Systematic Mapping and Temporal Reasoning of IoT Cyber Risks using Structured Data

Marta Zumaquero Gil
University of New South Wales
Sydney, NSW, Australia

Zhibo Hu
University of New South Wales
Sydney, NSW, Australia

Minzhao Lyu
University of New South Wales
Sydney, NSW, Australia

Gustavo Batista
University of New South Wales
Sydney, NSW, Australia

Hassan Habibi Gharakheili
University of New South Wales
Sydney, NSW, Australia

ABSTRACT

Deploying Internet-of-Things (IoT) assets, such as cameras, printers, and building sensors, at scale introduces operational and cyber risks to organizations. While public repositories such as the National Vulnerability Database (NVD) or Exploit-DB provide valuable data on known cyber risks, each comes with its specific query format, and their knowledge is often fragmented, lacking a comprehensive perspective. Organizations often require the capability to assess current vulnerabilities and forecast future risks from distributed and nonunified sources. This paper aims to empower digital infrastructure teams to obtain a complete view of IoT cyber risks. First, we map public repositories for digital product vulnerabilities, exploits, and patches (solutions). This includes highlighting their interrelationships and the information they offer. Second, we develop a data schema to detail cyber risks associated with specific products, like equipment, operating systems, or applications. We build "vesData", a tool that takes a product name as input and automatically produces a machine-processable data structure of its risk knowledge. We apply our tool to obtain public risk data of about 20 consumer IoT products in our lab—our tool and data will be released openly. Third, we demonstrate a preliminary use case of our structured IoT risk data, which predicts new vulnerabilities, patches, and exploits for existing ones.

CCS CONCEPTS

• **Security and privacy → Network security**; • **Networks → Network monitoring**; • **Information systems → Data management systems**.

KEYWORDS

Cyber risk data, Internet-of-Things, cyber security

ACM Reference Format:
Marta Zumaquero Gil, Zhibo Hu, Minzhao Lyu, Gustavo Batista, and Hassan Habibi Gharakheili. 2024. Systematic Mapping and Temporal Reasoning of IoT Cyber Risks using Structured Data. In *ASIAN INTERNET ENGINEERING CONFERENCE 2024 (AINTEC '24), August 09, 2024, Sydney, NSW, Australia.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3674213.3674216

AINTEC '24, August 09, 2024, Sydney, NSW, Australia

ACM ISBN 979-8-4007-0985-2/24/08.
https://doi.org/10.1145/3674213.3674216

1 INTRODUCTION

The IoT market is large and diverse, with manufacturers operating at various maturity levels in terms of security, privacy, and reliability [17]. The disparity in security postures across the market has increased the scale and sophistication of cyber threats against IoT-rich networks and critical infrastructures [9]. Operators of IoT networks are, therefore, urged to develop or acquire a suite of capabilities for cyber asset attack surface management (CAASM) [11]. It is imperative for organizations to obtain real-time visibility into connected assets and their intended behavioral characteristics [18, 19, 23]. Once asset registers are established, a pressing need arises for a reliable assessment of vulnerabilities and systematic risk estimation and prediction. Understanding the extent of cyber risks associated with IoT devices enables network operators to effectively manage vulnerable devices on their networks, implementing control policies to reduce attack surfaces [13]. Additionally, the magnitude of cyber risk serves as an important metric for insurers in calculating cyber insurance premiums for organizations [40].

Currently, security firms and individual contributors curate and publish repositories [1, 24, 35], each containing specific types of cyber risk data like information on device vulnerabilities, past exploit instances, or solutions (patches) for certain vulnerabilities. Notable examples include the Common Vulnerabilities and Exposures (CVE) database by MITRE and the National Vulnerability Database (NVD) by NIST, which report vulnerabilities of networked devices along with their respective data coverage specifications. Relying solely on specific databases, such as CVE and NVD, may offer limited data coverage and types, leading to an inaccurate estimation of the actual cyber risks associated with a given IoT product. For a reliable risk assessment, it is essential to consider that some products (*e.g.,* air quality sensors or power switches) may have numerous vulnerabilities yet remain unattractive to attackers for various reasons (*e.g.,* limited footprints). However, a popular printer or camera with multiple exposures and exploits may receive support from the security community, frequently releasing solutions or patches. Accurate risk assessment and prediction require comprehensive cyber risk information obtained from diverse and heterogeneous data repositories. Furthermore, security teams prefer structured data formats for systematic consumption in analytical tasks.

Prior works [2, 10, 15, 16, 20, 26] developed tools to collect cyber risk data from public sources. However, they focused on certain risk

categories rather than aiming for a comprehensive collection of all data types (*i.e.,* vulnerability, exploit and solution) for intended IoT products. For example, CVEfixes [2] and CyVIA [20] concentrate on vulnerability data. CVEfixes automatically collects and curates all common vulnerability and exposure (CVE) records available on the NVD dataset, while CyVIA combines vulnerability data from multiple public repositories. Works in [15, 26] analyzed data from exploit-related databases. To our knowledge, no open-source tool or public repository provides a comprehensive set of cyber risk data from all three categories for given product identifiers (*e.g.,* names). To address the identified gap in this paper, we develop a tool named vesData, designed to automatically gather cyber risk records specific to a given IoT product from diverse public repositories and organize them into a detailed structured format. Our tool allows cybersecurity and digital infrastructure teams to gain comprehensive insights into vulnerabilities, exploits, and solutions associated with their deployed IoT products. This empowers them to quantify and predict the exposed cyber risk effectively. This paper makes three specific contributions.

Our **first** contribution (detailed in §3) involves a systematic review and mapping of 16 popular public cyber risk databases maintained by governmental entities or security organizations. Based on the type of cyber risk records about digital (IoT) products in each database, we categorize them as vulnerability, exploit, or solution repositories, respectively. Additionally, we highlight the connectivity among the repositories, illustrating instances such as records in NVD and CVE Details directly referencing the respective records in the CVE database.

Building upon the categorization of diverse public repositories, our **second** contribution (discussed in §4) introduces a structured data schema that comprehensively describes cyber risk information for a specific (IoT) product. Additionally, we present an automatic tool, vesData, designed to generate a data file instance for a given product name by iterating through selected repositories. By constructing data files for 23 IoT products (to be released publicly), we demonstrate the quality of data collected by our techniques and commendable timing performance.

To underscore the utility of our structured data schema for cyber risk measurement and prediction, the **third** contribution (discussed in §5) describes our preliminary effort in using temporal heterogeneous graph neural networks to predict the number of vulnerabilities, exploits, and solutions for three representative products: HP printer, Apache proxy, and Linux IoT kernel. Leveraging the structured data files of these three products, we demonstrate that the predicted cyber risk counts closely align with the ground-truth values. Furthermore, we emphasize the importance of incorporating all three categories of cyber risk data for reliable predictions.

2 RELATED WORK

Tools for Gathering Cyber Risk Data: Previous research efforts have introduced methods and open-source tools for collecting cyber risk data from public repositories, but their focus differs. For instance, *CVEfixes* [2] has developed a tool that retrieves the entire NVD database, augmenting each record with additional information (*e.g.,* commits, fixes) about vulnerable and patched codes available on platforms like GitHub, GitLab, and Bitbucket. This extended database aids researchers in categorizing software vulnerabilities based on attributes at an aggregate level. Similarly, *CyVIA* [20, 21] combines all records from the CVE and NVD databases to predict popular vulnerabilities (*e.g.,* unauthorized access and DoS) that may impact the global cyberspace in the future. In contrast, our vesData tool narrows its focus on device-specific risk data while encompassing three interconnected pillars: vulnerability, exploit, and solution. We contend that our data provides network operators and cybersecurity teams with a fine-grained and more comprehensive view of the attack surfaces exposed by their connected devices.

Inference Analysis of Cyber Risk Data: Previous studies primarily analyzed cyber risk data from individual public repositories or multiple repositories of a specific type, such as vulnerability, exploit, or solution. For example, works like [2, 3, 6, 12, 31, 33] focus on vulnerability data, utilizing the National Vulnerability Database (NVD) to assess current risk levels for specific products, predict future risks (at global/aggregate levels), and identify missing elements in records. In contrast, studies like [15, 26] examine exploit data from exploit-DB to gauge the popularity of various exploit types, particularly those induced by IoT malware. Additionally, works discussed in [10, 16, 20] concentrate on solution data for cyber risk. However, due to the limited coverage of cyber risk data in existing databases, which typically focus on only one of the three categories (vulnerability, exploit, or solution), prior analyses have not fully considered the semantic correlations between product-specific vulnerabilities, exploits, and solutions, especially in the temporal domain. This limitation becomes pronounced in §5.

3 MAP OF PUBLIC KNOWLEDGE REPOSITORIES ON CYBER RISKS

To establish a unified knowledge base for a given product, we identify 16 popular repositories of cyber risks and categorize them into three data domains: **vulnerability** of devices (§3.1), **exploit** events of a certain device type by cyber attackers (§3.2), or **solution** and countermeasure (§3.3) of exploit events. Fig. 1 visually represents this categorization. Solid lines connecting repositories signify shared or overlapped knowledge, while dashed lines indicate linked items (such as pointers or IDs).

3.1 Vulnerability

As depicted in the brown frame on top of Fig. 1, our first category of cyber risk knowledge is vulnerability, detailing the potential weaknesses a product (device) may expose to attackers. A total of 12 major repositories considered in this paper contain vulnerability information. Out of these, five repositories solely maintain vulnerability information and are exclusively framed within the brown region. NVD, which primarily carries vulnerability data and includes solution information, is framed in the brown and green regions but is discussed in this section. We now discuss the features of these six repositories. It is worth noting that the three red-titled and three green-titled repositories, which primarily maintain exploit and solution information, also host vulnerability data. These will be discussed in the following sections §3.2 and §3.3, respectively.

CVE: Visually situated as the center top box in the brown frame, CVE (Common Vulnerabilities and Exposures) [35] is a repository that catalogs computer security flaws, overseen by MITRE. MITRE updates the CVE repository with information gathered from various

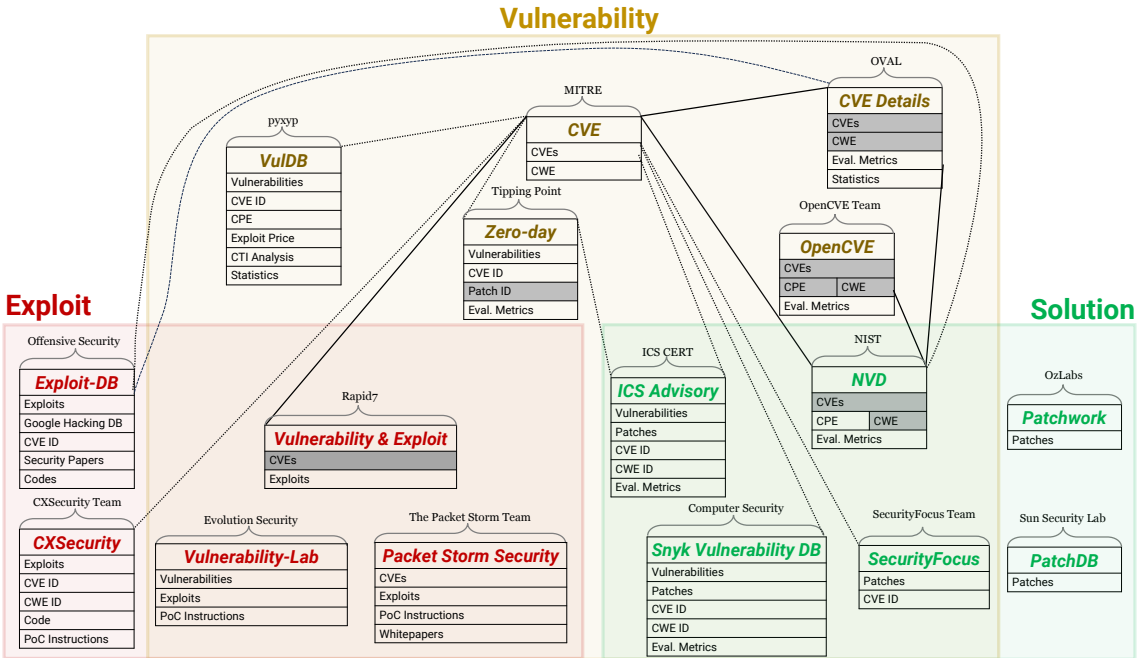


Figure 1: Map of public repositories on product cyber risks across vulnerability, exploit, and solution pillars.

sources, including Bug Bounty Programs, Hosted Services, National and Industry CERTs, vendors of a specific vulnerable product, and research groups working on vulnerability projects. Each record of vulnerability information is registered and tracked in the repository with an identifier in the format "`CVE-{year}-{ID}`". The ID is assigned by one of the 319 authorized organizations (by MITRE), known as CVE Numbering Authorities (CNA), such as IT/security vendors, manufacturers, cloud operators, open-source projects, and governmental bodies.

In addition to the identifier, a CVE record includes the date, a concise description of the security vulnerability, and references (links) to source reports and advisories. As of the time of writing this paper, the CVE repository hosts a total of 222,759 CVE records, with instances of records being occasionally merged and/or pruned over time. Within the CVE repository, MITRE also curates the Common Weakness Enumeration (CWE), a community-developed list of software and hardware weakness types. Ideally, each CVE record should be linked to a CWE type; however, such direct linkage is currently unavailable in the repository.

NVD: The National Vulnerability Database (NVD) serves as a U.S. government repository for standards-based vulnerability management data and is sponsored by the National Institute of Standards and Technology (NIST). The database is regularly updated by importing and accumulating newly added records from the CVE repository, encompassing 236,729 CVE records and corresponding CWE types. Furthermore, the NVD repository assigns evaluation scores to each CVE record using metrics provided in two schemes: Common Platform Enumeration (CPE), a structured naming scheme for information technology systems, software, and packages, and CVSS, a systematic scheme for assessing the severity of security vulnerabilities in software.

Other Repositories: In addition to the two primary vulnerability repositories, namely CVE and NVD, maintained by leading security organizations MITRE and NIST, other repositories built upon them, incorporating supplementary vulnerability metrics. For instance, CVE Details [4] by OVAL includes more than 164,000 CVE records, CWE types, and evaluation metrics from the NVD repository while augmenting them with additional statistics such as device vendors, firmware versions, and the presence of known exploits for each CVE record. CVE Details maintains the exploit information as a reference to Exploit-DB [7], as visually depicted by the dashed line connecting the two repositories in Fig. 1. Another example is OpenCVE [25], a repository built on top of the NVD database. It incorporates a distinct set of metrics totaling 236,691, including quantitative and qualitative risk assessment scores for vulnerabilities and exploits and the status of risk records (such as published, rejected, disputed, or reserved). In our vesData tool discussed later in §4.2, these metrics are attributed as scores in their respective categories.

Some repositories provide links to entries in the CVE or NVD repositories by storing their respective IDs rather than full records. An example is VulDB [36], positioned on the left side of the brown region in Fig. 1. VulDB is an official numbering authority certified by MITRE and a data publisher authorized by NIST. Therefore, besides the referenced CVE IDs, it maintains its own records, documenting device vulnerabilities reported since 1970. VulDB also hosts Cyber Threat Intelligence (CTI) analysis, the monetary cost of exploits, and vulnerability statistics for various device types. This macro-level information is particularly valuable for large-scale economic analysis. Similarly, the Zero-day repository by Tipping Point contains referencing IDs of CVE entries and unique vulnerabilities discovered by researchers of the Tipping Point Zero Day initiative since 2006. Those repositories are often small in size and host less structured data, such as text-based reports and source code)—thus, they may not be useful for IT, network, or cybersecurity teams.

Understanding the diverse array of public repositories, interconnected directly or indirectly, each specializing in specific facets of device vulnerabilities, empowers network operations and security

teams to gain richer insights to assess the vulnerability landscape of their deployed IoT assets and infrastructures. Recognizing the challenges inherent in collecting data from distributed sources, we address this practical concern in §4.2 by an automated tool.

3.2 Exploit

An exploit is a piece of code designed to take advantage of a specific vulnerability in a device, system, or application. When successfully executed, an exploit can lead to severe consequences, including unauthorized access, exfiltration of critical information, or denial of service. The security community maintains repositories to record known exploits along with sample executable code or processes. These repositories serve as valuable resources for manufacturers, security researchers, or cybersecurity teams, allowing them to test their deployed devices/systems against known vulnerabilities and enhance their defenses.

Purely Exploit Repositories: One of the primary exploit repositories is the Exploit Database (ExploitDB) [7], actively maintained by Offensive Security, the same organization behind the widely used penetration testing tool, Metasploit. Positioned in the left red region of Fig. 1, this repository aggregates exploit instances from diverse sources, including white papers, research articles, shell codes, and submissions to technical forums, totaling over 45,000 entries in the exploit list. Notably, it monitors updates from the Google Hacking database, specifically designed for exploits using Google's search engine syntax language. The repository also includes CVE IDs for entries from the CVE repository. Due to its comprehensive coverage and regular updates, many prior works ([15, 26]) rely on Exploit-DB as a primary data source. Another noteworthy exploit database is CXSecurity [5], a community-driven platform recognized for its rich collection of exploit code and Proof-of-Concept (PoC) instructions. Both Exploit-DB and CXSecurity exclusively contain full records for exploit instances rather than just IDs for vulnerability CVE entries found in other databases. Note that CXSecurity relies on user contributions for vulnerabilities. As a result, its trustworthiness and comprehensiveness may not match that of Exploit-DB.

Exploit Repositories with Vulnerability Records: In certain repositories primarily focused on exploit records, a smaller yet noteworthy fraction of vulnerability data exists. Examples include Packet Storm Security [27], Vulnerability-Lab [38], and Vulnerability and Exploit [37]. These repositories are visualized in the intersection region of brown and red frames and offer a unique mix of exploit and vulnerability records. The Vulnerability & Exploit repository, for instance, not only includes full records from the CVE database but also houses a rich collection of exploit instances. The other two curate their exclusive set of vulnerabilities, introducing new instances of vulnerability and exploit that may not be available in other databases.

3.3 Solution

The final category within cyber risk knowledge repositories is designated as "solution", as depicted in the green frame in Fig. 1. This category encompasses the strategies and measures to be employed by cybersecurity and/or asset management teams to address specific vulnerabilities that may lead to exploits. Examples of such solutions include the implementation of security patches (*i.e.,* blocks of code added to existing software), updating device firmware, enforcing access controls, and integrating monitoring techniques, all aimed at mitigating identified vulnerabilities.

The repository we want to highlight in this category is NVD[24], maintained by NIST. Beyond storing CVE and CWE vulnerability records, the NVD database includes crucial solution information for each CVE record. This encompasses a comprehensive list of external resources or references, patch submissions, curation details, and integration processes related to the identified vulnerability. In addition to the CVE/CWE records and their corresponding solutions, NVD incorporates CPE and evaluation metrics inherited from databases such as CVE Details and OpenCVE.

Three databases, namely ICS Advisory [14] by ICS CERT, Snyk Vulnerability DB [34] by Computer Security, and SecurityFocus [32], predominantly focus on solution information. They provide patches with referencing IDs linked to CVE and/or CWE records found in other vulnerability databases, as illustrated in Fig. 1. Consequently, these databases are designated with green titles and positioned within the region encircled by both the vulnerability (brown) and solution (green) frames.

The remaining two databases, namely PatchDB [28] and Patchwork [29], are exclusively framed within the green region in Fig. 1, indicating their sole focus on hosting patch information. PatchDB, described in detail in [39], boasts around 36,000 patches derived from real-world deployments. On the other hand, the Patchwork repository contains patch data meticulously maintained by Sun Security Lab.

3.4 Our Selected Databases

From the 16 databases discussed earlier in this section, we strategically choose one representative repository for each of the three domains: CVE for vulnerabilities, Exploit-DB for exploits, and NVD for solutions. These repositories stand out as the most popular, well-curated, regularly updated, and extensively referred to by both industry and academia. As elaborated in the next section, their inclusion is integral to developing our tool, vesData.

4 INTEGRATING AND GATHERING OF IOT CYBER RISK DATA

Building upon insights gleaned from public databases categorized into vulnerabilities, exploits, and solutions, we present our schema named vesData (§4.1). This schema integrates data fields sourced from diverse sources into a structured, machine-readable file. We, next, introduce our tool designed to accept a product name as input, traverse selected public repositories to extract relevant data records, and produce a structured JSON-formatted file as output (§4.2). To validate the efficacy of our tool, we test it with 23 IoT products, measuring its response time performance (§4.3).

4.1 Structured Cyber Risk Data

We envision a minimal structure of cyber risk data comprising four sections: Product, Vulnerabilities, Exploits and Solutions, as depicted in Fig. 2. Individual sections are discussed as follows.

The "`ves-product`" section (① in Fig. 2) provides high-level information about the product identity and risks, featuring essential

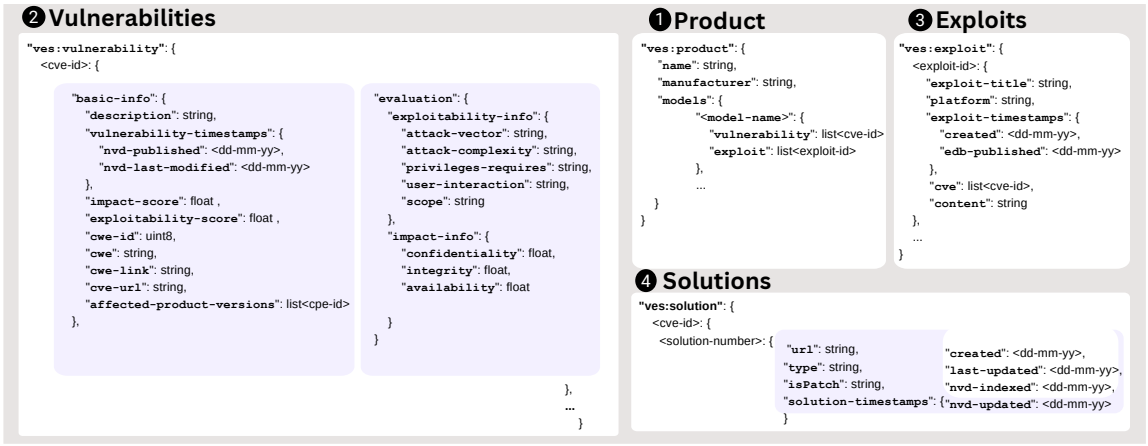


Figure 2: The structure of cyber risk data generated by our vesData tool.

fields such as "`name`" (*e.g.,* HP printer) and "`manufacturer`" (*e.g.,* HP). It is important to note that a product may exist in multiple "`models`" (*e.g.,* F2A70A, B5L26A), each associated with a specific set of "`vulnerability`" and "`exploit`" IDs. Individual IDs are expanded within their respective sections—note that each solution is mapped to and indexed by a unique vulnerability (CVE) ID, discussed later in this section.

The section ② of our data structure ("`ves-vulnerability`") hosts detailed information for each CVE ID (mentioned above) under two subblocks: "`basic-info`" and "`evaluation`". The basic-info subblock stores details, such as vulnerability description, published and modified timestamps, impact and exploitability scores, corresponding CWE and CVE information, and affected product firmware versions (different from product names earlier recorded by the "ves-product" component). The evaluation subblock provides information about the exploitability and impact of potential attacks. Specifically, the "`exploitability-info`" section contains knowledge about the attack vector, complexity, required system privileges, user interactions, and the scope of the attack. Also, the "`impact-info`" provides measures across the three security pillars: confidentiality, integrity, and availability.

Moving to section ③ in Fig. 2, the "`ves-exploit`" component in our data structure lists records, each distinguished by a unique "`<exploit-id>`". Each record encompasses details such as the name of the associated exploit, victim platform, creation and publication timestamps, exploited vulnerabilities identified by CVE IDs, and content, including code and a comprehensive description of the methodology.

Lastly, "`ves-solution`" is depicted in section ④ of Fig. 2, encompassing solution records, each cross-referenced with a corresponding CVE ID from the vulnerability section. A solution record may comprise multiple published solutions, identified by their assigned numbers ("`<solution-number>`") from pertinent security authorities. Furthermore, each record includes details such as the address to the solution source, solution type (*e.g.,* vendor advisory, mailing list, proof of concept, or security focus), patch type (indicating whether a solution is being patched), and timestamps for creation, indexing, and updates.

Our existing data format comprehensively encompasses all cyber risk data types extractable from public repositories, as outlined in §3. Its structure is extensible, providing flexibility to accommodate emerging data types in the future.

4.2 vesData Tool: Collecting Data from Diverse Repositories

Let us now present the design of our vesData tool, which, upon receiving a user-specified IoT product name (*e.g.,* "*Amazon Echo*"), extracts relevant cyber risk data from chosen public repositories using its web crawler and scraper modules. The tool compiles this information into a JSON-formatted text file, as illustrated in Fig. 2, for its output. As outlined in §3.4, our proof-of-concept implementation of the tool leverages the most widely-used repository in each of the three categories (CVE for vulnerability, Exploit-DB for exploit, and NVD for solution). Scaling up the proof-of-concept is deferred to future work.

The data collection process is organized into distinct modules that operate sequentially, each fulfilling a specific purpose. The output JSON file is dynamically constructed on the fly as records are retrieved from repositories instead of being assembled only at the conclusion of the process. The process commences with the "Product Identifier" module, which utilizes the provided product name to retrieve the product identity (CPE) from the NVD database, completing the section ① in our data schema. The subsequent module, the "Vulnerability Extractor", fetches vulnerability records (CVE) and related metrics associated with the product identifier from CVE and NVD databases, populating the section ② in our schema. The third module, "Exploit Collector" within vesData, uses extracted vulnerability IDs to gather corresponding exploit records from Exploit-DB, completing the section ③. Lastly, the "Solution Finder" module searches NVD databases for solution records aligned with product name, vulnerability names/IDs, or exploit names, concluding the construction of the section ④.

Considering the diverse access modes provided by different public repositories—such as web-based browsing for CVE and NVD and script-based APIs for Exploit-DB—our tool incorporates tailored functions to facilitate data retrieval. It can either crawl/scrape HTML webpages using HTTP tags or make direct API calls. To optimize the efficiency of the data-gathering process and mitigate the risk of database blocking resulting from frequent connections, we implement batch requests (*e.g.,* 100 individual requests in a batch)

Table 1: Representative (IoT) products we applied our vesData tool.

Product name	Genre	Size	Time (min)
Linux Kernel	IoT firmware	2.6 MB	519.000
HP printer	Smart printer	1.7 MB	7.000
Apache server	IoT server	630 KB	311.000
TP-Link camera	Smart camera	35 KB	3.638
Smart Things	IoT hub	30 KB	0.068
TP-Link router	Wireless router	29 KB	3.507
Amazon Echo	Smart speaker	16 KB	0.874
Bosch camera	Smart camera	16 KB	0.092
Samsung SmartCam	Smart camera	14 KB	0.054
Netatmo camera	Smart camera	3 KB	0.404

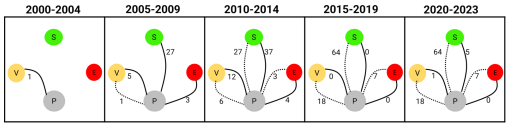
with reasonable time gaps (*e.g.,* a minute) between two successive batches. This approach is particularly emphasized for repositories that support bulk operations, such as NVD and Exploit-DB.

4.3 Applying vesData to Representative Assets

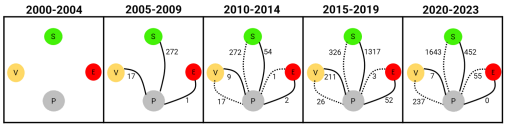
This section demonstrates the performance and usability of our vesData tool, implemented in Python. We executed the tool on a Windows laptop configured with an Intel Core i7 CPU with 16GB RAM, conducting tests by inputting the names of 23 (IoT) products available in our research lab. Seventeen of the products have associated data records in the repositories we utilized. Table 1 lists ten of them, sorted by their data sizes. The source code of vesData and the cyber risk data (in JSON files) associated with the tested products are publicly available at [22].

Response Time: In our tests, vesData performed well without interruptions. Note that data availability in public repositories varies for each product, influenced by factors such as its popularity, support, and interest from the community, including the manufacturer, cloud providers, and potentially malicious actors. A higher volume of data records for a specific product (*e.g.,* Linux IoT kernel or Apache IoT server) likely leads to larger sizes of constructed data and longer response times for our tool, shown as the third and last columns in Table 1. The tool incorporates error-handling mechanisms (*e.g.,* skipping common web search errors like page not found, internal server error, or unauthorized) to enhance data collection efficiency. Table 1 summarizes the response time of our tool when applied to each product. It took less than ten minutes to construct the cyber risk data file for most tested products (20 out of 23). Notably, products with longer track records and greater complexity, such as Linux kernel and Apache server, require relatively more time (*i.e.,* about ten hours and five hours, respectively) for our tool to iteratively search, filter, and fetch relevant data entries in the source repositories.

Practical Limitations: We recognize two specific limitations in our tool identified during the evaluation. Firstly, precision in providing the product name is crucial for comprehensive data retrieval from various public repositories. The tool relies on an exact match as recognized by the technical community. For instance, a misspelling in the product name (*e.g.,* "Netamo camera" instead of "Netatmo camera" with a missing "t") could lead to incomplete data retrieval, as the incorrect name is not recognized by string matching and API calls.Secondly, certain public repositories impose limitations on



(a) HP printer.



(b) Linux kernel.

Figure 3: Temporal graphs of cyber risks for: (a) HP printer and (b) Linux kernel.

query rates. For instance, the NVD database allows a maximum of 5 requests per 30 seconds for users of the free API and 50 requests per 30 seconds for those with paid API credentials. Similarly, Exploit-DB permits 50 requests per day for free users. Consequently, for product names associated with a significant number of records in certain databases (*e.g.,* for Linux Kernel), the execution time may be considerably higher than that of lighter-weight products.

5 TEMPORAL PREDICTION OF CYBER RISKS USING STRUCTURED DATA

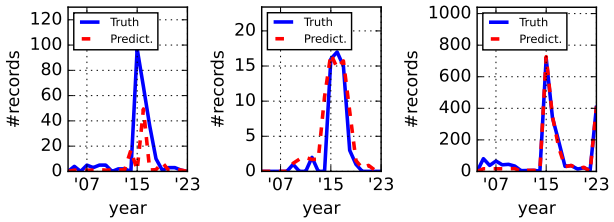
This section showcases how structured data facilitates advanced inference in temporal risk prediction or reasoning. We focus on a fundamental yet representative use case: *predicting the future counts of vulnerabilities, exploits, and solution instances for a specific product.* These predictions can serve as valuable reference points for organizations with large deployments of certain products (say, HP printers), aiding them in more effectively planning their security budgets.

5.1 Temporal Graph Forecasting

Considering the nature of our data, it can be conceptualized as a graph with four major nodes: product (Ⓟ), vulnerabilities (Ⓥ), exploits (Ⓔ), and solutions (Ⓢ). The central node, Ⓟ, connects with the other three nodes through edges. Each edge (with a set of attributes) represents a unit of record in the corresponding section of the vesData format, as illustrated in Fig. 2. Inspired by prior work [8, 10], we employ heterogeneous temporal graphs that seem capable of capturing dynamic interactions between diverse entities that evolve. The entire graph corresponding to a given product's cyber risk data file can be decomposed into a sequence of graphs, each representing a distinct time period with a preferred resolution, such as years or months.

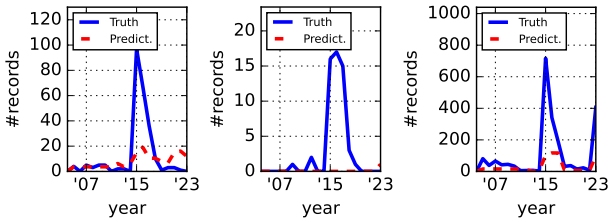
In Fig. 3, we present temporal graphs for two representative products—HP printer and Linux kernel—spanning from 2000 to 2023 with a 5-year resolution. For simplicity, we depict only the number of records (*e.g.,* vulnerabilities) in the data file as edge attributes. Solid edges represent new records in the respective period, while dotted edges signify existing records from the past.

To demonstrate the efficacy of capturing the relational patterns within the generated graphs of cyber risk data in forecasting future vulnerabilities, exploits, and solutions for a specific product, we employ a sophisticated deep neural network model known as the Heterogeneous Temporal Graph Neural Network (HTGNN) [8].



(a) # vulnerabilities. (b) # exploits. (c) # solutions.

Figure 4: Performance of models for Linux Kernel with complete risk data (all three categories).



(a) # vulnerabilities. (b) # exploits. (c) # solutions.

Figure 5: Performance of models for Linux Kernel with partial risk data (only the relevant category).

This model makes predictions by leveraging spatial and temporal dependencies within our generated temporal heterogeneous graphs, with 1-year resolution. Our approach involves standard procedures for training machine learning models, encompassing data pre-processing, model training, parameter tuning, and model selection. However, the details of these steps are not extensively elaborated here. Utilizing a set of historical graphs (*e.g.,* five) as input, each trained HTGNN model for an IoT product aims to forecast the potential count of vulnerabilities, exploits, and solutions expected for the product over the next several years (*e.g.,* one or three). Acknowledging that certain fields in our structured data schema, such as description and scope, are in text format and not directly consumable by graph neural networks, we employed a pre-trained natural language processing (NLP) model called "GloVe" [30] to transform human-readable texts into machine-friendly vector representations/attributes. We emphasize that the fine-tuning of models to achieve highly reliable predictions is beyond the scope of this paper. This aspect is reserved for future work and further exploration. Instead, our primary objective is to showcase the intrinsic value of our structured data for systematic reasoning and inference.

5.2 Evaluations & Preliminary Insights

We train distinct neural network models tailored to the unique knowledge base of three representative products: "HP Printer", "Apache Server", and "Linux Kernel". We develop three specialized models for each product, each dedicated to predicting outcomes within a specific category—namely, vulnerabilities, exploits, or solutions. These specialized models are designed to take the graph representation of the entire vesData for the past N years (with N set to 5 in our default configuration) as input. Based on this historical data, their primary objective is forecasting the number of records in the next year. The models are trained on data from 2000 to 2014 (considered as seen data). They are next evaluated on both seen data (from 2004 to 2014, for learning validation) and unseen data (from 2015 to 2023, for open set evaluation).

Fig. 4 illustrates the results for the Linux product, which boasts the largest amount of historical data among the three representative products under study. The dashed red lines represent the predicted number of records, while the solid blue lines signify the ground truth count. It is evident that the performance is relatively decent, considering no extensive efforts were made to fine-tune the model architecture and input parameters. The prediction curve consistently mirrors the trend of the ground-truth curve across vulnerabilities (Fig. 4a), exploits (Fig. 4b), and solutions (Fig. 4c). Significantly, the predicted numbers of exploits and solutions post-2014 show minimal errors, amounting to less than 10% compared to the ground truth. Similar observations (but relatively more errors) were noted in the results for the other two products with fewer data records.

We next explore the necessity of having comprehensive knowledge for meaningful inference. Specifically, we examine how model performance is influenced when trained on only the relevant fraction of risk data. For instance, in predicting the number of vulnerabilities, rather than providing the entire graph as input, we feed only the vulnerability section of the knowledge graph to the models. The results for the Linux kernel are illustrated in Fig. 5. In contrast to our baseline results in Fig. 4, where input comprises all three categories of data, the performance has become notably unreliable. Evidently, none of the predictions based on individual data categories—solely vulnerability in Fig.5a, solely exploit in Fig.5b, or solely solution in Fig. 5c—managed to capture the levels observed in the ground-truth values. This leads to the conclusion that the relationships among all three types of risk data are crucial for achieving more accurate predictions. We acknowledge that our exploration of risk inference and temporal reasoning on a per-product basis is preliminary. One may consider expanding the inference scope to encompass a network of diverse product types. As highlighted before, this section primarily aimed at showcasing the value of structured data, laying the foundation for potential further research endeavors in this domain.

6 CONCLUSION

The current reliance on public repositories like NVD and Exploit-DB poses challenges due to diverse query formats and fragmented knowledge. Organizations require a unified approach to assess vulnerabilities and predict risks from distributed sources. This paper first mapped public repositories for digital product vulnerabilities, exploits, and patches, highlighting interrelationships and information provided. We developed a data schema, vesData, automating the generation of machine-processable risk knowledge for specific products—our data is openly released along with the tool. Finally, we represented the structured data by heterogeneous temporal graphs, allowing the tracking of IoT cyber risk evolution over time. We trained neural network models on cyber risk graphs for representative products and evaluated them to predict new vulnerabilities and potential patches/exploits for existing ones. Our future work will explore ways to unlock more values from vesData by integrating it with large language models (ChatGPT and Gemini).

REFERENCES

[1] 2024. NTIA. https://ntia.gov/
[2] Guru Bhandari et al. 2021. CVEfixes: Automated Collection of Vulnerabilities and Their Fixes from Open-Source Software. In *Proc. PROMISE*.
[3] Grzegorz J Blinowski and Paweł Piotrowski. 2020. CVE Based Classification of Vulnerable IoT Systems. In *International Conference on Dependability and Complex Systems*. Brunów, Poland.
[4] CVE-Details. 2024. https://www.cvedetails.com/
[5] CXSecurity. 2024. https://cxsecurity.com/
[6] Y. Dong et al. 2019. Towards the Detection of Inconsistencies in Public Security Vulnerability Reports. In *Proc. USENIX Security*. Santa Clara, CA, USA.
[7] Exploit-DB. 2024. https://www.exploit-db.com/
[8] Yujie Fan, Mingxuan Ju, Chuxu Zhang, and Yanfang Ye. 2022. Heterogeneous Temporal Graph Neural Network. In *Proc. SDM*.
[9] Center for Strategic & International Studies. 2024. Significant Cyber Incidents. https://bit.ly/42EBjhI
[10] Peng Gao et al. 2021. A System for Automated Open-source Threat Intelligence Gathering and Management. In *Proc. International Conference on Management of Data*. Virtual Event, China.
[11] Gartner Insights. 2024. Cyber Asset Attack Surface Management (CAASM) Reviews and Ratings. https://www.gartner.com/reviews/market/cyber-asset-attack-surface-management
[12] H. Guo et al. 2022. Detecting and Augmenting Missing Key Aspects in Vulnerability Descriptions. *ACM TSEM* (Apr 2022).
[13] Ayyoob Hamza et al. 2022. Combining Device Behavioral Models and Building Schema for Cybersecurity of Large-Scale IoT Infrastructure. *IEEE Internet of Things Journal* (Jul 2022).
[14] ICS-Advisory. 2024. http://tinyurl.com/uwd9sphs
[15] Raphaël Khoury et al. 2021. An Analysis of the Use of CVEs by IoT Malware. In *Proc. FPS*.
[16] Zhen Ling et al. 2018. IoT Security: An End-to-End View and Case Study. *arXiv preprint arXiv:1805.05853* (2018).
[17] Franco Loi et al. 2017. Systematically Evaluating Security and Privacy for Consumer IoT Devices. In *Proc. ACM IoT S&P*. Dallas, Texas, USA.
[18] Minzhao Lyu, Hassan Habibi Gharakheili, Craig Russell, and Vijay Sivaraman. 2023. Enterprise DNS Asset Mapping and Cyber-Health Tracking via Passive Traffic Analysis. *IEEE Transactions on Network and Service Management* 20, 3 (2023).
[19] Minzhao Lyu, Hassan Habibi Gharakheili, and Vijay Sivaraman. 2022. Classifying and tracking enterprise assets via dual-grained network behavioral analysis. *Computer Networks* 218 (2022), 109387.
[20] Adeel Malik et al. 2021. Robust Cyber-Threat and Vulnerability Information Analyzer for Dynamic Risk Assessment. In *Proc. IEEE MeditCom*.
[21] Adeel A. Malik and Deepak K. Tosh. 2023. Dynamic Vulnerability Classification for Enhanced Cyber Situational Awareness. In *Proc. IEEE SysCon*. Vancouver, Canada.
[22] martazg01. 2024. vesData. https://github.com/martazg01/vesData.
[23] NetScout. 2023. Visibility Is Key to Preventing Outbound and Cross-bound DDoS Attacks. https://bit.ly/3HW2NFO
[24] NVD. 2024. https://nvd.nist.gov/
[25] OpenCVE. 2024. https://www.opencve.io/welcome
[26] Pascal Oser et al. 2022. Risk Prediction of IoT Devices Based on Vulnerability Analysis. *ACM Transactions on Privacy and Security* (2022).
[27] Packet Storm Security. 2024. http://tinyurl.com/na5sus3d
[28] PatchDB. 2024. https://sunlab-gmu.github.io/PatchDB/
[29] Patchwork. 2024. https://patchwork.ozlabs.org/
[30] Jeffrey Pennington et al. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing*.
[31] Carlos A Rivera A et al. 2021. Is This IoT Device Likely to Be Secure? Risk Score Prediction for IoT Devices Using Gradient Boosting Machines. In *Proc. EAI MobiQuitous*. Virtual Event.
[32] securityfocus. 2024. http://tinyurl.com/2w4xsw23
[33] Yun Shen et al. 2018. Tiresias: Predicting Security Events Through Deep Learning. In *Proc. ACM CCS*. Toronto, Canada.
[34] Snyk -Vulnerability. 2024. https://security.snyk.io/
[35] The MITRE Cooperation. 2024. CVE. https://cve.mitre.org/
[36] VulDB. 2024. https://vuldb.com/
[37] Vulnerability & Exploit. 2024. http://tinyurl.com/2r3dezxs
[38] Vulnerability-Lab. 2024. http://tinyurl.com/3ec57br8
[39] Xinda Wang et al. 2021. Patchdb: A Large-Scale Security Patch Dataset. In *Proc IEEE/IFIP DSN*. Virtual Event.
[40] Kinza Yasar. 2023. Cyber Insurance. https://bit.ly/3wdvL1k