UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS

OSINT-Based Data-Driven Cybersecurity Discovery

“ Documento Definitivo”

Doutoramento em Informática

Fernando Baptista Leal Alves

Tese orientada por:
Prof. Doutor Alysson Neves Bessani
e pelo Prof. Doutor Pedro Miguel Frazão Fernandes Ferreira

Documento especialmente elaborado para a obtenção do grau de doutor

2023

UNIVERSIDADE DE LISBOA

FACULDADE DE CIÊNCIAS

OSINT-Based Data-Driven Cybersecurity Discovery

Doutoramento em Informática

Fernando Baptista Leal Alves

Tese orientada por:

Prof. Doutor Alysson Neves Bessani

e pelo Prof. Doutor Pedro Miguel Frazão Fernandes Ferreira

Júri:

Presidente:

- Manuel João Caneira Monteiro da Fonseca, Professor Associado com Agregação e Presidente do Departamento de Informática, da Faculdade de Ciências da Universidade de Lisboa

Vogais:

- Doutora Annalisa Appice, Full Professor da Università degli studi di Bari Aldo Moro (Itália)
- Doutor Henrique João Lopes Domingos, Professor Associado da Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa
- Doutor Bruno Emanuel da Graça Martins, Professor Associado do Instituto Superior Técnico da Universidade de Lisboa
- Doutor Alysson Neves Bessani, Professor Associado com Agregação da Faculdade de Ciências da Universidade de Lisboa (orientador)
- Doutora Ibéria Vitória De Sousa Medeiros, Professora Associada da Faculdade de Ciências da Universidade de Lisboa

Documento especialmente elaborado para a obtenção do grau de doutor

Este trabalho foi financiado pelo projecto Europeu H2020 DiSIEM (H2020-700692), pela Fundação para a Ciência e a Tecnologia (FCT) através do projecto ThreatAdapt (FCT-FNR/0002/2018), e pela unidade de investigação LASIGE (UIDB/00408/2020 e UIDP/00408/2020).

2023

Agradecimentos

Em primeiro lugar quero agradecer aos meus orientadores por me suportarem no desenvolvimento deste trabalho. Trabalhei com o professor Alysson anos vários antes do início do meu doutoramento, e ao longo de toda esta parceria aprendi imenso sobre como organizar, desenvolver, e escrever um trabalho científico. Ele incutiu-me com o incisivo sentido crítico dele, que já traz e irá certamente irá trazer sucesso no meu futuro. O professor Pedro foi fundamental para a conclusão bem-sucedida deste trabalho ao trazer um louvável rigor e cuidado na aplicação das várias técnicas usadas, bem como na escrita dos vários documentos produzidos ao longo destes anos. Foi também alguém com quem aprendi muito sobre o desenvolvimento científico do ponto de vista experimental e exploratório, e de avaliação das melhores técnicas a usar conforme um dado contexto. Ambos têm o meu profundo agradecimento, e espero também ter deixado algo de valor convosco.

Ainda ligado a este trabalho, quero agradecer ao Eric, ao André, e ao Nuno, colegas que colaboraram comigo no desenvolvimento de partes desta tese. Com os vossos contributos, este trabalho ficou mais rico.

Quero também agradecer à Joana e à Rita, que tiveram a (mui extensa) paciência que eu acabasse este trabalho, e que apostaram em mim apesar da falta do término dos meus estudos.

Por fim, menciono as restantes pessoas que me suportaram durante estes anos por estarem presentes e fazerem a diferença. Da minha família agradeço ao meu pai, aos meus tios Manuel e Lúcia, e às minhas primas Bruna, Inês e Paula, por todo o suporte que me deram e que continuam a dar. Finalmente, quero agradecer aos meus amigos Adriano, André Lamúrias, André Santos, Beatriz, Caetano, Cristina, Diana, Diogo, Francisco, Freezer, Inês Modesto, Inês Pampulha, Joana Matos, Joana Serra, João Silva, João Sousa, Mariana Barbosa, Mariana Fernandes, Miguel Garcia, Miguel Santos, Nuno, Pedro, Pimpão, Rafa, Rita, Rúben, Simões, Sofia, Susana, Tiago, Vasco, Vinicius, Vlada, e Yana. Obrigado por todas as horas de companhia e parvoíce, muito necessárias todos os dias.

Por fim, quero deixar uma menção especial para o meu irmão Daniel, que tem sido uma presença ubíqua na minha vida.

Para o Daniel.

Abstract

Cybersecurity is a topic of growing concern as the number and gravity of cyber-attacks are continuously increasing. Receiving the latest updates, patches, and news is crucial to maintaining an IT infrastructure's high-security level. An alternative to purchasing expensive security news feeds is to collect Open Source Intelligence: a wealth of knowledge published daily by users, security companies, researchers, and hackers, among others. In particular, Twitter has become an information hub for obtaining cutting-edge information about many subjects, including cybersecurity. This thesis is focused on the collection and processing of cybersecurity-related tweets. Firstly, we conducted a qualitative and quantitative study about the security data found on Twitter and compared it to databases that publish confirmed vulnerabilities or exploits. Our study shows that Twitter is a relevant cybersecurity source. The remainder of the work is about developing a framework for collecting, processing, and delivering security tweets. Its pipeline comprises text filtering, text feature extraction, a binary classifier, clustering, and Indicator of Compromise generation. We show how to obtain a tweet classifier model following tweet characteristics and machine learning best practices. Our clustering strategy adopts the k-means algorithm to an unknown number of clusters, and to cluster and update based on a stream of tweets instead of the classical batch operation. From the clusters we generate Indicators of Compromise, which are structured data formats used in cybersecurity; this step eases the integration of our tool with existing cybersecurity tools. Finally, we showcase one such integration with the Security Information and Event Management system of a nation-wide electrical utility company.

Keywords: Open Source Intelligence, Cybersecurity, Security Operations Centre, Twitter

Resumo

A cibersegurança tem vindo a atrair cada vez mais as atenções devido ao crescimento contínuo do número e da gravidade dos ataques efectuados. Para proteger eficazmente um sistema informático é necessário receber continuamente notícias e eventos relacionadas com cibersegurança, incluíndos novas vulnerabilidades e ataques. Uma alternativa viável a subscrições pagas (dado que muitas têm preços elevados) é obter esta informação através de fontes abertas (*Open Source Intelligence [OSINT]*), visto que especialistas em cibersegurança publicam diariamente vastos conteúdos sobre o tópico. O *Twitter* encontra-se em destaque como plataforma de *OSINT* por ser um agregador natural de conteúdos, incluíndo cibersegurança.

Esta tese foca-se na recolha e processamento de *tweets* que falam sobre cibersegurança. Primeiro, efectuámos um estudo qualitativo e quantitativo sobre as informações de cibersegurança publicadas no *Twitter*, e comparamos estes resultados com as informações presentes em bases de dados dedicadas a armazenar vulnerabilidades e ataques; o nosso estudo mostra que o *Twitter* é uma fonte relevante e completa de informação sobre cibersegurança. O restante trabalho foi dedicado ao desenvolvimento de uma plataforma dedicada à recolha, processamento, e agregação de *tweets* sobre cibersegurança. A nossa plataforma é composta por fases de processamento de texto, conversão numérica, classificação binária, agregação, e criação de indicadores de segurança (*Indicators of Compromise [IoC]*). Primeiro, mostramos como criar um modelo de classificação adequado para *tweets* usando boas práticas de aprendizagem automática. Depois, criámos uma nova estratégia de aplicação do algorítmo *k-means* de modo a não ser necessário definir previamente o número de grupos a obter, ao mesmo tempo que a agregação é feita sobre um fluxo contíno de *tweets* e não sobre um conjunto estático. A partir destes grupos geramos *IoCs* para que a nossa plataforma possa facilmente alimentar outras ferramentas de cibersegurança. Por fim, demonstramos a integração da nossa plataforma com o sistema de gestão de eventos de cibersegurança de uma fornecedora eléctrica nacional.

Palavras chave: *Open Source Intelligence*, cibersegurança, Centro de operações de segurança, *Twitter*

Resumo Alargado

A cibersegurança tem vindo a atrair cada vez mais atenções devido ao crescimento contínuo do número e da gravidade dos ataques efectuados. Para proteger eficazmente um sistema informático, os analistas de segurança necessitam de receber continuamente notícias e eventos relacionadas com cibersegurança e estar a par das últimas vulnerabilidades descobertas, dos novos ataques e do impacto que causam. Também precisam de estar a par de novas versões do *software* que empregam para poderem gerir eficazmente as actualizações e configurações do *software* em uso na infraestrutura informática que monitorizam. Efectuar mudanças no *software* instalado nos equipamentos implica várias fases operacionais, tal como testes de compatibilidade entre versões, além de que o esforço das equipas de operações para criar e instalar novos pacotes tem um custo associado. Ao ter uma visão sobre o panorama de cibersegurança relacionado com a infraestrutura monitorizada, os analistas conseguem prioritizar os sistemas a serem actualizados e coordenar esforços eficazmente com as equipas de operações.

Existem empresas especializadas que recolhem, processam, e enriquecem informação sobre cibersegurança, de forma a facilitar a analise das equipas de segurança. Contudo, estas empresas costumam pedir preços altos por este serviço, que consequentemente, não está ao alcance de todos, especialmente de empresas mais pequenas ou com orçamento limitado para cibersegurança. Uma alternativa viável a subscrições pagas é obter esta informação através de fontes abertas (*Open Source Intelligence [OSINT]*). Diariamente são publicados por especialistas em cibersegurança, investigadores, *hackers*, entre outros, vastos conteúdos sobre os últimos avanços na àrea; como tal, recolhendo *OSINT* é possível obter visibilidade sobre a àrea. O *Twitter* encontra-se em destaque como plataforma de publicação e recolha de *OSINT* por ser um agregador natural de conteúdos, incluíndo cibersegurança.

Esta tese foca-se na recolha e processamento de *tweets* que falam sobre cibersegurança. Primeiro, efectuámos um estudo qualitativo e quantitativo sobre as

informações de cibersegurança publicadas no *Twitter*, e comparamos estes resultados com as informações presentes em bases de dados dedicadas a armazenar vulnerabilidades e ataques. O nosso estudo teve três principais resultados: nenhuma base de dados de vulnerabilidades se destaca em termos de completude ou em ser a primeira a publicar, e como tal, devem ser usadas complementarmente; o *Twitter* é uma fonte completa de informação sobre cibersegurança, e fala sobre vulnerabilidades tão cedo quanto as fontes dedicadas; e que o *Twitter* é uma fonte complementar de dados de cibersegurança dado incluir ligações secundárias que analisam as ameças em muitas das suas publicações, e porque nalguns casos publica sobre vulnerabilidades antes de qualquer uma das outras fontes estudadas.

O restante trabalho foi dedicado ao desenvolvimento de uma plataforma dedicada à recolha, processamento, e agrupamento de *tweets* sobre cibersegurança. Primero, dedicámo-nos à recolha e classificação de *tweets*; para tal, é necessário fazer uma pre-preparação da plataforma: recolher um conjunto de contas de *Twitter* a utilizar como fontes de dados (de preferência contas dedicadas a cibersegurança), e um conjunto de palavras-chave que vão ser usadas como filtro. Ao recolher *tweets* apenas de contas pre-seleccionadas aumentamos o volume de dados potencialmente relevante para os analistas e reduzimos a probabilidade de o sistema classificar erradamente estes *tweets*. As palavras-chave são utilizadas para filtrar os *tweets* recolhidos; este filtro deve ser configurado descrevendo todos os elementos em uso na infraestrutura informática monitorizada de modo a remover *tweets* irrelevantes para os analistas. Os *tweets* que passam o filtro são convertidos num formato numérico adequado para a sua utilização por algorítmos de aprendizagem automática—neste caso utilizámos o algoritmo *Term Frequency – Inverse Document Frequency*. Uma vez em formato numérico, os *tweets* são alimentados a um algoritmo de classificação binária. Neste trabalho testámos dois algorítmos: *Support Vector Machines*, e uma *Multi-Layer Perceptron Neural Network*. Uma avaliação quantitativa utilizando mais de 195.000 *tweets* recolhidos durante oito meses mostra que a nossa proposta consegue classificar correctamente a maioria dos *tweets* relacionados com cibersegurança (taxa de verdadeiros positivos acima dos 90%), enquando descarta incorrectamente uma pequena percentagem de *tweets* (taxa de falsos negativos abaixo dos 10%).

Uma vez que conseguimos seleccionar com alta precisão que *tweets* devem ser apresentados aos analistas, debruçámo-nos sobre como agregar e apresentar os *tweets* seleccionados. A agregação de *tweets* que discutem o mesmo assunto evita a apresentação de informação repetida, um passo crucial para que os analistas possam usar a plataforma de forma eficaz e sem perderem tempo com alertas redundantes. Como tal, desenvolvemos um novo método para agregar um fluxo de *tweets*, considerando a validade temporal dos grupos gerados. O nosso algoritmo de agregação de fluxo de *tweets* baseado no algoritmo *k-means* trouxe três inovações. (1) Não é necessário definir o número de grupos a obter pelo algoritmo—este valor é encontrado dinamicamente durante a execução do mesmo. (2) O algoritmo não descarta *outliers*, algo que as propostas semelhantes à nossa presentes na literatura fazem; nós argumentamos que um sistema dedicado a ciberseguraça não pode descartar *outliers* dado que estes são, com elevada probabilidade, *tweets* que falam acerca de novos assuntos e de interesse aos analistas. (3) Por fim, o nosso algoritmo cria um monitor dos assuntos activos ao considerar que a duração da validade dos dados é gerida por grupo e não por *tweet* (os algoritmos de agrupamento de fluxo associam um prazo de validade a cada ponto, e os pontos "frescos" são agrupados continuamente de modo a que o algoritmo possa gerir ao longo do tempo que pontos que devem pertencer a que grupo); as propostas da literatura associam uma validade a cada elemento (isto é, a cada *tweet*), o que no caso de cibersegurança, quereria dizer que poderíamos remover *tweets* de grupos (assuntos) que ainda estão a sofrer actualizações, visto que o período de discussão de um assunto pode variar entre poucos dias (por exemplo, notícias sobre uma nova versão) a mais de um mês (por exemplo, discussão sobre uma vulnerabilidade crítica). O nosso algoritmo só remove um grupo do monitor quando este deixa de ter novos *tweets* por mais de uma semana, isto é, deixou de haver discussão sobre este assunto.

A partir dos grupos gerados no passo anterior geramos *Indicators of Compromise (IoCs)*, uma estrutura de dados usada em cibersegurança para partilha de informação. Assim, podemos integrar a nossa plataforma de recolha e processamento de *OSINT* com sistemas de gestão de cibersegurança, como plataformas de partilha de vulnerabilidades ou *Security Information and Event Management (SIEM)*. Por fim, demonstramos a integração da nossa plataforma com o *SIEM* de uma fornecedora eléctrica nacional. Esta integração serviu para validar a

nossa plataforma e demonstrar a sua utilidade ao ligar os *IoCs* gerados a partir de *tweets* com eventos recolhidos pelo *SIEM*, e assim enriquecer a informação presente no *SIEM* e fornecer a fonte de vulnerabilidades detectadas.

Palavras chave: *Open Source Intelligence*, cibersegurança, Centro de operações de segurança, *Twitter*

Contents

List of Figures

List of Tables

1
Introduction

The Internet has become an ubiquitous presence in our lives. The number of connected goods are continuously increasing, and now our lamps, coffee machines, and fridges are connected to the Internet. However, every connected device is a target for cyber-attacks, no matter the device's nature or use; there are numerous reports of botnet agents detected in medical devices [137], the Ukrainian power grid was partially shut down via a cyber-attack [59], and even our televisions are vulnerable [40]. These are just a few examples of devices that were not targetable remotely before the connectivity trend, and could only be attacked through (much harder to get) physical access.

Cybersecurity has become a necessity as ubiquitous as the Internet. Every entity must take appropriate security measures to adequately protect themselves from attacks from embedded devices, personal computers, and large companies with thousands of machines. However, the level of complexity for maintaining a system secure dramatically varies with the types of apparatus; personal computers are manageable by a non-specialized user, but small embedded devices (like sensors) may require specialized staff and equipment. Moreover, the number and heterogeneity of devices in use greatly increase the complexity and cost of management. A dedicated team is required to deploy updates for large Information Technology (IT) infrastructures, since there are many administrative considerations in place, such as the purpose of each machine, software compatibilities, the support given by the software developers (*e.g.*, operating system support), and available downtimes.

Cybersecurity also encompasses detecting and reacting to attacks. Here, once again, the challenge is more significant for companies with large IT footprints, who, due to their heterogeneous infrastructure, have machines with widely different exposure and attack sur-

faces. While dedicated staff can keep the various systems updated, it is impossible to manually correlate events happening on such infrastructure and organize a proper and coordinated response.

To streamline the cybersecurity management of organizations with a wide IT infrastructure, Security Information and Event Management (SIEM) systems [95] can be employed. SIEMs aggregate event data produced by operating systems, applications, security devices, and the network infrastructure. The primary data source is log data, but SIEMs can also ingest data in other formats, such as NetFlow [25] or STIX [50]. The input data is normalized and combined with contextual information about users, assets, and vulnerabilities, thus allowing many different types of correlations, analyses, and reports. SIEMs can detect malicious behaviours like serial failed login attempts on multiple machines, attempts to access unauthorized networked resources, or malicious traffic patterns on the whole IT infrastructure. These systems can perform forensic analysis since they record the events collected; note that to characterize hacker campaigns or Advanced Persistent Threats (APT) it is necessary to analyze from several months to years of logs, data that normally would not be stored.

From our interaction with SIEMs and Security Operations Center (SOC) operators, we learned that their effectiveness is directly related to the quality of the rules defined for the correlation engine. We identified three main factors that influence the quality of the rules: the experience of the operators, the completeness of the logs collected, and the freshness of the cybersecurity intelligence present in the SOC. While we cannot influence the first factor, and the second one could provide an interesting topic for future research, we focus on the latter.

As with any cybersecurity system, SIEMs must be constantly updated with the latest cybersecurity content to maximize their threat coverage and detection. Usually, cybersecurity systems are solely updated by the company that provides them, *i.e.*, if one buys an anti-virus from company `A`, it will be solely updated by `A`'s feed. This is not the case with SIEMs, as these systems are extensible and can receive intelligence from almost any source. Companies purchase additional cybersecurity feeds from specialized companies not only to increase the quality of the SIEM's database, but also to complement the detection capabilities of the SOC and increase the company's security level. Examples of such curated feeds include LookingGlass [19] and Anomali ThreatStream [3], which provide specialized feeds tailored to the customer. However, focusing on a few companies' feed is a reductive approach since a wealth of knowledge is published daily by all kinds of cybersecurity information sources, such as cybersecurity analysts, researchers, and hackers.

To broaden the horizons of cybersecurity intelligence, companies turn to Open Source Intelligence (OSINT). According to the U.S. Intelligence Community [125], OSINT is "*(...) publicly available information appearing in print or electronic form. (...) It may be disseminated to (...) the mass media, (...) gray literature, which includes conference proceedings, company shareholder reports, and local telephone directories. (...) Open Source involves no information that is: classified at its origin; is subject to proprietary constraints (other than copyright); is the product of sensitive contacts with U.S. or foreign persons; or is acquired through clandestine or covert means.*". In summary, OSINT is information publicly available on the news and Internet. A wide variety of results have been achieved through OSINT usage, such as detecting earthquakes [112], retrieving threats from the dark web [101], or creating an Android anti-malware solution [111]. We highlight blog posts, news, social media, and articles of scientific or technical nature from the wide variety of OSINT sources, since these are the simplest to collect and process [87]. These sources are provided in an unstructured format, even though unstructured text processing is a current Natural Language Processing (NLP) challenge. Since SIEMs and other cybersecurity systems are not prepared to process unstructured text, to take advantage of OSINT one must employ some framework that collects and parses it to a machine-readable format, such as Logstash [33], IntelMQ [17], or SpiderFoot [32].

Although there are many OSINT sources, Twitter is in the spotlight for two main reasons. First, Twitter is well-recognized as an important source of short notices about occurring events and web activity [14]. This is also true for cybersecurity-related events, as demonstrated by the highly-active accounts of most cybersecurity feeds and researchers, who tweet cybersecurity-related news [96, 111]. Second, since a tweet is limited to 280 characters (mostly 40–60 words), these messages are potentially simple to process automatically, enabling very high levels of accuracy and low false-positive rates by using standard machine learning techniques. The cybersecurity research community has also taken an interest in Twitter due to its information richness on a wide variety of topics (*e.g.*, [96, 106, 111]).

1.1 Motivation

Despite mounting evidence that a Twitter-based security can be used to increase the cybersecurity intelligence of a SOC [55, 93, 98, 111, 114], there were no systematic studies evaluating such approach. Using Twitter for cybersecurity became a common practice on the literature without a clear evaluation of its timeliness, richness, or usefulness. The first

step for this work was to validate Twitter usage for cybersecurity through a systematic study of their characteristics and advantages. Since this study provided positive results (see Chapter 3), we worked on creating a Twitter-based security feed.

Since Twitter is an unstructured data source, it is necessary to employ a specialized tool to collect, parse, select, and transform tweets into a format that can be ingested by other systems (such as SIEMs). We set to build one such tool, and found that there are two requirements for efficient OSINT (including Twitter) usage [116, 118]: adequate data selection, and post-processing in the form of data aggregation and deduplication. There are numerous threat intelligence tools (*e.g.*, SpiderFoot [32], IntelMQ [17]) that can collect cybersecurity-related OSINT (including tweets). However, these focus on collecting data from various sources or on capabilities such as ordering events generated from multiple sources. They use simple keyword-based filters to narrow the large volume of collected information, and do not employ sophisticated methodologies to address the aforementioned requirements.

In the literature, there are many proposals for selecting cybersecurity-related tweets (*e.g.*, [96, 111, 114]). Nevertheless, we found a gap since there is no proposal that addresses the following four aspects:

- These systems should not depend on a secondary data source to validate the collected tweets, since waiting for validation will most likely cancel any timing advantage from using Twitter;
- The data collection should not be restricted in any format to not discard relevant alerts;
- The system must include a method to aggregate similar tweets to not create redundant alerts;
- The system must be designed for processing the tweet stream and include the time dimension in all its components (*e.g.*, the aggregation must be updated over time).

1.2 Thesis Hypothesis

Based on the previous discussion, we defined the following hypothesis for this thesis:

An information retrieval system can provide threat intelligence systems with timely and informative Indicators of Compromise obtained from Twitter.

1.3 Research Questions and Contributions

This thesis is focused on two OSINT-centred aspects: its processing (sources, collection, selection, and summarisation); and its practical use. Based on those points, we defined the following three research questions.

RQ 1. What are the best cybersecurity OSINT sources?

Our first research goal is related to OSINT sources. As discussed, Twitter was already considered a relevant data source. However, there were no quantitative or qualitative studies concerning its use in a cybersecurity context. Therefore, our first research question is focused on quantifying some aspects of using OSINT (and in particular Twitter and National Vulnerability Database (NVD)) as a cybersecurity OSINT source:

1.1 Is NVD the richest and timeliest vulnerability database?

1.2 Does Twitter provide a rich and timely vulnerability coverage?

1.3 How are vulnerabilities discussed on Twitter?

Our findings show that: vulnerability databases complement one another in richness and timeliness (i.e., no single source contains all the vulnerabilities; no single source can be relied on to provide the earliest vulnerability reporting date); Twitter is a rich and timely vulnerability information source; and finally, Twitter complements other OSINT sources.

These results were reported in the following publication:

Fernando Alves, Ambrose Andongabo, Ilir Gashi, Pedro M. Ferreira, and Alysson Bessani. **Follow the blue bird: a study on threat data published on twitter.** In *European Symposium on Research in Computer Security, pp. 217-236. Springer, Cham, 2020.*

RQ 2. How to create a system capable of collecting and selecting only relevant tweets for a given IT context?

There are several research works for the collection of cybersecurity OSINT, including some that collect data solely from Twitter. However, the vast majority of works either restrict the data to a fixed set of topics, or require external validation of the data collected. These

approaches are not practical nor flexible, thus not meeting the requirements of SOC operators, who manage a changing and heterogeneous infrastructure. Therefore, our classifier is designed without the said restrictions. Finally, our work includes an extensive methodology on how to select the best models to build the system.

These results were reported in the following publication:

Fernando Alves, Pedro M. Ferreira, Alysson Bessani. **Design of a Classification Model for a Twitter-based Streaming Threat Monitor.** In *1st International Workshop on Data-Centric Dependability and Security (together with IEEE/IFIP DSN'19).*

RQ 3. How to create a framework capable of summarizing and presenting the selected information for the convenience of SOC operators?

As mentioned above, there are several previous works that collect tweets for cybersecurity purposes. However, these works fall short on their applicability, as they do not consider how the collected data will be presented to the final users. SOC operators (or cybersecurity analysts in general) have a limited time budget to get updated with the latest news. Therefore, unlike previous work, this research goal has two main objectives:

3.1 To collect and select only tweets relevant to the IT context under an analysts' care;

3.2 To group similar items and present only one element representing the group data considering the time dimension (*i.e.*, continuous aggregation), to simplify the data analysis.

To this end, we developed SYNAPSE, a framework that collects OSINT from specific sources, selects the relevant information, and summarizes it for the convenience of SOC operators. More specifically, this framework comprises a data collector to retrieve tweets, a feature extractor to convert them into numerical feature vectors, a supervised machine learning approach to select the relevant data, and a summarising function to aggregate similar items.

SYNAPSE is designed to summarise current cybersecurity events, useful for any cybersecurity analyst. It is able to select relevant events with high confidence through its machine learning classifiers, while the novel stream clustering algorithm makes it able to aggregate events over time—a requirement for any such system in a SOC. Further, as SYNAPSE's

output will be in Indicator of Compromise (IoC) format, it enables the direct connection between our framework and any system capable of receiving IoCs. For example, the analyst can observe the provided IoCs, and choose which ones are fed to a SIEM.

SYNAPSE and its associated results were reported in the following publication:

Fernando Alves, Aurélien Bettini, Pedro M. Ferreira, Alysson Bessani. **Processing tweets for cybersecurity threat awareness.** In *Information Systems (2020)*.

This work also contributed to the following publications:

Nuno Dionísio, Fernando Alves, Pedro M. Ferreira, Alysson Bessani. **Cyberthreat detection from twitter using deep neural networks.** In *2019 International Joint Conference on Neural Networks (IJCNN). IEEE, 2019*.

Nuno Dionísio, Fernando Alves, Pedro M. Ferreira, Alysson Bessani. **Towards end-to-end Cyberthreat Detection from Twitter using Multi-Task Learning.** In *2020 International Joint Conference on Neural Networks (IJCNN). IEEE, 2020*.

1.4 Document Structure

The remainder of this document is structured as follows: Chapter 2 presents previous research work concerning OSINT usage and processing; it also reviews the requirements for efficient OSINT usage in a security environment. Chapter 3 details the results of our study on OSINT sources and its usage; Chapter 4 describes our tweet collector and classifier; it also presents lessons learned as how to use Twitter as a security data source. Chapter 5 describes our Twitter collection and processing tool and its integration with SIEM systems; in there we describe how to build a tool that follows the recommendations and lessons learned in the previous chapters. The conclusions and future research directions appear in Chapter 6.

2
Background and Related Work

This chapter describes research work relevant to this thesis. First, we present a review of OSINT sources and their publication, followed by a review of works that collect cybersecurity tweets. Second, we describe research work on recommending tweets.

2.1 OSINT for Cybersecurity

This section describes OSINT collection and processing. First, the types of data sources used to collect OSINT are described, followed by Twitter-based related work.

2.1.1 Data Sources

The data sources used to gather OSINT can be divided into three major groups:

Structured data sources: Resources that provide structured data in a well-defined format. The data obtained from these sources comes in a machine-parsable format.

Unstructured data sources: Feeds that provide unstructured data where the main content is in free-text format. Feeds in text format (such as news posts) are typically more information-rich, although this data type requires further processing,

Dark web: Only accessible using TOR-compatible software, is a part of the Internet known for hacker sites and forums, and exploit marketplaces. Both are rich information sources for malicious activity. The data found here is expected to be in an unstructured format.

In the following, we describe these sources in detail.

2.1.1.1 Structured Data Sources

Structured data sources present machine-parsable information, and thus can be directly fed to a system. These sources usually provide an API for programmatic access to their content. Two relevant examples are vulnerability repositories and IP/rules sources.

Vulnerability/exploit sources. Sources that collect confirmed vulnerabilities or exploits provide their content in a structured format. Vulnerability databases describe each entry using several numeric fields (such as the CVSS score [7]) and a text description, complemented with a set of external links pointing to content that describes the vulnerability in further detail (such as the software developer's webpage or scientific articles). Structured datasets are the most reliable information sources since their content is officially confirmed before publication. This reliability comes at a price; usually, there is a time-lapse between detecting a vulnerability and its presence in this type of database.

Two of the most important structured sources are Common Vulnerabilities and Exposures (CVE) and NVD. MITRE Corporation [23] maintains the Common Vulnerabilities and Exposures list [6] (in short, CVE), a compilation of known vulnerabilities described in a standard format. A global index of known vulnerabilities simplifies complex analyses such as detecting APT. Therefore, indexing known vulnerabilities in CVE became standard practice for security practitioners, including software vendors. Each CVE entry has an ID (CVE-ID), a short description, and the creation date.

The NVD [24] mirrors and complements CVE entries on their database. Every hour, NVD contacts CVE to obtain newly disclosed vulnerabilities (we contacted NVD directly to get this information). Each vulnerability indexed in NVD undergoes a thorough analysis, including attributing an impact score based on the Common Vulnerability Scoring System (for both versions 2.0 [9] and 3.0 [8]), and links related to the vulnerability, such as advisory sites or technical discussions. NVD uses the CVE-ID in place of an ID of its own.

Besides CVE and NVD, many online databases compile known vulnerabilities and provide unrestricted use of their contents, such as the Security Database [31], PacketStorm [28], Exploit Database [11], or Vulners [30]. The complementary information provided by each database differs, but in general, these provide a description, some analysis of the security issues raised by the vulnerabilities, known exploits, and possible fixes or mitigation actions.

IP and rule sources. In the case of blacklists/whitelists, or sets of rules (*e.g.*, firewall rules), the data is available in text files with an address/rule per line. Each line can be fed directly to the corresponding software.

2.1.1.2 Unstructured Data Sources

Unstructured sources include news feeds, blog posts, forums, and scientific articles. The most used sources are news feeds and blogs, since these are simpler to filter, thus aiding in content selection.

Unstructured sources provide text data describing events of all sorts, including security ones. Blogs and news may contain more information (*e.g.*, a quick fix to a vulnerability), but pose a complex challenge for automated processing since extracting concepts from free text is still an NLP challenge. Therefore, as appealing as they may be, using them as OSINT sources is far from trivial. Nevertheless, some authors show it is possible to collect data from technical blog posts and scientific literature, since technical writing tends to have a stable structure and much less ambiguity when compared to other types of writing [87, 139].

One of the unstructured data sources that is receiving particular attention is Twitter [36], a micro-blog service where users can publish text and media content. Twitter is a popular feed since a quick review of tweet titles provides an overview of current news and trends; news sites, bloggers, and other information sources post tweets containing the post's title to increase the visibility of the content they produce. Tweets tend to provide concise information due to their character limit (140 from 2006 to 2017, 280 since then), making them attractive for publishing quick status updates. Tweets are also attractive for automated processing, as small, concise messages are simpler to process than large texts.

2.1.1.3 Dark Web

Accessible only through TOR-compatible software, the dark web offers anonymity to the users accessing it and to the services hosted on it. Therefore, it is the ideal place for buying, selling and discussing all types of illegal commodities and services. This is also true for botnets, exploits, malware, viruses and all kinds of malicious IT assets.

The dark web is known for active exploit discussion and development. Collecting information about threats during their development phase, or about threats for sale yet to be used is extremely valuable, as it allows defenders to act before the attackers. This approach has

been successfully undertaken by Nunes *et al*. [101], who obtained data on zero-day vulnerabilities on dark web marketplaces and hacker forums.

2.2 Twitter for Cybersecurity

In this section, we review research works focused on collecting tweets for cybersecurity. All works described below use Twitter as their OSINT source and aim to find cybersecurity OSINT about a given IT infrastructure, but apply different techniques to process and provide those tweets to the end-user.

Okutan *et al*. [102] integrate tweets with posts from the GDELT news service and Hackmageddon to detect new threats related to one of three topics: Defacement, Denial of Service, and Malicious Email/URL. Khandpur *et al*. [82] use semantic trees to validate if tweets mention one of three topics: distributed denial of service attacks, data breaches, and account hijacking. Liu *et al*. [89] use semantic trees to group cybersecurity tweets by their semantic. Sabottke *et al*. [111] show that information about exploits are published on Twitter two days before they are included in NVD (on average). These entities are used to detect complex events and categorise them into one of seven topics. Behzadan *et al*. [52] use two convolutional neural networks, one to assess the relevance of a tweet for cybersecurity, the other to assign it to one of seven topics. Ji *et al*. [81] use a multitask neural network where each task is classifying if tweets are related to seven topics. Bose *et al*. [54] extract keyword-based and social-based features to cluster them and detect trending events concerning 11 different topics. Niakanlahiji *et al*. [100] use regex expressions to extract IoCs from tweets and form discussion threads by chaining tweet replies. Mittal *et al*. [96] use a knowledge base created from security concepts to evaluate if a tweet is relevant for cybersecurity. Le Sceller *et al*. [84] designed a framework that collects tweets on a keyword basis and is capable of extending the keyword set automatically, focused on six topics. Ritter *et al*. [106] search Twitter for occurrences of three specific topics: DoS attacks, data breaches, and account hijacking. Yagcioglu *et al*. [133] fuse various machine learning techniques to select tweets relevant for cybersecurity by following a cybersecurity taxonomy. Sapienza *et al*. [114] validate tweets mentioning new threats using dark web sources. The authors published an extension of this work that also uses technical blogs to increase the quality of the approach [115]. Lee *et al*. [85] complement Twitter with blogs to form a timeline of cybersecurity events. Le *et al*. [83] use CVE descriptions (therefore, only positive samples) to train a classifier and infer if a tweet is relevant for cybersecurity. Trabelsi *et al*. [130] cluster tweets by subject. Threats

Table 2.1: Summary of the related work and comparison of the techniques used.

Authors	**Number of topics**	**No external validation**	**Aggregation capabilities**	**Stream processing**
Okutan *et al*. [102]	3	✗	✗	✗
Khandpur *et al*. [82]	3	✓	✗	✗
Liu *et al*. [89]	5	✓	✓	✓
Hyejin Shin *et al*. [121]	5	✓	✓	✗
Sabottke *et al*. [111]	7	✗	✗	✗
Behzadan *et al*. [52]	7	✓	✗	✗
Ji *et al*. [81]	7	✓	✗	✗
Bose *et al*. [54]	11	✓	✓	✗
Niakanlahiji *et al*. [100]	13	✓	✗	✓
Mittal *et al*. [96]	Taxonomy	✗	✗	✗
Le Sceller *et al*. [84]	Taxonomy	✗	✓	✗
Ritter *et al*. [106]	Taxonomy	✓	✗	✗
Yagcioglu *et al*. [133]	Taxonomy	✓	✗	✗
Sapienza *et al*. [114]	Unrestricted	✗	✗	✗
Sapienza *et al*. [115]	Unrestricted	✗	✗	✗
Lee *et al*. [85]	Unrestricted	✗	✗	✗
Le *et al*. [83]	Unrestricted	✓	✗	✗
Trabelsi *et al*. [130]	Unrestricted	✗	✓	✗
Han-Sub Shin *et al*. [120]	Unrestricted	✓	✗	✗

not referred by NVD are considered novel and handled like zero-day vulnerabilities. Han-Sub Shin *et al*. [120] build a classifier fusing two models, one trained to detect cybersecurity relevant data and another trained from generic data. Hyejin Shin *et al*. [121] focus on the words present in tweets to identify new or re-occurring alerts.

Table 2.1 summarises these works according to four features: topic or taxonomy-restricted data collection, tweet validation through external knowledge, data aggregation, and stream processing capabilities. With the exception of Le's *et al*. [83] approach—which shares similarities with our data collection methodology—all approaches are either: dependent on a secondary cybsersecurity data source to validate the tweets [85, 114, 115, 130]; or have restrictions on the tweet collection: follow only a fixed number of cybersecurity top-

ics [52, 82, 84, 96, 102, 106, 129, 133], or to specific events such as exploits [111], or possible zero day vulnerabilities [130]. Both variants are not practical for a real use case scenario. The first solution is likely to lose the Twitter timeliness advantage [57, 93, 98]. The latter is too restrictive, as companies cannot afford to be protected only against some attacks. Furthermore, by setting a number of topics to focus on, one of two things will happen when a tweet discussing an unpredicted topic appears: either the tweet is discarded because it does not fit the predefined model (and possibly important information is lost), or it is placed in a topic it does not belong to. In both cases, the user is likely to lose confidence that the system is performing correctly.

As can be seen in Table 2.1, the research community largely overlooked the post-processing of the collected cybersecurity data, which is not discussed in the majority of papers. Among the previously mentioned works, only five summarise the collected tweets. Liu *et al.* [89] employ semantic trees to group tweets according to various formats, such as continuous occurrences or connecting seemingly disparate events. The authors set the number of clusters to form by observing cybercrime statistics. The presented methodology is useful for detecting APT, but the authors do not clarify how it can be used in the daily operations of security operation centres. Hyejin Shin *et al.* [121] organise the tweets into the proposed five topics used in the paper. A set of keywords describes each topic, and the tweet is valid only if it has one of those words. Bose *et al.* [54] use the DBSCAN density-based clustering algorithm [71] to aggregate tweets by events. However, their results show high heterogeneity of threat types by event, meaning that, in practice, a SOC operator would have to inspect all elements of each event manually. Le Sceller *et al.* [84] use Local Sensitive Hashing [79] to group tweets by similarity to calculate a relevance metric. The system disregards clusters with less than ten elements. We believe this methodology is inappropriate for cybersecurity as the system will discard many non-critical security tweets. Trabelsi *et al.* [130] use k-means to group the tweets by threat. However, the authors do not expose the methodology used to find the critical parameter k, nor show any validation of the clustering methodology. Furthermore, none of these works show a proper evaluation of the methodologies.

We may consider that the proposals that collect tweets based on taxonomies or topics [52, 54, 81, 82, 84, 89, 96, 100, 102, 106, 111, 121, 133] could present the collected data organized according to those topics. However, this organisation is too coarse-grained to be used in practice. The following two tweets provide an example:

- "*#0daytoday #Joomla Easy Youtube Gallery 1.0.2 SQL Injection Vulnerability [webapps #exploits #Vulnerability #0...*";

- "*#0daytoday #Wordpress Ocim MP3 Plugin SQL Injection Vulnerability [webapps #exploits #Vulnerability #0day #Ex...*".

While both discuss the same *topic* (SQL injection), they discuss different issues. To be useful, the data needs to be organised so that each tweet set discusses *exactly the same subject*, and to the best of our knowledge, no proposal in the literature addresses this requirement.

As can be seen in Table 2.1, there is a lack of a system that is able to aggregate cybersecurity a stream of tweets about any topic, validate the data without requiring external sources, and aggregate them. In Chapters 4 and 5, we propose a system that fulfils all described requisites.

3
Twitter Study

A growing trend for obtaining cybersecurity news is to collect OSINT from the Internet [125]. OSINT sources include vulnerability databases (*e.g.*, the NVD), online forums (*e.g.*, Reddit), social networks (*e.g.*, Twitter), and scientific literature. Although more technical, exploit databases (*e.g.*, ExploitDB) are a useful OSINT source providing code excerpts known as Proofs of Concept (PoC) that show how to exploit a vulnerability. PoCs can be analysed by a specialised audience capable of using the exploit's code to understand and counteract vulnerability exploitation, thereby removing the vulnerability.

The research community has shown many different uses for OSINT, from its collection and processing [46, 52, 66, 67, 84, 96, 106, 115, 130], vulnerability life cycle analysis [49, 55, 94, 109, 119], to evaluating vulnerability exploitability [44, 55, 56, 70, 99, 111]. There are two predominant OSINT sources in the literature: NVD (*e.g.*, [44, 55, 99, 111, 117]), and Twitter (*e.g.*, [46, 84, 93, 96, 130]). The first provides curated vulnerability data, while the latter is more generic, concise, and covers more topics.

As Twitter's usage grew, various information sources began to link their content on Twitter to increase visibility and attract attention. Twitter's continued growth placed it among the most relevant communication tools; the vast majority of companies have a Twitter account to interact with the world. All this activity also caught the attention of the research community. The information flow and interaction graphs meant new research opportunities, such as detecting emerging topics [60, 103], or finding events related to a specific topic, such as riots [45], patients experience with cancer treatment drugs [53], or earthquakes [112]. Twitter popularity instigated the development of tools to collect tweets (*e.g.*, Tweet Attacks Pro [35]), APIs for programming languages (*e.g.*, Tweepy [34]), and many OSINT-collecting tools de-

veloped specific plugins to collect tweets (*e.g.*, Elastic Stack [37]), including cybersecurity-oriented ones (*e.g.*, SpiderFoot [32]).

However, to the best of our knowledge, there is no evidence in the literature highlighting one security data source as advantageous over the others. For instance, the following questions are yet to be answered: Why use NVD solely when there are several reputable vulnerability databases? Is NVD the richest (in terms of the number of vulnerabilities reported) and timeliest (does it contain the earliest reporting date of a vulnerability) vulnerability database? Why use Twitter to gather cybersecurity OSINT? Does Twitter provide any advantage over vulnerability databases? Is it useful for security practitioners?

In this chapter, we present an extensive study on OSINT sources, comparing their timeliness and richness. We analysed the vulnerability OSINT sources indexed on vepRisk [47], which aggregates several vulnerability databases, advisory sites, and their relationships. We compared Twitter against these data sources to understand if there are any advantages in using it as a cybersecurity data source. To explore this topic, we formulated three research questions:

RQ1: Is NVD the richest and timeliest vulnerability database?

RQ2: Does Twitter provide a rich and timely vulnerability coverage?

RQ3: How are vulnerabilities discussed on Twitter?

Our findings show that: vulnerability databases complement one another in richness and timeliness (*i.e.*, no single source contains all the vulnerabilities; no single source can be relied on to provide the earliest vulnerability reporting date); Twitter is a rich and timely vulnerability information source; and finally, Twitter complements other OSINT sources. In summary, the contributions of this chapter are:

- A comparison between some of the most reputable and complete vulnerability databases in terms of timeliness and coverage;
- An analysis of the coverage and timeliness of Twitter with respect to vulnerability information;
- An analysis about "early alerts" on Twitter, *i.e.*, vulnerabilities disclosed or discussed on Twitter before their inclusion on vulnerability databases;
- An analysis of how vulnerabilities are discussed on Twitter;

- Insights on how to collect timely tweets;
- Insights regarding Twitter's usage for cybersecurity threat awareness.

3.1 Additional Related Work: Cybersecurity-Related OSINT Studies

To the best of our knowledge, Sauerwein *et al.* performed the most similar study to the one present in this chapter [117]. For two years, the authors collected all tweets with a CVE-ID in their text. They show a comparison of the tweet publishing dates with the disclosure dates of those CVEs on NVD. The results show that 6232 vulnerabilities (25.7% of their dataset) were discussed on Twitter before their inclusion on NVD. However, this study falls short in some aspects. Firstly, the NVD is not always the first database to report new vulnerabilities, which changes the vulnerabilities' first confirmed report date (see Section 3.4). Secondly, the authors search only for CVE-IDs on Twitter, which will not capture issues that have been disclosed to the public but not (yet) indexed on CVE or NVD. Finally, the analysis is focused solely on the vulnerabilities' life cycle and Twitter appearance, overlooking vulnerability characterisation such as their impact.

There is some research work providing evidence that relevant and timely cybersecurity data is available on Twitter [57, 93, 98], *i.e.*, that some vulnerabilities were published on Twitter before their inclusion on vulnerability databases. However, these are case studies concerning a single vulnerability, and compare the tweets referring them solely with NVD. Other Twitter-based contributions include correlating security alerts from tweets with terms found in dark web sources [114], studying the propagation of vulnerabilities on Twitter [127], and finding that exploits are published on Twitter (on average) two days before the corresponding vulnerability is included in NVD [111].

In a similar research line, Rodriguez *et al.* [107] analysed vulnerability publishing delays on NVD when compared to other OSINT sources: Security Focus, ExploitDB, Cisco, Wireshark, and Microsoft advisories. The authors report that NVD presents publishing delays (from 1 to more than 300 days) from 33% to 100% of the cases when comparing with those databases, *i.e.*, sometimes it publishes after these databases, or it always publishes after these databases. However, the authors consider only the year 2017. Similarly, the Recorded Future company reports that for 75% of the vulnerabilities NVD presents a 7-day disclosure delay [29]. However, the company does not reveal how it obtained these results.

3.2 Vulnerability Disclosure Procedure

Ideal procedure. Fig. 3.1 depicts the ideal coordinated vulnerability disclosure with CVE, which proceeds as follows. Alice becomes aware of a vulnerability in her company's software (1); she contacts the CVE to register the new vulnerability and obtain a CVE-ID (2). At this point, a CVE entry is *reserved*, meaning that an ID for the vulnerability is assigned and the creation date is registered. However, no information about the vulnerability is made public—not even who is the requester (in this case, Alice), the affected software, or the nature of the issue. Once Alice develops a patch for the vulnerability (3), it contacts CVE again for an organised public disclosure of the vulnerability, *i.e.*, both Alice's company and CVE disclose the vulnerability to the public at the same time (4). Once the vulnerability becomes public on CVE, NVD can fetch its data and perform its analysis.

There is a significant difference between the dates of NVD and CVE entries. In CVE, it is the date when entries became *reserved*, but not yet *public*. NVD entries are always public, using the date when they were indexed, even prior to their analysis completion. Thus, in practice, a vulnerability has the same public disclosure date on both CVE and NVD, which is the NVD creation date. Therefore, the NVD vulnerability disclosure date is the actual date a vulnerability becomes public.

Other cases. Nevertheless, the vulnerability disclosure process described above is not always followed. Considering the model defined by Frei *et al.* [74], the most relevant cases for security practitioners happen when a vulnerability disclosure time is prior to a security database disclosure time, independently of the other timings.

Other disclosure scenarios include:

- Bob discovers a vulnerability but does not inform Alice, thus making impossible a coordinated disclosure;

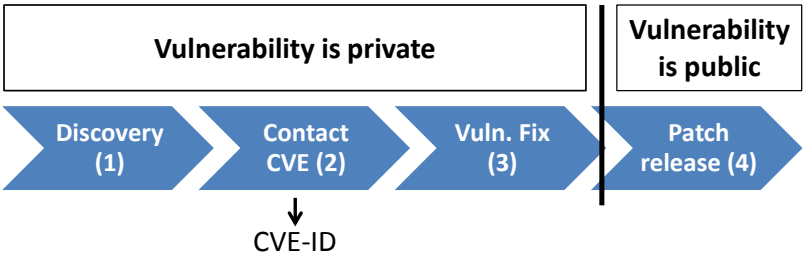


Figure 3.1: Vulnerability disclosure procedure with CVE.

- Alice discloses the vulnerability before the CVE;
- The vulnerability is indexed on CVE much later due to indexing and standardisation efforts.

3.3 Methodology

The objective of this study is to compare some aspects of the information present on vulnerability databases with Twitter. Instead of searching, collecting, and parsing a set of databases, we use the vepRisk database [47]. It contains several types of security-related public data, including all entries published on NVD, Security Database, Security Focus, and PacketStorm databases, from their creation until the end of 2018.

We chose Twitter as an OSINT source, as it is a known aggregator of content posted by all kinds of users (hackers, security analysts, researchers, etc.), news sites, and blogs, among others who tweet about their content to increase visibility [14]. Thus, Twitter became an information hub for almost any kind of content. Unlike vulnerability databases—that contain only security data—Twitter includes discussions over a vast universe of topics. Since the results of this study are based on tweets mentioning indexed vulnerabilities, we decided to search for tweets mentioning the vulnerabilities indexed on NVD. Finally, to ensure the validity of our results, we opted to *manually* match tweets to vulnerabilities. These decisions raised two questions: 1) what part of the vulnerability description are we going to use as a search term? and 2) How to reduce the number of vulnerabilities to manually inspect?

The NVD description of some vulnerabilities includes a "colloquial" name for which the vulnerability became known. For example, CVE-2014-0160 is known as the "Heartbleed bug". These names are within two categories: a generic description of the vulnerability class (*e.g.*, "Microsoft Search Service Information Disclosure"), or some "creative" designation related to the vulnerability (*e.g.*, "Heartbleed" is a vulnerability on the "heartbeat" TLS packets which can be exploited to leak or "bleed" information). These colloquial names are easily recognisable since they always appear in the NVD vulnerability description after the "aka" acronym (for *also known as*). Therefore, to guide the search on Twitter, all vulnerabilities with a colloquial name were selected, and the names were used as query terms. This decision also reduced the number of vulnerabilities to analyse to 9,093, an amount of data manually processable. Additionally, vulnerabilities with colloquial names are more likely to be discussed on Twitter since most were "named" due to media attention. The IDs of the

9,093 vulnerabilities with a colloquial name that were used in this study are listed online [1].

We were unable to use the Twitter API to collect the tweets for the study as it only provides access to tweets published in the previous week. However, the Twitter web page allows searching for tweets published at any point in time. To automate the querying process, a library called `GetOldTweets` [12] was employed. It mimics a web browser performing queries on the Twitter page, enabling fast and programmatic retrieval of any number of tweets from any time.

Regarding matching tweets and vulnerabilities, we consider that a tweet t unequivocally refers a specific vulnerability v if and only if: (1) t mentions in its text v's CVE-ID even if the vulnerability has not yet been disclosed on NVD, or (2) t contains a link mentioned in v's NVD description, even if the web page pointed by the link is currently down, or (3) t mentions a security advisory that is also referred by v's NVD links about that threat, or (4) t or t's links mention an ID associated with v. Two assumptions are made: 1) if an ID is present on a tweet, then the advisory has been published; and 2) a security analyst that receives a tweet containing a security advisory ID can search for this advisory, thus having the same result as publishing the advisory link on the tweet. If a vulnerability is mentioned by up to a thousand tweets, all tweets were manually inspected. The colloquial name of some vulnerabilities is also a word commonly used on tweets, such as "CRIME" (CVE-2012-4930) or "RESTLESS" (CVE-2018-12907). For those cases where a search term can return more than 350,000 tweets, the manual inspection was done in two steps. First, the description is analysed to understand the vulnerability characteristics and related terminology. Then, a large set of informed searches were performed on the tweet set in search of tweets potentially referring the vulnerability. In total, *about a million tweets were manually inspected*, and any links present in potential matches were also examined to confirm the matches. The data labelling was performed solely by us, and it took roughly eight months to complete. All potential matches were triple checked to ensure their validity.

The time range considered in this study begins in March 2006 (Twitter's creation date) until the end of 2018. The tweets were collected between early 2017 and the end of 2019. The resulting dataset contains 3,461,098 tweets. The tweets publishing times were adjusted to the day time scale to match the time granularity provided by the vulnerability databases. Therefore, all time comparisons performed in this study used the publishing day.

3.4 Vulnerability Database Comparison

As NVD is considered a standard for consulting vulnerability data, many research works use only it as their vulnerability database (*e.g.*, [94, 105, 109, 117]). This is a natural choice since NVD includes multiple resources for further understanding of the issue at hand. However, other reputable vulnerability databases, with their own disclosure procedures and timings, provide useful information for security practitioners. Therefore, it is interesting to understand if there is evidence that supports using only NVD for practice or research work. To investigate this point, we collected data about two different aspects: the number of entries and their publishing date. The first measures the coverage of the database, while the second is related to its timeliness and practical usefulness.

Table 3.1 shows the number of entries in each of vepRisk's databases: NVD, PacketStorm (PS), Security Database (SD), and Security Focus (SF). It also shows the number of entries shared between each database pair. Tables 3.2 and 3.3 are related to timeliness. Table 3.2 is divided in two blocks. The first shows the number of occurrences where one database was the *first* to disclose a vulnerability ahead of other databases. The second block shows the number of occurrences where various groups of databases were *simultaneously first* to disclose a vulnerability. Table 3.3 complements the previous table by showing the percentage of times each database was one of the first to disclose a vulnerability, *i.e.*, if a database disclosed an unknown vulnerability at the same time as other databases.

There are five key takeaways obtained from analysing the tables:

1. NVD is not the most complete vulnerability database, with the Security Database and PacketStorm containing more entries;

2. NVD is not the most timely database. Alone, it was never the first to publish a vulnerability;

3. No database stands out as the most timely;

4. Security Database contains all of NVD's entries (this was manually verified);

5. With the exception of NVD, all databases publish different vulnerabilities.

Therefore it is important to follow a set of data sources instead of relying solely on one.

Table 3.1: The number of entries in each database (in bold) and the number of shared entries between database pairs.

	NVD	**PS**	**SD**	**SF**
NVD	**110,353**	-	-	-
PS	9,290	**129,130**	-	-
SD	110,353	9,344	**117,098**	-
SF	60,378	8,597	60,843	**98,445**

Table 3.2: The number of times one database or a group of databases were the first to disclose a vulnerability.

Database(s)	**# Occurrences**	**%**
NVD	0	0.00
PS	853	0.77
SD	0	0.00
SF	40,208	36.44
NVD, SD	51,238	46.43
PS, SF	1,265	1.15
NVD, SD, SF	16,580	15.02
NVD, PS, SD	85	0.08
NVD, PS, SD, SF	124	0.11

Table 3.3: The percentage of times each database was one of the first to disclose a vulnerability.

Database	**# Occurrences**	**%**
NVD	68,027	61.64
Security Database	68,027	61.64
Security Focus	58,177	52.72
PacketStorm	2,327	2.11

3.5 Twitter Vulnerability Coverage and Timeliness

3.5.1 Coverage

A first validation on using Twitter for cybersecurity is verifying if vulnerability data reaches Twitter. We searched for tweets mentioning each of the CVE-IDs published on NVD after Twitter's creation. Of the 94,398 CVE-IDs searched, 71,850 (76.11%) were mentioned in tweets. However, by analysing Figure 3.2 it is possible to observe that since the beginning of 2010, CVEs have become regularly discussed on Twitter. In fact, from 2010 forward, the coverage became above 97.5%, validating the hypothesis that vulnerability data reaches Twitter.

The drastic increase in tweets mentioning CVEs in 2010 may be connected to the sudden growth Twitter underwent in that period [15]. Nevertheless, the turning point in cybersecurity threat awareness and the importance of coordinated vulnerability disclosure mechanisms *may* have been at the beginning of 2010, when Google publicly disclosed that their infrastructure in China was targeted by an advanced persistent threat codenamed "Operation Aurora" [27]. Later on, it was discovered that other major companies were targeted, such as Adobe Systems, Rackspace, Yahoo, and Symantec. This event *could* have triggered two crucial social phenomena: that companies are attacked and should not be ashamed of it, and should disclose the details of these attacks in a coordinated effort to detect, understand, and prevent them; and that the users prefer transparency in cybersecurity events since when data breaches occur, typically it is the user data that is affected.

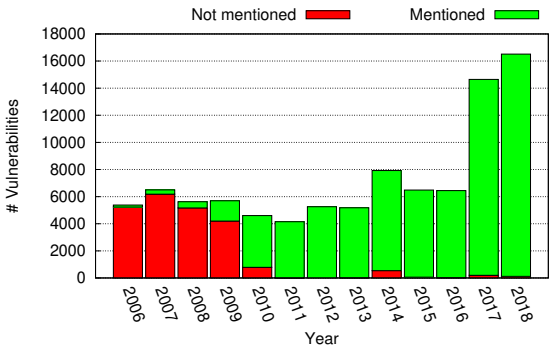


Figure 3.2: Twitter's CVE coverage.

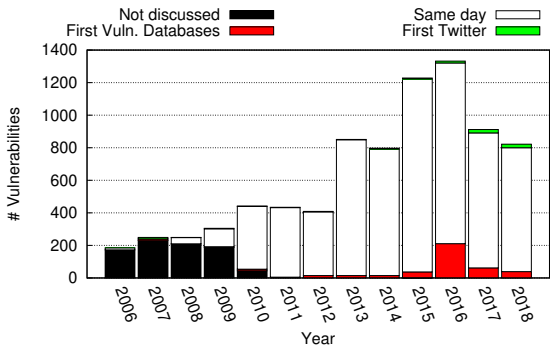


Figure 3.3: A timeliness comparison between vulnerability databases and Twitter.

3.5.2 Timeliness

Regarding timeliness, we performed the analysis only for the 9,093 vulnerabilities that were manually analysed to ensure the correctness of the results. Figure 3.3 shows, for those vulnerabilities, which source discussed them first: either one of the vulnerability databases considered in this study, Twitter, or Twitter and at least one of the databases, simultaneously. There are also the cases where the vulnerability was not discussed on Twitter, which were predominant before 2010. Although we are not evaluating the whole databases, the figure shows a predominance of same-day publishing cases (84.56% when considering 2006–2018, and 93.73% in 2010–2018). We consider that these results validate the hypothesis that Twitter is a timely source of vulnerability data. In the next section, we present an in-depth study of the cases where the vulnerabilities were discussed on Twitter ahead of vulnerability databases.

3.6 Early Vulnerability Alerts on Twitter

Of the 9,093 vulnerabilities analysed, 89 were referred by tweets before being published on at least one of the vulnerability databases considered. Even though these vulnerabilities represent a small percentage of the vulnerability sample under study (0.98%), we decided to characterise them to understand if searching for early alerts on Twitter is a worthy endeavour. The most mentioned vendors in the early alerts are the Ethereum blockchain (17 mentions), Microsoft products (5), Debian (5), Oracle (4), Linux (4), and Apple (4), while the most mentioned assets are Javascript (9), SSL/TLS (8), Xen Hypervisor (4), Safari Browser (3), Mercurial version control (3), and Cloud Foundry (3). These mentions provide evidence of the usefulness of these alerts, as both vendors and assets are some of the major players in their respective fields.

All vulnerabilities with Twitter early alerts can be found online [1], together with their publishing dates on the vulnerability databases and some extra notes. In the following sections, these early alerts are further analysed on their impact and usefulness. We conclude the section with a discussion of the significance of these results.

3.6.1 Timeliness

3.6.1.1 Twitter Versus Vulnerability Databases

Figure 3.4 presents the distribution of early alerts over the years considered. The number of early alerts increased in the last two years, which matches the increase of vulnerabilities published on databases since the beginning of 2016.

Concerning publishing timing, the majority of early alerts were available up to thirty days ahead of vulnerability databases (78.65%–70 cases). Notably, four early alerts appeared between 31 and 50 days ahead, four between 51 and 100 days ahead, nine between 101 and 200 days ahead, and finally, the two cases with the highest antecedence were 371 and 528 days ahead. The number of days Twitter is ahead of vulnerability databases increased continuously since 2008, but only after 2016 we found more than 10 cases. No relevant patterns were discovered in the early alerts due to the small number of occurrences.

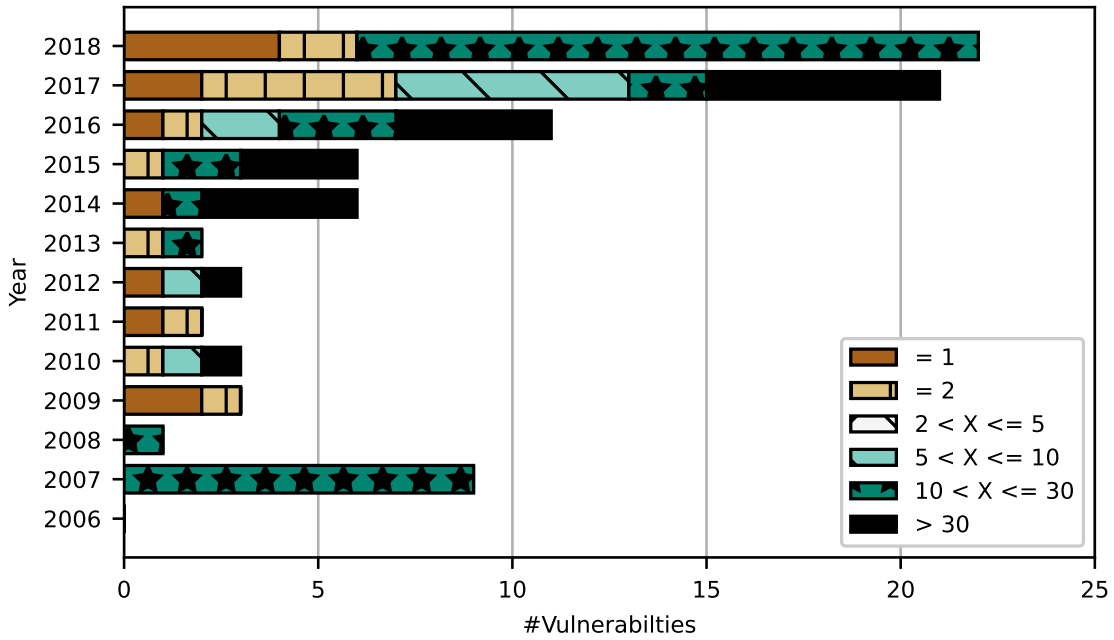


Figure 3.4: The number of early alerts found per year.

3.6.1.2 Twitter Versus Advisory Sites

Besides vulnerability databases and social media, advisory sites are an essential source of vulnerability information. Many companies use websites to announce software patches, along with which vulnerabilities are fixed. Therefore, we compare Twitter with advisory sites as these are specialised OSINT sources directly connected to the software vendors.

We manually searched for advisory notices for each of the early alerts, obtaining only 33 advisories (approximately of early alerts). Table 3.4 presents the number of times either Twitter or the advisory was the first publisher, or when both published on the same day.

The majority of early alerts are not paired with an advisory, but the tweets referring them contain links that describe these vulnerabilities. This observation reinforces the idea that Twitter is a useful cybersecurity discussion hub by connecting various knowledge resources in a single place.

3.6.2 Vulnerability Impact

Although the existence of early alerts is relevant by itself, it is essential to assess the impact of the vulnerabilities. Table 3.5 presents how many early alerts have low, medium, high, and critical (CVSS 3.0 only) CVSS scores according to the CVSS 2.0 and 3.0 scoring systems. As the CVSS 3.0 was released in 2015, 31 early alerts are ranked only according to CVSS 2.0 (the N/A line in Table 3.5b).

Almost all early alerts are ranked by CVSS 2.0 as having a medium or high impact (about 94%). When considering the CVSS 3.0, no alerts are ranked with low impact, and five are graded with a critical score.

Despite the small number of early alerts, the CVSS score points out that these are relevant vulnerabilities and should not be disregarded. For example, CVE-2017-14581, rated with a

Table 3.4: The number of times Twitter, advisory site, or simultaneously both, were the first to publish an advisory notice.

1st publisher	#	%
Twitter	11	12.36
Both on same date	13	14.61
Advisory site	9	10.11
No advisory publication	56	62.92

Table 3.5: The CVSS 2.0/3.0 impact of the early alert vulnerabilities.

(a) CVSS 2.0.

CVSS 2.0	#	%
Low	5	5.62
Medium	64	71.91
High	20	22.47

(b) CVSS 3.0.

CVSS 3.0	#	%
Low	0	0.00
Medium	16	17.98
High	37	41.57
Critical	5	5.62
N/A	31	34.83

7.5 CVSS 3.0 score, describes a flaw in SAP systems which makes them vulnerable to denial of service attacks. Another example is CVE-2016-7089, which describes that WatchGuard firewalls are vulnerable to privilege escalation via code injection. This vulnerability belongs to the set of issues disclosed by the "Shadow Brokers" [13], and has a public exploit on ExploitDB [39].

3.6.3 Vulnerabilities Exploited at Disclosure Time

A vulnerability only has an actual impact once it is exploited. Table 3.6 shows the exploitation status of the early alert vulnerabilities, both at the Twitter publishing and disclosure dates. The majority of vulnerabilities are not paired with observations of their exploits in the wild (64% or 57). A quarter of these cases (23.6% or 21) are known to be exploited. In a few cases (12% or 11), a PoC was referred by the vulnerability notice, describing how to exploit the vulnerability. As it is impossible to know if that PoC was used, we categorised these separately from the cases where the exploitation was confirmed.

We matched the early alerts with CVE-mentioning exploits present in ExploitDB [11] to complement the previous result. Only one case was found, published before the earliest vulnerability database and after the disclosing tweet. This information was used to update Table 3.6, adding the "At disclosure" column.

Considering vulnerabilities known to have been exploited and those with a PoC, the total amounts to about 34%. Current studies estimate that the percentage of vulnerabilities that are exploited in the wild is 5.5% [70], meaning that these early alerts include many appealing targets for hackers.

Table 3.6: The exploitation status of the early alerts.

Exploitation status	Twitter publishing		At disclosure	
	#	%	#	%
Exploited	21	23.60	22	24.72
PoC	11	12.36	11	12.36
No data	57	64.04	56	62.92

Table 3.7: The actionability provided by the early alert tweets.

Action types	#	%
Patch	34	38.20
Configuration/patch	5	5.62
Configuration	12	13.48
None	23	25.84
No data	13	14.61
Unreadable	1	1.12
N/A	1	1.12

3.6.4 Actionability

Perhaps even more important than knowing the impact or exploitation status of a vulnerability is to avoid exploitation. This can be achieved by applying patches or configurations to protect the vulnerable system. Table 3.7 shows which vulnerability mitigation measures can be reached by following the hyperlinks found in early alert tweets. In almost 40% of the cases, the tweet includes a link pointing to a patch that solves the vulnerability. For another 40% of vulnerabilities there is no patch available ("None"), or that information is not clear or the topic is not discussed ("No data"). The unreadable case is due to a page not written in English, where some parts of the text were not clear even after translation. The N/A entries are due to dead links, which blocked the analysis.

In the majority of cases (57%), the early alerts provided some information on how to protect the vulnerable systems from exploitation, either by patch or configuration. If the cases where we could not get more information (the "No data" cases) provided some solution, then the protection rate would increase to more than 70%. Therefore, we conclude that besides impact and exploitation relevance, early alerts are also useful due to the actionability they enable, as they inform security practitioners of possible actions to protect their systems.

3.6.5 Twitter Discussion and Vulnerability Impact

The tweet discussion volume is one of the metrics used by techniques such as topic detection (*e.g.*, [60, 72, 103]), which identifies new discussion trends. Therefore, we wanted to understand if the Twitter vulnerabilities discussion volume could be correlated to their severity or impact, *i.e.*, if the number of tweets discussing a vulnerability can be used as an importance metric. As described in Section 3.3, we performed a similar search and manual analysis only for the early alerts, starting on the day of each vulnerability Twitter disclosure, and without an end date. Out of the 695,413 collected tweets, about 88,000 tweets were manually inspected.

To search for correlations between the discussion about a vulnerability and its importance, we used Spearman correlation [69] (ϱ) between various pairs of indicators, which provides a value between 1 (positive correlation) and -1 (negative correlation). The 0 value indicates no correlation. Since we are correlating Twitter discussion volume with vulnerability importance, we used various indicators to define both variables. For the Twitter discussion volume we used the total number of tweets discussing vulnerabilities and the peak number of tweets posted on a single day for each vulnerability. For both definitions, besides using all the tweets, we considered one correlation variant using only the vulnerabilities whose tweet count and daily peak were above ten, embodying the idea that only those vulnerabilities actually caused significant discussion. Then, to define vulnerability importance, we selected the CVSS 2.0 and 3.0 ratings (for the vulnerabilities that have it), and the CVSS-based global vulnerability impact. The CVSS-based impact measures the amount of damage that can be caused by exploiting the vulnerability. Before computing the correlations, we further divided the vulnerabilities into three groups: all vulnerabilities, those with descriptive names, and those with colloquial names (see Section 3.3).

The correlations are presented in Table 3.8 (CVSS 2.0), Table 3.9 (CVSS 3.0), and Table 3.10 (CVSS-based impact). Between brackets is shown the number of samples for each correlation. The number of samples used in each correlation is shown between parenthesis. We use the terminology by Dancey and Reidy [63] to interpret the correlation strength.

In the majority of cases the correlation is weak or very weak ($0 \leq |\varrho| < 0.4$). The first exception occurs when considering the number of tweets of vulnerabilities with a creative name correlated with the CVSS 3.0, where a moderate ($0.4 \leq |\varrho| < 0.6$) correlation was found. The negative result suggests that, to a certain extent, the impact of the vulnerabilities decreases as the volume of discussion increases. In the same case, for the peak number of

Table 3.8: The correlations between the discussion aspects and the CVSS 2.0 score.

All				**Creative Naming**			
#Tweets		#Peak		#Tweets		#Peak	
All (227)	$> 10(122)$	All (227)	$> 10(97)$	All (46)	$> 10(38)$	All (46)	$> 10(38)$
-0.07	0.01	-0.07	-0.03	-0.23	-0.17	-0.16	-0.03

Descriptive Naming			
#Tweets		#Peak	
All (181)	$> 10(84)$	All (181)	$> 10(59)$
-0.03	0.05	-0.03	-0.03

Table 3.9: The correlations between the discussion aspects and the CVSS 3.0 score.

All				**Creative Naming**			
#Tweets		#Peak		#Tweets		#Peak	
All (191)	$> 10(97)$	All (191)	$> 10(72)$	All (37)	$> 10(31)$	All (37)	$> 10(31)$
-0.06	-0.06	-0.06	-0.30	-0.55	-0.51	-0.43	-0.33

Descriptive Naming			
#Tweets		#Peak	
All (154)	$> 10(66)$	All (154)	$> 10(41)$
-0.02	0.10	-0.03	-0.23

Table 3.10: The correlations between the discussion aspects and the vulnerability impact.

All				**Creative Naming**			
#Tweets		#Peak		#Tweets		#Peak	
All (227)	$> 10(122)$	All (227)	$> 10(97)$	All (46)	$> 10(38)$	All (46)	$> 10(38)$
-0.14	-0.10	-0.13	0.00	0.02	0.07	0.10	0.48

Descriptive Naming			
#Tweets		#Peak	
All (181)	$> 10(84)$	All (181)	$> 10(59)$
-0.05	-0.09	-0.05	-0.24

tweets, by analysing the two results we consider the correlation weak. The second exception happens when correlating the peak number (> 10) of tweets with a creative name with the

CVSS-based impact, where a moderate positive correlation was found. In this case, the positive result suggests that the impact increases as the volume of tweets increases. Overall, there is moderate evidence that the impact and severity of vulnerabilities with a creative name are, respectively, positively and negatively related to the volume of tweets.

By examining the most discussed early alerts (> 100 collected tweets) for common grounds, we discovered that all vulnerabilities are related to some of the most widely used IT vendors (*e.g.*, Microsoft, Cisco, Intel). These products tend to attract mediatic attention since any flaws will likely impact a great audience. Although we cannot provide a clear correlation to prove this finding as we cannot find exact usage numbers for each product contemplated in our analysis, it is natural that any vulnerabilities whose exploitation cause worldwide damages will get more attention (*e.g.*, "wannacry").

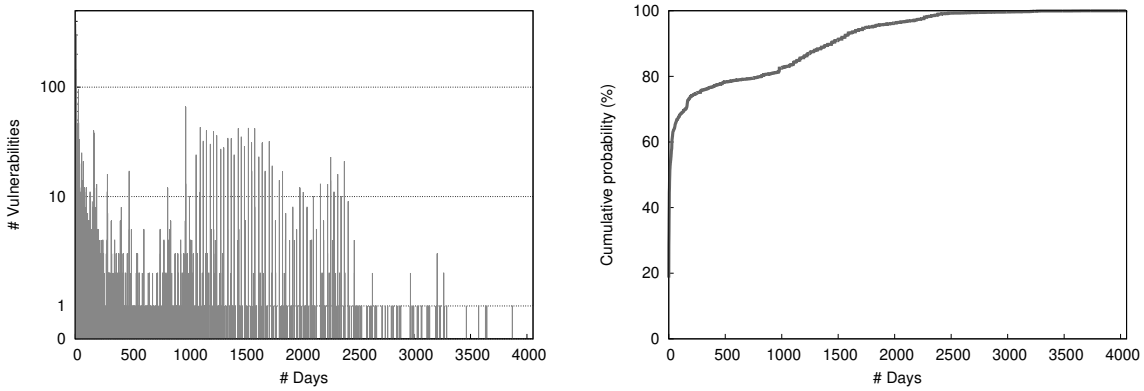
3.7 How Vulnerabilities are Discussed on Twitter

In this section, we characterise some aspects of how vulnerabilities are discussed on Twitter. By identifying these aspects we provide guidelines for topic detection techniques oriented at capturing cybersecurity events. The following results are based on the analysis of the 9,093 vulnerabilities considered in this study.

3.7.1 Duration and Number of Tweets

Figure 3.5 presents the discussion duration and associated cumulative probability. We observed that half of the vulnerabilities were discussed for up to eight days. However, it is interesting to see that the other half is middling, spread across up to 2,000 days. In some cases, the discussion can continue for up to almost 3,800 days. Discussion periods are extensive on some vulnerabilities mainly due to three different reasons: being used as comparative examples when discussing new events (*e.g.*, CVE-2014-0160, the "Heartbleed" bug); being (partly) reused on new attacks or as a part of a campaign, (*e.g.*, CVE-2017-11882 [26]); as case studies, therefore being remembered by their impact, specificity, or technical details.

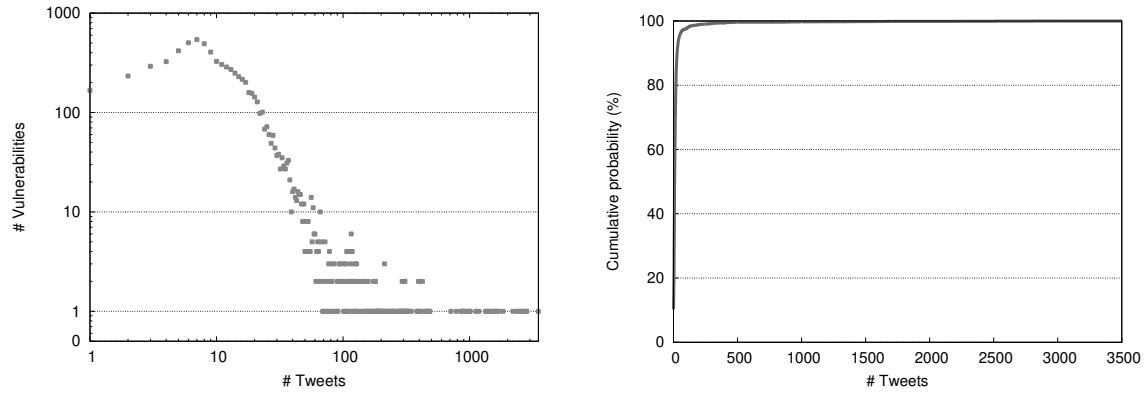
Figure 3.6 presents how many tweets discuss the vulnerabilities and the associated cumulative probability. From the graph we are omitting two outliers: one vulnerability discussed by 7,749 tweets, and another discussed by 15,733. Half of these vulnerabilities are discussed by two to thirteen tweets. These results are not surprising considering that most vulnerabilities are uneventful. The large majority of vulnerabilities are described, patched, and forgot-



(a) Discussion duration in days.

(b) Cumulative probability of the discussion duration.

Figure 3.5: The vulnerability discussion duration on Twitter.



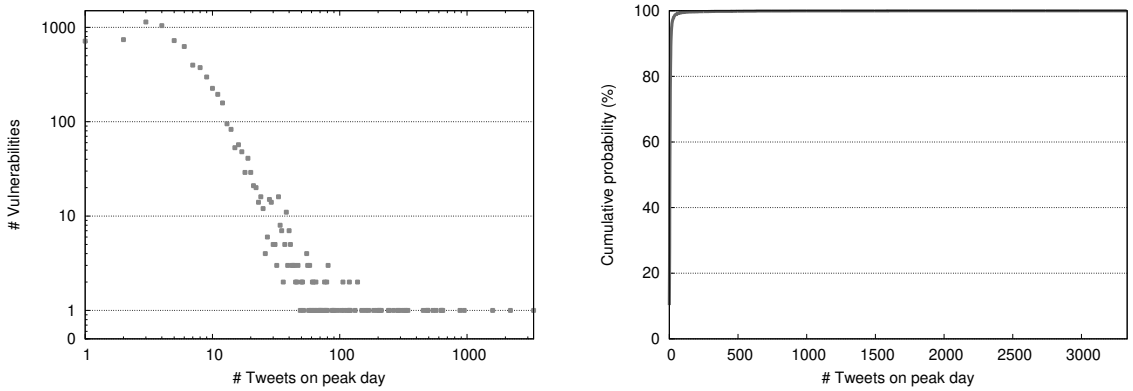
(a) Number of tweets discussing a vulnerability.

(b) Cumulative probability of the number of tweets discussing a vulnerability.

Figure 3.6: The number of total tweets discussing a vulnerability.

ten. Also, only a small percentage of vulnerabilities is exploited in the wild [70], which are the ones more likely to attract more attention. Only 351 vulnerabilities were discussed by more than 50 tweets, showing that although this content is posted on social media, relatively few vulnerabilities attract attention. However, taking a closer look at those 351 vulnerabilities, 14 of them have low severity ratings according to CVSS 2.0, 124 have medium severity, and 213 have high severity. Although it is not implied that vulnerabilities with medium and high severity are going to be widely discussed, these results indicate that those referred by more tweets tend to have higher severity ratings.

Figure 3.7 presents the daily peak discussion, *i.e.*, the maximum number of tweets dis-



(a) Peak number of tweets in a single day.

(b) Cumulative probability of the peak number of tweets in a single day.

Figure 3.7: The peak number of tweets in a single day.

cussing the vulnerability in a single day. It also presents the cumulative probability associated with the peak discussion. Three-quarters of the vulnerabilities have daily peaks between one and ten tweets. This is an important factor for topic detection techniques, as most of these identify new trends based on detecting bursts of tweets discussing the same event [132, 134].

3.7.2 Accounts

The tweets discussing security content used in this study were posted by 194,016 different accounts. We performed a quick analysis to understand if any account(s) stand out as sources of cybersecurity tweets. Out of the 194,016, only 5,863 of them published more than one relevant tweet, and only 228 posted more than 5. The highest tweet count for a single account is 73 tweets. Therefore, in this study, we did not find any best accounts to follow for cybersecurity content.

3.8 Summary of Findings

In the following, we summarise the findings associated with each of the research questions formulated in this study.

RQ1: Is NVD the richest and timeliest vulnerability database? The NVD does not stand out as the most complete vulnerability database, as there are others that index more vulnerabilities. Also, NVD is not the most timely database. In fact, it was never the first

database to publish a new vulnerability ahead of the others. However, NVD is known to have a strict publishing policy, allowing for consultation and comments from product vendors, which means that vulnerability publication may be delayed.

RQ2: Does Twitter provide a rich and timely vulnerability coverage? Since the beginning of 2010 Twitter provides a timely and rich coverage of known vulnerabilities. Moreover, there is a small subset of vulnerabilities (less than 1% of those we inspected) that are discussed on Twitter before their inclusion on vulnerability databases. Although these are very few cases, our analysis shows that they are relevant, impactful, and in many cases provide useful security recommendations. Overall, we consider Twitter as a useful cybersecurity news feed that should be taken into account by security practitioners.

RQ3: How are vulnerabilities discussed on Twitter? Vulnerability discussion on Twitter is carried out mostly in small bursts of two to thirteen tweets. Most vulnerabilities stopped being discussed within eight days, although tweets about them can appear for several years. Vulnerabilities discussed by a higher-than-usual volume of tweets (more than 50) tend to have higher impacts.

3.9 Insights for Practical Usage

Beyond the comparative analysis presented in this chapter, there are a set of insights that we gathered while analysing the tweets collected. Below we present practical takeaways related to OSINT usage and its advantages.

No vulnerability database stands out as the best. NVD is an essential OSINT source, especially due to its thorough analysis and important link aggregation, but other reputable databases should be considered as a complement for four main reasons: *Timeliness*—NVD is not the most timely database (see Section 3.4); *Actionability*—NVD does not directly provide suggestions to mitigate or avoid vulnerabilities, unlike other databases (*e.g.*, PacketStorm, Security Focus); *Known exploits*—NVD does not collect information about known exploits, unlike other databases (*e.g.*, PacketStorm, Security Focus); *Completeness*—NVD is not the most complete database (see Table 3.1). Therefore security practitioners should use a database ensemble to collect security events.

Twitter is relevant. OSINT is provided by many reputable sources and should be taken seriously. Besides the significant research efforts (*e.g.*, [46, 84, 93, 96, 130]), there are companies and tools dedicated to OSINT sharing and enrichment. Sections 3.5 and 3.6 demonstrate that tweets provide timely, relevant, and useful cybersecurity news.

Twitter is a natural data aggregator. Another clear advantage of using Twitter to gather information is its natural data aggregation capability. The 89 early alerts mention 73 different products from 59 different vendors. When considering the 9,093 vulnerabilities analysed, these numbers extend to 1,153 products from 346 vendors. Forty-two CVEs are not indexed in the Common Platform Enumeration—a database of standard machine-readable names of IT products and platforms [5] used by CVE.

Security advisories may not be provided by smaller companies. The majority of vendors mentioned in the 89 early alerts do not provide an advisory site or news blog, while the vendors who provide advisories may not provide an API or a feed subscription. Since advisory sites link their content to Twitter at publication time, one can receive security updates by following the advisory accounts or by accessing Twitter's stream API and applying appropriate filters.

Twitter is important but not omniscient. We believe that a plausible trend for OSINT is to use Twitter as a front-end of the latest events. Since tweets have a relatively small size, messages tend to be concise, efficiently summarising the content of the associated links. This is one of Twitter's characteristics that made it so popular: reading a set of tweets is much faster than inspecting a collection of websites. Therefore, Twitter naturally provides an almost standardised summary, quick and straightforward to process, which is very attractive for Security Operation Centres.

Tweets will not replace the current security publishing mechanisms in place. Once a security-related tweet is received, a visit to the associated site is practically mandatory to understand the issue at hand or to search for patches, among other relevant data. It is also arguable that a similar feed can be obtained by using an RSS feed. However, through Twitter, it is possible to monitor multiple accounts and to gather additional information not provided by RSS, such as timely breakthroughs or further discussion concerning the issue.

Collecting OSINT is a continuous process. Another takeaway for security practitioners is that it is essential to follow news about all layers of the software stack by including keywords related to network protocols (*e.g.*, SSL, HTTP) or purchased web services (*e.g.*, cloud services, issue tracking services). This may seem obvious to the reader given this chapter's discussion. However, as part of the work described in Chapter 5, when we asked security analysts of three industrial partners (nation-wide and global companies with dedicated Security Operations Centres) for keywords to describe their infrastructure (to guide our tweet collection), they did not include network protocols or hardware elements.

Moreover, beyond receiving updates about selected assets, it is vital to obtain trending

security news. It is hard to describe all relevant elements of a large company thoroughly, and maybe not all software in use is indexed or known. By extending the collection elements with (for example) topic detection techniques (*e.g.*, [60, 72, 103, 132]), one is more likely to cover all software in use. As Twitter can provide all these types of news and the research community has studied thoroughly topic detection on this platform, having trend detection might be mandatory for effective OSINT collection.

Diverse sources complement each other. Finally, and to complement the previous insight, it is important to follow a diverse set of accounts to observe the broad universe of software vendors. The early alerts were posted by 53 different accounts (for 89 alerts), demonstrating that diversity of sources is crucial for awareness. Moreover, during this study we collected tweets posted by about 194,000 accounts, reinforcing the idea that Twitter is a cybsersecurity discussion hub. It is also important to discuss critical cases like the exploitation of CVE-2017-0144, which became known as "wannacry". The vulnerability was published on CVE/NVD and Microsoft's security advisory, and patched a few months before the wannacry crisis. Therefore, by following Microsoft or CVE/NVD one would be aware of the issue and could avoid the ransomware.

Once the vulnerability started being exploited, several online discussions suggested a set of configurations that blocked the exploit. Therefore, those that did not patch their systems (and for the Windows versions that were not patched by Microsoft) could benefit from OSINT once the attacks began. The wannacry ransomware generated a massive discussion on Twitter: describing the issue, how to avoid it, and informing about the kill switch that eventually disabled it.

3.10 Conclusions, Discussion and Limitations

Conclusions and Discussion

In this chapter, we provide an analysis of the richness of coverage of vulnerabilities and timeliness (in terms of reporting dates of vulnerabilities) of some of the most important OSINT sources, namely Twitter and several vulnerability databases. Our key findings are the following: no source could be considered clearly better than others, and therefore diverse OSINT sources should be used as they complement each other; when considering only confirmed vulnerabilities, NVD should not be the unique vulnerability database subscribed; since 2010, Twitter provides an almost perfect vulnerability coverage; Twitter discusses vul-

nerabilities ahead of databases for very few cases (about 1% for the vulnerabilities examined), and is as timely as the vulnerability databases for the remaining cases; and finally, most of the vulnerabilities reported early on Twitter have a high or critical impact, with the tweet leading to usable mitigation measures. Beyond the collected facts, we provide a set of insights for the security practitioner interested in using OSINT for cybersecurity. These insights are based on our experience of manual inspection of almost one million tweets, and analysing many thousands of vulnerabilities. We believe this knowledge should be valuable for security analysis and research both in industry and academia.

All the vulnerabilities and early alerts reported in this chapter were manually evaluated, matched with an existing CVE-ID, and deemed correct. However, when collecting tweets live, it may not be possible to detect adversarial tweets trying to lead security teams to change their systems mistakenly or to take actions against non-existing threats. A set of research efforts is dedicated to detecting malicious tweets (*e.g.*, [43, 91, 124]). These techniques should be used by security teams to escape from poisoning attacks.

Limitations

Due to the amount of manual effort involved, we chose to perform an in-depth analysis only for the early alerts. Those are the most useful cases for security practitioners, and it is essential to comprehend their characteristics better. Although it would be interesting to extend the analysis to more vulnerabilities, the effort involved makes it infeasible.

The results presented in this chapter are somewhat pessimistic in terms of the number of vulnerabilities that were found to have early alerts on Twitter. There could be more cases with media attention or early alerts that were not captured by our methodology since we cover a reduced amount of vulnerabilities. There is also the possibility of human error, as manual processing of tweets can lead to mistakes and missing some matches—all early alerts were triple checked to avoid false positives.

Another factor that we cannot control (since we are performing a forensic analysis) is that some early alert tweets could have been deleted before this study, and thus not captured. Many dead links also invalidated possible matches, especially when the tweet links used some shortening system, such as “dlvr.it”, “hrbt.us”, “url4.eu”, “bit.ly”, or “ow.ly”.

4

Building a Tweet Classifier

There are two requirements for efficient OSINT usage (including Twitter) [116, 118]: adequate data selection, and post-processing in the form of data aggregation and deduplication. In this chapter we tackle the former, while in Chapter 5 we tackle the latter. There are numerous threat intelligence tools (*e.g.*, [17, 32]) and proposals in the literature for the collection and selection of tweets (*e.g.*, [96, 111, 114]). However, we found that these proposals do not adequately address the challenges since the proposed techniques restrict the data collection to a set of topics [52, 82, 84, 96, 102, 106, 111, 129, 130, 133], or require validation from a secondary source [85, 114, 115, 130]. These limitations reduce the diversity of information gathered since, for example, the user will not receive news about vulnerabilities outside the ones specified by the topics.

We propose a new collection and classification pipeline that addresses the aforementioned flaws. We collect tweets with no topic restrictions from a set of cybersecurity-related accounts. This way, the collected data is more likely related to all kinds of cybersecurity subjects, including for example patches—a type of security data overlooked in the literature. The tweets are filtered using a set of keywords defined by the SOC analyst that represent the managed IT infrastructure. The collected tweets have their text normalized so that features can be extracted. Finally, a supervised classifier selects tweets relevant for the cyberdefense of said infrastructure. A quantitative evaluation considering over 195.000 tweets from 80 accounts over more than 8 months, shows that our proposed classifier finds the majority of security-related tweets concerning an example IT infrastructure (true positive rate above 90%), and incorrectly selects a small number of tweets as relevant (false positive rate under 10%).

4.1 Classifier Setup

4.1.1 Data Collection

The data collector module requires a set of accounts, from which it will collect every posted tweet using Twitter's stream API—an approach already found in the literature [83, 85, 115]. These accounts can be from security analysts and organisations, vendors, hackers, researchers, among others. They are chosen considering the likelihood of users tweeting about the security of elements belonging to the monitored IT infrastructure. Since usually security analysts already follow OSINT sources and Twitter accounts, it is just a matter of providing these sources to our system.

Simply collecting tweets by keywords is a method likely to retrieve large amounts of irrelevant information. For instance, tweets with the word "windows" include all Windows-related topics (the OS) and all tweets referring glass windows. By collecting tweets only from selected security-related accounts, a more substantial fraction of tweets is related to cybersecurity.

4.1.2 Filtering

Despite the account-based collection approach, most likely the collected data will include tweets unrelated to the infrastructure under the analyst's care. These have to be dropped by a filter. The filtering approach assumes that a tweet referring a threat to a particular IT infrastructure asset has to mention that asset. Therefore, a second input is required: a set of keywords describing the assets of the monitored IT infrastructure. Only tweets that include at least one of the keywords will pass the filter. Keywords further restrict the scope of the security events, hence decreasing the number of irrelevant tweets beyond the filter.

To maximise the effectiveness of our system, the keywords defining the monitored assets must be as complete and specific as possible. For example, if the analyst is in charge of securing a Linux cluster running virtual machines to serve a web service with a database, the keyword set could be {`linux, ssh, virtualbox, vbox, mysql, apache, php`}.

4.1.3 Pre-processing and Feature Extraction

Pre-processing normalises the tweet representation. First, all characters are converted to lower case, and stopwords and hyperlinks are removed—the latter are shortened URLs that

provide little information. Numbers, dots, and hyphens are replaced by their textual representation (*e.g.*, "2" to "two"), as these are relevant to distinguish software versions (*e.g.*, Mozilla Firefox 4.5.1-2). Finally, all non `[a-z]` characters are removed. For instance, after pre-processing, the tweet "*#Oracle #Linux 6 / 7 : Unbreakable Enterprise kernel (ELSA-2016-3573) https://t.co/vLTel8NodG*" becomes "*oracle linux six seven unbreakable enterprise kernel elsa hyphen two thousand and sixteen hyphen three thousand five hundred and seventy three*". The original tweets are stored for presentation.

The tweets must be converted to a numerical format to become suitable for supervised learning classification techniques. This work uses the well-known Term Frequency – Inverse Document Frequency (TF-IDF) method [86]. TF-IDF computes weights to words (features) based on their occurrence frequency in each document and on the group of documents considered. The weight of a word increases with its frequency of occurrence in a single document but is scaled down by the frequency of occurrence in all documents. By mapping each consecutive word token to a corresponding vector position, tweets are converted to a constant size, zero-padded, TF-IDF numeric vector. Finally, to limit the size of the vector we employ the hashing trick technique [131].

4.1.4 Classification

For the classification of tweets according to their security relevance, two classifiers have been explored: Support Vector Machines (SVM) [62] and Multi-Layer Perceptron (MLP) Neural Networks (NN) [108, 110]. The SVM is a broadly-used classifier achieving good results across a multitude of application domains. We consider the SVM implementation available in the Apache Spark Machine Learning library (MLlib) [4], which employs a linear kernel, thereby assuming the input vectors are linearly separable.

Since MLlib does not provide a non-linear SVM kernel, MLlib's MLP NN implementation was considered to account for the assumption that input vectors may not be linearly separable. The MLP is a well-established and frequently used NN architecture that has a long track record of good and consistent results over a vast number of classification tasks.

4.2 Experimental Setup

This section describes the experimental work carried out for validation. All code is written in Scala and deployed on the Apache Spark Framework [4].[1] We chose Spark as its data-structures are scalable and designed for large datasets. Also, Spark includes a scalable machine learning library called MLlib, used to implement all ML algorithms employed in this chapter.

4.2.1 Infrastructure Definition

We used a hypothetical IT infrastructure to define the filter during the experimental evaluation. This infrastructure (presented in Table 4.1) is composed of software elements typically found in the IT world, such as the most common browsers and operating systems.

4.2.2 Tweet Collection and Labelling

We collected three datasets during three periods of time. Table 4.2 presents their collection periods, the sets of accounts used, and the number of tweets. After being collected and filtered using the keywords in Table 4.1, each tweet was manually labelled as positive or negative, thus creating labelled datasets suitable for supervised learning. The tweets were labelled as positive when mentioning information relevant for the cybersecurity (*e.g.*, updates, vulnerabilties, exploits) of a part of the IT infrastructure.

Two sets of accounts, `S1` and `S2`, were used for tweet collection, as shown in the third row of Table 4.2. The accounts are listed in Table 4.3.

4.2.3 Feature Extraction

We used Spark's implementation of TF-IDF with default parameters, except for the feature vector size. In order to find a suitable vector size to describe the tweets, eleven values

[1] The source code is available at https://github.com/fernandoblalves/ScalaTweets.

Table 4.1: The hypothetical infrastructure designed for tweet collection and filtering.

oracle, cisco, internet explorer, google chrome, chrome, firefox, microsoft edge, edge, wordpress, joomla, wp, microsoft windows, ms, linux, operating system, operating systems

Table 4.2: Datasets collection and labeling details.

Dataset:	**D1**		**D2**		**D3**	
Time period (from/to)	01/11/2015 01/04/2016		01/04/2016 15/05/2016		15/05/2016 10/07/2016	
Account sets	S1		S1, S2			
Total tweets collected	71024		57579		66608	
Class distribution	Pos.	Neg.	Pos.	Neg.	Pos.	Neg.
	1697	2008	536	4292	1680	2153

Table 4.3: Sets of accounts used to create the datasets.

S1 Accounts: inj3ct0r, TrustedSec, Anomali, briankrebs, Secunia, exploitdb, alienvault, slashdot, dstrom, Info_Sec_Buzz, vuln_lab, threatintel, dangoodin001, ivspiridonov, ThreatFeed, pikisec, SANSInstitute, johullrich, drericcole, F1r3h4nd, MaldicoreAlerts, USCERT_gov, gcluley, hal_pomeran, SecurityWeek, SecurityNewsbot, sans_isc, e_kaspersky
S2 Accounts: TenableSecurity, securitywatch, securityaffairs, zer0element, notsosecure, CyberExaminer, SCMagazine, DMBisson, lennyzeltser, IT_securitynews, teamcymru, WordPress, MicrosoftEdge, JoomlaTips, sjzaib, SecurityMagnate, Cisco, Dell, linuxtoday, securityninja, cyberopsy, OWASP_Java, _WPScan_, d_plusk, threatpost, Rootsector, Microsoft, linuxfoundation, ChidoDike, Sec_Cyber, ptracesecurity, msftsecurity, LinuxSec, hack3rsca, CiscoSecurity, NytroRST, joomla, Windows, crackerhacker00, fstenv, HPE_Security, googlechrome, wordpressdotcom, packet_storm, RokaSecurity, Oracle, firefox, wpbeginner, YoKoAcc, SecurityCrap, jasonlam_sec, threatmeter

were tested: $\{30, 50, 80, 100, 200, 300, 500, 750, 1000, 1500, 3000\}$. This range covers from low to high dimensional vectors, and with it, we should be able to find an appropriate vector size for the datasets.

4.2.4 Classifier Configuration

Relevant hyper-parameters and design variables were varied to find a good design for this application. For the SVM, we varied C (the regularization parameter) within $\{0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5\}$, and the step size (a parameter for the Stochastic Gradient Descent) within $\{0.1, 0.5, 1, 1.5, 2, 5\}$. For the MLP, the number of layers varied from 2 to 8 and the

number of neurons per layer within $\{5, 7, 10, 12, 14, 16, 18, 20\}$.

Each model was evaluated through a 10-fold cross-validation procedure using dataset `D1`. The maximum number of training iterations was set to 100 for the SVM and 200 for the MLP, which were deemed to achieve parameter convergence for the range of the design parameters.

To select the best classifiers, we performed a Pareto-optimal search. For each type of classifier we plotted a Pareto front figure (Figure 4.1), with lines connecting the dominant configurations regarding *True Positive Rate* (TPR, x-axis) and *True Negative Rate* (TNR, y-axis). Each point shows the average value obtained by a specific configuration over the 10-fold cross-validation procedure. The highlighted triangular and circular points are, respectively, the dominant configurations and the configurations chosen to be used (the SVM case) in the experiments. We use the classical true positive definition: a sample labelled as positive and classified as positive; in our case, a tweet manually labelled as relevant and classified as relevant. The negative samples use the equivalent definition.

Based on this analysis, we selected the parameter configurations with the best TPR$\times$TNR balance: those with the smallest distance to the optimum. The best SVM configuration uses a step size and C values of 0.05 and 5, respectively, and the best MLP had 5 layers with 10 neurons each. Both models use feature vectors with a size of 3000, revealing a clear advantage in using high-dimensional feature vectors.

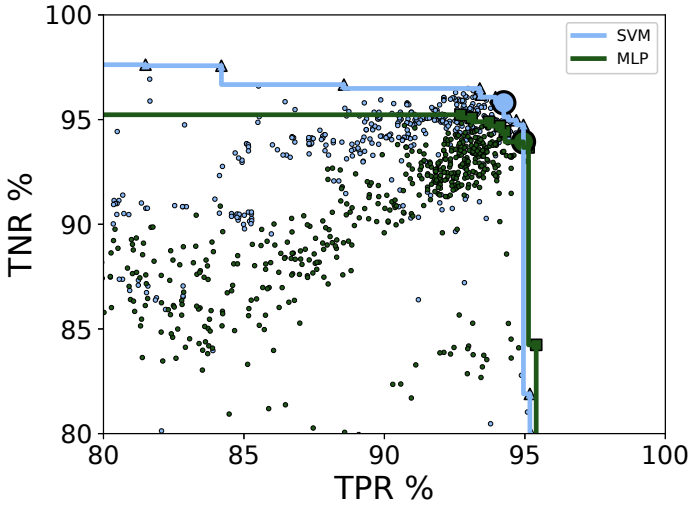


Figure 4.1: The Pareto fronts for SVM and MLP cross-validated using *D1*.

4.3 Results

The tweet processing pipeline components were evaluated using the selected models and datasets D2 and D3. These consider only tweets in the future of those in the training set (D1), and include information posted by an additional and substantially larger set of accounts (S2) not considered in the training stage. This methodology embodies the idea that in a real deployment, models will classify future tweets possibly from a different set of accounts.

Considering that 10-fold cross-validation was employed during the model selection phase, it should be noted that the selected model configurations were trained for the evaluation phase using the whole D1 dataset. The feature vectors of D2 and D3 tweets were generated using the TF-IDF model determined using dataset D1. This guarantees that TF-IDF weights attributed to words in D2 and D3 will be coherent with those used to train the classifiers.

Figure 4.2 shows the True Positive Rate (TPR) and True Negative Rate (TNR) of the SVM and MLP classifiers described in Section 5.4, considering also the average result of the 10-fold cross-validation over D1.

Overall, the results are slightly worse for D2 and D3 when compared to D1 (as expected), since new data presents unmodeled patterns to the classifiers. Focusing on the results obtained for D2 and D3, in general, the classifiers maintain very high TPR and TNR, except for the MLP TPR. In both cases, the TNR is higher than the TPR. The imbalance between positively and negatively labelled data in the training data sets (more negative samples) can explain a higher TNR.

In summary, *the SVM approach achieved the best results, displaying true positive and true negative rates around 90% and showing a small degradation of results in D2 and D3.*

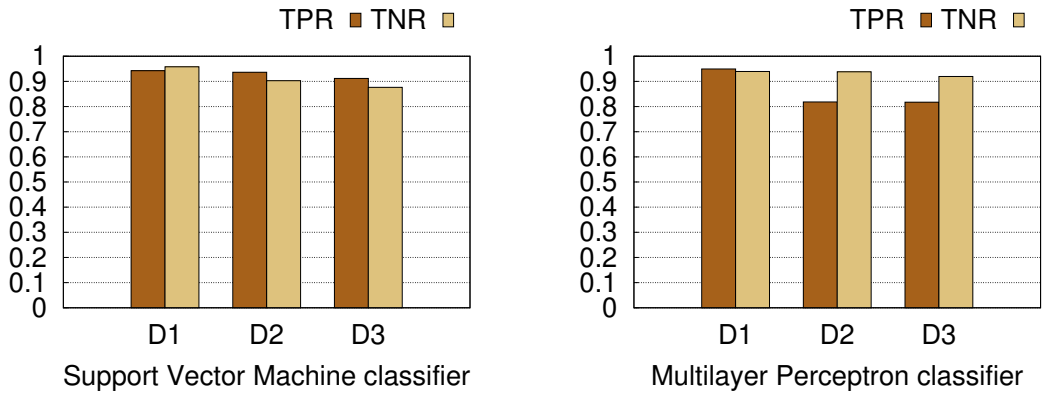


Figure 4.2: SVM (left) and MLP (right) classifier results.

4.4 How to Find Timely Tweets

The results presented in this chapter are based on data that was collected and manually curated several months. However, security practitioners are interested in capturing these posts live. Therefore, this section provides insights for data collection methods based on the experience collected in analysing the datasets of this thesis.

Systems that collect threat intelligence are designed to detect relevant news items while discarding non-relevant ones. The various systems proposed in the literature vary in terms of the complexity of the data selection approach, and as it was infeasible to test all of them, we selected three approaches to test against our data. The first is a simple heuristic-based approach. A tweet is considered relevant if it mentions a software element and a threat word from the VERIS [38] or ENISA [10] cybersecurity taxonomies. The second one is equal to the first but also detects the word "CVE". The third is a more sophisticated approach. We used a convolutional neural network-based approach (CNN) described in Section 5.7. The test simply measures if the approach correctly detects the target tweets.

Table 4.4 shows the results of these approaches. We distinguish pre- and post-2010 periods as the coverage differs significantly. The percentages on the first four columns in the table are obtained from the labelled data used in Sections 3.6 and 3.7, respectively. The last two columns are obtained by running the techniques mentioned above on the dataset described in Section 3.5. These are speculative results as the data is not labelled, but should transpose from the relatively large labelled data.

Using the simplest heuristic method is the worst method of the three except in one case. This means that the trivial approach works but lacks expressiveness regarding how vulnerabilities are discussed. Adding the word "CVE" to the detection mechanism enables it to

Table 4.4: Percentage of correctly detected tweets according to the various datasets and methods. The header row includes the dataset size between brackets.

	Early (89)	**Early post 2010** (76)	**Colloquial** (9,101)	**Colloquial post 2010** (7,923)	**All CVEs** (94.398)	**All CVEs post 2010** (77,409)
Heuristic	56.18%	63.16%	57.01%	63.98%	71.14%	70.99%
Heuristic + CVE	62.92%	71.05%	88.46%	99.24%	84.56%	93.73%
CNN	57.30%	45.68%	89.51%	87.76%	87.74%	87.59%

detect tweets about already indexed vulnerabilities that do not follow the "software name and threat type" tweet text formula, drastically increasing the detection rates. Therefore, this is a suitable detection technique.

Finally, the CNN presents rather poor results regarding detecting early alerts but otherwise is consistently close to 90% accuracy. Although sometimes the CNN has a lower accuracy rate than the heuristics, this test does not cover false-positive rates, where the CNN is expected to largely outperform heuristics.

We performed a follow-up analysis of the early alert tweets in an effort to understand the CNN results. We used BERT [64] to obtain semantic-rich feature vectors from the tweets, and cosine similarity [135] as a similarity measure. The tweets were grouped by similarity, where each tweet was grouped with its most similar peers, as long as each group had an average similarity rating above 0.8. In almost all cases, the CNN gave the same classification for all members of each group. By observing these groups we can assert that the CNN accurately classified as relevant tweets with a more cybersecurity-oriented speech ("*New "Lucky Thirteen" attack on TLS CBC...*" or "*Misfortune Cookie: The Hole in Your Internet Gateway...*"), while incorrectly classifying less structured tweets ("*Only in the IT world can you say things like "header smuggling" (...) or in regex "Did you escape the caret?*" or "*The text for TBE-01-002 references TBE-01-004 - which does not seem to be included in the report. Is that intentional?*"). As our CNN was trained mostly using tweets directly discussing cybersecurity, any tweets not conforming to the pattern are likely to be discarded. Thereby we conclude that a diverse training set for neural networks is required for a complete detection coverage.

4.5 Conclusions

This chapter proposed a tweet collector and classifier structure that adhere to the requirements for efficient usage in a SOC. It implements a pipeline that gathers tweets from a set of accounts, filters them based on the monitored infrastructure, and classify the remaining tweets as either relevant or not. Results show that our system maximises the relevant information (true positive rate of 90%), minimises irrelevant information (false positive rate of 10%). Finally, we provide a set of insights for the security practitioner interested in using OSINT for cybersecurity. These insight are based on our experience of manual inspection of almost one million tweets, and analysing many thousands of vulnerabilities.

5
SYNAPSE

In the previous chapter we designed and validated a tweet classifier; in this chapter, we tackle the second challenge for efficient OSINT usage: post-processing in the form of data aggregation and deduplication. To this end, we extend the work from the previous chapter and create an end-to-end framework called SYNAPSE, a Twitter-based streaming threat monitor that generates a continuously updated summary of the threat landscape concerning a monitored infrastructure.

The tweets classified as relevant are aggregated to avoid the presentation of repeated information by employing a novel stream clustering method adapted to the context of cybersecurity. To comply with the challenge of post-processing a tweet *stream*, our k-means application strategy aims at organizing the tweets such that each cluster discusses the same subject, *i.e.*, the same news-item regarding the same product. Through this method, security analysts can observe only one element from each cluster. Finally, to enable SYNAPSE's integration with SIEMs (*e.g.*, IBM QRadar [16]) and threat intelligence/sharing tools (*e.g.*, MISP [20]), SYNAPSE creates IoCs from the obtained clusters by featuring some specific keywords from the tweets.

We also perform a quantitative evaluation of the clustering strategy and show that it accurately aggregates tweets by specific issues. When compared to a naive text-filtering approach (as employed by most threat intelligence systems used in practice), it decreases the number of tweets presented by approximately 80%, with the number of summarised IoCs being only 21% of the tweets classified as relevant. This volume of data can either be inspected manually or processed by a SIEM as OSINT-generated events. Further, a qualitative analysis of the largest 65 clusters generated by SYNAPSE revealed two paramount findings.

Firstly, 43% of the IoCs describe high-impact security alerts, and for half of these, the tweet publication preceded the vulnerability publication on the NVD by eight days (on average). Secondly, 70% of the analysed clusters provided serviceable intelligence, including exploits whose vulnerabilities were not matched to NVD entries.

Finally, SYNAPSE was integrated with the Security Operation Center of a nation-wide electric utility. Together with SOC operators, we were able to design solutions that integrate tweet-based IoCs in the SOC's daily operation. The resulting rules enriched internal events with external data, and increased the SOC awareness to critical cybersecurity events.

In summary, our contributions are:

1. An end-to-end streaming threat monitor architecture for collecting, classifying, and clustering tweets related to a specified infrastructure (Section 5.2);

2. A novel application strategy and adaptation of well-known clustering techniques to the context of cybersecurity threat awareness (Section 5.3);

3. A detailed system evaluation using three real-world datasets and a qualitative analysis of the security alerts generated thereof (Section 5.5);

4. Methods for generating MISP-compatible IoCs from tweets that enable the integration of SYNAPSE into SOC operation (Sections 5.2.2 and 5.6);

5. Highlights of the integration and real-world validation in a SOC of the techniques proposed (Section 5.6.4).

5.1 Additional Related Work: Stream Clustering

In the following, we briefly review stream clustering algorithms, which is related work specific to this chapter. A system considering a production environment has to process a stream of tweets. This means that batch aggregation techniques (of which the standard is clustering [41]) are not adequate as these are not designed to describe data evolution over time. Thus, a suitable stream clustering algorithm becomes necessary [123]. Stream clustering algorithms receive data points over time; each new data point `d` is either added to an existing cluster, or considered an outlier and discarded. However, existing algorithms (*e.g.*, [77, 136, 138]) have two shortcomings for our context: they require *a priori* definition

of the target number of clusters (k), and they discard outliers [123].. When processing a cybersecurity news feed, the number of active threats under discussion is unknown in advance, and outliers cannot be discarded as they are likely to represent new threats.

In the following are, to the best of our knowledge, the only algorithms that either have variable k or do not discard outliers. Feng *et al*. [73] cluster only the tweets' hashtags, using text similarity to adapt the number of clusters to the collected data. However, this algorithm would potentially miss important information in the security field, as the clustering would not consider the full tweet text, only hashtags. Saki *et al*. [113] use a density-based clustering approach, therefore avoiding the definition of k. However, their technique discards outliers, which could lead to missing important emerging threats. Shou *et al*. [122] approach allows the value of k to vary up to an upper limit, but its outlier detection mechanism discards topics that do not gain traction, ignoring possibly important threats that remain unknown for long periods of time.

5.2 SYNAPSE Pipeline

Figure 5.1 presents SYNAPSE's architecture and data processing stages: tweet gathering, filtering, feature extraction, classification, clustering, and IoC generation. The first four elements were described in Chapter 4, while the two latter ones are described next.

5.2.1 Clustering

SYNAPSE uses clustering to aggregate similar tweets in the news feed stream. The Clustream algorithm [42] was chosen as the basis for this pipeline stage as its structure and characteristics were closest to our requirements. However, it required adaptation to SYNAPSE's context to achieve threat aggregation as described in the next section.

5.2.2 MISP-Compatible IoC Generation

After the clustering phase, the clusters of tweets are transformed into the IoC format to allow their inclusion in SIEMs or threat intelligence platforms. There are several standards for sharing IoCs, such as STIX [50] or MISP [21]. The format must be extensible and adaptable as tweets are unstructured and contain unpredictable content. For these reasons, we selected both MISP and CEF [92] formats to generate IoCs.

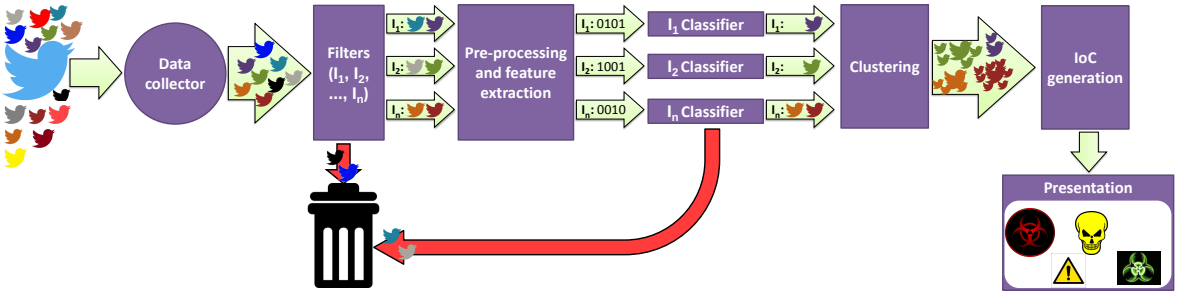


Figure 5.1: SYNAPSE's architecture. Collected tweets pass through various stages and those classified as relevant are aggregated, transformed in IoCs, and delivered to analysts.

We use a combination of MISP items to generate the IoC. One MISP *Event* is composed of two *Objects* containing security indicators called *Attributes*: one describing the content of the exemplar tweet (Section 5.3); the other representing the cluster of tweets. Events are classified using tags, added according to a set of threat categories related to ENISA and VERIS cyberthreat taxonomies [22]. The OSINT tag is added to emphasise the automatic creation based on tweets. The classification is achieved by using regular expressions to match taxonomy elements in the exemplar's message, generating one tag for each match. Further, SYNAPSE includes in its IoCs security bulletin IDs (*e.g.*, CVE-IDs, Ubuntu security notice IDs) in a special field to streamline the correlation of OSINT with other events.

Figure 5.2 depicts the taxonomy employed to represent IoCs in MISP (top of the figure). The exemplar tweet is the core of the IoC, while its cluster is an extra element to increase informativeness. The bottom of the figure shows a MISP Event generated from a cluster and its exemplar (the example cluster shown in Table 5.1). The OSINT object contains extracted information from the exemplar such as the tweet's message, any links therein, and the *Cluster_Analysis* object contains the remainder cluster data. A simple classification was applied: the OSINT tag marks the event as created from tweets, and the "Denial of Service" tag (from VERIS) classifies the threat.

5.3 Tweet Stream Clustering

Since Twitter users can tweet or retweet about the same subject, SYNAPSE is expected to collect many similar tweets. Thus, to cover information about the IT infrastructure, the analyst would have to manually inspect a large amount of redundant data for each threat.

To alleviate this burden, clustering is used to group similar tweets classified as relevant

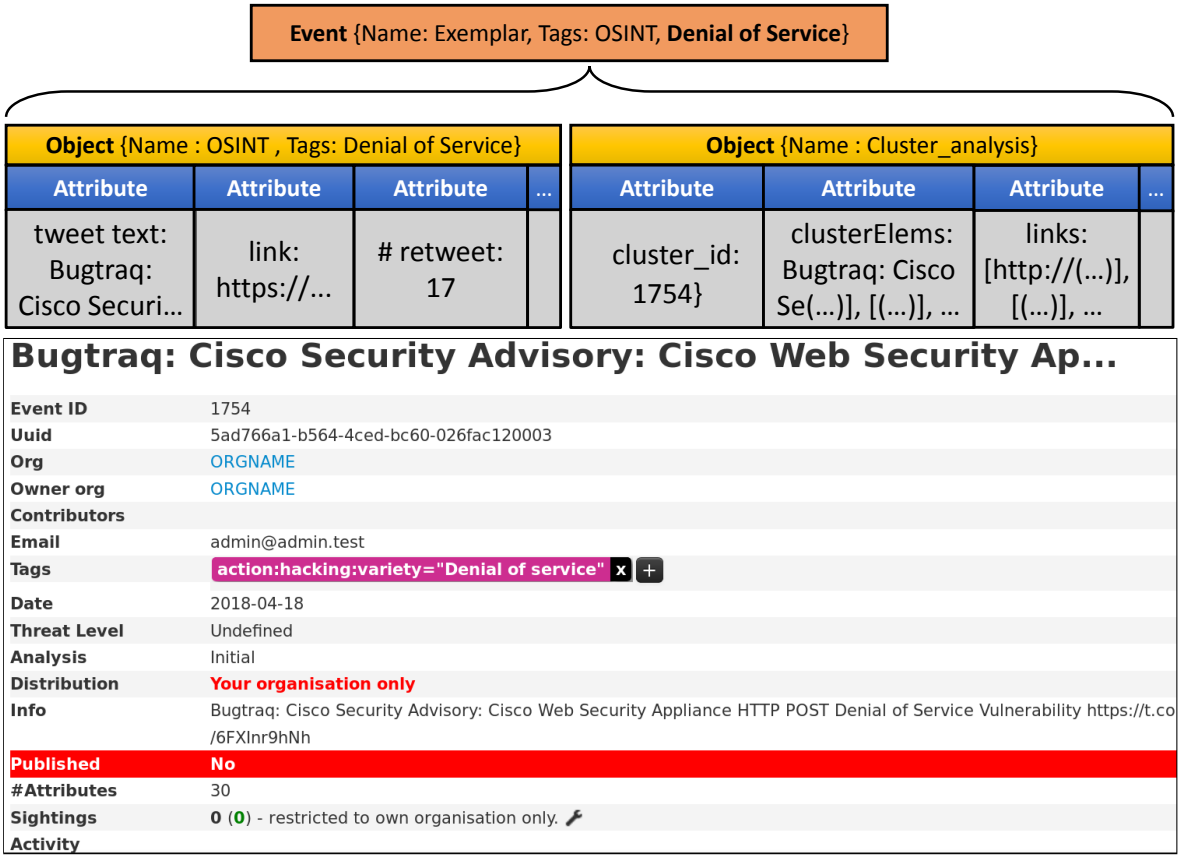


Figure 5.2: Representation of a cluster into the MISP taxonomy [21] and an OSINT-generated event in MISP.

for the protection of the IT infrastructure. Ideally, the information collected about a specific threat gets aggregated in one cluster, from which a single representative tweet—the *exemplar*—is presented to the analyst. By clustering the stream of relevant tweets, distinct active threats are summarised in a set of clusters and updated as more tweets are collected. It is through this mechanism that SYNAPSE can create an active threat monitor outlining the current threat landscape, *i.e.*, the current threats that potentially require more immediate attention from SOC analysts.

5.3.1 Data Stream Aggregation Challenges

Clustering is commonly applied in batch, as an exploratory data technique where a static data set is clustered into k groups [41]. The number of clusters, k, is either defined *a priori*

Table 5.1: An example of a cluster and its *exemplar* (in Bold).

Bugtraq: Cisco Security Advisory: Cisco Web Security Appliance HTTP POST Denial of Service Vulnerability https://t.co/6FXInr9hNh
Bugtraq: Cisco Security Advisory: Cisco Web Security Appliance HTTP POST Denial of Service Vulnerability https://t.co/6FXInr9hNh
Bugtraq: Cisco Security Advisory: Cisco Web Security Appliance HTTP Length Denial of Service Vulnerability https://t.co/TgU0T9vlZt #bugtraq
Bugtraq: Cisco Security Advisory: Cisco Web Security Appliance HTTP POST Denial of Service Vulnerability https://t.co/feZlTxQKVC #bugtraq
#cybersecurity Bugtraq: Cisco Security Advisory: Cisco Web Security Appliance HTTP POST Denial of Service https://t.co/XUUctUnQ8F #infosec
#vulnerability #security : Bugtraq: Cisco Security Advisory: Cisco Web Security Appliance HTTP POST Denial of Serv https://t.co/9bW0ls00kx
#internet #security: Cisco Web Security Appliance HTTP POST Denial of Service Vulnerability https://t.co/cXQUTWUBbD

or estimated to satisfy performance metrics [41]. In a dynamic setting such as SYNAPSE's streaming context, defining k beforehand is not possible, as the number of threats being discussed at a given time is unknown. If at any moment SYNAPSE was processing t threats and clustering was set to find $k \neq t$ clusters, the result would contain clusters including unrelated threats, various clusters related to the same threat, or both cases. *Therefore, SYNAPSE requires a clustering algorithm able to adapt k over time.*

Furthermore, an essential feature of most stream clustering algorithms is the ability to detect and remove outliers that may disrupt the quality of the clustering. In the security context, performing outlier removal could prevent the discovery of emerging threats. Moreover, all tweets reaching SYNAPSE's clustering stage were classified as relevant, and should not be discarded. *Therefore, SYNAPSE requires a clustering algorithm capable of maintaining performance indicators (e.g., intra and inter-cluster cohesion) without removing outliers.*

5.3.2 DynamicClustream

The lack of solutions that fit the requirements of threat intelligence tools (see Chapter 2), motivated us to adapt the Clustream [42] algorithm for SYNAPSE, thus creating the DynamicClustream. The Clustream algorithm clusters a data stream in two phases. The online

phase performs a simple and efficient clustering of the inbound stream by keeping only a summary of the data collected, thus abiding to the speed requirements of a data stream [41]. The offline phase is performed in background to provide a more complete analysis of the collected data through a more effective and computationally demanding clustering algorithm. Clustream includes an outlier detection mechanism that excludes data points unfit for any of the existing clusters by analysing the distance from that point to all clusters. A decision is only taken once it becomes clear if a data point is an element of a new trend or an isolated occurrence. The components that distinguish DynamicClustream from Clustream are detailed in the following.

5.3.3 High-level Overview

Assume there is always a global cluster state S, defined as a set of sets, describing the clusters formed from a previously processed time-window of tweets. When a new tweet t is received, the online clustering component attempts to place t in one of S's clusters. If a direct placement is not possible, a new cluster containing only *t* is created and the offline clustering component is triggered to compute a new clean cluster state considering the tweets in the clusters of S plus t.

Once a new cluster state S is in place, a final step is taken to obtain each cluster's *exemplar* tweet, *i.e.*, the tweet representing the cluster, that will be shown to the analyst. The exemplar tweet is selected by choosing the tweet with the smallest Euclidean distance to the centroid of the cluster. An example of a generated cluster (and its exemplar) appears in Table 5.1. The online and offline components of DynamicClustream are presented in Algorithm 1, with locking details for ensuring atomic updates on S omitted for better readability.

5.3.4 Online Clustering Component

The online clustering component uses a lightweight approach to assign a new tweet t to the current clustering state S. To do so, the membership of t is tested in all clusters (line 3) by employing the WTS cohesion measure (introduced below). This is done by adding t to each cluster $C_i \in S$ and calculating the corresponding WTS value. t belongs to C_i when WTS is above a certain threshold τ. If t does not fit in one of the existing clusters, a new cluster solely containing t is created (lines 4–5). If t belongs to a single cluster, it is added to that cluster (lines 6–7). When t fits more than one cluster, it is added (temporarily) to the cluster with the highest membership rate, and the offline clustering is scheduled (lines 9–10).

Algorithm 1: DynamicClustream online and offline clustering.

```
1  S ← ∅                                                  // global cluster state
2  Function OnlineClustering(t):
3      i ← GetNumHits(S, t)
4      if i = 0 then
5          AddNewCluster(S, t)
6      else if i = 1 then
7          UpdateCluster(S, t)
8      else                                               // needs offline clustering
9          PlaceInClosestCluster(S, t)
10         schedule OfflineClustering(S)
11 Function OfflineClustering(SavedState):
12     T ← Flatten(SavedState)
13     ε* ← +∞; k ← 2; Clusters ← ∅
14     S* ← ∅
15     while T ≠ ∅ do
16         do
17             Clusters, ε ← KMeansClustering(T, k)
18             if ε < ε* then
19                 ε* ← ε
20                 k ← k + 1
21         while ε = ε* and k < |T|
22         forall C ∈ Clusters do
23             if WTS(C) ≥ τ then
24                 S* ← S* ∪ {C}
25                 T ← T \ C
26     S ← MergeClusterState(S*, Flatten(S) \ Flatten(S*))
```

In SYNAPSE's application scenario it makes no sense to remove outliers. Instead, when new tweets do not belong to S, we treat them as the onset of a threat by adding new clusters with a single element which in time may receive additional tweets. This outlier processing mechanism allows adapting the number of clusters, k, to the novelty in the dataflow. Furthermore, it is through the online component of DynamicClustream that the active threat monitor is implemented: the system categorises new tweets as *new threats* or as *updates* to known ones, thus maintaining an updated threat summary about an IT infrastructure.

5.3.5 Cohesion Measure

Cluster cohesion and cluster separation are concepts used to assess the validity of a partition generated by a clustering algorithm [48], which in most cases have a purely geometric interpretation. In SYNAPSE, cohesion is based on the similarity of tweets within a cluster and not on a geometric measure such as the distance to the cluster centroid, thus defining a context-based cluster validation approach, argued to be more effective [78].

To reinforce the one-to-one relation between clusters and threats, the cohesion measure must detect clusters whose tweets refer to the same threat. Assuming that a threat is expressed by a minimum number of words appearing in all tweets, the proposed cohesion measure—named *Within-cluster Threat Similarity* (WTS)—is defined as $\frac{\omega}{w_m}$, where ω is the number of words shared by all the cluster's tweets and w_m is the number of words of the smallest tweet in the cluster. WTS is 0 if no words are shared by the tweets of a cluster, and 1 when all tweets share the words of the smallest tweet in the cluster. It assumes that if all cluster tweets share a sufficiently large number of words, then they mention the same threat.

The degree of separation of two clusters C_i and C_j is measured by the Jaccard index [135]. It is determined as $J = \frac{|C_i \cap C_j|}{|C_i \cup C_j|}$, corresponding to the ratio between the number of common words to C_i and C_j and the number of unique words of C_i and C_j. The lower its value, the more separated the clusters are.

5.3.6 Offline Clustering Component

The offline component applies the *k*-means clustering algorithm [90] repeatedly to provide more robust clusters. *k*-means is a widely used algorithm that has provided good efficiency and empirical success over the last 50 years [80]. However, it is commonly employed for exploratory data analysis, not for automatic text summarisation.

The *k*-means algorithm requires the specification of the number of clusters, k, which is unknown in this case. At a given time we do not know how many potential threats to our infrastructure are being discussed. Therefore, we defined a novel strategy to find the so-called elbow point [128], *i.e.*, the point beyond which by increasing k there is no significant improvement in the clusters' Sum of Squared Errors (SSE). This procedure automatically determines k, thus avoiding the specification of a threshold to find the elbow point or the visual inspection of the within-class-variance versus k graph.

5.3.6.1 *k*-means Application Strategy:

Starting at $k_1 = 2$, a *k*-means model is trained for each successive $k_i = i + 1$ number of clusters, which produces a corresponding SSE, denoted by ε_i. As the initial cluster centres are randomly chosen, there is a given variance σ_i associated to ε_i. As we keep increasing k_i, we expect ε_i to decrease up to the point where the magnitudes of ε_i and σ_i become of the same order. At this point $\varepsilon_{i+1} - \varepsilon_i$ might become zero or even negative, indicating that there is no significant SSE improvement in increasing k_i. Therefore, the iteration is stopped when the error (ε) stops decreasing or (the limit case where) the number of clusters corresponds to the number of tweets to be clustered, and k_i is selected as the number of clusters (lines 16–21).

By testing this approach, we found that small clusters had only very similar tweets, but other large clusters contained unrelated tweets. The cause might be two-fold: (1) *k*-means assumes spherical clusters that it tends to produce equally sized, which might not be adequate; and (2) the strategy to find k is not guaranteed to find the *best* k. To overcome this limitation, we use the WTS cohesion measure to quantify how closely related the tweets in a cluster are, and implement a *re-clustering method* that splits these clusters into smaller ones with related tweets. If WTS $\geq \tau$ (a specified threshold), indicating high cohesion, it enables the validation of clusters as *final*.

5.3.6.2 Re-Clustering Method:

All tweets of non-final clusters are gathered (line 22–25) and re-clustered (lines 16–21) using *k*-means to allow similar tweets to be grouped. Then, the new clusters generated are again tested using their WTS, and the process is repeated for the non-final clusters. Eventually, all clusters are considered final, ideally each related to a single threat, and S^* is merged with S (line 26), i.e., S^* is updated with the tweets received since the algorithm started by executing a procedure similar to lines 3–10.

5.3.6.3 Offline Clustering Scheduling:

At any time, there may be only one instance of the offline component in execution. Since multiple tweets received in a short time interval may trigger offline clustering, we employ the schedule keyword (line 10) to avoid overlapping executions. The idea is that each call to **schedule** OfflineClustering() notifies the system that offline clustering is required after this

point, and saves the current cluster state for its next execution. Once the algorithm is started again (using the latest saved state), it process all tweets pending in S (line 12).

5.3.7 Time-Window Model

To fully adapt Clustream to our context we also changed the clustering ageing model used to remove clusters. This model is necessary to complete the adaptation of the cluster state to the data stream flow.

Clustream's window model is global in the sense that all data points are aged and removed using the same rule. However, this methodology does not fit SYNAPSE application domain, as different cybersecurity topics have different lifetimes. For example, news about an update are expected to last a few days, while advances about an active threat may continue for a month or more. Thus, in the cybersecurity field it makes more sense to adopt a local window model, monitoring ageing *by cluster* (by threat). As a consequence, whole clusters rather than single points should be removed in forthcoming clustering states.

In DynamicClustream a cluster C_i is removed from the cluster state S if it has been stale for a period of time longer than θ, *i.e.*, if θ time passes without C_i receiving a new data point. In this way, topics that no longer receive traction are stowed away, *while active topics retain all their elements, regardless of the time passed, which may be crucial for understanding the evolution of a threat*.

5.4 Experimental Setup

This section describes the experimental work carried out to validate SYNAPSE. All code is written in Scala and deployed on the Apache Spark Framework [4].[1] We chose Spark as its data-structures are scalable and designed for large datasets. Also, Spark includes a scalable machine learning library called MLlib, used to implement all ML algorithms employed in this chapter.

5.4.1 Clustering

SYNAPSE uses the *k*-means algorithm in the offline clustering component, configured with fifty iterations, a minimum of two clusters, and the remaining parameters with their

[1]The source code is available at https://github.com/fernandoblalves/ScalaTweets.

default values. Clustering was performed on the set of tweets classified as positive by the selected models.

The WTS cluster cohesion measure was set to $\tau = \frac{2}{3}$. This value was selected after preliminary experiments, reflecting the rationale that two tweets can be in the same cluster if and only if they share at least two-thirds of their words.

We compare our data presentation strategy with the one employed by threat intelligence tools and SIEMs capable of collecting OSINT (*e.g.*, AlienVault OTX [2], Spiderfoot [32]). For that, we set up a Logstash [18] instance fed by the same dataset as SYNAPSE, which selected as relevant tweets mentioning at least one of our infrastructure assets and containing at least one security concept.

The security concept keywords were selected using the following methodology. First, a list of documents is obtained by selecting all tweets labelled as positive from all datasets. After that, we removed stopwords, applied the TF-IDF method, and selected the words with TF-IDF value lower than a threshold ρ. Finally, the list was manually filtered for security-irrelevant content (such as numbers). We considered ρ values of 0.1, 0.2, and 0.3. After inspecting the results, $\rho = 0.2$ was chosen due to the provision of the most substantial amount of generic words without showing words related to a specific context. The Logstash security concept keyword set corresponding to $\rho = 0.2$ appears in Table 5.2.

For the time-window model we applied a θ value of seven days, *i.e.*, a cluster without updates for seven days is removed from the online clustering state. The same θ value was applied to the Logstash approach but globally, *i.e.*, all relevant tweets were removed from the active threat pool after a week.

5.5 Results

The tweet processing pipeline components were evaluated using the selected models and datasets `D2` and `D3`. These consider only tweets in the future of those in the training set (`D1`),

Table 5.2: The words used in the Logstash filter.

access, acl, admin, advisory, allow, arbitrary, aslr, assurance, attack, auth, buffer, bug, bypass, certificate, code, command, corruption, csrf, cve, cyber, denial, deployment, dereference, disclosure, execute, exploit, hack, heap, identity, injection, interception, leak, overflow, privilege, remote, root, scripting, security, stack, threat, unauthenticated, vuln, xss

and include information posted by an additional and substantially larger set of accounts (S2) not considered in the training stage. This methodology embodies the idea that in a real deployment, models will classify future tweets possibly from a different set of accounts.

Considering that 10-fold cross-validation was employed during the model selection phase, it should be noted that the selected model configurations were trained for the evaluation phase using the whole D1 dataset. The feature vectors of D2 and D3 tweets were generated using the TF-IDF model determined using dataset D1. This guarantees that TF-IDF weights attributed to words in D2 and D3 will be coherent with those used to train the classifiers.

5.5.1 Clustering

The following experiments evaluate SYNAPSE's ability to aggregate the dataflow into meaningful clusters, where each cluster is expected to describe a single threat. Further, the DynamicClustream's window model is evaluated to assess its capability to detect the continuous discussion of threats.

The initial clustering evaluation focuses on the basic algorithm's capability of properly aggregate tweets, *i.e.*, producing clusters with high internal cohesion and low inter-cluster similarity. Then we analyse the end-to-end benefit of SYNAPSE and discuss the effectiveness of the proposed outlier detection mechanism and time-window model which convey the active threat monitor functionality to SYNAPSE.

Datasets D2 and D3 were merged and fed to SYNAPSE. At the end of each day, for all clusters in the current cluster state, we calculated the average WTS and the Jaccard distance between all pairs of clusters. For the latter, we saved the largest value, which corresponds to the most similar cluster pair. Since SYNAPSE's objective is to obtain distinct clusters, each devoted to a single threat, the WTS should always be high (*i.e.*, the elements in each cluster are very similar), and the maximum Jaccard distance should be low (*i.e.*, there are no clusters that should be merged).

Figure 5.3 shows the WTS and maximum Jaccard distance obtained, comparing the proposed DynamicClustream clustering algorithm (DC-WTS and DC-J) to its execution in clustering only mode, without considering re-clustering (NR-WTS and NR-J). The importance of including the re-clustering step (lines 22-25 of Algorithm 1) is clear since it raises the WTS to above 90% independently of the number of clusters and tweets present in the cluster state. The Jaccard distance, although with small values, is higher when using the re-clustering algorithm. Yet, this is an expected result. First, re-clustering produces significantly more

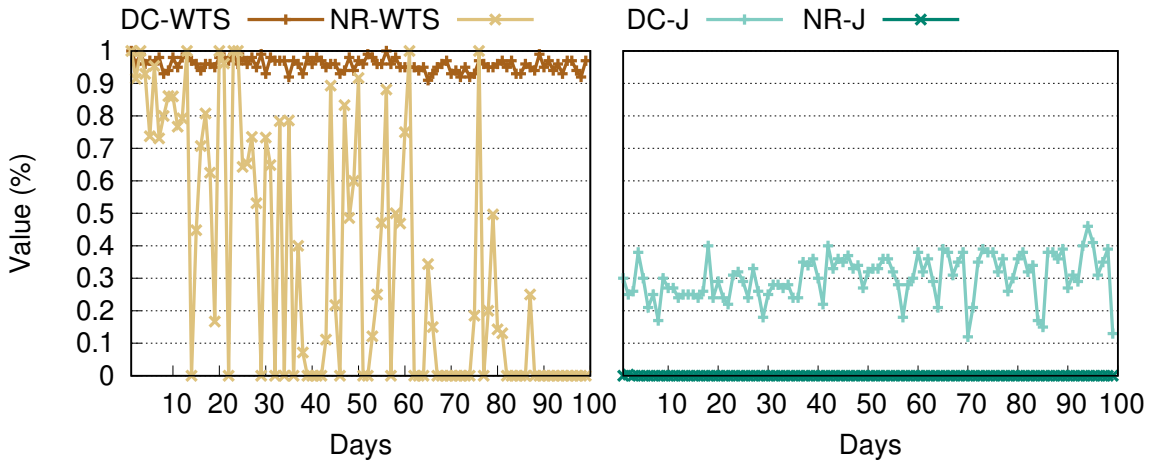


Figure 5.3: Comparing WTS and Jaccard distance over time, for DynamicClustream with and without the re-clustering step.

clusters, therefore naturally decreasing their degree of separation. Second, since tweets in clusters mentioning different threats are likely to share commonly used security concept words and sentence structure, their similarity is increased.

Regarding the number of clusters obtained using either approach, the re-clustering algorithm naturally increases the number of clusters, as shown in Figure 5.4. Nevertheless, we argue that in practice, the DynamicClustream algorithm improves the balance between maximising the relevance of the information presented and minimising the time required for its analysis. The WTS results provide guarantees that each cluster has similar tweets, likely about a single threat. Therefore, we can be confident that the set of cluster exemplar tweets provides a complete and accurate summary of the current threat landscape, thus not requiring additional time to analyse more tweets. Without the WTS cohesion validation, each cluster may discuss various threats—a highly plausible assumption based on the very low WTS values in Figure 5.3 for the NR-WTS case—meaning that all tweets of each cluster would have to be analysed.

5.5.2 End-to-End Benefit

The results presented in Figure 5.5 highlight the end-to-end benefit of using SYNAPSE, and reinforces the importance of its clustering stage. The figure shows the reduction in the number of tweets that have to be analysed, when compared to the tweet stream, to the classifier output and to the naive Logstash filter described in Section 5.4.

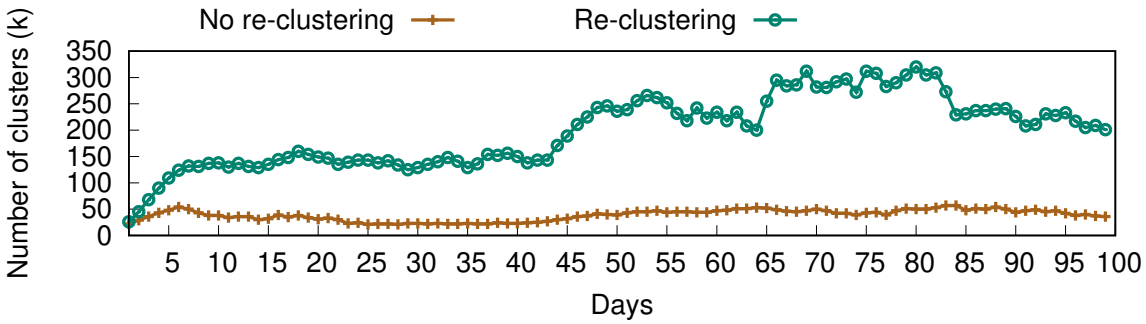


Figure 5.4: Number of clusters obtained by the DynamicClustream algorithm with and without the re-clustering step.

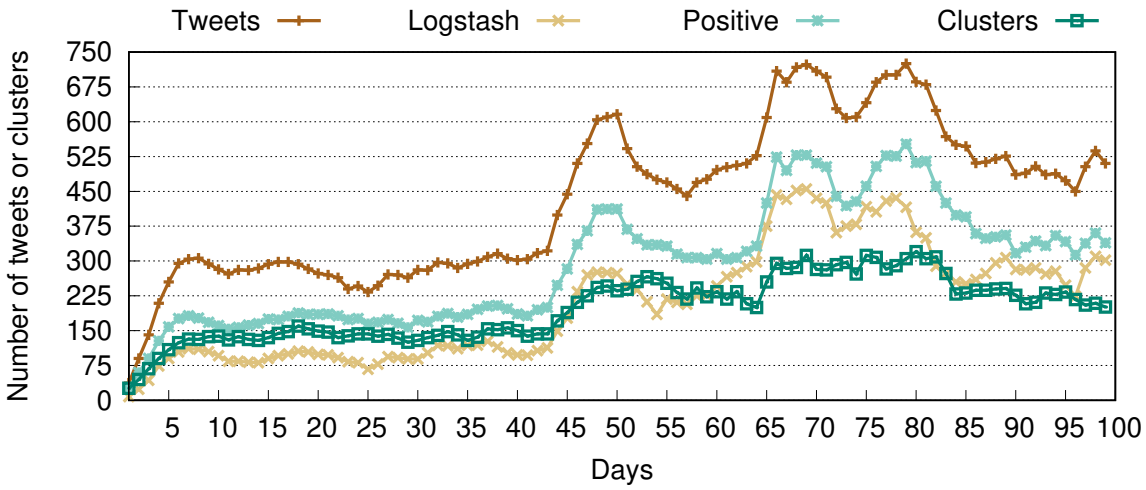


Figure 5.5: The number of tweets collected and those filtered by Logstash, classification only, and classification and clustering.

The results show the need for efficient OSINT retrieval tools. Even with the naive keyword-based approach provided by the Logstash filter, the number of tweets marked as relevant would be extremely high, rendering the approach useless to SOC analysts. The introduction of a trained classifier decreases the amount of information by 65%. By attaching a clustering stage, we further reduce the information to be shown by almost 80%, which is a significant improvement.

5.5.3 Active Threat Monitor

To demonstrate the necessity of the active threat monitor implemented by the proposed stream clustering algorithm, we measured the active time for each of the 820 clusters formed during SYNAPSE's operation on the union of datasets `D2` and `D3`. We define the duration of a cluster as the difference in days between the date of its creation and the date of the last added tweet. Figure 5.6 depicts the distribution of the number of clusters over the cluster duration in days. The results clearly show that a global time-window model enforcing a fixed duration for each tweet would fail to detect active topics through time, since the threat discussion duration varies greatly (between 1 and 57 days), even in a dataset that covers only 100 days.

5.5.4 Analysis of Generated IoCs

Besides the ability to accurately select and aggregate tweets relevant to the security of an IT infrastructure, SYNAPSE provides useful threat intelligence for SOC analysts. To demonstrate this, we present some information about the timeliness, actionability, and relevance of the IoCs generated from the dataset used in previous experiments.

From the data collected over 3 months, SYNAPSE generated 820 clusters (IoCs) containing 1754 tweets. From these, we selected those with 5 or more tweets for analysis, obtaining 65 clusters comprising 466 tweets. These clusters are listed in Appendix A. The remaining

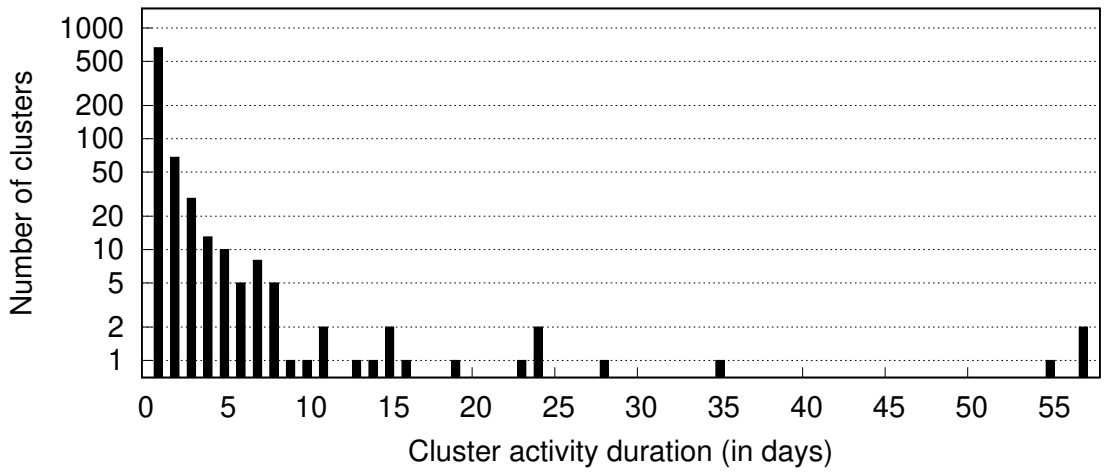


Figure 5.6: The distribution of the number of clusters over the cluster duration in days.

755 clusters have 1 (577 clusters), 2 (101), 3 (55), and 4 tweets (22). Our focus on larger clusters was motivated by the expectation that relevant threats are probably those that attract more attention and, ultimately, are mentioned in more tweets.

All tweets within each cluster were manually analysed. From these, as well as from any hyperlink therein, we extracted all CVEs mentioned (if any) and their Common Vulnerability Scoring System v3.0 (CVSS) [7] impact score, the types of actions that can be performed to respond to the alarm, and a comparison between the date of the earliest tweet in the cluster and the CVE's publication date on NVD.

The actionability information was divided into three categories: a patch is available (45 occurrences); a configuration to avoid the vulnerability exploitation is suggested (2 occurrences); and no directly actionable information is provided (14 occurrences). The latter is mostly associated with clusters mentioning exploits to vulnerabilities, with the tweet hyperlinks leading to proofs-of-concept. However, an expert might still make use of this information to prevent exploitation, as discussed in previous work [111]. Patches are mostly announced together with their associated vulnerabilities, regardless of indexing on NVD. In the end, 71% (46) of the clusters provided directly usable intelligence, including exploits whose vulnerabilities were not matched to NVD entries. Among the 65 clusters, 36 mentioned a total of 122 different CVEs (15 clusters mentioning more than one CVE). Of these, only two have low impact score, about a quarter have medium impact (33), more than half are categorised with high impact (68), and more than a tenth have critical impact (14). Considering their relevance, 43% (28) of the IoCs were related to CVSS scores above or equal to 7 (high severity) and 12% (8) to scores above or equal to 9 (critical severity). Regarding timeliness, 20% of the alerts were raised 8 days (on average) before their corresponding vulnerabilities were published on NVD.

As an illustration of the richness of the obtained data, Table 5.3 shows 10 representative IoCs selected from those analysed. In the table, the date column shows the date of the earliest tweet in the cluster and, when a number is shown within parenthesis, it denotes the number of days before publication on NVD. Two additional columns provide information about the threat type (as automatically classified by SYNAPSE) and relevant notes about the cluster content.

From the 10 clusters presented, 6 announce vulnerabilities before publication on NVD, all of them with patches available. Further, 7 are classified with a *high* CVSS and two with *critical* impact. For example, the 7th IoC of the table shows a critical Cisco router vulnerability patched and published three days before its inclusion on NVD. Finally, since

Table 5.3: Examples of tweets whose content has high impact or important actionability.

Cluster exemplar text (without links)	#	Asset	Date	Action	Threat type	Notes
#ubuntu #security : USN-3006-1: Linux kernel vulnerabilities	19	Linux	10/06	Patch	vulnerabilities	Several vulnerabilities patched, some were not yet included on NVD, half with CVSS ≥ 7.0
High - USN-3016-1 - Linux kernel vulnerabilities A security issue affects these releases of Ubuntu and its derivat	12	Linux	27/06	Patch	vulnerabilities	Several vulnerabilities patched, some were not yet included on NVD, half with CVSS ≥ 7.5
Microsoft Internet Explorer CVE-2016-3205 Scripting Engine Remote Memory Corruption Vulnerability Type: Vulnerabil	8	IE	14/06 (1)	Config	vulnerability, remote	This cluster contains various threats with CVSS ≥ 7.5; configurations are suggested to mend the issue before it is patched
#CISCO fixed severe #vulnerabilities in Network Management and #Security Products #SecurityAffairs	9	Cisco	30/06 (2)	Patch	vulnerabilities	Patch for critical vulnerabilities (CVSS ≥ 8.6) announced on Twitter before being published on NVD
Bugtraq: [security bulletin] - Linux Kernel Flaw, ASN.1 DER decoder for x509 certificate DER	6	Linux	06/06 (21)	Patch	certificate	A highly important Linux kernel flaw (CVSS 7.8) was disclosed 21 days before being included in NVD
Vuln: Oracle Java SE and JRockit CVE-2016-3427 Remote Security Vulnerability Vulnerable:Red Hat Enterprise Linux	21	Oracle	05/07	Patch	vulnerability, remote	This cluster contains three different threats (one with CVSS 9.0); patches are available
Bugtraq: Cisco Security Advisory: Cisco RV110W, RV130W, and RV215W Routers Arbitrary Code Execution Vulnerability	5	Cisco	15/06 (3)	Patch	vulnerability, execution	A critical vulnerability (CVSS 9.8) was disclosed and patched before its inclusion on NVD
Bugtraq: Cisco Security Advisory: Cisco Products IPv6 Neighbor Discovery Crafted Packet Denial of Service	5	Cisco	25/05 (4)	Patch	denial of service	A high impact vulnerability (CVSS 7.5) was disclosed and patched before its inclusion on NVD
#ubuntu #security : USN-2975-2: Linux kernel (Trusty HWE) vulnerability	5	Linux	16/05 (42)	Patch	vulnerability	A high impact vulnerability (CVSS 7.8) was disclosed and patched before its inclusion on NVD (42 days in advance)
Bugtraq: Wordpress Levo-Slideshow 2.3 - Arbitrary File Upload Vulnerability	9	WPress	07/06	Config	vulnerability	An exploit is provided; a software correction is suggested

not all occurrences are patched at disclosure time, some actionable IoCs contain suggested configurations to avoid exploitations. As an example, the last row in the table shows a WordPress exploit with suggested remediations.

These results show the edge obtained by using Twitter as a security data source. A SOC analyst using SYNAPSE would obtain timely and relevant data about patches to known vulnerabilities, thus possibly reducing the vulnerable system's exposure time. Further, the results also show that vendors publish important impact data before it is included in NVD.

5.6 SOC Integration

An essential aspect of threat intelligence tools such as SYNAPSE is the integration in a SOC. In the following, we describe practical issues related to this integration.

5.6.1 Adversarial Model

When using Twitter as a cybersecurity information source, it is important to consider what would happen if some of the monitored accounts fall under the control of the adversary. In a nutshell, two things can happen: (1) the adversary may not tweet about the threats he is interested in exploiting using the accounts he controls; or (2) the adversary may create tweets with false threats to make SOC analysts waste their time in solving potential non-existent problems. These attacks would reduce the confidence of the SOC analysts in the system, reducing its usefulness. To avoid or reduce the impact of these attacks, SYNAPSE should have (as future work): 1) a method to evaluate the usefulness of the accounts in use, and automatically search for useful accounts; and 2) a method to receive feedback from analysts (discussed further below).

5.6.2 Training the System

Our approach requires the creation of labelled datasets for training the classifiers. To do that, the SOC analysts need first to configure the keywords defining the infrastructure. A second configuration step is to define the Twitter accounts that will be monitored.

After those two steps, the system should present all filtered tweets as if they are important, and a button for the analyst to mark a tweet as "irrelevant".[1] Notice that, to avoid bias, it is relevant to inform the analysts that the system is under training. When enough positively-labelled tweets are collected, the classifiers can be trained in background and then placed in operation.

It is expected that the classifier's performance decreases with time, as the operational data gets progressively different from the training data. To maintain the utility of the classifiers in use, it is essential to minimise this effect. Incremental learning is a technique that can be used for this purpose, where the classifier's model is continuously trained with new labelled

[1] The "irrelevant" button must always be available, even when the system is not being trained, in order to collect wrongly classified tweets for future retraining.

examples [76]. By training the model with the latest events, it is continuously adapted to changes in input format (in this case, changes in tweet format or language).

Another possibility is to replace the model with a new model trained with only the latest data, *e.g.*, the last three months of tweets. This way the model is periodically adapted to the current threat landscape, so that old data will not impact the classifier's quality.

5.6.3 Changing Keywords and Monitored Accounts

Adding or removing keywords from the datasets require retraining the classifier. Removing a keyword requires removing the tweets that were filtered by this keyword and retrain the model without them. To add a keyword, one needs first to complement the existing labelled dataset (in the same way as described before) with tweets related to the new keyword, and then retrain the model with the reformulated data set. Changing the set of monitored Twitter accounts is not a burden for the system since the structure of threat descriptions is expected to be similar across all security accounts. The datasets employed in our experimental evaluation consider this possibility.

5.6.4 SYNAPSE Integration with a Real SOC/SIEM

In the following we present some highlights of SYNAPSE integration with the SIEM of a nation-wide electric power utility. The first step was to provide a dashboard so the SOC operators could visually inspect the latest security events.

Figures 5.7 and 5.8 present the developed dashboard, representing respectively the enriched tweet clusters and the collected data volume. However, SOC operators require data centralization in the SIEM to guarantee streamlined workflows; data originating from various sources has to be correlated and their attention cannot be dispersed throughout multiple dashboards. Therefore, we developed a connector to place SYNAPSE's IoCs in the SIEM (which was trivial as SYNAPSE was designed considering integration with threat intelligence tools), and the SOC operators developed a new SIEM dashboard so they could observe SYNAPSE data.

Once SYNAPSE was directly connected to the SIEM we raised our attention to how to use tweets together with infrastructure events. Discussing with the SOC operators how the data could improve the SIEM capabilities, three integration actions were implemented. The first was to create a rule in the SIEM that triggered an alarm when a cluster with more

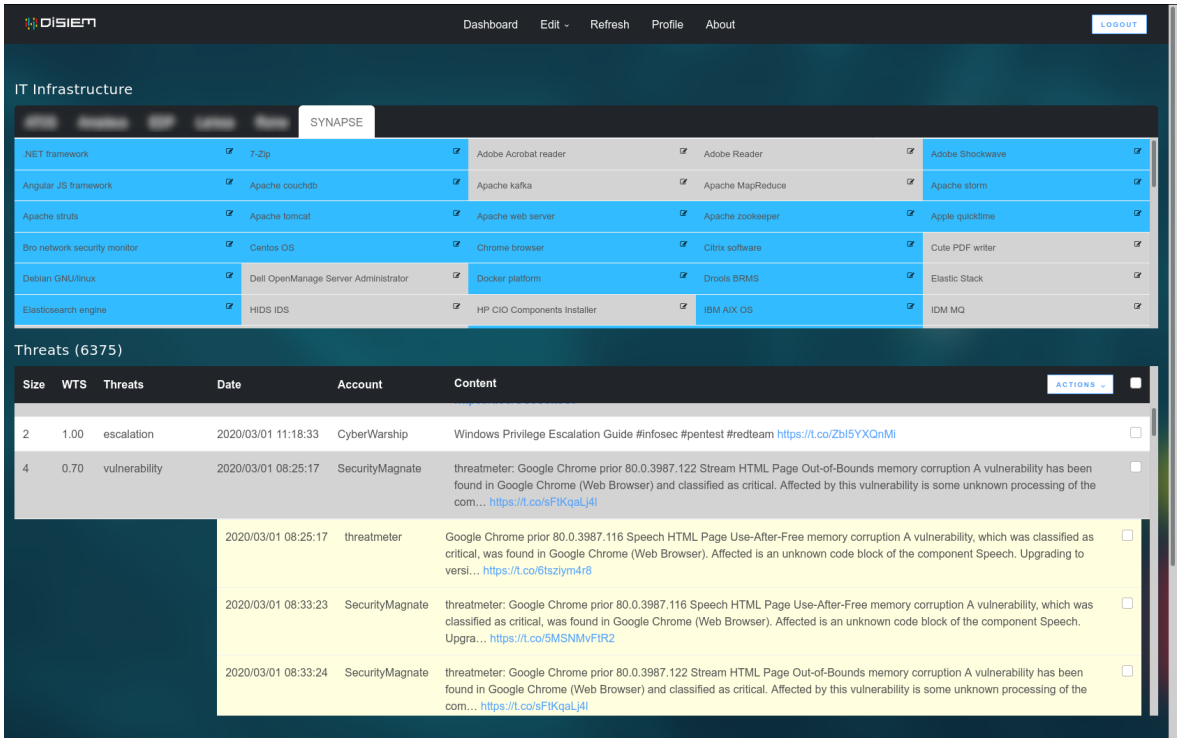


Figure 5.7: An overview of SYNAPSE's dashboard. It is possible to view all collected threats (as depicted), or to select only some assets. Furthermore, each threat can be analysed only by its exemplar or in its entirety.

than five tweets was received. Although simple, this rule selects only events likely to be of importance.

The second action complements internal events with external descriptions. The SOC includes in its infrastructure a firewall that tags identified threats with their corresponding CVE-IDs. Since SYNAPSE's IoCs include security bulletin IDs (such as CVE-IDs), the SIEM was able to match the events. The connection between internal events (firewall detected threats) and external events (tweets with security bulletins for those threats) improved the quality of SOC operation as the firewall events only mention IDs, lacking an accompanying description of the threats—something provided by the tweets.

The third action is related to prioritizing security actions. Managing a large IT infrastructure raises many complex problems. One of them is updating software in many different machines with different purposes. Updating system images (composed of operating system and various software elements) implies a patching phase followed by a testing and compliance phase aimed at detecting incompatibilities with the different software in use. In practice,

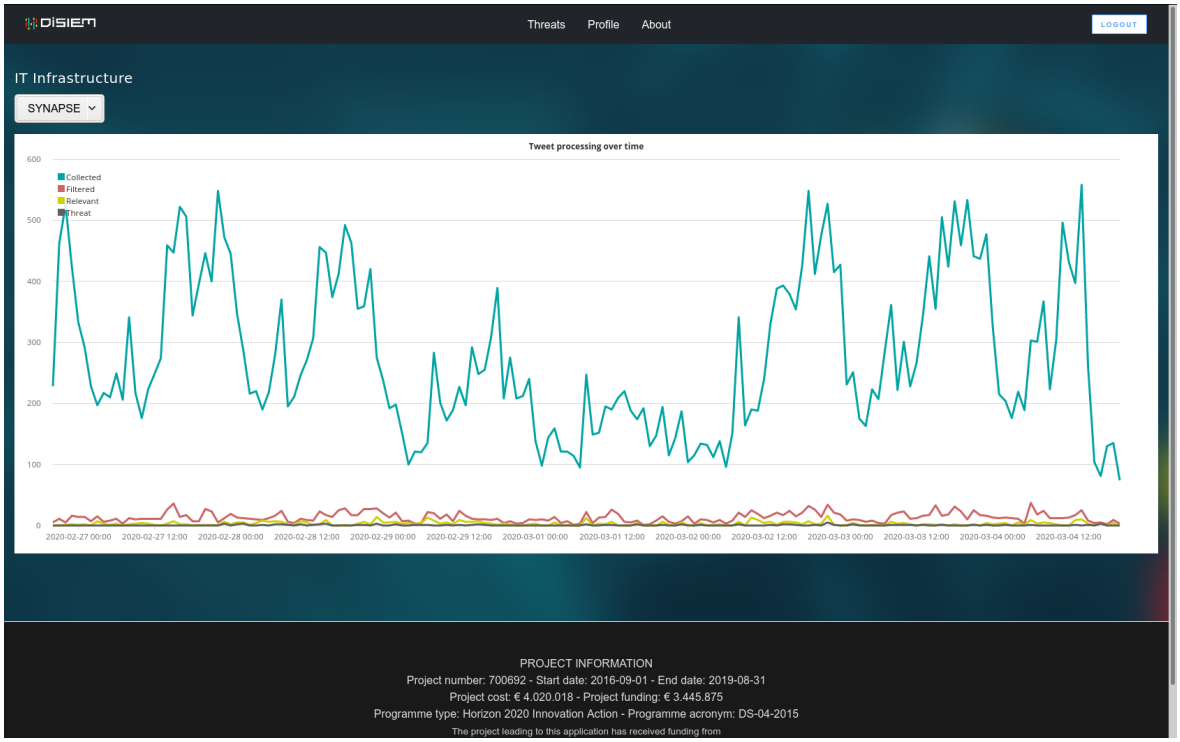


Figure 5.8: The collected data volume. The lines represent: in blue, the total number of tweets collected; in red, how many refer the mentioned infrastructure assets; in yellow, the number of tweets deemed relevant for cybersecurity; and in brown, the number of threats detected by SYNAPSE (post aggregation).

updating system images can take about a month, sometimes more. SYNAPSE was praised by the SOC operators as it provided them with much needed awareness of the current cyberthreats, and thus they were able to better prioritize the patching process because it was simpler for them to understand the criticality of each patch.

5.7 Deep Learning Extensions

The base SYNAPSE work was dedicated to establishing an end-to-end streaming feed from Twitter to the SOC. This pipeline was later improved with more refined machine learning techniques. This work was reported in a master's thesis [68], and summarised in the following.

5.7.1 Classifier refinement and Named Entity Recognition

This first extension improved two components, namely the pipeline's classifier and the IoC generator.

5.7.1.1 Classifier

Instead of having a simple MLP or an SVM, the SYNAPSE classifier can be replaced with neural network architectures known for better performance, namely a Convolutional Neural Network (CNN). The convolutional layers are better at identifying connections between the separate words, thus better at processing sentences and more accurately determining their relevance. The increase in performance is notable (in some cases almost 40% increase in TNR), always outperforming SVMs and MLPs, while always keeping a solid balance between TPR and TNR.

5.7.1.2 IoC Generation

The IoC generator identified some words of interest on a tweet based on a word set. This mechanism can be improved by complementing the word filter with a Named Entity Recognizer (NER) [97]. This component relies on a Bidirectional Long Short-term Memory neural network (BiLSTM) to identify key elements in a sentence, such as subjects or verbs. In this case, our NER was tailored to identify five elements: names of companies or organizations, names of products or assets, versions of products or assets, references of vulnerabilities or threats, and identifiers such as CVE IDs. These elements were added to the IoCs to enrich their content. SYNAPSE's NER was able to identify all pertaining elements of a sentence with an F1 score above 90%.

5.7.2 Multi-Task Deep Neural Network

Multi-Task Learning (MTL) [58, 61] is a technique where a single model is trained to perform different tasks, such as classification and NER. Several works in natural language processing show that MTL can improve a model's performance in tasks in the same domain and generalization capabilities, thus reducing overfitting [51, 88]. The second SYNAPSE deep learning extension was to use MTL on the models described in the previous section. This MTL model was created as a fork after the CNN/LSTM layers; the tweet is sent to both

paths, one performs classification, the other performs NER, just as described in the sections above. This technique led to an improvement of 0.5% F1 score for both tasks.

5.8 Conclusions

This chapter proposes SYNAPSE, a Twitter-based streaming threat monitor for threat detection in security operation centres. It implements a pipeline that gathers tweets from a set of accounts, filters them based on the monitored infrastructure, and classify the remaining tweets as either relevant or not. Relevant tweets are grouped in dynamic clusters and presented as indicators of compromise that can be either manually inspected or fed to SIEMs and other threat intelligence tools. We performed an evaluation over these IoCs, showing that highly relevant, timely and actionable information was collected, illustrating the value of our end-to-end approach. Finally, we present how SYNAPSE was integrated with a SIEM and their events correlated, together with a set of insights for future works.

6
Conclusions and Future Work

6.1 Conclusions

This thesis is focused on validating and using Twitter for cybersecurity. The first step was a study comparing Twitter to vulnerability databases, as well as a short overview of these databases. This study yielded that Twitter is a useful cybersecurity data source. In more detail, we gathered the following results:

- No OSINT source can be considered clearly better than others and therefore diverse OSINT sources should be used as they complement each other;
- When considering only confirmed vulnerabilities, NVD should not be the unique vulnerability database subscribed;
- Since 2010, Twitter provides an almost perfect vulnerability coverage;
- Twitter discusses vulnerabilities ahead of databases for very few cases (about 1% for the vulnerabilities examined), and is as timely as the vulnerability databases for the remaining cases;
- Most of the vulnerabilities reported early on Twitter have a high or critical impact, with the tweet leading to usable mitigation measures.

We also collected a set of insights for the security practitioners interested in using Twitter for cybersecurity, originated from our inspection of of almost one million tweets, and analysing many thousands of vulnerabilities.

After validating that Twitter should be used for cybersecurity, we set to build a tweet collection system that follows the requirements gathered during the study. Thus we built SYNAPSE, a Twitter-based streaming threat monitor for threat detection in security operation centres. The SYNAPSE pipeline begins with a tweet gathering connector, that collects every tweet posted by a set of accounts. These tweets are filtered according to a user-provided list of assets of interest. A supervised machine learning classifier selects the tweets relevant for cybersecurity, which are sent for clustering on a novel stream-clustering algorithm. The clusters are transformed into IoCs, that can be either manually inspected or fed to SIEMs and other threat intelligence tools. Results show that our system maximises the relevant information (true positive rate of 90%), minimises irrelevant information (false positive rate of 10%), and aggregates related information (only 21% of the relevant tweets are presented). Further, SYNAPSE is designed to handle the tweet stream, thus being wholly designed to include the time dimension—something lacking in the literature. SYNAPSE was developed using insights from SOC operators we worked with, and SYNAPSE was integrated with the SIEM managed by those operators. We also present some insights based on that integration effort for any future tools developed with similar purposes.

6.2 Future Research Directions

The study we performed filled a gap on the literature, and SYNAPSE advanced the state-of-the-art in the area of tweet-processing tools. However, there are still research directions to explore.

Account management over time. The set of accounts used by SYNAPSE to collect tweets is set at the system's start and unchanged, unless the user manually does it. The ideal setting would be to have a recommender system autonomously managing the accounts used by SYNAPSE, *i.e.*, removing closed accounts, removing accounts that no longer tweet about cybersecurity, and discovering and adding relevant accounts.

The recommender system would use the infrastructure keywords to collect directly from the Twitter stream instead of using the selected accounts (see Section 4.1 for details). The collected tweets would go through the classifier, and from those deemed as relevant we collect the accounts posting them. From here, two strategies would be studied for account to be added: to achieve a threshold of minimum relevant tweets, or to set a ratio of relevant to non-relevant tweets.

Manage SYNAPSE's classifier over time. Although SYNAPSE is designed considering the time dimension, the classifier is static, *i.e.*, once trained it is never updated. Over time, it is expected that the classifier will lose performance as the accounts change, the tweet wording changes, or as new terminology is introduced. Therefore, the classifier needs to be aware of these changes.

The classifier would have to be managed in two dimensions: the dataset used to train, and the model training over the said dataset. During SYNAPSE operation, new tweets would have to gathered for model training; this is considered in SYNAPSE's SoC integration (Section 5.6). We would have to study which tweets to use for model management: either a fixed number of tweets, or all tweets within the latest time interval, or selecting the best tweets for training. The latter criteria would involve a new measure that would evaluate the quality of a tweet regarding its writing and actionability, which are two most relevant aspects a tweet can bring to SOC.

About the model management, we would test a series of possibilities, such as incremental learning [65, 75, 104, 126], full model retraining, model redesign, and combinations of these approaches. Another aspect to study is when to manage the model, since model training is computationally expensive. Model management could be started when: a threshold of new labelled tweets is gathered, periodically, through model metrics, or a combination of these approaches. Model metrics could be gathered by evaluating the model against a new dataset (collected as mentioned above), and if the model would perform significantly worse, then the management would be triggered. Model redesign would bring the greatest challenge, since a pareto search is expensive, especially when considering deep learning models; a possible research direction would be to reduce the search from a brute force to a guided search, where only certain parts of the model undergo changes, and under a set of heuristics.

Large Language Models (LLM) in SYNAPSE. LLMs are trained with massive amounts of information, and is is possible to extract knowledge from it. In the context of SYNAPSE, LLMs could be used for two tasks: 1) replace the classifier and identify if a tweet is relevant for the cybersecurity of the SOC; and 2) provide context and actionability about collected threats, along with the sources for that information.

Acronyms

APT Advanced Persistent Threats. 2, 10, 14

CVE Common Vulnerabilities and Exposures. xiii, 10, 12, 19–22, 25, 29, 37–39, 48, 54, 67, 71, 73

IoC Indicator of Compromise. 7, 51–54, 66–68, 70, 71, 73, 74, 76, 95

IT Information Technology. 1, 2, 5, 6, 11, 41, 42, 44, 54, 55, 58, 66, 71, 95

MTL Multi-Task Learning. 73

NER Named Entity Recognizer. 73, 74

NLP Natural Language Processing. 3, 11

NVD National Vulnerability Database. 5, 10, 12, 13, 17–25, 35, 36, 38, 52, 67, 68

OSINT Open Source Intelligence. 3, 5–7, 9, 11, 12, 17–19, 21, 28, 36–39, 41, 42, 51, 54, 55, 62, 65, 75

SIEM Security Information and Event Management. 2–4, 7, 51, 53, 62, 70, 71, 74, 76

SOC Security Operations Center. 2, 3, 6, 41, 49, 52, 55, 65, 66, 68–72, 76, 77

References

[1] Additional paper data. https://github.com/fernandoblalves/Follow-the-Blue-Bird-Paper-Additional-Data. [Accessed 10-07-2020].

[2] AlienVault OTX, The World's First Truly Open Threat Intelligence Community. https://otx.alienvault.com/. [Accessed 13-02-2019].

[3] Anomali threatstream threat intelligence management. https://www.anomali.com/products/threatstream. [Accessed 18-07-2022].

[4] Apache Spark. http://spark.apache.org. [Online; accessed 05-December-2017].

[5] Common platform enumeration. https://cpe.mitre.org/about/. [Accessed 15-04-2020].

[6] Common vulnerabilities and exposures (cve). http://cve.mitre.org/. [Accessed 15-04-2020].

[7] Common Vulnerability Scoring System SIG. https://nvd.nist.gov/vuln-metrics/cvss. [Accessed 13-06-2018].

[8] Common vulnerability scoring system version 3.0. https://www.first.org/cvss/v3-0/. [Accessed 15-04-2020].

[9] Cvss v2 archive. https://www.first.org/cvss/v2/. [Accessed 15-04-2020].

[10] ENISA threat taxonomy. https://www.enisa.europa.eu/topics/threat-risk-management/threats-and-trends/enisa-threat-landscape/etl2015/enisa-threat-taxonomy-a-tool-for-structuring-threat-information. [Accessed 13-06-2018].

[11] Exploit database. www.exploit-db.com/. [Accessed 15-04-2020].

[12] Get old tweets programatically. https://github.com/Jefferson-Henrique/GetOldTweets-java. [Accessed 15-04-2020].

[13] Hackers say they hacked nsa-linked group, want 1 million bitcoins to share more. https://www.vice.com/en_us/article/ezpa9p/hackers-hack-nsa-linked-equation-group. [Accessed 15-04-2020].

[14] How people use Twitter in general. https://www.americanpressinstitute.org/publications/reports/survey-research/how-people-use-twitter-in-general/. [Accessed 15-04-2020].

[15] How Twitter evolved from 2006 to 2011. https://buffer.com/resources/how-twitter-evolved-from-2006-to-2011. [Accessed 15-04-2020].

[16] IBM QRadar SIEM. https://www.ibm.com/pt-en/marketplace/ibm-qradar-siem. [Accessed 15-02-2019].

[17] IntelMQ. http://github.com/certtools/intelmq/. [Online; accessed 05-December-2017].

[18] Logstash: Collect, Parse, Transform Logs. https://www.elastic.co/products/logstash. [Accessed 13-06-2018].

[19] Lookingglass cyber. https://lookingglasscyber.com/. [Accessed 18-07-2022].

[20] MISP - Open Source Threat Intelligence Platform & Open Standards For Threat Information Sharing. http://www.misp-project.org/. [Accessed 13-06-2018].

[21] MISP data models. http://www.misp-project.org/datamodels/. [Accessed 13-06-2018].

[22] MISP taxonomies. http://www.misp-project.org/datamodels/. [Accessed 13-06-2018].

[23] The mitre corporation. https://www.mitre.org/. [Accessed 15-04-2020].

[24] National vulnerability database. https://nvd.nist.gov/. [Accessed 15-04-2020].

[25] Netflow version 9 flow-record format. https://www.cisco.com/en/US/technologies/tk648/tk362/technologies_white_paper09186a00800a3db9.html. [Accessed 01-05-2022].

[26] New targeted attack in the middle east by APT34, a suspected iranian threat group, using cve-2017-11882 exploit. https://www.fireeye.com/blog/threat-research/2017/12/targeted-attack-in-middle-east-by-apt34.html. [Accessed 15-04-2020].

[27] Operation aurora. https://en.wikipedia.org/wiki/Operation_Aurora. [Accessed 15-04-2020].

[28] Packet storm. https://packetstormsecurity.com/. [Accessed 15-04-2020].

[29] The race between security professionals and adversaries. https://www.recordedfuture.com/vulnerability-disclosure-delay/. [Accessed 15-04-2020].

[30] Search engine for security intelligence — vulners. https://vulners.com/. [Accessed 15-04-2020].

[31] Security database. https://www.security-database.com/. [Accessed 15-04-2020].

[32] Spiderfoot. https://www.spiderfoot.net/documentation/. [Accessed 15-04-2020].

[33] The Open Source Elastic Stack. http://www.elastic.co/products. [Online; accessed 05-December-2017].

[34] Tweepy. https://www.tweepy.org/. [Accessed 15-04-2020].

[35] Tweet attacks pro. http://www.tweetattackspro.com/. [Accessed 15-04-2020].

[36] Twitter. https://twitter.com/. [Accessed 15-04-2020].

[37] Twitter input plugin. https://www.elastic.co/guide/en/logstash/current/plugins-inputs-twitter.html. [Accessed 15-04-2020].

[38] Veris taxonomy. http://veriscommunity.net/enums.html#section-incident_desc. [Accessed 13-06-2018].

[39] Watchguard firewalls - 'escalateplowman' ifconfig privilege escalation. https://www.exploit-db.com/exploits/40270. [Accessed 15-04-2020].

[40] Yousra Aafer, Wei You, Yi Sun, Yu Shi, Xiangyu Zhang, and Heng Yin. Android SmartTVs Vulnerability Discovery via Log-Guided Fuzzing. In *30th USENIX Security Symposium (USENIX Security 21)*, 2021.

[41] Charu C Aggarwal. *Data mining: the textbook*. Springer, 2015.

[42] Charu C Aggarwal et al. A framework for clustering evolving data streams. In *Proceedings of the 29th VLDB*, 2003.

[43] Subutai Ahmad, Alexander Lavin, Scott Purdy, and Zuha Agha. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, 262, 2017.

[44] Mohammed Almukaynizi, Eric Nunes, Krishna Dharaiya, Manoj Senguttuvan, Jana Shakarian, and Paulo Shakarian. Proactive identification of exploits in the wild through vulnerability mentions online. In *2017 CyCon US*, 2017.

[45] Nasser Alsaedi, Pete Burnap, and Omer Rana. Can we predict a riot? Disruptive event detection using Twitter. *ACM TOIT*, 17(2), 2017.

[46] Fernando Alves, Aurélien Bettini, Pedro M. Ferreira, and Alysson Bessani. Processing tweets for cybersecurity threat awareness. *Information Systems*, 2020.

[47] Ambrose Andongabo and Ilir Gashi. vepRisk - A Web Based Analysis Tool for Public Security Data. In *13th EDCC*, 2017.

[48] Olatz Arbelaitz, Ibai Gurrutxaga, Javier Muguerza, Jesús M. Pérez, and Iñigo Perona. An extensive comparative study of cluster validity indices . *Pattern Recognition*, 46(1), 2103.

[49] Ashish Arora, Ramayya Krishnan, Anand Nandkumar, Rahul Telang, and Yubao Yang. Impact of vulnerability disclosure and patch availability-an empirical analysis. In *Third Workshop on the Economics of Information Security*, 2004.

[50] Sean Barnum. Standardizing cyber threat intelligence information with the structured threat information expression (stix). *Mitre Corporation*, 11:1–22, 2012.

[51] Jonathan Baxter. A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine learning*, 1997.

[52] Vahid Behzadan, Carlos Aguirre, Avishek Bose, and William Hsu. Corpus and deep learning classifier for collection of cyber threat indicators in Twitter stream. In *IEEE Big Data 2018*, 2018.

[53] Jiang Bian, Umit Topaloglu, and Fan Yu. Towards large-scale Twitter mining for drug-related adverse events. In *Proc. of the SHB*, 2012.

[54] Avishek Bose et al. A novel approach for detection and ranking of trendy and emerging cyber threat events in twitter streams. In *Proceedings of the 2019 IEEE/ACM ASONAM*, 2019.

[55] Mehran Bozorgi, Lawrence K Saul, Stefan Savage, and Geoffrey M Voelker. Beyond heuristics: learning to classify vulnerabilities and predict exploits. In *16th ACM SIGKDD*, 2010.

[56] Benjamin L Bullough, Anna K Yanchenko, Christopher L Smith, and Joseph R Zipkin. Predicting exploitation of disclosed software vulnerabilities using open-source data. In *3rd ACM IWSPA*, 2017.

[57] Rodrigo Campiolo, Luiz Arthur F. Santos, Daniel Macêdo Batista, and Marco Aurélio Gerosa. Evaluating the utilization of Twitter messages as a source of security alerts. In *SAC 13*, 2013.

[58] Rich Caruana. Multitask learning. *Machine learning*, 1997.

[59] Defense Use Case. Analysis of the cyber attack on the ukrainian power grid. *Electricity Information Sharing and Analysis Center (E-ISAC)*, 2016.

[60] Mario Cataldi, Luigi Di Caro, and Claudio Schifanella. Emerging topic detection on Twitter based on temporal and social terms evaluation. In *10th MDM/KDD*, 2010.

[61] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, 2008.

[62] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3), 1995.

[63] Christine P Dancey and John Reidy. *Statistics without maths for psychology*. Pearson Education, 2007.

[64] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018. arXiv:1810.04805.

[65] Christopher P Diehl and Gert Cauwenberghs. Svm incremental learning, adaptation and optimization. In *Neural Networks, 2003. Proceedings of the International Joint Conference on*, 2003.

[66] Nuno Dionísio, Fernando Alves, Pedro M Ferreira, and Alysson Bessani. Cyberthreat detection from Twitter using deep neural networks. In *IJCNN 2019*, 2019.

[67] Nuno Dionísio, Fernando Alves, Pedro M Ferreira, and Alysson Bessani. Towards end-to-end cyberthreat detection from Twitter using multi-task learning. In *IJCNN 2020*, 2020.

[68] Nuno Dionísio. Improving cyberthreat discovery in open source intelligence using deep learning techniques, 2018. Available at `http://hdl.handle.net/10451/36434`.

[69] Yadolah Dodge. *The concise encyclopedia of statistics*. Springer Science & Business Media, 2008.

[70] Michel Edkrantz, Staffan Truvé, and Alan Said. Predicting vulnerability exploits in the wild. In *2nd IEEE CSCloud*, 2015.

[71] Martin Ester et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the KDD*, 1996.

[72] Mateusz Fedoryszak, Brent Frederick, Vijay Rajaram, and Changtao Zhong. Real-time event detection on social data streams. *25th ACM SIGKDD - KDD '19*, 2019.

[73] Wei Feng et al. Streamcube: hierarchical spatio-temporal hashtag clustering for event exploration over the twitter stream. In *Proceedings of the 31st ICDE*, 2015.

[74] Stefan Frei, Dominik Schatzmann, Bernhard Plattner, and Brian Trammell. Modeling the security ecosystem-the dynamics of (in) security. In *Economics of Information Security and Privacy*. Springer, 2010.

[75] Piyabute Fuangkhon and Thitipong Tanprasert. An adaptive learning algorithm for supervised neural network with contour preserving classification. In *International conference on artificial intelligence and computational intelligence*, 2009.

[76] Xin Geng and Kate Smith-Miles. *Incremental learning*. Springer, 2015.

[77] Sudipto Guha et al. Clustering data streams. In *Proceedings 41st FOCS*, 2000.

[78] Isabelle Guyon et al. Clustering: Science or art. In *Proc. of the 9th NIPS workshop on clustering theory*, 2009.

[79] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the 30th ACM STOC*, 1998.

[80] Anil K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8), 2010.

[81] Taoran Ji et al. Feature driven learning framework for cybersecurity event detection. In *IEEE/ACM ASONAM*, 2019.

[82] Rupinder Paul Khandpur et al. Crowdsourcing cybersecurity: Cyber attack detection using social media. In *Proceedings of the 26th ACM CIKM*, 2017.

[83] Ba-Dung Le et al. Gathering cyber threat intelligence from twitter using novelty classification. In *Proceedings of the 18th CW*, 2019.

[84] Quentin Le Sceller, ElMouatez Billah Karbab, Mourad Debbabi, and Farkhund Iqbal. Sonar: Automatic detection of cyber security events over the Twitter stream. In *12th ARES*, 2017.

[85] Kuo-Chan Lee et al. Sec-buzzer: cyber security emerging topic mining with open threat intelligence retrieval and timeline event annotation. *Soft Computing*, 2017.

[86] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of massive datasets*. Cambridge University Press, 2014.

[87] Xiaojing Liao, Kan Yuan, XiaoFeng Wang, Zhou Li, Luyi Xing, and Raheem Beyah. Acing the ioc game: Toward automatic discovery and analysis of open-source cyber threat intelligence. In *2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016.

[88] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*, 2019.

[89] Xiuwen Liu et al. Event evolution model for cybersecurity event mining in tweet streams. *Information Sciences*, 2020.

[90] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. of the 5th BSMSP*, 1967.

[91] Juan Martinez-Romo and Lourdes Araujo. Detecting malicious tweets in trending topics using a statistical analysis of language. *Expert Systems with Applications*, 40(8), 2013.

[92] Navjot Marwaha. System and method for providing common event format using alert index, November 21 2006. US Patent 7,139,938.

[93] Nikki McNeil, Robert A Bridges, Michael D Iannacone, Bogdan Czejdo, Nicolas Perez, and John R Goodall. Pace: Pattern accurate computationally efficient bootstrapping for timely discovery of cyber-security concepts. In *12th ICMLA*, 2013.

[94] Miles A McQueen, Trevor A McQueen, Wayne F Boyer, and May R Chaffin. Empirical estimates and observations of 0Day vulnerabilities. In *42nd HICSS*, 2009.

[95] David R Miller, Shon Harris, Allen Harper, Stephen VanDyke, and Chris Blask. *Security Information and Event Management (SIEM) Implementation (Network Pro Library)*. McGraw Hill, 2010.

[96] Sudip Mittal, Prajit Kumar Das, Varish Mulwad, Anupam Joshi, and Tim Finin. Cybertwitter: Using Twitter to generate alerts for cybersecurity threats and vulnerabilities. In *2016 ASONAM*, 2016.

[97] Behrang Mohit. Named entity recognition. In *Natural language processing of semitic languages*. Springer, 2014.

[98] Ole Christian Moholth, Radmila Juric, and Karoline Moholth McClenaghan. Detecting cyber security vulnerabilities through reactive programming. In *HICSS 2019*, 2019.

[99] Kartik Nayak, Daniel Marino, Petros Efstathopoulos, and Tudor Dumitraş. Some vulnerabilities are different than others. In *17th RAID*, 2014.

[100] Amirreza Niakanlahiji et al. Iocminer: Automatic extraction of indicators of compromise from twitter. In *Big Data*, 2019.

[101] E. Nunes, A. Diab, A. Gunn, E. Marin, V. Mishra, V. Paliath, J. Robertson, J. Shakarian, A. Thart, and P. Shakarian. Darknet and deepnet mining for proactive cybersecurity threat intelligence. In *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, 2016.

[102] Ahmet Okutan et al. Predicting cyber attacks with bayesian networks using unconventional signals. In *Proceedings of the 12th CISRC*, 2017.

[103] Saša Petrović, Miles Osborne, and Victor Lavrenko. Streaming first story detection with application to Twitter. In *11th NAACL HLT*, 2010.

[104] Robi Polikar, Lalita Upda, Satish S Upda, and Vasant Honavar. Learn++: An incremental learning algorithm for supervised neural networks. *IEEE transactions on systems, man, and cybernetics, part C (applications and reviews)*, 31(4), 2001.

[105] Alexander Reinthal, Eleftherios Lef Filippakis, and Magnus Almgren. Data modelling for predicting exploits. In *NordSec*, 2018.

[106] Alan Ritter, Evan Wright, William Casey, and Tom Mitchell. Weakly supervised extraction of computer security events from Twitter. In *24th WWW*, 2015.

[107] Luis Gustavo Araujo Rodriguez, Julia Selvatici Trazzi, Victor Fossaluza, Rodrigo Campiolo, and Daniel Macêdo Batista. Analysis of vulnerability disclosure delays from the national vulnerability database. In *WSCDC-SBRC 2018*, 2018.

[108] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6), 1958.

[109] Yaman Roumani, Joseph K Nwankpa, and Yazan F Roumani. Time series modeling of vulnerabilities. *Computers & Security*, 51, 2015.

[110] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, DTIC Document, 1985.

[111] Carl Sabottke, Octavian Suciu, and Tudor Dumitraș. Vulnerability disclosure in the age of social media: exploiting Twitter for predicting real-world exploits. In *24th USENIX Security Symposium*, 2015.

[112] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes Twitter users: Real-time event detection by social sensors. In *19th WWW*, 2010.

[113] Fatemeh Saki and Nasser Kehtarnavaz. Online frame-based clustering with unknown number of clusters. *Pattern Recognition*, 57:70–83, 2016.

[114] Anna Sapienza, Alessandro Bessi, Saranya Damodaran, Paulo Shakarian, Kristina Lerman, and Emilio Ferrara. Early warnings of cyber threats in online discussions. In *2017 ICDMW*, 2017.

[115] Anna Sapienza, Sindhu Kiranmai Ernala, Alessandro Bessi, Kristina Lerman, and Emilio Ferrara. Discover: Mining online chatter for emerging cyber threats. In *WWW'18 Companion*, 2018.

[116] Clemens Sauerwein et al. Threat intelligence sharing platforms: An exploratory study of software vendors and research perspectives. In *Towards Thought Leadership in Digital Transformation: 13. Internationale Tagung Wirtschaftsinformatik*, 2017.

[117] Clemens Sauerwein, Christian Sillaber, Michael M Huber, Andrea Mussmann, and Ruth Breu. The tweet advantage: An empirical analysis of 0-day vulnerability information shared on Twitter. In *33rd IFIP SEC*, 2018.

[118] Nikolaos Serketzis et al. Actionable threat intelligence for digital forensics readiness. *Information & Computer Security*, 2019.

[119] Muhammad Shahzad, Muhammad Zubair Shafiq, and Alex X Liu. A large scale exploratory analysis of software vulnerability life cycles. In *34th ICSE*, 2012.

[120] Han-Sub Shin, Hyuk-Yoon Kwon, and Seung-Jin Ryu. A new text classification model based on contrastive word embedding for detecting cybersecurity intelligence in twitter. *Electronics*, 2020.

[121] Hyejin Shin, WooChul Shim, Jiin Moon, Jae Woo Seo, Sol Lee, and Yong Ho Hwang. Cybersecurity event detection with new and re-emerging words. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, 2020.

[122] Lidan Shou et al. Sumblr: continuous summarization of evolving tweet streams. In *Proceedings of the 36th ACM SIGIR*, 2013.

[123] Jonathan A Silva et al. Data stream clustering: A survey. *ACM Computing Surveys*, 46(1), 2013.

[124] Saini Jacob Soman and S Murugappan. Detecting malicious tweets in trending topics using clustering and classification. In *2014 ITC*, 2014.

[125] Robert David Steele. Open source intelligence: What is it? why is it important to the military. *American Intelligence Journal*, 17(1), 1996.

[126] Nadeem Ahmed Syed, Syed Huan, Liu Kah, and Kay Sung. Incremental learning with support vector machines. 1999.

[127] Romilla Syed, Maryam Rahafrooz, and Jeffrey M Keisler. What it takes to get retweeted: An analysis of software vulnerability messages. *Computers in Human Behavior*, 80, 2018.

[128] Robert Tibshirani et al. Estimating the number of clusters in a data set via the gap statistic. *J. Royal Stat. Soc., Series B*, 63(2), 2001.

[129] Alberto Tonon et al. Armatweet: detecting events by semantic tweet analysis. In *Proceedings of the European Semantic Web Conference*, 2017.

[130] Slim Trabelsi, Henrik Plate, Amine Abida, M Marouane Ben Aoun, Anis Zouaoui, et al. Mining social networks for software vulnerabilities monitoring. In *7th NTMS*, 2015.

[131] Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. Feature hashing for large scale multitask learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009.

[132] Wei Xie, Feida Zhu, Jing Jiang, Ee-Peng Lim, and Ke Wang. Topicsketch: Real-time bursty topic detection from Twitter. *IEEE TKDE*, 28(8), 2016.

[133] Semih Yagcioglu et al. Detecting cybersecurity events from noisy short text, 2019.

[134] Xiaohui Yan, Jiafeng Guo, Yanyan Lan, Jun Xu, and Xueqi Cheng. A probabilistic model for bursty topic discovery in microblogs. In *29th AAAI*, 2015.

[135] Mohammed J Zaki et al. *Data mining and analysis: fundamental concepts and algorithms*. Cambridge University Press, 2014.

[136] Tian Zhang et al. Birch: an efficient data clustering method for very large databases. In *ACM Sigmod Record*, 1996.

[137] Guanglou Zheng, Guanghe Zhang, Wencheng Yang, Craig Valli, Rajan Shankaran, and Mehmet A Orgun. From wannacry to wannadie: Security trade-offs and design for implantable medical devices. In *2017 17th International Symposium on Communications and Information Technologies (ISCIT)*, 2017.

[138] Aoying Zhou et al. Tracking clusters in evolving data streams over sliding windows. *Knowledge and Information Systems*, 15(2), 2008.

[139] Ziyun Zhu and Tudor Dumitras. Featuresmith: Automatically engineering features for malware detection by mining the security literature. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016.

A

Complete Cluster Data

Table A.1 presents the 65 IoCs largest clusters generated by SYNAPSE, as described in Section 5.5.1.

By running SYNAPSE's IoC generation module, each cluster was tagged with the type of threats mentioned by its tweets. The most common tags are "vulnerability" (23) and "vulnerabilities" (13), reflecting that most threats are related to vulnerability disclosure. Other two common tags are "exploit" (18) and "0day" (15) (or "zero-day"), which indicate exploitable vulnerabilities. Less used tags include "remote" (6) (remote execution attacks), "denial of service" (6), "SQL injection" (5), and "Buffer overflow" (4) (or BO).

Out of the 13 assets composing the hypothetical IT infrastructure described in Table 4.1, only 9 ($\sim$ 70%) had related IoCs. The distribution of IoCs over the assets is shown in Figure A.1. WordPress is the asset with more related IoCs (19), followed by Linux (14) and Cisco (12). All analysed IoCs mentioned a single asset.

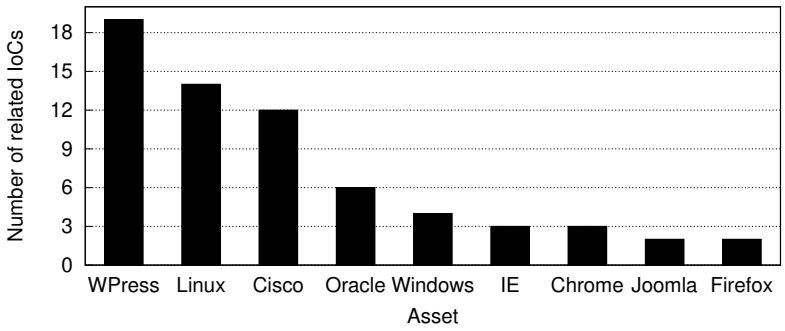


Figure A.1: Number of IoCs for each asset.

A. COMPLETE CLUSTER DATA

Table A.1: Largest generated clusters represented as IoCs.

Cluster exemplar text (without links)	**#**	**Asset**	**Date**	**Action**	**Threat type**	**Notes**
High - USN-3016-1 - Linux kernel vulnerabilities A security issue affects these releases of Ubuntu and its derivat	12	Linux	27/06	Patch	vulnerabilities	Several vulnerabilities patched, some were not yet included on NVD, half with CVSS ≥ 7.5
#0daytoday #Sun Secure Global Desktop and Oracle Global Desktop 4.61.915 - ShellShock Exploit [#0day #Exploit]	11	Oracle	06/06	None	exploit, 0day	An exploit is presented; an expert might use this data for protection
#ubuntu #security : USN-2993-1: Firefox vulnerabilities	10	Firefox	09/06 (4)	Patch	vulnerabilities	Patches are available for vulnerabilities, half with CVSS ≥ 8.8
Bugtraq: CM Ad Changer 1.7.7 Wordpress Plugin - Cross Site Scripting Web Vulnerability	10	WPress	13/06	Patch	vulnerability	A patch is available; an exploit is provided
Bugtraq: Wordpress Levo-Slideshow 2.3 - Arbitrary File Upload Vulnerability	9	WPress	07/06	Config	vulnerability	An exploit is provided; a software correction is suggested
Bugtraq: Oracle Orakill.exe Buffer Overflow	9	Oracle	14/06	Patch	Buffer overflow	A patch is available; an exploit is provided
#CISCO fixed severe #vulnerabilities in Network Management and #Security Products #SecurityAffairs	9	Cisco	30/06 (2)	Patch	vulnerabilities	Patch for critical vulnerabilities (CVSS ≥ 8.6) announced on Twitter before being published on NVD
#ubuntu #security : USN-3016-1: Linux kernel vulnerabilities	8	Linux	27/06	Patch	vulnerabilities	Several vulnerabilities patched, some were not yet included on NVD, half of the vulns with CVSS ≥ 7.5
Microsoft Internet Explorer CVE-2016-3205 Scripting Engine Remote Memory Corruption Vulnerability Type: Vulnerabil	8	IE	14/06 (1)	Config	vulnerability, remote	This cluster contains various threats with CVSS ≥ 7.5; configurations are suggested to solve the issue before it is patched
NA - CVE-2016-2825 - Mozilla Firefox before 47.0 allows remote... Mozilla Firefox before 47.0 allows remote attack	8	Firefox	13/06	Patch	attack, remote	A patch is available for a vulnerability with CVSS 6.5

Vuln: Oracle Java SE and JRockit CVE-2016-3427 Remote Security Vulnerability Vulnerable:Red Hat Enterprise Linux	21	Oracle	05/07	Patch	vulnerability, remote	This cluster contains three different threats (one with CVSS 9.0); patches are available
#ubuntu #security : USN-3006-1: Linux kernel vulnerabilities	19	Linux	10/06	Patch	vulnerabilities	Several vulnerabilities patched, some were not yet included on NVD, half with CVSS ≥ 7.0
#0daytoday #Cisco EPC 3928 - Multiple Vulnerabilities [webapps #exploits #Vulnerabilities #0day #Exploit]	16	Cisco	07/06	None	exploit, vulnerabilities, 0day	An exploit is presented; an expert might use this data for protection (half of the vulns with CVSS ≥ 7.5)
#0daytoday #Joomla En Masse com_enmasse Component 5.1 - 6.4 - SQL Injection Vulnerability [#0day #Exploit]	12	Joomla	15/06	None	SQL injection, exploit, injection, vulnerability, 0day	An exploit is presented; an expert might use this data for protection
#0daytoday #WordPress Social Stream Plugin 1.5.15 - wp_options Overwrite Vulnerability [#0day #Exploit]	8	WPress	14/06	Patch	exploit, vulnerability, 0day	A patch is available; an exploit is provided
Microsoft Internet Explorer 11 Garbage Collector Attribute Type Confusion #exploit	8	IE	18/06	Patch	exploit	A patch is available for a vulnerability with CVSS 8.8; an exploit is provided
CVE-2016-1388 Cisco Prime Network Analysis Module (NAM) before 6.1(1) patch.6.1-2-final and 6.2.x before 6.2(1) an	8	Cisco	03/06	Patch		This cluster contains 4 threats, 3 with CVSS $\geq$ 7.8; patches are available
#Oracle #Linux 6 : #openssl (ELSA-2016-0996) #Nessus	8	Linux	16/05	Patch		This cluster contains seven threats: 3 critical (CVSS 9.8) and 3 high (CVSS 7.5); patches are available
Vuln: Linux Kernel Multiple Local Memory Corruption Vulnerabilities	7	Linux	08/07	Patch	vulnerabilities	Patches are available for vulnerabilities with CVSS 7.1 and 7.8
Vuln: Linux Kernel CVE-2016-0723 Local Race Condition Vulnerability	7	Linux	08/07	Patch	vulnerability	A patch is available for vulnerability with CVSS 6.8
Vuln: Linux kernel CVE-2013-7446 Use After Free Denial of Service Vulnerability	7	Linux	05/07	Patch	denial of service, vulnerability	A patch is available for vulnerability with CVSS 5.3

A. COMPLETE CLUSTER DATA

Bugtraq: Cisco Security Advisory: Cisco Firepower System Software Static Credential Vulnerability	7	Cisco	29/06 (3)	Patch	vulnerability	A patch is available for vulnerability with CVSS 8.6
#0daytoday #WordPress Ultimate Membership Pro Plugin 3.3 - SQL Injection Vulnerability [#0day #Exploit]	7	WPress	29/06	Patch	SQL injection, exploit, injection, vulnerability, 0day	A patch is available; an exploit is provided
#0daytoday #Google Chrome - GPU Process MailboxManagerImpl Double-Read Vulnerability [#0day #Exploit]	7	Chrome	15/06	Patch	exploit, vulnerability, 0day	A patch is available; an exploit is provided
#0daytoday #WordPress Gravity Forms Plugin 1.8.19 - Arbitrary File Upload Exploit [#0day #Exploit]	7	WPress	17/06	None	exploit, 0day	An exploit is presented; an expert might use this data for protection
#0daytoday #WordPress Uncode Theme 1.3.1 - Arbitrary File Upload Exploit [webapps #exploits #0day #Exploit]	7	WPress	06/06	N/A	exploit, 0day	All tweet links are broken; nothing can be inferred
#0daytoday #WordPress Double Opt-In for Download Plugin 2.0.9 - SQL Injection Vulnerability [#0day #Exploit]	7	WPress	06/06	Patch	SQL injection, exploit, injection, vulnerability, 0day	A patch is available; an exploit is provided
#cybersecurity Hackers offering Microsoft Windows zero-day exploit for $90000 #infosec	7	Windows	01/06	N/A	exploit	Just informative tweets
#Oracle ATS Arbitrary File Upload #PacketStorm	7	Oracle	24/05	None		An exploit is presented; an expert might use this data for protection
Vuln: Linux Kernel 'usb/core/hub.c' NULL Pointer Dereference Denial of Service Vulnerability	6	Linux	08/07	Patch	denial of service, vulnerability	A patch is available for vulnerability with CVSS 6.8
#0daytoday #Linux - ecryptfs and /proc/$pid/environ Privilege Escalation Vulnerability [#0day #Exploit]	6	Linux	21/06 (6)	None	exploit, escalation, vulnerability, 0day	An exploit is early presented for a vulnerability with CVSS 7.8; an expert might use this data for protection
CVE-2016-3221 The kernel-mode drivers in Microsoft Windows Vista SP2, Windows Server 2008 SP2 and R2 SP1, Windows	6	Windows	16/06	Patch		A patch is available for a vulnerability with CVSS 7.8
NA - CVE-2016-3201 - Microsoft Windows 8.1, Windows Server 2012 Gold... Microsoft Windows 8.1, Windows Server 2012	6	Windows	16/06	Patch		A patch is available for a vulnerability with CVSS 6.5

#0daytoday #Joomla com_affiliatetracker - SQL Injection Vulnerability [webapps #exploits #Vulnerability #0day	6	Joomla	13/06	N/A	SQL injection, exploit, injection, vulnerability, 0day	All tweet links are broken; nothing can be inferred
[shellcode] - #Linux x86_64 Shellcode Null-Free Reverse TCP Shell #ExploitDB	6	Linux	16/06	None	exploit	An exploit is presented; an expert might use this data for protection
Bugtraq: [security bulletin] - Linux Kernel Flaw, ASN.1 DER decoder for x509 certificate DER	6	Linux	06/06 (21)	Patch	certificate	A highly important Linux kernel flaw (CVSS 7.8) was disclosed 21 days before being included in NVD
[webapps] - WordPress WP Mobile Detector Plugin 3.5 - Arbitrary File Upload: WordPress WP Mobile Detector Plu...	6	WPress	06/06	Patch		A patch is available; an exploit is provided
Bugtraq: Cisco Security Advisory: Cisco Prime Network Analysis Module IPv6 Denial of Service Vulnerability	6	Cisco	01/06 (1)	Patch	denial of service, vulnerability	A patch is available for a vulnerability with CVSS 5.3
Bugtraq: Cisco Security Advisory: Cisco Prime Network Analysis Module Unauthenticated Remote Code #bugtraq	6	Cisco	01/06 (1)	Patch	remote	A patch is available for a critical vulnerability with CVSS 9.8
WordPress Patches Zero Day in WP Mobile Detector Plugin #InfoSec	6	WPress	03/06	Patch	zero day	A patch is available
CVE-2016-1381 Memory leak in Cisco AsyncOS 8.5 through 9.0 before 9.0.1-162 on Web Security Appliance (WSA) device	6	Cisco	25/05	Patch	leak	A patch is available for a vulnerability with CVSS 7.5
Oracle E-Business Suite Vulnerabilities Related To Common Components Oracle E-Business Intelligence component in O	6	Oracle	23/05	None	vulnerabilities	The tweet links provide no useful information
NA - cisco-sa-20160518-wsa4 - Cisco Web Security Appliance Connection Denial of Service Vulnerability A vulnerabil	6	Cisco	18/05 (6)	Patch	denial of service, vulnerability	A high impact vulnerability (CVSS 7.5) was disclosed and patched before its inclusion on NVD
#ubuntu #security : USN-2947-1: Linux kernel vulnerabilities	6	Linux	06/04	Patch	vulnerabilities	A patch is available to solve multiple vulnerabilities, one of them critical (CVSS 9.8)
Vuln: Cisco Video Communication Server and Expressway CVE-2016-1444 Authentication Bypass Vulnerability	5	Cisco	08/07	Patch	vulnerability	A patch is available for a vulnerability with CVSS 6.5

A. COMPLETE CLUSTER DATA

Vuln: Google Chrome Prior to 49.0.2623.75 Multiple Security Vulnerabilities	5	Chrome	06/07	Patch	vulnerabilities	A patch is available to solve multiple high to critical vulnerabilities (5 with CVSS 8.8 and 5 with CVSS 9.8)
[webapps] - WordPress Real3D FlipBook Plugin - Multiple Vulnerabilities: WordPress Real3D FlipBook Plugin - M...	5	WPress	04/07	None	vulnerabilities	An exploit is presented; an expert might use this data for protection
Vuln: Linux Kernel 'btrfs/inode.c' Information Disclosure Vulnerability	5	Linux	05/07	Patch	vulnerability	A patch is available for a vulnerability with CVSS 4.0
Medium - CVE-2016-5835 - WordPress before 4.5.3 allows remote attackers... WordPress before 4.5.3 allows remote at	5	WPress	29/06	Patch	attack, remote	A patch is available for a vulnerability with CVSS 7.5
#vulnerability #security : WordPress Contus Video Comments 1.0 File Upload	5	WPress	22/06	None	vulnerability	An exploit is presented; an expert might use this data for protection
[webapps] - WordPress Ultimate Product Catalog Plugin 3.8.1 - Privilege Escalation: WordPress Ultimate Produc...	5	WPress	20/06	Patch	escalation	A patch is available; an exploit is provided
#0daytoday #WordPress Premium SEO Pack 1.9.1.3 - wp_options Overwrite Exploit [webapps #exploits #0day #Exploit]	5	WPress	21/06	None	exploit, 0day	An exploit is presented; an expert might use this data for protection
CVE-2016-0200 Microsoft Internet Explorer 9 through 11 allows remote attackers to execute arbitrary code or cause	5	IE	16/06	Patch	attack, remote	The cluster contains two different threats; patches are available to solve 4 vulns with CVSS 8.8
Bugtraq: Cisco Security Advisory: Cisco RV110W, RV130W, and RV215W Routers Arbitrary Code Execution Vulnerability	5	Cisco	15/06 (3)	Patch	vulnerability, execution	A critical vulnerability (CVSS 9.8) was disclosed and patched before its inclusion on NVD
#0daytoday #WordPress Newspaper Theme 6.7.1 - Privilege Escalation Exploit [webapps #exploits #0day #Exploit]	5	WPress	06/06	Patch	exploit, escalation, 0day	A patch is available; an exploit is provided
[webapps] - WordPress Simple Backup Plugin 2.7.11 - Multiple Vulnerabilities: WordPress Simple Backup Plugin ...	5	WPress	06/06	None	vulnerabilities	An exploit is presented; an expert might use this data for protection

CVE-2016-1701 The Autofill implementation in Google Chrome before 51.0.2704.79 mishandles the interaction between	5	Chrome	06/06	Patch		All tweets refer a different vulnerability, all from the same date, all with CVSS $\geq$ 7.5; patches are available
#0daytoday #WordPress WP PRO Advertising System Plugin 4.6.18 - SQL Injection Exploit [#0day #Exploit]	5	WPress	06/06	None	SQL injection, exploit, injection, 0day	An exploit is presented; an expert might use this data for protection
[webapps] - WordPress Creative Multi-Purpose Theme 9.1.3 - Stored XSS: WordPress Creative Multi-Purpose Theme...	5	WPress	06/06	Patch	XSS	A patch is available; an exploit is provided
#WordPress WP Mobile Detector 3.5 Shell Upload #PacketStorm	5	WPress	04/06	Patch		A patch is available; an exploit is provided
#hackers Selling Unpatched Microsoft Windows Zero-Day Exploit for $90.000	5	Windows	03/06	N/A	exploit	Just informative tweets
Oracle E-Business Suite Vulnerabilities Related To E-Business Intelligence Oracle E-Business Intelligence compon	5	Oracle	30/05	None	vulnerabilities	The tweet links provide no useful information
Bugtraq: Cisco Security Advisory: Cisco Products IPv6 Neighbor Discovery Crafted Packet Denial of Service	5	Cisco	25/05 (4)	Patch	denial of service	A high impact vulnerability (CVSS 7.5) was disclosed and patched before its inclusion on NVD
#ubuntu #security : USN-2975-2: Linux kernel (Trusty HWE) vulnerability	5	Linux	16/05 (42)	Patch	vulnerability	A high impact vulnerability (CVSS 7.8) was disclosed and patched before its inclusion on NVD (42 days in advance)
Bugtraq: Cisco Security Advisory: Cisco Web Security Appliance HTTP POST Denial of Service Vulnerability	5	Cisco	18/05 (6)	Patch	vulnerability	A high impact vulnerability (CVSS 7.5) was disclosed and patched before its inclusion on NVD