

No need to hack when it's leaking

GITHUB HEALTHCARE LEAKS

PROTECTED HEALTH INFORMATION ON THE PUBLIC WEB

*A collaborative report by
Jelle Ursem & DataBreaches.net
August, 2020*



EXECUTIVE SUMMARY

Personally Identifiable Information (PII) and Protected Health Information (PHI) that may be protected with industry-standard security while in an entity's secure environment lose that protection when a developer exposes login credentials in public GitHub repositories. In this report, we describe nine data leak incidents that potentially impacted an estimated 150,000 - 200,000 patients, **and possibly many more.**

i If you think that this can't happen to you because you don't use GitHub and don't even know what GitHub is, well, keep reading. One of your vendors or business associates may have an employee using it, as one provider discovered the hard way.

The leaks were commonly caused by developers:

- Embedding hard-coded login credentials in code instead of making them a configuration option on the server the code runs on;
- Using public repositories instead of private repositories;
- Failing to use two-factor or multifactor authentication for email accounts; and/or
- Abandoning repositories instead of deleting them when no longer needed.

Errors went undetected for months or years due to entities:

- Failing to audit their developer's security and compliance with security policies;
- Failing to have a monitored account for researchers to report security concerns; and
- Failing to respond to attempts at responsible disclosure for fear that the notification is a social engineering attack.

Recommendations for avoiding leaks on GitHub include:

- Forcing password changes periodically;
- Using 2FA or MFA for email accounts;
- Prohibiting the use of public repositories by your developers and requiring the use of private repositories; and
- Prohibiting the use of hardcoded login credentials in repositories.

OTHER RECOMMENDATIONS ARE INCLUDED IN THE REPORT.



CONTENTS

2 EXECUTIVE SUMMARY

4 OVERVIEW AND METHODS

6 MISUSE OF GITHUB

- 6 The Uber and Lynda Breaches
- 6 GnosticPlayers
- 6 Ransom Attempts
- 7 ShinyHunters
- 7 No Need to Hack When It's Leaking

8 FINDINGS

- 8 1. Xybion
- 10 2. MedPro Billing
- 13 3. Texas Physician House Calls
- 14 Malware
- 15 4. VirMedica
- 16 5. MaineCare
- 18 6. Waystar
- 21 7. Shields Health Care Group
- 23 8. AccQData
- 25 9. 'Unnamed Entity'

27 THE 'TYPHOID MARY OF DATA LEAKS'

29 KEY FINDINGS AND RECOMMENDATIONS

- 29 Findings
- 30 Recommendations

32 DON'T SHOOT THE MESSENGER

34 CONCLUSION

35 ABOUT US



OVERVIEW AND METHODS

It started when Jelle Ursem, a security researcher in the Netherlands, wondered, “Hey — let’s see if somebody is actually stupid enough to upload medical customer data to GitHub.”

☒ GitHub search results from one search.

It took Ursem less than 10 minutes to find that yes, medical data had been exposed on GitHub — and a lot of it. Ursem, who unabashedly claims to be ‘the lamest hacker you know’, uses variations on simple search phrases like ‘**companynname password**’ (or in this case, ‘**medicaid password FTP**’) to quickly find potentially vulnerable hardcoded login usernames and passwords for systems. You don’t even need to be a nerd to be able to do this, he notes — literally anyone could do the same.

After identifying potential ‘targets’, Ursem just... logs in — with the front door key. It doesn’t matter if the credentials Ursem finds relate to a database, an Office365 or Gmail account or a Secure File Transfer host. “**You just point the right software at it and hit ‘connect’.** *It really is that simple,*” Ursem explains, and the results are plentiful.

ⓘ For those who are not familiar with GitHub and would like an overview, see [Brown K: What Is GitHub, and What Is It Used For?](#) November 13, 2019.

☒ [Link to Article](#)

Retrieved July 12, 2020.

Once logged in to a Microsoft Office365 or Google G Suite environment, Ursem is often able to see everything an employee sees: contracts, user data, internal agendas, internal documents, emails, address books, team chats, and more.

→ **Just think about what that would mean if someone is able to snoop around as if they were your employee without ever triggering any alarms. Are you imagining that? Are you starting to feel anxious?**

In the past year and half, Ursem has contacted more than 400 entities to alert them to leaks. Without going into much detail, these entities include some of the biggest brands and companies in the world ranging from Fortune 500, publicly traded companies to private sector, from banking to entertainment, and from commercial to governmental.

For the findings reported in this paper, Ursem quickly found plenty of exposed protected health information, but it would take him months — and in some cases, the assistance of DataBreaches.net — to get some of these leaks secured.

Remember that no matter how big or small you are, there's a real chance that one of your employees has thrown the front door key under the doormat and has forgotten that the doormat is transparent.



In this report, we describe some of Ursem's findings from nine entities he recently tried to notify with DataBreaches.net's assistance. We look for patterns in what led to the unintended exposures, and we discuss the problems we encountered when we attempted to engage in responsible disclosure.

⌚ Spoiler alert:

You can think of that part as '*The Good, the Bad, and the Downright Infuriating*'.

We also share some recommendations for entities to prevent experiencing GitHub leaks. To that end, we will also introduce you to a developer we discovered online when we kept tripping over his mistakes in leaving credentials of his clients exposed. We refer to him as the '**Typhoid Mary of Data Leaks**'.



MISUSE OF GITHUB

Rather than just making general claims that bad things can happen, we thought it might motivate some entities more if we tell you about a few actual cases involving criminals finding and misusing leaked credentials.

THE UBER AND LYNDA BREACHES

We do not know exactly when criminals first started abusing it **or misusing GitHub**, but by 2016, threat actors were already using **password-reuse attacks**. Two threat actors exploited information stored on GitHub to gather data that helped them hack, and then attempt to extort, Uber and Lynda.com.

Brandon Glover and **Vasile Mereacre** used a custom-built account checker to determine if stolen data they possessed were also used as GitHub account credentials. If they were, they would search for valid GitHub account credentials for corporate employees and access employee accounts to search for Amazon Web Services credentials. Once they found the Amazon Web Services credentials, they found and exfiltrated valuable corporate data to use to try to **extort the victim companies**.

i DataBreaches.net recently heard from a prolific hacker who gave her a long list of his attacks and methods. After describing how they attacked some forums, he noted, “The rest are from checking private databases against GitHub.” This has probably been going on for years.

i Davenport, Shawn. GitHub Security Update: Reused password attack. June 16, 2016. [Link to Article](#)
Retrieved July 12, 2020.

GNOSTICPLAYERS

In 2019, prolific threat actors known as **GnosticPlayers** would also capitalize on the ability to gain information and access via GitHub. More than three dozen firms were hacked because the threat actors were able to brute force access to repositories and then use that access to search for employee credentials to valuable databases. In many cases, entities had no idea they had been hacked until the media contacted them to ask about their data being up for sale on ‘dark web’ forums and marketplaces.

i Uber paid \$100,000 ransom that they tried to cover up by having the threat actors sign non-disclosure agreements. Eventually, it all came out and Uber was fined \$148 million for covering up the breach that compromised the personal information of 57 million riders and hundreds of thousands of drivers. Chappell, Bill: Uber Pays \$148 Million Over Yearlong Cover-Up Of Data Breach. Sept. 27, 2018: [Link to Article](#)

RANSOM ATTEMPTS

Glover, Mereacre, and GnosticPlayers are not the only examples of threat actors misusing GitHub. In 2019, we also saw a rash of attacks on repositories similar to what we had seen with misconfigured MongoDB installations years earlier: attackers gained access to repositories,

wiped out their data, and then demanded ransom [to restore the data](#). In other recent cases, databases have just been wiped out maliciously and replaced with a “meow”. Almost 4,000 such databases were wiped out in quick succession in July, 2020.

ⓘ Cimpanu, Catalin:
A hacker is wiping Git
repositories and asking for a
ransom. May 3, 2019.

🔗 [Link to Article](#)
Retrieved July 12, 2020.

Franceschi-Bicchieri,
Lorenzo: Someone
Is Hacking GitHub
Repositories and Holding
Code Ransom. May 19,
2019.

🔗 [Link to Article](#)
Retrieved July 12, 2020.

SHINYHUNTERS

In May, 2020, threat actors calling themselves ‘ShinyHunters’ emerged to offer what they claimed was 500 GB of data from [Microsoft’s own private repositories](#). Their description and claims were not totally accurate, and investigation by Microsoft revealed that for the most part, the data they had acquired were sample projects and code snippets that had been intended for publication anyway. But ShinyHunters had accessed Microsoft private repositories, and that was enough to help them start to establish a reputation that the press would pay attention to. They subsequently disclosed numerous other hacks and data dumps.

ⓘ Abrams, Lawrence:
Microsoft’s GitHub account
hacked, private repositories
stolen. May 6, 2020.

🔗 [Link to Article](#)
Retrieved July 12, 2020.

NO NEED TO HACK WHEN IT’S LEAKING

The number of threat actors misusing or attacking GitHub repositories is anyone’s guess, because we suspect that no one actually checks or audits logs unless something makes them urgently aware of the need to check their security. But attacks are only one risk entities face. Perhaps the more pernicious risk is the risk of leaks that go undetected but may be capitalized on by threat actors or those who would hack but lack sufficient skills.



FINDINGS

“GitHub search is the most dangerous hacking tool out there”

— Jelle Ursem

In this section, we describe **nine leaks** discovered on GitHub by Ursem.

1. XYBION

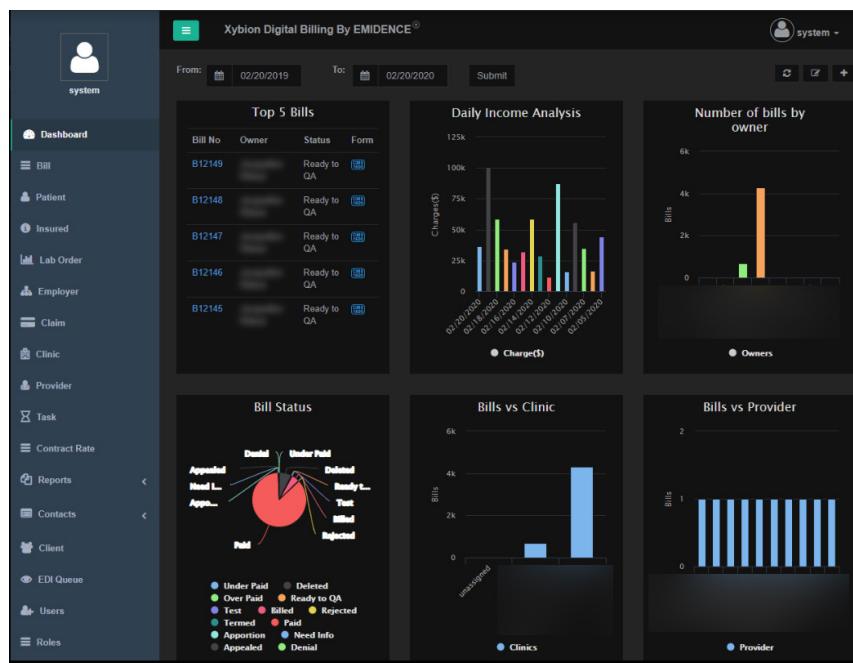
Xybion is a software, services and consulting company with a presence in workplace health issues. In February, 2020, Ursem discovered that one of their developers had left some code in a public repository that provided a system user's username and password. That code, in conjunction with other exposed code, gave Ursem full access to one of Xybion's billing backoffices, including data on almost 7,000 patients and more than 11,000 health insurance claims. The data seemed to have been publicly available since October 31, 2018.

Ursem first tried to contact the company himself. When they did not respond to his notification, and concerned that PHI was involved, Ursem gave the information to a well-known IT news reporter. In June 2020, when Ursem discovered that the data were still vulnerable to access, he contacted DataBreaches.net for notification help. When DataBreaches.net was also unable to get a response from Xybion, Dissent called one of Xybion's clients to alert them to the situation and to suggest that they contact Xybion to urge them to call DataBreaches.net. The next day, Xybion called DataBreaches.net and the data was secured.

In phone calls with DataBreaches.net and then Ursem, a Xybion representative apologized for ignoring all of the notification attempts. He acknowledged that they had received them all, but stated that the staff feared they might be social engineering attempts and so they hadn't responded. In a follow-up discussion with Ursem, Xybion constructively sought suggestions for ways to enhance their security and incident response. Now more than one month later, however, there is still nothing on their website that tells a researcher or anyone how to report a data security concern to a monitored account. Nor do we know if Xybion has implemented any of the changes discussed with Ursem. Xybion did not respond to a follow-up inquiry offering them an opportunity to issue a statement about the incident and asking them what changes they have actually implemented or have in the works.

i We originally intended to report on four leaks, but every time Ursem went on GitHub, he seemed to find more leaks. We finally called a halt to his searching for now so that we could get this report out.

Nor do we know if they have notified any of the patients whose PHI was exposed and/or if they have notified the U.S. Department of Health & Human Services or any state regulators.



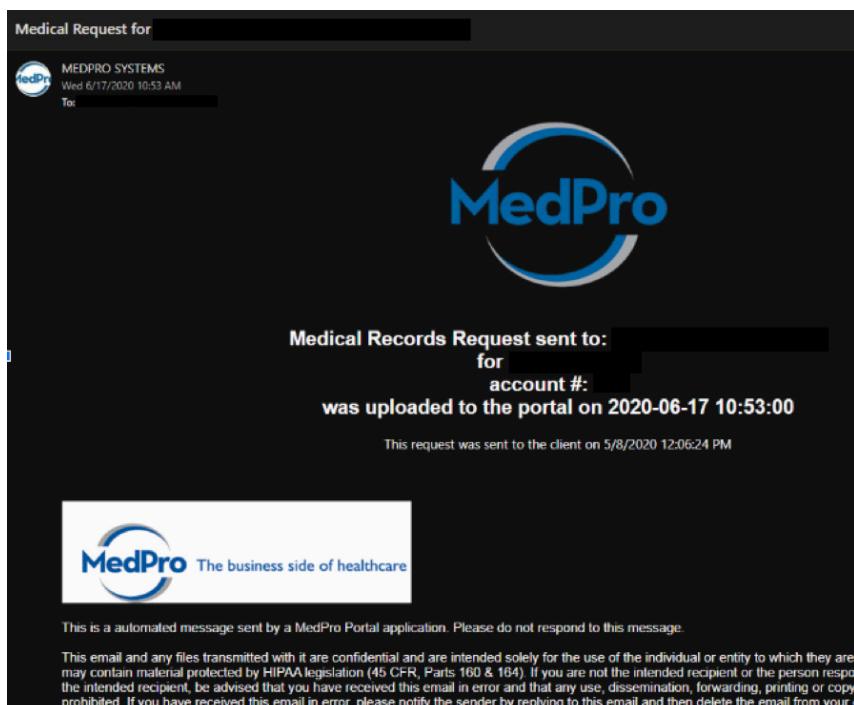
 **Xybion's Billing Dashboard**, showing how much money flows through the system, amount of paid and rejected insurance claims, billing sources and providers.

Xybion did not respond to a follow-up inquiry asking them what changes they have actually implemented or have in the works.

2. MEDPRO BILLING

MedPro in Florida provides medical billing and management services for the mental health and substance abuse community. Due to their developer's errors, they exposed protected health information for years. Patients' PHI was exposed on an SFTP server, in an Outlook mail account containing requests to pull patient data to support insurance claims, and also in a backup database. Any credentials needed to access them could be found in the developer's public repository, where several (working) windows domain access credentials were also listed.

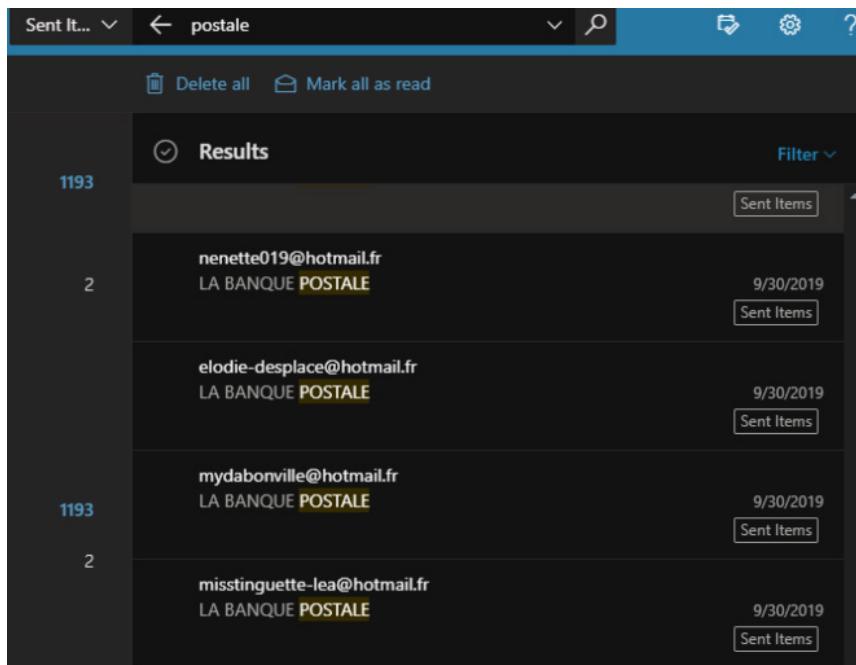
The earliest exposed files on the SFTP server appeared to date from 2015. The GitHub repositories have been online since 2016.



 **Anyone could view the record showing the MedPro had processed a request for a named patient of a named client and the patient's account number. Image redacted by Jelle Ursem.**

Email requests concerning named patients' PHI were not the only contents of the exposed mail account, however. The firm's email account appeared to have been compromised by spammers.

"It appears that this system was set up once, the application started running, and then this mailbox was probably never looked into again. There should have been a lot of monitoring for a system that processes PHI. Instead, there seemed to be none," Ursem tells DataBreaches.net.



 An example of French spam sent out via medprosystems.net

Both Ursem and DataBreaches.net attempted to contact MedPro multiple times and by multiple means. Ursem's earliest calls (in 2019) were met with an automated system that required him to know an extension that he didn't know, and repeated calls never were answered by a person or led to a person. More recently, when DataBreaches.net tried to reach the firm, emails to the addresses on the firm's website bounced back as not working. Ursem reluctantly finally left a detailed phone message. When there still was no response, DataBreaches.net started reaching out to some clients to alert them in the hopes that they could reach MedPro's owner and get them to lock down their data.

Surprisingly, the first client we reached out to — whose login credentials appeared in the public repository — informed us that they were not, and had never been, a client of MedPro's. They were understandably disturbed that the repository made it appear that they were a client and that their supposed login credentials were exposed.

DataBreaches.net eventually reached a few actual clients of MedPro and alerted them to the problem.

To this day, however, MedPro did not contact Ursem to get more information about their security issues. One of their clients did call DataBreaches.net back to thank us, and to say that when they called MedPro about the concerns, MedPro tried to say that the leak was the client's fault. To be clear: this leak was the result of the developer's

public repository exposing credentials that permitted access to some of MedPro's sensitive information. What Ursem found was in no way the fault of MedPro's clients, although we hope that in the future they will remember this incident and audit any vendor's or business associate's security a bit more.

We will have more on MedPro's incident response later in this report.

3. TEXAS PHYSICIAN HOUSE CALLS

Sumana Ketha, M.D. owns a number of domains and entities in Irving, Texas, including [Texas Physician House Calls](#). At various times, Dr. Ketha appears to have employed different developers who use GitHub as a private data store.

Neither of two developers protected their repositories, and both exposed patients' protected health information. One developer exposed 1,214 files in .pdf format on June 13, 2017, and they remained exposed. The other developer exposed approximately 100 files with protected health information in February, 2019, and they remained exposed. But there was more. Looking at the code, Ursem was able to find a backdoor that bypassed regular authentication. Experimenting with this, Ursem quickly found PII and PHI of approximately 4000 patients.

InstaMed Patient Batch:												
Show 100 entries												
Account Number	Last Name	First Name	Middle Name	Date Of Birth (YYYYMMDD)	Gender	Address	City	State	Zipcode	Phone Number	IsActive	Email
1008					F		Grand Prairie	TX			Y	
1016					F		Grand Prairie	TX			Y	
1019					F		Irving	TX			Y	
1020					M		Irving	TX			Y	
1049					M		Irving	TX			Y	
1050					F		Dallas	TX			Y	
1051					F		Dallas	TX			Y	
1053					M		Dallas	TX			Y	

 **Highly redacted**
Screenshot of a patient list
Ursem was able to access.

Ursem's research also revealed that one of the domains, risecorp.com, had apparently been hacked in the past and some content redirected users to a spam site in the U.K.

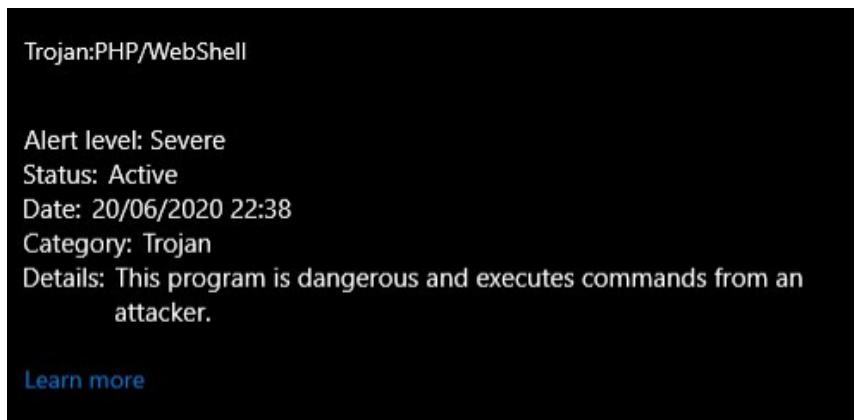
```
  <div class="wpb_wrapper">
    <header class="kd-section-title vc_col-lg-6 vc_col-sm-8 text-center kd-animated fadeIn vc_custom_1586938820338 kd-animate" data-animation-delay="200">
      <h2 class="separator_off">Our Client Reviews</h2>
      <h6 class="subtitle">
        "We are very proud of the "
        <a href="``NetentFreeSpins.co.uk``">service we provide</a>
        " and stand by every product we carry. Read our
        testimonials from our happy customers."
      </h6>
    </header>
  </div>
```

 **Risecorp.com HTML**
contained (an invalid) link
to a gambling site

But even more concerning was some malware Ursem found — malware that due to poor incident response, is still active on one of their live servers.

MALWARE

Ursem also discovered that they had unwillingly and most likely unknowingly integrated and uploaded malware into their codebase at two spots. *“ALL of the client data that has lived on this server since at least June 13, 2017 should be considered compromised”, Ursem asserted, adding, “How can you NOT detect you've been hacked for 3 whole years?”*



The screenshot shows a Windows Defender alert for a Trojan:PHP/WebShell. The alert details are as follows:

- Trojan:PHP/WebShell
- Alert level: Severe
- Status: Active
- Date: 20/06/2020 22:38
- Category: Trojan
- Details: This program is dangerous and executes commands from an attacker.

[Learn more](#)

☒ Windows Defender recognized the malware, which left us wondering: Are the developers running any antivirus at all?

The malware in question is a ‘**php web shell**’ that phones home to a server in the Ukraine. It accepts remote commands and is, for its form, quite sophisticated. It can automatically suggest exploits to root the server, finds files containing passwords and credentials, can upload, download, alter and delete files, and will remove traces of its traffic from the http log if it has the rights to do so.

After about six calls to different phone numbers, we made contact with someone in Dr. Ketha’s employ. That individual appeared to be in a rush to get off the phone and didn’t give Ursem the time to tell him about all of the other problems he had discovered. A follow-up text message Ursem sent him informing him that there were other issues to be discussed did not result in any response at all.

4. VIRMEDICA

VirMedica describes itself as the leader in e-access technology solutions designed to streamline patient access to life-enhancing therapies. As such, it is a business associate under HIPAA. VirMedica was acquired by CareMetx last year. CareMetx is a hub services provider for pharmaceutical and biotechnology manufacturers.

Ursem first discovered their leak in February. He noticed that the code had been up since January 2018 and attempted to notify them by phone and via their website 'contact us' form. **He received no response** at the time, and after alerting an IT journalist to the leak in case that journalist wanted to pursue it, Ursem did not attempt to notify VirMedica again until June, when he realized that protected health information was still exposed.

FTP login credentials were publicly available, as were large .csv and Excel files. Headers from some of the files indicated that they contained numerous types of protected health information, including demographic information on patients, their diagnoses, health insurance information, provider information, and insurance subscriber information. Other files were in .pdf format and also contained insurance information.

i This is a useful example of why entities should respond to notifications, however cautiously. As dedicated as Ursem is to responsible disclosure, he was busy with other things and after they failed to respond to his first notification attempt, he didn't even think about them again until months later. How many threat actors might have found the leak and exploited it during that time?

One of the files Ursem downloaded contained tens of thousands of records, although many appeared to be for the same individuals. In a statement to Ursem and DataBreaches.net, VirMedica's external counsel stated that after de-duplication, the files contained records on approximately 40,000 patients. In their statement, VirMedica confirmed that the downloaded data, which has since been securely deleted by both Ursem and DataBreaches.net, contained PHI fields:

“including but not limited to Patient Name (First, and Last); DOB; Medication; Primary Diagnosis; CPT Codes; Health Plan Policy Numbers, and Medicare Beneficiary Identifiers (in some instances). There were no Social Security Numbers in the SSN field. Many other fields found in the data set were not routinely populated or were not populated at all. Data accessed by the security researcher also included de-identified test data that appeared identifiable.”

VirMedica's counsel also informed Ursem and DataBreaches.net that their investigation had confirmed that Ursem's IP address was the only IP address that had accessed the files.

5. MAINECARE

On June 19, while looking into another leak, Ursem discovered a leak involving **MaineCare**, a program that is state- and federally-funded to provide healthcare coverage to Maine residents. Discovering that the leak exposed approximately 75,000 individuals' personal and protected health information and also gave him unexpected administrator rights to their entire website, Ursem immediately called the state to alert them.

He reached an employee who seemed to grasp the seriousness of the notification and tried to find someone to call him back. That was on Friday afternoon. No call came that day, or the next, or the next. In fact, it took many more calls and multiple means of contact and more than a week to get the state and its contractor, **DXC Technology** to respond. In a conference call with all hands on deck, Ursem was told that the first line handling the contact attempts had failed to stress the importance of the notification and our attempts had been filed under 'potential phishing attempt.'

 **DXC Technology**
acquired Molina Healthcare,
the entity leaking the
MaineCare data, back in
2018.

- While we applaud entities being cautious about possible social engineering or phishing attempts, reports need to be escalated and reviewed by someone capable of assessing their legitimacy.

In this case, a developer had exposed an impressive amount of sensitive data about MaineCare / Molina Health and himself, including:

- His passport
- His visa / H1B info
- His payslips
- His mortgage and contracts.
- His employee number
- Molina VPN login + password
- Molina domain accounts
- Molina help desk credentials
- Molina / MaineCare technical design documents marked confidential
- Molina Internal server infrastructure
- Mainecare SQL data sources with credentials
- mainecare.maine.gov production usernames, passwords and locations for Maine healthcare- and Medicaid-related websites, which, after logging in, provide access to PHI.

 Mphasis The Next Applied		
Reg C		
Pay slip for the month of September 2019		
Name		PAN
Employee No		Working
Designation	Senior Software Engg - Systems	Days Pa
Band	Band 4	PF No.
Level	Level 4	LOP
Location	CHENNAI	LOP PR MONTI
Bank Name	Citibank	Bank A
Earnings		Rs. Deduct
Basic Pay		Provide

While the state and its contractor secured the data and systems, we note that almost two months later, there is still no contact information on MaineCare's site that indicates how to responsibly disclose security concerns.

6. WAYSTAR

[Waystar](#) is a tech company that provides revenue cycle management (RCM) solutions for healthcare entities. It was built on the combination of two RCM organizations in 2017: Navicure and ZirMed.

In February, 2020, Ursem discovered that a developer's repository was exposing some of Waystar's clients' patients' protected health information. The data appeared to have been online since 2019. Ursem attempted to make contact with Waystar via an on-site contact form. Getting no response to his attempts, he promptly forgot about Waystar until months later when he re-discovered that the repository was still exposing PHI. On June 17, DataBreaches.net reached out to Waystar via LinkedIn. Once again, they did not respond.

After a few more exasperating attempts to get them to respond, and in a now-deleted tweet, Ursem gave Waystar one last chance to contact him before we went public with his findings.



 **SchizoDuckie**  @SchizoDuckie · 6h

Dear [@Waystar](#). I have been trying to responsibly disclose a PHI data breach to you since February. This is your last chance to contact me before I drop the document I'm writing up in collaboration with [@PogoWasRight](#) from [databreaches.net](#) this weekend

Within an hour their CTO, Chris Schremser, accepted DataBreaches.net's June 17th invite to connect on LinkedIn, explaining that they had not responded for fear that her message to them had been a social engineering attack. In that message, the CTO also claimed that although they hadn't responded to her, they had investigated and their investigation:

“... did reveal FTP credentials (usernames and passwords) for a small number of clients had been posted to an internet site. While that post was subsequently deleted, it was not before these credentials were captured and circulated on the ‘dark web’. We immediately remediated any credential that had not already been remediated through our normal course of operations. We also reviewed all FTP activity logs since the credentials were posted to ensure we did not see any activity that included inappropriate downloads of PHI. We did not find any inappropriate activity on these specific accounts, or more broadly across our FTP accounts.

As we completed the review of the firm's findings, the Waystar team agreed we should connect with you to describe the steps we had taken and engage with you to ensure we have remediated correctly the items you also may have knowledge of."

As they would learn when they finally connected with Ursem, there was a lot more they needed to do. And what was this bit about credentials circulating on the 'dark web'? We were not even aware of those.

When given an opportunity to provide a statement about the incident for inclusion in this report, Waystar submitted the following statement:

"As custodians of our clients' data, we take security seriously. On August 6th, we learned and confirmed that customer credentials were available on Github, and these credentials allowed for the compromise of three customer accounts. We have been working with a third-party cybersecurity firm to remediate this incident and conduct a thorough review of our security measures. At this point we are confident this is an isolated incident and have no evidence of any other unauthorized activity."

Their statement somewhat underplays the extent of the problems. What Ursem had found was six repositories for ZirMed and Navicure. Those repositories contained access credentials for clients of Waystar and Navicure. The credentials provided access to SFTP servers housing **EDI data** and Ursem sampled around 3000 records of PHI data. So their statement about three customer accounts is correct only insofar as Ursem sampled data from three.

EDI: Electronic Data Interchange standard.
A way for computers to exchange information in text files not designed to be read by humans.

Of special note, Ursem's attention was caught by what appeared to be a pattern involving their passwords. In short order, he figured out their password generation algorithm.

To their credit, on August 6, after communicating directly with Ursem for the first time, Waystar promptly created a responsible disclosure section with contact information linked from every page on their website. In a follow-up call on August 10 with Ursem, Waystar also acknowledged the fact that Ursem had, in fact, discovered a bug in their password generation mechanism that had been previously overlooked. Waystar subsequently found another bug in their cryptography related to storing passwords, which they fixed at the same time.

Waystar's incident is a cautionary tale about the need to investigate attempts at responsible disclosure. If we had given up trying to alert them to their problems, the consequences to Waystar, its customers, and its patients might have been severe.

7. SHIELDS HEALTH CARE GROUP

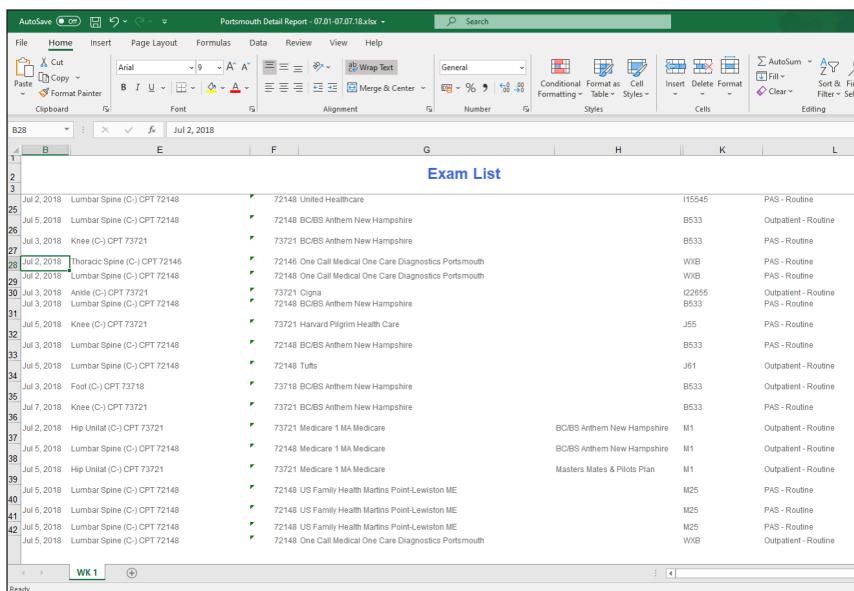
Shields Health Care Group is an MRI provider with locations throughout New England. On August 6, while following up on the Waystar/ZirMed findings, Ursem discovered that they, too, **were exposing PHI**.

The exposure, once again, was due to SFTP credentials being hardcoded into a GitHub repository.

The server contains a wealth of records about the company, including:

- their complete claims income and revenue history
- PHI on 5400 of their Portsmouth location patients
- the amount of clients with bad debt flags on their claims,
- and for some patients, complete large PDF files with full medical history.

ⓘ The server also lists some firmware upgrades for their on-site firewall, which would be a nice target for a sophisticated attacker to get even deeper into the organization.



Exam List				
Jul 2, 2018	Lumbar Spine (C-) CPT 72148	72148 United Healthcare	I15645	PAS - Routine
Jul 5, 2018	Lumbar Spine (C-) CPT 72148	72148 BCBS Anthem New Hampshire	B533	Outpatient - Routine
Jul 5, 2018	Knee (C-) CPT 73721	73721 BCBS Anthem New Hampshire	B533	PAS - Routine
Jul 2, 2018	Thoracic Spine (C-) CPT 72146	72146 One Call Medical One Care Diagnostics Portsmouth	WXB	PAS - Routine
Jul 2, 2018	Lumbar Spine (C-) CPT 72148	72148 One Call Medical One Care Diagnostics Portsmouth	WXB	PAS - Routine
Jul 3, 2018	Ankle (C-) CPT 73721	73721 Cigna	I22655	Outpatient - Routine
Jul 3, 2018	Lumbar Spine (C-) CPT 72148	72148 BCBS Anthem New Hampshire	B533	PAS - Routine
Jul 5, 2018	Knee (C-) CPT 73721	73721 Harvard Pilgrim Health Care	J55	PAS - Routine
Jul 3, 2018	Lumbar Spine (C-) CPT 72148	72148 BCBS Anthem New Hampshire	B533	PAS - Routine
Jul 5, 2018	Lumbar Spine (C-) CPT 72148	72148 Tufts	J61	Outpatient - Routine
Jul 3, 2018	Foot (C-) CPT 73718	73718 BCBS Anthem New Hampshire	B533	Outpatient - Routine
Jul 7, 2018	Knee (C-) CPT 73721	73721 BCBS Anthem New Hampshire	B533	PAS - Routine
Jul 2, 2018	Hip Unilat (C-) CPT 73721	73721 Medicare 1 MA Medicare	BCBS Anthem New Hampshire	Outpatient - Routine
Jul 5, 2018	Lumbar Spine (C-) CPT 72148	72148 Medicare 1 MA Medicare	BCBS Anthem New Hampshire	Outpatient - Routine
Jul 5, 2018	Hip Unilat (C-) CPT 73721	73721 Medicare 1 MA Medicare	Masters Mates & Pilots Plan	Outpatient - Routine
Jul 5, 2018	Lumbar Spine (C-) CPT 72148	72148 US Family Health Martins Point-Lewiston ME	M25	PAS - Routine
Jul 6, 2018	Lumbar Spine (C-) CPT 72148	72148 US Family Health Martins Point-Lewiston ME	M25	PAS - Routine
Jul 5, 2018	Lumbar Spine (C-) CPT 72148	72148 US Family Health Martins Point-Lewiston ME	M25	PAS - Routine
Jul 5, 2018	Lumbar Spine (C-) CPT 72148	72148 One Call Medical One Care Diagnostics Portsmouth	WXB	Outpatient - Routine

☒ One of the exposed files showed patients' first and last name, their date of birth, type of procedure, and their insurer. Redacted by Ursem to delete those columns.

Both Ursem and DataBreaches.net reached out to Shields via phone over the next few days. In a conversation with Ursem, a representative of Shields informed him that the developer in question worked for Surgical Information Systems.

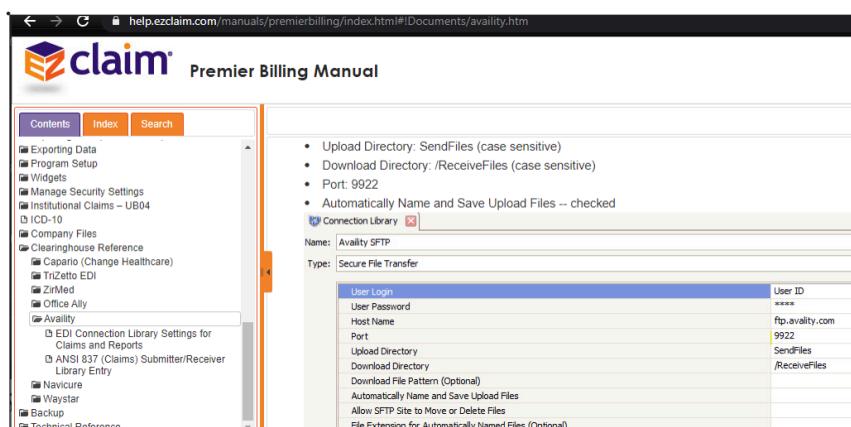
Happily for Shields, their investigation and analysis of logs showed that no one had engaged in unauthorized access to the patient files other than Ursem's one access. But as a spokesperson wrote to us, **it was a wake up call**:

“I’m sure you hear this all time, but we truly take security very seriously at this company. We appreciate you tipping us off, it was a good wake up call. Not only do we need to worry about making sure we are following best practices and keeping up with industry standards, but we also have to make sure that our vendors are doing the same. Especially when patient data is involved.”

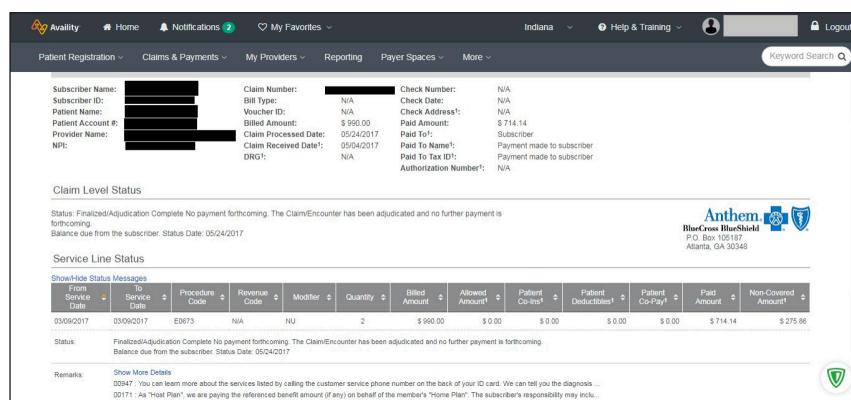
Out of all the entities we attempted to contact, Shields was one of the easiest to get a call back from, and they promptly investigated our report and then remediated their problem.

8. ACCQDATA

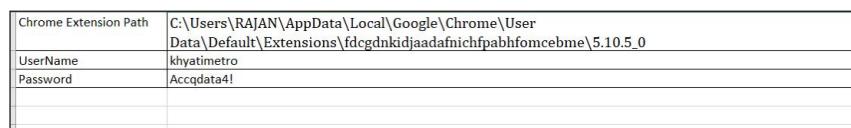
At one point, Ursem's research into Waystar/ZirMed also led him to a document for **EZClaim medical billing software**. Their manual listed a number of organizations that could assist with billing. Among them was Availity. Like Waystar, Availity provides RCM and EDI clearinghouse services. In looking into Availity, Ursem discovered two repositories containing protected health information that appeared to belong to them. One of the repositories appeared to contain more than 4300 records. As Ursem dug deeper into it, it seemed to originate not from Availity, but from a company named Acc-q Data Network LLC ("AccQData"). That hypothesis found some confirmation in invoices stored in the repository that showed the developer billing AccQData for his services.



 An image from EZClaim's manual listed a number of clearinghouses that inspired Ursem to see if they were leaking any credentials on GitHub.



 Availity record with patient name, subscriber info, and other claims data all exposed in plain text. Redacted by DataBreaches.net



 The developer also left login credentials exposed in the repository.

Before we had figured out that the repositories Ursem had found probably belonged to AccQData, Ursem had contacted Availity by phone on August 7 to alert them to the exposure. The first-line support

employee from Availity handled the report professionally, gathered all the information that was important, and promised to relay the information to the security team, thereby providing a text-book example of how it should be done.

Unfortunately Availity's good example wasn't the end of this story.

On August 8 Databreaches.net followed up on Ursem's phone call to verify that the information had been received and would be processed. Hours later, after calling AccQData and speaking to someone there, she received a phone call from "Raj", who may have been the developer himself. In a bizarre phone call that left Dissent wondering whether she might need to invest in better audio for her phone, he seemed to be accusing Ursem of being a blackmailer and mentioned something about the FBI investigating.

This was not the first threat Dissent had received in the process of trying to alert entities to their data leaks, and while we have no fear of the FBI, we do not take kindly to people defaming us — especially after we went out of our way to help them. So Dissent sent an email to AccQData.

Hello,

I received a phone call from "Raj" in response to my attempt to alert AccQData to a GitHub data leak involving AccQData patient billing data. The leak had been discovered by whitehat researcher Jelle Ursem. Ursem had previously called Availity and spoken with Shawn to alert them to the leak, and he had also emailed AccQData last night. My phone call this morning was to ensure that his email didn't wind up in some spam folder. Gladys told me she would have some [contact me](#), and I did get a call from "Raj."

Because I am hearing impaired and Raj was speaking quickly with a bit of an accent, I am not sure I understood what he was saying, and want to confirm it all with you or correct any misunderstanding.

1. It sounded like he was saying that the FBI was investigating the data leak. Is that correct?
2. It also sounded like Raj was saying that the researcher, Jelle Ursem, was a blackmailer or was attempting to blackmail AccQData. I hope I heard that incorrectly. Ursem is an ethical researcher. He has discovered and responsibly disclosed more than 400 leaks on GitHub in the past 1+ years. This is just another one. Ursem and I are collaborating on a whitepaper about GitHub leaks of PHI. This leak is one of 9 leaks that are included in that report. No patient's PHI will be exposed in our report. We both contacted AccQData to make sure the firm had time to lock down the data before our report [comes out](#). I am pleased to learn that it is now locked down.

I really hope I just heard Raj incorrectly, but if he was actually accusing Ursem of attempted blackmail, then that is libelous, and I trust the FBI will explain Raj's mistake to him after they read Ursem's email. AccQData owes Ursem a big thank you, not libelous accusations.

"Shooting the messenger" who notifies you of a mistake or breach is never appropriate. If AccQData wishes to make a statement for publication in our report in response to the leak, I will be happy to consider including it, but I will not repeat libelous accusations against a whitehat.

Kind regards,

 **Gritting her teeth,
Dissent sent an email to
AccQData about Raj's
bizarre phone call.**

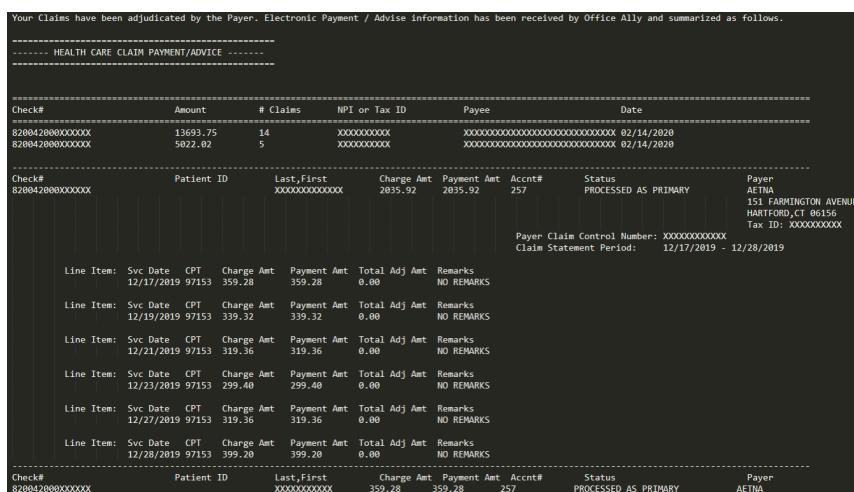
DataBreaches.net has not received any reply from AccQData. While we understand that people may be desperate to cover up their errors, their attempt to shift blame to us is not acceptable and will not stand.

9. ‘UNNAMED ENTITY’

→ *[At the time of publication, one of the leaks has yet to be secured. For that reason, we are not naming the entity, although we describe the incident.]*

A **multi-modality therapy provider** in the U.S. offers a variety of therapeutic modalities for children with autistic spectrum disorder.

On August 6, Ursem found an SFTP server belonging to a business associate of theirs that contained what may have been every insurance claim made by the provider until earlier this year.



The image shows a screenshot of a claims data file. The file header reads: "Your Claims have been adjudicated by the Payer. Electronic Payment / Advise Information has been received by Office Ally and summarized as follows." Below this is a section titled "----- HEALTH CARE CLAIM PAYMENT/ADVICE -----". The main content is a table of claims data. The columns include: Check#, Amount, # Claims, NPI or Tax ID, Payee, Date, Patient ID, Last,First, Charge Amt, Payment Amt, Acct#, Status, Payer, and Remarks. There are several entries for different dates (12/17/2019, 12/19/2019, 12/21/2019, 12/23/2019, 12/27/2019, 12/28/2019) and patient numbers (97153). The Payer listed is AETNA. The table also includes a section for Line Items with columns: Svc Date, CPT, Charge Amt, Payment Amt, Total Adj Amt, and Remarks. The file footer includes: "Payer Claim Control Number: XXXXXXXXXX", "Claim Statement Period: 12/17/2019 - 12/28/2019", and "Check#, Patient ID, Last,First, Charge Amt, Payment Amt, Acct#, Status, Payer".

Claims data for some pediatric therapy clients. Redacted by Ursem.

While digging further through the repository, Ursem also found that the developer had included a .sql file with the full name, address, birthdate and other identifying properties of 40 children that may have been the initial sample of pediatric patients registered into the system.

It took a few attempts and methods to reach them, but DataBreaches.net got a call back from their clinical director, who asked what he needed to do immediately to stop the problem. He also indicated that they had terminated services with the business associate months earlier.

In a follow-up the next day, the director expressed his gratitude to us for alerting him to the issue. And then he provided us with an unexpected update, informing us that a developer who had worked for them in the past left their usernames and passwords accessible, and when he contacted their former business associate to change the SFTP password and delete the data, that firm reportedly informed him that legally, they could not delete the data. To make matters even worse it

appeared that in 2015, someone used the login credentials, retrieved the source code, sql database structure, and data and posted it all on GitHub. The director had reached out to GitHub to ask them to delete the data, and when we last heard from him, was awaiting their response.

This provider's experience is a HIPAA nightmare and a cautionary tale, as if another one was needed, as to why login credentials should never be publicly accessible.



THE 'TYPHOID MARY OF DATA LEAKS'

How much damage can one developer do? A lot — if he is good at getting himself hired and keeps repeating the same mistakes as he goes from employer to employer without ever deleting his old or no-longer-need repositories. In Ursem's research, he came across a few names who were 'repeat offenders' when it came to exposing access credentials in public repositories. We will only tell you about one — the one we think of as the 'Typhoid Mary of Data Leaks'.

To give you a preview of where this is going: when Ursem was trying to wrap his head around the scope of the MedPro billing leak mentioned earlier in this report, he found so many things wrong that it was hard to know where to start reporting on it. It seemed that if there was any way this developer could do something wrong or mess something up, he would. And he seemed to be surprisingly unaware that everything he was doing was visible to others. Even after Github hit him with a DMCA Takedown request for an Ebook he improperly shared back in 2018, he continued to expose everything. If that takedown notice wasn't a wake-up call that others could see all his work, we don't know what would be.

In collaboration with DataBreaches.net, Ursem started analyzing and documenting the kinds of information that the developer had exposed, while we continued to try to make contact with the relevant entities to alert them to their exposure.

Some readers may characterize the following as errors. Most developers would probably characterize them as sins:

- Using Github as his personal sync solution for personal and work data
- Leaking credentials for no less than five individual entities/employers
- Hardcoding credentials to everything from (admin) Office365 accounts to database credentials and SFTP sites
- Not removing or hiding code that was written for previous employers

- Storing 800mb SQL backups with PHI from MedPro's clients' patients on Github
- Wrongful attribution of a client that wasn't even a client, leading Ursem and DataBreaches.net on a wild goose chase tracking down an innocent party — and making the non-client look like they foolishly hired a developer who would expose their login credentials publicly
- Exposing access to the telephone central system for a large entity in debt collection
- Exposing an estimated 200,000+ PII and PHI records (he had non-PHI disasters as well as PHI disasters).
- Uploading credentials for a web application error tracker that also logged PII
- Exposing credentials that lead to highly sensitive records for people with a history of substance abuse
- Apparently not adding sensitive systems to audit and monitoring controls.

Even though we have not been in direct contact with this person, we know that the message of his wrongdoings has sunk in with either him or his current employer now as he made all his repositories private on June 22, 2020. Sadly, some of them had already been cloned by third parties and remain publicly available as silent witnesses to the havoc he caused. Thankfully the cloned repositories do not include PHI and the passwords inside them have been changed. Ursem and DataBreaches.net are still trying to get the last of the fallout of his actions plugged and are trying to make sure that the companies affected are aware of the fact that their data has been out in the open for sometimes years on end.



KEY FINDINGS AND RECOMMENDATIONS

FINDINGS

Of the nine entities in this report, three were health care providers, one was a health plan, and the remainder were business associates or in third-party relationships. All of the three healthcare providers informed us that the developers in their case were contractors or employees of their business associates.

All of the leaks reported in this paper occurred because developers:

- embedded hard-coded login credentials in their code instead of making it a configuration option on the server the code runs on;
- used public repositories instead of private repositories;
- failed to use two-factor or multifactor authentication for email accounts; and/or
- abandoned repositories instead of deleting them when no longer needed.

Service providers also increased the risk of leaks by:

- failing to deploy IP address whitelists;
- not enforcing password resets; and
- not providing responsible disclosure mechanisms

We note that even had they used private repositories, threat actors can credential stuff or brute force their way into private repositories where they can then search for access credentials for corporate accounts.

The problems we observed were exacerbated by failures to audit business associates, vendors, and developers, and by failures to respond to responsible disclosure attempts.

At least three of the nine entities intentionally did not respond to early notification attempts and would later claim that they had been fearful the notifications were a social engineering attack. Their failure to respond left PHI exposed even longer and they risked never finding out about

their leaks had we given up after the first or second attempt. None of the three apparently made any attempt to google or investigate either Ursem or DataBreaches.net to determine if we might be legitimate.

As an additional finding, we note that many of the developers involved in these leaks were outsourced or contracted developers that were likely hired in cost-saving measures. One of them even provided evidence for that in his own leaking repository. His contract rate was RS 500/hour, which translates to less than \$7 USD an hour, for a person tasked with processing sensitive medical data. To be blunt, sometimes you get what you pay for.

Those who have not grown up in the U.S. may not fully appreciate our laws or the language of their instructions concerning keeping medical and personal data highly secured and private. We are not suggesting laziness or callous disregard of privacy or security on any developers' part. There may be a language problem or lack of training in the need to keep personal and sensitive information secure and confidential.

RECOMMENDATIONS

Security is never perfect, but entities can improve their security posture and incident response.

- Provide a way for researchers to responsibly disclose security incidents: Create a 'disclosure@yourcompany' e-mail address on your Contact / About us webpage that's monitored by your CISO, CSO or CTO, or MSP/IT provider.
- Train employees and especially your first line support and social media team on procedures for escalating notifications they receive.
- Teach them how to avoid a phishing attempt, but make sure they always escalate it to have it investigated by someone who has the skills to determine credibility of the communication.
- Regularly search GitHub for your firm's name and domain name(s). Even if you do not use a developer, one of your business associates or vendors might.
- Regularly force password changes and do not allow password reuse. If you have objections to this recommendation, at least rotate passwords used by former employees after they left.
- **Strong passwords are just as useless as weak passwords if you upload them to GitHub and do not change them for three years.**

- Lock down connections by IP address. Is there really a need for your webservice or secure FTP site to communicate with the whole world? Or do you just want to open the door for a couple specific people?
- Use 2FA or MFA for every third party service you use that supports it.
- Make sure to enforce admin approval of devices used for MFA and that every account gets logged in to at least once when enabling it. Ursem has encountered more than one case where he would have been able to establish his own phone number as the MFA authenticator.
- Require developers to use private repositories; prohibit public repositories. Recognize, however, that attackers may attempt to brute force or credential stuff to gain access to private repositories.

So:

- Never allow developers to embed passwords or authentication tokens in code repositories;
- Prohibit the use of real (production) data in GitHub repositories; and
- When developers terminate, ensure that their repository(ies) is/are deleted.
- Vet your business associates. Do they hold themselves to any information security standards like [ISO-27001](#)?

Even if you do all of the above, incidents may occur, so make sure you have a way for people to contact you to alert you. And then be prepared to respond promptly and appropriately.



DON'T SHOOT THE MESSENGER

While a number of the nine entities were genuinely appreciative of our volunteer efforts to help them secure their protected health information, two of the nine entities were abusive in their responses to our attempts to notify them that they had patient data exposed and vulnerable. Both of them claimed that the FBI was investigating or would be contacted. That's fine with us, as we have done nothing wrong.

But what all entities need to know is that when some people irresponsibly threaten those trying to alert them to a problem, they discourage others from trying to be helpful. Threats do have a chilling effect for many people. While there are those who might want to see us be understanding of people accusing us falsely because they are upset or misunderstood, we prefer to take a firm line that shooting the messenger is not acceptable, period.

In addition to the bizarre response by "Raj" in response to the AccQData incident, there was also one individual who called DataBreaches.net about MedPro and threatened Dissent with **reporting us to the FBI** or something. She did not seem to understand that we took to calling MedPro's clients because MedPro had not responded to multiple attempts to make contact with them directly. Even after the call -- and the caller hung up after yelling at Dissent -- MedPro never contacted Ursem to find out more about his findings. Had they done so, they might not still have a public repository on GitHub. **And no, we are not going to try to help them again.**

MedPro's response stands in sharp contrast to the email we received from one of their clients whom we had reached out to when we were still trying to get MedPro to lock down their data. **That now-former client wrote to us:**

"We appreciate your organization's efforts and dedication to ensuring third-party vendors who serve and support healthcare providers maintain the highest security standards and IT best practices available."

ⓘ The caller had their number hidden and did not give their name, but a client of MedPro suggested that the caller had been the owner of MedPro herself. Unfortunately, we have no way of verifying this claim.

ⓘ For an example of how heavy-handed responses to responsible notification may also come back to bite you in terms of bad publicity and then lack of needed cooperation, see these posts by DataBreaches.net: [Link to Article](#)

If you want researchers to let you know when there is a problem that you need to address, then reinforce responsible disclosure. Most researchers we know do not expect any bug bounty, reward, mention on your website, or even swag (although Ursem assures DataBreaches.net that most researchers would definitely appreciate it). **But don't underestimate the value of a thank-you note.** Even those of us who are not marketing our services still like hearing that our volunteer efforts to help secure data are appreciated.



CONCLUSION

In this paper, we presented nine examples involving leaks of PHI, but the problems we describe and the recommendations apply to other sectors as well. While it took Ursem only minutes for each case to find a total of a couple hundred thousand PHI records, it took a lot of time — in many cases, months — to get entities to respond to attempts to responsibly notify them of the leaks. Not one of these entities had a clearly posted means to contact them to report a data security concern. We tried phone, e-mail, Facebook, Twitter, LinkedIn, and even reached out to associates or clients if need be to relay the notification. Most people will not persist like that, and shouldn't need to. Regardless of what sector you work in, facilitating responsible disclosure is critical to your security.

There are undoubtedly many more leaks that can be found on GitHub, and we know that at least some threat actors are already using GitHub as a way to find login credentials in repositories. It is no longer sufficient to just search Google or Shodan or BinaryEdge for your firm's data or to search for signs of your firm's data on the dark web. **You also need to search GitHub.**



ABOUT US

JELLE URSEM

is a Developer / Devops engineer by day and ethical security researcher by night. Hailing from The Netherlands, he cheerfully disregards the yellow tape US researchers have to work around to not get sued and opts to directly inform companies that are at risk of being hacked or extorted. Over the past 1.5 years, he has accumulated over 400, mostly privately handled, responsible disclosures to his name. To contact Jelle about his research or this report, send an e-mail to jelle@esctunes.com. You can follow Ursem on Twitter [@SchizoDuckie](https://twitter.com/SchizoDuckie).

DATABREACHES.NET

is a non-commercial blog created in 2008 to report on leaks and data breaches. While much of the site's more than 27,000 posts represent news aggregation, the site also includes original reporting and commentary by '**Dissent Doe, PhD**' a licensed mental health professional. Dissent's reporting and watchdog complaints with federal agencies have resulted in enforcement action in a number of cases. Dissent can be reached via e-mail to breaches@databreaches.net. You can follow Dissent on Twitter [@PogoWasRight](https://twitter.com/PogoWasRight).

❖ *Design for this report was graciously donated by Gertrude Lok of 6500kelvin.nl*