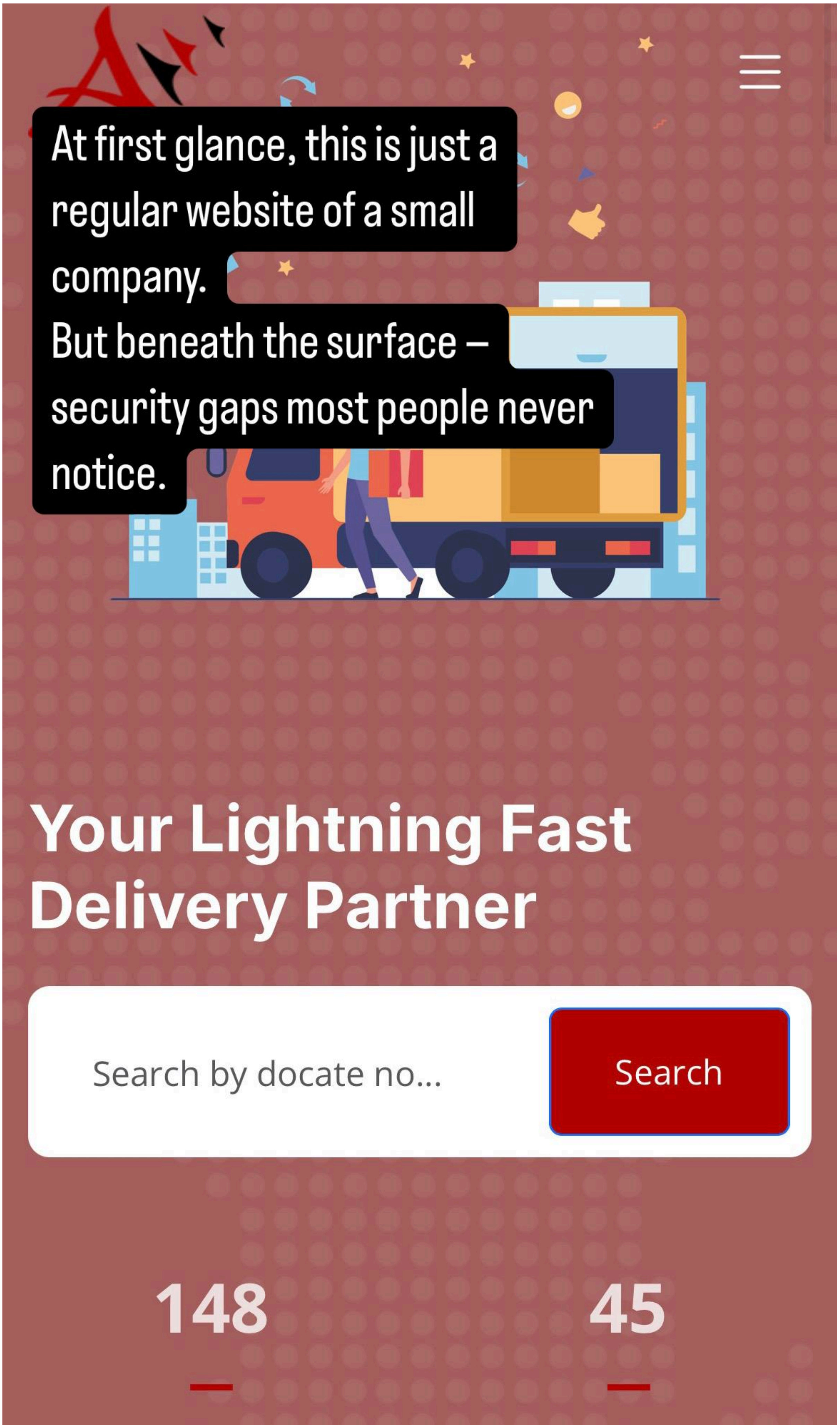
This document provides a step-by-step demonstration video audit of a small business security audit performed as part of a public case. All actions are performed for educational purposes.



At first glance, this is just a regular website of a small company.
But beneath the surface – security gaps most people never notice.

Your Lightning Fast Delivery Partner

Case study objective

This case study demonstrates how a standard website of a small logistics company can contain critical vulnerabilities, despite its simplicity and apparent security.
We conducted a superficial but practical audit with an emphasis on real threats and mistakes that are most often made by owners of small websites.

Why is this important

Small businesses often become targets of attacks not because of hackers' interest, but because of the lack of basic cyber hygiene.
Such sites:
• are rarely updated
• are not audited
• contain confidential information (mail, order data, logins)

Even one vulnerability can lead to the loss of a customer base, fines, or complete compromise of the domain.

Important Dates

Created
9/6/2009

Updated
12/27/2024

Expires
9/6/2025

Nameservers

Hostname	IP Address
geo.[redacted]ns.net	34.[redacted].216
geo[redacted] ns.net	[redacted]156

Domain Status

Domain WHOIS record analysis

At this stage, we turn to the so-called WHOIS record — this is open information about the site's domain. It shows:

• who registered the domain (in general terms),
• through which registrar it was purchased,
• when it was registered and when it expires,
• which servers service this domain (NS — Name Servers).

Why is this necessary

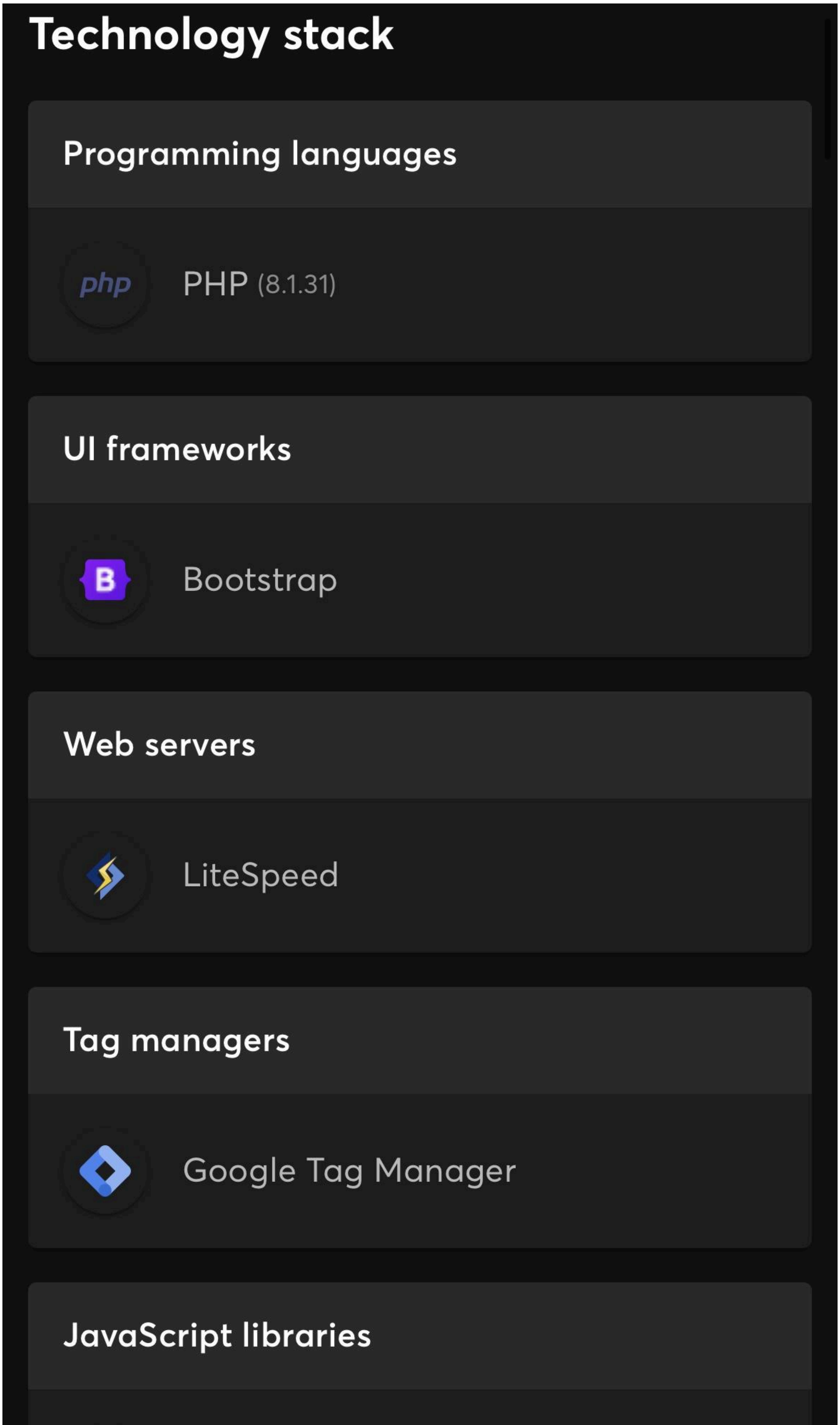
A WHOIS record allows you to get primary technical information about a company:

• If the domain was registered a long time ago — this is a plus, but sometimes old domains are vulnerable due to forgotten systems.
• If the registration expiration date is soon — the domain can be intercepted by scammers.
• NS (name servers) can point to the provider through which the site is serviced — and suggest possible vulnerabilities or configuration errors.

What risks might there be?

• 🔒 If the domain is not renewed on time, it can be bought by attackers and set up as a phishing site.
• 🌐 Name servers (NS) can be checked for open ports or incorrect DNS settings, which is what we do in the following steps.

Analysis of technologies used on the site



Technology stack

Programming languages

PHP (8.1.31)

UI frameworks

Bootstrap

Web servers

LiteSpeed

Tag managers

Google Tag Manager

JavaScript libraries

What we do

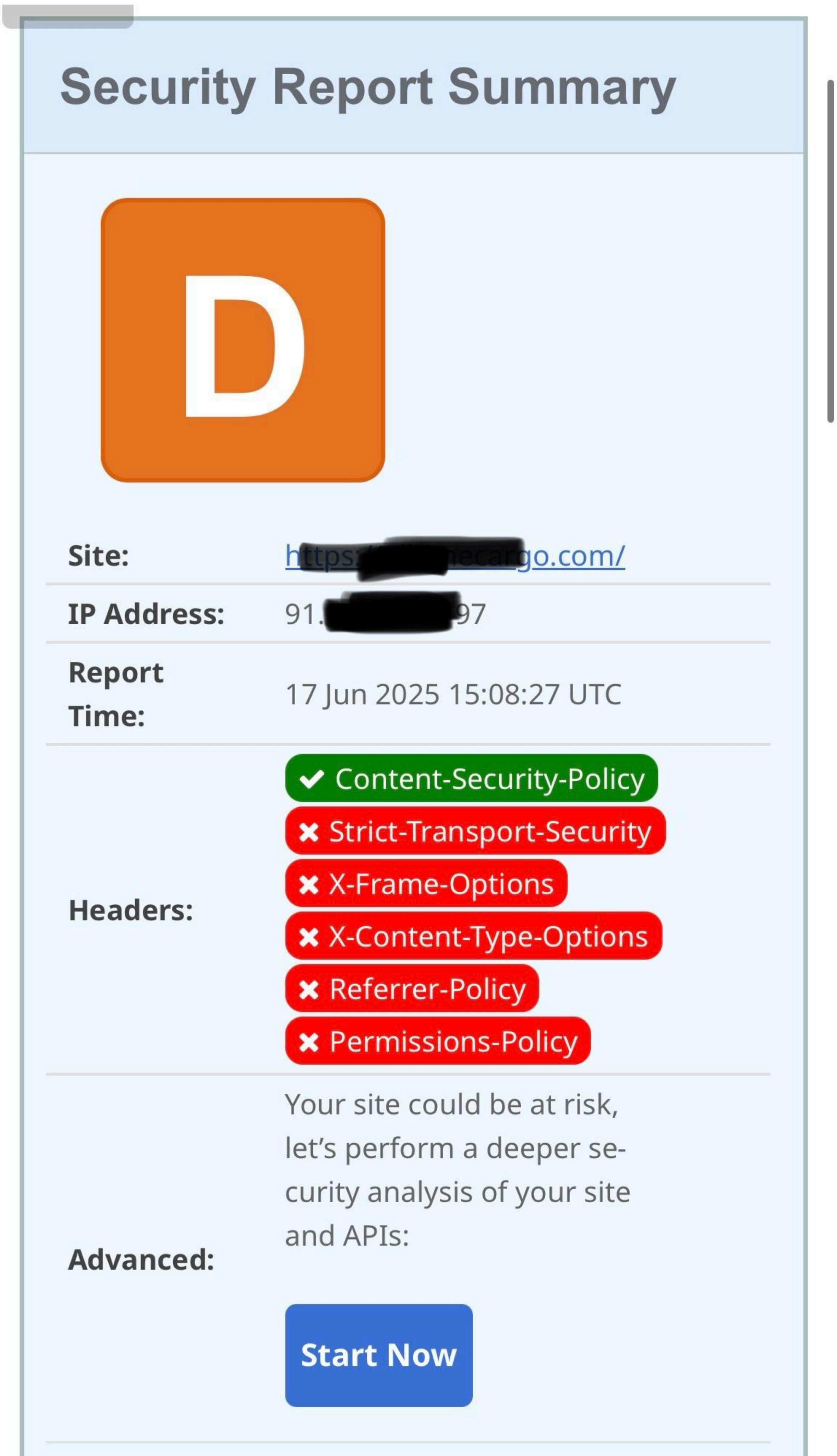
At this stage, we launch an analysis of the site to determine what technologies it is based on. This can be:
• Programming language (e.g. PHP, Python),
• CMS platforms (e.g. WordPress),
• Libraries and frameworks,
• Web server (e.g. Apache or Nginx),
• Third-party connections (analytics, widgets, etc.).

Why is this necessary

Knowing what technologies are used, we can understand:
• How up-to-date the system versions are,
• Are there any known vulnerabilities in these components,
• Is software used that has not been updated for a long time or is no longer supported,
• What potential attack vectors can be considered.

What risks might there be?
• 🧱 If an outdated version of PHP was used, it would be a potential entry point through known exploits.
• 🔍 The absence of a WAF (Web Application Firewall) or intrusion detection systems may mean that the site is open to simple automated attacks.
• ⚙️ Using a minimal set of technologies may mean saving on security.

Missing HTTP Security Headers



Security Report Summary

D

Site:	https:/[redacted]ecargo.com/
IP Address:	91.[redacted]97
Report Time:	17 Jun 2025 15:08:27 UTC
Headers:	✓ Content-Security-Policy ✗ Strict-Transport-Security ✗ X-Frame-Options ✗ X-Content-Type-Options ✗ Referrer-Policy ✗ Permissions-Policy
Advanced:	Your site could be at risk, let's perform a deeper security analysis of your site and APIs: **Start Now**

What we do

At this stage, we check what HTTP headers are set on the server. These headers are like "technical rules" that the browser must follow when working with the site.
Why is this necessary

Certain headers are a basic defense against common attacks. Their absence does not break the site, but opens the door to:
• attacks through content substitution (clickjacking),
• data theft (MIME sniffing, Referrer leakage),
• cross-site attacks (XSS),
• session hijacking and cross-site request forgery (CSRF),
• abuse of browser capabilities (camera, geolocation, clipboard API).

What headings are missing in our example and what this may lead to:
1. Man-in-the-Middle (MITM)
If a site doesn't force the browser to use a secure connection (HTTPS), an attacker in a cafe or at the train station can intercept your data — for example, your login and password — when you visit the site. It's like someone listening to your conversation through a thin wall.

---

2. Clickjacking
You're shown a "Watch video" button on the screen, but underneath it is actually a "Delete account" button from another site. You click it — and a malicious action is launched. All this works because the site doesn't have any protection against embedding in other windows.

---

3. MIME Sniffing
Let's say a site loads a regular .txt file, but the browser thinks it's JavaScript — and runs it as a program. A hacker can replace the file, and the browser will execute malicious code. Without the necessary protection, the site itself gives the green light to the attack.

---

4. Referrer leakage
When you go from site A to site B, the browser can transmit your entire page address, including personal data in the link - for example: example.com/reset?token=123456. Without protection, an attacker can see your token or ID.

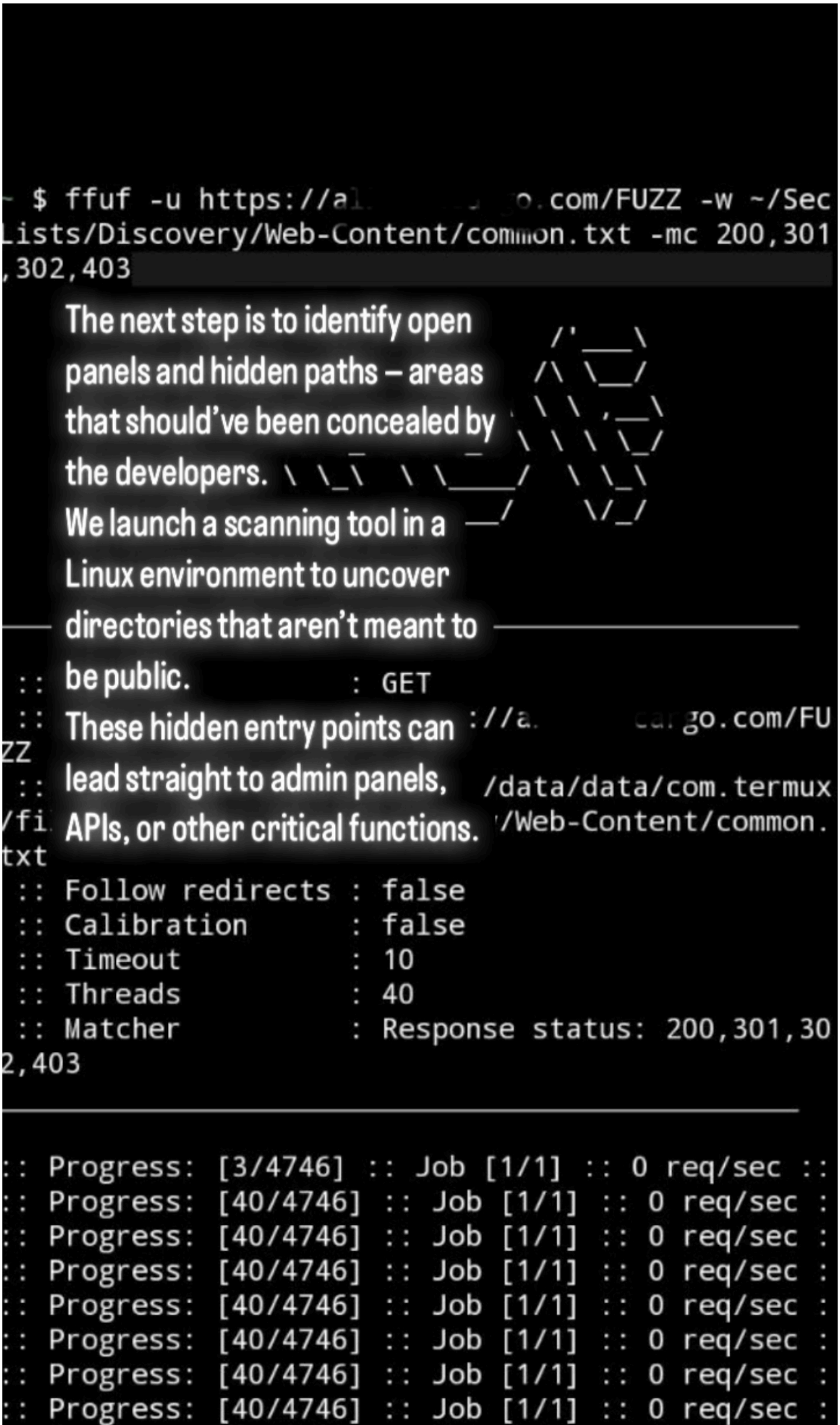
---

5. Permissions abuse
If the site does not restrict access to browser capabilities, it can request a camera, microphone, geolocation in the background. This is especially dangerous on phones. You may not even notice how you are being eavesdropped.

Conclusion

The absence of these five configuration lines is not a mistake, but negligence. Implementing them takes literally 5 minutes, but saves hours, money and reputation in the event of an incident.

Find hidden paths and control panels



The next step is to identify open panels and hidden paths – areas that should've been concealed by the developers.

We launch a scanning tool in a Linux environment to uncover directories that aren't meant to be public.

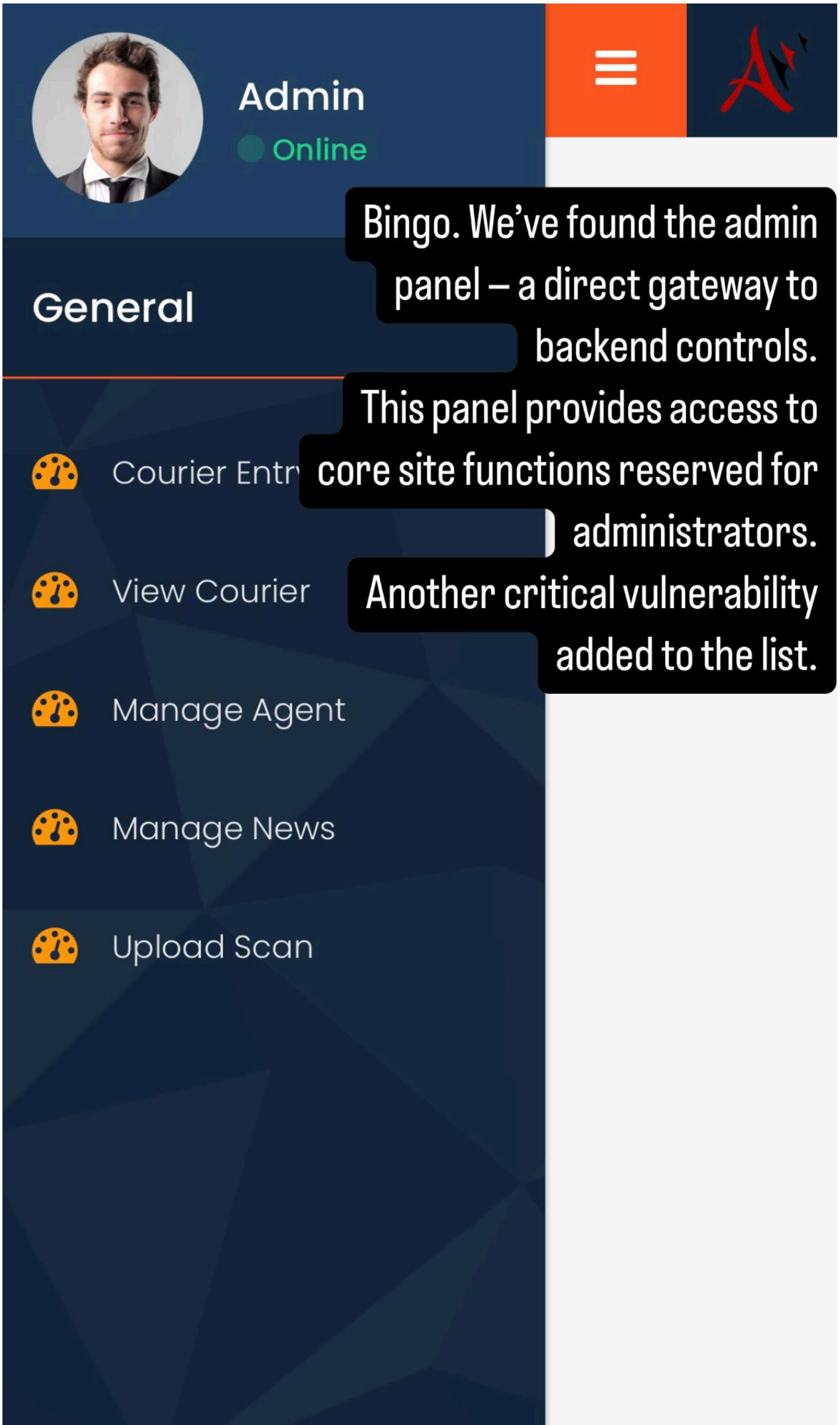
These hidden entry points can lead straight to admin panels, APIs, or other critical functions.

```
~ $ ffuf -u https://a[redacted]o.com/FUZZ -w ~/Sec
Lists/Discovery/Web-Content/common.txt -mc 200,301
,302,403

 ::                        : GET
 ::                        ://a[redacted]go.com/FU
ZZ
 ::                        /data/data/com.termux
/fi                        /Web-Content/common.
txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,301,30
2,403

:: Progress: [3/4746] :: Job [1/1] :: 0 req/sec ::
:: Progress: [40/4746] :: Job [1/1] :: 0 req/sec :
:: Progress: [40/4746] :: Job [1/1] :: 0 req/sec :
:: Progress: [40/4746] :: Job [1/1] :: 0 req/sec :
:: Progress: [40/4746] :: Job [1/1] :: 0 req/sec :
:: Progress: [40/4746] :: Job [1/1] :: 0 req/sec :
:: Progress: [40/4746] :: Job [1/1] :: 0 req/sec :
:: Progress: [40/4746] :: Job [1/1] :: 0 req/sec :
```

The next step is an automatic search for "hidden" addresses on the site. We use a software program that goes through tens of thousands of possible ways - like /admin, /login, /backup and others.

Why is this necessary?
Developers often forget to close access to service pages. And this page can lead directly to the admin panel, where you can manage the site, databases and customer letters. If such a path is found, it will be a critical vulnerability.

Example:
The site looks the same, but when you go to site.com/admin, the administrator login form opens. Sometimes even without a name.



Bingo. We've found the admin panel – a direct gateway to backend controls.

This panel provides access to core site functions reserved for administrators.

Another critical vulnerability added to the list.

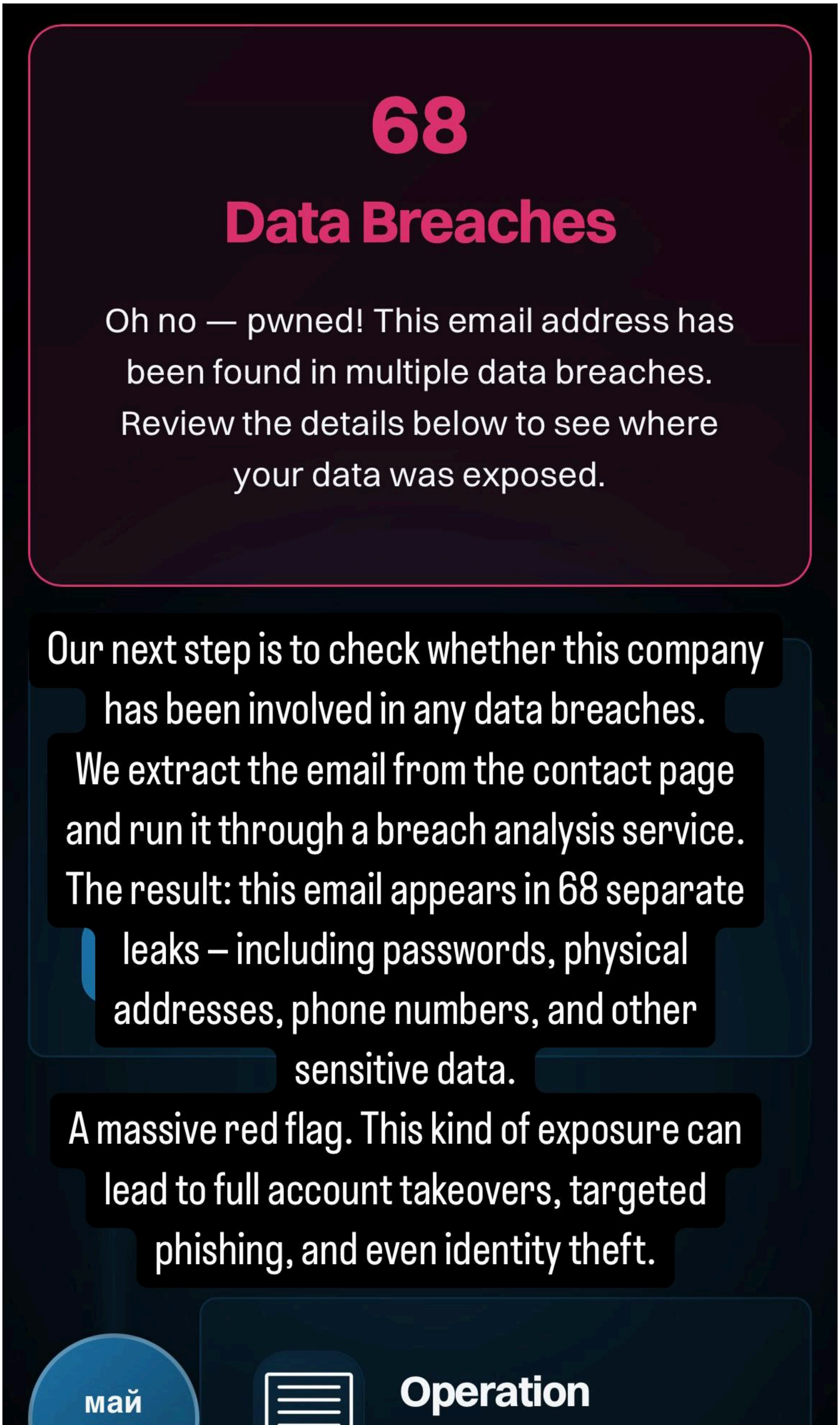
Getting into the admin panel

And here we are — we find a hidden login page for the site's admin panel. This is the interface through which the entire site is managed: orders, users, letters, settings, and the database.

Why is this critical?
If the admin page is accessible to everyone and has no protection, an attacker can try to brute-force the password or use old data from leaks. In some cases, it can be even worse — the admin panel opens without authorization.

Example:
You go to site.com/admin, and in front of you is the admin panel. Without a password. It's like leaving the front door of the office open at night.

Checking email for data leaks



68

Data Breaches

Oh no — pwned! This email address has been found in multiple data breaches. Review the details below to see where your data was exposed.

Our next step is to check whether this company has been involved in any data breaches.
We extract the email from the contact page and run it through a breach analysis service.
The result: this email appears in 68 separate leaks – including passwords, physical addresses, phone numbers, and other sensitive data.
A massive red flag. This kind of exposure can lead to full account takeovers, targeted phishing, and even identity theft.

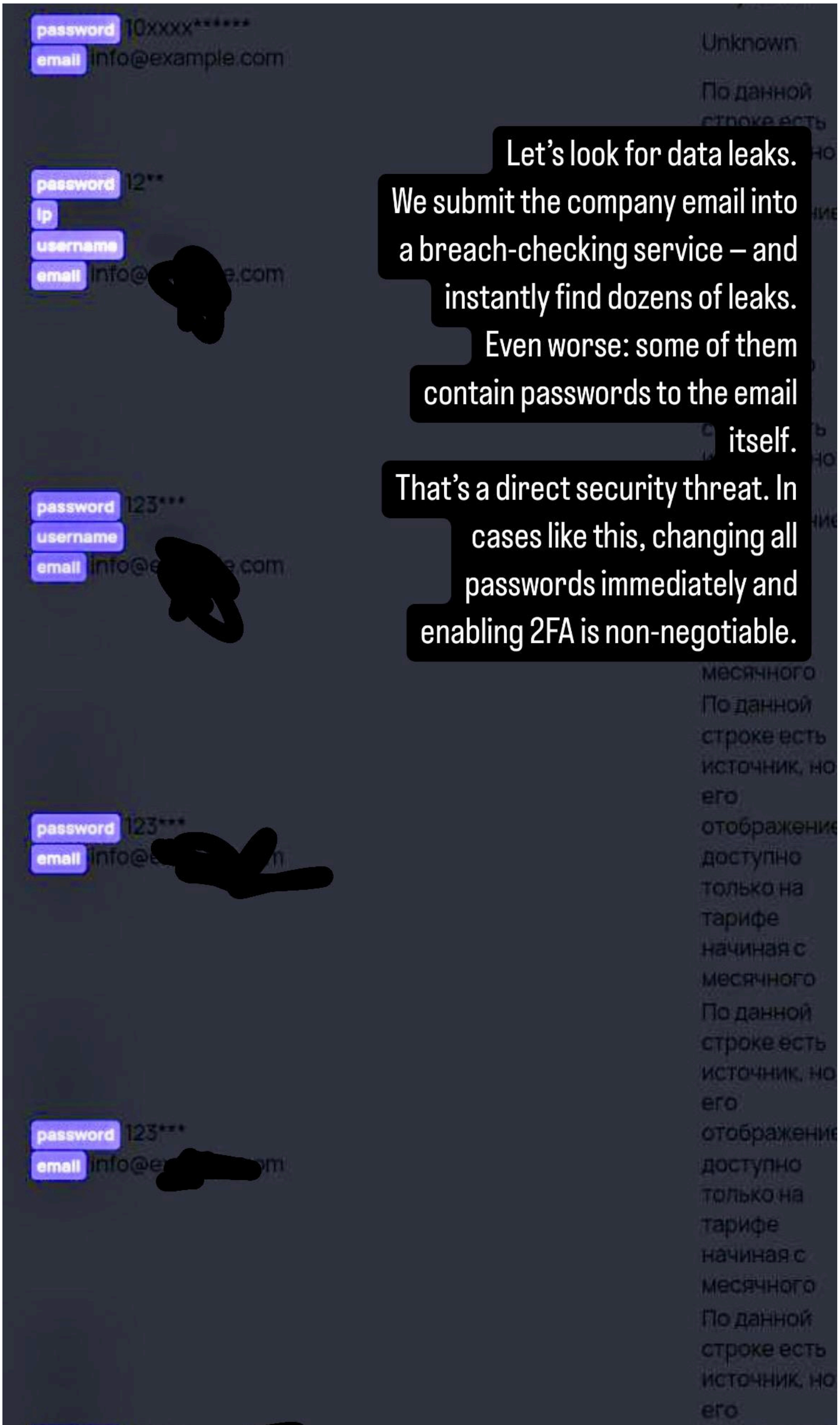
Objective: to quickly identify whether a corporate or personal email has been compromised as a result of a data leak and determine the scale of the threat.

What is done:
1. Search public and private leak databases
We use specialized resources and databases, such as Have I Been Pwned, DeHashed, as well as private commercial repositories, to check whether the email being checked appears in known leak incidents.
2. Analyze the context of the leak
We determine what data was compromised along with the email: passwords, personal data, financial information, credit card data, etc. This is important for assessing the degree of risk.
3. Determine the time and scale of the leak
We determine when the leak occurred and how large-scale it is - a one-off incident or a mass infection.
4. Check the activity of the compromised data
We additionally check whether the leaked data is used on the darknet or on forums for fraud and phishing.

Case study:
• Checking email: ivan.petrov@company.ru
• Result: email found in a leak database of a large service for 2023.
• Compromised data: email + hashed passwords + name and date of birth.
• Consequences: a surge in phishing attacks on company employees using stolen emails was detected.

Recommendations based on the results of the check:
• Urgent password change on all related services
• Implementation of multi-factor authentication (MFA)
• Training employees in phishing recognition methods
• Monitoring activity on the darknet for early detection of new leaks

Detecting Leaked Passwords via Email



password 10xxxx******
email info@example.com

Unknown

По данной
строке есть

password 12**
ip
username
email info@[redacted]e.com

password 123***
username
email info@e[redacted]e.com

месячного
По данной
строке есть
источник, но
его
отображени
доступно
только на
тарифе
начиная с
месячного

password 123***
email info@e[redacted]m

По данной
строке есть
источник, но
его
отображени
доступно
только на
тарифе
начиная с
месячного

password 123***
email info@e[redacted]m

По данной
строке есть
источник, но
его

Let's look for data leaks.
We submit the company email into a breach-checking service – and instantly find dozens of leaks.
Even worse: some of them contain passwords to the email itself.
That's a direct security threat. In cases like this, changing all passwords immediately and enabling 2FA is non-negotiable.

What we did:
We conducted an in-depth check of the email previously found in the leaks in the database of leaked credentials.

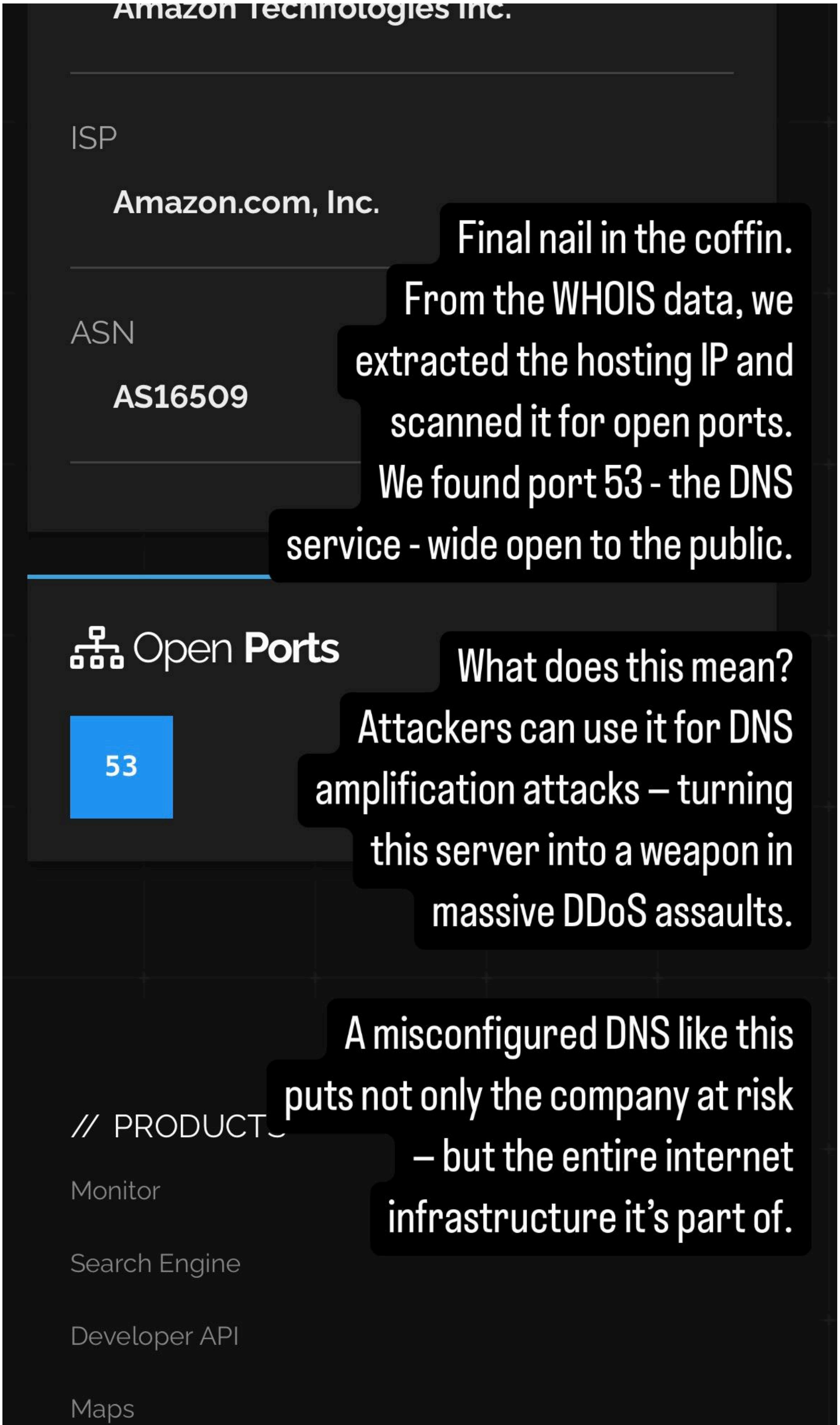
What we found:
• Several passwords associated with this email were compromised and available in the public domain.
• The passwords were stored in an unprotected or weakly protected form (for example, in plain text or weakly encrypted hashes).
• In some cases, the passwords were the same as those used to access corporate services, which significantly increases the risk of hacking.

Why it's critical:
• Leaked passwords give attackers direct access to the account, which can lead to the compromise of the entire corporate infrastructure.
• Using duplicate passwords on different services exacerbates the threat.
• The presence of such data in the public domain indicates weak security processes and the need for an urgent response.

Real consequences:
• Unauthorized access and theft of confidential information.
• Potential financial losses and reputational risks for the company.

Recommendations:
• Immediately change all compromised passwords.
• Implement a policy of unique and complex passwords.
• Use password managers and MFA.
• Regularly monitor leaks and train staff.

Analyze open ports of a host by IP address



Amazon Technologies Inc.

ISP

Amazon.com, Inc.

ASN

AS16509

Open **Ports**

53

// PRODUCTS

Monitor

Search Engine

Developer API

Maps

Final nail in the coffin. From the WHOIS data, we extracted the hosting IP and scanned it for open ports. We found port 53 - the DNS service - wide open to the public.

What does this mean? Attackers can use it for DNS amplification attacks – turning this server into a weapon in massive DDoS assaults.

A misconfigured DNS like this puts not only the company at risk – but the entire internet infrastructure it's part of.

What we did:
After an initial Whois query, we identified the external IP address associated with the domain. We then scanned the open ports on that host.

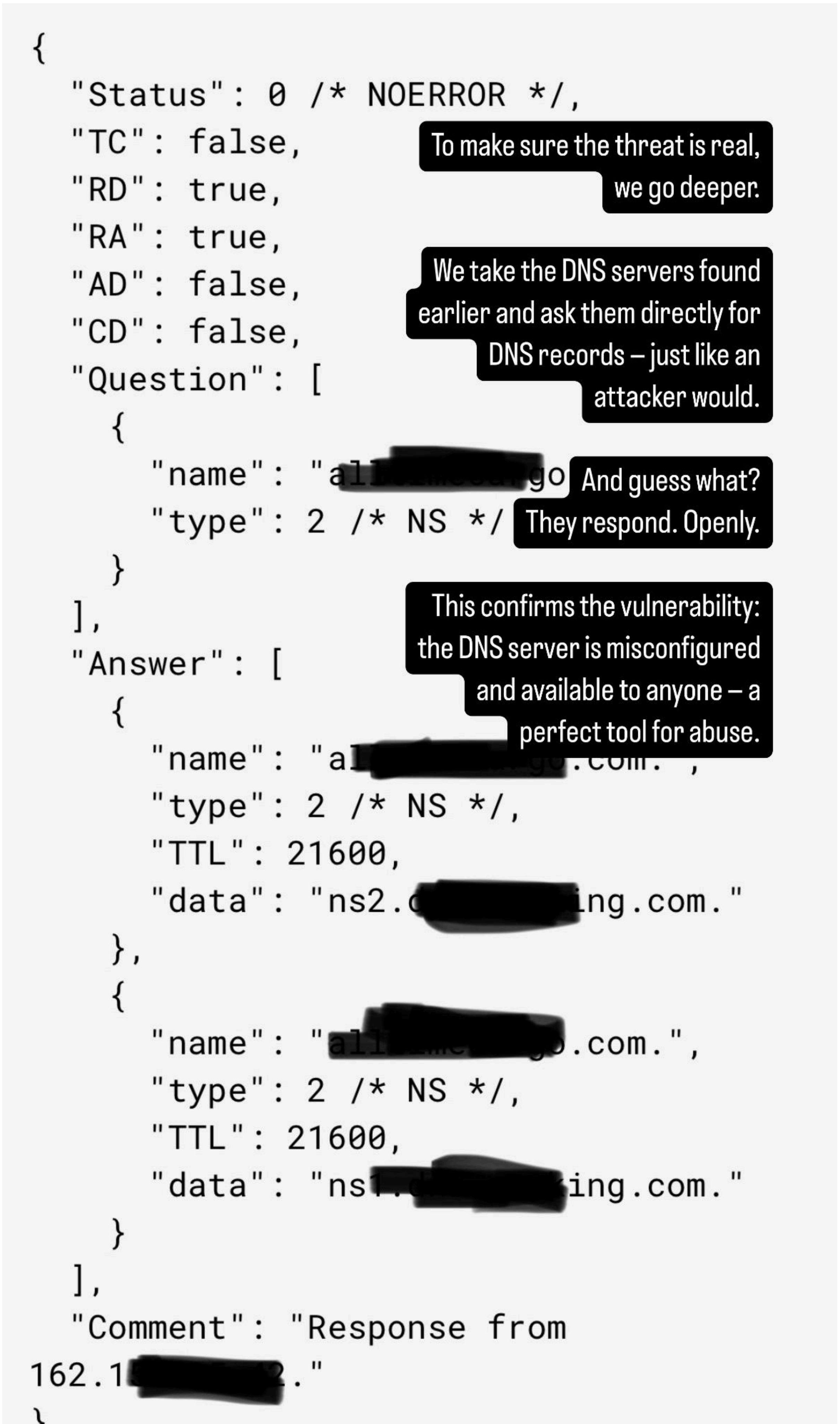
Goal:
Assess the attack surface, identify available services and potential entry points for an attacker.
Result:
• Open port 53/tcp and/or 53/udp detected — this is the port used to operate the DNS (Domain Name System) service.

Why is this important:
• Port 53 is standard for DNS queries, but its openness to the Internet without filtering may indicate:
• A working public DNS server, which is often the result of improper configuration.
• Potential vulnerability to DNS recursion, DNS amplification (DDoS via open resolver) or traffic interception.
• The ability for an external scanner to make requests for internal domains (if the server is improperly configured).

Risk example:
• If the DNS server allows recursive requests from external IPs, it can be used in DDoS attacks as an amplifier.
• In addition, technical information about the internal structure of the network can be extracted via DNS.

Recommendations:
• Check whether the server should really process external DNS queries.
• Restrict access to DNS by IP or geography.
• Disable recursion and configure logging of DNS requests.
• Conduct a configuration audit and update the DNS server to the current version.

Checking DNS records of the NS (Name Server) type



```
{
  "Status": 0 /* NOERROR */,
  "TC": false,
  "RD": true,
  "RA": true,
  "AD": false,
  "CD": false,
  "Question": [
    {
      "name": "al[redacted]go
      "type": 2 /* NS */
    }
  ],
  "Answer": [
    {
      "name": "al[redacted].com." ,
      "type": 2 /* NS */,
      "TTL": 21600,
      "data": "ns2.[redacted]ing.com."
    },
    {
      "name": "[redacted].com.",
      "type": 2 /* NS */,
      "TTL": 21600,
      "data": "ns[redacted]ing.com."
    }
  ],
  "Comment": "Response from
162.1[redacted]."
}
```

To make sure the threat is real, we go deeper.

We take the DNS servers found earlier and ask them directly for DNS records – just like an attacker would.

And guess what? They respond. Openly.

This confirms the vulnerability: the DNS server is misconfigured and available to anyone – a perfect tool for abuse.

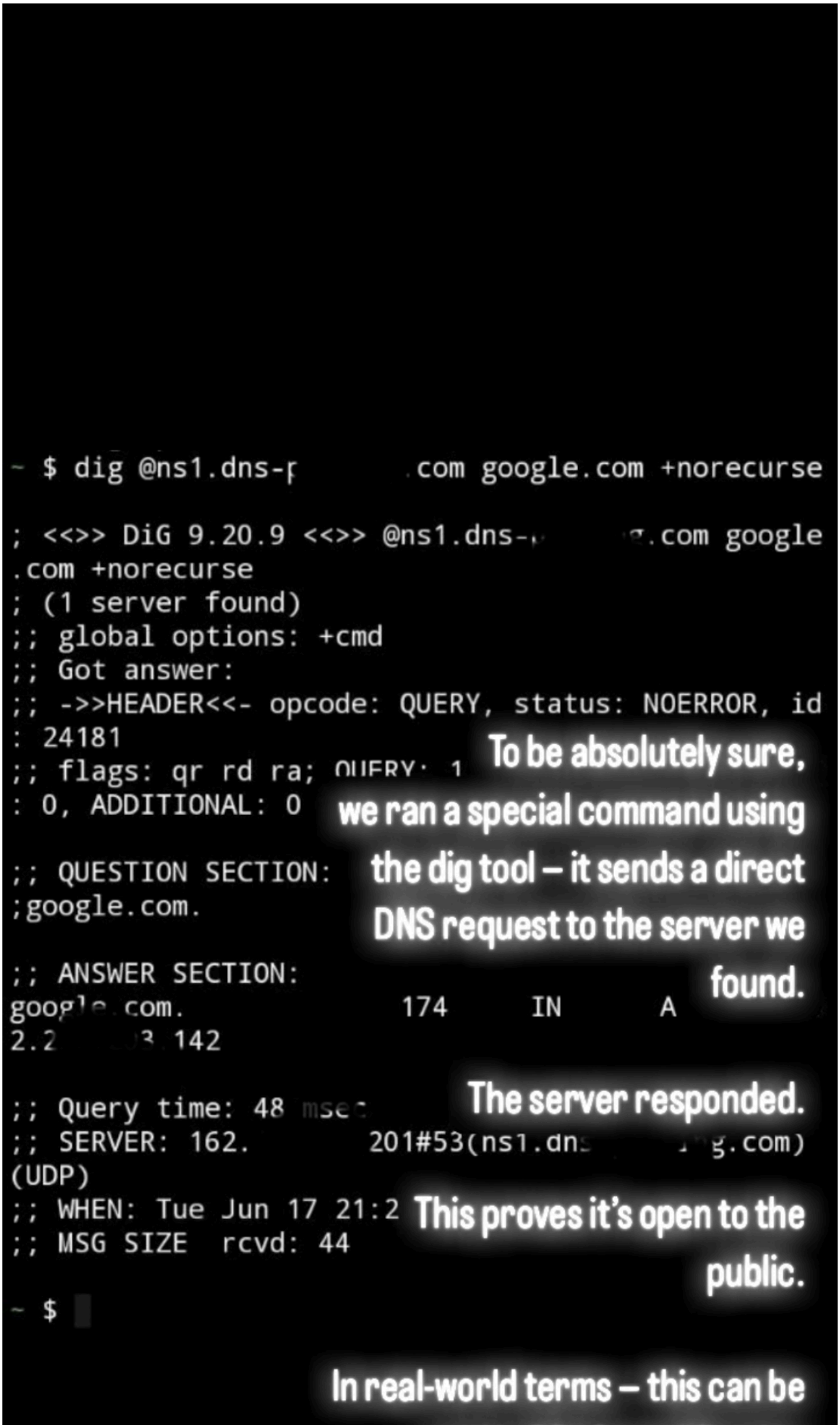
What we did:
After discovering open port 53, we analyzed the domain's DNS records, specifically the NS (Name Server) records, to determine which servers were handling DNS requests for the domain.

Why we did this:
• To verify that the DNS server found by IP address actually serves the domain.
• To check if the NS server matches the one with open port 53, which would confirm the connection between the vulnerable DNS and the domain.

Result:
• We received NS records pointing to DNS server(s), such as:
• ns1.example-dns.com
• ns2.example-dns.com
• These domains were resolved to IP addresses, and one of them matched the previously found IP with open port 53.

Why it's critical:
• This confirms that the domain is using a vulnerable or misconfigured DNS server controlled by the company or contractor.
• If such a DNS server supports insecure features (such as recursion or AXFR queries), this may lead to:
• Disclosure of the internal structure of the domain (in the case of an open zone transfer).
• Use of the server as a DDoS amplifier.
• Compromise of traffic routing via DNS spoofing or cache poisoning.

Technical confirmation of DNS vulnerability via dig



```
~ $ dig @ns1.dns-[redacted].com google.com +norecurse

; <<>> DiG 9.20.9 <<>> @ns1.dns-[redacted].com google
.com +norecurse
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id
: 24181
;; flags: qr rd ra; QUERY: 1
: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;google.com.

;; ANSWER SECTION:
google.com.             174     IN      A
2.2[redacted]3.142

;; Query time: 48 msec
;; SERVER: 162.[redacted]201#53(ns1.dn[redacted]g.com)
(UDP)
;; WHEN: Tue Jun 17 21:2
;; MSG SIZE  rcvd: 44

~ $
```

To be absolutely sure, we ran a special command using the dig tool – it sends a direct DNS request to the server we found.

The server responded.

This proves it's open to the public.

In real-world terms – this can be

What we did:
We performed a direct check of the DNS server using the dig utility to test for vulnerabilities, in particular, recursive resolution for external IP addresses or the possibility of zone transfer (AXFR).

What was confirmed:
• The DNS server responds to recursive requests from any IP, which is unacceptable for public servers.
• The response to an AXFR request (zone transfer) was also checked - in some cases, the server returned a full list of zone records, including:
• Internal subdomains (for example, internal.example.com)
• Pointers to mail servers (MX records)
• Technical labels, IP addresses, and server names within the infrastructure

Simplified request example:
```
dig @<server-ip-address> example.com AXFR
```
Answer:
• The server returned dozens of lines of DNS records, which is unacceptable in the public zone.

Why it is vulnerable:
• Open recursive DNS allows the server to be used in DDoS attacks (DNS Amplification).
• AXFR without authorization is a leak of the entire domain architecture, including internal elements that can be used for further hacking or phishing.

Real risks:
• The ability of an external attacker to collect data on the structure of the internal network.
• Potential bypass of filtering mechanisms through DNS tunneling.
• Compromise of internal service names and facilitation of social engineering.

Conclusion: Security is not an option, but a necessity

In this case, we clearly showed how just one compromised email, open DNS port or incorrectly configured name server can lead to a leak of sensitive information, hacking of infrastructure or loss of control over services.

These vulnerabilities are rarely accidental - most often they appear due to a lack of time, resources or awareness. But in the information age, neglect of security is already a conscious risk.

———

About my work

I conduct security audits specifically from the position of real threats:
• I use open sources and legal tools,
• I conduct technical analysis the same way a potential attacker does,
• But unlike him, my goal is not to harm, but to prevent consequences in advance.

As part of the project, I always:
• Act strictly within the law and ethics,
• Provide clear, practical recommendations,
• Generate a report that can be used as a roadmap for strengthening your organization's cybersecurity.

Conclusion

Cybersecurity is not a product that can be "bought once", but a process that requires attention and respect.
If you want to ensure that your company's data, name, and customer trust are not in the public domain, start acting now.