SpiderFoot: The Definitive Guide to Automated OSINT and Attack Surface Intelligence

Foreword: The Digital Detective's Automated Assistant

In the sprawling, interconnected digital landscape of the 21st century, information is both the greatest asset and the most significant liability. For cybersecurity professionals, the ability to navigate this ocean of data—to find the signal in the noise—is the defining skill of the era. Open-Source Intelligence (OSINT) has evolved from a niche discipline into a foundational pillar of modern security operations, underpinning everything from offensive reconnaissance to defensive threat hunting and corporate due diligence. The challenge, however, is one of scale. The sheer volume of publicly available information about any given target is staggering, making manual collection and analysis an inefficient, if not impossible, task.

This is the world into which SpiderFoot was born. It is more than just a tool; it is an automated reconnaissance framework, a digital detective's tireless assistant designed to methodically and relentlessly gather, correlate, and organize data from the farthest corners of the internet. It acts as a force multiplier, automating the laborious and time-consuming aspects of OSINT, thereby freeing the human analyst to focus on what they do best: analysis, interpretation, and strategic decision-making.

This guide is designed to be the definitive resource for mastering SpiderFoot. It will take you from the fundamental concepts of its architecture to the practical intricacies of installation, from launching your first basic scan to orchestrating advanced, multi-faceted investigations. We will explore its dual nature as both an offensive weapon for red teams and an indispensable defensive shield for blue teams. We will delve into its vast module library, unlock its full potential with API integrations, and transform its raw data output into actionable intelligence through powerful visualization and analysis techniques.

Whether you are a penetration tester mapping an attack surface, a threat intelligence analyst tracking an adversary's infrastructure, a digital forensics investigator building a case, or a corporate security professional assessing your organization's digital footprint, this Ebook will serve as your comprehensive manual. It aims to equip you not only with the "how" of using SpiderFoot but also the "why" behind its methods, empowering you to leverage this powerful platform to its maximum potential and turn the tide of information in your favor.

Chapter 1: Unveiling SpiderFoot: An Introduction to Automated OSINT

1.1 What is SpiderFoot? A History and Overview

At its core, SpiderFoot is an open-source intelligence (OSINT) automation tool designed to automate the process of gathering intelligence about a given target.[1] It operates as a reconnaissance framework, systematically querying a vast array of public data sources to collect and correlate information, presenting it to the user in a navigable and intuitive format.[3]
A target, or "seed," for a SpiderFoot scan can be almost any digital entity, including a domain name, IP address, email address, username, phone number, or even a person's name.[1]
The project was created by Steve Micallef and has been in active development since 2012, establishing itself as a mature and robust platform within the cybersecurity community.[1]
Written in Python, it is inherently cross-platform, capable of running on Linux, macOS, and Windows environments.[5] Users can interact with SpiderFoot through two primary interfaces: a clean, browser-based web UI that provides graphical representations of data, or a powerful command-line interface (CLI) for scripting and automation.[1]
A pivotal moment in SpiderFoot's history occurred in November 2022, when it was acquired by Intel 471, a prominent cyber threat intelligence company.[9] As part of this acquisition, founder Steve Micallef joined Intel 471 as its Vice President of Attack Surface Technology. This event marked a significant evolution for the tool, signaling its transition from a highly regarded open-source project into a commercially significant platform with professional backing. The acquisition implies a long-term commitment to the tool's future, with increased resources likely driving further development. Intel 471 announced plans to invest in SpiderFoot's existing tools and integrate its capabilities into their enterprise-grade TITAN cybercrime intelligence platform.[10] This professional validation underscores the tool's value and suggests that future enhancements, even in the open-source version, will be informed by the rigorous demands of enterprise-level threat intelligence.

1.2 The Dual Nature: Offensive Reconnaissance vs. Defensive Intelligence

SpiderFoot is engineered with a fundamental duality, making it an invaluable asset for both offensive and defensive security operations.[1] Its utility is not defined by the tool itself, but by the intent of the user operating it.

For offensive practitioners, such as penetration testers and red teams, SpiderFoot is a formidable reconnaissance weapon. It automates the initial, critical phase of an engagement: information gathering. By pointing SpiderFoot at a target organization, an attacker can rapidly map the external attack surface, discovering assets like subdomains, IP address ranges, and cloud storage buckets.[1] This information is crucial for identifying potential entry points. Furthermore, the tool can gather intelligence vital for crafting sophisticated social engineering attacks, such as identifying employee names, email addresses, and social media profiles, which can be leveraged in highly targeted phishing or spear-phishing campaigns.[4]

Conversely, for defensive practitioners, such as blue teams and corporate security analysts, SpiderFoot serves as a powerful mirror, reflecting the organization's own digital footprint as seen from an attacker's perspective.[1] By turning the tool on their own infrastructure, defenders can proactively identify and remediate exposures before they are exploited. This includes discovering forgotten or unauthorized assets, often referred to as "shadow IT," which may not be compliant with corporate security policies.[3] It can uncover vulnerabilities like susceptibility to subdomain hijacking, identify open ports on servers, and monitor for mentions of the company's domain or employee credentials in public data breach dumps.[1]

This dual-use capability makes SpiderFoot a quintessential "purple team" tool. The very same scan that reveals a potential attack vector to a red teamer simultaneously highlights a critical vulnerability that a blue teamer must address. The data is neutral; its interpretation and subsequent action depend on the user's role. This makes the tool exceptionally well-suited for collaborative security exercises where offensive and defensive teams work in concert to improve an organization's overall security posture.

1.3 The SpiderFoot Ecosystem: Open Source vs. SpiderFoot HX

The SpiderFoot ecosystem consists of two distinct offerings: the classic, free open-source version and a commercial, cloud-based platform known as SpiderFoot HX.[1] Understanding the differences between these two is crucial for selecting the right tool for a given use case, budget, and operational environment.

The **open-source version** of SpiderFoot is a powerful, feature-rich tool that can be self-hosted on any system with the appropriate Python environment. It provides access to the full suite of over 200 modules, the web UI, the CLI, the SQLite database backend, and the YAML correlation engine.[4] It is ideal for individual researchers, students, and professionals who are comfortable managing their own infrastructure, dependencies, and API key configurations. Its primary limitations are operational: it is inherently a single-user application, and the user is solely responsible for installation, updates, security, and the manual acquisition and configuration of all third-party API keys.[12]

SpiderFoot HX is the commercial Software-as-a-Service (SaaS) offering, designed to address the operational challenges of using the open-source version at scale in a professional setting. It takes the core engine of SpiderFoot and hosts it in a fully managed, cloud-based environment, eliminating the need for users to handle installation, updates, or server maintenance.[1] The value proposition of HX is built directly upon solving the pain points of the open-source version.

The relationship between the two versions is symbiotic. The open-source tool serves as a powerful demonstration of the platform's capabilities, attracting a wide user base. Professionals and organizations that require greater performance, scalability, collaboration, and support are then naturally drawn to the commercial HX platform. This model ensures the continued development and relevance of both offerings, with the open-source version benefiting from the core engine advancements driven by the commercial product.

The following table provides a detailed comparison of the key features distinguishing the two versions, offering a clear framework for deciding which is best suited for your needs.

Feature	SpiderFoot Open Source	SpiderFoot HX
Hosting & Management	Self-hosted and self-managed. User is responsible for installation, updates, and security.	100% Cloud-based and professionally managed for the user.[1]
Performance	Standard performance, dependent on local hardware and network.	Tuned for performance, with scans running approximately 5-10x faster.[12]
Team Collaboration	Single-user application.[12]	Supports multiple named users with role-based access control for team collaboration.[12]
Security	User is responsible for securing the instance (e.g., firewall, authentication).	Professionally managed security with two-factor authentication (2FA) available.[12]
Targets per Scan	Limited to a single target per scan.[12]	Supports scanning multiple targets simultaneously in a single scan.[1]
Attack Surface Monitoring	Manual process; requires re-running scans and comparing results.	Built-in feature with automated change detection and notifications via email, Slack, and REST endpoints.[1]
API/Tool Integration	User must acquire and configure all API keys and install external tools (e.g., Nmap) themselves.[12]	Includes HX-only modules and comes with many third-party tools and APIs pre-installed and configured.[1]
Advanced Features	Core features including modules, correlations, and visualizations.	Includes exclusive features like Investigations, Screenshotting, and data feeds to Splunk/ElasticSearch.[1]
TOR Integration	User must configure TOR	Built-in and seamlessly

	integration themselves.[12]	integrated for anonymous scanning.[1]
Support	Community support via Discord and GitHub.[1]	Dedicated customer support.[1]
Cost	Free and open-source (MIT/GPL licensed depending on version).[1]	Subscription-based with tiered pricing (e.g., Hobby, Freelancer, Business, Enterprise).[12]

Chapter 2: Architecture and Core Concepts

2.1 Under the Hood: The Publisher-Subscriber Model

To effectively wield SpiderFoot, one must first understand its architectural heart: a publisher-subscriber model for data handling and module interaction.[1] This design is the engine that drives the tool's automated, cascading data enrichment process. In this paradigm, every piece of information discovered during a scan is treated as an "event." Modules are designed to both "publish" new events and "subscribe" to events of specific types that they are interested in.

The process begins with the initial seed target provided by the user (e.g., a domain name). This seed is the first event. A module subscribed to the DOMAIN_NAME event type, such as a DNS resolver module, will be notified. This module processes the domain name, resolves it to an IP address, and then "publishes" a new event of type IP_ADDRESS.

This new IP_ADDRESS event, in turn, triggers all modules that are subscribed to it. For instance, a port scanning module, a WHOIS lookup module, and a threat intelligence module might all activate simultaneously. The port scanner might discover an open port and publish an OPEN_PORT event. The WHOIS module might find contact information and publish an EMAIL_ADDRESS event. The threat intelligence module might find the IP on a blacklist and publish a MALICIOUS_IP event. Each of these new events can then trigger yet another wave of subscribed modules, creating a powerful, automated chain reaction of discovery and analysis.[1]

This publisher-subscriber model is the source of SpiderFoot's greatest strength: its ability to perform massive, broad, and divergent data correlation automatically. It excels at taking a single data point and fanning out to discover a vast network of related information. However, this same architecture presents a notable limitation for certain types of analysis. The model is fundamentally event-driven, with modules reacting to single data points as they are published. Modules do not have easy, native access to the complete, holistic picture of all data found so far in the scan.

This becomes a challenge for modules that perform best with convergent analysis—that is, modules that require multiple, disparate data points about a single entity to function optimally. A documented example of this limitation involves a proposed custom module for the PIPL people search service, which yields far better results when queried with a combination of a name, email address, and physical address simultaneously.[15] Within SpiderFoot's architecture, a module cannot simply ask, "Give me the name, email, and address for this person." Instead, it must subscribe to HUMAN_NAME, EMAIL_ADDRESS, and PHYSICAL_ADDRESS events separately. This makes it difficult to build a complete profile before querying the external service without resorting to complex workarounds, such as storing data in memory within the module and waiting for a signal that the scan is complete.[15] This architectural nuance is a critical consideration for any developer planning to write custom modules for the platform.

2.2 The Power of Modules: Over 200 Data Sources

The functional capabilities of SpiderFoot are encapsulated within its extensive library of over 200 modules.[1] These modules are the individual workers of the framework, each designed to perform a specific data collection or analysis task. They are the primary mechanism through which SpiderFoot interacts with the outside world, querying an immense variety of data sources to build an intelligence picture of the target.[3] The sheer breadth of these integrations is what makes SpiderFoot a "Swiss Army knife" of OSINT; it acts as a master orchestrator, unifying dozens of otherwise disparate tools and workflows into a single, cohesive investigation.[3]

The modules can be broadly categorized based on the type of information they seek and the methods they employ:

- **DNS and Network Infrastructure:** These modules form the foundation of most network-based reconnaissance. They include tools for DNS resolution (sfp_dnsresolve), raw DNS record retrieval (MX, TXT, etc.), DNS zone transfers, port scanning (sfp_ports), and banner grabbing to identify running services.[1]
- **Threat Intelligence and Blacklists:** A crucial category for security assessments, these modules query dozens of threat intelligence feeds and blacklist providers. Integrations include well-known services like SHODAN, VirusTotal, HaveIBeenPwned, GreyNoise, AlienVault OTX, and AbuseIPDB to determine if a target IP, domain, or email has been associated with malicious activity.[1]
- **Identity and Social Media:** These modules focus on finding information related to people and online personas. They can extract email addresses, phone numbers, and human names from web content, and enumerate social media accounts across a wide range of platforms like Twitter and Facebook (sfp_twitter, sfp_facebook).[1]
- **Web Content and Metadata:** This category includes modules that scrape web pages for links and information, analyze web content, and extract metadata from discovered documents, images, and binary files. This metadata can often reveal sensitive

information like usernames, software versions, and internal file paths.[1]

- **Cloud and Storage:** With the rise of cloud computing, these modules are increasingly important. They are designed to enumerate and test for publicly accessible Amazon S3 buckets, Azure blobs, and DigitalOcean Spaces associated with a target.[1]
- **Dark Web and Data Breaches:** These modules extend the search into the darker corners of the internet. They can query TOR search engines like Ahmia and check public data breach aggregations for mentions of the target's assets.[1]
- **External Tool Integration:** Beyond its native modules, SpiderFoot can also act as a wrapper to call other popular command-line tools, incorporating their results into its own data stream. This includes integrations with Nmap for advanced port scanning, WhatWeb for web technology fingerprinting, and DNSTwist for identifying typosquatted domains.[1]

This modular design is the key to SpiderFoot's longevity and adaptability. As new OSINT sources and techniques emerge, the framework can evolve simply by adding new modules, ensuring it remains a current and relevant tool in the ever-changing landscape of cybersecurity.

2.3 The Brain: The YAML Correlation Engine and SQLite Backend

While the modules are responsible for data collection, the intelligence processing and storage are handled by two other core components: the SQLite database backend and the YAML-configurable correlation engine.

Every piece of data discovered by every module during a scan, along with its metadata (such as the source module and its relationship to other data), is stored in a local SQLite database file.[1] This approach has several advantages. SQLite is a lightweight, serverless, and self-contained database engine, making deployment simple and portable. More importantly, it provides a structured and queryable repository of all scan findings. This allows advanced users to go beyond the capabilities of the web UI and perform highly specific, custom queries directly against the database using standard SQL, enabling limitless possibilities for deep analysis.[4]

The introduction of the YAML-configurable correlation engine in SpiderFoot version 4.0 marked a significant evolution for the tool, elevating it from a pure data collection framework to a basic data *analysis* platform.[4] This engine addresses the primary challenge of OSINT: data overload. A comprehensive scan can generate thousands of data points, and sifting through them to find the most critical information can be a daunting task.

The correlation engine automates this initial triage. It consists of a set of rules, written in the human-readable YAML format, that define conditions of interest.[18] These rules are essentially pre-canned queries that SpiderFoot runs against the SQLite database after a scan completes. The platform comes with over 37 pre-defined rules designed to flag common high-risk findings.[4] Examples of these built-in correlations include:

- Identifying hosts or IP addresses that have been reported as malicious by multiple, independent threat intelligence sources.

- Flagging "outlier" web servers that are running different software from the rest of the target's infrastructure, which can be an indication of shadow IT.
- Highlighting internet-exposed databases or open ports that reveal specific software versions, which could be vulnerable.
- Noting the discovery of data in document metadata.[18]

By automatically analyzing the scan results and surfacing these potentially critical findings, the correlation engine shifts some of the analytical burden from the user to the tool. It helps analysts focus their attention on the needles in the haystack, accelerating the process of turning raw data into actionable intelligence.

Chapter 4: Your First Investigation: Launching and Running Scans

4.1 The Command Center: Navigating the Web UI and CLI

SpiderFoot offers two distinct interfaces for command and control, catering to different workflows and user preferences: a graphical Web User Interface (Web UI) and a powerful Command-Line Interface (CLI). This dual-interface design is a deliberate choice that maximizes the tool's flexibility, allowing it to serve as both an interactive investigative workbench and a scriptable component in automated security pipelines.[16]

The Web User Interface (UI)

For most users, especially those new to the tool, the Web UI is the primary method of interaction. It is launched by running the main Python script with the -l (listen) flag, specifying an IP address and port for the embedded web server to bind to. A typical launch command is:

python3./sf.py -l 127.0.0.1:5001

Once executed, this command starts the server, and the user can access the interface by navigating to http://127.0.0.1:5001 in a web browser.[1] The UI is praised for its clean layout, ease of navigation, and its ability to present results in an intuitive, graphical format.[5] The main dashboard is organized around three core tabs:

- **New Scan:** This is the starting point for any investigation, where the user defines the target and configures the parameters of the scan.
- **Scans:** This tab displays a history of all past and currently running scans, allowing the user to monitor progress, review results, or manage scans (e.g., stop, delete, or clone).
- **Settings:** This is the configuration hub for the entire SpiderFoot instance. Here, users

can manage global settings, configure TOR integration, and, most importantly, enter API keys for the various modules.[5]

The Command-Line Interface (CLI)

For power users, scripters, and those looking to integrate SpiderFoot into automated workflows, the CLI is the interface of choice. SpiderFoot provides two distinct command-line tools [2]:

1. sf.py: This is the main script used to launch the web server, but it can also be used to execute "headless" scans directly from the terminal without ever starting the UI. A wide range of flags allows for precise control over the scan. For example, the following command would run a full "footprint" scan against target.com and output the results in CSV format:
   ```
   python3 sf.py -s target.com -u footprint -o csv
   ```
2. **sfcli.py:** This script provides an interactive command-line shell that connects to an already running SpiderFoot server instance. This allows an analyst to manage scans, view results, and interact with the server remotely, which is particularly useful for managing long-running scans on a remote server from a local machine.[2]

The availability of these robust CLI options is what enables SpiderFoot to be used for tasks like scheduled, recurring reconnaissance, making it a valuable tool for continuous attack surface monitoring and automated threat intelligence gathering.[16]

4.2 Defining the Mission: Selecting Targets and Scan Profiles

Initiating a new investigation in SpiderFoot begins on the "New Scan" tab. This process involves two critical decisions: defining the target and selecting the appropriate scan methodology. These choices are not merely procedural; they are the most important strategic decisions a user makes, as they directly influence the scan's duration, stealth, thoroughness, and the ultimate quality of the intelligence gathered.

First, every scan must be given a unique name for future reference. Next, the user must provide the "Seed Target," which is the initial piece of information that the investigation will pivot from. SpiderFoot supports a comprehensive list of seed target types, including:

- IP Address (IPv4 or IPv6)
- Domain/Sub-domain Name
- Hostname
- Network Subnet (CIDR)
- ASN (Autonomous System Number)
- E-mail Address
- Phone Number
- Username

- Person's Name
- Bitcoin Address [1]

After defining the target, the user must select a scan profile. This choice represents a trade-off between speed, stealth, and comprehensiveness. SpiderFoot offers three primary ways to configure a scan [19]:

1. **By Use Case:** This offers pre-defined templates for common scenarios:
   - **All:** The most comprehensive but also the slowest and noisiest option. It enables all modules to gather every possible piece of information.[16]
   - **Footprint:** Focuses on mapping the target's public-facing infrastructure and identity.
   - **Investigate:** A balanced approach that performs footprinting while also checking for malicious indicators.
   - **Passive:** The most stealthy option. This profile gathers intelligence exclusively from third-party sources (like search engines and threat feeds) and never interacts directly with the target's servers. This is ideal for investigations where avoiding detection is paramount, though it may miss data that is only available via direct contact.[16]
2. **By Required Data:** This approach allows the user to specify the *types* of data they wish to find (e.g., "Email Addresses," "Leaked Passwords"). SpiderFoot will then automatically select and enable only the modules necessary to retrieve that specific information, making for a very efficient scan.[19]
3. **By Module:** This provides the most granular level of control, allowing the user to manually enable or disable each of the 200+ modules individually. This is best for advanced users who have a very specific investigative goal and a deep understanding of the modules and their dependencies. A misconfigured scan of this type can lead to incomplete results; for example, if a module that relies on data from another disabled module is run, it will produce no output.[19]

The selection of a scan profile is therefore a strategic decision based on the specific goals of the investigation, balancing the need for comprehensive data against the constraints of time and the requirement for operational security.

4.3 Monitoring the Hunt: Understanding Scan Status and Live Results

Once a scan is launched, SpiderFoot provides a dynamic and interactive environment for monitoring its progress. A key usability feature of the platform is its real-time feedback loop, which transforms it from a static, batch-processing tool into an active investigative workbench. Analysts are not required to wait for a potentially hours-long scan to complete before they can begin their analysis; they can view and interact with data as it is discovered.[8]

Upon starting a scan, the user is taken to the scan's dashboard. The main visual indicator of progress is a series of bar charts, with one bar for each module enabled in the scan. These bars fill up as each module runs and completes its tasks, providing a quick, at-a-glance overview of the overall scan status.[5] Below these charts, a more detailed status log shows which modules are currently active and what data they are processing.

This real-time discovery process is crucial for an agile investigation. It allows an analyst to pivot their strategy mid-stream. For example, if a broad scan on a corporate domain quickly uncovers a leaked email address belonging to a key executive, the analyst can immediately launch a new, highly targeted scan on that email address to search for associated accounts or breach data, without having to wait for the initial, broader scan to finish its full run. This iterative approach, where findings from one scan become the seeds for the next, more closely mirrors the natural flow of a human-led investigation and can dramatically accelerate the time it takes to arrive at critical insights. From the "Scans" tab, the user has full control over running processes, with the ability to pause, resume, stop, or delete a scan at any time.[19]

Chapter 5: Mastering the Arsenal: Modules and API Integration

5.1 A Tour of the Module Library

The true power and versatility of SpiderFoot lie in its modular architecture. The framework's ability to query over 200 distinct data sources is facilitated by its extensive library of individual modules, each designed for a specific intelligence-gathering task.[1] This modularity is what allows SpiderFoot to remain adaptable and current; as new OSINT resources become available or old ones change, the community can develop and integrate new modules without needing to alter the core application.[24]

The modules can be grouped into several key functional categories, providing a comprehensive toolkit for any investigation:

- **Network and Infrastructure Intelligence:** These are the foundational modules for any target that has an online presence. Key modules include sfp_dnsresolve for resolving domain names to IP addresses, sfp_whois for retrieving domain registration data, and sfp_ports for identifying open TCP ports on a host. More advanced modules in this category can perform DNS brute-forcing, attempt DNS zone transfers, and conduct banner grabbing to fingerprint services running on open ports.[16]
- **Social Media & Public Profiles:** Focusing on the human element, this category includes modules like sfp_twitter and sfp_facebook to find social media accounts related to a target. The powerful sfp_accounts module checks for the existence of a given username across nearly 200 different websites, from forums to developer platforms.[1]
- **Dark Web & Data Breaches:** These modules extend the search into less savory parts of the internet. sfp_darkweb can search for mentions of a target on dark web forums

and marketplaces, while sfp_pwned integrates with the HaveIBeenPwned service to check if email addresses or usernames associated with the target have appeared in known data breaches.[16]
- **Threat Intelligence:** A critical category for any security assessment, these modules cross-reference discovered assets (IPs, domains, file hashes) against dozens of external threat intelligence feeds. This includes integrations with industry-standard services like sfp_shodan for finding internet-exposed devices, sfp_virustotal for checking files and URLs against antivirus engines, and many other blacklist providers to identify known malicious infrastructure.[1]

Understanding these categories and the key modules within them allows an analyst to move beyond the pre-configured scan profiles and build highly customized scans tailored to the specific objectives of their investigation.

5.2 Unlocking Full Potential: The Critical Role of API Keys

While the open-source version of SpiderFoot is remarkably powerful out of the box, its effectiveness operates on a tiered model. The baseline capabilities, which rely on public web scraping and DNS lookups, are substantial. However, the "professional tier" of data gathering is only accessible by integrating the tool with third-party services via Application Programming Interfaces (APIs). In the SpiderFoot UI, modules that can be enhanced or require an API key are marked with a padlock symbol.[5]

API keys are the credentials that allow SpiderFoot to programmatically request data from specialized external services. Many of the most valuable data sources, particularly in the realm of threat intelligence and data breach analysis, do not make their full datasets available for public scraping and instead require authenticated API access.[11] While SpiderFoot can function effectively without any keys, the richness and depth of the data it can collect are magnified exponentially with each API that is integrated.[26]

For any serious practitioner, configuring API keys is a non-negotiable step to unlocking the platform's full potential. The process typically involves registering for an account with the third-party service (many of which offer generous free tiers for low-volume use), locating the API key in the account dashboard, and then pasting it into the appropriate field within SpiderFoot's "Settings" tab. Investing the time to acquire and configure these keys transforms SpiderFoot from a good OSINT tool into a great one, capable of pulling in data from sources that are inaccessible to non-authenticated queries.

5.3 A Practical Guide to Integrating Key Third-Party APIs

Configuring API keys in SpiderFoot is a straightforward process that dramatically enhances the quality and scope of the intelligence it can gather. The following guide provides instructions for acquiring and integrating keys for some of the most valuable and commonly used services. The general process for all integrations is to navigate to the "Settings" tab in the SpiderFoot web UI, select the desired module from the list on the left, and paste the acquired key into the designated field.

Service Name	Data Provided	Cost Model	Acquisition URL	SpiderFoot Module
SHODAN	Information on internet-exposed devices, services, open ports, and vulnerabilities.	Tiered API	https://www.shodanhq.com	sfp_shodan
VirusTotal	Checks files, URLs, domains, and IPs against over 70 antivirus scanners and domain blacklisting services.	Tiered API	https://www.virustotal.com	sfp_virustotal
HaveIBeenPwned	Checks if an email address has been compromised in known data breaches.	Free API	https://haveibeenpwned.com/API	sfp_pwned
SecurityTrails	Comprehensive DNS history, WHOIS data, domain information, and associated infrastructure.	Tiered API	https://securitytrails.com	sfp_securitytrails
AlienVault OTX	Open Threat Exchange platform for threat intelligence on IPs, domains, and file hashes.	Tiered API	https://otx.alienvault.com/	sfp_alienvault
Hunter.io	Finds email addresses associated with a domain.	Tiered API	https://hunter.io	sfp_hunter
Censys.io	Scans the internet	Tiered API	https://censys.io	sfp_censys

	to provide data on hosts, websites, and certificates.			
GreyNoise	Identifies internet "background noise" and scan traffic to differentiate targeted attacks from mass scanning.	Tiered API	https://www.greynoise.io/	sfp_greynoise
RiskIQ	Attack surface management data, passive DNS, WHOIS, and threat intelligence.	Commercial API	https://www.riskiq.com	sfp_riskiq
BuiltWith	Identifies the technology stack (web servers, frameworks, analytics tools, etc.) used by a website.	Tiered API	https://www.builtwith.com	sfp_builtwith

Step-by-Step Integration Example (SHODAN):

1. **Navigate to the Service:** Open a web browser and go to https://www.shodanhq.com.[26]
2. **Create an Account:** Sign up for a free account or log in if you already have one.
3. **Locate the API Key:** After logging in, click on "Developer Center." Your API key will be displayed prominently in a box on the page.[7]
4. **Copy the Key:** Select and copy the entire API key string.
5. **Configure in SpiderFoot:** In the SpiderFoot web UI, go to "Settings" -> "SHODAN". Paste the copied key into the "API Key" field and click "Save Changes".[26]

This process can be repeated for the other services listed in the table. While some services are commercial, many offer free tiers that are more than sufficient for individual researchers and small-scale investigations, providing access to a wealth of professional-grade data at little to no cost.[26]

Chapter 6: From Data to Intelligence: Analysis and Visualization

6.1 Navigating the Web of Data: The Browse and Graph Views

Once a scan begins to yield results, SpiderFoot provides several interfaces for exploring the discovered information. The primary challenge in any OSINT investigation is moving from a massive collection of raw data points to a coherent, actionable intelligence picture. SpiderFoot's analysis views are designed to facilitate this transition.

The **Browse** tab is the most straightforward way to view results. It presents the data in a categorized list format, with each category corresponding to a specific data type (e.g., "IP Address," "Email Address," "Malicious IP Address").[5] This view is excellent for getting a quick overview of

what was found and for examining the raw details of individual findings. Users can click on any category to see the specific data elements discovered within it. The interface also allows for filtering and sorting, as well as toggling different display formats, such as a "tiled" view for image-heavy results like screenshots, or an "aggregated" view that shows only unique data points and their frequency, which is useful for identifying outliers or highly recurrent data.[27]

While the Browse view is essential for examining individual facts, the **Graph** view is the primary tool for analysis. This view transforms the flat list of data into a dynamic, interactive node-based graph that visualizes the relationships between different entities.[11] Each piece of data is a node, and lines (or edges) connect nodes that are related. For example, a central node representing the target domain might be linked to several IP address nodes, which in turn are linked to open port nodes and WHOIS record nodes.

This visualization is the key to understanding context and making connections that would be nearly impossible to spot in a raw data list. It allows an analyst to instantly identify clusters of activity, shared infrastructure between seemingly unrelated assets, and pivotal data points that bridge different parts of the investigation.[28] By illustrating

how the data is connected, the Graph view moves the user from simple data collection to genuine intelligence analysis, forming the basis of a coherent investigative narrative.

6.2 Tracing the Threads: Understanding Data Lineage and Correlations

A fundamental requirement for credible intelligence is the ability to validate findings and understand their provenance. OSINT is notoriously susceptible to false positives and misleading information, so an analyst must be able to trace the origin of every piece of data.[3] SpiderFoot provides this capability through its robust data lineage feature.

Within the results view, each piece of data is presented alongside the "source data" that was analyzed to find it. By clicking a series of small arrow icons, a user can traverse this discovery path backward, step-by-step, all the way to the original seed target.[30] This creates a transparent and auditable trail for every finding. For example, if a scan flags an IP address as malicious, the analyst can use the lineage feature to see precisely which threat intelligence module(s) reported it. This context is critical: a flag from a single, less-reputable source carries far less weight than a flag that is corroborated by multiple, highly-trusted feeds like AlienVault OTX and GreyNoise. This ability to assess the source and confidence of a finding is essential for building a defensible investigative report.

In addition to lineage, each data element is enriched with valuable metadata that aids in interpretation:

- **Data Type:** The category of the information (e.g., INTERNET_NAME, EMAIL_ADDRESS).
- **Source Module:** The specific module that generated the data (e.g., sfp_shodan).
- **Children:** The number of new data points that were discovered by analyzing this specific element.
- **Correlations:** The number of correlation rules that were triggered by this data point.
- **Distance from Target:** How many "hops" away the data is from the original seed target.[30]

This metadata provides immediate context, allowing an analyst to quickly sort and prioritize results. For instance, a data element with a high number of "children" is clearly a rich source of information and warrants further investigation. Similarly, an element that triggers multiple "correlations" has been automatically identified by the system as being of high interest or risk.

6.3 Advanced Filtering and Custom Database Queries

As scans can generate thousands of data points, the ability to effectively filter and search the results is paramount to managing data overload. SpiderFoot's web UI includes a powerful search bar that operates across the entire scan's dataset. This allows for both simple and complex queries to isolate specific information.[22] The search functionality supports:

- **Exact Value Matching:** Searching for a specific string, such as a known IP address or username.
- **Pattern Matching:** Using wildcards to find data that fits a certain pattern, suchas *:22 to find all hosts with port 22 open.
- **Regular Expressions (Regex):** For highly complex and specific pattern matching, allowing for sophisticated queries to find data formatted in a particular way.[22]

While the UI provides excellent tools for most use cases, the ultimate power for custom analysis lies in the backend SQLite database. This feature acts as an "escape hatch" for power users, ensuring that no analytical question is unanswerable, provided the user has basic SQL proficiency.[1] If an analyst needs to perform a complex, multi-faceted query that is not supported by the UI's filters—for example, "Show me all email addresses found by modules X and Y, but not by module Z, that are also associated with a domain found in a data breach"—they can directly access the scan's
.db file with any standard SQLite client. This provides unrestricted access to the raw, structured data, enabling limitless custom reporting and analysis that goes far beyond the built-in capabilities. This direct database access ensures that the tool's data is fully transparent and extensible for any analytical need.

6.4 Exporting Intelligence for External Analysis (CSV, JSON, GEXF, Neo4j)

SpiderFoot is designed not as a monolithic, closed ecosystem, but as a powerful data collection component that can integrate into a broader intelligence analysis toolchain. This philosophy is evident in its robust data export functionality, which allows users to extract scan results in several standard, open formats for use in other applications.[1]

The primary export formats available directly from the web UI include:

- **CSV (Comma-Separated Values):** Ideal for importing data into spreadsheets like Microsoft Excel or Google Sheets for tabular analysis, sorting, and reporting.[1]
- **JSON (JavaScript Object Notation):** A lightweight, structured data format perfect for programmatic use. JSON exports can be easily ingested by custom scripts or other security tools for further automated processing.[1]
- **GEXF (Graph Exchange XML Format):** A standard format for graph data. Exporting to GEXF allows users to import their scan results into advanced, dedicated graph visualization and analysis platforms like Gephi, enabling more powerful network analysis than the built-in graph view provides.[1]

This commitment to open data formats demonstrates an understanding that users often have preferred tools for specific analytical tasks. By allowing data to be easily moved out of SpiderFoot, the developers empower users to integrate its world-class data collection capabilities into their existing and preferred workflows.

For users requiring the most advanced graph analysis capabilities, the community has developed a dedicated tool, spiderfoot-neo4j, which facilitates the import of SpiderFoot scan data into a Neo4j graph database.[32] Neo4j is a highly scalable and powerful native graph database that offers advanced querying languages (like Cypher) and graph data science algorithms. The

spiderfoot-neo4j tool can even use these algorithms, such as PageRank or Harmonic Centrality, to analyze the graph and suggest new, closely related targets for subsequent scans, further blurring the line between data collection and automated analysis.[32]

Chapter 7: Advanced Operations and Techniques

7.1 Automated Correlation: Writing Custom Rules to Find the Needles in the Haystack

The YAML correlation engine is one of SpiderFoot's most powerful advanced features, enabling users to move beyond simple data collection and into the realm of automated risk identification. While the platform ships with over 37 pre-defined rules for flagging common security issues, its true potential is realized when users write custom rules that codify their organization's specific security concerns and threat model.[4] This transforms SpiderFoot from a general OSINT scanner into a bespoke attack surface monitoring system.

A correlation rule is a YAML file that defines a specific set of conditions to search for within a scan's results. If the conditions are met, a "Correlation Result" is generated, highlighting the finding for the analyst. The structure of a rule is logical and consists of several key sections [18]:

- **meta:** Contains metadata about the rule itself, such as a unique ID, version, title, and a description of the risk it's designed to find.
- **collections:** This is the data extraction part of the rule. It defines one or more queries to pull specific data elements from the backend SQLite database. Queries can be based on data type (e.g., INTERNET_NAME), the module that generated the data (e.g., sfp_shodan), or the content of the data itself using exact matches or regular expressions.
- **aggregation (optional):** This section allows for the grouping of collected data. For example, data could be grouped by IP address or domain name to count occurrences.
- **analysis (optional):** This section performs logical tests on the collected and aggregated data. This is where thresholds can be set (e.g., "flag if count is greater than 5") or outlier analysis can be performed.

To create a custom rule, the recommended practice is to copy the template.yaml file from the correlations directory, rename it, and modify it to suit the new purpose.[4] For example, an organization could create a rule to detect potential subdomain takeover vulnerabilities. The rule's

collections section would query for CNAME DNS records. The analysis section would then check if the target of the CNAME record points to a cloud service known to be vulnerable (e.g., an unclaimed S3 bucket or Heroku app), and if so, generate a high-priority correlation.

By writing custom rules, a security team can automate the triage process, teaching SpiderFoot to proactively alert them to the specific conditions that represent the greatest risk to their unique environment.

7.2 Going Dark: Leveraging TOR for Anonymity

For sensitive investigations, particularly those conducted as part of offensive security engagements or in situations where the target must not be aware of the investigator's identity, operational security (OPSEC) is paramount. SpiderFoot includes a built-in integration with The Onion Router (TOR) network to provide a layer of network-level anonymity for scans.[1]

When enabled in the global settings, this feature routes all of SpiderFoot's direct HTTP/HTTPS requests through the TOR network. This means that for any module that directly contacts the target's servers (e.g., web scrapers, port scanners), the traffic will appear to originate from a random TOR exit node, not from the analyst's true IP address.[21] This is a critical feature for preventing the target from identifying and blocking the source of the scan.

However, it is crucial to understand the capabilities and limitations of this feature. While it effectively anonymizes direct network contact, it is not a magic bullet for complete anonymity. The primary trade-off is performance; routing traffic through the multi-layered TOR network is inherently slower than a direct connection, so scans will take longer to complete.[21]

More importantly, the TOR integration only applies to modules that make direct HTTP/HTTPS requests from within the SpiderFoot framework. It does not anonymize all of SpiderFoot's activity. For instance:

- **API-based Modules:** Modules that query third-party services like SHODAN or VirusTotal will still use the analyst's configured API key. The request to the third-party service will be made from the analyst's IP address, not through TOR. This could potentially deanonymize the investigation to the third-party provider.
- **DNS Queries:** Standard DNS queries may not be routed through TOR, potentially leaking the analyst's IP address to DNS servers.
- **External Tool Calls:** If SpiderFoot is configured to call external tools like Nmap, those tools will run in their own process and will not be routed through TOR unless separately configured to do so.

Therefore, while the TOR feature is an invaluable tool for enhancing OPSEC, it must be used as one component of a broader security strategy. Analysts conducting highly sensitive investigations should consider additional measures, such as running SpiderFoot on a dedicated virtual machine that is itself configured to route all traffic through a secure network.

7.3 Extending the Framework: An Introduction to Custom Module Development

One of SpiderFoot's most compelling features for advanced users and developers is its extensibility. The entire platform is built around its modular architecture, and users have the ability to write their own custom modules in Python to integrate new data sources or add unique analytical capabilities.[14] The commercial SpiderFoot HX platform also supports bringing your own custom Python modules to run in its cloud environment.[4]

Developing a custom module requires a solid understanding of Python and, more critically, a deep understanding of SpiderFoot's event-driven, publisher-subscriber architecture. The process is not trivial, and a "simple" idea can become architecturally complex. A developer cannot simply write a script and plug it in; the module must be designed to correctly interact with the framework's event handling system.[15]

A basic module must inherit from the sflib.SpiderFootPlugin class and define several key properties:

- **_setup() method:** Initializes the module, defining its name, description, the event types it consumes (watchedEvents), and the event types it produces (produces).
- **handleEvent() method:** This is the core logic of the module. It is called every time the module receives an event it has subscribed to. The method receives the event data as

an argument, performs its specific task (e.g., querying an external API, parsing data), and then uses the self.notify() method to publish any new events (new data) it discovers.

The primary challenge, as discussed in Chapter 2, is the single-event nature of the handleEvent() function. A module reacts to one event at a time and does not have native access to the full context of all other data found in the scan. This makes it difficult to build modules that need to aggregate multiple pieces of information before taking action.[15] Developers must either design their modules to work within this single-input paradigm or build their own complex state-management logic to store and correlate data internally as it arrives.

A practical example of a community-developed custom module demonstrates how to query local datasets. This module listens for data types like email addresses or usernames, and when one is found, it searches for that identifier in local files (e.g., downloaded breach compilations) specified by the user in the module's settings.[34] This serves as an excellent template for users looking to integrate their own private or proprietary data sources into the SpiderFoot workflow. For those less experienced with Python, some users have reported success in using Large Language Models (LLMs) to assist in writing modules by providing the model with the relevant API documentation.[35]

Chapter 8: Real-World Applications: From Theory to Practice

8.1 Case Study: Comprehensive Attack Surface Mapping

Objective: To map the external digital attack surface of a target organization, starting with only its primary domain name. The goal is to identify all internet-facing assets, including known, unknown (shadow IT), and potentially rogue infrastructure.

Methodology:

1. **Scan Setup:** A new scan is created in SpiderFoot.
   - **Scan Name:** _Attack_Surface_Map
   - **Seed Target:** The organization's main domain (e.g., example.com).
   - **Scan Profile:** "By Use Case" -> "Footprint." This profile is optimized for asset discovery, enabling modules for DNS enumeration, web scraping, IP lookups, and cloud asset discovery. For a more aggressive scan, modules like sfp_dnsbrute (DNS brute-forcing) and sfp_portscan_tcp (port scanning) can be manually enabled.
2. **Execution and Initial Analysis:** The scan is launched. As results populate, the analyst focuses on key data types in the "Browse" view:

    - **Sub-domains/Hostnames:** This reveals web servers, mail servers (mail.example.com), development environments (dev.example.com), and other infrastructure.
    - **IP Addresses:** All discovered hostnames are resolved to their corresponding IP addresses.
    - **Netblocks/ASNs:** The IP addresses are mapped to their owning network blocks and Autonomous System Numbers, revealing the hosting providers (e.g., AWS, Google Cloud, DigitalOcean) and physical data centers in use.[36]
    - **Cloud Storage:** The sfp_s3bucket_finder and sfp_azureblob_finder modules search for publicly accessible storage buckets named after the target, which are common sources of data leaks.[1]
3. **Identifying Shadow IT and Vulnerabilities:** The true value emerges when analyzing the aggregated data and correlations.
    - The analyst uses the Graph view to visualize the relationships between domains, IPs, and services. This can reveal subdomains hosted on entirely different infrastructure from the main corporate assets, a classic sign of shadow IT.
    - The analyst reviews the "Software Used" and "Web Server" data types. The correlation engine's "Outlier Web Servers" rule automatically flags servers running different software from the norm (e.g., one Apache server in a fleet of Nginx servers), which could be an unmanaged, potentially vulnerable system.[18]
    - The sfp_vuln_scanner modules (integrating with services like Shodan) will flag discovered services with known vulnerabilities (CVEs).

Outcome: The final report, built from the SpiderFoot results, provides the organization with a comprehensive inventory of its external attack surface. It identifies not only the officially managed assets but also uncovers a forgotten development server with an outdated PHP version and a publicly listable S3 bucket containing old marketing materials. This allows the security team to bring the shadow IT assets under management, patch the vulnerabilities, and secure the exposed cloud storage, significantly reducing their overall risk exposure.[37]

8.2 Case Study: Threat Hunting a Malicious Domain

Objective: To investigate a domain (hikmahmuliautama.co.id) reported in a phishing campaign to gather intelligence on the threat actor's infrastructure and potentially identify other malicious assets. This case will leverage the advanced capabilities of SpiderFoot HX.

Methodology:

1. **Scan Setup (Passive First):** To avoid tipping off the threat actor, the initial scan is configured for stealth.
    - **Scan Name:** _Threat_Hunt
    - **Seed Target:** hikmahmuliautama.co.id
    - **Scan Profile:** "By Use Case" -> "Passive." This ensures no direct contact is made with the target's servers. All data is gathered from third-party sources like threat

intelligence feeds, search engines, and public databases.[21] TOR integration is enabled for any modules that might make direct contact, adding another layer of anonymity.[21]

2. **Initial Findings and Pivoting:** The passive scan yields immediate results:
   - **Threat Intelligence:** Multiple modules (sfp_abuseipdb, sfp_virustotal, etc.) flag the domain's IP address as malicious, confirming its involvement in nefarious activities.[23]
   - **WHOIS and DNS:** WHOIS data may be privacy-protected, but historical DNS records from services like SecurityTrails can reveal previous IP addresses or name servers used by the actor.
   - **File Metadata Analysis:** The sfp_filemeta module, even in a passive scan (by querying search engine caches), might find documents hosted on the site. The metadata within these files (author names, creation dates, software used) can provide invaluable clues about the actor.[39]
3. **Automated Correlation and Infrastructure Expansion:** This is where SpiderFoot's ability to pivot shines.
   - **Shared Infrastructure:** The "Co-Hosted Site" data type reveals other domains hosted on the same IP address. These are immediately suspect and can be added as targets for new scans.
   - **Google Analytics Correlation:** A powerful technique involves the sfp_webanalytics module. It identifies the Google Analytics ID used on the malicious site. SpiderFoot then automatically queries services like SpyOnWeb to find *every other domain on the internet* that uses the same analytics ID. This is a highly effective method for definitively linking a threat actor's entire web infrastructure, even when hosted on different servers with different registration details.[39]

Outcome: Starting with a single domain, the analyst uses SpiderFoot to rapidly unmask a network of interconnected malicious sites. The investigation reveals five additional domains linked by a shared IP address and a common Google Analytics ID. The metadata from a PDF found on one site contains a username, which, when used as a seed for a new scan, uncovers social media profiles associated with the threat actor. This body of evidence, all gathered and correlated automatically by SpiderFoot, can be provided to law enforcement or used to create robust blocking rules to protect the organization from the actor's entire campaign.[38]

8.3 Case Study: OSINT for Corporate Due Diligence

Objective: To conduct an open-source due diligence investigation on a potential startup acquisition ("PartnerCorp") to identify any non-obvious risks related to their security posture, reputation, or potential liabilities.

Methodology:

1. **Planning and Scoping:** The objective is to assess PartnerCorp's external risk profile.

The investigation must remain ethical and legal, focusing strictly on publicly available information.[40]

2. **Scan Setup:** A multi-faceted investigation is initiated using several seed targets.
    - **Scan 1 (Infrastructure):**
        - **Target:** partnercorp.com
        - **Profile:** "Footprint" to map their technical assets and security posture.
    - **Scan 2 (Reputation):**
        - **Target:** "PartnerCorp" (as a Human Name/Username)
        - **Profile:** "Investigate" to find mentions on social media, forums, and news sites.
    - **Scan 3 (Breach History):**
        - **Target:** partnercorp.com (as an E-mail Address, to check the domain against breach databases)
        - **Profile:** "By Required Data" -> "Data Breach."
3. **Data Analysis and Risk Identification:** The analyst synthesizes the results from all scans to build a risk profile.
    - **Security Posture:** The infrastructure scan reveals that PartnerCorp's main website is missing key security headers (Content-Security-Policy, Strict-Transport-Security), and their mail server's DNS records lack DMARC and SPF records, making them vulnerable to email spoofing. The port scan identifies an exposed, unauthenticated Redis instance on a development server.
    - **Reputation:** The reputation scan uncovers several negative reviews from former employees on tech forums and identifies the CEO's personal social media accounts, which contain politically charged content that may not align with the acquiring company's brand values.
    - **Data Breach History:** The breach scan gets a hit from the sfp_pwned module, indicating that email addresses using the @partnercorp.com domain have appeared in the "Collection #1" data breach, suggesting a high likelihood of compromised employee credentials being available to threat actors.[41]

Outcome: The OSINT due diligence report, compiled from the SpiderFoot scans, provides the acquiring company with critical risk information not present in the official financial statements. The poor security posture, exposed database, and history of employee credential compromise represent a significant potential liability. The CEO's public statements present a potential brand reputation risk. This information allows the acquiring company to make a more informed decision, potentially leading to a renegotiation of the acquisition price to account for the necessary security remediation costs, or even a decision to walk away from the deal entirely.[40]

8.4 Case Study: Proactive Brand and Reputation Monitoring

Objective: To establish an automated, proactive monitoring system for a fictional brand, "AcmeCorp," to detect brand impersonation, executive impersonation, and reputational threats in near real-time.

Methodology:

1. **Scan Setup (Continuous Monitoring with HX):** This use case is best served by SpiderFoot HX's attack surface monitoring features, which allow for scheduled, recurring scans and automated change notifications.[13]
   - **Targets:** A multi-target scan is configured with several seed types:
     - Domain Name: acmecorp.com
     - Human Name: "John Doe" (CEO's Name)
     - Username: AcmeCorp, AcmeCorpSupport
     - Custom Keyword: "AcmeCorp"
   - **Scan Profile:** A custom profile is created, enabling specific modules relevant to brand protection:
     - sfp_dns_dnstwist: To detect newly registered, typosquatted, or cybersquatted domains (e.g., acmecorpp.com, acme-corp.net).
     - sfp_accounts: To find new social media or forum accounts created using the brand name.
     - sfp_spider: To crawl the web and search for mentions of the brand name and executive names on blogs and forums.
     - sfp_pwned: To monitor for new breaches containing @acmecorp.com email addresses.
   - **Scheduling:** The scan is scheduled to run weekly.
   - **Notifications:** Alerts for any *new* findings are configured to be sent to the security team's Slack channel and email distribution list.[1]
2. **Automated Threat Detection:** The system runs automatically in the background.
   - **Week 1:** The initial scan establishes a baseline of AcmeCorp's known digital footprint.
   - **Week 2:** The scan runs again. SpiderFoot's change detection algorithm compares the new results to the baseline. It finds that a new domain, acmecorp-support.com, has been registered. The sfp_dns_dnstwist module flags this as a potential cybersquatting attempt. A notification is automatically sent to the security team.
   - **Week 3:** The scan detects a new Twitter account named @AcmeCorpHelp that is responding to customer complaints. The sfp_accounts module flags this new account.
3. **Triage and Response:**
   - Upon receiving the notification about acmecorp-support.com, the security team investigates and confirms it is a malicious phishing site designed to steal customer credentials. They initiate a domain takedown process with the registrar.
   - The @AcmeCorpHelp Twitter account is identified as an impersonation account attempting to scam customers. The team reports the account to Twitter for suspension.

Outcome: By using SpiderFoot HX for automated brand monitoring, AcmeCorp transforms its security posture from reactive to proactive. Instead of waiting for a customer to report a phishing email or a scam, the security team is automatically alerted to the creation of malicious infrastructure and impersonation accounts, often before they can be weaponized. This allows for rapid response, minimizes brand damage, and protects customers from fraud.[1]

Chapter 9: The OSINT Landscape: A Comparative Analysis

9.1 SpiderFoot vs. Maltego: Automation vs. Visualization

In the world of OSINT, SpiderFoot and Maltego are two titans, but they occupy different, albeit overlapping, roles in an investigator's toolkit. Understanding their core philosophies—automation versus visualization—is key to leveraging them effectively.

SpiderFoot is fundamentally an **automation engine**. Its primary strength lies in its ability to cast an incredibly wide net, automatically querying over 200 data sources with minimal user intervention.[43] The workflow is one of "fire and forget": the user provides a target, selects a broad scan profile, and SpiderFoot works tirelessly in the background to collect a massive volume of data. It answers the question, "

What is out there?" Its purpose is to handle the laborious, time-consuming data collection phase of an investigation, freeing the analyst from manually querying dozens of websites and APIs.[17] While it has visualization capabilities, its core competency is automated data aggregation.

Maltego, on the other hand, is a premier **link analysis and visualization platform**. Its strength is not in broad, automated collection but in providing a powerful, interactive canvas for an analyst to manually explore and map relationships between data points.[44] Maltego operates using "Transforms," which are small scripts that take one piece of data as input and retrieve related information. The workflow is iterative and analyst-driven. An investigator might start with a domain, run a transform to get an IP address, then run another transform on the IP to find other domains, visually building out the graph step-by-step. It excels at answering the question, "

How is this all connected?".[47]

The choice between them is not "either/or" but "when and why." They are highly complementary and are often used in conjunction. A common and highly effective workflow is to use SpiderFoot for the initial, broad data collection phase. Once SpiderFoot has generated its comprehensive report, the analyst can export the key findings and import them into Maltego for a deeper, more granular manual investigation, using Maltego's superior visualization engine to uncover complex relationships that the automated tool might have missed.[45]

9.2 SpiderFoot vs. theHarvester: Breadth vs. Specificity

Comparing SpiderFoot to theHarvester is a study in the trade-off between comprehensive breadth and lightweight specificity. Both are excellent open-source reconnaissance tools, but they are designed for different scales of investigation.

theHarvester is a **lightweight, fast, and highly focused tool**. Its purpose is specific: to gather email addresses, subdomains, virtual hosts, and employee names associated with a target domain from a curated list of public sources like search engines and PGP key servers.[43] It is the digital equivalent of a scalpel, designed to perform a quick, initial reconnaissance sweep to gather low-hanging fruit. It is excellent for the very first stage of a penetration test or security assessment where the goal is to quickly establish a basic footprint of the target.[44]

SpiderFoot, in contrast, is a **comprehensive, broad-spectrum framework**. It is the Swiss Army knife of OSINT. While it performs all the same functions as theHarvester, that capability is just one small part of its overall feature set. SpiderFoot extends its reach to hundreds of other data types and sources, including threat intelligence feeds, data breach records, social media platforms, cloud storage, dark web searches, and much more.[43] Its goal is not just to find emails and subdomains, but to build the most complete intelligence profile possible on the target.

An analyst's choice between the two depends on the immediate need. For a quick, targeted query to find email addresses for a potential phishing campaign, theHarvester is an excellent and efficient choice. For a deep, exhaustive investigation into a target's entire digital ecosystem, SpiderFoot is the far more powerful and appropriate tool. As with Maltego, the tools can be used together. An analyst might run theHarvester to get a quick list of subdomains, and then feed that entire list into SpiderFoot as multiple targets for a deep and comprehensive scan.

The following table provides a strategic summary of these three leading OSINT tools, enabling a practitioner to select the right tool for the job based on their investigative requirements.

Characteristic	SpiderFoot	Maltego	theHarvester
Primary Function	Automated, broad-spectrum data collection and correlation.	Manual, interactive link analysis and data visualization.	Focused, lightweight collection of emails, subdomains, and hosts.
Strength	Massive automation across 200+ data sources; "fire and forget" operation.	Unparalleled graph visualization for uncovering complex relationships; highly extensible via custom Transforms.	Speed and simplicity for initial reconnaissance; very easy to use.
Weakness	Can produce data overload; less intuitive	Steeper learning curve; Community Edition has	Limited scope of data sources and types;

	for deep manual link analysis.	limitations; less suited for broad, automated collection.	lacks deep analysis features.
Typical Use Case	Comprehensive attack surface mapping; threat infrastructure investigation; initial phase of a deep-dive investigation.	Deep-dive analysis of complex networks (e.g., cybercrime, fraud); visualizing relationships from data collected by other tools.	Quick, initial footprinting of a target domain during a penetration test; gathering a list of potential email targets.
Data Sources	[43]	[44]	[43]

Chapter 10: Operational Best Practices and Ethical Considerations

10.1 Managing False Positives and Data Overload

A direct consequence of SpiderFoot's power is the sheer volume of data it can generate. A comprehensive scan on a large target can easily return tens of thousands of data elements. This presents two inherent challenges for the analyst: data overload and false positives. As one source aptly puts it, SpiderFoot "casts a wide net, not a sniper rifle".[3] Success with the tool requires strategies to manage this deluge of information.

The first line of defense against data overload is to define a clear objective before starting a scan. Running an "All" modules scan on every target is inefficient and often counterproductive. By using more targeted scan profiles—such as "By Required Data" or a custom selection of modules—the analyst can significantly reduce the amount of irrelevant noise from the outset.[16]

Once a scan is complete, the correlation engine is the next critical tool for triage. By automatically flagging high-risk or anomalous findings, it helps the analyst focus on the most likely areas of interest, saving them from manually sifting through every result.[4]

However, no amount of automation can completely eliminate the need for human verification. False positives are an unavoidable reality of OSINT.[3] A username scan might flag an account that belongs to a different person with the same name, or a threat feed might list an IP address that was malicious in the past but is now clean. The data lineage feature is invaluable here, allowing the analyst to check the source and context of a finding.[30] Ultimately, SpiderFoot is an intelligence *gathering* assistant, not an intelligence *analysis* replacement. The final, critical steps of interpretation, correlation, and manual verification rest with the human analyst, whose expertise is required to transform the tool's raw data into credible, actionable intelligence.[3]

10.2 The Legal and Ethical Boundaries of OSINT

The practice of OSINT operates in a complex legal and ethical gray area. While the discipline is defined by its use of publicly available information, the term "publicly available" does not grant an investigator carte blanche to collect and use that data for any purpose.[40] Responsible and professional use of a powerful tool like SpiderFoot requires a firm understanding of and adherence to legal and ethical boundaries.

The primary legal consideration is that laws regarding data privacy, computer access, and surveillance vary significantly by jurisdiction. An action that is perfectly legal in one country may be a criminal offense in another. Practitioners must be aware of the laws that apply to their location, their target's location, and the location of the data they are accessing.[3]

From an ethical standpoint, the key principle is to operate with a clear scope of authority. For penetration testers and security assessors, this means having a signed contract that explicitly grants permission to perform reconnaissance on a client's systems.[8] For corporate investigators or threat intelligence analysts, it means operating within the bounds of company policy and for legitimate business purposes, such as due diligence or threat hunting.

Using OSINT to harass, intimidate, or stalk individuals is unethical and illegal. Even when investigating a legitimate target, the analyst must consider the privacy implications of collecting and storing personally identifiable information (PII). A passive scan, which avoids direct contact with the target, is often the most ethically sound starting point for an investigation.[16] The power of a tool like SpiderFoot comes with a significant responsibility to wield it ethically, respecting privacy and operating within a well-defined legal and professional framework.

10.3 Maintaining and Updating Your SpiderFoot Instance

To ensure its continued effectiveness, a self-hosted SpiderFoot instance requires regular maintenance. The OSINT landscape is in constant flux: websites change their structure, breaking web scrapers; services update their APIs, requiring module adjustments; and new data sources emerge. SpiderFoot is not a "set it and forget it" tool.[3]

The most critical maintenance task is to keep the software up to date. For users who installed via git clone, this is a straightforward process: navigate to the SpiderFoot directory and run git pull to download the latest changes from the official repository. After pulling the updates, it is essential to re-run the pip install -r requirements.txt command to install any new or updated Python library dependencies that the new version of the tool may require.[3]

API key management is another ongoing task. API keys can expire, or services may change their access policies. Periodically reviewing the configured keys in the "Settings" tab to ensure they are still valid is good practice.

Finally, over time, the local SQLite database can grow very large with old scan data. While storage is cheap, large database files can slow down the UI and the query process. Periodically, an analyst may wish to archive or delete old scan data from the "Scans" tab to keep the instance performing optimally. For users who find this maintenance burdensome, the commercial SpiderFoot HX platform handles all of these tasks automatically as part of its managed service offering.[1]

Appendix: The Complete Module and API Reference

This appendix serves as a quick-reference guide to a selection of the most notable modules available in SpiderFoot. It allows practitioners to rapidly understand the capabilities of each module, the type of data it provides, and whether it requires an API key for full functionality.

Module Name (sfp_*)	Description	Data Source Link	API Requirement
sfp_abusech	Checks if a host, domain, or IP is malicious according to Abuse.ch feeds.	https://www.abuse.ch	Free API
sfp_abuseipdb	Checks if an IP address is malicious according to the AbuseIPDB.com blacklist.	https://www.abuseipdb.com	Tiered API
sfp_accounts	Searches for associated user accounts on nearly 200 websites (Reddit, eBay, etc.).	N/A	Internal
sfp_ahmia	Searches the Ahmia search engine for mentions of the target on the TOR network.	https://ahmia.fi/	Free API
sfp_alienvault	Obtains threat intelligence from AlienVault Open Threat Exchange (OTX).	https://otx.alienvault.com/	Tiered API
sfp_archiveorg	Identifies historic versions of webpages and files from the Wayback Machine.	https://archive.org/	Free API
sfp_arin	Queries the ARIN registry for network	https://www.arin.net/	Free API

	ownership and contact information.		
sfp_binaryedge	Obtains data from BinaryEdge.io on breaches, vulnerabilities, and passive DNS.	https://www.binaryedge.io/	Tiered API
sfp_bing	Uses the Bing search engine to identify subdomains and related links.	https://www.bing.com/	Tiered API
sfp_bitcoinabuse	Checks Bitcoin addresses against a database of suspect/malicious addresses.	https://www.bitcoinabuse.com/	Free API
sfp_builtwith	Queries BuiltWith.com to identify a website's technology stack.	https://www.builtwith.com	Tiered API
sfp_censys	Obtains information on hosts, websites, and certificates from Censys.io.	https://censys.io	Tiered API
sfp_dns_dnstwist	Calls the DNSTwist tool to find typosquatted and cybersquatted domains.	N/A	External Tool
sfp_dns_zonetransfer	Attempts to perform a full DNS zone transfer against a target's nameservers.	N/A	Internal
sfp_email	Extracts email addresses found in scraped web content.	N/A	Internal
sfp_filemeta	Extracts metadata from documents, images, and other binary files.	N/A	Internal
sfp_fullcontact	Gathers domain and email information from FullContact.com.	https://fullcontact.com	Commercial API
sfp_github	Identifies associated	https://github.com	Tiered API

	public code repositories on GitHub.		
sfp_greynoise	Queries GreyNoise to identify internet-wide scanning traffic.	https://www.greynoise.io	Tiered API
sfp_hunter	Uses Hunter.io to find email addresses associated with a domain.	https://hunter.io	Tiered API
sfp_haveibeenpwned	Checks email addresses against the HaveIBeenPwned data breach service.	https://haveibeenpwned.com	Free API
sfp_ipinfo	Queries ipinfo.io for geolocation and network data about an IP address.	https://ipinfo.io	Tiered API
sfp_nmap	Calls the Nmap tool for advanced TCP/UDP port scanning and service identification.	N/A	External Tool
sfp_portscan_tcp	Performs a basic TCP port scan on identified IP addresses.	N/A	Internal
sfp_riskiq	Queries RiskIQ's passive DNS and threat intelligence databases.	https://riskiq.com	Commercial API
sfp_s3bucket_finder	Searches for potential Amazon S3 buckets associated with a target.	https://aws.amazon.com/s3/	Free API
sfp_securitytrails	Queries SecurityTrails for historical DNS data and domain intelligence.	https://securitytrails.com	Tiered API
sfp_shodan	Queries SHODAN for internet-exposed devices, services, and vulnerabilities.	https://www.shodanhq.com	Tiered API
sfp_spider	A web spider that crawls websites to find	N/A	Internal

	links, emails, and other data.		
sfp_subdomain_takeover	Checks for CNAME records pointing to unclaimed cloud services, indicating a takeover vulnerability.	N/A	Internal
sfp_twitter	Searches Twitter for mentions of the target and associated user profiles.	https://twitter.com	Tiered API
sfp_virustotal	Checks IPs, domains, and file hashes against the VirusTotal malware database.	https://www.virustotal.com	Tiered API
sfp_whatweb	Calls the WhatWeb tool to identify technologies used on a website.	N/A	External Tool
sfp_whois	Performs a WHOIS lookup to find registration details for a domain or IP.	N/A	Internal

Data Sources: [1]

Works cited

1. SpiderFoot automates OSINT for threat intelligence and mapping your attack surface. - GitHub, accessed June 18, 2025, https://github.com/NextKool/Spiderfoot
2. spiderfoot | Kali Linux Tools, accessed June 18, 2025, https://www.kali.org/tools/spiderfoot/
3. Spiderfoot OSINT Made Simple: Fast Install and Recon on Mac/Linux - DevDigest, accessed June 18, 2025, https://www.samgalope.dev/2025/05/05/spiderfoot-osint-made-simple-fast-install-and-recon-on-mac-linux/
4. smicallef/spiderfoot: SpiderFoot automates OSINT for threat ... - GitHub, accessed June 18, 2025, https://github.com/smicallef/spiderfoot
5. Beginners guide to SpiderFoot - Hackercool Magazine, accessed June 18, 2025, https://www.hackercoolmagazine.com/beginners-guide-to-spiderfoot/
6. Lessons learned from my 10 year open source Python project - Reddit, accessed June 18, 2025, https://www.reddit.com/r/Python/comments/smta85/lessons_learned_from_my_1

0_year_open_source/

7. Spiderfoot Documentation | PDF | Domain Name System - Scribd, accessed June 18, 2025, https://www.scribd.com/document/352062606/Spiderfoot-Documentation
8. spiderfoot tutorial - OSINT tool - Sweshi's Cyber Security Tutorials, accessed June 18, 2025, https://sweshi.com/CyberSecurityTutorials/Penetration%20Testing%20and%20Ethical%20Hacking/spiderfoot%20tutorial.php
9. Cyber Threat Intelligence Buy: Intel 471 Acquires SpiderFoot - | MSSP Alert, accessed June 18, 2025, https://www.msspalert.com/news/cyber-threat-intelligence-buy-intel-471-acquires-spiderfoot
10. Intel 471 Acquires SpiderFoot - PR Newswire, accessed June 18, 2025, https://www.prnewswire.com/news-releases/intel-471-acquires-spiderfoot-301665787.html
11. OSINT: Using Spiderfoot for OSINT Data Gathering - Hackers Arise, accessed June 18, 2025, https://hackers-arise.com/osint-using-spiderfoot-for-osint-data-gathering/
12. Open Source Intelligence (OSINT) Automation Tool "Spider Foot HX ..., accessed June 18, 2025, https://www.tegakari.net/en/2022/03/spiderfoot-hx/
13. spiderfoot - Automate OSINT for, accessed June 18, 2025, https://romanr301.github.io/spiderfoot/spiderfoot/index.html
14. README.md · kali/master · undefined - spiderfoot - GitLab, accessed June 18, 2025, https://gitlab.com/kalilinux/packages/spiderfoot/-/blob/kali/master/README.md
15. Custom PIPL Module · Issue #1077 · smicallef/spiderfoot - GitHub, accessed June 18, 2025, https://github.com/smicallef/spiderfoot/issues/1077
16. Information Gathering using Spiderfoot: A Practical Walkthrough - Infosec Train, accessed June 18, 2025, https://www.infosectrain.com/blog/information-gathering-using-spiderfoot-a-practical-walkthrough/
17. Getting Started With Spiderfoot – A Beginner's Guide - Nixintel, accessed June 18, 2025, https://nixintel.info/osint-tools/getting-started-with-spiderfoot/
18. spiderfoot/correlations/README.md at master - GitHub, accessed June 18, 2025, https://github.com/smicallef/spiderfoot/blob/master/correlations/README.md
19. Gather all information about target with Spiderfoot - Studytonight, accessed June 18, 2025, https://www.studytonight.com/post/gather-all-information-about-target-with-spiderfoot
20. SpiderFoot – A Automate OSINT Framework in Kali Linux | GeeksforGeeks, accessed June 18, 2025, https://www.geeksforgeeks.org/spiderfoot-a-automate-osint-framework-in-kali-linux/
21. Spiderfoot HX: Investigating A Domain Used For Phishing - Nixintel, accessed June 18, 2025,

https://nixintel.info/osint-tools/spiderfoot-hx-investigating-a-domain-used-for-phishing/
22. Spiderfoot - Offensive Security Cheatsheet, accessed June 18, 2025, https://cheatsheet.haax.fr/open-source-intelligence-osint/tools-and-methodology/frameworks-automated/spiderfoot/
23. Using Spiderfoot HX To Investigate A Malicious IP Address - Nixintel, accessed June 18, 2025, https://nixintel.info/osint-tools/using-spiderfoot-hx-to-investigate-a-malicious-ip-address/
24. FullHunt Open-Source: Integration with Amass + SpiderFoot, accessed June 18, 2025, https://fullhunt.io/blog/2021/12/07/fullhunt-integration-with-amass-spiderfoot.html
25. SpiderFoot 3.0: OSINT reconnaissance tool - Andrea Fortuna, accessed June 18, 2025, https://andreafortuna.org/2020/02/07/spiderfoot-3-0-osint-reconnaissance-tool/
26. How to Use SpiderFoot for OSINT Gathering - Null Byte, accessed June 18, 2025, https://null-byte.wonderhowto.com/how-to/use-spiderfoot-for-osint-gathering-0180063/
27. Visualizing, Filtering and Sorting Data in SpiderFoot HX - YouTube, accessed June 18, 2025, https://www.youtube.com/watch?v=ys4F_WHDce0
28. Spiderfoot Guide | OSINT & Recon | Kali Linux - YouTube, accessed June 18, 2025, https://www.youtube.com/watch?v=5SbJZKAoJcQ
29. Spiderfoot Training Video · Issue #1140 - GitHub, accessed June 18, 2025, https://github.com/smicallef/spiderfoot/issues/1140
30. Understanding SpiderFoot HX Scan Results - YouTube, accessed June 18, 2025, https://www.youtube.com/watch?v=-UtFI5a7Zfo
31. How to do OSINT with Spiderfoot - YouTube, accessed June 18, 2025, https://www.youtube.com/watch?v=cgw0IgmiqSk
32. Import, visualize, and analyze SpiderFoot scans in Neo4j, a graph database - GitHub, accessed June 18, 2025, https://github.com/blacklanternsecurity/spiderfoot-neo4j
33. spiderfoot | Knowledgebase, accessed June 18, 2025, https://knowledgebase.beehive.systems/tools/osint/spiderfoot
34. Spiderfoot - Beyond the Web · - Security by Accident, accessed June 18, 2025, https://security-by-accident.com/beyond-the-web-spiderfoot/
35. SpiderFoot w/ TruePeopleSearch Custom Module / Feedback and Ideas Needed - Reddit, accessed June 18, 2025, https://www.reddit.com/r/OSINT/comments/1friq1o/spiderfoot_w_truepeoplesearch_custom_module/
36. Mapping the attack surface of telecommunication networks from the public internet - Aaltodoc, accessed June 18, 2025, https://aaltodoc.aalto.fi/bitstreams/61ea526a-a513-46d8-81c8-c0ffb55ade7c/download
37. Unveiling Your External Attack Surface - Perspective Intelligence, accessed June

18, 2025, https://perspectiveintelligence.co.uk/unveiling-your-external-attack-surface-an-in-depth-exploration-of-attack-surface-intelligence/

38. Using Spiderfoot to combat domain name abuse/security threats, accessed June 18, 2025, https://realtimeregister.com/blog/using-spiderfoot-to-combat-domain-name-abuse-security-threats/
39. Nixintel Open Source Intelligence & Investigations Crypto Scam ..., accessed June 18, 2025, https://nixintel.info/osint-tools/crypto-scam-investigation-using-spiderfoot-hx-for-osint-automation/
40. How to Use the OSINT Framework: Sources, Tools, & Steps - Bitsight, accessed June 18, 2025, https://www.bitsight.com/learn/cti/osint-framework
41. 7 Top OSINT Software Tools | Liferaft, accessed June 18, 2025, https://liferaftlabs.com/blog/7-top-osint-software-tools
42. The Benefits Of OSINT Services And Tools For Businesses - Corma Investigations, accessed June 18, 2025, https://corma-investigations.com/series/osint-lookdeeper/understanding-the-different-open-source-intelligence-osint-services-and-tools-and-how-they-benefit-businesses/
43. OSINT Tools for Expert Intelligence Gathering - Tech with JD - Techie, accessed June 18, 2025, https://blog.jadhusan.com/osint-tools-for-expert-intelligence-gathering/
44. Top 7 OSINT Tools Every Cybersecurity Professional Should Know | The Complete Guide, accessed June 18, 2025, https://www.webasha.com/blog/top-7-osint-tools-every-cybersecurity-professional-should-know-the-complete-guide
45. What are your preferred OSINT tools? : r/hacking - Reddit, accessed June 18, 2025, https://www.reddit.com/r/hacking/comments/vr7bah/what_are_your_preferred_osint_tools/
46. 15 Best OSINT tools in 2025 - Lampyre, accessed June 18, 2025, https://lampyre.io/blog/2025/03/11/15-best-osint-tools-in-2025/
47. Top 15 OSINT Tools for Expert Intelligence Gathering - Recorded Future, accessed June 18, 2025, https://www.recordedfuture.com/threat-intelligence-101/tools-and-technologies/osint-tools
48. Top 10 open-source intelligence platforms for amateur spies - Cyber Magazine, accessed June 18, 2025, https://cybermagazine.com/top10/The-Top-10-open-source-intelligence-platforms-for-amateur-spies
49. SHADOW INTELLIGENCE WITH OSINT AND SPIDERFOOT - ijrpr, accessed June 18, 2025, https://ijrpr.com/uploads/V6ISSUE4/IJRPR42883.pdf