CASE STUDY 3

Browser-Based Location Disclosure: A Controlled Seeker Link Simulation Case Study

By: Joseph Gonzalez

ZERODAY TECH LABS

Scope	Controlled lab simulation using a researcher-owned phone posing as the victim device; educational and analytical purposes only
Environment	Kali Linux, Seeker, ngrok tunnel, mobile browser, researcher-owned Android phone, redacted public URL/IP/location artifacts
Primary Evidence	Seeker launch screenshot, ngrok tunnel screenshot, request listener screenshot, mobile location-permission screenshot, redacted terminal capture of device/location fields
Primary Attack Vector	Permission-based disclosure through a location-themed public link that persuades the user to approve browser location access

Educational and analytical use only. Public URL, IP, device, and location details shown in figures are redacted.

Introduction

Mobile browsers routinely ask for sensitive permissions during ordinary web activity, especially when a site claims it needs location to personalize the experience. That creates a different exposure from credential phishing or malicious app sideloading: the user does not need to install software or type a password. The risk appears at the permission prompt.

This case study examines a controlled Seeker simulation using a temporary public tunnel and a researcher-owned phone posing as the victim device. The goal is to document what the page looked like, how the browser presented the location request, what data appeared after permission was granted, and why this matters for individual users outside the lab.

Methodology

A controlled cybersecurity simulation was conducted in an isolated lab using Kali Linux, Seeker, an ngrok tunnel, and a mobile browser on a researcher-owned Android phone. The scope was limited to self-controlled infrastructure, a temporary test link, and consent-based interaction by the researcher. No real victims, third-party devices, production accounts, or unauthorized tracking were involved.

1. Overview

This scenario shows how a web page can request location access and record browser-provided device and network details once the user grants permission. The analytical question was whether a believable location-themed page, delivered through a temporary public tunnel, could produce observable location and device data from a phone that voluntarily accepted the browser prompt.

2. Threat Analysis: Vector, Technique, and Human Factor

Attack vector used	A public link to a location-themed web page that requests browser location access and records returned device, network, and location fields after approval.
Social engineering techniques	Permission phishing, pretexting, contextual deception, link-based lure, and origin inattention.
Human factor exploited	Mobile prompt fatigue, task fixation, trust in the page theme, focus on large approval buttons, and reduced attention to temporary or unfamiliar domains.
Defensive relevance	Location prompts should be treated as security decisions. Users should deny unfamiliar requests, verify the site origin, review saved permissions, and revoke access immediately after a suspicious approval.

Key reader takeaway: this is not a silent operating-system compromise. The browser did ask. The exposure happened because the page made the request feel normal enough for the user to approve.

3. Lab Setup and Public Link Delivery

The lab host was used to start the Seeker environment and select a location-themed template. Figure 1 shows the launch environment. Figure 2 shows the temporary tunnel used to make the local test page reachable from the phone. The important transition is from a local-only page to a public link that a mobile browser could open.

Figure 1. Seeker launch environment on the lab host, showing the template-selection stage used for the controlled simulation.

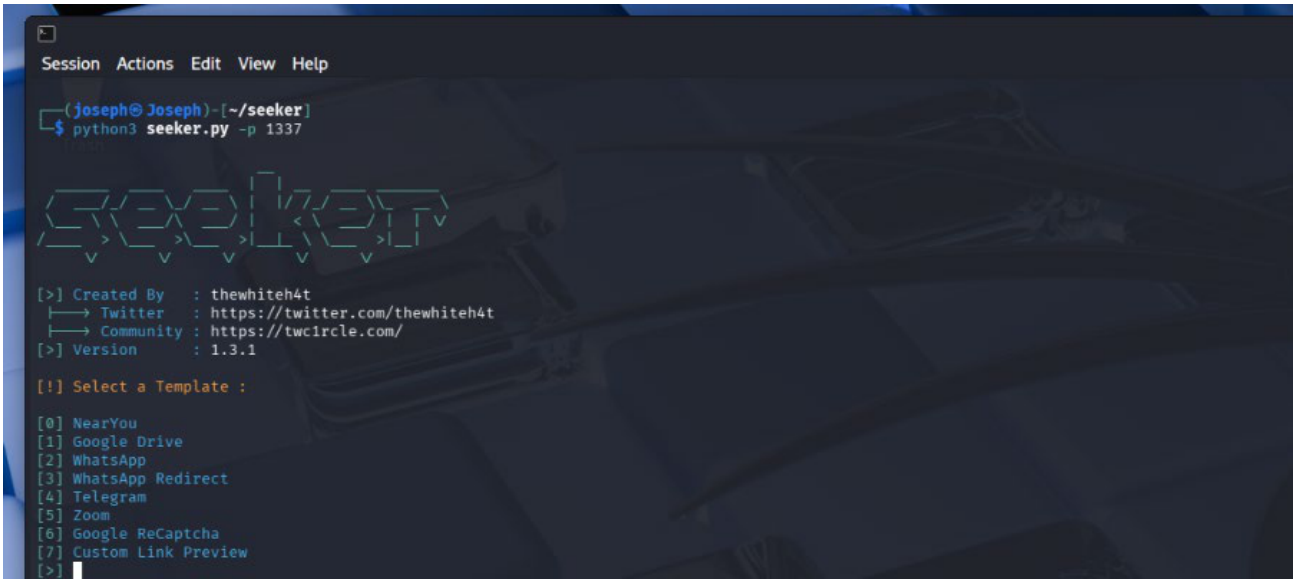

Figure 2. Temporary tunnel session used to expose the local test page through a public forwarding path, with URL and account details redacted.

4. Permission Prompt and Evidence Collected

After the public test link was prepared, the phone opened the location-themed page as the simulated victim device. Figure 3 shows request activity in the tunnel session after the page was visited. Figure 4 shows the mobile browser displaying a location permission request for the site. This is the key decision point: the exposure depends on the user granting access, not on an invisible bypass of the browser permission model.

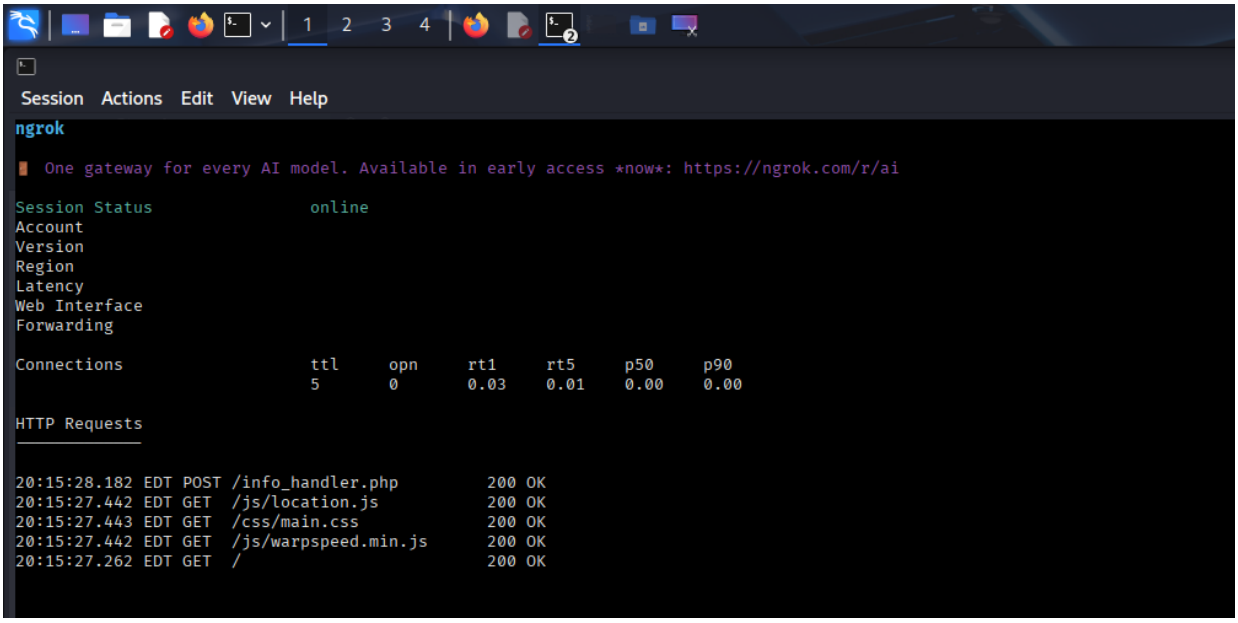

Figure 3. Tunnel listener view showing web requests from the phone after the public test link was opened, with forwarding details redacted.

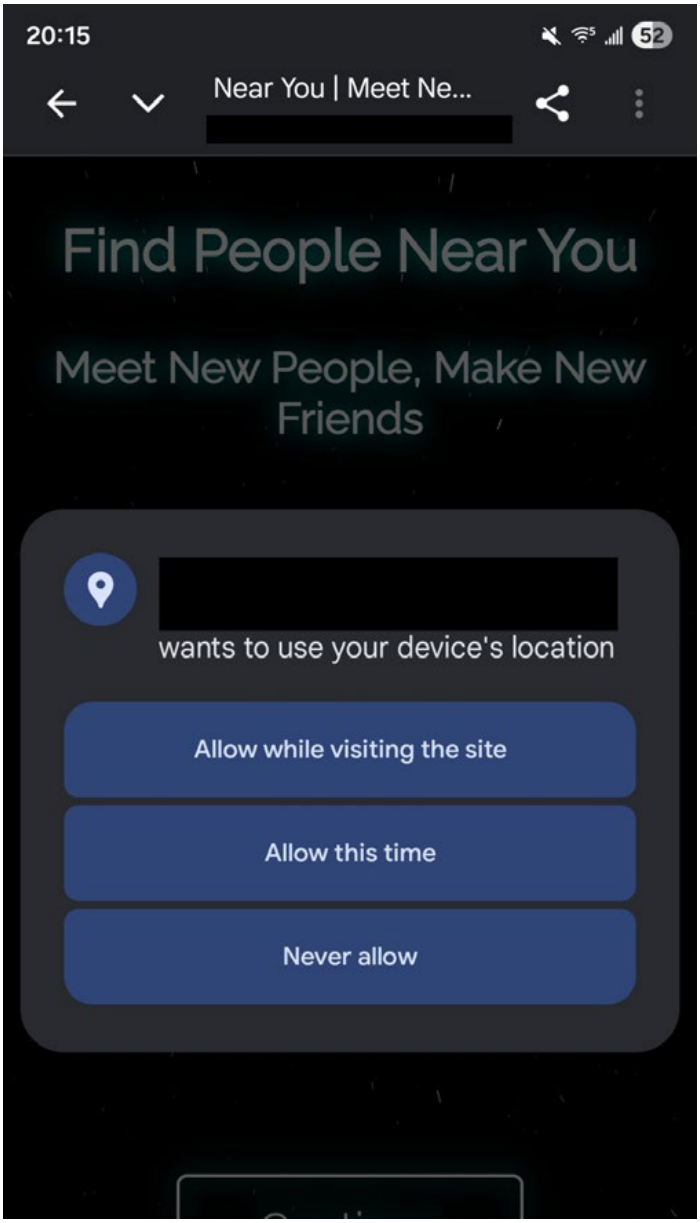

Figure 4. Researcher-owned phone posing as the victim device, opening the link and receiving a browser location-permission prompt with the public URL redacted.

Additional terminal evidence appears with the next artifact. Redactions remove private IP, URL, and location details while preserving the forensic value of the test.

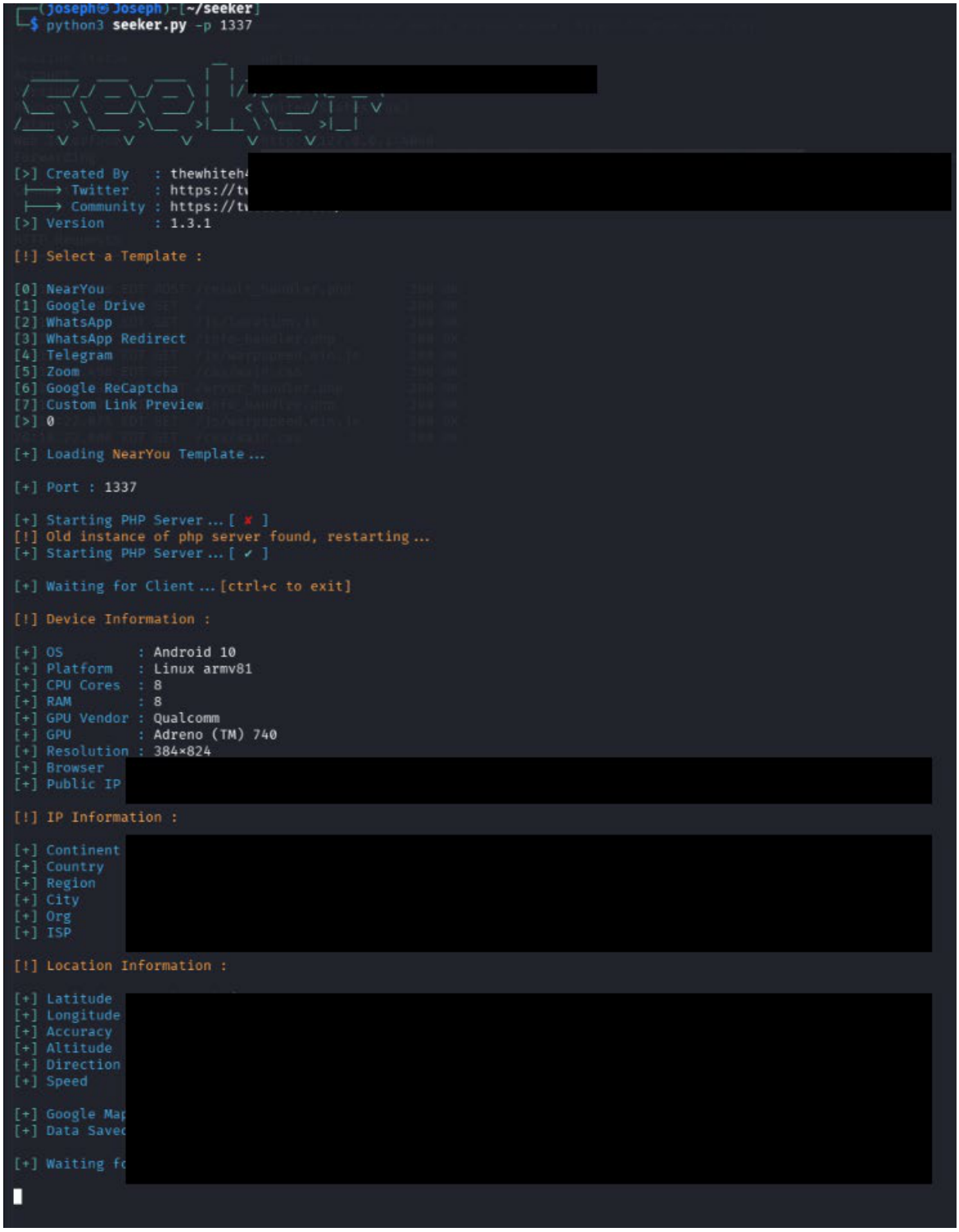

Figure 5. Redacted Seeker terminal capture showing browser-provided device, network, and location fields after permission was granted during the controlled test.

5. Why the Disclosure Occurred

The disclosure occurred because the site asked for a permission that browsers are designed to support, and the user granted it. The near-you theme made the request feel contextually appropriate: a page claiming to show nearby people naturally appears to need location access.

The browser did show the site origin, but users often focus on the task language and the large approval buttons rather than verifying the temporary domain. The tunnel also mattered because it turned a local service into a reachable public page,

allowing the phone to interact with the lab-hosted content from outside the localhost context.

6. Real-World Impact to the Individual

For an individual user, granting location to a deceptive page can expose more than a single data point. Location can reveal where someone lives, works, studies, worships, or spends time with family. When combined with public IP address, device type, browser version, and network provider information, it can support targeted phishing, harassment, doxxing, stalking, or highly personal follow-up lures.

The risk is serious on phones because people treat mobile permission prompts as routine. A user may approve location while distracted, then forget that the site ever asked. Even without malware, the collected data can help an attacker build a profile, time future messages, or craft a more believable follow-up.

7. Mitigation, Hardening, and Prevention

The practical fix is to reduce unnecessary location grants and treat permission prompts as security decisions. Users should deny location requests from unfamiliar links, avoid opening location-themed links from messages or social media, and check the browser address bar before approving access.

If a suspicious location prompt was approved, the user should revoke the site permission, clear the site data, close the tab, and avoid interacting with the page again. Long-term prevention includes updating the browser and operating system, preferring official apps and known domains, and disabling precise location when it is not needed.

8. Reader Preparation Checklist

Before approving location	Verify the domain, ask why the site needs exact location, and deny requests from unfamiliar links by default.
After a suspicious approval	Revoke the site permission, clear site data, close the tab, and avoid using the page again.
Long-term habit	Review browser permissions regularly, reduce precise-location sharing, and teach family members that Allow this time still releases sensitive data.

9. Ethical Consideration

This case study was conducted strictly for educational and analytical purposes. The site, tunnel, terminal session, and phone interaction were operated by the researcher in a bounded test scenario. The phone used in the screenshots intentionally acted as the victim device, and sensitive URLs, IP details, and location values were redacted before the report was prepared.

The defensive lesson is straightforward: browser permission prompts are not harmless pop-ups. They are trust decisions. When a site asks for location, the safest habit is to pause, verify the origin, and deny the request unless the need is legitimate and expected.