花山熊強 – ブルムバーグ バルンハルツ[†]

† 奈良先端大, 先端科学技術研究科
〒630–0101 奈良県生駒市高山町 8916–5
E-mail: †bernhards.blumbergs.bb2@is.naist.jp

A Multiple Vantage Point-based Concept for Open-Source Information Space Awareness

Bernhards BLUMBERGS[†]

† Graduate School of Science and Technology, Nara Institute of Science and Technology
8916–5 Takayama, Ikoma, Nara 630–0192, Japan
E-mail: †bernhards.blumbergs.bb2@is.naist.jp

Abstract The proposed concept bolsters visibility and provides context to cyber incident response teams countering malicious activities, such as online fraud, phishing, information leakage, defacement, and disinformation. Simultaneously accessing the same information source from multiple vantage points within the Internet may provide different results and analysis of identified changes may benefit incident response efforts. A single snapshot employs various connection methods, extracts a set of data from the information source, cross-correlates acquired data, and represent identified changes between the various vantage points of the same information source. The analysis of collected snapshots over a period of time represents the dynamics of an information source and identifies anomalies.

Key words Open-source information, information source interaction, information space analysis, situational awareness, incident response

1. Introduction

Information space monitoring has become a prominent area with increasing importance in establishing capabilities, such as situational awareness, disinformation monitoring, data breach identification, threat intelligence gathering, and tracking cyberattacks. As the demand for situational awareness and information processing increases, this field has become highly competitive with worldwide companies offering various information space monitoring and analysis solutions [1]. Cyber threat intelligence services [2] [3], open-source and dark-net data collection solutions [4] [5], and social network information aggregation platforms [6] [7] represent a saturated and ever-growing demand for data collection and analysis for a broad range of purposes. The service providers will use a variety of approaches to collect, analyze, and represent the data, which will depend on the data source itself, its specific use case, and intelligence consumer requirements. Such data acquisition sources may include metadata and telemetry analytics collected from vendor's products, deploying sinkholes and honeypots, or collecting information from clear-net, deep-net, or dark-net resources. A comprehensive cyber incident response would employ a multi-faceted approach through aggregation and parsing of various information sources for establishing as complete situational awareness as it is possible. Existing solutions and tools provide a high level of visibility, deliver valuable cyber threat intelligence, and are an integral part of incident response capability for a computer emergency response team (i.e., CERT, CSIRT, or CIRC) based on operational requirements, solution capabilities, and availability. Existing solutions aimed at monitoring and collecting data from online reachable data sources, such as forums, social network posts, webpage content, and dark-net marketplaces, will interact with the target source from a single position on the Internet. Such a position typically is not considered or is chosen by the information collector depending on reasons, such as resource availability for node deployment, closer proximity to the source of information, or maintaining a level of anonymity. Although valid information from the given single position on the Internet for further processing and analysis may be received, it is strictly limited to that particular perspective and how the information source responds to it.

The author proposes a hypothesis, that an information source may provide a different response based on the way how it is being accessed, and it may reveal the dynamic nature of the information source and deliver valuable contextual information to enrich the incident response process. In this work, a term of *vantage point* is understood as a combination of a connection method and the position within the Internet from which an information source is being accessed. A vantage point would employ a specific access method with the intent to observe a change in response by the information source. Accessing an information source from a different geographically assigned IP address space, using well-known VPN or proxy service brokers, routing through anonymization network exit nodes, or changing connection parameters (e.g., user-agent) may produce responses with varying results. Receiving varied results might be also expected due to well-accepted practices and techniques, such as geographical load balancing (e.g., cloud services), serving regionally relevant content (e.g., regional entity or display language change), or limiting content access (e.g. streaming services). By deploying a set of collectors, the data from the information source may be collected simultaneously from multiple vantage points (Fig. 1). A term of *snapshot* is used to represent a collection of responses from the same information source via the used multiple vantage points. In this technical report, a work-in-progress is described towards developing a concept for multiple vantage point-based interaction with open-source information resources to provide information source behavior change analysis relevant to the incident response team activities. The proposed approach is aimed at increasing the incident response team capabilities and in further detail is discussed in chapter 3..

This technical report is structured as follows: chapter 2. gives an overview summary of related work in the field and chapter 3. presents the proposed conceptual model and its core functionality.

2. Related Work

Related work in the field of open-source data gathering and information space awareness considers academic research, vendor whitepapers, and solution descriptions, as well as technical blogs and write-ups. The search was conducted by the use of Internet academic search engines (e.g., Google Scholar, Semantic Scholar) and academic databases (e.g., IEEE Xplore, ACM, Springer, and Elsevier). A primary search time frame of the last five years was selected to represent the latest developments and advances in this constantly evolving field, although earlier relevant works may also be considered if referenced by the reviewed papers. This section focuses on the following relevant aspects 1) information source response dynamics, 2) website content mining approaches, and 3) website content extraction and analysis.

Based on a variety of used keywords and within the defined scope of the search criteria, no related work was identified, which would recognize or analyze the information source variations depending on the used access method. Identified and further reviewed papers do not consider or acknowledge the possible changes in the retrieved content and employ a single connection method limited to available resources, such as the researcher's university or private network connections, public cloud or virtual private server providers, or use of anonymization network (e.g., Tor). Although reasonably valid data may be extracted in this manner it is limited to a single vantage point and in a broader scope may provide incomplete results introduced by the bias of the information source response nature.

The dynamic nature of the websites, content delivery mechanisms, and inclusion of paid content pose a challenge for identifying and extracting the actual data, which represents the main intended information of the webpage. There has been ongoing research in this area and multiple approaches have been proposed for content extraction from websites also considering dynamic content extraction, social networks, and user-generated content [8] [9] [10] [11] [12] [13]. The majority of researchers recognize text processing, webpage document object model (DOM) parsing, and natural language processing techniques to extract the data from dynamic web applications [14] [15]. Some researchers rely on crowdsourcing the content identification to human operators (e.g., Amazon Mechanical Turk) [16].

Acquired website content analysis introduces a broad variety of approaches based on the analytical requirements leading up to cyber threat intelligence gathering within clear-net and dark-net websites. Identified research includes approaches, such as, text and web mining approaches with semantic textual patterns and predictive modeling [17]; time series-based modeling for predicting website content changes and future dynamics [18]; Google Analytics keywords-based similarity analysis to derive insight into website content usability [19]; graph theory and linear algebra to support the discovery of patterns and intelligence within existing datasets [20]; malicious website detection opportunities through HTTP header and page content feature analysis [21] [22]. Two notable works, aligned with the research scope of this technical report, are oriented toward the analysis of dark-net resources. Lawrence et.al. [23] proposes a modular *D-miner* framework specialized in dark-net website data mining. The framework relies on Python and its third-party libraries, such as Selenium, Requests, and BeutifulSoup4. Interaction with dark-net marketplaces is implemented by adjusting crawling speeds not to trigger the rate-limiter and crowdsourcing CAPTCHA solving to a paid *DeathByCaptcha* service. Samtani et.al. [24] present an *AZSecure Hacker Assets Portal* to collect, analyze, and report dark-net data sources contributing to cyber threat intelligence into hacker intentions and motivations, employed cyber assets, and improved situational awareness. The authors provide an overview of the dark-net data sources and assess their intelligence value, and introduce the crawling and anti-crawling technique countermeasures to collect data from hacker forums, marketplaces, shops, and IRC

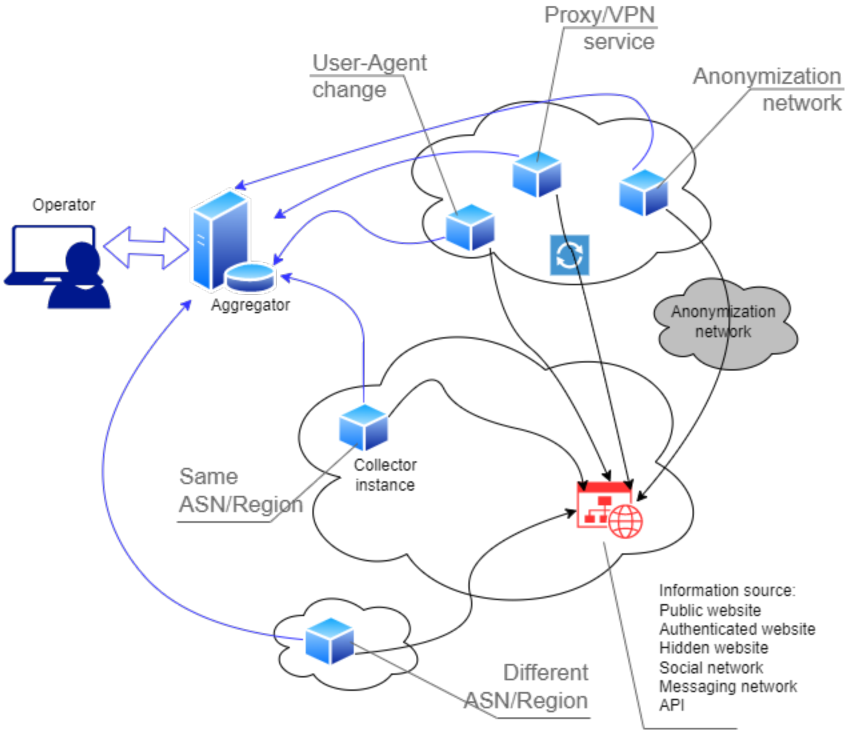

図 1　Information source interaction via multiple vantage points

channels.

The field of website data mining, information extraction, and analytics has been well-researched and presents multiple significant practical and theoretical contributions. However, the described data mining approaches and solutions have been focused on reaching the source from a single vantage point and working with acquired data to derive further intelligence. While such an approach is fully acceptable and would yield desired in-depth results in most cases, however, it limits the analyst on comprehending the dynamics of the information source when accessed from multiple vantage points with various techniques (e.g., change of IP address space, various connection methods, and HTTP user-agent variations). Such a multiple-vantage point-based approach creates a snapshot of the same resource under different vantage point conditions and presents the differences to the analyst for further decision-making and intelligence gathering. This work builds upon existing work in the field and expands it with an additional perspective to permit CERT teams to maintain better situational awareness over their defined information space.

3. Concept Description

This section introduces the core design considerations and implementation approaches and presents the current work in progress toward proof-of-concept prototype development.

The proposed proof-of-concept introduces an approach of a multiple vantage point snapshot (Fig. 1), where the contents of the same information source are simultaneously collected from various positions on the global Internet in this work referred to as *vantage points*. A complete set of such simultaneously collected information source representations in this work is referred to as a *snapshot*. A snapshot represents the dynamic nature of the same information source from different vantage points. Further automated analysis of the data may identify and represent any changes at a given time between the instances within the same snapshot. Furthermore, the creation of multiple snapshots over time against the same data source and its cross-correlation among the past snapshots may represent the nature of changes in the data source over the period of time. The identified changes, in a structured and visual format, are represented to a human analyst for further analysis and evaluation. The human analyst is provided with the identified changes and detailed information related to a sequence of snapshots, a single snapshot, or a particular vantage point collector engaged in data collection. This work limits its scope to multi-vantage point-based data collection, snapshot change analysis and correlation, and representation to the analyst. Analysis and deriving the meaning of the changes in most cases will be context-driven, dependent on the information source under analysis, and the reasons for engaging in the conduct of data collection.

The main focus areas of the approach fall within the following categories:

（1） identification of changes within a snapshot. From an incident response perspective, for example, this may permit assessing and tracking online fraud and phishing activities by observing how a web resource changes based on an employed vantage point. A malicious or compromised web resource may be serving its content to users originating from a particular geographically allocated IP address space, which would provide the contextual information of the likely targeted victims, scope, and adversarial intentions;

（2） identification of changes between the snapshots. From a disinformation-tracking perspective, for example, minute changes in the web resource's existing content over a period of time may reveal modifications in conveyed information context, as well as lead up to the detection of website defacement or take-down activities;

（3） detection of artifacts within snapshot instances. From a data leakage and asset tracking perspective, for example, may permit tracking of particular keywords or digital artifact hashes to detect their use or leakage. That may include also the organization's user accounts linked to affiliated domain names.

The proof-of-concept codebase is written in Python3 language, and its functionality and implementation consist of the following stages: 1) collector deployment, 2) collector configuration, 3) information source interaction, 4) data collection, 5) snapshot data reporting, and 6) snapshot processing. These stages are represented in Fig. 2 and are described in detail further in this chapter. Parts of the proof-of-concept source code will be released by the author publicly under the GNU GPLv3 license on GitHub (*https://github.com/lockout*).

Collector engine uses Docker container technology to create a self-contained environment, enable flexible automation via Docker Compose, and to permit scalable cloud-based deployments (e.g., Amazon Web Services, Google Cloud Platform, and Microsoft Azure). The collector instance is developed on top of the *Python:latest* Docker image and upon every deployment is fully updated, installs required base system software packages and Python dependencies, configures the environment, and sets up the required collector Python source files. The collector handles both the underlying connection establishment (i.e., the vantage point) and data collection from the target information source. The data connection types include the implicit deployment IP range within the cloud services assigned region, Tor anonymization network, VPN connection, and the use of proxy services. At the Docker instance system service level, Tor and OpenVPN daemons are customized based on the received configuration profile in the next stage and are managed from the Python module using *subprocess.Popen*. The proxy server connection may be established either at the system service level or by configuring the web connection properties from Python configuration, however, the latter option is the preferred one to ensure control and management of proxy connections and avoid system data leaking through those connections. To establish a vantage point for the collector, the following approaches are considered and implemented (Fig. 1): 1) geographically allocated IP address space deployments (e.g., cloud or virtual private server hosting services), 2) network proxy services (e.g., public or private HTTP, HTTPS, or SOCKS servers), 3) VPN services (e.g., public or private OpenVPN or WireGuard servers), 4) anonymization networks with Internet exit nodes (e.g., local Tor router as a SOCKS proxy), and 5) change of HTTP User-Agent (e.g., various mobile or desktop browser user-agents). A single collector instance is capable of establishing all implemented connection methods and vantage point modifications to interact with the target data source. Although different technologies, the use of regional IP address space deployments, connection redirectors (e.g., *socat* relay [25]), and VPN and proxy services in essence accomplish the same goal of changing the outbound IP address through which the connection to the information source is being established. The use of public or commercial VPN or proxy services may trigger the information source to respond differently if it is configured to detect well-known services. Such behavior may be desired from a broader data collection perspective. In specific cases, self-deployed bespoke redirector, VPN, or proxy services may be used, however, to lower the complexity of the infrastructure and its management overhead it is expected that dedicated collector instance deployments are used in the preferred IP address space as distinct vantage points instead of self-deployed connection broker instances. To acknowledge broader use-case possibilities and illustrate the flexibility of the proposed approach, an infrastructure of geographically distributed deployments of redirectors, proxies, and VPN services may permit the use of a single collector instance to reach the target information source through a variety of vantage points.

A JSON-based configuration is used to define the required variables and settings needed for the collector instance, collection process, and interaction with the information source. JSON, being a widely accepted standard, permits human- and machine-readable configuration file creation, detailed customization, and future expansion of additional requirements. The collector instance configuration profile contains four main sections: 1) meta information, 2) task management configuration, 3) connection preferences, and 4) data collection settings. Also, snapshot data reporting is represented in a structured schema in JSON format. This format has been chosen due to its wide support and parsing compatibility. A significant consideration is to enable the collected JSON-formatted result ingestion into other tools used by the incident response team, to permit a flexible integration, data parsing, and representation from systems and their dashboards already in use. The collected result report contains two main sections – meta information and collected data. The collector configuration profile is split into multiple sections – meta information, task management and reporting, connection establishment, and data collection (i.e., harvesting). The data collection has a few subcategories related to web session establishment and control, content collection, and artifact detection. As an example, a collector profile provides the configuration and required variables and has the following structure, where tags < > (not part of a valid JSON syntax) are used to represent a variable or Boolean value specified by the analyst:

```
{
"profile_meta" : {
    "syntax_version" : "<version>",
    "profile_id" : "<profileID>",
    "profile_description" : "<description>",
```

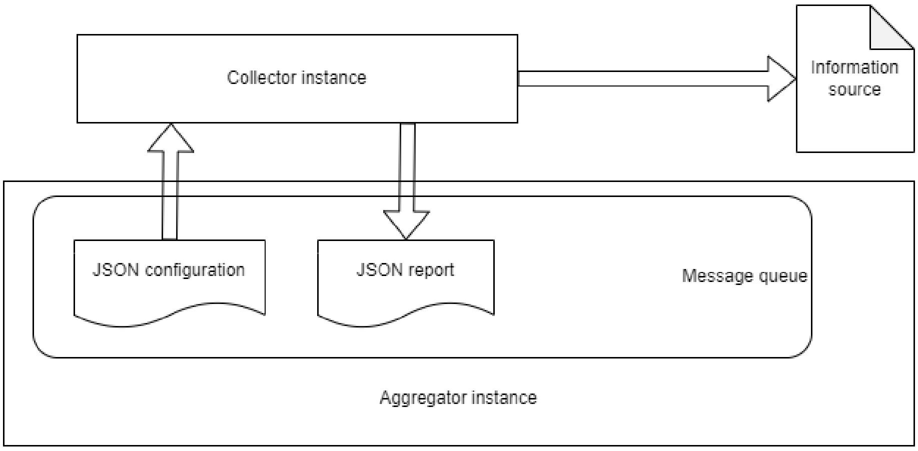

図 2　Stages of the collection process flow

```
        "timestamp" : <timestamp>
    },
    "manager" : {
        "task_id" : "<taskID>",
        "task_count" : <iterations>,
        "task_sleep" : <sleepTime>,
        "queue_url" : "<url>",
        "queue_access" : "<accessToken>"
    },
    "connector": {
        "ip_region" : "<identifier>",
        "proxy" : "<proxyUri>",
        "tor" : <bool>,
        "vpn" : [ <vpnSettings> ]
    },
    "harvester" : {
        "harvest_url" : [
            "<url-1>",
            "<url-2>",
            "<url-n>"
        ],
        "harvest_allowredirect" : <bool>,
        "harvest_requesttimeout" : <seconds>,
        "harvest_depth" : <depth>,
        "harvest_useragent" : "<userAgemt>",
        "harvest_privatemode" : <bool>,
        "harvest_headers" : <bool>,
        "harvest_timer" : <bool>,
        "harvest_timeout" : <seconds>,
        "content_html" : <bool>,
        "content_data" : <bool>,
        "content_image" : <bool>,
        "content_imagesize" : [
            <horizontal>,
            <vertical>
            ],
        "content_image_optimize" : <bool>,
        "content_image_jpeg" : <bool>,
        "content_image_quality" : <quality%>,
        "content_sha256" : <bool>,
        "content_ssdeep" : <bool>,
        "artefact_collect" : <bool>,
        "artefact_pattern" : "<regex_pattern>",
        "artefact_sha256" : <bool>,
        "artefact_ssdeep" : <bool>,
        "artefact_keywords" : [
            "<keyword1_regex-1>",
            "<keyword2_regex-n>"
            ]
        }
    }
```

The main component is the information source interaction and data collection. A Google Chromium headless browser is used and automated by the use of Selenium [26]. Mozilla Firefox and Google Chromium browsers were evaluated for the use of website interaction and Selenium automation, however, the Chromium browser proved to have better and easier support for parameter specification via command line arguments (e.g., User-Agent, network proxy). Although the information source interaction may be accomplished by the sole use of the Python Requests framework [27], an additional layer of interaction provided by a browser may be required to capture website visual content and in the future support attempts to detect and solve CAPTCHA and perform user behavior emulation. The prototype implements the use of both Selenium and Requests, depending on the configuration profile, and may transfer session parameters between both. Various metadata and content data are collected from the target information source. The choices of the retrieved data are related to the unique identification of the target information source, its parameters, and served content. Collected resource metadata consists of a visited URL which may differ from the initial one in case of HTTP redirection, HTTP header data (e.g., session cookies, identifiers, used user agent, and session parameters), the SSL certificate, and interaction time measurement. This collected metadata may allow additional assessment of the information source (e.g., passive SSL or DNS lookups) and identification of its unique behavior in relation to the employed vantage point. Collected information source data represents its response to the used vantage point and is collected for further parsing, analysis, and information extraction. If the target information source is a website, HTML content is extracted, its SHA256 hash and *ppdeep* fuzzy-hash [28] is calculated and recorded. These hashes are used for further analysis of the extracted content between other instances within the same snapshot. While SHA256 may be used to identify precisely the same content, the context-triggered piecewise hash (CTPH), known as a fuzzy-hash – provides an estimation of similarity to identify subtle deviations. Although multiple fuzzy-hashing algorithms and implementations exist (e.g., *ssdeep* [29]), *ppdeep* has been chosen due to lower base system and Python library dependency requirements and is based on the *SpamSum* [30] algorithm upon which derivations, such as *ssdeep* [31] have been developed. Additional HTML content analysis, such as, regex-based detection of keywords or artifacts may be carried out by using BeautifulSoup library [32]. Although additional crawling of the data source may be performed up to the specified depth, it is not considered to be feasible since such activity would yield large volumes of collected data, add overhead to data analysis, and make it harder for the analyst to assess the captured results. Instead, in some specific cases, it may be considered to create a list of URLs for the specific data source and provide them in the collector configuration file. A website graphical representation (i.e., screenshot) is collected via Selenium Chromium web driver, which permits not only textual content-based assessment of the information source but also allows a visual comparison, change detection, and evaluation by the use of third-party libraries, such as PIL/Pillow [33] or Open Computer Vision [34]. The default screenshot is captured in a high-resolution lossless PNG format but may be converted to a JPG with additional compression. The screenshots are included in the JSON report in Base64 encoded format. The author believes that humans are better at evaluating visual data instead of structured textual information, therefore the analyst is presented with a snapshot, where each vantage point capture instance is represented by its screenshot. This visual information may give immediate feedback in case severe changes may be observed, for example, information source significant alterations or defacement. Selecting a single capture instance displays the retrieved textual data, which includes the metadata, content data, and highlights identified regex-based keywords. If two capture instances either within the same snapshot or between two different snapshots are selected, the visual differences and their structural similarity measure (SSM) between the primary and secondary images are assessed and displayed, as well as, computed content hash differences and textual data diff being represented alongside it.

A central aggregator instance serves as the main interface point between the various collectors and a human analyst. The anticipated functionality of the aggregator is the following: 1) analyst user interface, 2) collector instance management, and 3) collected result representation. The web user interface permits the analyst to complete and manage all tasks related to the collector instance management and collected result analysis. The collector instance management would include activities, such as collector instance customization and automated cloud deployment/removal, task profile creation and assignment to collector instances, and running task management. The aggregator and collector interaction is handled through a *Nats* [35] Docker instance – a lightweight and resilient messaging queue for real-time data streaming with *Nats Jetstream*. The aggregator will put the assigned tasks on the queue and the collectors, at predefined time intervals, will push their status report on the queue, retrieve assigned tasks, and report collected results as soon as they are complete. To ensure that collected results may be properly identified and grouped by the aggregator as a snapshot representing the same task, each collector instance, upon initializing the task, will create a unique collector identifier (i.e., worker ID) based upon a combination of variables – collector deployed location identifier (e.g., geographical location, IP address region, or service provider name), connection profile type (e.g., IP connection, proxy, VPN, Tor), assigned configuration profile ID, and performed task ID – *connectorIpRegion:agentConnection:profileId:taskId*. If the assigned task has multiple iterations, then a separate field for the task identifier will combine the assigned task ID with its iteration count – *taskId:cycle*. The UNIX Epoch time is used as a timestamp for each collector report and may be further used to sort and assess the results. As an example, a collector instance session report consists of single or multiple interaction reports if multiple URLs have been provided within the same task and has the following structure, where tags < > (not part of a valid JSON syntax) are used to represent a variable populated during the initialization or runtime:

```
sessionReport = {
    "task_meta" : {
        "syntax_version" : "<version>",
        "worker_id" : "<worker_id>",
        "task_id" : "<task_id}>,
        "timestamp" : <timestamp>
    },
    "task_data" : [
        <interactReport-1>,
        <interactReport-2>,
        <interactReport-n>
    ]
}
interactReport = {
    "meta" : [
        <url>,
        <userAgent>,
        <HTTPheaders>,
        <sessionCookies>,
        <sslFingerprint>,
        <hashSha256>,
        <hashPpdeep>,
        [<startTime>, <endTime>]
    ],
    "content" : [
        <HTMLcontent>,
        <resourceImage>
    ]
}
```

At the moment of technical report writing, the prototype is in active development and the aggregator only implements the message queue for issuing tasks and collecting results. However, the upcoming prototype development milestones in priority order are as follows:

（1） central aggregator base system design with considerations towards data collection and parsing efficiency. It is anticipated to store the collected snapshot textual data in a folder structure on a ZFS file system [36], which implements data deduplication at the file system level;

（2） collected data representation in web UI and snapshot data graphical and textual comparison and change identification;

（3） content mining algorithm implementation to extract the actual information by excluding all the unnecessary clutter, such as web page navigation and metadata, banners, and paid content;

（4） extraction of data from other reachable sources within the clear-net, deep-net, and dark-net. This may include sources, such as social networks (e.g., Twitter), messaging platforms (e.g., Telegram groups), dark-net websites (e.g., .onion content), or data acquisition via API calls. Such sources may require slightly different gathering methodology when compared to web pages due to their more dynamic nature and access restrictions;

（5） identifying and solving CAPTCHA to overcome content access restrictions;

（6） artifact and their hash collection from the target information source, which may include any hosted files or resources. Such collection may be conducted either by performing regex-based filename matching or collecting specific types of files and calculating their hash match against a provided value. At this moment, although having considerable benefits, it is not anticipated to deliver the collected files themselves to the central aggregator as it may increase storage requirements. Instead, a successful match would be reported with a corresponding URL pointing to that resource;

（7） collector automated deployment, configuration, and management from the web UI.

4. Acknowledgements

This research is conducted under the Japan Society for the Promotion of Science (JSPS) fellowship program and supported by the KAKENHI grant.

文献

[1] M. Moore, “The cyber 100: Cybersecurity companies you should know,” https://onlinedegrees.sandiego.edu/top-100-cybersecurity-companies/. Accessed: 2023/05/23.

[2] Crowdstrike, “Threat intelligence,” https://www.crowdstrike.com/products/threat-intelligence/. Accessed: 2023/05/23.

[3] Trellix, “Global threat intelligence,” https://www.trellix.com/en-us/products/global-threat-intelligence.html. Accessed: 2023/05/23.

[4] Mandiant, “Digital threat monitoring,” https://www.mandiant.com/advantage/digital-threat-monitoring. Accessed: 2023/05/23.

[5] Palantir, “Platforms,” https://www.palantir.com/platforms/. Accessed: 2023/05/23.

[6] Flashpoint, “Flashpoint cyber threat intelligence,” https://flashpoint.io/ignite/cyber-threat-intelligence/. Accessed: 2023/05/23.

[7] Outpost24, “Social media threat intelligence,” https://outpost24.com/products/cyber-threat-intelligence/social-media-monitoring. Accessed: 2023/05/23.

[8] M. Pujar and M.R. Mundada, “A systematic review web content mining tools and its applications,” International Journal of Advanced Computer Science and Applications, pp.••–••, 2021.

[9] M.A. Russell and M. Klassen, Mining the Social Web, Third Edition, O’ Reilly Media, Inc., California, 2018.

[10] J. Gibson, B. Wellner, and S. Lubar, “Adaptive web-page content identification,” Proceedings of the 9th Annual ACM International Workshop on Web Information and Data Management, p.105–112, WIDM ’07, Association for Computing Machinery, New York, NY, USA, 2007.

[11] E. Suganya and S. Vijayarani, “Content based web search and information extraction from heterogeneous websites using ontology,” International Journal of Scientific and Technology Research, pp.••–••, 2020.

[12] K.A. Neuendorf, The Content Analysis Guidebook, SAGE Publications, Inc., California, 2017.

[13] Y.-D. Song, M. Gong, and A. Mahanti, “Measurement and analysis of an adult video streaming service,” Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, p.489–492, ASONAM ’19, Association for Computing Machinery, New York, NY, USA, 2020.

[14] W. Nadee and K. Prutsachainimmit, “Towards data extraction of dynamic content from javascript web applications,” 2018 International Conference on Information Networking (ICOIN), pp.750–754, 2018.

[15] J. Alarte, D. Insa, J. Silva, and S. Tamarit, “Main content extraction from heterogeneous webpages,” WISE, pp.••–••, 2018.

[16] J. Kiesel, F. Kneist, L. Meyer, K. Komlossy, B. Stein, and M. Potthast, “Web page segmentation revisited: Evaluation framework and dataset,” Proceedings of the 29th ACM International Conference on Information and Knowledge Management, p.3047–3054, CIKM ’20, Association for Computing Machinery, New York, NY, USA, 2020.

[17] D. Thorleuchter and D. Van denPoel, “Using webcrawling of publicly available websites to assess e-commerce relationships,” 2012 Annual SRII Global Conference, pp.402–410, 2012.

[18] M.C. Calzarossa and D. Tessera, “Analysis and forecasting of web content dynamics,” 2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA), pp.12–17, 2018.

[19] S. Kohli, S. Kaur, and G. Singh, “A website content analysis approach based on keyword similarity analysis,” Proceedings of the The 2012

IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Volume 01, p.254–257, WI-IAT '12, IEEE Computer Society, USA, 2012.

[20] C.K. Leung, E.W. Madill, and S.P. Singh, "A web intelligence solution to support recommendations from the web," IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, p.160–167, WI-IAT '21, Association for Computing Machinery, New York, NY, USA, 2022.

[21] J. McGahagan, D. Bhansali, M. Gratian, and M. Cukier, "A Comprehensive Evaluation of HTTP Header Features for Detecting Malicious Websites," 2019 15th European Dependable Computing Conference (EDCC), pp.75–82, 2019.

[22] J. McGahagan, D. Bhansali, C. Pinto-Coelho, and M. Cukier, "A Comprehensive Evaluation of Webpage Content Features for Detecting Malicious Websites," 2019 9th Latin-American Symposium on Dependable Computing (LADC), pp.1–10, 2019.

[23] H. Lawrence, A. Hughes, R. Tonic, and C. Zou, "D-miner: A framework for mining, searching, visualizing, and alerting on darknet events," 2017 IEEE Conference on Communications and Network Security (CNS), pp.1–9, 2017.

[24] S. Samtani, W. Li, V. Benjamin, and H. Chen, "Informing cyber threat intelligence through dark web situational awareness: The azsecure hacker assets portal," Digital Threats, vol.2, no.4, pp.••–••, oct 2021.

[25] E. Amoany, "Getting started with socat, a multipurpose relay tool for linux," https://www.redhat.com/sysadmin/getting-started-socat. Accessed: 2023/06/18.

[26] Software Freedom Conservancy, "Selenium automates browsers," https://www.selenium.dev/. Accessed: 2023/05/26.

[27] K. Reitz, "Requests: HTTP for Humans," https://requests.readthedocs.io/. Accessed: 2023/05/26.

[28] M. Ulikowski, "ppdeep," https://github.com/elceef/ppdeep. Accessed: 2023/05/26.

[29] J. Kornblum, "ssdeep Project. ssdeep - Fuzzy hashing program," https://ssdeep-project.github.io/ssdeep/index.html. Accessed: 2023/06/15.

[30] A. Tridgell, "SpamSum README," https://www.samba.org/ftp/unpacked/junkcode/spamsum/README. Accessed: 2023/06/15.

[31] J. Kornblum, "Identifying almost identical files using context triggered piecewise hashing," Digital Investigation, vol.3, pp.91–97, 2006. The Proceedings of the 6th Annual Digital Forensic Research Workshop (DFRWS '06).

[32] L. Richardson, "Beautiful soup," https://www.crummy.com/software/BeautifulSoup/. Accessed: 2023/05/29.

[33] J.A. Clark, "Pillow, the friendly PIL fork," https://python-pillow.org/. Accessed: 2023/05/29.

[34] OpenCV, "OpenCV," https://opencv.org/. Accessed: 2023/05/29.

[35] Cloud Native Computing Foundation, "Connective Technology for Adaptive Edge & Distributed Systems," https://nats.io/l. Accessed: 2023/06/16.

[36] OpenZFS, "OpenZFS," https://openzfs.org/l. Accessed: 2023/06/16.