available purchases, Rutherford's tool provides a deceptively powerful framework for creating the early basis of a ticket bot.

References

[1] Rutherford, Alistair. "ticket-check-bot", https://github.com/alistairrutherford/ticket-check-bot.

[2] Clement, Olivia. "*Dear Evan Hansen* Advance Sales Climb to $10 Million After Opening", *Playbill*, December 9, 2016.

[3] "Drama", *The Pulitzer Prizes*, http://www.pulitzer.org/prize-winners-by-category/218

[4] Hager, Emily B., Channon Hodge, and Tim Nackashi, "Fair Ticketing: Fans Before Scalpers", *The New York Times*, May 27, 2013.

[5] "History", *The Schubert Organization*, http://www.shubert.nyc/about-us/history/.

[6] "How *Rent* Revolutionized Modern Musical Theatre", *The Learned Fangirl*, June 11, 2016.

[7] Maddaus, Gene. "Ticketmaster Says Bot Army Bought 30,000 'Hamilton' Tickets", *Variety*, October 2, 2017.

[8] "Our History", *Ticketmaster*, https://www.ticketmaster.com/about/our-history.html.

[9] Paulson, Michael. "'Hamilton' Hits a New High: The Most Money Grossed in a Week on Broadway", *The New York Times*, November 28, 2016.

[10] Paulson, Michael. "'Hamilton' Tries New Sales Method to Battle Bots and Scalpers", *The New York Times*, August 15, 2017.

[11] Paulson, Michael, and Ben Sisario, "'Hamilton' and 'Harry Potter' Productions Try to Outwit Scalpers", *The New York Times*, February 12, 2017.

[12] Schneiderman, Eric T. "Obstructed View: What's Blocking New Yorkers from Getting Tickets", *New York State Office of the Attorney General*, January 28th, 2016.

[13] "SeatGeek and Telecharge Announce Ticketing Integration", *SeatGeek*, November 11, 2014.