INFORMATION-GATHERING USING GOOGLE DORKING AND NSLOOKUP COMMAND

PROJECT MEMBERS:

- KRYSTIAN CHRUPEK (LINKEDIN)
- KABIR ONI (LINKEDIN)
- MAWSUMI HAQUE (LINKEDIN)

WARNING!

All techniques demonstrated in this project (Google dorking, nslookup and other OSINT-related tools) are used **for educational and research purposes only!**

This project does not intend to:

- gain unauthorized access,
- violate privacy,
- or misuse any gathered information.

These tools should be used **only on systems and resources you own or have explicit permission to test.**
Any other use may be **illegal** and is done **at the user's own responsibility.**

PROJECT OVERVIEW

- Introduction
- Requirements
- Information gatehring techniques
- Script and execution + the output
- Conclusion
- References

ABOUT THE PROJECT

The goal of this project is to create a single, simple **Python** script that demonstrates and automates two fundamental information-gathering techniques (OSINT) used in reconnaissance: **Google dorking** and **DNS queries using nslookup**. The project is purely **educational** — it shows how to lawfully and responsibly collect publicly available data to analyze an attack surface and verify exposed assets.

Concretely, the script will:

- perform **Google dork** queries (examples of operators used: cache:, allintext:, allinurl:, filetype:, inurl:, site:, related:) to locate publicly accessible pages, files, and information related to the target scope;
- run DNS queries (nslookup / DNS library in Python) to fetch key DNS records such as **MX (Mail Exchange)**, **NS (Name Server)**, **A (IPv4)** and **SOA (Start of Authority)**, enabling mapping of mail infrastructure and domain name configuration.

The script's output will be presented as a clear report (screenshots/sample outputs) showing what information each technique can reveal and what security/administration conclusions can be drawn. The project emphasizes ethical usage: all demonstrations are performed only on public resources or in controlled test environments, and no unauthorized access or misuse of collected data is intended.

REQUIREMENTS

- Python (version 3.9 or newer) – to run the script
- Windows 10/11 (Script tested o Windows)
- Internet access
- Python standard library: os (used to execute system commands from the script)

INFORMATION GATHERING TECHNIQUES

- Google dorking

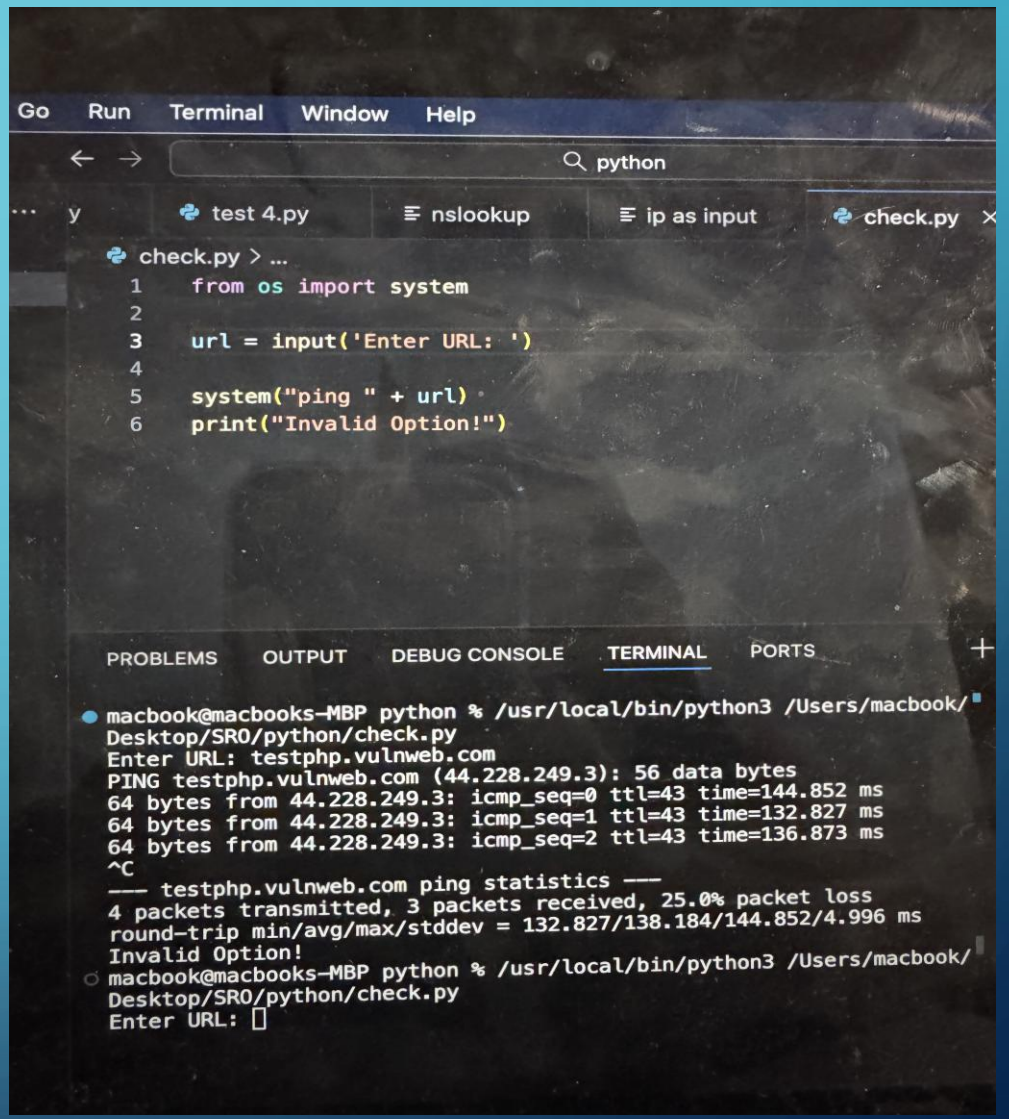
Using advanced Google search operators (e.g. site:, filetype:, inurl:, intitle:, cache:) to discover publicly available but not easily visible resources related to a target – such as exposed documents, login pages, backups, or pages indexed by mistake. This helps map what information about the target is already on the internet.

- DNS queries with nslookup

Querying DNS servers to retrieve key domain records (A – IPv4 address, MX – mail servers, NS – name servers, SOA – domain authority info). This allows us to understand how the domain is configured, which servers handle mail/traffic, and to identify parts of the target's infrastructure.

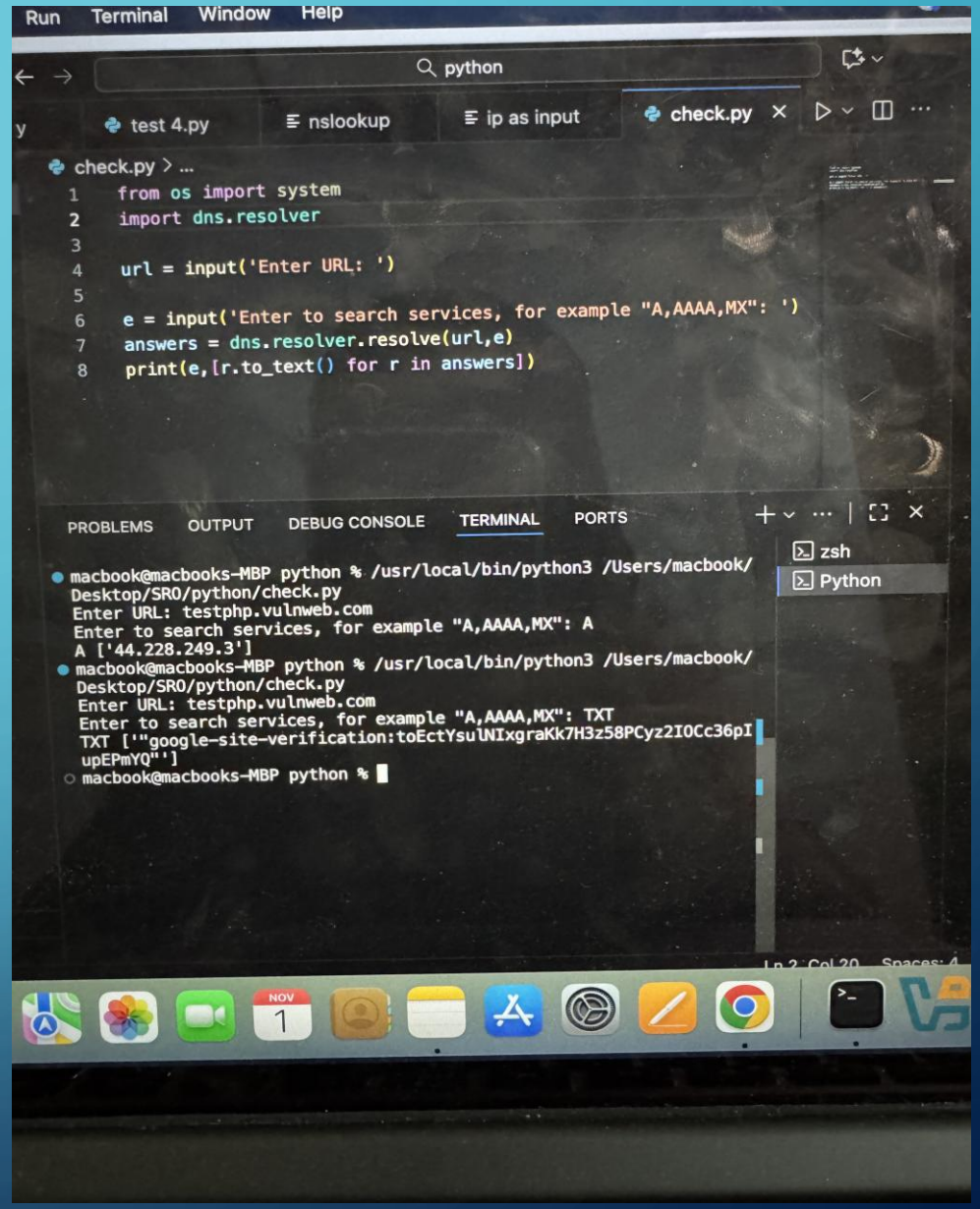
ICMP – PING COMMAND

- Scrypt execute the ICMP echo requests to the url that is taken from user, using the ping function from the system() method from os module, where the os module is a built-in Python module that provides functions for interacting with the operating system in a portable manner and system() method in Python is a powerful utility that allows you to execute shell commands directly from a Python script.

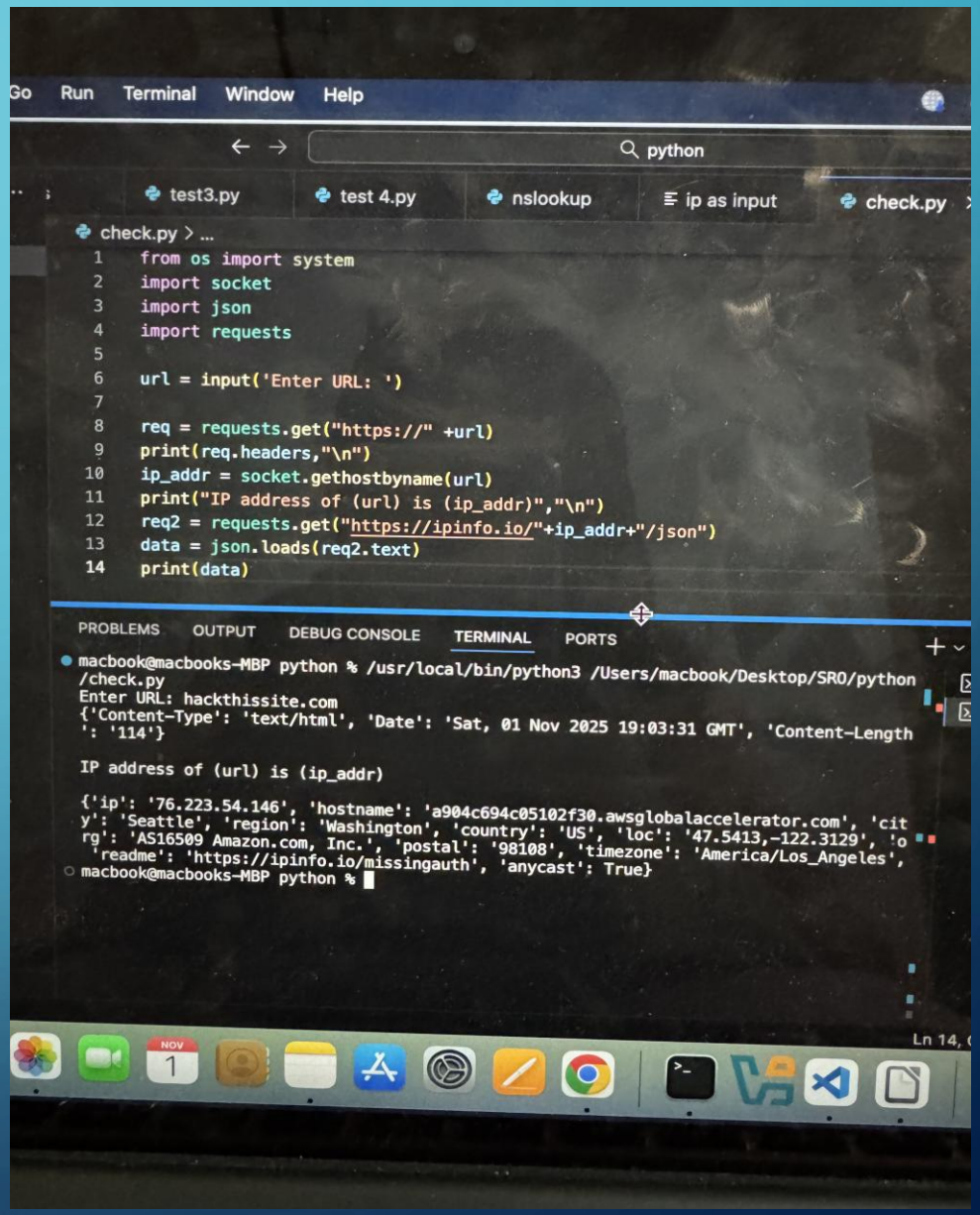


DNS Resolver

This particular part of the script using dns.resolver library is converting human-friendly domain names into machine-readable IP addresses ,so that web browsers can connect to servers. Then **url** is taking the input and **e** is taking different DNS Records as input from user and finally store this in '**answers**' . At the end it is printing whichever services user want to check , like , A, AAAA, MX , TXT ect.

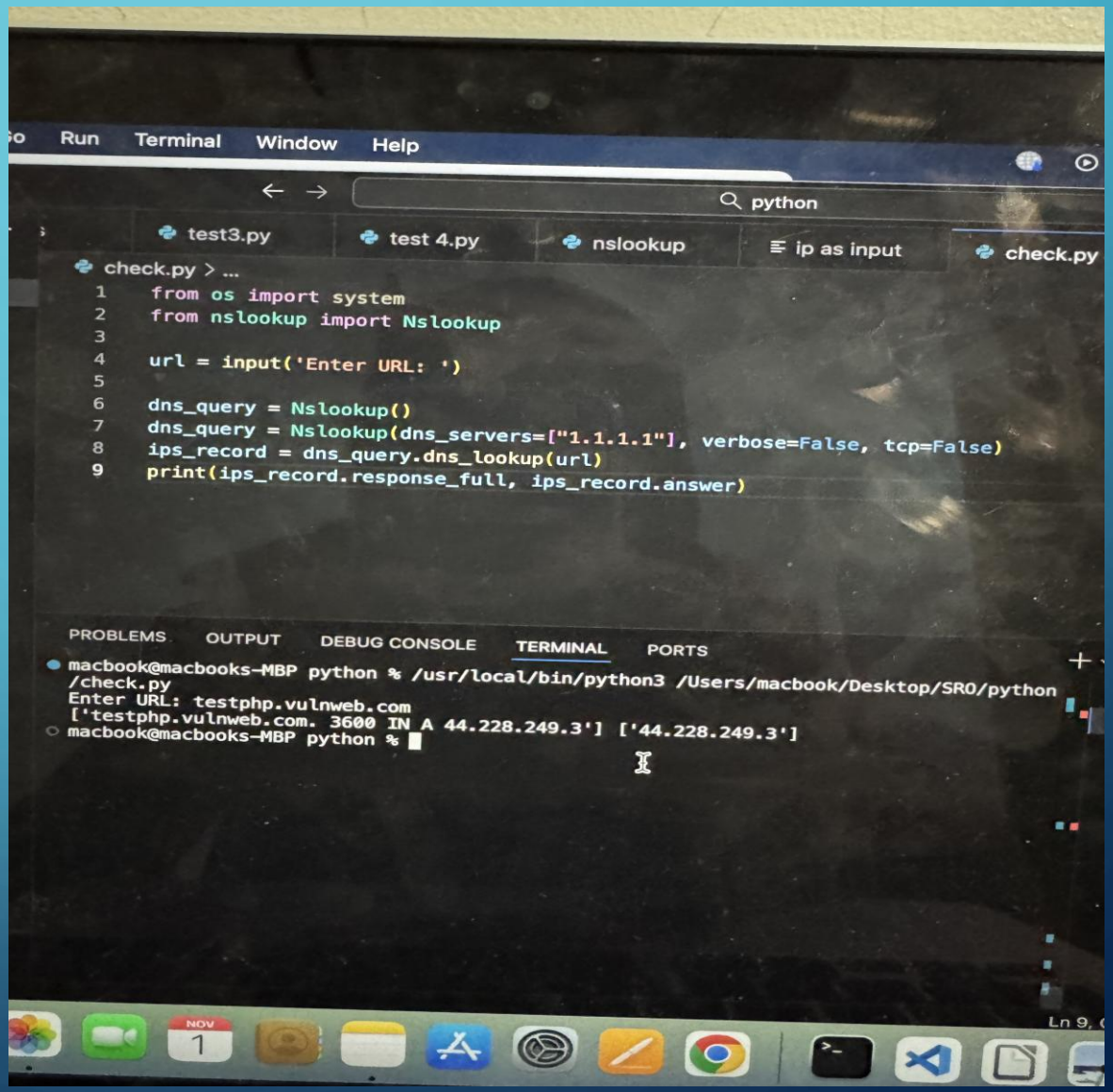


This particular part of the script is gathering information from inverting url in IP address by using **socket** library which connect directly with the machine. Here **requests** library used for taking the url as http:// , which is connect to the website. After converting the url to IP address by using ip_addr, req2 is using the information from ipinfo.io . Here json library used for the function Jason which is like a javascript to organise the data by calling Jason.loads.

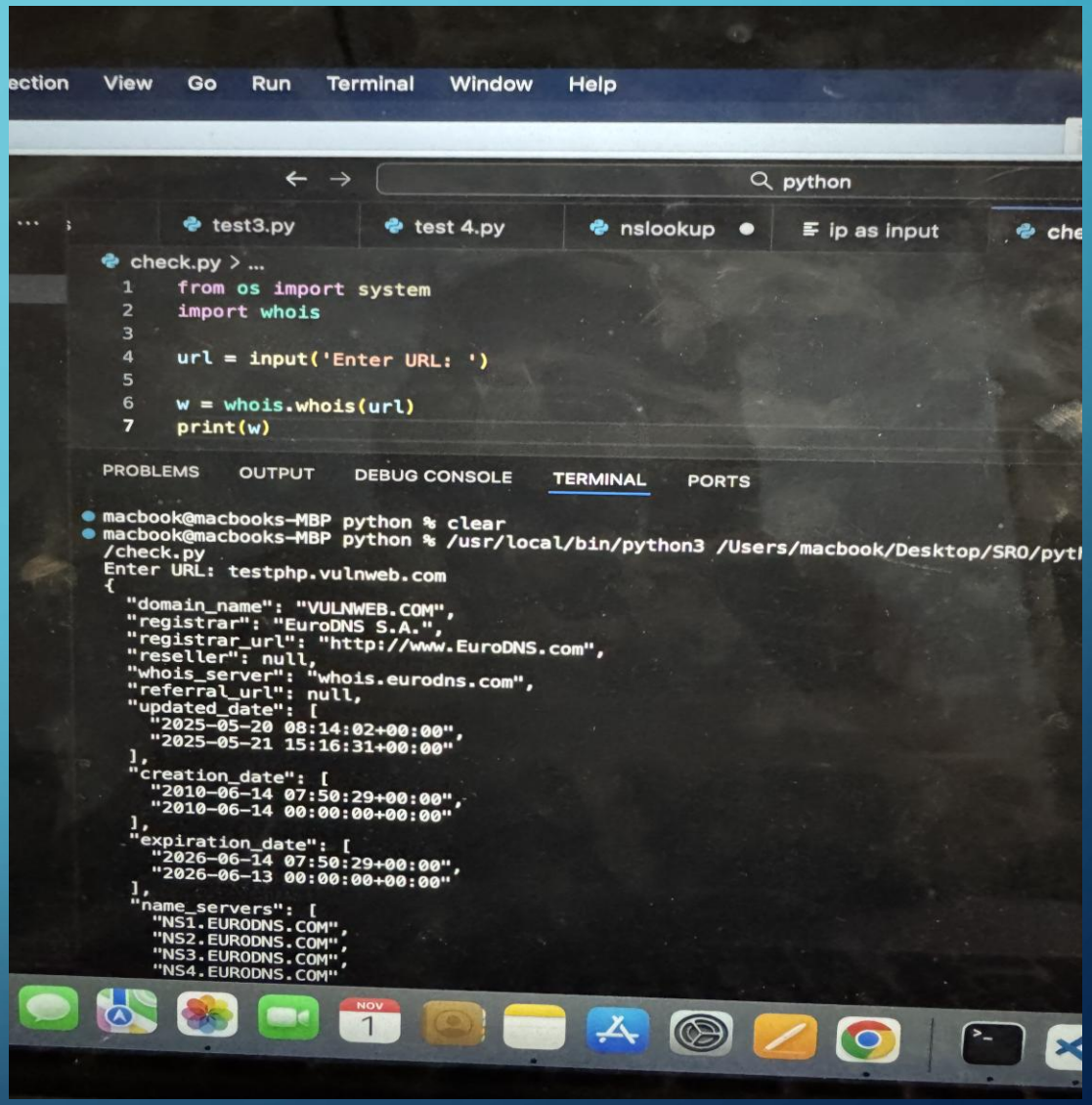


NSLookup

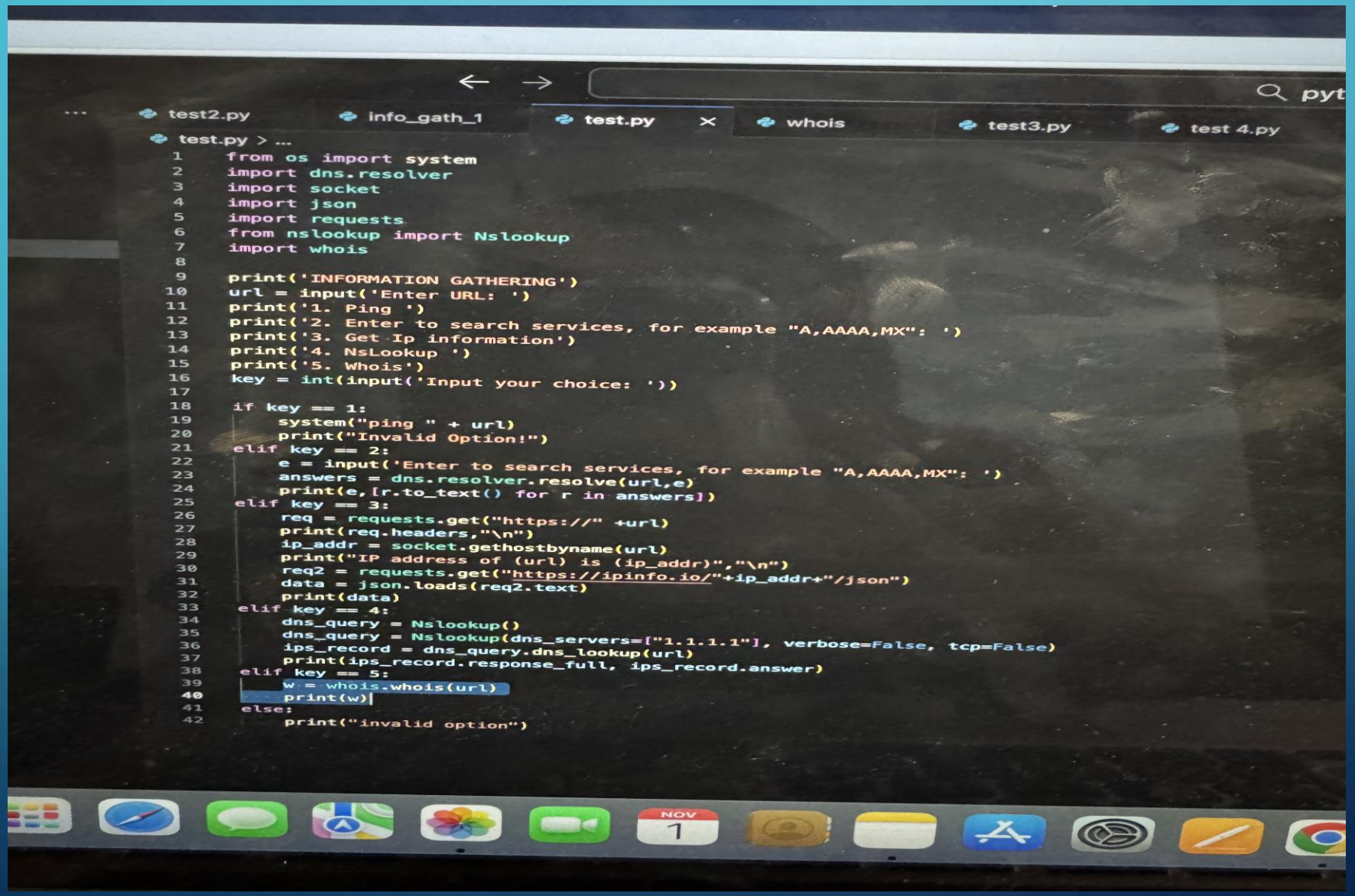
Using Nslookup function from nslookup library to give the information exactly like how we can gather information from cmd/terminal command. Here, ns_query = Nslookup() Initializes Nslookup, dns_query = Nslookup(dns_servers=["1.1.1.1"], verbose=False, tcp=False) this line arguments for setting custom dns servers (defaults to system DNS),verbosity (default: True) and using TCP instead of UDP (default: False). The line ips_record = dns_query.dns_lookup(domain) gets the dns information from domain, and soa_record = dns_query.soa_lookup(domain) gets the sea information from domain.



This particular script will produce parsed WHOIS data for a given domain,Query a WHOIS server directly instead of going through an intermediate web service like many others do. There is a library called whois in python which is used to get all the information here.



FINAL SCRITP WITH ALL OF THE COMPONENTS:



CONCLUSION

- The project demonstrated that combining two simple techniques (Google dorking and DNS/nslookup) can reveal a surprising amount of publicly available information about a domain.
- Using advanced Google operators makes it easier to reach resources that are indexed but not meant to be easily found (login pages, backups, documents, admin panels).
- DNS record analysis (A, MX, NS, SOA) allows for an initial mapping of the target's infrastructure and related services.
- Python can be used as an automation layer for reconnaissance tasks – with the os module it is possible to call system networking tools directly from the script.
- Automation must take into account limitations (search engine rate limiting, OS differences) and follow ethical rules – public data only, authorized targets only.

REFERENCES

- Mosh Hamedani, Python Full Course for Beginners (YouTube, “Programming with Mosh”)
- Python 3 Standard Library documentation (modules: os, subprocess)
- Google Search Operators – official help / support pages
- Microsoft Docs – nslookup command
- AI-assisted explanations (ChatGPT – to clarify OSINT techniques and script structure)
- https://www.youtube.com/watch?v=CqIW4dI8a9c
- https://pypi.org/project/nslookup/
- https://hackernoon.com/python-build-a-domain-lookup-tool


THANK YOU