Inconsistent, Incomplete, and Insecure: A Survey of Account Security Interfaces

Arkaprabha Bhattacharya
Cornell University

Alaa Daffalla
Cornell University

Kevin Lee
Independent Researcher

Rosanna Bellini
New York University

Nicola Dell
Cornell Tech

Thomas Ristenpart
University of Toronto & Cornell Tech

Abstract

Despite improvements in account security, compromise remains widespread and damaging, especially when the attacker has close physical or social proximity to the victim (e.g., interpersonal abuse settings). To help users identify unauthorized access, web services provide account security interfaces (ASIs): notifications and logs that provide information to help infer adversarial compromise. We present the largest measurement study of ASIs to date, evaluating 100 popular services.

Our study highlights an unsatisfying status quo: 29 services provided users with no way to distinguish account accesses. After categorizing ASIs using a new typology, we show that services were inconsistent in the types they deployed. Further, ASIs were often incomplete and confusing, even for expert researchers. Finally, of 61 services that offered an ASI to convey device or location descriptions, 41 (67.2%) were vulnerable to spoofing attacks that successfully obfuscate the source of the access. Based on these findings, we present six principles for improving future ASI deployments.

1 Introduction

Account compromise is widespread and, especially in interpersonal abuse contexts, remains a primary gateway to harmful attacks like monitoring, control, harassment, impersonation, and stalking [14, 24]. Web services frequently advise users who suspect compromise to examine account activity for "suspicious" behavior [5, 30]—a daunting task in the presence of multiple devices and complex account ecosystems. To facilitate this investigation, many web services provide what prior work [14] calls *account security interfaces* (ASIs): user-facing tools that enable people to identify signs of account compromise via records of account access, activity history, connected devices, and security-specific account actions.

Unfortunately, emerging study of ASIs suggests that these tools are fraught with design and implementation issues. Daffalla et. al [14] studied four major web services (Google, Apple, Facebook, and WhatsApp), surfacing signs that ASIs are under-delivering on their intended use. These four web services' (services hereafter) ASIs were challenging to navigate, confusing to interpret, and easily manipulated by technically unsophisticated adversaries, allowing potential threats to spoof or conceal illicit account accesses [14]. However, this prior work focused on only four services; how their results generalize across the web is thus an open question.

We therefore conducted a broader measurement study of ASIs, examining 100 popular services (based on Tranco ranking [63]). Via a systematic investigation, we discovered 200 ASIs across 71 web services. This means that 29 popular services provided *no user-visible logs or notifications* about account access, a startling security problem for users.

For the remaining 71 services, we analyzed the 200 discovered ASIs, developing a typology that expands the ASI types found in prior work [14]. This typology provides consistent terminology to use when describing different ASI types. Our typology consisted of seven ASI types: two types of off-service notifications (security notifications, verification notifications), and five types of on-service logs (security logs, activity logs, device logs, audit logs, and session logs). We observed a number of usability and correctness issues with ASIs, some of which are shown in Figure 9 in the Appendix.[1]

We found that services deployed different combinations of ASIs, which often presented inconsistent records of account activity. Further, some companies (e.g., Microsoft) attempted to unify ASIs across affiliated services (*office.com*, *outlook.com*, etc.) via account management portals. These portals failed to provide granular information about what affiliated service an access came from. Moreover, we discovered that navigating to ASIs can require users to click through multiple intermediary pages with labels and descriptions that use inconsistent terminology.

Finally, we did an evaluation of how informative ASIs are for helping users diagnose what devices have accessed their accounts. We performed manual stepthroughs [36] of the ASIs on each service, observing the information shown as

[1] Screenshots here and elsewhere in the paper have been lightly edited to remove whitespace and improve clarity.

we simulated illicit adversarial access from a second device. Beyond the 29 services with no ASIs, a further 10 services either had ASIs with no information or only showed temporal details about our accesses (39 services in total). These may not be directly useful for determining if a second device has accessed the account. We also demonstrated that 41 of the remaining services with ASIs that provided device information (OS, browser, etc.) were vulnerable to spoofing.

In sum, our paper makes several contributions. First, we survey ASIs across 100 popular services that support user accounts from the TRANCO list, expanding on Daffalla et al. [14] which looked at four, and other works [40, 51] that exclusively looked at notifications. . Our findings show that at least 39 services provided insufficient information to determine if a second device has accessed the account.

We analyze the types of ASIs provided, constructing a new typology of seven ASI types, including two new log types and a new notification type previously unidentified by Daffalla et al [14]. We show a lack of consistency across ASIs, difficulty of navigation to web-based ASIs, and incompleteness of logs. Finally, we demonstrate that most services are vulnerable to spoofing attacks that degrade users' ability to determine what devices are accessing their accounts. Thus, the issues highlighted by Daffalla et al. [14] are prevalent across popular services.

Together, our results highlight the poor state of ASIs in practice, motivating the need for future work to improve such tools. We facilitate this in three ways. First, we disclosed the identified issues to services, the details of which can be found in Appendix A.1. Second, we present a set of six principles for improving future ASI implementations. Finally, we released our data for researchers to build off our results.[2]

2 Threat Model and Related Work

Online account security is becoming increasingly complex, with rising prevalence of vulnerabilities such as password guessing attacks [31], cookie hijacking [19], and account pre-hijacking [58]. Services have deployed several mechanisms to help keep accounts secure, namely, stronger login mechanisms (e.g., passkeys [38] or multi-factor authentication [26]), and affordances that help users diagnose illicit access [14].

For users who face a higher risk of harm from compromise (e.g., survivors of interpersonal abuse [25], journalists [41], refugees [55], activists [15]), stronger login mechanisms can fall short [7, 14]. For these groups, identifying illicit access, and in some scenarios, who illicitly accessed their account, is critical. Here, we describe a threat model centered on identifying illicit access, and discuss work in account security.

Threat model. Our threat model considers an adversary who has gained unauthorized account access (beyond the interpersonal scenarios considered in [14]). The adversary's goal is to access the target's account without populating any interfaces that inform the target, or by making their activity appear as if it originated from the target's own device, similar to Daffalla et al. [14]. To achieve this, the adversary can reconfigure software settings, install widely available applications, and modify browser configurations.

Sending authentication challenges. To defend against account compromise, services may employ multifactor authentication (MFA) or risk-based authentication (RBA). In MFA, a service sends an authentication challenge (typically a one-time code) to a user chosen contact (e.g., email, phone number, or authenticator app), on every new login. For RBA, a service assesses each login based on a set of identifiers (e.g., device fingerprint [37]), and sends an authentication challenge to the user (e.g., through a one-time code or secret question) when a login attempt is deemed suspicious.

Much prior work has examined the adoption and efficacy of these tools [16, 28, 37, 44, 49, 65]. For example, Quermann et al. [45] characterized user authentication and recovery tools on 48 services, finding that most still relied on passwords and few enabled two-factor authentication by default. Gavazzi et al. studied the availability of MFA and RBA for 208 popular services [26], finding that less than half of the services they assessed supported MFA, and less than a quarter supported RBA. The limited adoption of security features reinforces the need for effective tooling for identifying illicit access.

Prior studies have also found that these mechanisms fall short in design and usability. Ghorbani et al. [27] manually explored 2FA journeys on 85 popular services, finding minimal consistency in design patterns for both configuration and usage. Moreover the few consistent design elements, such as 2FA descriptions, did not follow established UX best practices. While these studies studied the adoption and usability of RBA and MFA, they did not engage with how services inform users of illicit access post-compromise.

Post-compromise detection and remediation. Prior studies have devised automated methods for detecting account compromise. However, most focus on detection tools for external observers, such as service providers. For example, Thomas et al. [60] identified 13 million hijacked accounts on Twitter from 8.7 billion tweets through the use of classifiers. They found that account compromise often spreads through phishing and malware campaigns across users' social graphs. Ruan et al. [50] developed a compromise detection system based on a user's social network behaviors, showing high accuracy in differentiating a sample of Facebook users. While these are valuable methods for service providers, they do not address how users can manually identify illicit access on their account. Our study fills this gap by assessing logs and notifications that aim to enable users to infer suspicious activity.

Studies have also evaluated service guidance for account remediation: the process by which users regain control of their accounts post-compromise [39, 42]. Neil et al. [42] performed

[2] Available at: `https://doi.org/10.5281/zenodo.18223588`

a measurement of account remediation advice, identifying five stages of remediation: compromise discovery, account recovery, limiting access, service restoration, and prevention [42]. Markert et al. [39] extended this in a transcontinental analysis of remediation advice across 158 websites, confirming that services fail to adequately advise users on all account remediation. Most services in their study fell short of providing sufficient advice for detecting compromise. While providing tangible recommendations for improving account remediation advice, these studies did not analyze the actual features services implement for account remediation. We build on these studies by highlighting which services do and do not provide affordances to facilitate compromise detection, and discuss the (lack of) guidance on many of them.

Security notifications. One way that users may learn that their accounts are compromised or are at risk of compromise is through notifications. Many prior studies have examined security notifications for operating systems and application security [11], Transport Layer Security (TLS) [2], phishing and malware detection [18,20,54], and browser security [3]. In authentication, prior work has studied the usability of password reuse and breach alerts [29,62].

Services have since extended such warnings for account security purposes. Shay et al. conducted a large-scale survey with 294 participants to understand people's experiences with account hijacking [53], finding that only 29% of participants were notified about an account compromise by the service provider; most people learned about compromise from other people who told them about suspicious activity originating from their accounts. Since then, a handful of studies investigated how users might learn about account compromise via service-provided security notifications [4,29,40,47,51,53].

For example, Markert et al. [40] studied how users react and respond to suspicious login notifications sent to a user's email. They found that after a suspicious login notification, around 80% of respondents did not take preventative measures (e.g., password reset) to protect their accounts. The authors attributed this to users' lack of understanding of the notifications and suggested refining notification designs to clearly explain why they were sent. They also found that while most users expected services to send login notifications, some people experienced notification or warning fatigue. Although this work examined notification designs, little is known about how a lack of clear explanations may extend to other interfaces for diagnosing illicit access (e.g., on-service logs). Our study answered this question by analyzing the design patterns and descriptions of a broader set of ASIs.

Prior studies on security notifications do not consider how consistent they are with other service affordances for diagnosing illicit access. This is critical for ensuring users have accurate and actionable information. Our study filled this gap by identifying key terminological inconsistencies and completeness issues across both notifications and logs.

Identifying illicit accesses on accounts. The closest prior work to ours was by Daffalla et al. [14], who examined account security interfaces (ASIs): on-service interfaces and notifications that provide users with a history of account accesses. The authors began by investigating the utility of ASIs in technology abuse clinics. Then, they explored ASIs on four services: Google, Facebook, Apple and WhatsApp, finding four distinct design types: session lists, device lists, account activity logs, and access notifications. They also tested the integrity of information reported on ASIs, finding that device information was easily spoofed and ASIs often enabled adversaries to hide their logins. While this work provided key insights into both notifications and on-service logs for identifying illicit access, it is unknown whether their findings extend to a larger sample of services, and whether they were comprehensive in discovering all types of ASIs.

Thus, we performed a broader measurement of ASIs, characterizing the designs, deployment strategies, navigation patterns, and security of ASIs on 100 popular services. For measuring each service, we designed manual stepthroughs that replicated the methodology of Daffalla et al. [14]: we populated each service's ASIs with a series of logins, some of which used spoofed device identifiers. To ensure that our measurement was comprehensive, we expanded the scope of ASIs beyond just account access-related interfaces. Here, an ASI is any log or notification of account activity that can help a user infer if illicit access occurred. Examples include logs of account accesses, connected devices, password changes, account activity logs, and security notifications.

3 Methods

Our goal was to provide the largest to date characterization of ASIs. Thus, we created and followed systematic manual protocols to: (1) discover ASIs across 100 popular services; (2) characterize the ASIs discovered via an ASI typology and, (3) measure the utility and integrity of the discovered ASIs.

Service selection. We studied ASIs via manual UI stepthroughs, a method of UI testing that requires researchers to individually interact with each service. Prior work suggests that account security feature deployments are sparser and weaker outside the most popular services [8,26,34,42], meaning a study of the most popular sites may provide a reasonable coverage of the ASIs users may encounter. Thus, we scoped our study to 100 popular services, selected from the Tranco list (ID: 7XN5X) generated on May 7, 2024 [3]. Tranco is a service that orders URLs based on a popularity heuristic hardened against malicious actors [33]. Using the list, we manually selected the top 100 services that supported an obvious flow for account creation and did not return an error code (e.g., HTTP 404), redirect to a URL already on the list, or

[3] Available at https://tranco-list.eu/list/7XN5X

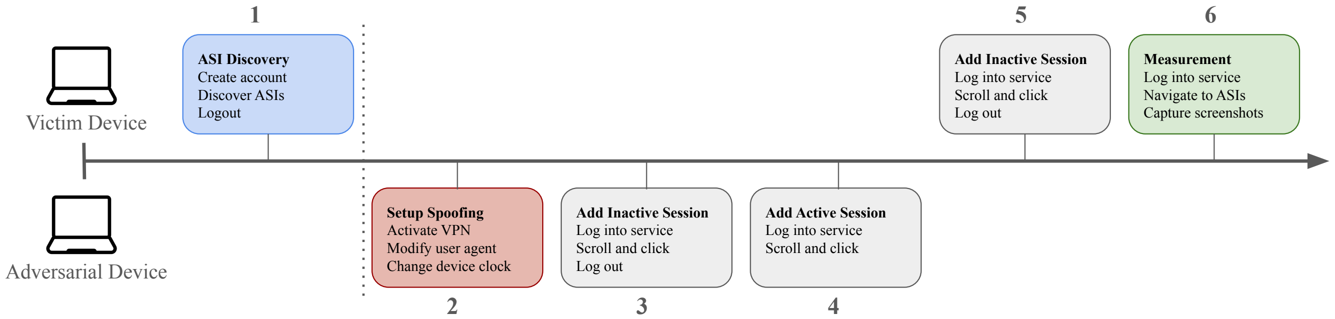


Figure 1: An overview of our methodology for discovering and characterizing ASIs. We begin by discovering ASIs using the victim device setup with system defaults. A second (attacker) device is set up with spoofed device information. We generate an inactive and an active session on the attacker device, and an inactive and active session on the victim device. On the final active session of the victim device, we stepthrough the service to capture information and screenshots of all previously discovered ASIs.

lead to a non-English website. While previous measurement studies have assessed non-English websites effectively using automated translation tools [17], we chose not to use this approach, as any distortion in translation could obscure terminology used in ASI descriptions. We also omitted services that required interacting with customer service or providing personal information (e.g., credit card, social security number, etc.) for account creation. We found that our 100 services achieved saturation (no new results) for different ASI types and information attributes.

Of the 100 services, 94 were consumer-facing and six were enterprise-facing. Fifteen were social media (e.g., *instagram.com*), twelve productivity services (e.g., *office.com*), and another twelve offered multiple functionalities (e.g., *google.com*). The others comprised long tail of areas (e.g., communication/messaging (7) and e-commerce (4)).

ASI discovery protocol. We designed a semi-structured protocol for discovering ASIs on our selected services (Step 1 in Figure 1) with the goal of capturing the ASIs a general user represented via our threat model may find by navigating the webpage using visible UI elements (e.g., buttons, hyperlinks). For consistency, we scoped our study to focus only on web-based ASIs. Three researchers split the 100 services and were tasked with: (1) creating an account on the service; (2) logging into the service; and, (3) manually searching for any logs of user activity that could help infer illicit access by navigating the website using UI elements. Given the overhead of manual search, we capped each search at 30 minutes.

We counted each separately labeled log as a distinct ASI. Sometimes, multiple logs appeared on one webpage. We separated these based on whether they were under different headers or visually separate UI containers. If a log contained multiple entries that could be expanded or clicked out to the same type of UI element, we only counted the expanded ASI once. The researchers captured any loosely relevant logs or UI containers when in doubt to ensure our subsequent analysis was as comprehensive as possible. Any disagreements on whether an interface was an ASI were resolved via team discussion.

We also gathered any access-relevant notifications sent via email or phone during account setup. However, we exclude 2FA challenge emails from consideration if they contained no access information other than the challenge request.

Populating ASIs. To populate discovered ASIs, we performed manual stepthrough analyses of each service. Stepthrough analyses are part of a collection of methods, known as walkthrough approaches [35] originating from software engineering [22], that aim to identify how users may deviate from intended system pathways. This approach involves systematically navigating each screen, feature, and activity flow of an interface along with careful documentation [36].

Our protocol first simulated active and inactive logins by both a legitimate user and an adversary with illicit access to the account. An active session simulated an adversary that was currently logged into the service. An inactive session simulated an adversary that had logged out of the service. Then, we used stepthroughs to investigate each service's ASIs.

An overview of our procedure is shown in Steps 2–6 in Figure 1. In Step 2, we set up a device to simulate an attacker that can login to the target account. To achieve this, we used (i) a VPN with proxy in a different geographic location (we used ProtonVPN and a Socket Secure (SOCKS) proxy); (ii) a modified HTTP user-agent string within Chrome; and (iii) the clock set to be three hours behind the time of test.

Then, in Step 3 we created an inactive/logged out session on the attacker's device. We used the attacker device to log in, scroll through the UI, click on a button/link, and logout. We then cleared the browser profile's state (browsing history, cache, autofill, and cookies).

In Step 4, we created an active session on the attacker's device. We logged in again from the attacker device, scrolled and clicked, but did not log out.

In Step 5, we created an inactive/historical session on the victim's device. We logged in, scrolled and clicked, logged out, and cleared the browser profile's state.

Finally, in Step 6 we created an active session on the victim's device. We logged in, navigated to all discovered ASIs, documented the visible information on ASIs, and took screen captures of all ASIs from the victim's perspective. We also collected any notifications sent to an email or phone number on the account. Each stepthrough therefore involved five sessions: three on the victim's device (Steps 1, 5, & 6) and two on the adversary's device (Steps 3 & 4).

We captured both manual and timed screenshots in Step 6 of our protocol to enable collection and analysis of ASI navigation paths. To do this, we developed a Google Chrome Extension that records screenshots (1) at every user click or keyboard press, (2) at regular five second intervals, and (3) on demand via a button press. Details of our extension are available via our Open Science repository.

Three researchers performed stepthroughs for all 71 services with at least one ASI. We used three pairs of devices to run the protocol across these services: three Macbook Pros (two 2023 and one 2022 version), all running OS X 14.5 Sonoma, one Macbook Air (2020 version) running OS X 12.7 Monterey, and two Dell XPS devices running Windows 10. All testing was done on Google Chrome 125, in empty Chrome profiles with default configurations, between May 21st, 2024 and July 21st, 2025 (see Open Science repository), with each stepthrough lasting no more than 20 minutes.

Building an ASI typology. To systematically characterize ASI designs, we developed a typology by analyzing the screenshots from our stepthroughs. Typologies are hierarchical systems of categories used to organize objects according to their similarities and differences [57]. Typologies may be constructed using ideal-type analysis: an exploratory method that involves identifying features to partition large datasets into types, making it useful for characterizing ASIs.

To begin, we used deductive and inductive coding [9, 52] to annotate features from screen captures of each ASI. Two researchers performed multiple passes over each screenshot, coding the labels that services used to describe the ASI (e.g., 'Account Activity'), the information attributes included (e.g., date, time, IP address), and any actionable buttons/links (e.g., 'logout', 'change password'). Six rounds of coding resulted in two stable codebooks, one for information attributes (16 codes), and one for functionality (19 codes), which we provide in Figure 10 in the Appendix. Inter-rater reliability (IRR) was calculated on the last round between two coders using Krippendorff's alpha (at 0.87 "Very Good") [32].

The coded screenshots, along with the location and service descriptions of each ASI were then used to build our typology. Researchers clustered ASIs into similar types based on shared features, via a series of weekly meetings over two months to resolve disagreements around each type. This resulted in seven distinct ASI types (see Figure 2), expanding the 4 from Daffala et al. [14]: two types of notifications and five types of logs. To verify that we achieved good coverage of ASI designs, we counted services in ranked order until no new information attributes or ASI types were found.

Limitations. While we are confident that we achieved good coverage of ASI designs, we acknowledge that the top 100 may not generalize across every ASI deployment for all device ecosystems. Specifically, our selection omitted some services that are lower ranked in popularity, but may contain private information (e.g., financial or health data) and did not consider mobile apps. Future research might scale our methods to a more diverse range of services and different app ecosystems.

In addition, although our discovery process was thorough, it was not exhaustive: manually searching service pages may have overlooked some ASIs. Furthermore, some services have ASIs that are difficult to find, or are hidden functionality not meant for general users. For example, after completing our protocol and stepthroughs, we found that *gmail.com* has a small link labeled "Details" at the bottom right of the webpage which opens an ASI (see Figure 8 in the Appendix). We note that including this ASI would not have yielded new design elements (information attributes, user actions, etc.) or ASI types, signaling that we achieved good coverage.

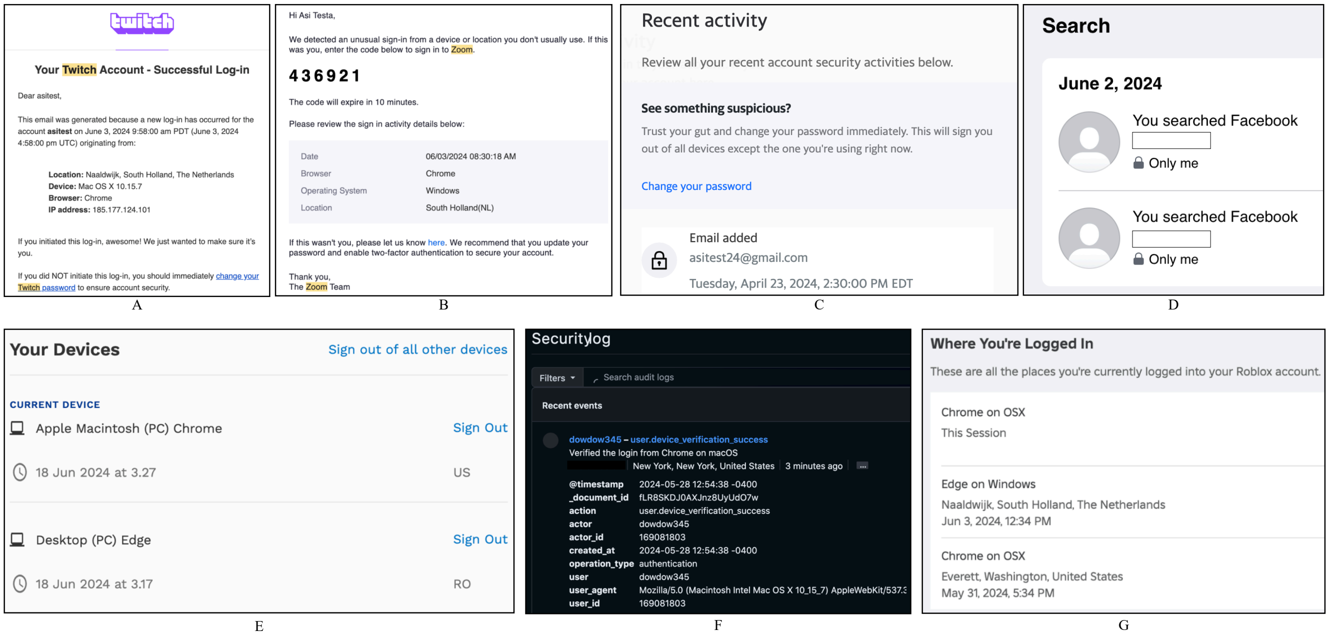
Our stepthroughs may also have missed some ways that users identify illicit accesses, as some services may populate ASIs or send notifications under specific account conditions not triggered by our protocol. That said, a service's ASIs should still sufficiently log the adversarial accesses from our stepthroughs. Thus, we consider ASIs that do not enable users to infer account access from multiple devices as deficient.

Our choice of browser (Chrome, representing 50% of the global browser market share [6]) may also have played an impact; browsers vary in the device information they expose, potentially affecting how web services identify devices and sessions [46].

4 ASIs (Not) Found in Practice

ASI-less services are widespread. We found that 29 services —almost one third of the 100 we explored—had no notifications or security-relevant logging for users. We list these 29 ASI-less services in Figure 7 in the appendix; examples include *booking.com*, *avast.com*, and *flickr.com*. While some ASI-less services may not be considered security-critical by some users (e.g., seven were news sites), service accounts may still contain important private information requiring careful access control. Since we selected services based on popularity (Section 3), this suggests that many popular online services provide no way for users to monitor illicit access.

Types of ASIs. We discovered 200 unique ASIs from the 71 remaining services (see Figure 11 in the Appendix). While most ASIs displayed account activity from one service, some unified activity across multiple services affiliated with the same company (e.g., *apple.com* and *icloud.com* are affiliated with Apple). We call these *unified ASIs*. We captured seven ASI types encompassing a broad variety of designs. All seven



twitch

Your Twitch Account - Successful Log-in

Dear asitest,

This email was generated because a new log-in has occurred for the account **asitest** on June 3, 2024 9:58:00 am PDT (June 3, 2024 4:58:00 pm UTC) originating from:

Location: Naaldwijk, South Holland, The Netherlands
Device: Mac OS X 10.15.7
Browser: Chrome
IP address: 185.177.124.101

If you initiated this log-in, awesome! We just wanted to make sure it's you.

If you did NOT initiate this log-in, you should immediately change your Twitch password to ensure account security.

A

Hi Asi Testa,

We detected an unusual sign-in from a device or location you don't usually use. If this was you, enter the code below to sign in to Zoom.

436921

The code will expire in 10 minutes.

Please review the sign in activity details below:

Date	06/03/2024 08:30:18 AM
Browser	Chrome
Operating System	Windows
Location	South Holland(NL)

If this wasn't you, please let us know here. We recommend that you update your password and enable two-factor authentication to secure your account.

Thank you,
The Zoom Team

B

Recent activity

Review all your recent account security activities below.

See something suspicious?

Trust your gut and change your password immediately. This will sign you out of all devices except the one you're using right now.

Change your password

Email added
asitest24@gmail.com
Tuesday, April 23, 2024, 2:30:00 PM EDT

C

Search

June 2, 2024

You searched Facebook
Only me

You searched Facebook
Only me

D

Your Devices Sign out of all other devices

CURRENT DEVICE

Apple Macintosh (PC) Chrome Sign Out
18 Jun 2024 at 3.27 US

Desktop (PC) Edge Sign Out
18 Jun 2024 at 3.17 RO

E

Securitylog

Filters ▾ Search audit logs

Recent events

dowdow345 – user.device_verification_success
Verified the login from Chrome on macOS
[redacted] | New York, New York, United States | 3 minutes ago | ...

```
@timestamp      2024-05-28 12:54:38 -0400
_document_id    fLR8SKDJ0AXJnz8UyUdO7w
action          user.device_verification_success
actor           dowdow345
actor_id        169081803
created_at      2024-05-28 12:54:38 -0400
operation_type  authentication
user            dowdow345
user_agent      Mozilla/5.0 (Macintosh Intel Mac OS X 10_15_7) AppleWebKit/537.3
user_id         169081803
```

F

Where You're Logged In

These are all the places you're currently logged into your Roblox account.

Chrome on OSX
This Session

Edge on Windows
Naaldwijk, South Holland, The Netherlands
Jun 3, 2024, 12:34 PM

Chrome on OSX
Everett, Washington, United States
May 31, 2024, 5:34 PM

G

Figure 2: The seven types in our ASI typology: **A**: security notification (Twitch); **B**: verification notification (Zoom), **C**: security log (Yahoo); **D**: activity log (Facebook); **E**: device log (Forbes); **F**: audit log (Github); **G**: session log (Roblox).

and their information attributes appeared in the 12 highest ranked services, signaling that our sample covered the breadth of ASI designs, with the remaining 59 adding no new results.

Our typology (Figure 2) consisted of two types of notifications and five types of logs. Notifications report a single event or activity, typically via off-service communication channels (e.g., text or email). Notifications in our typology have two types: *security notifications* and *verification notifications*. Logs are on-service UIs that capture traces of users' activities [21]. Our typology has five log types: *security logs*, *activity logs*, *device logs*, *audit logs*, and *session logs*. While different ASI types may share certain attributes, they are still unique in what they show to users. Figure 3 provides a decision tree for determining a given ASI's type. We now discuss each ASI type in detail.

Security notifications (Figure 2-A). Security notifications alert users about significant security changes to the account. Daffalla et al. referred to these as *access notifications* [14], but we use security notifications since they can show other kinds of security activity. Some security notifications used misleading terminology. For example, *shopify.com* sent five notifications titled "A new device has logged in to your Shopify account ", when we only used two devices during our stepthroughs and what was really reported were new sign-ins. Terminological inconsistency was widespread amongst ASIs.

We discovered 30 distinct security notification designs across 30 services. *apple.com* and *icloud.com* deployed the same two security notification designs: one via email, and one to the device. These were the only examples of unified security notifications. While most sent notifications to email, we also saw three services send security notifications as an on-service pop-up (e.g, *twitter.com*), and one sent a notification via SMS (*amazonvideo.com*).

Most security notifications described login attempts, including browser and OS type and version, location, date, and time. Users were typically provided a quick link button to change their password if the notification was deemed suspicious, with some services combining this with reporting the login (e.g., *grammarly.com*'s *'this wasn't me!'* feature).

Not all services sent security notifications upon each login: some services only sent them when the user logged in with a new device, while others triggered them when a new browser or app was used. This may stem from risk-based methods that determine what constitutes a login worth reporting.

Verification notifications (Figure 2-B). Verification notifications are authentication challenges containing login, device, and, access-related information that users can use to infer who is accessing their account, distinguishing them from security notifications, which provide access-related information but no authentication challenge/code. We found eight verification notification designs across 9 services (two Amazon-affiliated services used the same unified verification notification design). Most verification notifications show OS/browser type and version, and the location of the login attempt. Many verification notifications provided options for securing user accounts should the challenge be suspicious. Two services, *twitch.tv* and *salesforce.com*, sent a verification notification on every login. Meanwhile, seven others, like *zoom.us*, did

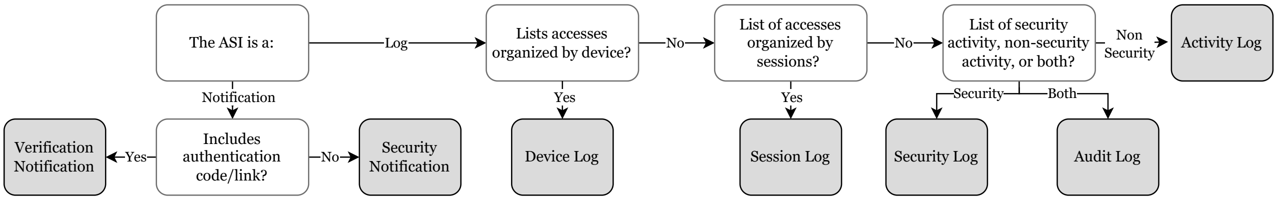

The ASI is a:
Log
Notification
Lists accesses organized by device?
No
Yes
List of accesses organized by sessions?
No
Yes
List of security activity, non-security activity, or both?
Non Security
Activity Log
Security
Both
Security Log
Audit Log
Session Log
Device Log
Includes authentication code/link?
Yes
No
Verification Notification
Security Notification

Figure 3: Decision tree representing how to decide an ASI's type based on our typology.

not (despite our protocol requiring cookie deletion). Each of these seven captured exactly one of the two adversarial sessions in our stepthrough. Like security notifications, we think this is rooted in service policies around when to send an authentication challenge to users.

Security logs (Figure 2-C). We define security activity as any user interactions that are relevant to logins, credential changes, or the configuration of different security features. A security log, then, is a descriptive list (often in reverse chronological order) of security activity on an account. While Daffalla et al. [14] referred to these as *activity logs*, we use security log to emphasize their focus on security-specific events. An entry in a security log typically includes a description of what happened, such as changes to authentication information (i.e., password, secret answers, recovery information), adding or removing 2FA, or log-in actions (e.g., attempts and successes). Security logs vary in specificity, ranging from providing a user agent string, IP address, date and time of each security action (e.g., *ubuntu.com*), to a simple description devoid of details such as 'your password has been changed' (e.g., *mozilla.org*).

We found 34 security logs across 31 services; some services shared unified security logs while others had more than one security log. Security logs can, but are not required to show entries pertaining to accesses. For example, *yahoo.com*'s "Recent activity" security log only showed one entry, which described an email being added upon account creation. Account accesses on *yahoo.com* were shown in a session log on the same page, "Review your connected devices and apps".

Activity logs (Figure 2-D). Activity logs show non-security account activity that may enable users to infer account access, distinguishing them from security logs which show security events. Examples of activity logs included search histories (e.g., streaming sites like *dailymotion.com*) and purchase histories (e.g., e-commerce sites like *ebay.com*). These were the most common ASI type; we found 58 activity logs across 28 services, and several services with multiple activity logs.

Since the actions a user can perform depend on the service, implementation of activity logs varied. However, all sites involving purchases contained logs that displayed order history, product warranty list, or past purchases, which were often accompanied by options for users to search or filter its entries. For web services that did not revolve around expenditure, search and browsing histories were more common.

Activity logs did not provide specific information attributes about access, because they, at first glance, are less security relevant. However, many activity logs displayed specific information about activity that can be used to infer access, such as the email address associated with a purchase or the date/time a search was performed. Many of these logs remained empty, as our stepthroughs did not generate such activity.

Device logs (Figure 2-E). Device logs provide a list of devices that have accessed an account, and do not show any other security or non-security activity, distinguishing them from security and activity logs. They are often identified by page headings like *'Your Devices'* (e.g., from *forbes.com*'s log shown in Figure 2). Daffalla et al. referred to these as *device lists* [14], but we use log since they display a device history. Devices in the log are typically shown as a list with icons representing devices like smartphones, tablets, and laptops.

We found 29 device logs across 35 services; several services shared unified device logs. Some device logs only display specific device types, such as ones the service sells itself, or ones that have downloaded their desktop application. For example, *roku.com* provides two device logs that log Roku-specific streaming and smart home devices. Since uniquely identifying devices accessing a web service is challenging, we envision some services may prioritize logging the devices they have more detailed information about. We discuss this further in Section 7. Some device logs organized devices according to their functionality or frequency of use. For instance, *google.com* categorizes devices by a device model extracted from the user agent string.

Although the exact information varies, most device logs display date, time, location, and operating system to aid identification. Some device logs also report active sessions on a connected device. Rarely, device-specific functionality is provided, such as removing an associated device from an account, or indicating a device's physical location (e.g., 'Find My' on *apple.com*). Logging into an account via our research devices failed to populate some device logs, discussed in Section 6.

Audit logs (Figure 2-F). An audit log (also called "audit trail", e.g., by *bit.ly*) is a chronological record of events, actions, and changes to an account. Unlike security logs and activity logs, which offer a focused view of events (e.g, payment history, security activity) for the user, audit logs record both security and non-security activity for multiple users. Unlike device

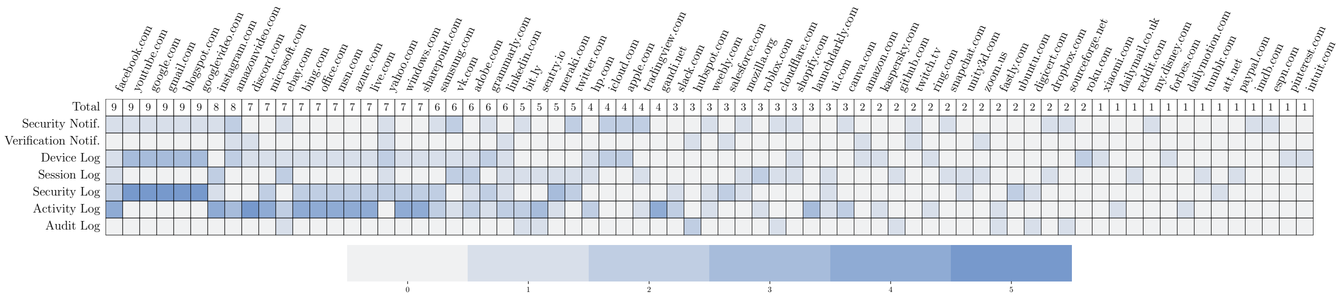


Figure 4: Heatmap with columns for each of the 71 services with at least one ASI and rows for the total number of distinct ASIs discovered for that service broken down by ASI type. For example, *facebook.com* has one device log, one session log, two distinct security logs (on separate pop-up pages with different headings), one security notification, and four activity logs (on separate pages), whereas *intuit.com* has just one device log.

logs, audit logs do not show a history of device accesses. We found ten audit logs across nine services; *hubspot.com* deployed two different audit logs.

Audit logs are often technical in nature, displaying information suitable for developers, researchers, or other "super" users of a service. Some examples include showing activity related to a developer project (e.g., repository on *github.com*), service subscriptions (e.g., trial on *sentry.io*), and generated assets (e.g., API token on *fastly.com*). The specific attributes on audit logs were not easily interpretable; unique record numbers, activity identification numbers, authentication log-in tokens, and more, were common, which might reasonably confuse a non-expert user. Further, they were typically found in non-security settings pages.

Since audit logs are often information rich, they were also lengthy, sometimes requiring several pages to display the data from our stepthroughs, as in the case of *github.com*. Thus, all audit logs in our dataset provided a search function to find or filter entries by a specific user or action. Users can also download the data displayed on audit logs, indicating that services anticipate the use of external analysis tools.

Session logs (Figure 2-G). We define a session as the period of time between when a user logs into a service and when they log out or are logged out by service policy. Session logs only report a history of the sessions started on an account. Daffalla et al. referred to these as *session lists* [14], but we use logs since they provide a trace of sessions on the account.

One device can start multiple sessions on a service, so a session log may show multiple entries from the same device organized by session, distinguishing it from device logs which organize accesses by device. Unlike security logs, session logs will not report any other security activity, and must report the most recently started session.

We found 31 session logs across 26 services, with several services deploying more than one. Session logs come in three variations: active, mixed, and unlabeled. Active session logs display sessions where the user is currently signed in on an account, but do not differentiate between dormant logged-in sessions and actively in-use sessions (e.g., *'Where You're Logged In'* from Figure 2-G). Mixed session logs delineate between active sessions and inactive sessions. Finally, unlabeled session logs usually show all sessions in descending order from 'most recent', without distinguishing whether they are active or inactive. Sessions were often displayed only if initiated within a specific time window (28–60 days).

A few active and mixed session logs differentiated the current session from others. For example a session log on *snapchat.com* displayed a 'current session' tag on the appropriate entry. Some active session logs provided real-time indicators, such as the IP of the current session, which allows users to quickly identify the current one (e.g., *meraki.com*).

Many services conflated sessions and devices. *twitter.com*[4] even states this incorrect conflation in its description, noting: *'Sessions are the devices you are using or that have used your X account.'* Similarly, *yahoo.com* titles its session log: "Review your connected devices and apps," although the entries corresponded to individual sessions. This conflation may confuse a user who may misinterpret each entry as a device. An alternativet design is used by *google.com*, which displays the sessions associated with each type of device, and clearly states: *'There might be multiple activity sessions from the same device.'* However, since this log was organized by device, we labeled it a device log instead of a session log.

Some services showed sessions in more than one place, duplicating information. However, none of them acknowledged the overlap between logs, which may confuse users. For example, *github.com* showed sessions in both its session log and audit log. Similarly, *meraki.com* logged sessions in one active session log and two security logs.

meraki.com's session log communicated (via a time limit of 30 days) when a session would automatically expire and thus be removed from a session log. However, a corresponding security log on the same page did not distinguish which logins belonged to expired, user-ended, or active sessions, making

[4] *twitter.com* redirects to *X.com*; however, we retain the original domain name, twitter.com, since that is what is included in the TRANCO list.

these ASIs inconsistent. Additional inconsistencies in ASI deployments are described in Section 5.

Session logs sometimes included account remediation advice. Typically, they recommended that users should either sign out of one or multiple sessions or change their password. A few services, like *reddit.com*, provided a force logout function (including the current session) that required an additional security challenge (e.g., re-entering one's password).

5 Patterns of ASI Deployments

We now highlight how individual services deploy (often multiple) ASIs for users. As shown in Figure 4, the number and types of ASIs discovered across these 71 services vary greatly. At one end of the spectrum is, for example, *facebook.com*, for which we discovered 9 distinct ASIs. At the other end are 13 services that each have a single ASI (the rightmost columns in Figure 4). These 13 varied in what ASI type was deployed. Most common was having a device log (4 services), followed by session log (3 services), email notifications (3 services), activity log (2 services), and security log (1 service).

An important aspect of a service's ASIs is if the service alerts users about access via notifications, or expects users to proactively investigate ASIs. In total, 37 services sent some form of notification, of which four *only* provided off-service email notifications, with no accompanying logs. Thus, when a user receives a notification, they have no on-service ASIs to cross-reference for verifying suspected account compromise. Only providing off-service notifications can cause other issues: notifications may be inaccessible if the email is not monitored, or even suppressed by an adversary with access to the email account, as seen in interpersonal abuse contexts [24]. Of the four notification-only services, *twitch.tv*, *disney.com*, and *espn.com* sent email notifications on each login, with *twitch.tv* sending both a verification notification followed by a security notification. *imdb.com* only sent one notification despite our stepthrough generating multiple logins.

Although our experimental methodology may not have triggered all service notifications, our results show that 34 of the 71 services with ASIs did not provide any notifications whatsoever. For these, a user must proactively seek out one of the on-service ASIs to assess compromise status. If a user rarely uses the account, they may never realize that their account was compromised. Moreover, the adversary may lock the user out of their account, rendering these ASIs and any evidence of illicit access inaccessible (again, an important issue in interpersonal abuse settings [14]).

A better approach is providing information via multiple channels. We found that 33 services deployed at least one on-service log and one off-service notification. Often, this was a security notification and one or more device, session, or security logs. Others were more diverse, such as *ebay.com*, which deployed an email security notification, two session logs, one device log, two activity logs, and one audit log.

Many services with multiple ASIs displayed inconsistent information across these ASIs. Recall that our stepthroughs generated two active and three inactive sessions. Then, *twitter.com*'s ASIs demonstrated inconsistency , deploying a session log showing three active sessions (incorrect), and security logs named 'Account access history' and 'Logged-in devices and apps' remaining empty (incorrect). This was confusing for the authors and may be confusing for users.

Multi-service account management. Several large companies used unified ASIs to merge information from multiple affiliated services. For example, eight services (*azure.com*, *outlook.com*, *office.com*, *microsoft.com*, *windows.com*, *sharepoint.com*, *bing.com*, *msn.com*) are operated by Microsoft. Users login to any of these sites using the same credentials (via *login.live.com* or *login.microsoftonline.com*). To obtain access information, all affiliated sites redirected to ASIs hosted on *account.microsoft.com* and *account.live.com*,[5] where the available ASIs included activity logs and a security log entitled "Recent Activity." Google and Apple similarly utilized unified account management portals accessible from multiple affiliated sites via a single account.

Unified account management portals can coalesce access data in a single "one-stop shop" for users. However, we found that none of the logs on unified portals explicitly differentiated accesses made to different affiliated services. For example, a login to *outlook.com* looked no different from a login to *office.com* on the "Recent Activity" security log (*account.live.com/activity*). While this may make sense from an architectural perspective as the company utilizes a single authentication backend, it may be confusing for users who may not realize the connection between services or when there are inconsistencies. On Microsoft, we observed that logging into one of its eight affiliated services did not result in a record in the sole security log, a flaw that we return to in Section 6.

Some companies deployed hybrids between independent and unified ASIs. The two Meta domains *instagram.com* and *facebook.com* had distinct ASIs on each site. However, one set of ASIs on *accountscenter.facebook.com*, labeled under the "Meta accounts center", used the same UI design and was identically labeled to the ASI on *accountscenter.instagram.com*. In our stepthroughs, which used distinct email addresses for the two sites, the "Meta accounts center" on both services only reflected sessions on that particular service. However, we later realized that if one uses the same email account for both services, the independent portals are automatically unified, and the "Meta accounts center" on each site shows sessions from both services, regardless of which service was accessed. This portal also allows users to manually link the two services' ASIs, should they have used separate email addresses when setting up their accounts. These ASIs were also inconsistent. For example, *facebook.com*'s security log "Logins and Lo-

[5] In Figure 4 these columns refer to the same ASIs. When we discuss the number of unique ASIs found in our study, however, we did not double count these or other similar unified portals.

gouts" was empty in our stepthroughs (incorrect), while the "Meta accounts center" session log "Account login activity" included all five sessions (correct).

Other services partially integrated their ASIs. For example, *amazon.com* and *amazonvideo.com*, mostly offered distinct ASIs, with one exception: a user can reach a device log on *amazon.com* through *amazonvideo.com*, but the six ASIs on the latter cannot be reached from *amazon.com*. The email verification and security notifications sent by both services appeared to be unified, using the same design, with no way to differentiate the originating service.

As a final example, *weebly.com* provided a security log, "Login History", which we found to contain duplicate entries for each login (ten login entries, rather than the correct five). Moreover, their email security notification describes new logins as being from *square.com* instead of *weebly.com* (which was not included in our study, but owns Weebly).

How users get to ASIs. We observed diversity in how users must navigate to on-service logs, finding that an average of 3.34 (SD ±1.4) user clicks were required to reach one from the service landing page. The most interactions needed was six (on *instagram.com*, *facebook.com*, and *google.com*).

Reaching different on-service logs required researchers to navigate through settings or account management pages, configuration pages, and independent pages for the ASI itself. For example, we conjecture that users may be confused by navigation paths for *twitter.com*'s ASIs , because the service's security-relevant information was accessed through "Apps and Sessions" rather than via "Security" (the latter led to configuration tools such as password management and 2FA).

A simpler example is the session log on *roblox.com* labeled "Where you're logged in" which was reachable by:

"Settings Icon" → "Settings" → "Security" → □

where we use → to denote a user clicking on a button or link, and □ to denote arriving at the page containing the ASI. Similarly, *discord.com*'s device log was accessed via

"Settings Icon" → "Devices" → □ .

By contrast, the most complex path was on *instagram.com*:

"Profile" → "Settings Icon"
→ "Settings and Privacy"
→ "See more in accounts center"
→ "Password and Security"
→ "Where You're Logged In"
→ "Review Devices" → □ .

The last element was a subtle, text hyperlink (almost missed during ASI discovery) that led to an "Unrecognized Logins" ASI on the "Meta Accounts Center" portal.

ASI navigation can also differ greatly within a service, which may be confusing for users. For instance, *ebay.com* had an "Activity Log" accessed via:

"Profile Dropdown" ↻ "Account Settings"
→ "Account Preferences"
→ "Permissions"
→ "Activity log" → □

where ↻ indicates a user hovering to reveal a dropdown menu. By contrast, users find a "Purchases" activity log via

"Profile Dropdown" ↻ "Account Settings"
→ "Activity"
→ "Purchases" → □ .

6 ASI Utility and Spoofability

We now analyze the utility and integrity of ASIs: do they provide sufficient, accurate, and reliable information to identify whether illicit access has occurred? We begin by assessing if ASIs accurately report the five accesses from our stepthroughs. Then, we analyze the information attributes within ASIs, evaluating their utility for identifying illicit access. Finally, we examine how secure these attributes are to being spoofed or obfuscated by an adversary.

Log population policies. Services often failed to clarify how their ASIs were populated. Some, like *forbes.com*'s "Your Devices" device log provided no details on how entries were added to the ASI. Others, like *adobe.com*'s "Active Sessions" session log, only showed active sessions on the account, meaning an adversary could avoid populating the ASI by simply logging out (referred to as "access hiding" by prior work [14]).

Account access reporting. Many ASIs remained unpopulated after our experiments. While we expected this for activity logs and audit logs (since our stepthroughs did not involve any account interaction beyond logging in), some device logs also remained unpopulated. In two cases, *amazon.com* and *xiaomi.com*, no explanation was given to the user about what kinds of devices populate these logs.

Some services *under-reported accesses* on their ASIs, meaning adversarial sessions may go unreported, enabling access hiding. For example, of the 37 services that deployed notifications, 30 did not send one on every account access. Logs also underreported logins. For instance, Microsoft's "Recent Activity" log only reported access when the account management portal was accessed, but not when any of its eight affiliated services were accessed. Similarly, *unity.com*'s mixed session log only showed one entry when there should have been five. Other services *over-reported account accesses*. For example, *slack.com*'s security log showed 20 total logins rather than five. We suspect that these issues arise from implementation errors and unique service policies around logging.

ASI informativeness. To reason about the utility of ASIs, we started by selecting a subset of information attributes discovered in our measurements. We included attributes based on our extensive experience helping survivors of technology-facilitated abuse assess whether their accounts have been compromised using ASIs. This experience stems from a subset of the authors who have volunteered for years in a clinic similar to the one described in [14]. The resulting *informativeness scale* groups information attributes for entries in an ASI (see Figure 6 in the Appendix) into six levels:

(1) no access attributes;
(2) date, time;
(3) location at any granularity;
(4) OS or browser type;
(5) OS version, browser version, device model, or complete User Agent string; and
(6) static device identifiers (e.g., serial numbers).

These six levels are ordered in increasing order of their value for helping users distinguish accesses from distinct devices (thus, level 1 = low value; level 6 = high value).

We rank temporal attributes lowest, as they indicate little about the originating device; instead they rely on users remembering temporal details of their access histories. While some users may know when they accessed their account and when they didn't, we believe ASIs should not rely on such user knowledge. Location can be useful in distinguishing accesses from different devices, but not if the accesses are within the same geographic region (a common case in interpersonal abuse settings), and users have previously demonstrated confusion with identifying devices based on locality [14]. Browser and OS type can indicate different kinds of devices and are useful for identifying a known perpetrator's device. Version information provides further granularity, and static device identifiers like serial numbers or IMEI can uniquely identify a device. Only the two Apple-related domains (*apple.com*, *icloud.com*) had ASIs with static identifiers.

We assess each ASI's informativeness as follows. If a log entry on the ASI contains an information attribute in level L, but no entry on that ASI contains any information attribute in a higher level, then that ASI has level L informativeness. For notifications, we treat each individual notification message as an entry. For example, if all notifications only show location, date, and time, then that ASI is level 3. Similarly, if a device log shows the version of the operating system within one entry, it is a level 5 ASI. An ASI is level one if the ASI remained unpopulated at the conclusion of our stepthroughs. A service is at level L if all its ASIs are at level L or less.

Level (L)	Max ASI Attributes	# Services	
		Benign	**Spoofed**
1	No Access Attributes	6	6
2	Date / Time	4	45
3	Location	4	1
4	OS / Browser	17	3
5	OS Ver / Browser Ver / Device Model / User Agent	38	14
6	Static Device Identifiers	2	2

Figure 5: The informativeness of service ASIs without spoofing (**Benign**) and with spoofing (**Spoofed**). A service is level one if, after our stepthroughs, all its ASIs showed no entries. A service is marked as being at level $L > 1$ if its most informative ASI includes at most level L attributes.

Figure 5 provides a breakdown of the informativeness levels achieved by the 71 services that have ASIs. As shown, several services had lower levels of informativeness: six had no access-relevant information whatsoever, despite having ASIs; four only showed temporal information; and four included location. For these 14 (19.7% of services with ASIs), a user cannot determine whether other devices have accessed their account if they log in from the same geographic region and/or the user does not recall the date/time of their accesses.

The 17 services that show OS and browser type are more helpful, but will be of limited utility if illicit accesses come from the same family of devices. Versions may be more useful for those cases, making the 38 services showing this more informative. As mentioned above, only Apple services are in a position to, and do, achieve level six informativeness.

Spoofing. Spoofing is an attack that attempts to downgrade the informativeness of an ASI, either by changing an attribute or, less frequently, forcing them to not be displayed (six ASIs across six services). We tested whether information attributes are spoofable by changing the device clock (temporal attributes), using a VPN (location attributes), and modifying the user agent string (OS, browser, and device model).

Serial numbers on *apple.com* and *icloud.com* cannot be spoofed by unprivileged client-side code. Temporal attributes were robust to spoofing with two exceptions: we spoofed time on *reddit.com*'s session log and a *vk.com* security notification. It is likely that these services use (untrustworthy) client-side timestamps, rather than a trustworthy server-side clock.

By contrast, we successfully spoofed location information on 70 services, the only outlier being a device log on *intuit.com*.[6] We spoofed all device information (type and version) on 38 services: 26 that show version numbers or full UA strings and 12 that do not. Ten services maintained accurate browser information on one or more ASIs despite spoofed UA strings, but in many cases this seemed to be because services list even legitimate Edge browser logins as Chrome.[7] *ebay.com*, maintained correct OS information on one ASI, and *twitch.tv* maintained correct OS and browser informa-

[6] We conjecture that *intuit.com* is using a known privacy leak in WebRTC [48] to obtain the true local IP address and using that for geolocation.

[7] Both Edge and Chrome are Chromium-based browsers, and services incorrectly identify Edge as Chrome even when no spoofing is involved.

tion on one ASI. Five services (*google.com*, *googlevideo.com*, *blogspot.com*, *gmail.com*, *youtube.com*) maintained correct device model and browser information on two unified ASIs; these services may be using more advanced fingerprinting techniques that don't rely on the user-agent [1].

Based on these results, we measured how spoofing can downgrade a service's informativeness. We let the spoofed informativeness level L' of an ASI be the maximum level of attributes appearing on the ASI that are *not* modifiable or suppressible by spoofing. Then a service has spoofed informativeness level L' if its ASI with highest spoofed informativeness level is L'. This approach is lenient on the effect of spoofing on informativeness. In practice, if an ASI shows both spoofable and unspoofable information, its informativeness may decrease even more than we suggest.

We present a breakdown of the spoofed informativeness levels of all 71 services in Figure 5. Many services are successfully downgraded, meaning that users can not trust any attributes beyond the spoofed informativeness level. As before, six services provided no reliable information at level 1—the two services for which temporal attributes could be spoofed either had other ASIs that provided correct time, or more informative, unspoofable attributes (e.g., browser information). All four level 3 services had spoofable location (*intuit.com* was originally level 4). The services that retained levels 4 or 5 in the face of spoofing are due to Chromium issues or conjectured fingerprinting (discussed earlier).

Most notably, spoofing resulted in 45 services—10.25x more than without spoofing—that can only provide reliable temporal information about accesses. This means that if a user cannot distinguish accesses based solely on date or time, the majority of services do not provide a way identify illicit access. It also highlights how widespread vulnerability to spoofing is: 42 of 71 services could be downgraded at least one level via spoofing. Only five services hosted ASIs that acknowledged the unreliability of information attributes: security notifications on *twitter.com*, *disney.com*, and *espn.com*, and session logs on *reddit.com* and *linkedin.com* all cautioned the instability of geolocation. For all other services, users have no way of knowing whether the information attributes on their ASIs were reliable.

Mitigating account compromise. If illicit access is identified, users may need to take immediate action to secure their account. Services in our study approached this in four ways. Many provided little to no remediation guidance on their ASIs (e.g., "Recent account activity" log in Figure 9-B in the Appendix). Others suggested that users change or reset their passwords, providing a shortcut or link. Third, some services presented links to external help guides or to forms where users can report suspicious activity. Finally, services with session logs and device logs often requested that users log out suspicious devices or sessions ("Active sessions" log in Figure 9-D in the Appendix).

7 Discussion

Our findings demonstrated widespread variation in ASI design and deployment. While imperfect ASIs retain some utility (i.e., for some technically proficient users), such inconsistencies nevertheless stem from diverging interpretations of what counts as account security and what is prioritized. Regardless of service perspectives, we highlight the need for best practices and standardization.

We fill this gap in two ways. First, design teams can use our typology to determine what ASI types to deploy. For instance, e-commerce services may include activity logs of purchases, while enterprise services may use audit logs. Second, we propose six principles for ASI developers to use in implementing high quality ASIs:

(1) *Complete logging:* Session and device logs should show active, inactive, and logged-out sessions and devices.
(2) *High utility:* An ASI should show device information that helps users identify illicit access.
(3) *Clear terminology:* An ASI should use clear, fixed, terminology when referring to account activity.
(4) *Visibility into ASI behavior:* An ASI should say why an entry is included, and which service it came from.
(5) *Effective guidance:* An ASI should help users identify illicit access and take appropriate action.
(6) *Easy access:* Services should have both on- and off-service ASIs, and they should be easy to find.

While some services achieved some principles, none fulfilled all six. We now examine each principle in detail and describe ideas for implementation.

P1: Complete logging. An ASI achieves complete logging if it reports all active, inactive, and signed-out sessions/devices. This mainly applies to session logs, device logs, and security logs. *meraki.com*'s security log titled "Your recent logins" is a good example of an ASI with complete logging, showing all sessions in reverse chronological order while allowing users to limit the number of entries shown.

However, many logs did not achieve this principle, either remaining unpopulated or underreporting accesses from our stepthroughs. These inconsistencies may stem from implementation bugs or risk-based approaches (c.f., [26, 37, 61, 65]) that predict noteworthy accesses to report.

Incomplete logging can enable access hiding attacks [14]. Thus, we recommend that services should implement complete logging for on-service ASIs showing access information. To avoid information overload [11, 12], we suggest allowing users to hide/show old entries, or auto-delete entries after a significant time period (e.g., 60 days). Services could also hide or "grey out" unpopulated logs to limit confusion.

While security notifications also display account access, more than 83% of services that sent notifications did not do so on every login, failing to achieve complete logging. An inviting fix may be to require services to send notifications on every access; however, research has shown that excessive notifications can lead users to ignore critical warnings [10].

P2: High utility. An ASI has high utility if it displays device information that helps users identify illicit accesses. Services in our study fell short of high utility in two ways. First, about 20% of services only reported temporal and/or location information on their ASIs, making it difficult to flag illicit activity from the same region and time range as the target. Second, most ASIs were vulnerable to spoofing. Moreover, no services fully described the reliability of all information shown on their ASIs. Without cautioning, users may wrongly assume that the information in an ASI is trustworthy.

Static identifiers, like serial numbers, are most informative and robust to spoofing. However, browser sandboxing blocks most web services from using them (the exceptions being *apple.com* and *icloud.com*), making implementing a true device log technically infeasible. Re-architecting the browser sandbox [14], such as through encrypted access logging [43], could improve access to static identifiers. However, this risks degrading user privacy and requires significant coordination between services, browser developers, and device vendors.

Therefore, we recommend that ASIs should show static identifiers when available, and more broadly, services should not implement device logs *unless* they can uniquely identify devices via static identifiers. Any other ASIs should display a best-effort approximation of OS and browser information, and note if any displayed identifiers are vulnerable to spoofing.

Exposing device identifiers on logs may also allow adversaries to monitor users. Hence, services may implement alerts or on-service banners that notify users when a log is accessed, with strategies to avoid notification fatigue (e.g, adding a delay to how quickly users can dismiss the alert) [10].

P3: Clear terminology. Terminology for describing account activity should be clear and consistent within and across services. Several ASIs failed to do this, often by conflating concepts such as "session" and "device" in their descriptions. Services also defined security terms differently. For example, *twitter.com*'s session log defined sessions as *"the devices you are using or that have used your X account"*. Describing sessions as devices may mislead users into thinking their account is being accessed from more devices than is true. *roblox.com*'s session log presents a better definition, describing sessions as *"places you're currently logged into your Roblox account"*.

Terminological inconsistency can misrepresent account activity and confuse users, so we suggest that ASIs should draw from a glossary of fixed, clear terminology when describing account activity. Our typology and the terms it defines are a starting point. However, future work is needed to investigate how to help users understand these terms (e.g, user studies).

P4: Visibility into ASI behavior. An ASI achieves high visibility if users can understand why an entry is or is not included and, if the ASI is a unified ASI (e.g., *microsoft.com*'s security log), what service the entry came from. While part of this clarity comes from high utility and clear terminology, here, we focus on how well ASIs describe their behavior. Many ASIs failed to achieve high visibility. Some remained unpopulated without any explanation (e.g., *xiaomi.com*'s device log), while others overreported accesses (such as *slack.com*'s security log). Finally, ASIs in unified portals often failed to distinguish what service an entry came from.

Prior work has pointed out that visibility is important in ensuring users develop strong mental models of security and privacy [56, 64]. Guaranteeing high visibility into ASI behavior can help ensure that users can precisely identify illicit accesses without wondering what triggered each entry.

We make three recommendations for achieving high visibility. First, each ASI should have a text description of its behavior. For logs, this description should clearly explain what is shown by each entry, and what kinds of account activity it reports. Similarly, notifications should provide a clear description of why it was triggered, and what that could mean for their account. If a notification or log relies on risk-based methods, services should explain what caused a particular activity to be flagged. Finally, unified ASIs should describe what service was used to perform the reported account activity, and whether any affiliated services were impacted.

P5: Effective guidance. ASIs should guide users to identify illicit access and secure their accounts. In our study, some ASIs provided specific indicators of suspicious activity and clear actions to take. For example, *adobe.com*'s session log tells users: "if you see any unfamiliar devices or locations, click End Session". Others were vague, such as *yahoo.com*'s security log, which advised users to "[r]eview all your account security activities" with no indicators of what to look for, or provided no guidance at all (e.g., *unity.com*'s security log).

Our findings extend prior work showing a continued need for improved account remediation guidance on services [39, 42]. Hence, we recommend that ASIs should emphasize signals of illicit access. One approach is to use UI markers (e.g., an "alert" icon) to highlight suspicious entries in an ASI, along with an explanation of why it is flagged (e.g., *"This not the usual operating system you sign in from"*). Services may also allow users to dismiss this icon (e.g., by pressing a button saying *"Don't alert me of this in the future"*), to prevent false positives. Finally, ASIs should include direct links to flows for securing the account. One example is including buttons to account security wizards that walk users through account remediation.

P6: Easy access. Services should provide easy access to account security information, both on- and off-service. We found that many services failed to achieve easy access, either offering *only* on- or off-service ASIs or placing logs in widely varying navigation paths.

Consequently, we propose that future ASI deployments should be found in a consistent, standardized, navigation path across services. One service that does this well is *roblox.com*, as pointed out in Section 4. For consistency, a standard URI path for ASIs should be established, similar to how security.txt files are currently deployed [23] (e.g., make all logs accessible at *domain /.well-known /asis /*).

Additionally, drawing from our typology, we suggest that every service should deploy at least one off-service security notification and one on-service session log since these directly report account security information. Both deployments should satisfy the other highlighted principles.

8 Conclusion

We surveyed ASIs offered by 100 popular services. We discovered 200 ASIs across 71 services, with 29 services providing no account access information to users. We found that many services exhibited design flaws, lacked clarity on ASI functionality, and did not present sufficient device information when spoofed. We contribute an expanded ASI typology that could help services better standardize their ASI offerings. We disclosed our results to relevant organizations, finding that existing disclosure mechanisms are misaligned with holistic views of account security. Finally, we discuss next steps for improving and standardizing ASIs. We will make our dataset available upon request to enable future work.

Acknowledgements

This work was funded in part by NSF grant CNS-2452613 and the Google Cyber NYC program.

Ethical Considerations

Our study did not involve interactions with real account owners or user data. As such it is not considered human subjects research and did not require IRB review. Nevertheless, we are cognizant of the ethical dimensions of our work and took precautions to mitigate such risks [59] during study design.

To protect the research team, we collected all data via newly created research test accounts with non-identifiable information (e.g., test email, test password, etc.) to avoid revealing personal information. We also excluded services that requested sensitive identifiers (e.g., government identification, payment information, health records) for signup.

In consideration of service providers and disclosure vendors, we chose methods that relied on standard user actions performed through test accounts, which complied with each service's terms and conditions. We found several flaws and potential bugs with service deployments, which we disclosed to services upon completion of our experiments and prior to publication (see Appendix A.1 for details). We ensured that our communications were transparent and promptly answered any follow up questions until cases were closed.

Finally, we acknowledge that an adversarial user could leverage our insights into inconsistent access reporting, spoofing, and account compromise to further surveil users. To mitigate this risk, we withhold the low-level details as to how to replicate these attacks, and reported these vulnerabilities to the relevant vendors (Section 7).

Open Science

We offer detailed, separate artifacts describing the results of our research here: `https://doi.org/10.5281/zenodo.18223588`. First, we release the recording extension that was used for screen captures during our stepthroughs Section 3. The *Screenshots* folder captures the states of all ASIs during stepthroughs. We also release the characterization of ASIs as a dataset (*ASI-Measurement-Dataset.csv*).

References

[1] Gunes Acar, Christian Eubank, Steven Englehardt, Marc Juarez, Arvind Narayanan, and Claudia Diaz. The web never forgets: Persistent tracking mechanisms in the wild. In *ACM conference on computer & communications security*, pages 674–689, 2014.

[2] Devdatta Akhawe, Johanna Amann, Matthias Vallentin, and Robin Sommer. Here's my cert, so trust me, maybe? understanding tls errors on the web. In *International conference on World Wide Web*, pages 59–70, 2013.

[3] Devdatta Akhawe and Adrienne Porter Felt. Alice in warningland: a {Large-Scale} field study of browser security warning effectiveness. In *USENIX Security Symposium*, pages 257–272, 2013.

[4] Julio Angulo and Martin Ortlieb. "wth..!?!" experiences, reactions, and expectations related to online privacy panic situations. 2015.

[5] Apple. If you think your apple account has been compromised, 2024. Accessed: 2024-09-30.

[6] Awio Web Services LLC. Browser & platform market share: November 2025. `https://www.w3counter.com/globalstats.php?year=2025&month=11`, November 2025. Accessed: 2025-12-08.

[7] Rosanna Bellini, Emily Tseng, Noel Warford, Alaa Daffalla, Tara Matthews, Sunny Consolvo, Jill Palzkill Woelfer, Patrick Gage Kelley, Michelle L Mazurek, Dana Cuomo, et al. Sok: Safer digital-safety research involving at-risk users. In *IEEE Symposium on Security and Privacy*, 2024.

[8] Joseph Bonneau and Sören Preibusch. The password thicket: Technical and market failures in human authentication on the web. In *WEIS*, 2010.

[9] Virginia Braun and Victoria Clarke. *Thematic analysis.* American Psychological Association, 2012.

[10] Cristian Bravo-Lillo, Lorrie Cranor, Saranga Komanduri, Stuart Schechter, and Manya Sleeper. Harder to ignore? revisiting {Pop-Up} fatigue and approaches to prevent it. In *Symposium on Usable Privacy and Security*, 2014.

[11] Cristian Bravo-Lillo, Lorrie Faith Cranor, Julie Downs, and Saranga Komanduri. Bridging the gap in computer security warnings: A mental model approach. *IEEE Security & Privacy*, 9(2):18–26, 2010.

[12] Carolina Carreira, Alexandra Mendes, João F Ferreira, and Nicolas Christin. A systematic review of security communication strategies: Guidelines and open challenges. *arXiv preprint arXiv:2504.02109*, 2025.

[13] Ting-Han Chen, Carlotta Tagliaro, Martina Lindorfer, Kevin Borgolte, and Jeroen Van Der Ham-De Vos. Are you sure you want to do coordinated vulnerability disclosure? In *IEEE European Symposium on Security and Privacy Workshops*, pages 307–314, 2024.

[14] Alaa Daffalla, Marina Bohuk, Nicola Dell, Rosanna Bellini, and Thomas Ristenpart. Account security interfaces: important, unintuitive, and untrustworthy. In *USENIX Security Symposium*, pages 3601–3618, 2023.

[15] Alaa Daffalla, Lucy Simko, Tadayoshi Kohno, and Alexandru G Bardas. Defensive technology use by political activists during the sudanese revolution. In *IEEE symposium on security and privacy*, 2021.

[16] Emiliano De Cristofaro, Honglu Du, Julien Freudiger, and Greg Norcie. A comparative usability study of two-factor authentication. *arXiv:1309.5344*, 2013.

[17] Martin Degeling, Christine Utz, Christopher Lentzsch, Henry Hosseini, Florian Schaub, and Thorsten Holz. We value your privacy... now take some cookies: Measuring the GDPR's impact on web privacy. *arXiv:1808.05096*, 2018.

[18] Rachna Dhamija and J Doug Tygar. The battle against phishing: Dynamic security skins. In *Symposium on Usable privacy and security*, pages 77–88, 2005.

[19] Kostas Drakonakis, Sotiris Ioannidis, and Jason Polakis. The cookie hunter: Automated black-box auditing for web authentication and authorization flaws. In *ACM Conference on Computer & Communications Security*, pages 1953–1970, 2020.

[20] Serge Egelman, Lorrie Faith Cranor, and Jason Hong. You've been warned: an empirical study of the effectiveness of web browser phishing warnings. In *ACM Conference on human factors in computing systems*, pages 1065–1074, 2008.

[21] Sandro Etalle, Fabio Massacci, and Artsiom Yautsiukhin. The meaning of logs. In *International Conference on Trust, Privacy and Security in Digital Business*, pages 145–154. Springer, 2007.

[22] M. E. Fagan. Design and code inspections to reduce errors in program development. *IBM Systems Journal*, 15(3):182–211, 1976.

[23] Edwin Foudil and Yakov Shafranovich. A File Format to Aid in Security Vulnerability Disclosure. RFC 9116, RFC Editor, 2022.

[24] Diana Freed, Jackeline Palmer, Diana Minchala, Karen Levy, Thomas Ristenpart, and Nicola Dell. "A Stalker's Paradise" How Intimate Partner Abusers Exploit Technology. In *ACM Conference on human factors in computing systems*, pages 1–13, 2018.

[25] Diana Freed, Jackeline Palmer, Diana Elizabeth Minchala, Karen Levy, Thomas Ristenpart, and Nicola Dell. Digital technologies and intimate partner violence: A qualitative analysis with multiple stakeholders. *Proceedings of the ACM on human-computer interaction*, 1(CSCW):1–22, 2017.

[26] Anthony Gavazzi, Ryan Williams, Engin Kirda, Long Lu, Andre King, Andy Davis, and Tim Leek. A study of {Multi-Factor} and {Risk-Based} authentication availability. In *USENIX Security Symposium*, 2023.

[27] Sanam Ghorbani Lyastani, Sven Bugiel, and Michael Backes. A systematic study of the consistency of two-factor authentication user journeys on top-ranked websites. 2023.

[28] Maximilian Golla, Grant Ho, Marika Lohmus, Monica Pulluri, and Elissa M Redmiles. Driving {2FA} adoption at scale: Optimizing {Two-Factor} authentication notification design patterns. In *USENIX Security Symposium*, pages 109–126, 2021.

[29] Maximilian Golla, Miranda Wei, Juliette Hainline, Lydia Filipe, Markus Dürmuth, Elissa Redmiles, and Blase Ur. " What was that site doing with my Facebook password?" Designing Password-Reuse Notifications. In *CCS*, 2018.

[30] Google. Secure a hacked or compromised google account, 2024. Accessed: 2024-09-30.

[31] Daniel V Klein. Foiling the cracker: A survey of, and improvements to, password security. In *Proceedings of the 2nd USENIX Security Workshop*, pages 5–14, 1990.

[32] Klaus Krippendorff. Computing krippendorff's alpha-reliability, 2011.

[33] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. Tranco: A research-oriented top sites ranking hardened against manipulation. In *Network and Distributed System Security Symposium*, 2019.

[34] Kevin Lee, Sten Sjöberg, and Arvind Narayanan. Password policies of most top websites fail to follow best practices. In *Symp. on Usable Privacy & Security*, 2022.

[35] Clayton Lewis, Peter G. Polson, Cathleen Wharton, and John Rieman. Testing a walkthrough methodology for theory-based design of walk-up-and-use interfaces. In *ACM Conference on Human Factors in Computing Systems*, pages 235–242, 1990.

[36] Ben Light, Jean Burgess, and Stefanie Duguay. The walkthrough method: An approach to the study of apps. *New Media & Society*, 20(3):881–900, 2018.

[37] Xu Lin, Panagiotis Ilia, Saumya Solanki, and Jason Polakis. Phish in sheep's clothing: Exploring the authentication pitfalls of browser fingerprinting. In *USENIX Security Symposium*, pages 1651–1668, 2022.

[38] Sanam Ghorbani Lyastani, Michael Schilling, Michaela Neumayr, Michael Backes, and Sven Bugiel. Is fido2 the kingslayer of user authentication? a comparative usability study of fido2 passwordless authentication. In *IEEE Symposium on Security and Privacy*, 2020.

[39] Philipp Markert, Andrick Adhikari, and Sanchari Das. A transcontinental analysis of account remediation protocols of popular websites. *arXiv:2302.01401*, 2023.

[40] Philipp Markert, Leona Lassak, Maximilian Golla, and Markus Dürmuth. Understanding users' interaction with login notifications. In *ACM Conference on Human Factors in Computing Systems*, pages 1–17, 2024.

[41] Susan E McGregor, Polina Charters, Tobin Holliday, and Franziska Roesner. Investigating the computer security practices and needs of journalists. In *USENIX Security Symposium*, pages 399–414, 2015.

[42] Lorenzo Neil, Elijah Bouma-Sims, Evan Lafontaine, Yasemin Acar, and Bradley Reaves. Investigating web service account remediation advice. In *Symposium on Usable Privacy and Security*, pages 359–376, 2021.

[43] Carolina Ortega Pérez, Alla Daffalla, and Thomas Ristenpart. Encrypted access logging for online accounts: Device attributions without device tracking. In *USENIX Security*, 2025.

[44] Thanasis Petsas, Giorgos Tsirantonakis, Elias Athanasopoulos, and Sotiris Ioannidis. Two-factor authentication: is the world ready? quantifying 2fa adoption. In *European workshop on system security*, 2015.

[45] Nils Quermann, Marian Harbach, and Markus Dürmuth. The state of user authentication in the wild. *WAY*, 2018.

[46] Kristina Radivojevic, Nicholas Clark, Anna Klempay, and Paul Brenner. Defending novice user privacy: An evaluation of default web browser configurations. *Computers & Security*, 140:103784, 2024.

[47] Elissa M Redmiles. "Should I Worry?" A Cross-Cultural Examination of Account Security Incident Response. In *IEEE Security & Privacy*, 2019.

[48] Andreas Reiter and Alexander Marsalek. Webrtc: your privacy is at risk. In *Proceedings of the Symposium on Applied Computing*, pages 664–669, 2017.

[49] Joshua Reynolds, Nikita Samarin, Joseph Barnes, Taylor Judd, Joshua Mason, Michael Bailey, and Serge Egelman. Empirical measurement of systemic {2FA} usability. In *USENIX Security Symposium*, 2020.

[50] Xin Ruan, Zhenyu Wu, Haining Wang, and Sushil Jajodia. Profiling online social behaviors for compromised account detection. *IEEE transactions on information forensics and security*, 11(1):176–187, 2015.

[51] Sena Sahin, Burak Sahin, and Frank Li. Was this you? investigating the design considerations for suspicious login notifications. In *NDSS*, 2025.

[52] Johnny Saldana. *Thinking qualitatively: Methods of mind*. SAGE publications, 2014.

[53] Richard Shay, Iulia Ion, Robert W Reeder, and Sunny Consolvo. "My religious aunt asked why I was trying to sell her viagra" Experiences With Account Hijacking. In *ACM Conference on Human Factors in Computing Systems*, 2014.

[54] Steve Sheng, Bryant Magnien, Ponnurangam Kumaraguru, Alessandro Acquisti, Lorrie Faith Cranor, Jason Hong, and Elizabeth Nunge. Anti-phishing phil: the design and evaluation of a game that teaches people not to fall for phish. In *Symposium on Usable privacy and security*, pages 88–99, 2007.

[55] Lucy Simko, Ada Lerner, Samia Ibtasam, Franziska Roesner, and Tadayoshi Kohno. Computer security and privacy for refugees in the united states. In *IEEE symposium on security and privacy*, pages 409–423, 2018.

[56] Eric Spero and Robert Biddle. Out of sight, out of mind: Ui design and the inhibition of mental models of security. In *New Security Paradigms Workshop*, 2020.

[57] Emily Stapley, Sally O'Keeffe, and Nick Midgley. Developing typologies in qualitative research: The use of ideal-type analysis. *International Journal of Qualitative Methods*, 21:16094069221100633, 2022.

[58] Avinash Sudhodanan and Andrew Paverd. Pre-hijacked accounts: an empirical study of security failures in user account creation on the web. In *USENIX Security Symposium*, pages 1795–1812, 2022.

[59] The Menlo Report. Ethical principles guiding information & communication technology research. https://www.dhs.gov/sites/default/files/publications/CSD-MenloPrinciplesCORE-20120803_1.pdf.

[60] Kurt Thomas, Frank Li, Chris Grier, and Vern Paxson. Consequences of connectivity: Characterizing account hijacking on twitter. In *ACM Conference on Computer & Communications Security*, pages 489–500, 2014.

[61] Kurt Thomas, Frank Li, Ali Zand, Jacob Barrett, Juri Ranieri, Luca Invernizzi, Yarik Markov, Oxana Comanescu, Vijay Eranti, Angelika Moscicki, Daniel Margolis, Vern Paxson, and Elie Bursztein. Data breaches, phishing, or malware? understanding the risks of stolen credentials. In *ACM Conference on Computer and Communications Security (CCS)*, 2017.

[62] Kurt Thomas, Jennifer Pullman, Kevin Yeo, Ananth Raghunathan, Patrick Gage Kelley, Luca Invernizzi, Borbala Benko, Tadek Pietraszek, Sarvar Patel, Dan Boneh, et al. Protecting accounts from credential stuffing with password breach alerting. In *USENIX Security Symposium*, pages 1556–1571, 2019.

[63] Tranco List. A research-oriented top sites ranking hardened against manipulation - tranco. https://tranco-list.eu/#download.

[64] Ben Weinshel, Miranda Wei, Mainack Mondal, Euirim Choi, Shawn Shan, Claire Dolin, Michelle L Mazurek, and Blase Ur. Oh, the places you've been! user reactions to longitudinal transparency about third-party web tracking and inferencing. In *ACM Conference on Computer & Communications Security*, pages 149–166, 2019.

[65] Stephan Wiefling, Luigi Lo Iacono, and Markus Dürmuth. Is this really you? an empirical study on risk-based authentication applied in the wild. In *34th IFIP TC 11 International Conference*, pages 134–148. Springer, 2019.

A Appendix

A.1 Disclosures to Services

We conducted a disclosure process to notify companies about our findings, which we report below.

Challenges. Disclosure is challenging even for traditional vulnerabilities (e.g., in triaging issues consistently) [13], and our findings are not traditional software vulnerabilities. For example, lacking off-service notifications and complex navigation paths are design and deployment issues, but can limit users' ability to detect compromise. Further, while prior work [14] disclosed similar issues, it was to only four services that had trust and safety expertise.

Approach. We categorized our findings into four groups:

(1) Missing functionality refers to insufficient/missing functionality in ASIs that make it harder to identify compromise. Examples include: not sending notifications, not showing all sessions on a session log, and missing key information.

(2) Bugs refer to unexpected ASI behavior that is not caused by adversarial use. This included over-reporting logins (e.g., weebly.com's security log), and failing to report service logins in unified portals (e.g. microsoft.com).

(3) Vulnerabilities allow adversarial manipulation of ASIs to trick targets or evade detection. Our focus is on spoofing.

(4) Design flaws are issues with how ASIs present information to the user, such as missing an ASI description, an ASI description that conflates multiple access concepts, or a lack of timezone information in entries.

We drafted 58 total disclosure briefings for services with ASIs, submitting one each for services run by Apple, Google, Microsoft, and Amazon. Each briefing included the study purpose, reproduction steps, an explanation of discovered issues, any risk assessments, a draft of this paper and relevant screen captures. We reported 232 total issues, with an average of four per briefing. Missing functionality and vulnerabilities were most common, appearing in 55 and 48 briefings, respectively.

For each organization, we performed a Google search with the name of the service and the keywords "vulnerability disclosure" or "bug bounty" and scanned for any relevant disclosure platforms or security contacts. We also checked for contacts in security.txt files by visiting the /.well-known/security.txt and /security.txt subdomains of each service's top level URL, finding 34 in total. If none were found (three services), we opted to send emails to the address: security@[service-url], which, in the authors' experiences, often lead to disclosure mailboxes.

Some organizations have their own disclosure platforms (seven organizations) or dedicated security team emails (12). Others (36) use third party platforms to triage reports: HackerOne (30), BugCrowd (5), and YesWeHack (1). YesWeHack required passport-based identity verification for users, preventing us from disclosing to that organization due to researcher privacy concerns. HackerOne rate limits disclosures, which slowed our disclosure process immensely. When facing restrictions, we still attempted to contact services through an email from a security.txt file if available. We performed disclosures between March 5th and August 12th, 2025.

None of the disclosure platforms (whether organization-specific or third-party) have reporting workflows that map well onto the types of issues we wanted to report. For example, disclosure forms often only presented options that focus on compromise itself, rather than the detection of it. We made a best effort in selecting the most appropriate options.

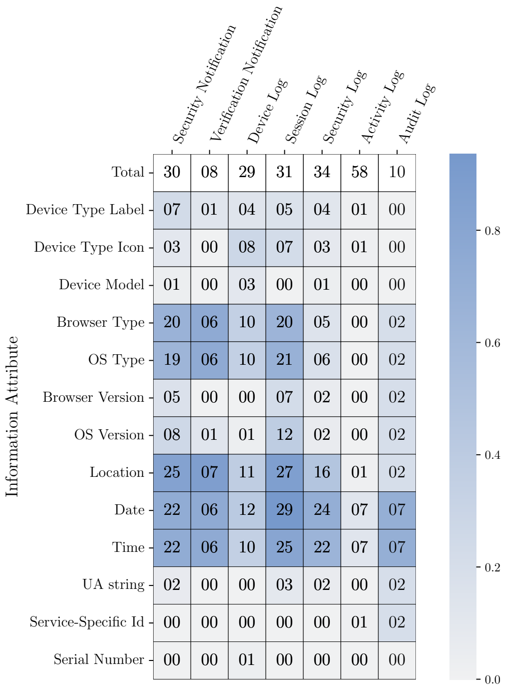
Disclosure responses. We got submission acknowledgement for 38 disclosures; 25 of which provided a response. Of the 25 with responses, we submitted three on organization-specific disclosure platforms, 16 via HackerOne or BugCrowd, and six via email. Most of our disclosures were classified as "Not applicable" or "Out of scope", reflecting a disconnect between our findings and disclosure program scopes.

Some responses indicated a misunderstanding of our study. For instance, snapchat.com's response asserts: *"attacks requiring physical access are strictly out of scope"* even though our stepthroughs tested a remote adversary. Others, such as *intuit.com* pointed to existing security functionality to argue that our results posed minimal risk: *"there does not appear to be any exploitable security implications... Intuit has additional protections in place, like MFA"* This misses our goal of improving compromise detection with ASIs.

Most responses rejected our disclosure because of the assumption of an adversary having account access. For example, reddit.com responded: *"[i]n this instance, the attacker must have access to the victim's account already and as such would not be accepted as a valid issue"*. Such responses suggest a misalignment between notions of what is security relevant.

Three responses correctly pointed out that user-agent and location spoofing is *"a known limitation of browser fingerprinting"* (shopify.com), concluding that it is not a security issue. This rightly stresses that future work is needed to rectify this (e.g., work by Ortega Pérez et al. [43]). Five organizations shared our results internally, e.g., grammarly.com stated: *"I am closing this report ... We are discussing this internally.*"

B Additional Figures

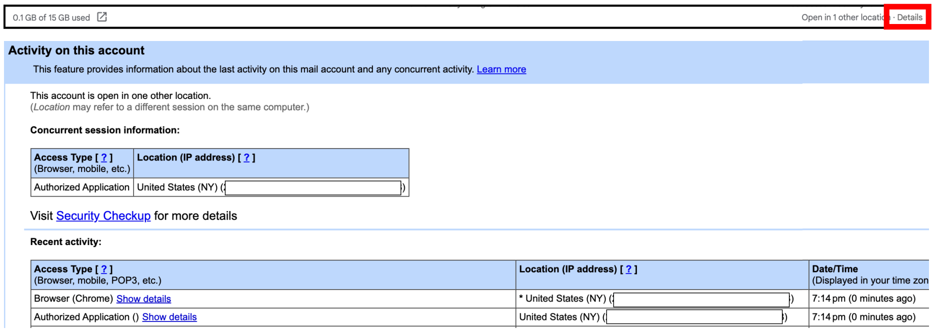


	Security Notification	Verification Notification	Device Log	Session Log	Security Log	Activity Log	Audit Log
Total	30	08	29	31	34	58	10
Device Type Label	07	01	04	05	04	01	00
Device Type Icon	03	00	08	07	03	01	00
Device Model	01	00	03	00	01	00	00
Browser Type	20	06	10	20	05	00	02
OS Type	19	06	10	21	06	00	02
Browser Version	05	00	00	07	02	00	02
OS Version	08	01	01	12	02	00	02
Location	25	07	11	27	16	01	02
Date	22	06	12	29	24	07	07
Time	22	06	10	25	22	07	07
UA string	02	00	00	03	02	00	02
Service-Specific Id	00	00	00	00	00	01	02
Serial Number	00	00	01	00	00	00	00

Figure 6: Heatmap showing the number and proportion of each ASI type that contains specific information attributes. The "total" row provides the total number of ASIs of each type. The remaining rows show how many ASIs of each type that contained that particular information attribute.

wikipedia.org
spotify.com
vimeo.com
wordpress.com
skype.com
tiktok.com
gravatar.com
w3c.org
nytimes.com
webex.com
cnn.com
theguardian.com
sciencedirect.com
bbc.co.uk
cisco.com
booking.com
issuu.com
tinyurl.com
ibm.com
soundcloud.com
reuters.com
avast.com
oracle.com
washingtonpost.com
linktr.ee
flickr.com
archive.org
epicgames.com
medium.com

Figure 7: The set of 29 services that have no ASIs, and thus do not provide users with any information about accesses.



0.1 GB of 15 GB used

Open in 1 other locatio · Details

Activity on this account

This feature provides information about the last activity on this mail account and any concurrent activity. Learn more

This account is open in one other location.
(*Location* may refer to a different session on the same computer.)

Concurrent session information:

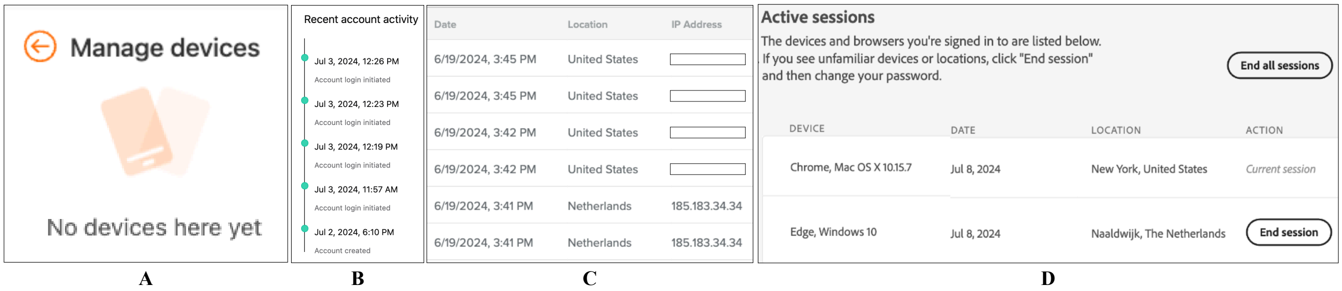
Access Type [?] (Browser, mobile, etc.)	**Location (IP address) [?]**
Authorized Application	United States (NY) ([redacted])

Visit Security Checkup for more details

Recent activity:

Access Type [?] (Browser, mobile, POP3, etc.)	**Location (IP address) [?]**	**Date/Time** (Displayed in your time zon
Browser (Chrome) Show details	* United States (NY) ([redacted])	7:14 pm (0 minutes ago)
Authorized Application () Show details	United States (NY) ([redacted])	7:14 pm (0 minutes ago)

Figure 8: A screenshot of a hard-to-find session log and security on Gmail, which is reachable by clicking on a small link titled "Details" at the bottom of the webpage (highlighted in red).



Manage devices

No devices here yet

Recent account activity

Jul 3, 2024, 12:26 PM
Account login initiated

Jul 3, 2024, 12:23 PM
Account login initiated

Jul 3, 2024, 12:19 PM
Account login initiated

Jul 3, 2024, 11:57 AM
Account login initiated

Jul 2, 2024, 6:10 PM
Account created

Date	Location	IP Address
6/19/2024, 3:45 PM	United States	[redacted]
6/19/2024, 3:45 PM	United States	[redacted]
6/19/2024, 3:42 PM	United States	[redacted]
6/19/2024, 3:42 PM	United States	[redacted]
6/19/2024, 3:41 PM	Netherlands	185.183.34.34
6/19/2024, 3:41 PM	Netherlands	185.183.34.34

Active sessions

The devices and browsers you're signed in to are listed below.
If you see unfamiliar devices or locations, click "End session"
and then change your password.

End all sessions

DEVICE	DATE	LOCATION	ACTION
Chrome, Mac OS X 10.15.7	Jul 8, 2024	New York, United States	*Current session*
Edge, Windows 10	Jul 8, 2024	Naaldwijk, The Netherlands	End session

Figure 9: Example ASIs encountered in our study, shown in increasing order of informativeness: **(A)** a device log from *xiaomi.com* that remains unpopulated, with no explanation as to why, even though our experiments logged into the service with two different devices; **(B)** a security log from *mozilla.org*, which only shows the date and time of login; **(C)** a security log from *weebly.com* labeled "Login History", which shows location and IP information, but omits details such as granular location data; **(D)** a session log from *adobe.com* which shows granular device information, but conflates sessions, devices, and browsers in its description.

Code	**Definition**	**Example**
ASI Functionality		
Allow/Deny verification request	UI element enabling completion of an authentication challenge.	"Verify Login" button on *discord.com* security notification.
Change password	UI element enabling a password change.	"Change your password" link on *ebay.com* session log.
Configure security alerts	UI element to adjust security notification delivery.	"Set up alerts" link on *instagram.com* session log.
Create/Delete token	UI element allowing token creation or deletion.	"Create new token" button on *sentry.io* activity log.
Expand ASI	UI element allowing expansion of log to show more entries.	"Hide sessions' button on *pinterest.com* device log.
Export data	UI element allowing data export from ASI.	"Download CSV" link on *cloudflare.com* audit log.
Filter ASI entries	UI interface allowing filtering of ASI entries.	"Category" dropdown on *hubspot.com* audit log.
Find device	UI element allowing users to try and locate a lost device.	"Find a lost device" button on *google.com* device log.
Log out (selected) session	UI element allowing signout of specific sessions.	"End session" button on *adobe.com* session log.
Log out all sessions	UI element that allows signout of all sessions on the account	"End all sessions" button on *adobe.com* session log.
Remove (selected) activity	UI element to remove specific entries from a log	"Delete" button on *facebook.com* activity log.
Remove all activity	UI element to clear a log	"Remove all" button on *dailymotion.com* activity log.
Remove devices	UI element to remove a device with account access	"Remove from account" button on *apple.com* device log.
Report sign in	UI element to report a suspicious sign in	"If this wasn't you, please let us know here" link on *zoom.us* verification notification.
Resend verification	UI element to resend a verification notification	"Resend verification" button on *sentry.io* activity log.
Review connected apps	UI element that redirects to another ASI to review apps	"Connected apps" link on *twitter.com* security log.
Search entries	UI interface allowing search of ASI entries.	"User" search box on *cloudflare.com* audit log.
Visit help/support	UI element that redirects to help/support pages	Support portal link on *cloudflare.com* security notification.
Visit on-service ASI	UI element that redirects to an on-service ASI	"Check activity" button on *google.com* security notification.
Information Attributes		
Date of activity	Text describing the day of an activity occurred.	"3 days ago" on *tumblr.com* session log.
Session entry	An entry/row in an ASI describing a session started on the account.	Single row in *adobe.com* session log.
Time of activity	Text on interface describing the time that activity occurred.	"13 hours ago" on *snapchat.com* session log.
Location	Text describing the location from which activity occurred.	"South Holland, The Netherlands" on *pinterest.com* device log.
Browser	Text describing the browser type/version of the device from which activity occurred.	"Chrome 125" on *tumblr.com* session log.
OS	Text describing the OS type/version of device from which activity occurred.	"Windows 10" on *tumblr.com* session log.
IP Address	Text describing the IP address of the device from which activity occurred.	"IP Address" in *meraki.com* security log.
Helpful label	Helpful text describing the activity that was logged.	"Account login initiated" on *mozilla.org* security log.
Device Model/Type	Text describing the model/type of the device from which activity occurred.	"Signed In" on *ui.com* security log.
Account/User Name	Text describing the account or username of account an activity was logged from.	"asitesta" on *sourceforge.com* audit log.
Email address	Text describing an email address.	"asitest24@gmail.com" on *meraki.com* security log.
Raw user agent string	Text displaying the raw user agent string of the device from which activity occurred.	"Use agent" column on *ubuntu.com* security log.
Authentication status	Text describing details of the authentication method used to complete a certain activity.	"TFA successful" on *unity.com* security log.
Service specific identifier	Text describing an identifier that is useful for service specific use cases or audits.	"tokenId" *fastly.net* audit log.
Static identifiers	Text describing a static identifier of the device from which activity occurred.	"Serial Number" on *apple.com* device log.
Session expiration	Text describing the max duration of a session or the date/time that a session will expire.	"Expires in" column in *meraki.com* session log.

Figure 10: The 19 codes for ASI functionality and 16 codes for ASI information attributes that were used to construct our typology, along with definitions of each code and examples from our studied services.

Domain	Total	Security Notification	Verification Notification	Device Log	Session Log	Security Log	Audit Log	Activity Log
att.net	1	0	0	0	0	1	0	0
dailymotion.com	1	0	0	0	0	0	0	1
espn.com	1	1	0	0	0	0	0	0
forbes.com	1	0	0	1	0	0	0	0
imdb.com	1	1	0	0	0	0	0	0
intuit.com	1	0	0	1	0	0	0	0
my.disney.com	1	1	0	0	0	0	0	0
paypal.com	1	0	0	0	1	0	0	0
pinterest.com	1	0	0	1	0	0	0	0
reddit.com	1	0	0	0	1	0	0	0
tumblr.com	1	0	0	0	1	0	0	0
xiaomi.com	1	0	0	1	0	0	0	0
dailymail.co.uk	1	0	0	0	0	0	0	1
amazon.com	2	0	1	1	0	0	0	0
digicert.com	2	0	0	0	0	1	1	0
dropbox.com	2	1	0	0	1	0	0	0
fastly.com	2	0	0	0	0	0	1	1
github.com	2	0	0	0	1	0	1	0
kaspersky.com	2	0	0	1	0	0	0	1
ring.com	2	0	0	1	0	0	0	1
roku.com	2	0	0	2	0	0	0	0
snapchat.com	2	1	0	0	1	0	0	0
sourceforge.net	2	1	0	0	0	0	1	0
twitch.tv	2	1	1	0	0	0	0	0
ubuntu.com	2	0	0	0	0	2	0	0
unity3d.com	2	0	0	0	1	1	0	0
zoom.us	2	0	1	0	1	0	0	0
canva.com	3	1	0	0	0	0	0	2
cloudflare.com	3	1	0	0	1	0	1	0
hubspot.com	3	0	1	0	0	0	2	0
launchdarkly.com	3	0	0	0	0	0	0	3
mozilla.org	3	1	0	0	1	1	0	0
roblox.com	3	0	0	0	2	0	0	1
salesforce.com	3	0	1	0	0	2	0	0
shopify.com	3	1	0	1	1	0	0	0
slack.com	3	0	0	0	0	1	0	2
ui.com	3	0	0	0	1	1	0	1
weebly.com	3	1	0	0	0	1	0	1
apple.com	4	2	0	2	0	0	0	0
gandi.net	4	0	0	0	0	0	0	4
hp.com	4	0	0	1	1	0	0	2
icloud.com	4	2	0	2	0	0	0	0
tradingview.com	4	2	0	0	1	0	0	1
bit.ly	5	1	0	0	1	0	1	2
meraki.com	5	0	0	0	1	3	0	1
sentry.io	5	0	0	0	1	0	1	3
twitter.com	5	2	0	0	1	2	0	0
adobe.com	6	0	0	1	2	1	0	2
grammarly.com	6	1	0	2	0	2	0	1
linkedin.com	6	0	1	1	1	1	0	2
samsung.com	6	1	0	1	0	2	0	2
vk.com	6	2	0	1	2	0	0	1
azure.com	7	0	0	1	0	2	0	4
bing.com	7	0	0	1	0	2	0	4
discord.com	7	0	1	1	0	0	0	5
ebay.com	7	1	0	1	2	0	1	2
microsoft.com	7	0	0	1	0	2	0	4
msn.com	7	0	0	1	0	2	0	4
office.com	7	0	0	1	0	2	0	4
live.com	7	0	0	1	0	2	0	4
sharepoint.com	7	0	0	1	0	2	0	4
windows.com	7	0	0	1	0	2	0	4
yahoo.com	7	1	1	2	1	2	0	0
amazonvideo.com	8	2	1	2	0	0	0	3
instagram.com	8	1	0	0	2	1	0	4
blogspot.com	9	1	0	3	0	5	0	0
facebook.com	9	1	0	1	1	2	0	4
gmail.com	9	1	0	3	0	5	0	0
google.com	9	1	0	3	0	5	0	0
googlevideo.com	9	1	0	3	0	5	0	0
youtube.com	9	1	0	3	0	5	0	0

Figure 11: The 71 services in our study with at least one ASI, showing the number and types of ASIs per service.