When Features Gets Exploited: Functional Abuse and the Future of Industrial Fraud Prevention

Elisa Chiapponi*, Umberto Fontana*†, Elyssa Boulila*‡,Claudio Costanza*, Vincent Rigal*, Olivier Thonnard*
*Amadeus IT Group, France †Télécom SudParis, France ‡EURECOM, France
{elisa.chiapponi, umberto.fontana, elyssa.boulilla, claudio.costanza, vincent.rigal, olivier.thonnard}@amadeus.com

***Abstract*—Functional abuse is an escalating cyber threat where attackers exploit legitimate website features for fraudulent activities and resource depletion. Unlike traditional attacks, these techniques circumvent security measures by misusing intended functionalities. This paper examines two advanced forms: SMS Pumping, which abuses SMS-based services to generate excessive messages for financial gain, and Denial of Inventory (DoI), which depletes stock availability by holding items in carts without purchase. Utilizing real-world attack data, we show why traditional anti-bot defenses are ineffective against these automated attacks and provide best practices to enhance mitigation strategies. This study is the first to present the evolution of these threats from a targeted business perspective, highlighting effective ad-hoc mitigation techniques and advocating for further research into adaptive countermeasures.**

***Index Terms*—Denial of Inventory, SMS Pumping, Bot Detection, Functional Abuse, Residential Proxies, Fraud, Fingerprint Rotation**

I. Introduction

As digital ecosystems become increasingly complex, a new wave of cyberattacks is exploiting the very features designed to enhance user experience. This phenomenon, known as functional abuse, involves leveraging legitimate website functionalities and built-in features for illicit purposes, often bypassing traditional security measures to commit fraud or consume resources. A well-known and straightforward example of such an attack is web scraping, which involves the automated and systematic extraction of publicly accessible data or processed output from a web application. This data is often used for purposes like reselling, reverse-engineering website algorithms, or competitor analysis. In this case, the exploited feature is the item display functionality, which exposes product prices and availability to users who aren't logged in.

Web scraping and other kinds of functional abuse activities are recognized as automated threats by the Open Worldwide Application Security Project (OWASP) [1] and they are commonly prohibited by the Terms of Service of the affected websites. However, these activities remain legally ambiguous, as demonstrated by notable court cases lost by the targets of such abuse [2], [3], and the inherent challenge of proving that a series of requests with different IP addresses and fingerprints are connected and solely intended to exploit the system.

Although not necessarily illegal, these activities undeniably cause financial and reputational harm to the affected websites. In recent years, we witnessed an increase in robotic attacks performing functional abuse. According to the report of one of the leading companies in the anti-bot sector [4], in 2023, 17% of API attacks were conducted by bots in order to exploit business logic vulnerabilities. Furthermore, in recent years, there has been a rise in both the sophistication of bots used as attack vectors [5], [6] and in the evolution of functional abuse. Initially, functional abuse primarily targeted features accessible without authentication or payment. However, attackers now exploit functionalities that require user login or payment, expanding the attack surface to more sensitive areas of applications [7], [8]. This is particularly concerning because it implies that an application's entire attack surface includes every endpoint designed to enhance user experience. Therefore, it is crucial to anticipate potential misuse of these functionalities and implement safeguards to prevent exploitation for an attacker's gain.

In this paper, we will focus on two types of advanced functional abuse activity, SMS Pumping and Denial of Inventory (DoI). SMS Pumping is a fraudulent attack that exploits an application's SMS-based services, such as One Time Password (OTP) or notification systems, to generate large volumes of messages, often for financial gain by abusing premium-rate numbers. In a DoI attack, malicious actors repeatedly add items to online carts or reserve services without completing the purchase, depleting available stock and preventing legitimate users from making transactions.

To the best of our knowledge, while functional abuse has been discussed in academic literature, this is the first work to present real-world insights regarding SMS Pumping and DoI attacks from the perspective of targeted businesses, highlighting the practical challenges in automatically detecting and mitigating these threats. In the rest of the paper, we will examine SMS Pumping and DoI in detail, analyzing their underlying mechanics (Section II). In Section III, we will explain why traditional bot detection measures—such as fingerprinting and behavior analysis—often fall short in isolating them. Section IV will present data-driven insights from real-world attacks, demonstrating both the limitations of traditional defenses and the evolving nature of attacker tactics. In Section V, we will outline key lessons learned, provide best practices for mitigating functional abuse, and call on the research community to further investigate effective countermeasures. Finally, Section VI concludes the paper.

II. THE ATTACKS

A. Denial of Inventory

Denial of Inventory (DoI), also known as Inventory Hoarding, is an attack that exploits a website's ability to temporarily reserve goods or services before completing a purchase. Attackers repeatedly hold large quantities of items or services from a limited stock, preventing legitimate customers from making purchases or reservations.

According to OWASP [1], DoI is commonly associated with removing e-commerce items from circulation by adding large quantities to a cart or basket without completing the purchase. A variation of this attack involves making pre-reservations without completing the payment process, targeting services such as holiday bookings, hotel rooms, or flight seats. In the case of airline bookings, this tactic is specifically known as Seat Spinning [9], [10].

While DoI attacks are typically automated, with bots repeatedly sending large volume of requests to maintain control over inventory, manual execution has also been observed. On social media, some users advertise manual Seat Spinning tactics to manipulate airline seating for better selection [11]. In Section IV-B, we will present indications of this behavior. Furthermore, recently large-scale attackers seem to operate more and more with a low traffic footprint to evade detection, as we will see in one of the presented case studies (Section IV-A).

The motivations behind DoI attacks vary. Some aim to weaken and defeat competitors by depleting stock and redirecting customers to alternative sites. Others manipulate supply and demand to resell items or appointments at inflated prices. In cases involving dynamic pricing, attackers strategically hold reservations and items at lower fares without an investment to force price drops before making a legitimate purchase. Additionally, DoI can serve as a direct attack on a business, causing operational disruption. In fact, this attack can be seen as a form of Denial of Service (DoS) at the application layer (Layer 7 of the OSI model).

Regardless of intent, DoI impacts targeted businesses, increasing system load, reducing sales, causing revenue losses and reputational damage by making services appear unavailable to genuine customers.

B. SMS Pumping

SMS Pumping, also known as Artificial Inflation of Traffic Attack, is a type of automated attack where fraudsters exploit application features used for sending Short Message Service (SMS) messages. Applications use these SMSs to deliver OTPs for logins, specific documents (e.g. boarding passes) and notifications to their users. Attackers exploit these functionalities to send large volumes of text messages. Their target is impacting the application owners and/or generate revenue on their side.

To reach these goals, attackers may collude with local mobile network operators that provide lists of mobile numbers to target and share part of the corresponding revenue with the attackers [12]. Alternatively, attackers can create new local carriers and identify them as terminator actors to a primary operator. The local carrier receives compensation for all the managed traffic thanks to agreements such as those enforced by the Federal Communications Commission (FCC) [13]. Attackers can send text messages to high-cost destinations or premium numbers, for which the revenue is higher than for normal rates [14]. Depending on the scheme put in place by the carriers, the owner of the number may be unaware that their number is used in this attack, since a transit operator could stop the traffic and obtain a termination fee [12].

SMS Pumping directly affects the owners of the targeted application, primarily because they need to cover the financial costs associated with the high volume of SMS messages. In 2023, Elon Musk claimed that Twitter, former name of the social network X, was losing approximately $60 million per year due to SMS Pumping through its two-factor authentication system [15]. While no independent source has confirmed this figure, in the same year, Twitter restricted SMS-based two-factor authentication to premium accounts only, to reduce the consequences of this fraud [16]. This decision suggests the significant scale and cost of such attacks.

Moreover, SMS Pumping indirectly affects the application. It can lead to operational issues, such as delays in the application's functionality due to the large amount of SMS to handle. Additionally, if the volume of SMS exceeds the application's quotas contracted with a network operator, legitimate users may be unable to leverage this feature. This could block them in their activities that could be critical for the interaction with the application (e.g. retrieve access through OTP after losing a password). This disruption can result in a significant drop in the application's reputation.

Finally, there could also be a negative impact on network operators. SMS Pumping attacks may harm the reputation of operators, who act as intermediaries between application owners and local carriers, potentially creating a perception of unreliability.

III. STATE OF THE ART

Information about SMS Pumping and DoI is available in white papers and blogs from companies specializing in fraud detection systems [4], [8], [17]–[22]. However, these sources may offer a partial perspective due to potential commercial interests, and they typically do not include data analysis, as they are primarily intended for marketing and sales purposes.

To the best of our knowledge, no academic paper specifically details these attacks or their evolution through data insights. However, Khan et al. discuss the usage of residential proxies as vectors for DoI attacks [23], and various nuances of telephony fraud and related attacks have been studied in earlier works [12], [24]–[26]. More recently, these issues have increasingly drawn the attention of law enforcement [27].

As explained in the introduction, bots are the primary vectors for functional abuse attacks, yet academic research has not directly addressed their role in SMS Pumping and DoI. While automated attack detection has been widely studied in recent years, existing approaches face limitations in mitigating

these advanced functional abuse threats. In this section, we provide an overview of detection techniques, highlighting their constraints in addressing these evolving attack strategies, particularly when considering both behavior-based and knowledge-based approaches [28].

A. Behavior-based approaches

Advances in machine learning are now favoring behavior-based bot detection techniques that isolate bots by examining their activity records. Data is collected from user interactions with a website through web logs. The logs are subsequently grouped in user sessions. After having identified a user session, various features are extracted from log data for the task. These set of features range from general navigational pattern (e.g. session volume, count of requests per HTTP method) [29]–[31] to more domain-specific characteristics [32]–[34]. Subsequently, the data is used by a classification or clustering algorithm to differentiate sessions.

However, research has demonstrated ways to evade simple behavior-based detection methods that rely on web logs. For instance, one study adjusted page visiting time according to page content [35]. In another study, Iliou et al. statistically modeled the time between subsequent requests with a degree of randomness to generate human-like navigation [36]. Building on this, [37] introduced a bot leveraging reinforcement learning to dynamically adjust its behavior and bypass detection mechanisms.

The primary challenge in applying simple behavior-based detection to DoI and SMS Pumping attacks is that these bots do not require a high request volume within a single session to achieve their objective. This renders session-based volume metrics (e.g. total HTTP GET/POST request counts) ineffective for their identification. Additionally, other features used to detect levels of exploratory session (e.g. depth of requested URLs, number of requests on the search page, access to trap file [38]) are specifically designed to detect simpler forms of functional abuse, such as web scraping. As a result, these conventional techniques struggle to detect more sophisticated, low-volume abuse patterns like DoI and SMS Pumping attacks, highlighting a fundamental limitation in behavior-based detection for these evolving threats.

The development of a framework capable of capturing users' local behavior within an application could significantly enhance protection against these types of attacks [39], [40]. Additionally, recent research has explored biometric attributes, such as mouse movement trajectories, for web bot detection [41]–[44]. These approaches appear promising for tackling complex fraud cases like those examined in this paper. However, to our knowledge, no study has yet applied these advanced techniques to the detection of DoI and SMS Pumping bots.

B. Knowledge-based approaches

Another approach to bot detection is browser fingerprinting [45]–[48]. This technique, while traditionally used for web tracking, is increasingly employed today for detecting automated attacks on real-world websites [6], [49]. Browser fingerprinting operates by collecting characteristics related to the client's browser, as well as the device's hardware and software configuration (e.g. screen dimensions, operating system, language settings, and CPU/memory usage). Additionally, websites can analyze how the client renders web pages, executes JavaScript, and processes audiovisual elements [50]. These collected attributes are utilized to identify patterns that differentiate browsers operated by human users from those controlled by automated scripts or bots. Moreover, artifacts added by instrumentation frameworks and headless browsers, such as `navigator.webdriver`, can be extracted to detect crawlers [47].

However, attackers have developed sophisticated evasion strategies, significantly reducing the effectiveness of their detection through fingerprinting. One of the most effective techniques is fingerprint rotation, where bots dynamically alter their browser and device attributes at regular intervals to avoid detection [49]. This means that even if a bot is flagged as suspicious based on its fingerprint, it can reappear moments later with a seemingly different identity, bypassing anti-bot measures.

Furthermore, bots increasingly mimic real user fingerprints to blend in with legitimate traffic. They achieve this by carefully modifying browser properties, disguising themselves as common configurations observed in genuine user populations. Prior research has explored identifying inconsistencies in manipulated fingerprints [51], but this remains a difficult challenge as attackers continuously refine their evasion tactics and researchers develop crawlers specifically designed to circumvent advanced anti-bot solutions relying on fingerprinting [52]. Additionally, many bot operators leverage residential proxies and CAPTCHA-solving services to add more legitimacy to their fingerprints [5].

These advancements in fingerprint evasion make it clear that traditional knowledge-based detection methods alone are insufficient. As attackers continuously adapt, more advanced detection strategies—such as combining fingerprinting with behavioral analysis and anomaly detection—are needed to counteract these evolving threats.

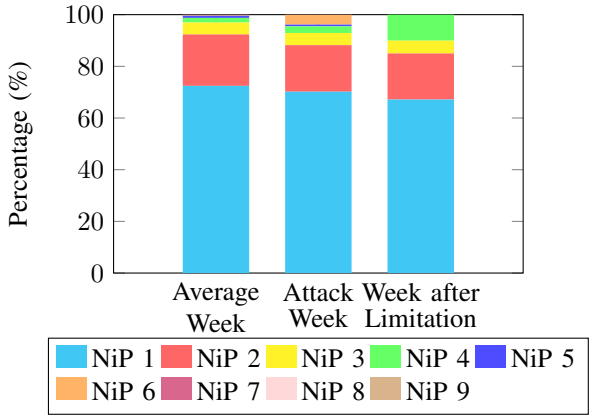
IV. Real-world cases studies

In this section, we will present key insights gathered from real-world attacks targeting flight reservation domains managed by Amadeus, a leading technology company in the airline industry.

A. From Large-Group to Low-Volume Reservations in Seat Spinning Attacks

In the airline industry, once a seat is selected on a flight, it is temporarily reserved for the passenger for a specific duration—ranging from 30 minutes to several hours depending on the domain—before payment is required. This feature is designed to enhance user experience, allowing customers to finalize their bookings without the pressure of immediate payment. However, this functionality can be exploited for

Fig. 1. Number in Party (NiP) distribution for an average week, the week in which the attack took place and there were no limitations on the maximum NiP, and the week after the limitation to less than 9 passengers per reservation was introduced.



Seat Spinning attacks, where malicious actors systematically reserve large numbers of seats without completing purchases, leading to stock depletion and service disruption.

In the early cases of Seat Spinning, we observed a significant surge in reservations involving a high number of passengers per booking. One notable instance of this occurred in May 2022, targeting a specific flight operated by *Airline A*. The fraudulent reservation requests were submitted continuously, with each new request sent as soon as the temporary hold on the previous one expired.

Fig. 1 presents a global view of reservation across all flights of *Airline A*, through the repartition of the Number in Party (NiP). This parameter indicates the number of people included in a single booking transaction, considering also the non finalized ones. The first stacked bar represents the typical NiP for seat reservations during an average week in 2022. The majority of reservations were for one or two passengers, with fewer bookings for groups of three or more. However, during the week in which the attack took place (second stacked bar), there was a sharp increase in reservations for groups of six passengers. This suggests that attackers strategically submitted fewer requests with high NiP values to maximize seat blockage while minimizing the number of required transactions. Interestingly, they did not target the highest possible NiP value for this airline, possibly to avoid triggering an immediate anomaly detection alert, as reservations for maximum-capacity groups are statistically rare. Although the attack was limited to a single flight, we can see that its impact on the overall reservation distribution was significant, particularly when considering that *Airline A* operates hundreds of flights per week.

To mitigate the attack, we implemented a temporary restriction capping the maximum number of passengers per reservation at four. Following this change, a noticeable shift occurred: there was a significant rise in four-passenger reservations, as visible in the last bar of Fig. 1. This indicates that legitimate group bookings adjust as well as the attackers' ones. Indeed, the targeted flight continued to experience a high volume of reservations at the new NiP limit, suggesting that attackers adapted their strategy and persisted with the attack despite the restriction.

To further counteract the attack, we conducted a detailed fingerprinting analysis of reservation requests from the beginning of the fraudulent requests. We introduced blocking measures based on fingerprinting patterns. Our observations revealed that attackers quickly adjusted to each new fingerprint-based rule, typically rotating their technical features (e.g. browser attributes, device information) within an average of 5.3 hours. The attack continued until two days before the flight's departure, at which point reservation holding activity ceased entirely.

This case study highlights key attacker behaviors that demonstrate both strategic planning and technical sophistication. First, the attackers conducted preliminary reconnaissance before executing the attack. They carefully studied the airline's reservation system, identifying the seat hold duration and maximum number of passengers per booking, likely to avoid triggering anomaly detection mechanisms. By understanding these parameters, they were able to devise an approach that maximized disruption while minimizing costs on their end.

Second, the attackers exhibited rapid adaptation in response to mitigation measures. When we imposed a restriction on the maximum number of passengers per booking, they promptly adjusted their strategy, shifting their reservation patterns to match the new limit while maintaining their overall goal of stock depletion. They also routinely rotated their identifiers—such as IP addresses, device fingerprints, and browser attributes—shortly after each new blocking rule was applied.

This ability to modify tactics in real time suggests the attack was highly dynamic, likely involving automated bots leveraging artificial intelligence. As a result, each new countermeasure was only effective for a limited period before attackers adapted, making long-term defense enforcement particularly challenging.

Since this use case, Seat Spinning attacks have evolved. Rather than starting with large group reservations and later adjusting to smaller groups during the attack, attackers now initiate fraudulent bookings with smaller NiP values. This tactic allows them to blend in with typical reservation patterns, delaying detection. As a result, identifying these attacks has become increasingly complex, requiring more advanced anomaly detection techniques to differentiate malicious activity from legitimate bookings.

B. Automated vs Manual Seat Spinning Attacks

As discussed in the introduction, bots are the preferred vector for functional abuse attacks, including Seat Spinning. To hold a reservation, a passenger's details—such as name, surname, and email—must be provided. In some cases, we observed entirely random entries (e.g., Name: affjgdui, Surname: ddfjrei, Email: ddfjrei@emailprovider.com[1]), while in others,

[1]The data used in the example is fictitious and provided solely for illustrative purposes.

TABLE I
TOP 10 COUNTRIES TOWARDS WHICH THE ATTACKS SENT SMS AND THE CORRESPONDING SURGE BETWEEN BEFORE AND AFTER THE ATTACK.

Country	Increase
Uzbekistan	160,209%
Iran	66,095%
Kirghizistan	37,614%
Jordan	12,251%
Nigeria	10,986%
Cambogia	4,990%
Singapore	67%
United Kingdom	44%
China	43%
Thailand	19%

attackers used repeated names across multiple reservations. Both these patterns facilitate the identification and isolation of fraudulent activity.

A particularly interesting case occurred in October 2024, where all reservations associated with a Seat Spinning attack in *Airline B* followed a structured pattern: the first passenger's name and surname remained unchanged, but the birthdate rotated systematically. Additionally, for other passengers in the same reservation, name-surname combinations overlapped, with varying birthdates. This strongly indicated automation, as well as a possible system flaw allowing repeated reservations under the same name. The clear repetition of details was instrumental in detecting and mitigating this attack.

However, not all seat-spinning attacks rely on automation. There are also manual attempts, most likely from individuals seeking to secure specific seats on an upcoming flight [11]. In December 2024, we detected an unusually high number of seat holding on *Airline C*, prompting further investigation. We found that the same fixed set of passenger names was being used repeatedly, though in different orders across bookings. Additionally, few entries contained slight misspellings of names and surnames, suggesting manual input rather than automation. Despite using a broad range of IP addresses to hide their location, the attacker's repetitive use of the same passenger details allowed us to flag and block the behavior.

Manual attacks pose a unique challenge for anti-bot detection systems. Since these attempts do not exhibit the typical behavioral patterns of automated bots, traditional bot-detection alerts are not triggered. This highlights the need for additional heuristics and behavioral analysis techniques to identify and mitigate manual functional abuse attempts effectively.

C. Advanced SMS Pumping

SMS Pumping attacks typically target OTP services, which are widely used in two-factor authentication systems and are easily accessible, since they are often required during login. However, fraudsters have begun to explore more sophisticated methods that require a deeper understanding of an application's mechanics and more complex interactions with the application to exploit it effectively.

One of these advanced attacks targeted *Airline D*'s boarding pass issuance system. Some airline websites allow users to receive their boarding passes, among other options, via SMS, after ticket issuance. In this case study, attackers purchased tickets, possibly with user interactions on the airline website, using fake data and (later discovered) stolen credit cards. They repeatedly requested the boarding pass through SMS via automated bot, leveraging residential proxies to rotate their bots' IP addresses while matching the countries associated with the mobile numbers. Additionally, they continuously altered their bots' fingerprints to bypass the anti-bot protection mechanisms in front of the application. This included spoofing browser attributes, modifying device characteristics, and mimicking human-like behaviors.

The attack occurred in December 2022 and, at the time, the application did not have rate limits on the number of boarding pass requests via SMS per booking reference. Hence, the attackers were able to issue few e-tickets and leverage them to generate a boarding pass for each of them. Subsequently, the attackers generated a high volume of SMS messages per boarding pass.

Globally, there was an increase in the number of sent boarding passes around 25%. Attackers used mobile numbers of 42 different countries, but prioritized specific regions. Table I illustrates the increase in SMS volume in the top 10 involved countries during the attack compared to the same period beforehand. We can see the huge increase in the number of sent SMS. Moreover, we can see that certain countries were disproportionately targeted. Notably, there was no significant correlation between the targeted countries and the attacked domain, suggesting that the attackers selected destinations based on the larger availability for them of mobile numbers to exploit and/or the potential for higher revenue per SMS in those regions.

The attack was detected only after the total number of boarding pass requests via SMS triggered the rate limit for the targeted path, as there were no SMS rate limits per user profile in place. The SMS option was then temporary removed and the attack ceased.

This case study demonstrates a real-world example of advanced SMS pumping, as it required knowledge of the application's functionality and an initial financial transaction to execute the attack. It underscores the fact that even application components behind a login and payment gateway remain susceptible to exploitation. Moreover, this case study highlights the increasing sophistication of attackers, who employed residential proxies, fingerprint rotation, and other evasion techniques to circumvent detection mechanisms.

V. DISCUSSION

In the previous section, we have shown the challenges posed by advanced functional abuse attacks such as SMS Pumping and DoI, highlighting the shortcomings of traditional bot detection and mitigation techniques. Unlike other automation attacks, functional abuse operates within the intended features of an application, often bypassing fingerprint-based and volume-based detection methods. This makes identifying and mitigating such threats particularly difficult. Addressing

these challenges requires a shift in approach. We outline key best practices for industry professionals and propose future research directions to improve defenses against these evolving threats:

- **Expanding the Scope of Protection:** One key realization is that the attack surface for functional abuse extends across the entire application, including post-login and post-payment functionalities that are not typically associated with bot attacks. A thorough security assessment of all application endpoints is crucial to identify potential vulnerabilities. Security teams should analyze how legitimate features could be repurposed for malicious gains and proactively design safeguards accordingly.
- **Balancing Usability and Security:** Many modern application features, such as SMS-based boarding passes or seat-holding mechanisms, are designed to improve the user experience. However, as we have shown, usability can inadvertently introduce security risks. This raises an important question: where should the line between usability and security be drawn?
  If a feature is essential, it should be protected with targeted security mechanisms. Potential mitigation strategies include:
  - **Ad-hoc rate limiting**: Placing caps on how frequently a user can request certain features, such as SMS-based services.
  - **Feature access restrictions**: Limiting high-risk functionalities (e.g. SMS reception, items holding for long periods of time) to trusted users, such as verified loyalty program members.
  - **Increased Layers of Anti-Bot Detection**: Implementing additional security checks such as CAPTCHAs, at critical points. Even if attackers can leverage CAPTCHA-solving services, these measures add cost and complexity to automated attacks.
- **Undermining the Economic Incentive for Attackers:** Since many functional abuse attacks are financially motivated, making them economically unviable is one of the strongest deterrents. A promising mitigation strategy for DoI attacks could be the deployment of honeypots, decoy environments that resemble the real website and to which attackers are redirected. This technique, previously explored for web scraping prevention [53], could ensure that attackers waste resources believing to hold items in a false environment while legitimate users remain unaffected. By keeping attackers engaged with a controlled replica, their need to rotate fingerprints or adjust tactics diminishes, reducing the overall impact of the attack.
  For SMS Pumping attacks, effective mitigation could involve collaboration with primary network operators. Many fraud schemes rely on fraudulent secondary operators. By working closely with network providers, we can enforce stricter validation measures for new secondary operators, and develop mechanisms to not compensate local carriers for SMS operations involved in functional abuse cases.
- **Advancing Behavioral-Based Detection:** As pointed out in Section III-A, advanced behavioral analysis offers a promising direction to detect functional abuse for future research and industry application. Recent work in bot detection has explored local behavioral modeling, such as graph-based navigation analysis and biometric indicators (e.g., mouse trajectory tracking). These approaches could be adapted to functional abuse detection by analyzing patterns that distinguish automated abuse from legitimate user interactions. Unlike static defenses, behavioral models can evolve dynamically, improving resilience against attackers who constantly modify their techniques. We encourage further research into applying behavioral artificial intelligence, anomaly detection, and reinforcement learning to enhance functional abuse detection. Integrating these methods into existing security frameworks could improve defenses beyond traditional anti-bot measures.

VI. Conclusion

In this paper, we explored advanced functional abuse attacks, specifically SMS Pumping and Denial of Inventory, providing insights from real-world cases and highlighting the practical limitations of traditional countermeasures against automated threats in this scenario. We discussed how defenses can evolve through a comprehensive approach that includes proactive endpoint monitoring, feature-specific protections balancing usability with security, economic deterrents to make attacks financially unsustainable, and behavioral-based anomaly detection to adapt to evolving attack tactics.

We encourage the research community to further investigate these complex issues and develop innovative solutions to detect and mitigate such threats. By advancing our understanding and improving detection techniques, we can build more resilient defenses that keep pace with the ever-changing landscape of automated attacks.

References

[1] C. Watson and T. Zaw, "OWASP Automated Threat Handbook Web Applications Version 1.2," OWASP Foundation, Tech. Rep., 2018.

[2] Wikipedia, "HiQ Labs v. LinkedIn," 2023. [Online]. Available: https://en.wikipedia.org/wiki/HiQ_Labs_v._LinkedIn

[3] H. Ravia and D. Hammer, "Court Dismisses Data Scraping Lawsuit Against Bright Data," 2024. [Online]. Available: https://www.pearlcohen.com/court-dismisses-data-scraping-lawsuit-against-bright-data/

[4] "2024 Bad Bot Report," Imperva a Thales company, Tech. Rep., 2024.

[5] E. Chiapponi, M. Dacier, O. Thonnard, M. Fangar, M. Mattsson, and V. Rigal, "An industrial perspective on web scraping characteristics and open issues," in *2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks - Supplemental Volume (DSN-S)*, 2022, pp. 5–8.

[6] B. A. Azad, O. Starov, P. Laperdrix, and N. Nikiforakis, "Web Runner 2049: Evaluating Third-Party Anti-bot Services," *Detection of Intrusions and Malware, and Vulnerability Assessment*, vol. 12223, pp. 135 – 159, 2020.

[7] "The surprising impact of sophisticated bots," HUMAN, Tech. Rep., 2022.

[8] Arkose Labs, "Tis the Season for Denial of Inventory Attacks," 2023. [Online]. Available: https://www.arkoselabs.com/blog/season-denial-of-inventory/

[9] Vercara a Digicert company, "Seat Spinning Attacks on Airline APIs," 2024. [Online]. Available: https://vercara.com/resources/seat-spinning-attacks-on-airline-apis

[10] "How bots affect airlines," Imperva, Tech. Rep., 2019.

[11] B. Cost, "'Genius' plane hack allows passengers to avoid dreaded middle seat — without paying," *New York Post*, 2024.

[12] M. Sahin, A. Francillon, P. Gupta, and M. Ahamad, "SoK: Fraud in Telephony Networks," in *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2017, pp. 235–250.

[13] Federal Communications Commission, "Intercarrier Compensation," 2024. [Online]. Available: https://www.fcc.gov/general/intercarrier-compensation-0

[14] "International premium rate number market," TransNexus, Tech. Rep., 2015.

[15] "Twitter is getting scammed by phone companies for $60M/year of fake 2FA SMS message," 2023. [Online]. Available: https://x.com/elonmusk/status/1626996774820024321

[16] Twitter, "An update on two-factor authentication using SMS on Twitter," 2023. [Online]. Available: https://blog.x.com/en_us/topics/product/2023/an-update-on-two-factor-authentication-using-sms-on-twitter

[17] "A2P SMS Under Siege - Artificially Inflated Traffic and its Impact on the Industry," ENEA and Mobilesquared, Tech. Rep., 2024.

[18] K. Lempereur, "What is SMS pumping, and how does it impact your business?" 2023. [Online]. Available: https://datadome.co/learning-center/what-is-sms-pumping-how-does-it-impact-your-business/

[19] Arkose Labs, "SMS Pumping Fraud: How to Spot and Stop It," 2025. [Online]. Available: https://www.arkoselabs.com/toll-fraud/sms-pumping-fraud/

[20] Synch, "Artificial Inflation of Traffic (AIT): A growing threat to the messaging ecosystem," 2023. [Online]. Available: https://sinch.com/blog/artificial-inflation-traffic-ait-growing-threat-messaging-ecosystem/

[21] DataDome, "Denial of inventory attacks, inventory hoarding, & shopping bots—how to prevent them," 2022. [Online]. Available: https://datadome.co/learning-center/prevent-shopping-bots-denial-of-inventory-attacks//

[22] Reblaze, "Inventory hoarding," 2024. [Online]. Available: https://www.reblaze.com/wiki/threats/inventory-hoarding/

[23] E. Khan, E. Chiapponi, M. Verkleij, A. Sperotto, R. Van Rijswijk-Deij, and J. Van Der Ham-De Vos, "A First Look at User-Installed Residential Proxies From a Network Operator's Perspective," in *2024 20th International Conference on Network and Service Management (CNSM)*, 2024, pp. 1–9.

[24] P. Gosset and M. Hyland, "Classification, detection and prosecution of fraud in mobile networks," in *Proceedings of ACTS mobile summit*, no. 1, 1999, pp. 2–4.

[25] M. Sahin and A. Francillon, "Understanding and detecting international revenue share fraud," in *NDSS 2021, Network and Distributed System Security Symposium, 21-24 February 2021, Virtual Conference*, ISOC, Ed., 2021.

[26] C. V. Richard A. Becker and A. R. Wilks, "Fraud Detection in Telecommunications: History and Lessons Learned," *Technometrics*, vol. 52, no. 1, pp. 20–33, 2010.

[27] C. Gibson, "Toll fraud, international revenue share fraud and more: How criminals monetise hacked cell phones and IoT devices for telecom fraud," EUROPOL and Trend Micro, Tech. Rep., 2018.

[28] H. Debar, M. Dacier, and A. Wespi, "A revised taxonomy for intrusion-detection systems," in *Annales des télécommunications*, vol. 55, no. 7. Springer, 2000, pp. 361–378.

[29] S. Rovetta, A. Cabri, F. Masulli, and G. Suchacka, "Bot or not? a case study on bot recognition from web session logs," in *Quantifying and Processing Biomedical and Behavioral Signals*. Springer International Publishing, 2019, vol. 103, pp. 197–206.

[30] M. Zabihimayvan, R. Sadeghi, H. N. Rude, and D. Doran, "A soft computing approach for benign and malicious web robot detection," *Expert Systems with Applications*, vol. 87, pp. 129–140, 2017.

[31] D. S. Sisodia, S. Verma, and O. P. Vyas, "Agglomerative approach for identification and elimination of web robots from web server logs to extract knowledge about actual visitors," *Journal of Data Analysis and Information Processing*, vol. 03, no. 1, pp. 1–10, 2015.

[32] S. Rovetta, G. Suchacka, and F. Masulli, "Bot recognition in a web store: An approach based on unsupervised learning," *Journal of Network and Computer Applications*, vol. 157, p. 102577, 2020.

[33] K. Li, M. Xiang, M. Kakaiya, S. Kaul, and X. Wang, "Web bot detection based on hidden features of HTTP access log," in *Tools for Design, Implementation and Verification of Emerging Information Technologies*. Springer Nature Switzerland, 2023, vol. 489, pp. 32–43.

[34] G. Buehrer, J. W. Stokes, K. Chellapilla, and J. C. Platt, "Classification of automated search traffic," *Weaving Services and People on the World Wide Web*, pp. 3–26, 2009.

[35] M. Campobasso, P. Burda, and L. Allodi, "CARONTE: Crawling adversarial resources over non-trusted, high-profile environments," in *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 2019, pp. 433–442.

[36] C. Iliou, T. Tsikrika, S. Vrochidis, and I. Kompatsiaris, "Evasive focused crawling by exploiting human browsing behaviour: a study on terrorism-related content," Feb. 2017.

[37] C. Iliou, T. Kostoulas, T. Tsikrika, V. Katos, S. Vrochidis, and I. Kompatsiaris, "Web bot detection evasion using deep reinforcement learning," in *Proceedings of the 17th International Conference on Availability, Reliability and Security*. ACM, 2022, pp. 1–10.

[38] M. Zabihi, M. V. Jahan, and J. Hamidzadeh, "A density based clustering approach for web robot detection," in *2014 4th International Conference on Computer and Knowledge Engineering (ICCKE)*. IEEE, 2014, pp. 23–28.

[39] J. Kadel, A. See, R. Sinha, and M. Fischer, "BOTracle: A framework for discriminating bots and humans."

[40] D. Doran and S. S. Gokhale, "An integrated method for real time and offline web robot detection," *Expert Systems*, vol. 33, no. 6, pp. 592–606, 2016.

[41] Z. Chu, S. Gianvecchio, and H. Wang, "Bot or human? a behavior-based online bot detection system," *From Database to Cyber Security: Essays Dedicated to Sushil Jajodia on the Occasion of His 70th Birthday*, pp. 432–449, 2018.

[42] H. Niu, J. Chen, Z. Zhang, and Z. Cai, "Mouse dynamics based bot detection using sequence learning," in *Biometric Recognition: 15th Chinese Conference, CCBR 2021, Shanghai, China, September 10–12, 2021, Proceedings 15*. Springer, 2021, pp. 49–56.

[43] A. Wei, Y. Zhao, and Z. Cai, "A deep learning approach to web bot detection using mouse behavioral biometrics," in *Biometric Recognition: 14th Chinese Conference, CCBR 2019, Zhuzhou, China, October 12–13, 2019, Proceedings 14*. Springer, 2019, pp. 388–395.

[44] A. Acien, A. Morales, J. Fierrez, R. Vera-Rodriguez, and O. Delgado-Mohatar, "Becaptcha: Behavioral bot detection using touchscreen and mobile sensors benchmarked on humidb," *Engineering Applications of Artificial Intelligence*, vol. 98, p. 104058, 2021.

[45] A. Vastel, W. Rudametkin, R. Rouvoy, and X. Blanc, "FP-Crawlers: studying the resilience of browser fingerprinting to block crawlers," in *MADWeb'20-NDSS Workshop on Measurements, Attacks, and Defenses for the Web*, 2020.

[46] G. Acar, M. Juarez, N. Nikiforakis, C. Diaz, S. Gürses, F. Piessens, and B. Preneel, "FPDetective: dusting the web for fingerprinters," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, 2013, pp. 1129–1140.

[47] J. Jueckstock and A. Kapravelos, "Visiblev8: In-browser monitoring of javascript in the wild," in *Proceedings of the Internet Measurement Conference*, 2019, pp. 393–405.

[48] P. Laperdrix, W. Rudametkin, and B. Baudry, "Beauty and the beast: Diverting modern web browsers to build unique browser fingerprints," in *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2016, pp. 878–894.

[49] S. Wu, P. Sun, Y. Zhao, and Y. Cao, "Him of Many Faces: Characterizing Billion-scale Adversarial and Benign Browser Fingerprints on Commercial Websites." in *NDSS*, 2023.

[50] E. Bursztein, A. Malyshev, T. Pietraszek, and K. Thomas, "Picasso: Lightweight device class fingerprinting for web clients," in *Proceedings of the 6th Workshop on Security and Privacy in Smartphones and Mobile Devices*, ser. SPSM '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 93–102.

[51] H. Venugopalan, S. Munir, S. Ahmed, T. Wang, S. T. King, and Z. Shafiq, "Fp-inconsistent: Detecting evasive bots using browser fingerprint inconsistencies," *arXiv preprint arXiv:2406.07647*, 2024.

[52] E. Boulila, M. Dacier, S. P. Vengadessa Peroumal, N. Veys, and S. Aonzo, *A Closer Look at Modern Evasive Phishing Emails*, 2025.

[53] E. Chiapponi, M. Dacier, O. Catakoglu, O. Thonnard, and M. Todisco, "Scraping Airlines Bots: Insights Obtained Studying Honeypot Data," *International Journal of Cyber Forensics and Advanced Threat Investigations*, vol. 2, no. 1, pp. 3–28, 2021.