

AI-Powered OSINT Reconnaissance

Responsible Use Warning: The techniques described in this lesson are intended *exclusively* for educational purposes and authorized security testing. Performing reconnaissance against systems or individuals without explicit written authorization is illegal in most jurisdictions. Always obtain proper permission before conducting any OSINT activities against real targets. Use lab environments and intentionally vulnerable platforms for practice.

What Is OSINT?

Open Source Intelligence (OSINT) is the practice of collecting, processing, and analyzing information from publicly available sources to produce actionable intelligence. The term originated in military and government intelligence communities, but it has become a cornerstone of modern cybersecurity, penetration testing, and red team operations.

The key word is *open*: OSINT does not involve hacking into systems or accessing private data. Instead, it leverages the vast amount of information that individuals and organizations inadvertently expose through everyday digital activity. A company's DNS records, an employee's LinkedIn profile, a forgotten paste on Pastebin, or a misconfigured S3 bucket listed on Shodan — all of these are fair game for OSINT.

In a red team context, OSINT is almost always the first phase of an engagement. Before you craft a phishing email, exploit a vulnerability, or attempt lateral movement, you need to understand your target. The more you know, the more precise your attacks become.

OSINT Sources

OSINT data comes from a wide variety of public channels. Here are the most important categories for security professionals:

Source Category	Examples	What You Can Find
Search Engines	Google, Bing, DuckDuckGo	Indexed pages, cached content, exposed files
Social Media	LinkedIn, Twitter/X, Facebook, GitHub	Employee names, roles, tech stacks, internal tooling
WHOIS / DNS	whois, dig, dnsdumpster.com	Domain ownership, IP ranges, mail servers, subdomains
Public Records	SEC filings, court records, patents	Corporate structure, acquisitions, legal disputes
Paste Sites	Pastebin, GitHub Gists, Ghostbin	Leaked credentials, config files, internal notes
Code Repositories	GitHub, GitLab, Bitbucket	API keys, internal URLs, infrastructure details
Internet-Wide Scanners	Shodan, Censys, ZoomEye	Exposed services, banners, certificates, IoT devices
Certificate Transparency	crt.sh, Censys	Subdomains, internal hostnames via TLS certificates

The OSINT Workflow

Effective OSINT follows a structured methodology. A widely adopted framework breaks the process into five phases:

- 1. Define Scope** — Establish what you are looking for and the boundaries of your engagement. Are you mapping an organization's external attack surface? Identifying key personnel for social engineering? Document your objectives and legal constraints.
- 2. Collect** — Gather raw data from the sources listed above. Cast a wide net initially, then narrow your focus. Automate repetitive collection tasks with tools.
- 3. Process** — Clean, deduplicate, and normalize the data. Convert screenshots to text, parse DNS records into structured formats, and organize findings into a searchable

knowledge base.

4. **Analyze** — Identify patterns, correlations, and actionable insights. This is where AI becomes a force multiplier — more on this below.
5. **Report** — Produce a clear, structured report that ties findings to risk. Include evidence, confidence levels, and recommendations.

Core OSINT Tools

Google Dorking

Google dorking uses advanced search operators to find information that standard searches miss. These operators filter results by file type, site, URL structure, and more:

```
# Find exposed PDF files on a target domain
site:example.com filetype:pdf

# Find login pages
site:example.com inurl:login OR inurl:admin

# Find directory listings (often misconfigured servers)
site:example.com intitle:"index of" "parent directory"

# Find exposed environment files
site:example.com filetype:env "DB_PASSWORD"

# Find publicly accessible config files
site:example.com ext:xml | ext:conf | ext:cnf | ext:reg | ext:inf | ext:rdp |

# Search for subdomains indexed by Google
site:*.example.com -www
```

Tip: The Google Hacking Database (GHDB) at exploit-db.com/google-hacking-database contains thousands of tested dork queries organized by category. It is an excellent starting point.

theHarvester

theHarvester is a command-line tool that aggregates data from multiple search engines and public data sources. It collects emails, subdomains, IPs, and URLs associated with a target domain:

```
# Basic email and subdomain enumeration
theHarvester -d example.com -b google,bing,dnsdumpster -l 200

# Use all available sources
theHarvester -d example.com -b all -l 500

# Output results to an HTML report
theHarvester -d example.com -b all -l 500 -f recon_report
```

Shodan

Shodan indexes internet-connected devices and services. Unlike Google, which indexes web page content, Shodan indexes service banners — the metadata that servers, IoT devices, and industrial control systems broadcast:

```
# Search for a specific organization
shodan search "org:Example Corp"

# Find all exposed Apache servers for a domain
shodan search "hostname:example.com Apache"

# Look for exposed databases
shodan search "port:27017 product:MongoDB" --limit 10

# Get detailed info about a specific IP
shodan host 93.184.216.34
```

Recon-ng

Recon-ng is a modular reconnaissance framework with a workflow similar to Metasploit. It provides a consistent interface for running dozens of OSINT modules:

```
# Launch and create a workspace
recon-ng
workspaces create example_recon

# Add a target domain as a seed
db insert domains example.com
```

```
# Load and run a module for subdomain enumeration
modules load recon/domains-hosts/certificate_transparency
run

# Harvest contacts from the discovered data
modules load recon/domains-contacts/whois_pocs
run

# Generate a report
modules load reporting/html
options set FILENAME /tmp/recon_report.html
run
```

How AI Supercharges OSINT

Traditional OSINT is labor-intensive. An analyst might spend hours manually reviewing search results, cross-referencing names across platforms, and trying to spot patterns in raw data. Large Language Models (LLMs) like Claude and ChatGPT change this dynamic in several critical ways:

1. Pattern Recognition at Scale

LLMs can ingest large volumes of unstructured text — employee bios, job postings, press releases, code comments — and extract structured intelligence from them. Patterns that would take a human analyst hours to spot can surface in seconds.

2. Natural Language Querying

Instead of writing complex regex patterns or custom parsers, you can ask an LLM to analyze data in plain English: *"From these 50 job postings, identify every technology, framework, and cloud provider mentioned, then rank them by frequency."*

3. Data Correlation

LLMs excel at connecting dots across disparate data sources. Feed them WHOIS records, LinkedIn profiles, and GitHub commits, and they can map relationships between people, organizations, and infrastructure.

4. Search Query Generation

Rather than manually crafting Google dorks, you can describe what you are looking for and let an LLM generate dozens of targeted queries:

```
import anthropic

client = anthropic.Anthropic()

message = client.messages.create(
    model="claude-sonnet-4-20250514",
    max_tokens=1024,
    messages=[
        {
            "role": "user",
            "content": (
                "Generate 10 Google dork queries to find potentially exposed "
                "configuration files, backup files, and admin panels for the "
                "domain example.com. Return only the queries, one per line."
            )
        }
    ]
)

queries = message.content[0].text.strip().split("\n")
for q in queries:
    print(q)
```

5. Automated Report Generation

After collecting raw OSINT data, you can use an LLM to synthesize it into a structured reconnaissance report:

```
import anthropic
import json

client = anthropic.Anthropic()

# Assume we have collected data from various sources
recon_data = {
    "domain": "example.com",
    "subdomains": ["mail.example.com", "vpn.example.com", "dev.example.com"],
    "emails": ["admin@example.com", "j.smith@example.com"],
    "technologies": ["Apache/2.4.41", "PHP/7.4", "WordPress 6.2"],
    "open_ports": [22, 80, 443, 8080, 3306],
```

```

        "whois_org": "Example Corporation",
    }

message = client.messages.create(
    model="claude-sonnet-4-20250514",
    max_tokens=2048,
    messages=[
        {
            "role": "user",
            "content": (
                "You are a penetration testing assistant. Analyze the following  

                OSINT reconnaissance data and produce a structured report. "  

                "Identify potential attack vectors, flag risky findings, and "  

                "prioritize them by severity \n\n"
            )
        }
    ]
)

```

Key Insight: AI does not replace the OSINT analyst — it amplifies them. You still need to define the right questions, validate findings, and exercise judgment. LLMs can hallucinate or miss context that a human would catch. Always verify AI-generated conclusions against primary sources.

A Simple OSINT Automation Script

The following Python script demonstrates a minimal OSINT pipeline that combines DNS enumeration with AI-powered analysis. It queries certificate transparency logs for subdomains, then uses Claude to assess the findings:

```

import requests
import json
import anthropic

def get_subdomains_from_crt(domain: str) -> list[str]:
    """Query crt.sh certificate transparency logs for subdomains."""
    url = f"https://crt.sh/?q=%.{domain}&output=json"
    resp = requests.get(url, timeout=15)
    resp.raise_for_status()

    subdomains = set()
    for entry in resp.json():
        name = entry.get("name_value", "")
        for sub in name.split("\n"):

```

```
        sub = sub.strip().lower()
        if sub.endswith(domain) and "*" not in sub:
            subdomains.add(sub)

    return sorted(subdomains)

def analyze_with_ai(domain: str, subdomains: list[str]) -> str:
    """Use Claude to analyze discovered subdomains for risk."""
    client = anthropic.Anthropic()
    message = client.messages.create(
        model="claude-sonnet-4-20250514",
        max_tokens=1024,
```

Ethical and Legal Considerations

OSINT occupies an interesting legal position: the data is public, but how you collect and use it matters. Consider these principles:

- **Authorization:** In a professional engagement, always have a signed scope of work or rules of engagement (ROE) before starting reconnaissance. Even passive OSINT can cross legal lines if it targets individuals without consent in certain jurisdictions (e.g., GDPR in the EU).
- **Proportionality:** Collect only what is relevant to your objective. Hoarding personal data "just in case" is both unethical and potentially illegal.
- **Data Handling:** Treat collected OSINT data with the same care as any sensitive information. Store it securely, share it only with authorized parties, and destroy it when the engagement ends.
- **Passive vs. Active:** There is an important distinction between passive reconnaissance (reading public data) and active reconnaissance (directly probing target systems). DNS lookups and Google searches are passive. Port scanning and vulnerability probing are active and typically require explicit authorization.
- **Disclosure:** If you discover exposed credentials, sensitive data, or critical vulnerabilities during authorized OSINT, follow responsible disclosure practices.

Legal Notice: Laws vary by country and region. Activities that are legal in one jurisdiction may be criminal in another. Familiarize yourself with the Computer Fraud and Abuse Act (CFAA) in the US, the Computer Misuse Act in the UK, and

equivalent legislation in your jurisdiction. When in doubt, consult legal counsel before proceeding.

Key Takeaways

Reconnaissance is the foundation of every successful red team engagement. The more thorough your OSINT phase, the more precise and effective your subsequent attacks will be.

- OSINT leverages publicly available data — no hacking required.
- Follow a structured workflow: scope, collect, process, analyze, report.
- Master the core tools: Google dorking, theHarvester, Shodan, Recon-ng, and certificate transparency searches.
- AI/LLMs act as a force multiplier for pattern recognition, data correlation, query generation, and report writing.
- Always verify AI-generated conclusions against primary sources.
- Operate within legal and ethical boundaries at all times.

Self-check: Can you explain the difference between passive and active reconnaissance? Can you write three Google dork queries from memory to find exposed files on a target domain? Can you describe how you would use Claude to analyze raw OSINT data?

Challenge Resources

Download Files

These files contain synthetic data for the OSINT challenge. All files are plain text (.txt) and safe to open.

↓ [whois-records.txt](#) ~2 KB

↓ [dns-records.txt](#) ~1 KB

↓ [social-profiles.txt](#) ~3 KB

↓ [certificate-transparency.txt](#) ~2 KB

↓ [paste-dump.txt](#) ~1 KB