Activity 4-1

**Step 1 – Access the Domain Dossier Website**

I opened my web browser and visited:  
[**https://centralops.net/co/domaindossier.aspx**](https://centralops.net/co/domaindossier.aspx)



This site provides passive reconnaissance features such as Whois lookup, DNS record inspection, traceroute, and network information.

**Step 2 – Perform Whois Lookup on mit.edu**

I typed mit.edu in the Domain Dossier search field, checked Domain Whois record, and clicked Go.

The tool returned information including:

* Registrar details
* Domain owner information
* IP addresses
* Name servers
* Domain creation and expiration dates
* Limited contact info due to GDPR restrictions



**Step 3 – Note the IP Addresses and Name Servers**

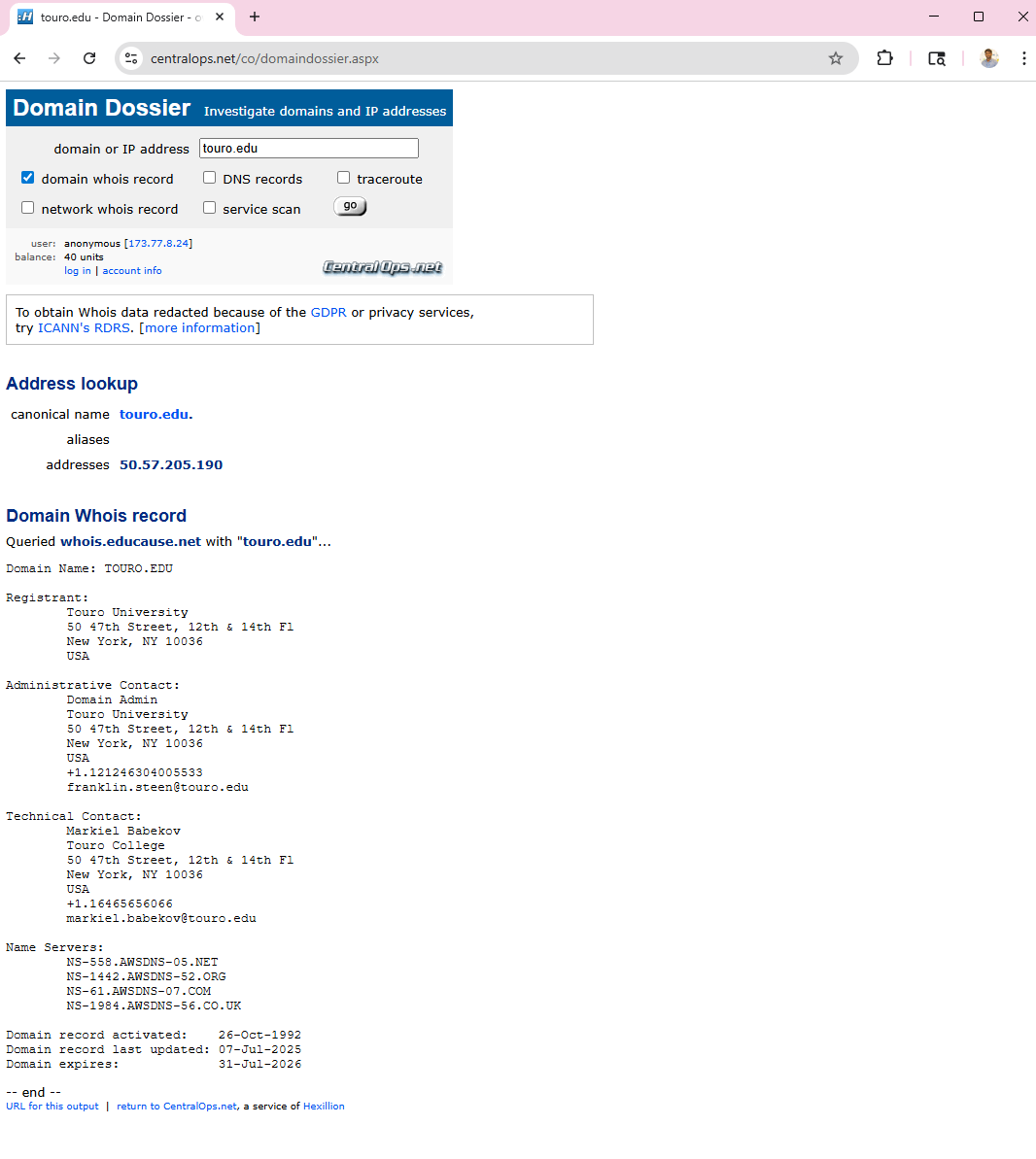
Example MIT results (your results may vary):

* IP Range: 18.0.0.0 – 18.255.255.255
* Name Servers:
  + ns1.mit.edu
  + ns2.mit.edu
  + ns3.mit.edu

**Step 4 – Tested Additional Organization Domains**

I repeated Steps 2 and 3 using several other domains.

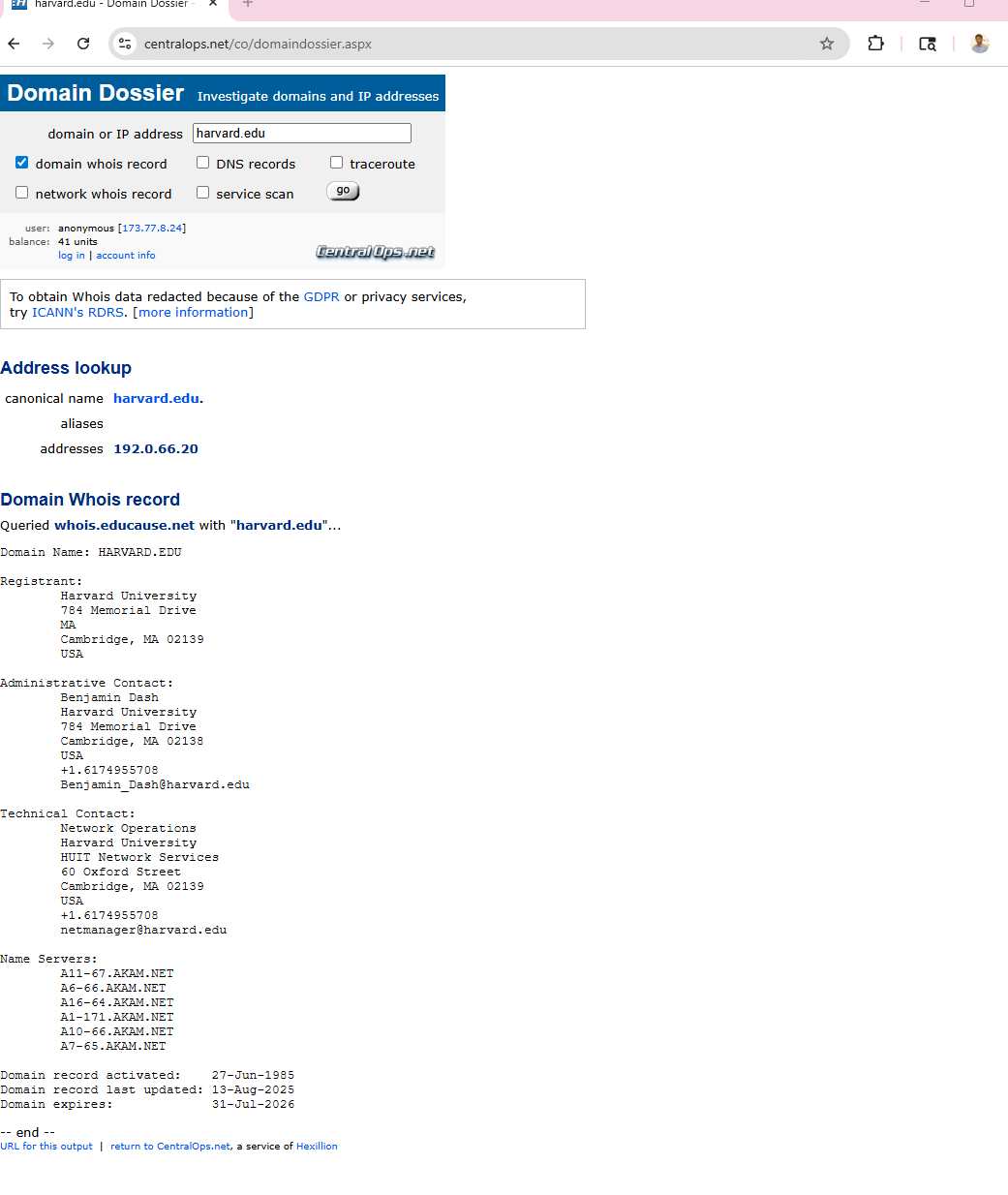
1. Touro.edu



* IP Address: 50.57.205.190
* Administrative Contact: Domain Admin
* Technical Contact: Markiel Babekov
* Name Servers:
  + NS-588.AWSDNS-05.NET
  + NS-1142.AWSDNS-S2.ORG
  + NS-431.AWSDNS-07.COM
  + NS-1984.AWSDNS-56.CO.UK

**Observation:**  
Touro exposes full names, phone numbers, and professional emails. This increases the risk of social engineering.

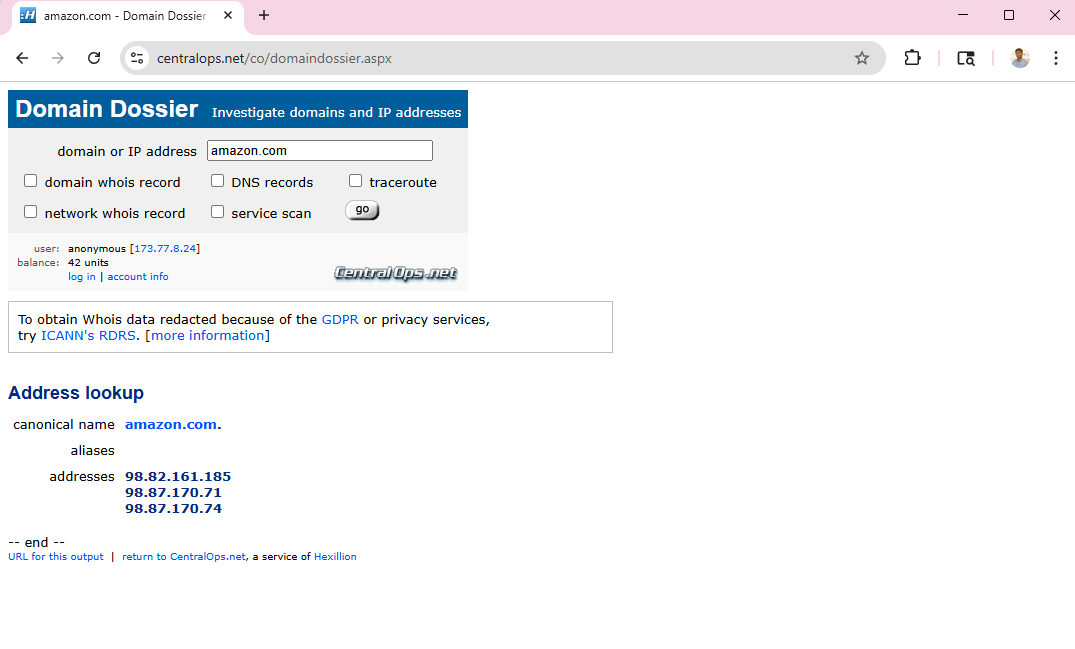
**2. Harvard.edu**



* **IP Address:** 192.0.66.20
* **Administrative Contact:** Benjamin Dash
* **Technical Contact:** Network Operations
* **Name Servers:**
  + A11-67.AKAM.NET
  + A6-66.AKAM.NET
  + A16-64.AKAM.NET
  + A1-117.AKAM.NET
  + A10-65.AKAM.NET
  + A7-65.AKAM.NET

**Observation:**  
Harvard includes both personal and role-based contacts. Role-based emails (like netmanager@harvard.edu) reduce exposure.

**3. Amazon.com**



* **IP Addresses:**
  + 98.82.161.185
  + 98.87.170.71
  + 98.87.170.74

Name servers and registrar information were hidden due to GDPR and privacy shielding.

**Observation:**  
Amazon shows very little data, which improves security against reconnaissance.

**Step 5 – Access the DomainTools Website**

**I opened a new tab and went to:**[**https://whois.domaintools.com**](https://whois.domaintools.com)



This site provides Whois records, hosting information, DNS details, and historical changes.

**Step 6 – Comparison of Results Between Domain Dossier and DomainTools**

Similarities

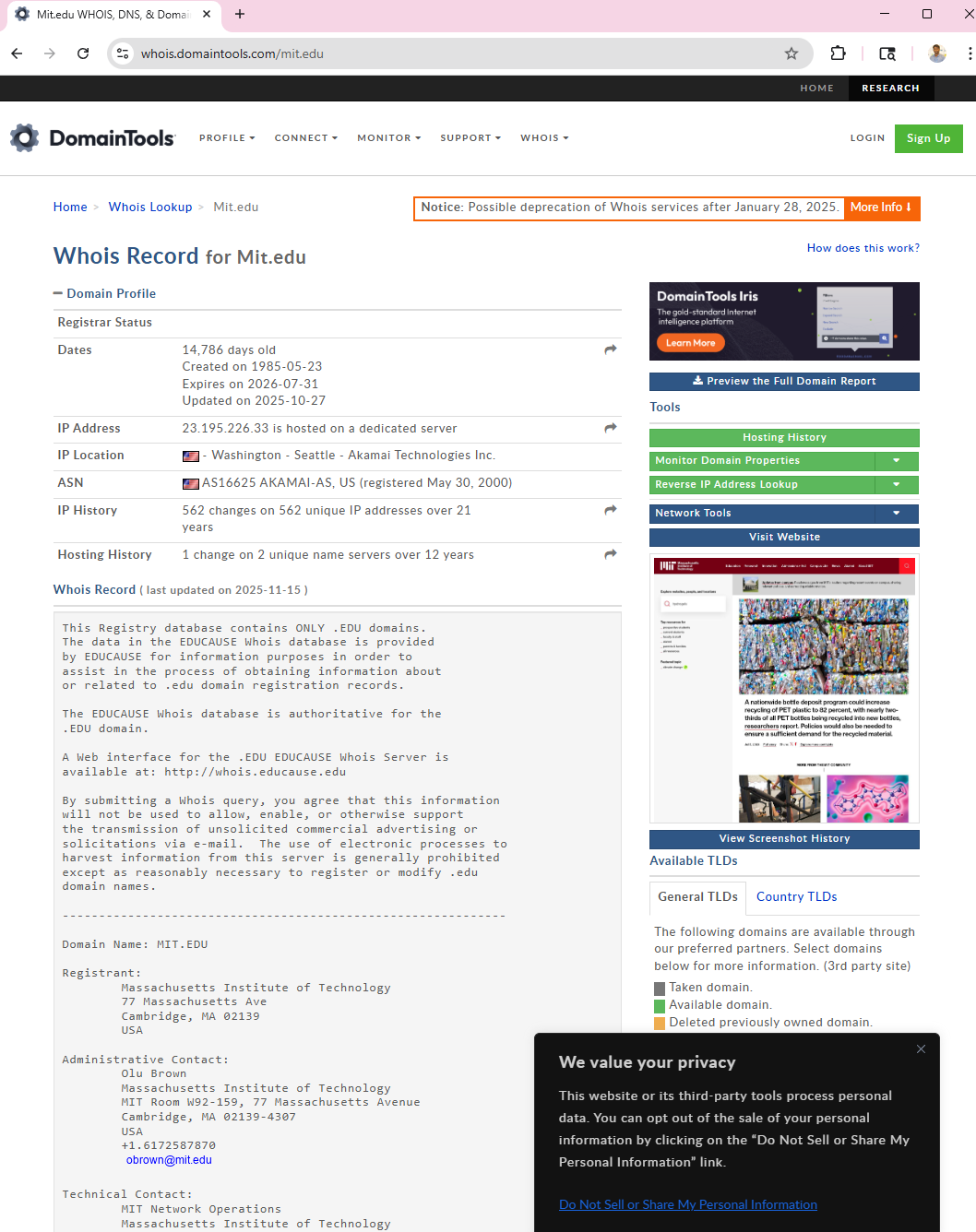
* Both returned identical domain creation and expiration dates.
* Both listed the same name servers for the domains.
* Both showed GDPR-redacted data for Amazon.

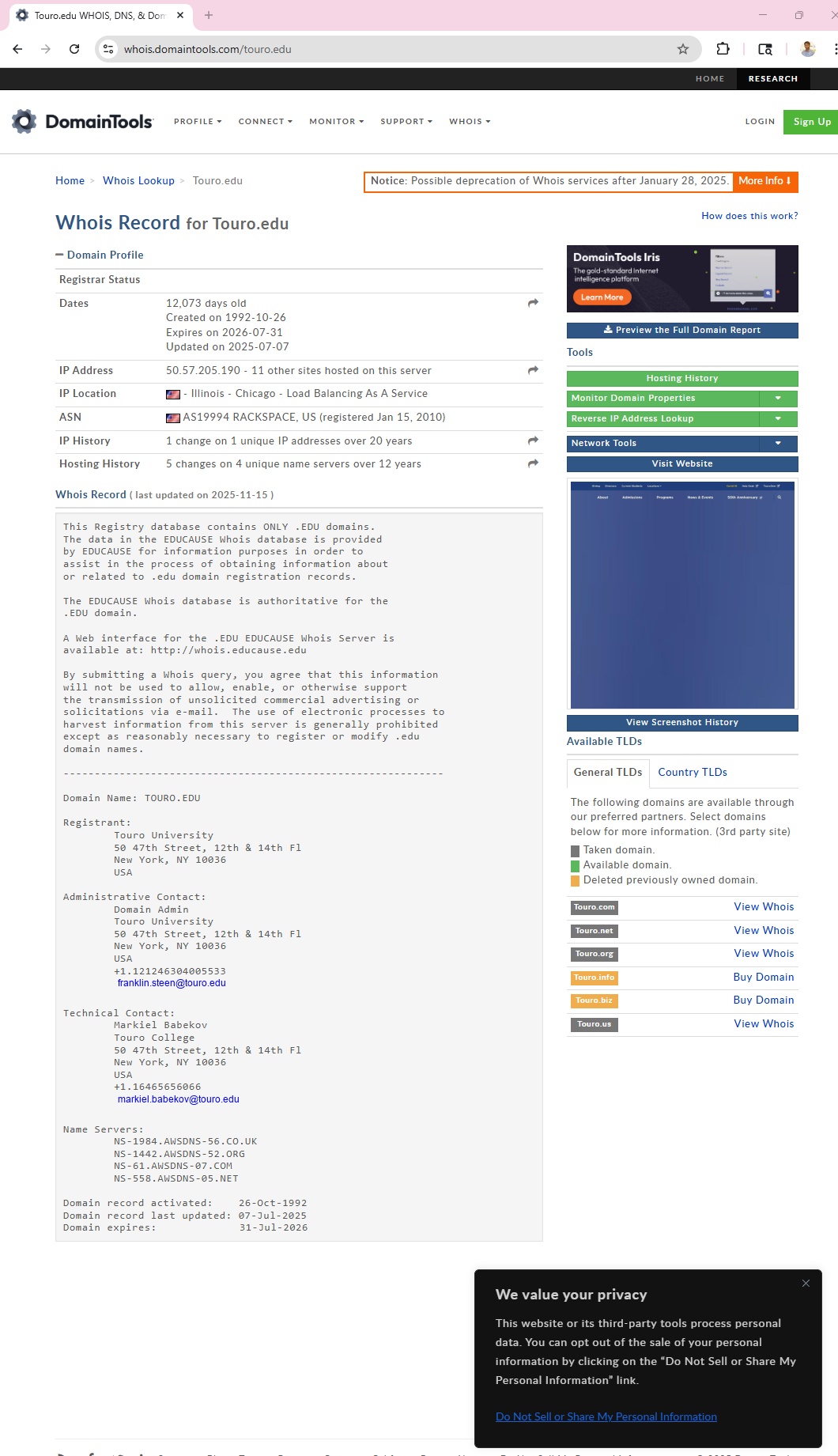
Differences

* DomainTools displayed information in a cleaner, more organized format.
* DomainTools included additional context such as hosting provider details and risk information.
* Domain Dossier provided extra technical functions like traceroute and service scan options.
* Some domains (e.g., Amazon) showed fewer details in DomainTools due to privacy masking.

Conclusion of Comparison

DomainTools is better for clean, business-oriented Whois viewing.  
Domain Dossier is better for technical reconnaissance.





**Step 7 – Browser Left Open**

I kept the browser tabs open for the next activity as instructed.

**Conclusion**

This activity demonstrated how much information an attacker or penetration tester can gather using simple Whois tools. Different organizations reveal different levels of information:

* **Universities** (Touro, Harvard) expose more contact information.
* **Large corporations** (Amazon) use heavy privacy protection.

Understanding these differences helps assess potential social engineering risks and network exposure.

**Activity 4-2: Identifying Company Email Accounts**

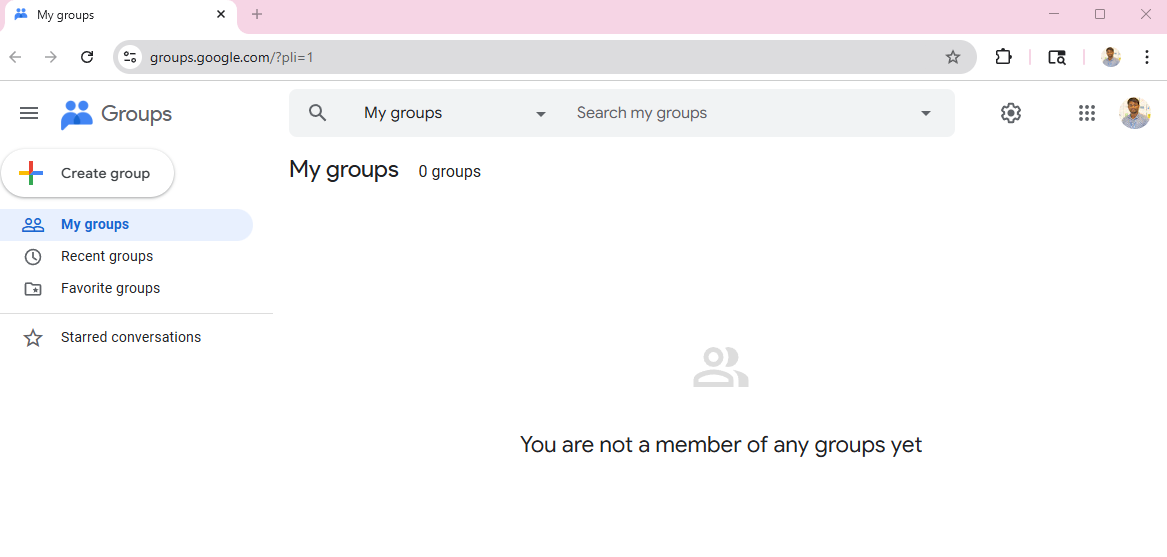
**Objective**

The purpose of this activity is to understand how attackers or security testers can identify company employees’ email accounts by searching public newsgroups on Google Groups. These emails may reveal personal information, internal technical problems, security configurations, or discussions that can become security risks.

Step 1 – Open Google Groups

I opened my web browser and click to:

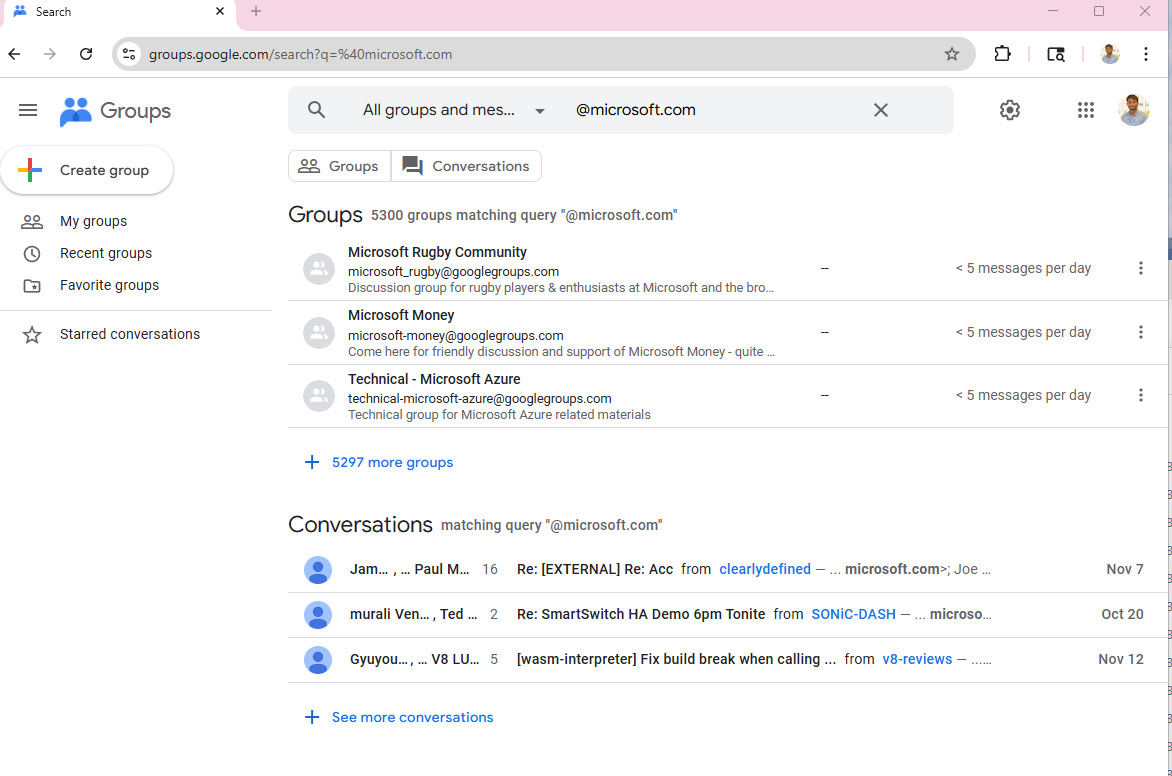
<https://groups.google.com>



**Step 2 – Search for @microsoft.com**

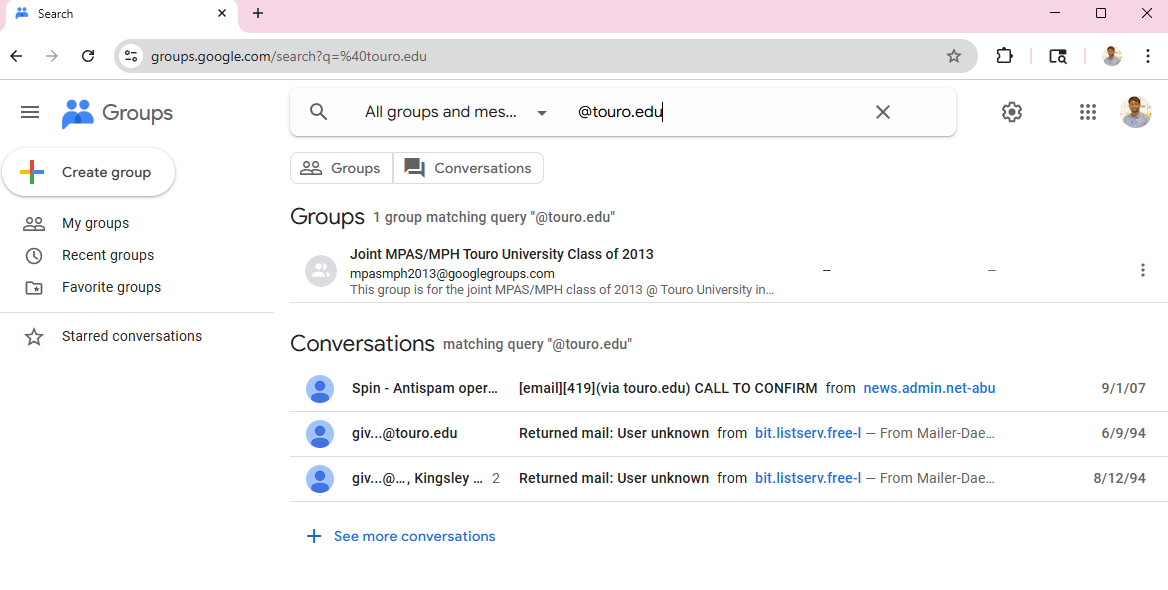
In the search bar, I selected All groups and messages, then typed:

@microsoft.com



This search shows messages posted using Microsoft employee email addresses.

I also tested a search for **touro.edu**, and it showed several discussions from users connected to my university, which confirmed that the same security risks apply to educational institutions.

****

This search shows messages posted using Microsoft employee email addresses.

**Step 3 – Review the Microsoft Results**

I scrolled through the results and looked for posts with **“Re:”**, because these are usually replies from real employees.

**Observations:**

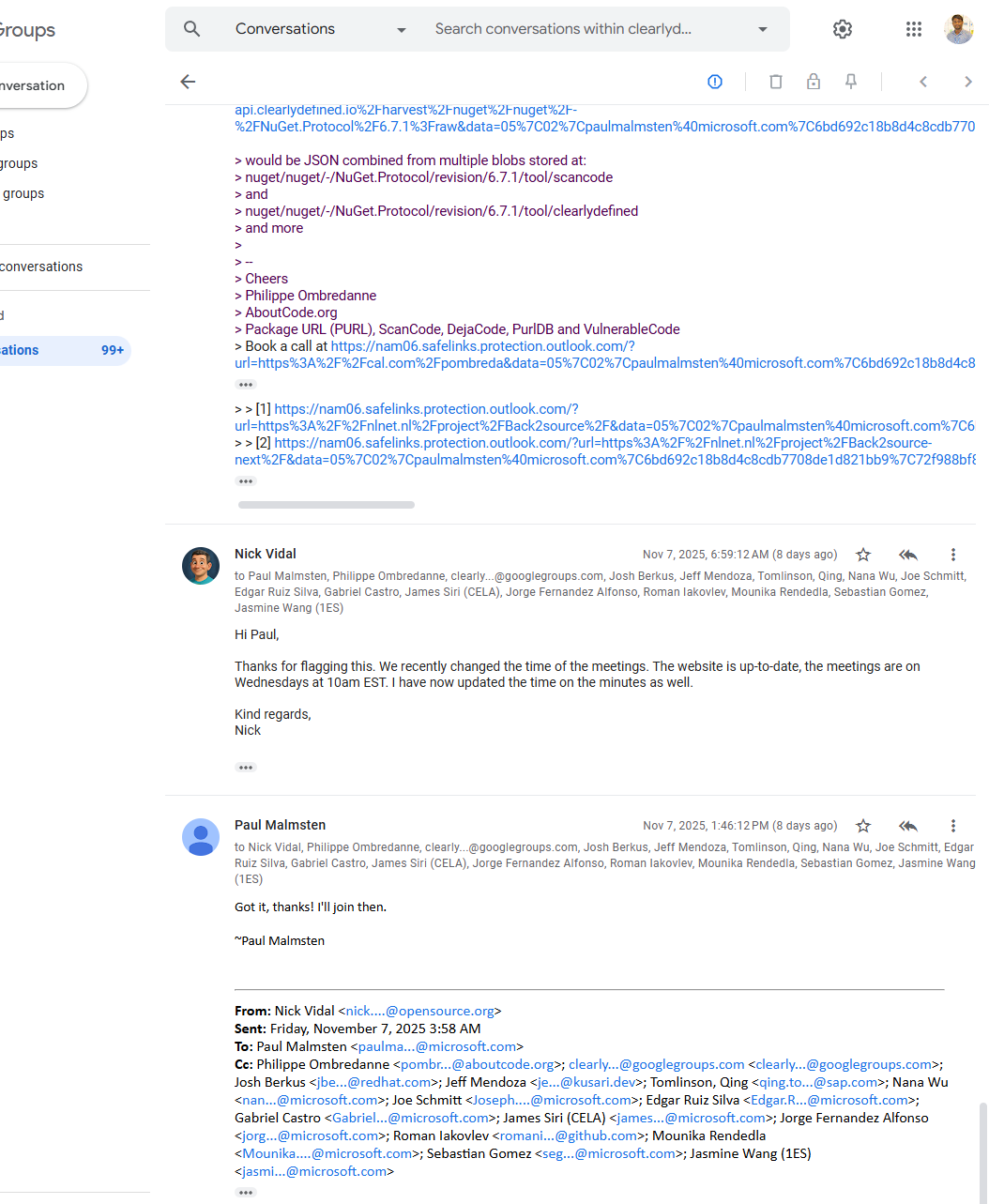
* Several posts were made using **official Microsoft email addresses**.
* Many replies contained **technical discussions**, troubleshooting steps, and guidance.
* Some posts revealed details about:
  + Internal issues with Microsoft products
  + Interaction between employees and customers
  + Technical environments used inside Microsoft

**Security Risks:**

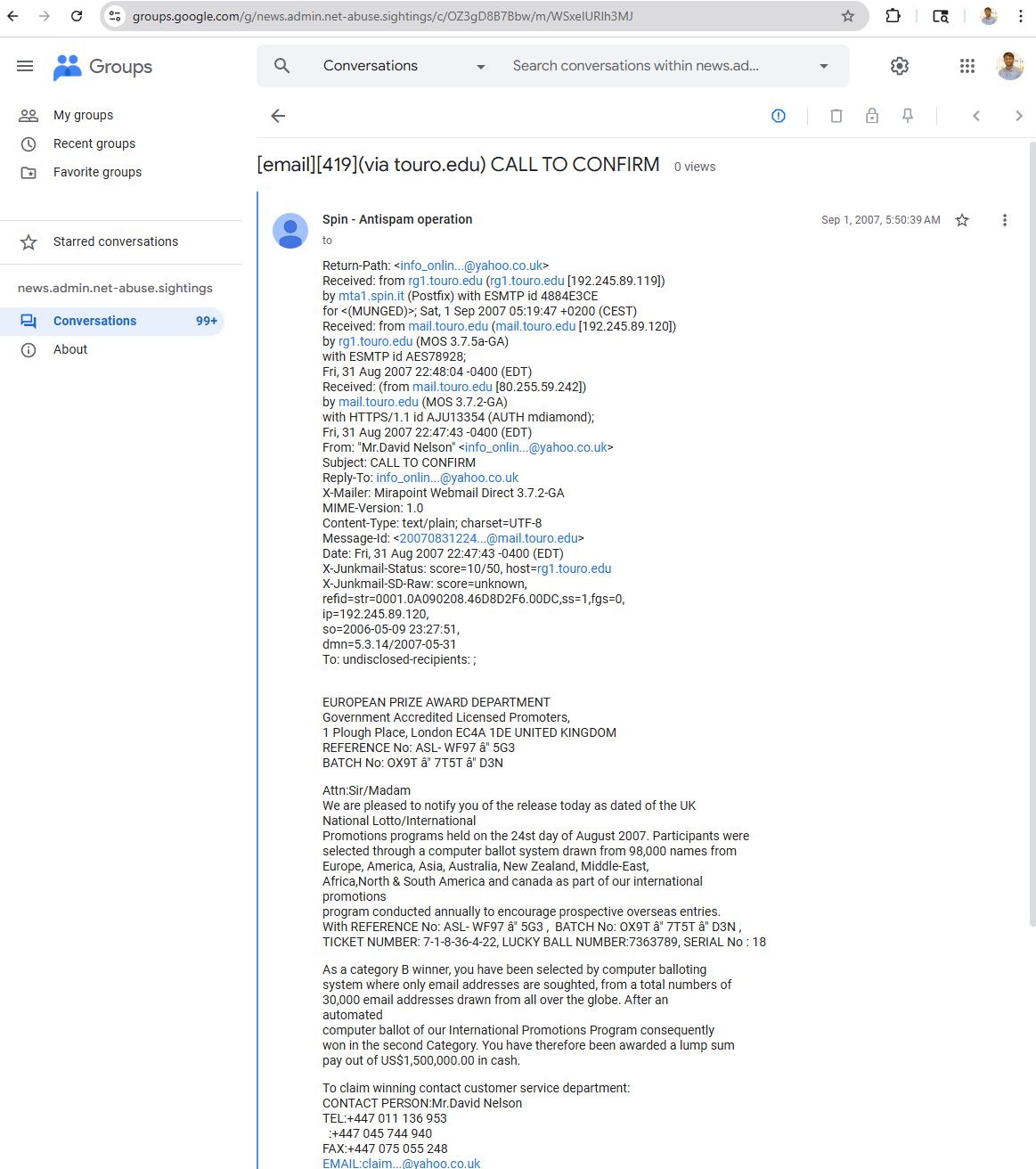
An attacker could use this information to:

* Identify employees and departments
* Determine what systems or products they use internally
* Conduct targeted phishing (especially replying to the same thread)
* Pretend to be part of the discussion
* Gather intelligence on internal problems

I opened one “Re:” message and viewed the full conversation. This clearly demonstrated how easy it is to trace employees through their official email usage.

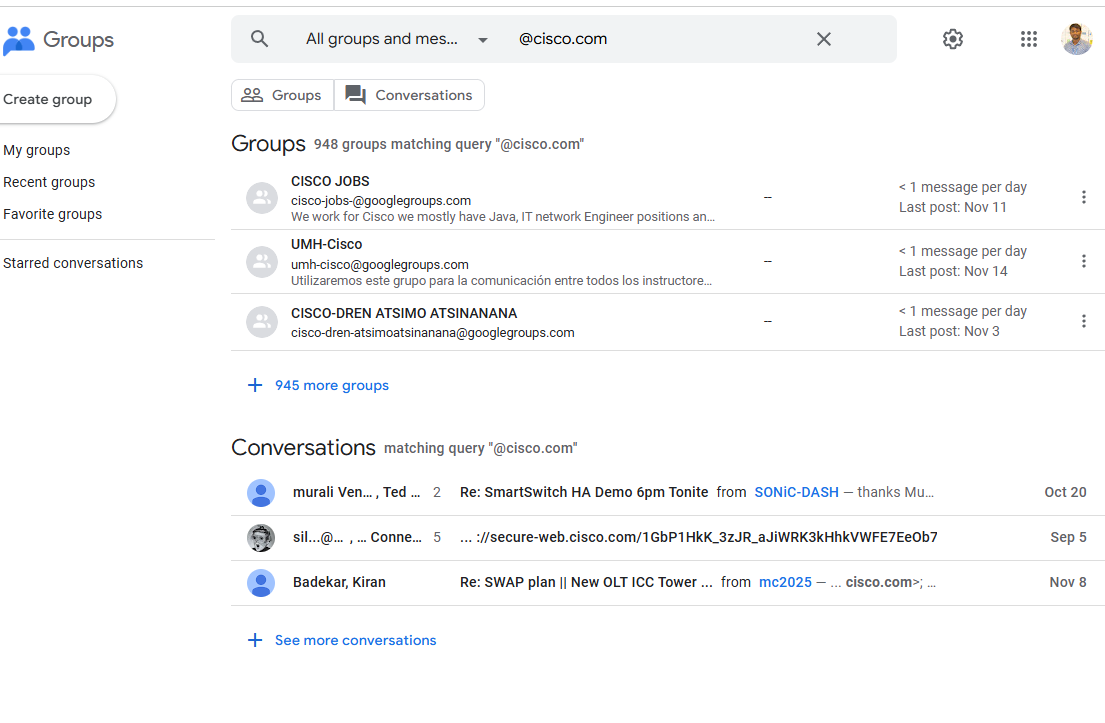
****

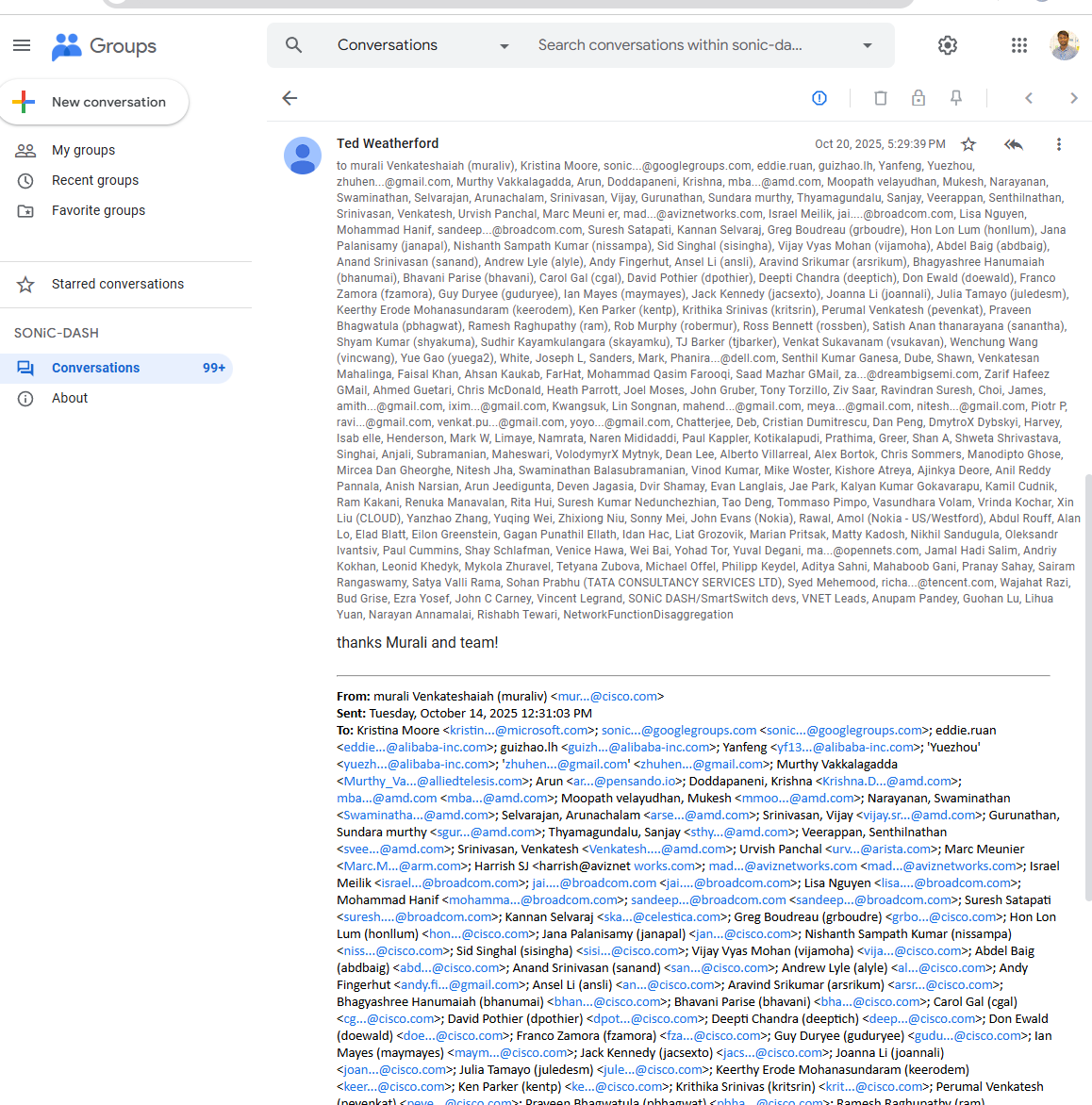
I also searched my own university domain touro.edu and found similar discussions, showing the same risks apply to educational institutions.

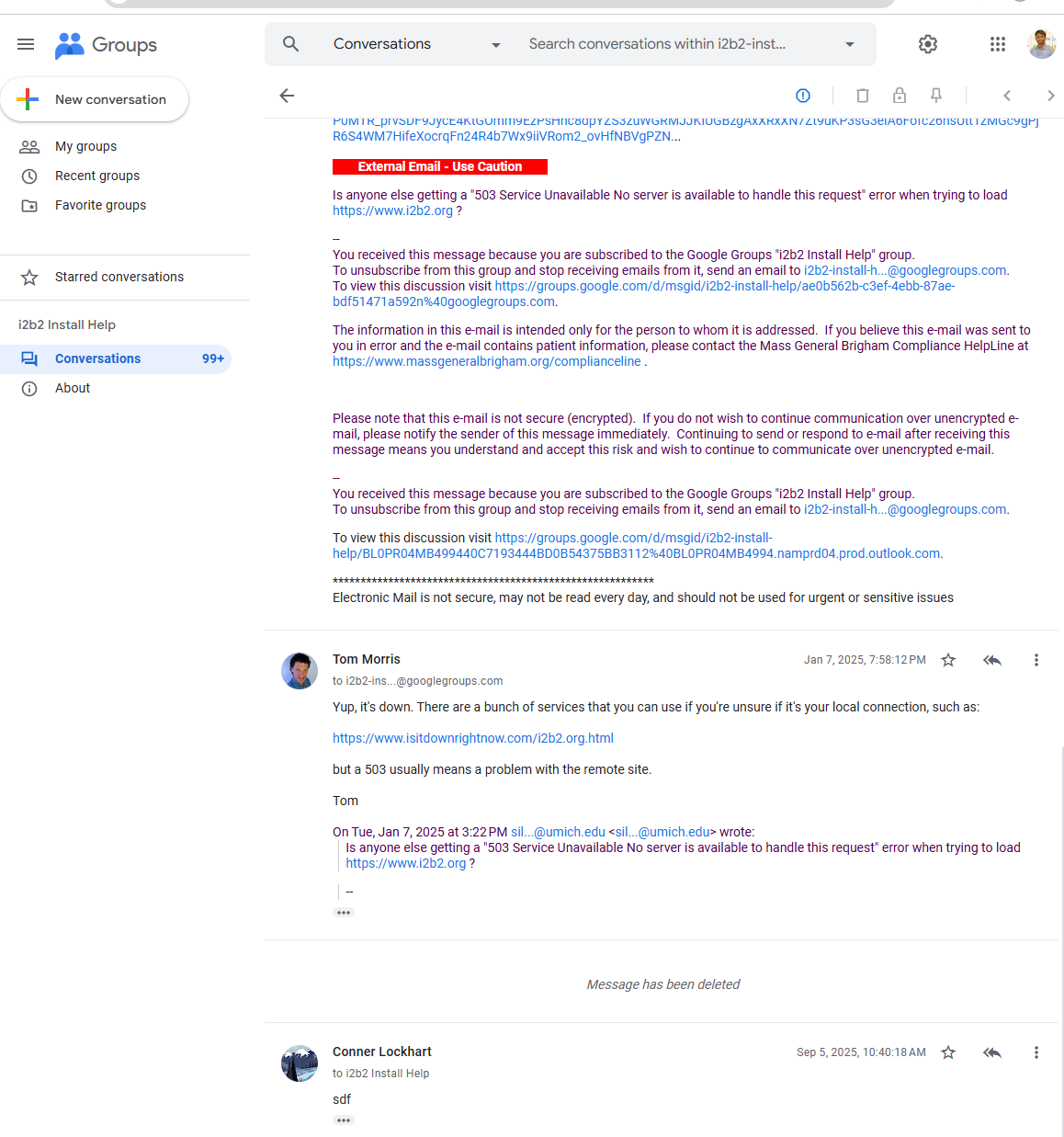
****

**Step 4 – Search for @cisco.com**

Next, I searched for:  
**@cisco.com**

A large number of results appeared, mostly from customers and IT professionals asking questions about Cisco networking products.





**Step 5 – Review the Cisco Results**

Observations:

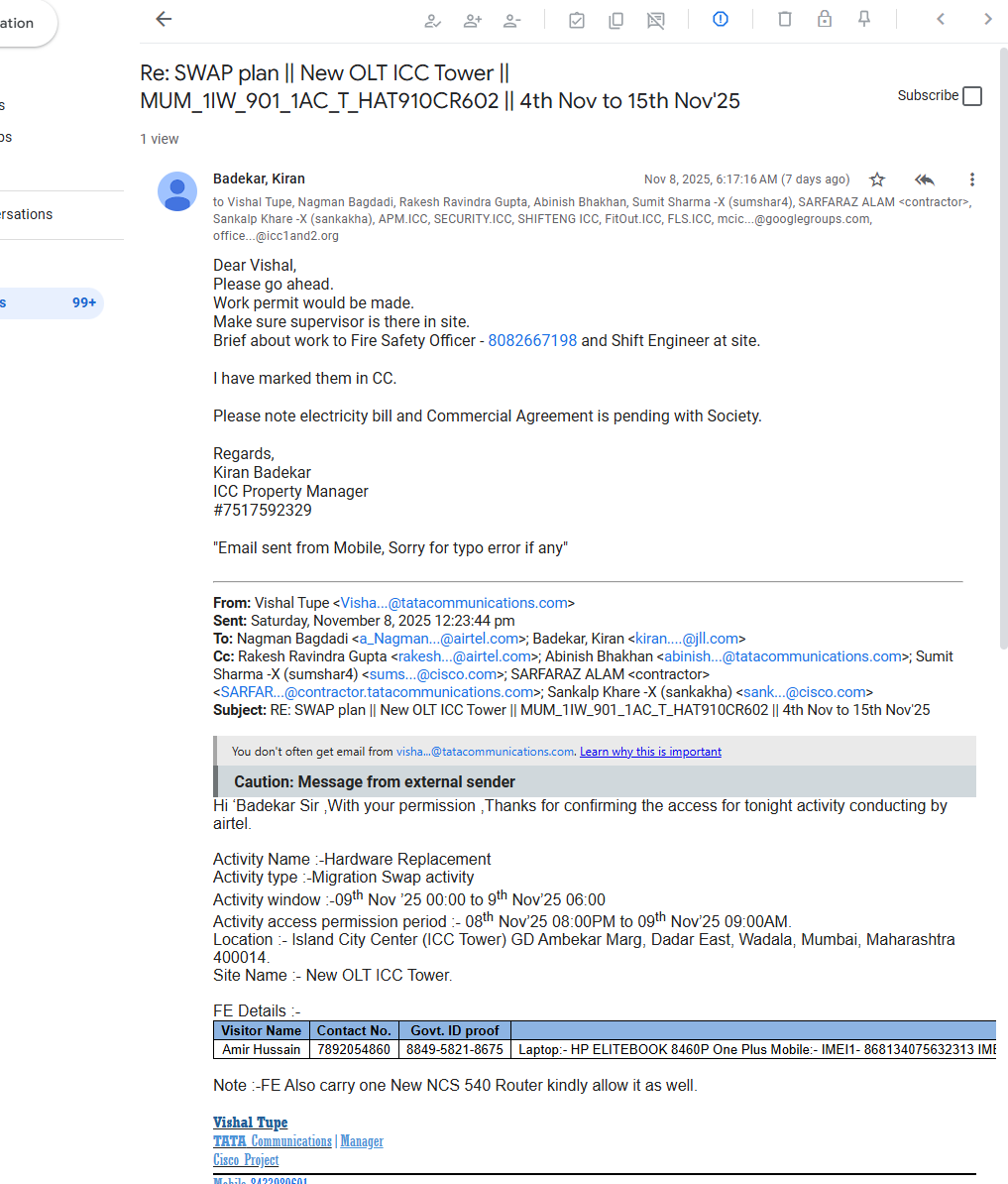
* Many postings contained technical questions about:
  + Cisco firewalls
  + Routers
  + VPN problems
  + Network performance issues
* Several users revealed:
  + Network device configurations
  + Firmware versions
  + Internal errors
  + Names of internal devices
  + Security misconfigurations

**Security Risks for Attackers**

Attackers could use this information to:

* Identify outdated or vulnerable Cisco equipment
* Target companies using specific Cisco hardware
* Send phishing emails pretending to be “Cisco Support”
* Understand customer networks and troubleshoot points
* Profile IT employees who manage sensitive systems

This type of information is extremely valuable for both penetration testers (ethical testing) and malicious hackers.



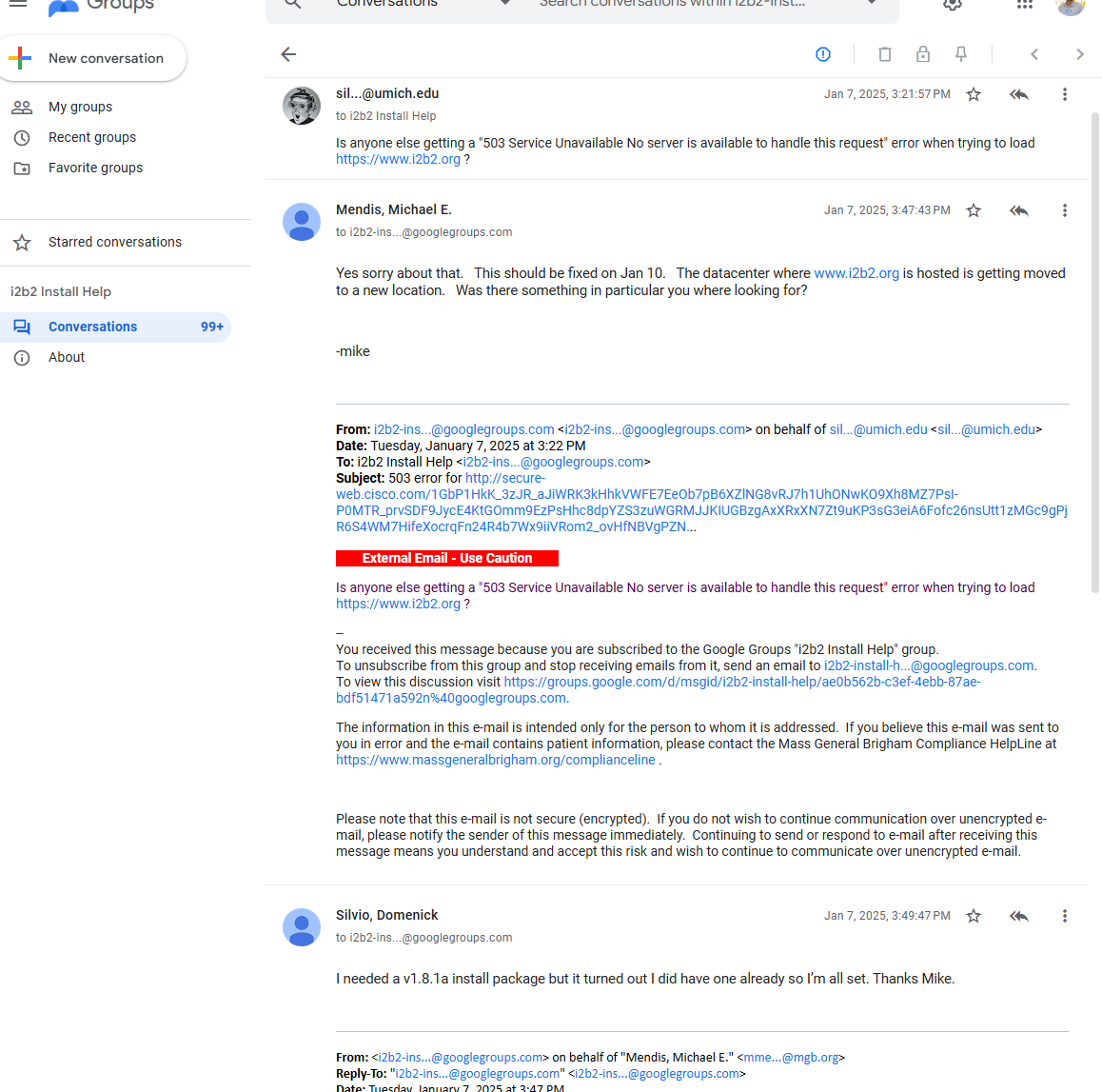
**Step 6 – Determine Useful Information**

**Findings:**

* Posts included employee names, email formats, and department information.
* Users shared internal network issues publicly, without realizing the risk.
* Email patterns like firstname.lastname@company.com were easy to identify.
* IT staff often revealed technical details that could expose vulnerabilities.

**Age of Returned Links**

* Many results were **very old**, often from the early 2000s.
* This proves that email discussions remain online for decades, and attackers can still use outdated data to map company structures.



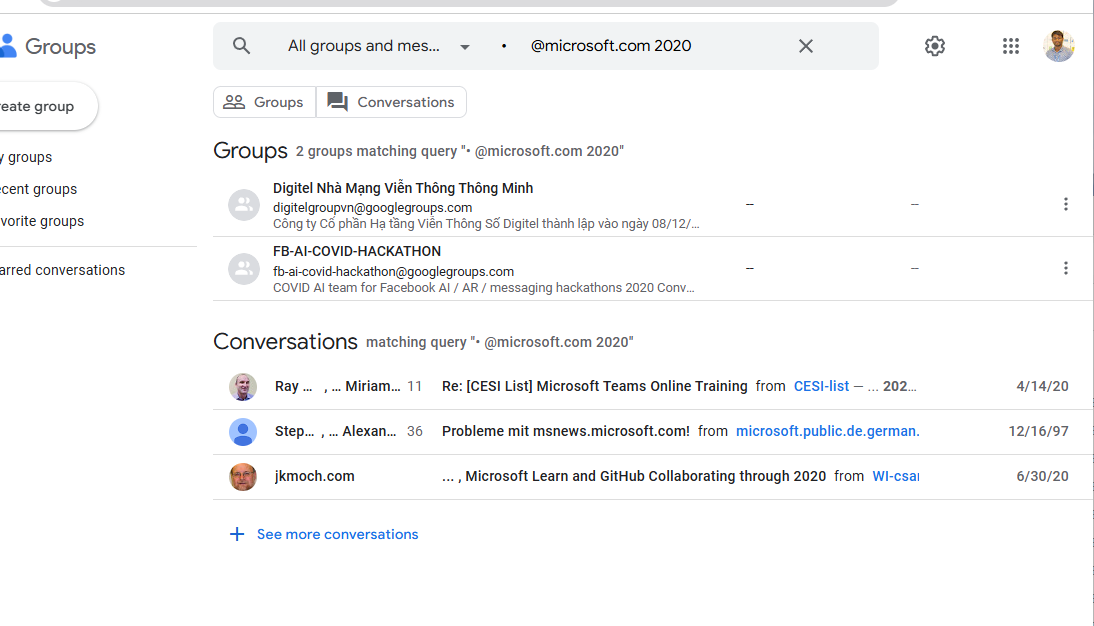
**Step 7 – Search for Recent Posts (2020 and 2021)**

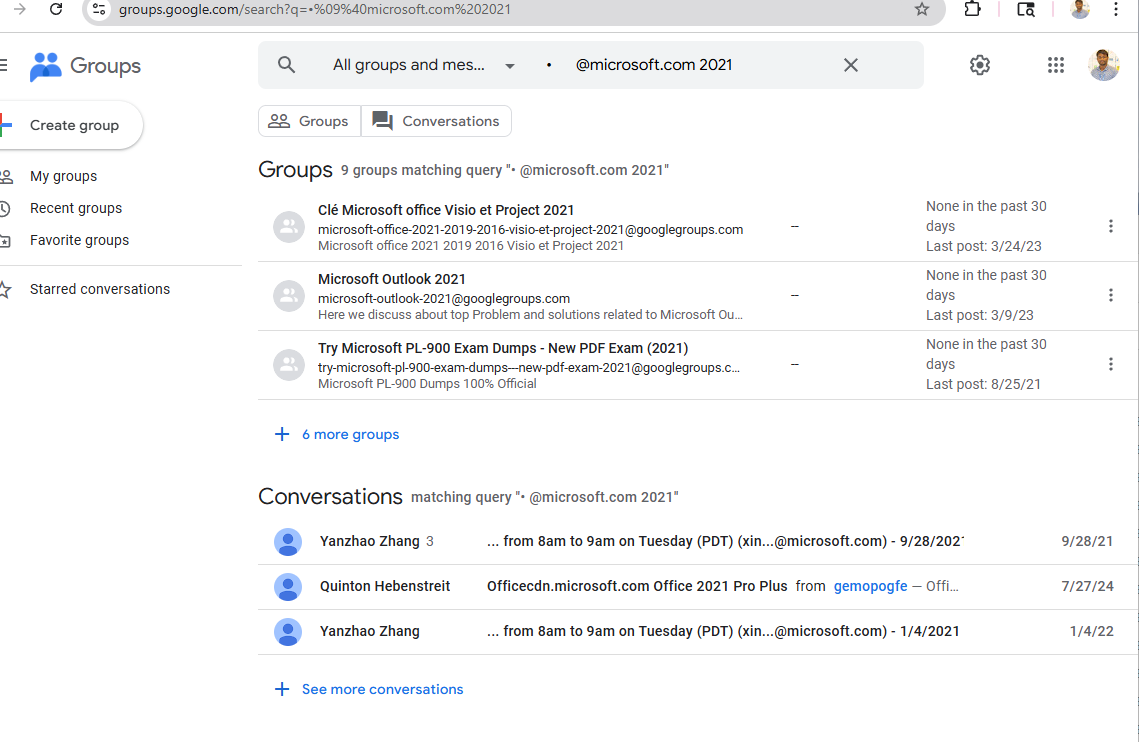
To view more recent discussions, I used:

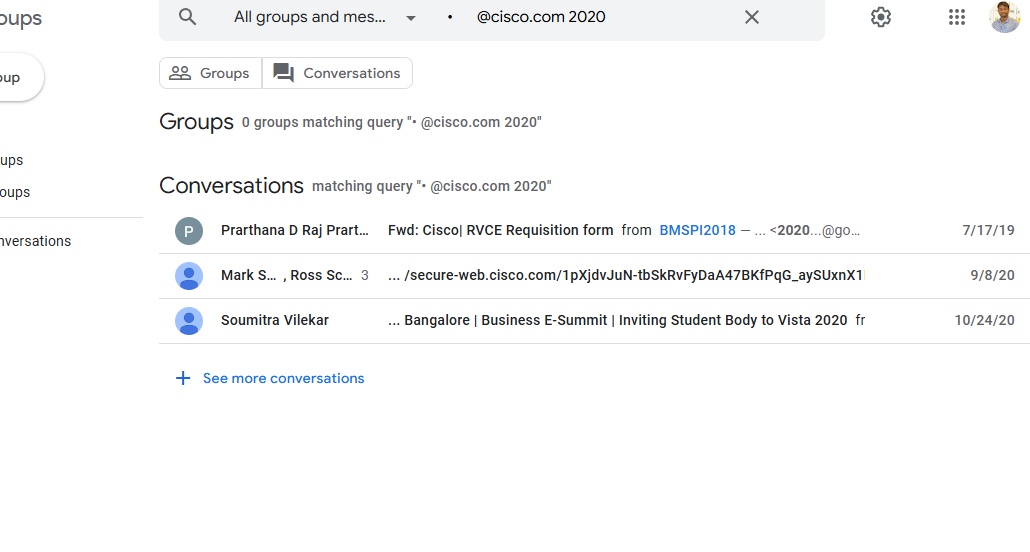
* @microsoft.com "2020"
* @microsoft.com "2021"
* @cisco.com "2020"
* @cisco.com "2021"

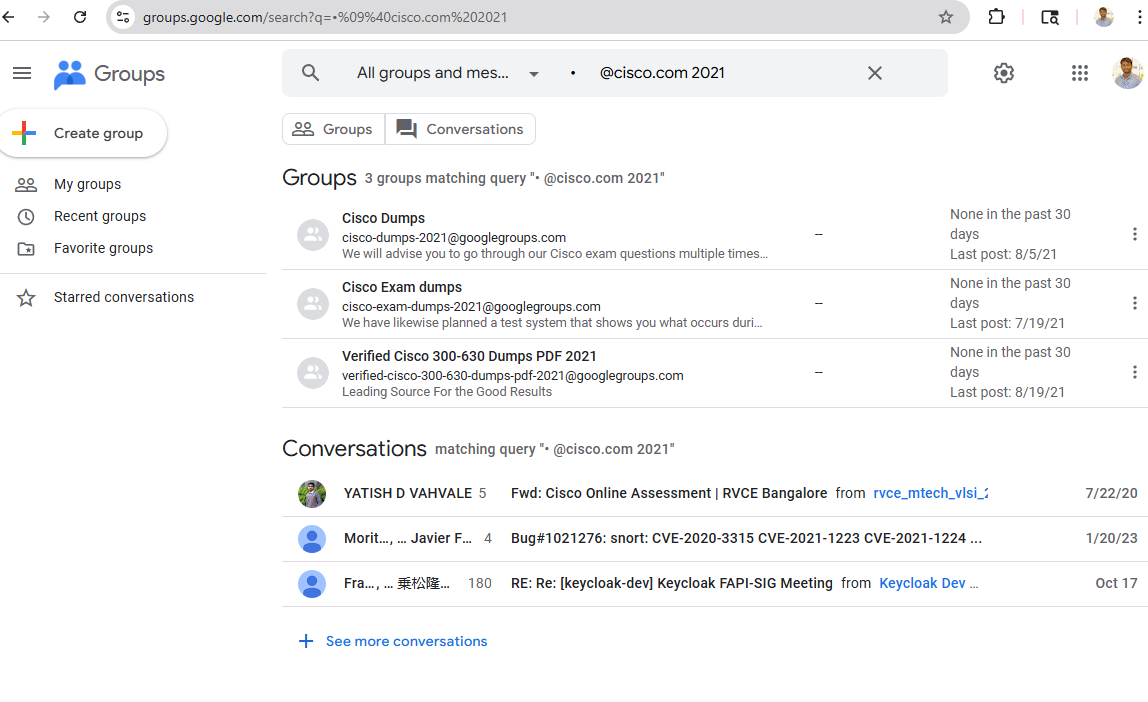
**Results:**

* Only a small number of recent posts appeared.
* Companies now discourage employees from using corporate emails publicly.
* Modern platforms hide or mask email addresses for privacy.

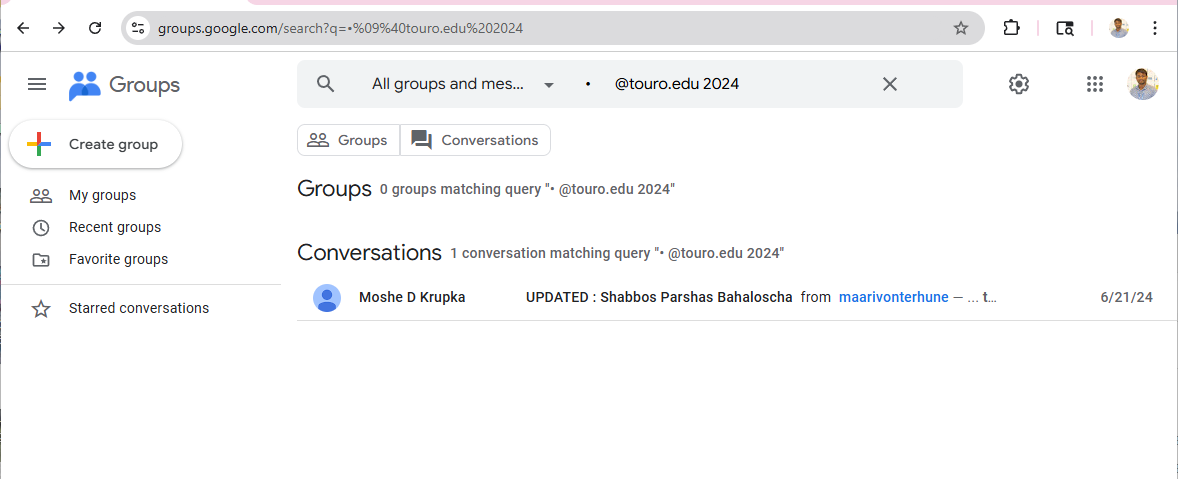
****

****

****

****

I also searched **touro.edu** again with year filters and saw far fewer results compared to older years.

****

**Conclusion**

This activity demonstrates how easy it is for attackers to gather email accounts and sensitive technical information from public forums. Employees often share details about network issues, devices, configurations, and internal processes sometimes without realizing the security risk.

Google Groups searches can reveal:

* Names and emails of employees
* Company email structures
* Technical problems inside organizations
* Customer support interactions
* Outdated equipment or systems

A penetration tester can use this information for footprinting and social engineering, while a malicious attacker could use it to plan targeted attacks.

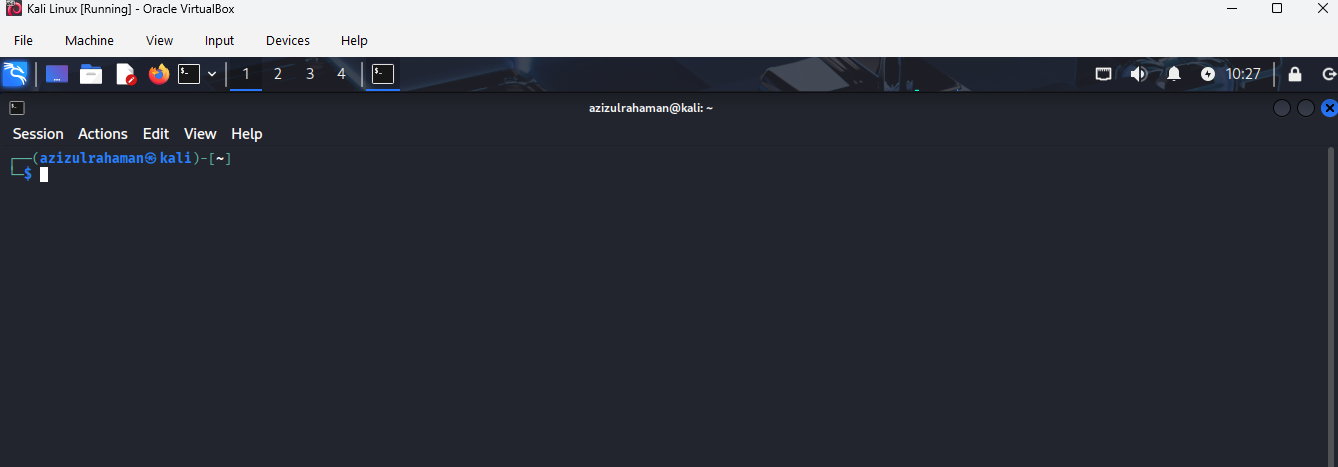
**Activity 4-3: Using HTTP Methods**

**Objective**

The purpose of this activity is to use basic HTTP methods with the nc (Netcat) command on port 80 to gather information from a web server. By analyzing the server’s responses, a security tester can determine allowed HTTP methods, server behavior, and potential vulnerabilities.

**Step 1 – Start Kali Linux and Run Netcat**

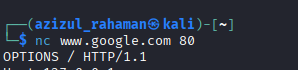
I opened my virtual kali linux machin then I opened terminal window. At the command prompt, I typed: nc www.google.com 80



This creates a connection to Google.com on port **80** (HTTP port).  
Most modern servers use **port 443**, and port 80 is less secure.

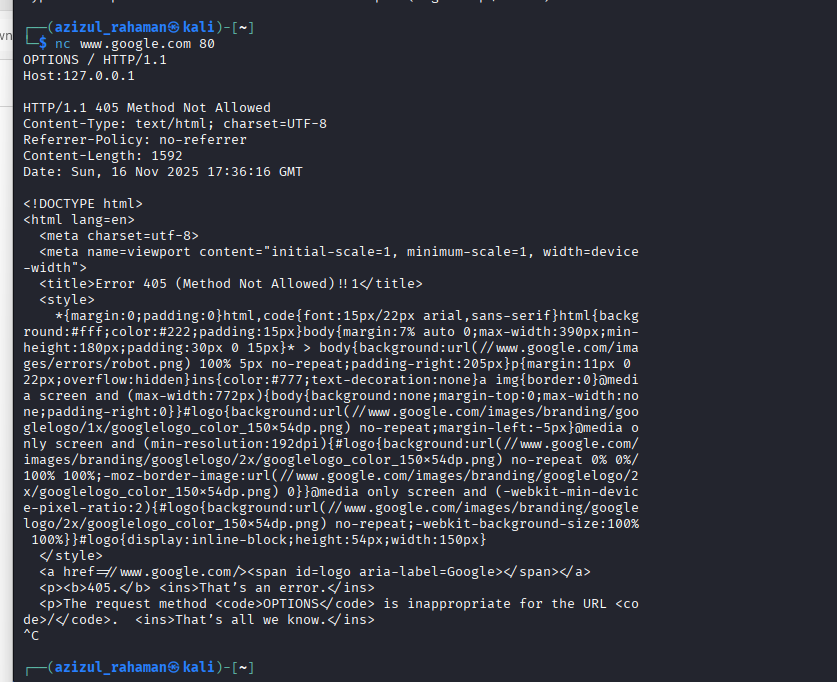
**Step 2 – Send OPTIONS Method**

In the Netcat session, I typed commends: OPTIONS / HTTP/1.1



**Step 3 – Send HOST Header**

I typed: HOST:127.0.0.1 and Then I pressed Enter twice.



This sends an **HTTP OPTIONS request**, which asks the server to list which HTTP methods it supports.

Google’s server responded, but the OPTIONS method was not allowed (as shown in Figure 4-8).  
However, the response still displayed the **allowed methods**, such as:

* GET
* HEAD
* POST

Google disables sensitive methods to prevent information exposure.

**Step 4 – What Information Is Useful to a Security Tester?**

Based on the OPTIONS results

**Useful Information from the nc Output:**

* Which HTTP methods the server allows
* Confirmation that **port 80 is open**
* How the server handles unsupported methods
* Basic server behavior (e.g., how it responds to malformed requests)

**Why This Matters:**

* If unsafe methods like **PUT** or **DELETE** were enabled, the server could be vulnerable.
* Attackers could upload files, delete files, or manipulate requests.
* Understanding allowed methods helps testers decide what to probe next.

**Other Options Commonly Available:**

* **GET** – Retrieve webpage contents
* **HEAD** – Retrieve only headers
* **POST** – Submit form or application data

Google allows only the safe and standard methods.

**Step 5 – Reconnect to Google.com**

I ran the Netcat command again: nc www.google.com 80

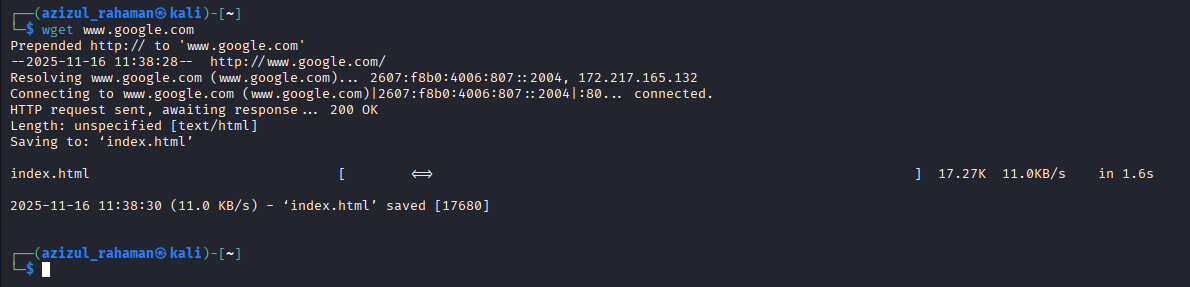


This opened a new connection for the next HTTP test.

**Step 6 – Send HEAD Request**

I typed: HEAD / HTTP/1.0

Then pressed Enter twice.



**Results (similar to Figure 4-9):**

The server returned headers only, including:

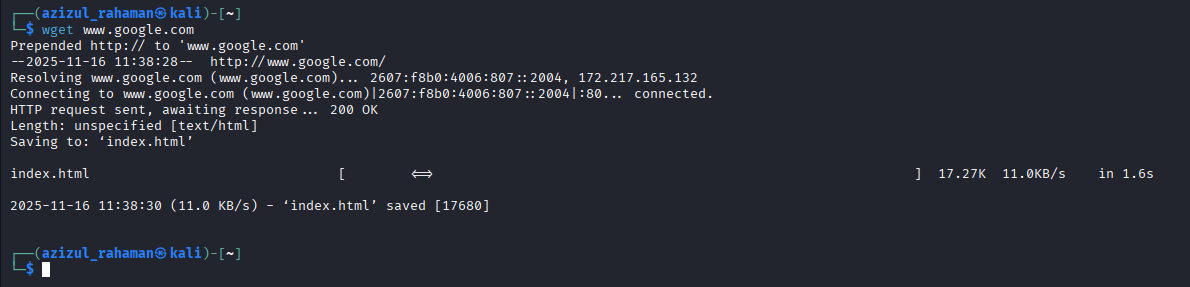
* **HTTP/1.0 200 OK**
* Date
* Server type
* Content-Type
* Content-Length: 0
* Connection: close

**Why HEAD Is Useful:**

* Reveals basic server info without downloading the webpage
* Helps fingerprint the OS or web server software
* Shows how the server handles connections
* Faster and safer than GET because it does not return the page body

**Step 7 – Use WGET to Download the Index Page**

I typed: wget [www.google.com](http://www.google.com)



This command downloaded the **index.html** file from Google.com.

**Observations (similar to Figure 4-10):**

* The file downloaded successfully.
* The HTML page can be opened in a text editor to analyze:
  + HTML structure
  + Metadata
  + Scripts
  + Redirects
  + Tracking details

This allows deeper inspection of how the website is built.

**Step 8 – Close Terminal**

I exited the terminal and logged off Kali Linux.