**Module 3 Lab**

**Methodology: Using OSINT Tools in Penetration Testing**

**1. Objective**

To simulate the reconnaissance phase of a penetration test using Open-Source Intelligence (OSINT) tools, in order to collect publicly available information about a target organization. This process demonstrates how attackers can leverage freely accessible data to build a threat model before active scanning begins.

**2. Tools and Environment**

* **OSINT Framework** (Web-based)
* **WhatsMyName** (GitHub project and web interface)
* **SMART (Start Me Aggregated Resource Tool)**
* **SpiderFoot** (Automated OSINT scanner)
* **Recon-ng** (Modular reconnaissance framework)
* **Kali Linux** (Pre-configured for ethical hacking)
* **Internet access**

**3. Passive Reconnaissance Process**

**3.1 Examining OSINT Framework**

* Accessed the **OSINT Framework** at <https://osintframework.com>
* Navigated categories such as **Username**, explored **WhatsMyName**
* Identified social media or online account presence based on username enumeration
* Observed that these resources help identify a user’s digital footprint and potential attack vectors via social engineering or credential reuse.

**3.2 Using WhatsMyName**

* Input a set of usernames and analyzed their presence across various platforms.
* Filtered and exported the results (CSV/PDF) for reporting.
* Evaluated the presence of user accounts on multiple services to detect potential reuse or abandoned accounts.

**Purpose**: Username enumeration assists in targeted phishing, account compromise, or brute-force attacks using known aliases.

**3.3 Investigating SMART (My OSINT Training)**

* Queried usernames, domains, and other keywords.
* Evaluated tools bookmarked and shared by the OSINT community.
* Noted that the SMART platform is a dynamic index of community-endorsed resources.

**Purpose**: Rapid access to niche tools categorized by functionality (e.g., metadata extraction, social media intelligence).

**4. Active-Passive Reconnaissance via SpiderFoot**

**4.1 Launching SpiderFoot**

* Ran SpiderFoot locally using spiderfoot -l 127.0.0.1:5001
* Accessed the GUI at http://127.0.0.1:5001

**4.2 Scanning Target (h4cker.org)**

* Selected **Footprint** use case for a balanced scan.
* Monitored data acquisition through SpiderFoot's graphing UI.
* Browsed modules and results categorized by data type (IP, ASN, email, leaked credentials).

**4.3 API Integration**

* Registered and configured API keys for enhanced scanning (e.g., HaveIBeenPwned, Shodan).
* Ran an API-focused scan to enrich findings with breach data, DNS records, and threat intelligence.

**Purpose**: SpiderFoot automates correlation of public data points, reducing manual effort in footprinting.

**5. Modular Reconnaissance with Recon-ng**

**5.1 Recon-ng Workspace Setup**

* Initialized Recon-ng using recon-ng
* Created isolated workspace: workspaces create test
* Explored workspace management with commands like workspaces list, workspaces remove, exit

**5.2 Module Discovery and Management**

* Queried available modules using modules search and marketplace search
* Searched and installed modules such as:
  + recon/domains-hosts/bing\_domain\_web
  + recon/domains-hosts/hackertarget
* Viewed module requirements (marketplace info) including dependencies and API keys.

**Purpose**: Recon-ng allows controlled, scriptable, and repeatable reconnaissance. Modules maintain configuration within workspace context.

**6. Data Analysis and Reporting**

* Reviewed SpiderFoot results to identify:
  + Whois data
  + Subdomain enumeration
  + Email leaks
  + Reputation indicators (blacklists)
* Analyzed Recon-ng module outputs for:
  + Host and IP enumeration
  + Email harvesting
  + DNS data
* Evaluated the scope of exposed information that could assist an attacker in pre-exploitation planning.

**7. Conclusion**

This lab demonstrated a standard OSINT workflow using industry-grade tools. From initial enumeration using OSINT Framework and WhatsMyName to automated reconnaissance via SpiderFoot and modular scanning through Recon-ng, the process replicates real-world threat actor behavior. Effective passive reconnaissance allows ethical hackers to identify weak points without engaging the target network, preserving stealth and legality.