MSc Artificial Intelligence Thesis

Are Advanced Persistent Threats all that matter for a National Security Center?

An analysis of temporal and ontological alignment.

Author:
Anniek Jansen

Supervisor:
Fabio Massacci

Second Reader:
Katja Tuma

GitHub Repository:
https://github.com/anniekjansen/ThreatIntelligenceAlignment

November 6, 2024

Are Advanced Persistent Threats all that matter for a National Security Center? An analysis of temporal and ontological alignment.

***Abstract*—This study examines the alignment and consistency of threat intelligence data between an Advanced Persistent Threat (APT) dataset and the National Cyber Security Centre (NCSC) datasets. The study addresses gaps in the existing literature on the validation and practical implementation of ontological frameworks for national security data. Using temporal and ontological analyses, the research evaluates the alignment of temporal attributes and consistency of technical details across both datasets. In particular, we introduce the notion of explained and unexplained changes in likelihood of vulnerability exploitation at the report of the NCSC. The findings indicate a strong alignment between the APT reserved and exploitation times and the NCSC publication timelines, albeit with a dangerous gap. Key discrepancies are identified in the technical details of the vulnerabilities related to the affected products and operating systems. These results highlight the need for improved collaboration among cyber threat intelligence organizations to ensure timely and consistent vulnerability reporting.**

***Index Terms*—Threat Intelligence, Vulnerability Assessment, Temporal Analysis, Ontological Framework**

1. Introduction

As cyber attacks become increasingly complex, threat intelligence has become vital for organizations to anticipate and mitigate risks [1]. It involves gathering, analyzing, and sharing information on current and potential threats. However, uncertainty in threat intelligence data complicates timely decision-making, making an effective representation of uncertainty crucial to improving vulnerability prioritization [2] [3].

National security centers respond to threat intelligence in different ways [4], [5], but they invariably publish a report of vulnerabilities that should be patched with more or less urgency. Such publications provide a variety of indicators, and it is still unclear which indicators should be used to drive the prioritization of vulnerabilities [6]. This is not a purely academic debate, as witnessed by the industry discussion between Common Vulnerability Scoring System (CVSS)[1] and Exploit Prediction Scoring System (EPSS)[2]. The EPSS claims to be based on live insights [7] but the authors' own honest analysis (Fig. 7 in [8]) shows that, once we remove the obvious tags being a Microsoft software (popularity is not helping here) and Remote Code Execution (unsurprisingly), the major source of their score is actually CVSS itself.

While the search for the best assessment method for vulnerability prediction and assessment might be an elusive quest [9], some research indicates that collateral information may help explain exploitation potential. Since the initial paper introducing case-control studies in security [10], independent studies have shown that analyzing what malicious actors sold or did can make a difference [11] [12] with possibly controversial results [13] (see the point-counterpoint discussion on ACM CACM [14] [15]).

If actions by malicious actors indeed influence cyber security outcomes, the interesting question arises whether such information also empirically dominates the alerts issued by national cyber security centers. Existing literature lacks studies that evaluate data alignment and consistency across different sources of threat intelligence. This research addresses these gaps by conducting temporal and ontological analyses using data from the National Cyber Security Centre (NCSC), the Dutch Cyber Security Center, and Advanced Persistent Threats (APT) datasets, the latter comprising over 350 APT reports [13]. Two key research questions guide this study; the first focuses on the alignment of threat publication timings between NCSC and APT data, and the second examines the consistency of technical details to evaluate threat intelligence quality.

The paper is structured as follows: Section 2 presents a comprehensive literature review, including key findings that underscore research gaps and inform the research questions outlined in Section 3. Section 4 describes the methodology for the data analysis of the NCSC dataset, while Section 5 covers the approach for the APT dataset. The temporal and ontological analyses are detailed in Sections 6 and 7. Finally, the study

1. CVSS: https://www.first.org/cvss/

2. EPSS: https://www.first.org/epss/

concludes with a discussion in Section 8, threats to validity in Section 9 and final conclusions in Section 10.

2. Systematic Literature Review

2.1. Process

This literature review investigates the state of knowledge on uncertainty representation in threat intelligence. A keyword-based search is performed to retrieve relevant sources from Google Scholar, Scopus, and the Association for Computing Machinery (ACM) Digital Library. The exact keywords are available in the replication guide (B.1). The collected sources are then screened based on their title, abstract, and a set of selection criteria. The final sources are clustered and analyzed using thematic analysis to assess various uncertainty techniques. The complete process, including the number of sources collected per step, is shown in Figure 1, following a methodology similar to [16].

The initial three keyword searches produced a large volume of academic sources, particularly from Google Scholar. Specifically, the first query generated $3,940$ sources from Google Scholar, while Scopus and the ACM Digital Library yielded 237 and 97 relevant sources, respectively. Table 15 summarizes the results of each search and the final selection of sources.

From the initial three searches, $1,411$ unique sources were collected, and the refined search added 34 more. Screening based on titles, abstracts, and selection criteria narrowed the selection to 51 and 11 sources, respectively. The refined search yielded 15 sources, with 7 passing the criteria. In addition, 4 new sources were identified using the direct snowball approach.

2.2. Assessment Dimensions

Sources are evaluated along four dimensions: technology, research method, study domain, and measure. Except for the technology criterion, each has a set of three sub-criteria.

Method. The validity of a source's conclusions is largely influenced by its research method. This study prioritizes methodologies that focus on developing and testing techniques to represent uncertainty. The key sub-criteria include case studies, experiments, and tools, while surveys and interviews are excluded, as they are not relevant to this approach.

Domain. The second dimension is the source's domain. Although the focus is on threat intelligence, related fields such as cyber security are also relevant. Threat intelligence focuses on monitoring and countering security threats, while cyber security aims to protect cyber systems from threats or cyber attacks in advance, as described by [17]. The methods used to represent uncertainty in cyber security are beneficial for this literature review as well. National security, with a focus on counterinsurgency and military operations, is also considered relevant.

Measure. This research investigates uncertainty, which is a complex concept influenced by various factors. To address this ambiguity, we review sources that explore related metrics such as confidence or reliability measures, as well as sources describing data quality metrics such as accuracy, completeness, and consistency.

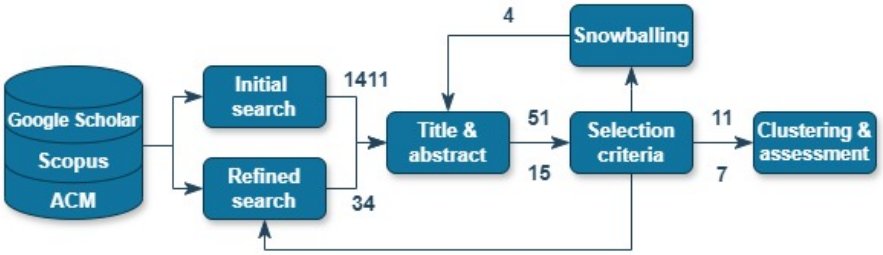
Technology. The final dimension is the technology used to represent uncertainty. Unlike the other criteria, it lacks strict sub-criteria to allow for a broad investigation of practical technologies. These include a wide range of techniques such as ontologies, Bayesian networks, quantitative techniques, vector representation, graph matching, explainable AI, and fuzzy description logic.

2.3. Literature Results

The analysis of the 18 sources provides a comprehensive perspective on existing research on the representation of uncertainty in threat intelligence. The literature focuses on technological approaches, with ontologies and Bayesian Networks as the most commonly used technologies. Other techniques include blockchain technology for assessing the reliability of threats, vector representation for data aggregation, and inexact graph matching for uncertainty evaluation through similarity scores. Furthermore, explainable artificial intelligence improves the understanding of intrusion detection alerts, while combined data functions and expert-based profiling offer richer uncertainty representation. Weighted evaluation methods continuously reassess uncertainty, and fuzzy description logic captures uncertain cyber knowledge. Table 1 presents a summary of the key findings per technology, including the missing criteria per cluster.

The literature employs case studies, experimentation, and validation tools, with case studies dominating and focusing on real-world examples. Experimentation is infrequent, primarily addressing blockchain technology and XAI, while tool-based validation is the most common approach, using various frameworks.

The primary emphasis is on threat intelligence, with most sources addressing the collection and analysis of cyber threat information. A limited number of sources focus solely on cyber



The initial search yielded 1,411 sources, with 34 identified in the refined search. After screening the titles, abstracts, and applying the selection criteria, 18 sources (11+7 resp.) were selected, including 4 obtained by snowball sampling.

Figure 1. Systematic Literature review process, including the number of sources collected per step.

TABLE 1. Summary of collected literature per technology.

Technology	Sources	Summary	Missing Criteria
Ontologies	[18] [19] [20] [21] [22] [23]	The URREF ontology is used primarily to represent uncertainty. Its integration with secondary ontologies enhances semantic uncertainty for more detailed datasets.	*Experiments* *National Security* *Data Quality*
Bayesian Networks	[19] [20] [21] [22] [24]	Bayesian networks effectively represent causal models for decision-making under uncertainty. Most sources combine BNs with the URREF ontology.	*Experiments* *Cyber Security* *Data Quality*
Vector Representation	[25] [26]	Vectors effectively represent and aggregate key relevant values associated with threats.	*Case Studies & Experiments* *National Security* *Uncertainty & Confidence/Reliability*
Inexact Graph Matching	[27] [28]	Inexact graph matching calculates similarity scores for each node, offering confidence measures based on comparisons with a template graph.	*Experiments & Tools* *Threat Intelligence & Cyber Security* *Data Quality*
Explainable AI	[29]	Explainable Artificial Intelligence, combined with data cleaning techniques, clarifies uncertainties inherent in threats	*Case Studies* *Cyber & National Security* *Uncertainty & Confidence/Reliability*
Combining Data Functions	[30]	An expansive research approach involves combining various technologies. Each function contributes to an updated knowledge base of threats.	*Experiments* *Cyber & National Security* *Confidence/Reliability & Data Quality*
Expert-based Profiling	[31]	Experts offer qualitative assessments of threat history and motivations, allowing probability calculations.	*Experiments* *Cyber & National Security* *Confidence/Reliability & Data Quality*
Weighted Evaluation	[32]	The weighted evaluation method continuously evaluates the trust and uncertainty of cyber threats using distinct parameters.	*Experiments & Tools* *Cyber & National Security* *Uncertainty & Confidence/Reliability*
Description Logic	[33]	A fuzzy description logic represents uncertainty in cyber knowledge using SROIQ description logic subsets.	*Experiments* *Cyber & National Security* *Confidence/Reliability & Data Quality*

security, while several studies apply research in the national security domain.

Representing uncertainty is a central theme, with many sources exploring its implications for data trustworthiness. Confidence and reliability measures are implemented in various studies, and data quality is assessed through metrics such as accuracy and completeness.

2.3.1. Method. Case studies are a key research validation method, used in 12 sources. Four sources focus on air traffic control [19] [20] [21] [22], two on person identification [27] [28], and one on social media for crisis management [23]. Another two examine military missions [30] [24], and three cyber threat cases [31] [32] [33].

Experiments are the least common method, used in only 3 of 18 sources. Two focus on blockchain for data aggregation [34] [35]. The other source evaluates XAI support through human analyst experiments [29].

Tool development is the most common method, used in 14 sources. These include frameworks for executing models and architectures such as CASE [18], blockchain-based cyber threat intelligence architecture [34], URREF ontology [19] [20] [21] [22] [23], privacy preserving protocol [25], intrusion detection ORISHA platform [35], Overt Vulnerability system named OVANA [26], FAIXID [29], MU tool [30], fuzzy description logic formalism [33], and a multi-level information fusion framework [24].

2.3.2. Domain. Most of the sampled literature focusses on threat intelligence, covering topics such as intrusion detection [35], vulnerability assessment [26] [29], and threat intelligence sharing [32]. Cyber security is closely related to threat intelligence due to its common focus on cyber systems. Two sources are directed solely towards the handling process of cyber data [18] [25]. These examine the CASE specification language on the handling of cyber-information [18] and privacy-preserving protocols for cyber-data [25]. Five of the sources relate to national secu-

rity with the emphasis on safeguarding a nation's sovereignty. The sources focus on counterinsurgency, commonly referred to as COIN [27] [28], and military missions [30] [24].

2.3.3. Measure. Ten sources emphasize uncertainty to represent the trustworthiness of data. In addition to numerous sources researching the URREF ontology [19] [20] [21] [22] [23], two articles focus on confidence and similarity scores [27] [28], while others apply combining functions to express qualitative measures [30], assess the likelihood of threats by expert-based profiling [31], or handle True/False description logic statements [33]. Six sources implement confidence or reliability measures. Specifically, one source explores the Admiralty Code Credibility Scale to assign a confidence property [18]. The two blockchain-based studies compute reliability and confidence scores by accumulating multiple data inputs [34] [35]. The graph matching studies employ various confidence and similarity measures [27] [28], while DBNs use reliability scores to analyze variables reaching confidence thresholds [24]. Four sources prioritize data quality, evaluating metrics such as accuracy [26] [29], completeness [25] [26] [29] [32], consistency [25] [29], timeliness [25] [29] [32], uniqueness [25] [26] and validity [25] [32].

3. Research Questions

Most of the reviewed literature implements ontological frameworks to represent uncertainty in threat intelligence. The URREF ontology stands out, but the ontology cluster lacks experimental validation, national security application, and data quality evaluation. This research addresses these deficiencies by performing temporal and ontological experiments using national security data and applying data quality metrics. This study investigates the alignment and consistency of threat intelligence data between the NCSC and APT datasets to represent uncertainty, addressing the following two research questions:

RQ1: *How do the reserved and exploitation times of the APT align with NCSC updates, and how do the justification and likelihood classifications of vulnerabilities impact the temporal differences between the publication and exploitation of threats?*

This research question is addressed using a comprehensive methodology that includes temporal analyses, statistical analyses, and the examination of temporal differences between publication times to identify significant relationships.

RQ2: *How do the product, operating system, and version details of the vulnerabilities from the APT align with the data from the NCSC, and how can this alignment be evaluated using the URREF ontology to assess the consistency of threat intelligence?*

The second research question focuses on the technical details of the vulnerabilities and will include an ontological analysis to examine the alignment and consistency of the two threat intelligence datasets. The ontological analysis will employ the URREF ontology to systematically evaluate and compare the technical attributes of vulnerabilities recorded in the NCSC and APT datasets.

4. NCSC Security Advisories

The first dataset consists of security alerts from the Dutch *Nationaal Cyber Security Centrum*[3]. These advisories are published in response to newly discovered vulnerabilities, providing descriptions, potential impacts, and solutions. The earliest entry dates back to June 4, 2002, while the most recent entry is from January 20, 2023.

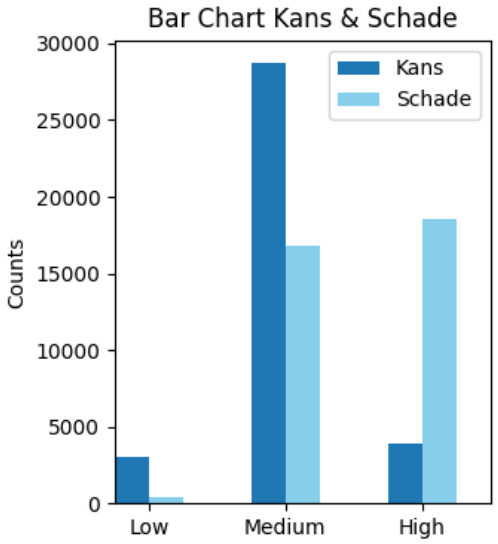
TABLE 2. Data summary of the NCSC dataset.

Data Summary	NCSC
Instances	35,684
Attributes	15
Duplicates	6,861
Missing values	7,639

Table 2 highlights that 6,861 of the 35,684 rows in the dataset are duplicates, suggesting possible data collection issues. Of the 7,639 missing values, the majority, specifically 7,483, are in the `Schadeomschrijving` (description of impact) column, reflecting optional reporting for the sub-descriptions. Other attributes with missing values are `Beschrijving` (description) and `Mogelijke oplossingen` (potential mitigations), with 95 and 61 missing values, respectively. Figure 2 shows the distribution of likelihood and impact levels, with most threats rated `Medium` likelihood.

Data pre-processing tasks cleaning the dataset to improve the accuracy and reliability of the research are detailed in the replication guide in Appendix B.2. After sorting by `NCSC ID` and `Update`, four new indicator variables were created, generating 13,450 instances and 6

3. NCSC Beveiligingsadviezen: https://advisories.ncsc.nl/



Most vulnerabilities have a medium likelihood (Kans) of occurrence, with an impact (Schade) nearly evenly divided between medium and high.

Figure 2. Distribution of likelihood and impact.

attributes. As summarized in Table 3, most updates did not affect the likelihood classification, with only 250 considered important.

TABLE 3. NCSC Update Justification Summary

Type	Count	Description
No change	11,507	Same description + same likelihood
Unjustified change	122	Same description + change in likelihood
Unimportant change	2,363	Change in description + same likelihood
Justified & important change	250	Change in description + change in likelihood

5. APT Dataset

The Advanced Persistent Threats (APT) dataset serves as a reference for this study, covering APT campaigns from 2008 to 2020 [36]. The dataset used is provided in the CSV file named *campaigns_vulnerability_vector_product_version_os*[4] and is derived from both structured and unstructured threat sources. Structured sources include MITRE Att&ck and the National Vulnerability Database (NVD), both publicly available. MITRE Att&ck categorizes and describes cyber attacks and intrusions[5], while the NVD is U.S. government platform for standards-based vulnerability data.

4. APT GitHub: https://github.com/giorgioditizio/APTs-database/blob/v1.0.1.1/data_analysis/campaigns_vulnerability_vector_product_version_os.csv

5. MITRE Att&ck: https://attack.mitre.org/

TABLE 4. Data summary of APT dataset.

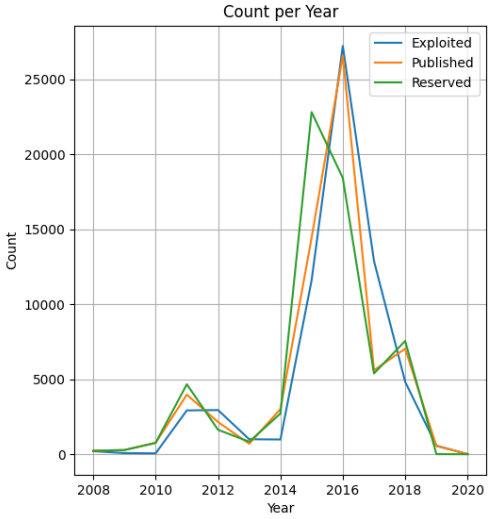
Data Summary APT	
Instances	65,552
Attributes	10
Duplicates	-
Missing values	1,463

Table 4 summarizes the dataset, which includes 65,552 instances across 10 attributes. Each APT is represented by rows detailing various `product`, `version`, `update` and `os` information, but only 118 unique IDs for the `vulnerability` attribute. Although there are no duplicate entries, 1,463 missing values are distributed across 7 attributes, with exactly 209 missing data points per attribute. For these entries, only `campaign`, `attack_vector`, `exploited_time` have available data. Certain attributes match those in the NCSC dataset:

`vulnerability` → `CVE-ID`
`published_time` → `Uitgiftedatum`
`product` → `Toepassingen`
`version` → `Versies`
`os` → `Platformen`

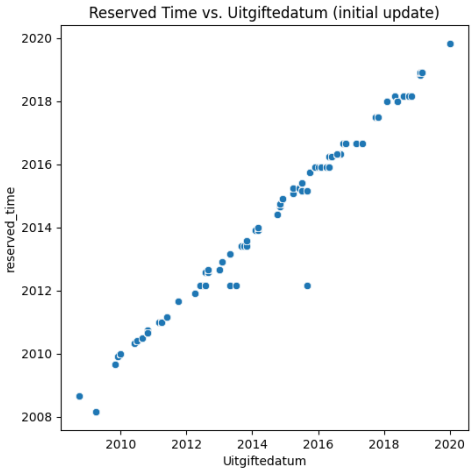
Similarly to the NCSC data, the APT dataset requires pre-processing to include only relevant data for the analysis. Attributes with a large proportion of missing values and inconsistencies (e.g. value denoted by "*") are eliminated, and time values are standardized. After processing, the dataset retains 65,552 instances and 9 attributes, with 209 instances removed due to missing values in key features. Full details are provided in the Replication Guide (B.3).

The conversion to a uniform time enhances the accuracy of the temporal analysis. Figure 3 illustrating the temporal progression of the threat publications. The `reserved_time` attribute, representing the initial reservation of the CVE by MITRE, generally precedes the `published_time` attribute, indicating that threats are typically reserved by MITRE before they are officially published by the NVD. After publication, the `exploited_time` attribute generally reflects the latest dates, suggesting that exploitation tends to occur after publication. This sequential pattern suggests a typical life cycle of a threat, starting with its reservation, followed by its official publication, and concluding with its exploitation. However, this could also mean that before a vulnerability is officially named, exploitation events may not be recorded due to the absence of a formal entry in the database.



Exploitation typically occurs after the vulnerability is reserved and published, indicating that most *recorded* exploits happen for known vulnerabilities.

Figure 3. Line plot of time-related columns of APT dataset.



A strong positive relationship exists between the time in which the vulnerabilities exploited by ATP has been reserved (ATP Reserve time) and NCSC publication of vulnerabilities, with only a few outliers.

Figure 4. Scatter plot `reserved_time` to Initial `Uitgiftedatum`.

6. RQ1: Temporal Analysis of Threats

This section addresses the first research question by analyzing how the reserved and exploitation times of the APT dataset correlate with the initial and subsequent updates from the NCSC platform. The analysis also considers how update justifications and likelihood classifications affect these temporal relationships. By examining these factors, the research aims to uncover patterns that inform better risk assessment and resource allocation in cyber security.

The temporal analysis covers four areas, each producing distinct results. It explores how APT reservation and exploitation times align with initial and subsequent NCSC updates, as well as whether justification and likelihood classifications affect timing differences between APT and NCSC publications, highlighting uncertainties in threat publication timing.

6.1. Reserved Time to Initial Release

The Pearson correlation between the APT reserved time and the initial NCSC release time is extremely high, with a coefficient of 0.98 and a p-value smaller than 0.001, indicating a strong positive linear and significant relationship. Figure 4 visually confirms this, showing that APT reserved times closely match the initial NCSC publication times.

The Mean Absolute Error (MAE) between APT reserved time and initial NCSC publication time is 1.98 months, suggesting that, on average, NCSC publications occur approximately two months after the vulnerabilities exploited by an APT have been reserved (for short *APT reserve time*). Table 5 shows the detailed results, indicating good alignment between the APT reservation time and initial NCSC publication.

TABLE 5. APT reserved time to NCSC initial publication.

	Reserved Time
Pearson correlation	0.982^{***}
Mean Absolute Error	1.98 months

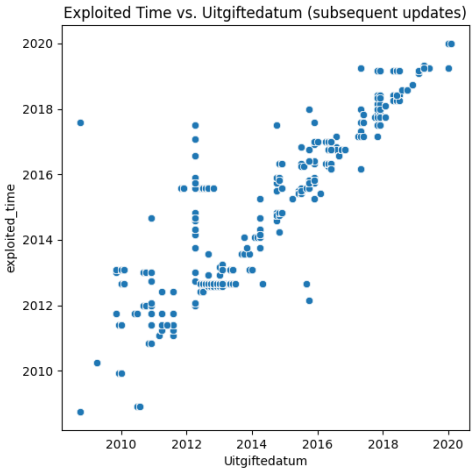
$^{***}(p < 0.001)$

Key Finding: The APT reserved time and the initial NCSC release time align closely, with an average delay of nearly two months. From a security perspective, this is bad news: defenders have (statistically significant) difficulties to keep up with attackers.

6.2. Exploited Time to Next Updates

The Pearson correlation between the APT exploitation time and the publication time of subsequent NCSC updates is also notably high, with a coefficient of 0.94 and a p-value lower than 0.001. Although the relation is very strong and significant, the correlation is slightly lower than the reserved-to-initial release analysis, suggesting that subsequent updates are more variable. Figure 5 shows a wider spread of data points, reflecting greater uncertainty as more updates on the same threat are published.

The MAE between APT exploitation times and subsequent NCSC updates is 2.59 months, indicating a higher deviation compared to the



There is a moderate positive relationship between APT exploitation and NCSC republication, with some variability and a couple outliers. Not everything is caught on the spot.

Figure 5. APT Exploitation and Updates of NCSC Alerts

reserved-to-initial analysis. This larger error reflects increased complexity and variability in predicting subsequent updates.

TABLE 6. Results APT exploited time to NCSC subsequent publications.

Exploited Time	
Pearson correlation	0.938^{***}
Mean Absolute Error	2.59 months

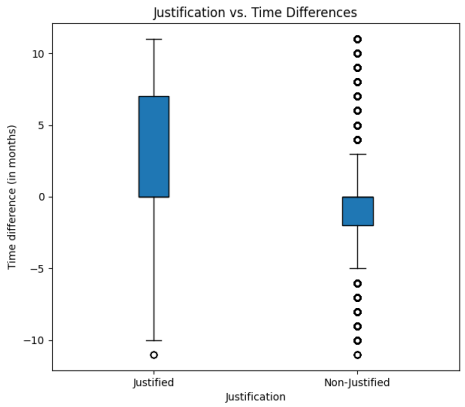
$^{***}(p < 0.001)$

Key Finding: APT exploitation time and subsequent NCSC publication times align closely, with more variability than initial releases. The deviation of about two and a half months reflects the complexity of this relationship as more updates on the same threat are released.

6.3. Update Justification on Timing

The analysis of how the justification of updates affects the timing differences between APT and NCSC publications shows a weak linear relationship, with a Pearson correlation of 0.07 and a p-value less than 0.001. Thus, the justification of updates does not substantially influence the publication timing of updates.

The box plot in Figure 6 reveals that unjustified updates show more variability and outliers than justified ones. Although the median of the unjustified updates is close to zero, the positive median of the justified group suggests that the publication by the NCSC generally follows the exploitation by the APT.



The distribution of vulnerabilities with non-justified changes exhibits greater variability and more outliers than justified updates.

Figure 6. Box plot of justification.

Table 7 confirms a significant difference in the mean exploitation times for threats with justified versus unjustified updates, supported by a T-statistic of 101.28 and a p-value less than 0.001.

TABLE 7. Results justification on time differences.

Justification	
Pearson correlation	0.067^{***}
T-statistic	101.28^{***}

$^{***}(p < 0.001)$

Key Finding: Although the weak linear relationship of the update justifications with timing differences, justified updates have fewer outliers and a more consistent publication pattern than unjustified ones.

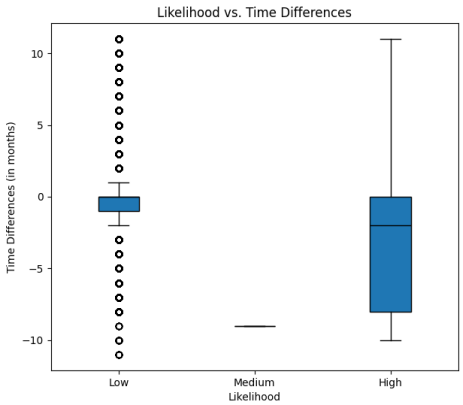
6.4. Likelihood on Timing

The final analysis investigates how likelihood classifications (`Low`, `Medium`, and `High`) impact the time differences between NCSC publication and APT exploitation. Table 8 reveals that the Pearson correlation is -0.44, with a p-value smaller than 0.001. This suggests a moderate inverse relationship, where higher-likelihood threats tend to have smaller time gaps. The ANOVA test, with an F-statistic of 247749 and a p-value smaller than 0.001, confirms significant timing differences between likelihood groups.

TABLE 8. Results likelihood on time differences.

Likelihood	
Pearson correlation	-0.438^{***}
F-statistic	247749^{***}

$^{***}(p < 0.001)$



High likelihood cases show more variability in timing, often with exploitation happening before publication, but the spread suggests that delays in either publication or exploitation are common.

Figure 7. Box plot of likelihood.

The box plot in Figure 7 shows that the `High` likelihood group has more variability in timing differences, while the `Medium` likelihood group shows a much more consistent timing pattern. The medians of all groups are below zero, suggesting that exploitation by the APT tends to happen before publication by the NCSC.

Key Finding: A moderate inverse relationship between likelihood and time differences suggests higher likelihood threats having smaller time gaps, explaining timing variability. However, numerous outliers suggest that several vulnerabilities may have history of their own.

7. RQ2: Ontological Analysis

Following the temporal analysis, the second research question focuses on the alignment and consistency of technical details between the NCSC and APT datasets. The URREF ontology is key for this analysis for the following reasons:

- **Structured Data Mapping**: The ontology provides a common framework for effectively aligning the technical details across both datasets.
- **Consistency Evaluation**: The ontology identifies and reports data discrepancies.
- **Re-usability & Communal Coherence**: Extending an established ontology ensures uniformity, thus improving comprehension and collaboration among stakeholders.

The analysis begins by mapping the threat intelligence data from both datasets to the URREF ontology, downloaded from its GitHub repository[6]. SPARQL queries are run on the populated ontology within the `GraphDB` environment to retrieve and manipulate RDF data. The queries explore various aspects of the data, enabling a detailed comparison of vulnerabilities, products, and operating systems.

The ontological analysis includes six investigations focused on vulnerabilities, products, operating systems, update justifications, and likelihood assessments. The SPARQL query results show relationships among these elements, highlighting patterns, trends, and uncertainties in threat intelligence.

7.1. Vulnerability Analysis

The analysis uses eight SPARQL queries, summarized in Table 9. Query 1.1 reveals that all 86 vulnerabilities share at least one affected product in both datasets. In contrast, Query 1.2 indicates that only 24 vulnerabilities have identical affected products. Queries 1.3 and 1.4 reveal 45 vulnerabilities tied exclusively to the NCSC dataset and only 6 linked to products unique to the APT dataset, suggesting minimal overlap.

Queries 1.5 and 1.6 explore vulnerabilities with shared operating systems for the affected products, while Queries 1.7 and 1.8 focus on affected versions. Both Query 1.5 and Query 1.7 confirm that 24 vulnerabilities share all products across both datasets based on at least one common operating system or version. However, no vulnerabilities have identical products with fully matching operating systems (Query 1.6) or versions (Query 1.8).

Key Finding: Among the 86 vulnerabilities, all share at least one affected product between NCSC and APT datasets, but only 24 share identical products, with no common operating systems or versions.

7.2. Product Analysis

The product analysis employs five queries. Query 2.1 identifies the top five most frequently shared products between the NCSC and APT datasets, along with vulnerability counts:

1) `Microsoft Windows` (25)
2) `Adobe Flash Player` (24)
3) `Microsoft Internet Explorer` (20)
4) `Microsoft Office` (14)
5) `Adobe AIR` (9)

6. URREF GitHub: https://github.com/adelphi23/urref/blob/master/UREF.owl

TABLE 9. Results queries vulnerability analysis.

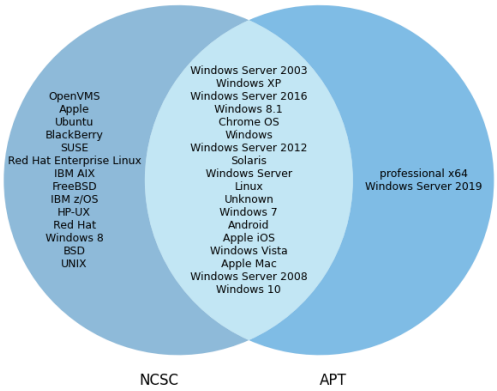
	SPARQL Query	Total Vulnerabilities	Percentage of Total
1.1	Any product in common between NCSC and APT	86	100%
1.2	All products identical for NCSC and APT	24	27.9%
1.3	No NCSC products present in APT	45	52.3%
1.4	No APT products present in NCSC	6	7.0%
1.5	All products and any operating system in common between NCSC and APT	24	27.9%
1.6	All products and all operating systems identical for NCSC and APT	0	0%
1.7	All products and any affected version in common between NCSC and APT	24	27.9%
1.8	All products and all affected versions identical for NCSC and APT	0	0%

In the NCSC dataset, `Unknown` ranks as the second most frequent product (Query 2.2) with 24 unique vulnerabilities. In contrast, the APT dataset shows the same top five affected products as listed above (Query 2.3). Further analysis reveals that 22 of 41 products (53.6%) are exclusive to the NCSC (Query 2.4), while 2 of 21 affected products (9.5%) are exclusive to the APT dataset (Query 2.5).

Key Finding: The most frequently shared affected products include Microsoft Windows and Adobe Flash Player. More than half of NCSC products are absent from the APT, with only two products unique to APT.

7.3. Operating System Analysis

The operating systems of the affected products run follow the same methodology as the product analysis, focusing on frequency and exclusivity. The results are visualized in the Venn diagram in Figure 8.



Despite many operating systems being listed exclusively by the NCSC, the majority are shared between the NCSC and APT datasets.

Figure 8. Venn diagram results OS analysis.

Query 3.1 identifies 18 distinct operating systems shared between both datasets, linked to all 86 vulnerabilities. Most operating systems appear at the intersection of the Venn diagram. Query 3.2 shows that 24 of the 32 unique operating systems (75%) in the NCSC are associated with all 86 vulnerabilities. Similarly, Query 3.3 reveals that 18 of the 20 unique APT operating systems (90%) are connected to all vulnerabilities. Queries 3.4 and 3.5 indicate that 14 of 32 unique operating systems (43.8%) are exclusive to the NCSC dataset, while APT features only 2 unique operating systems (10%): `Windows Server 2019` and `Professional x64` (Query 3.5).

Key Finding: The majority of affected operating systems are shared while only 10% are unique to APT. In other words, and quite reassuringly, almost anything exploited by an APT is going to make to the NCSC.

7.4. Update Justification Analysis

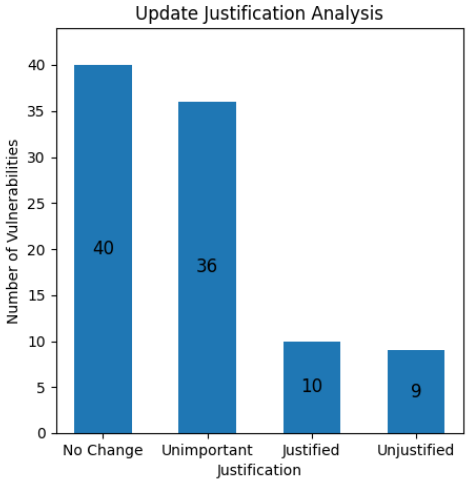
The analysis of the update justifications is performed using four different SPARQL queries. Figure 9 shows that 40 of the 86 vulnerabilities were updated without any modifications to their description or likelihood. 36 updates are deemed unimportant, while 10 involve justified updates, and 9 are classified as unjustified. The total number of justifications exceeds the 86 unique vulnerabilities. Only three vulnerabilities are linked to multiple justification types.

Query 4.3 summarizes unique vulnerabilities per justification for the most frequently shared products. For `Microsoft Windows`, 25 vulnerabilities have 4% classified as justified updates and 52% as unimportant. For `Adobe Flash Player`, 24 vulnerabilities do not exhibit unjustified cases, with 79.2% linked to updates without changes. The top five products predominantly reflect unimportant updates or without changes, as highlighted in Table 10.

Key Finding: Most NCSC vulnerability updates show no changes or are classified as unimportant, a trend consistent with the most affected products as well.

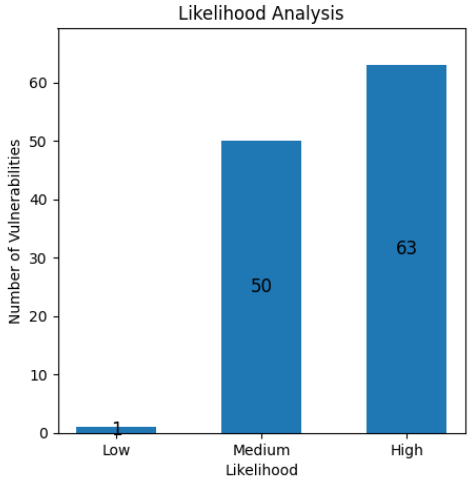
TABLE 10. Vulnerability counts for most frequently affected products per update justification.

Product	Justified		Unjustified		Unimportant		No Change		Total Vulnerabilities
MS Windows	1	4%	5	20%	13	**52%**	9	36%	25
Adobe Flash Player	1	4.2%	0	0%	4	16.7%	19	**79.2%**	24
MS Internet Explorer	3	15%	1	5%	10	**50%**	6	30%	20
MS Office	2	14.3%	2	14.3%	6	42.9%	7	**50%**	14
Adobe AIR	1	11.1%	0	0%	1	11.1%	7	**77.8%**	9



Most vulnerability updates exhibit either no change or are deemed unimportant, with only a small proportion being justified or unjustified. Multiple updates per vulnerability are possible.

Figure 9. Bar chart results justification analysis.



Most vulnerabilities exhibit a high likelihood of occurrence, followed by a significant large number of medium-probability threats. Multiple likelihoods per vulnerability are possible.

Figure 10. Bar chart results likelihood analysis.

7.5. Likelihood Analysis

The likelihood analysis examines threat occurrence probabilities. Figure 10 shows only one vulnerability classified with low likelihood, while 50 have medium and 63 high likelihoods, indicating most vulnerabilities are likely to be exploited. Again, the total likelihood labels exceed the number of unique vulnerabilities, with 27 vulnerabilities assigned multiple likelihood classifications, showing that their likelihood assessments changed over time.

Table 11 summarizes the unique vulnerabilities categorized per likelihood for the most frequently shared products. Notably, `Microsoft Windows` shows 72% of vulnerabilities as medium likelihood, while `Adobe Flash Player` indicates 75% as medium likelihood.

The number of vulnerabilities with high likelihood per update justification is detailed in Table 12. Justified and unjustified updates show high likelihood rates of both 100%, while updates classified as not having changes or unimportant updates show lower rates of 75% and 64%, respectively (Query 5.4).

Key Finding: Most vulnerabilities are classified as medium or high likelihood, with a third showing changing probability levels. High-likelihood vulnerabilities are common in Microsoft Internet Explorer and Adobe Air, and both justified and unjustified updates consistently lead to high-likelihood classifications.

7.6. Comparative Risk Analysis

The first comparative risk analysis, based on Query 6.1, investigates the odds ratio for `Justified` updates across the most frequently affected products. As summarized in Table 13, the odds ratios for `Microsoft Windows`, `Microsoft Internet Explorer`, and `Microsoft Office` are all 1.69, while `Adobe Flash Player` and `Adobe AIR` exhibit a higher ratio of 2.58. This indicates that these products are 1.69 and 2.58 times more likely, respectively, to be associated with vulnerabilities that have justified updates compared to the other justification categories.

Query 6.2 examines the likelihood of vulnerabilities classified as `High` for the same products. As shown in the right col-

TABLE 11. Vulnerability counts for most frequently affected products per likelihood.

Product	Low		Medium		High		Total Vulnerabilities
MS Windows	1	4%	18	**72%**	15	60%	25
Adobe Flash Player	0	0%	18	**75%**	12	50%	24
MS Internet Explorer	0	0%	10	50%	18	**90%**	20
MS Office	0	0%	10	71.4%	10	71.4%	14
Adobe AIR	0	0%	2	22.2%	9	**100%**	9

TABLE 12. Vulnerability counts of high likelihood per justification.

Justification	High Likelihood		Total Vulnerabilities
Justified	10	**100%**	10
Unjustified	9	**100%**	9
No Change	30	75%	40
Unimportant	23	64%	36

umn of Table 13, Microsoft Windows and Microsoft Office have odds ratios of 0.54 and 0.72, respectively. This indicates that it is less likely for these products to be associated with vulnerabilities that have a high likelihood but rather a low or medium likelihood. In contrast, Adobe Flash Player, Microsoft Internet Explorer and particularly Adobe AIR are more strongly associated with threats of high probability, with odds ratios of 2.42, 1.38 and 3.48, respectively.

TABLE 13. Justified and high likelihood odds ratios for most frequently affected products.

Product	Justified	High Likelihood
MS Windows	1.69	0.54
Adobe Flash Player	**2.58**	2.42
MS Internet Explorer	1.69	1.38
MS Office	1.69	0.72
Adobe AIR	**2.58**	**3.48**

The third comparative risk analysis, based on Query 6.3, investigates the odds ratios of High likelihood vulnerabilities per update justification. The results in Table 14 show that vulnerabilities with updates without changes have the highest odds ratio of 1.20, suggesting a stronger association with a high likelihood of occurrence. This is followed by unjustified (1.07) and justified updates (1.01), both of which show a modest association. Vulnerabilities with unimportant updates exhibit the lowest odds ratio of 0.79, indicating a lower risk of high likelihood.

The chi-squared tests for the three comparative risk analyses reveal no statistically significant relationships. The analysis of Justified updates per product yields a chi-squared statistic of 27.14 with 42 degrees of freedom and a p-value of 0.96. In contrast, the analysis of High likelihood per product shows a chi-squared statistic of 11.29, 42 degrees of freedom, and a p-value of 0.99. For the odds ratio of High likelihood per update justification, the chi-squared statistic is 0.41 with 3 degrees of freedom and a p-value of 0.94. All p-values exceed the 0.05 threshold, indicating that the relationships among products, update justifications, and likelihood classifications are not statistically significant, limiting reliable conclusions about risk factors in this context.

TABLE 14. Odds ratios for HIGH likelihood per update justifications.

Odds Ratios	Justification
No Change	1.20
Unjustified	1.07
Justified	1.01
Unimportant	0.79

Key Finding: Some products are more likely to be associated with justified updates and high likelihood vulnerabilities, but none of the risk analyses is statistically significant.

8. Discussion

8.1. Interpretation of Key Findings

The likelihood classification demonstrates an inverse relationship with the timing differences in publication. Vulnerabilities identified with a higher likelihood are published more quickly, reflecting the urgency of these threats. However, this increased urgency results in greater variability in reporting times, likely due to the complexities that require more thorough assessments. Consequently, while timely reporting is essential to address imminent threats, the inherent complexities may lead to fluctuations in publication timing.

Many affected products and operating systems are unique to the NCSC dataset, pointing to a wider coverage of NCSC vulnerabilities compared to the APT.

A considerable amount of vulnerability updates from the NCSC show no changes in the description and likelihood, implying that many

of the updates may not introduce significant new information. The low rate of justified updates, where both the description and likelihood are modified, suggests limited novel insights on vulnerabilities over time. However, when updates are classified as justified, they are always associated with high-likelihood threats, reflecting the dynamic nature of vulnerability assessment.

Finally, while the data suggest that certain products are more frequently associated with justified updates and high-likelihood vulnerabilities, the lack of statistical significance limits the ability to claim these findings. The increased risk observed for certain products could be due to random variation rather than to the inherent risk factors. More research is required to draw more reliable conclusions about the risks associated with specific products or updates.

9. Threats to Validity

9.0.1. Missing Likelihood Data. A key validity threat originates from missing likelihood data in the initial NCSC dataset, which lacked the likelihood classification sub-factors. This absence limited the analysis of key factors impacting threat probabilities. While a web scraper was intended to retrieve this data, only the most recent update's likelihood descriptions were accessible, obstructing prior version retrieval.

9.0.2. Manual Standardization. Manual standardization of product names, operating systems, and version numbers poses another validity threat. Although a mapping dictionary was developed, diverse formatting and complex versioning made this challenging. Variations like "*and earlier versions*" were inconsistently handled, and while `ChatGPT` assisted in the mapping, manual review introduced potential human error, impacting dataset accuracy.

9.0.3. URREF Ontology Usage. Extending the URREF ontology represents a validity concern. Although its reusability and community support made it a logical choice, it did not fully cover the specific domain of this study, potentially missing nuances in the data. A custom ontology could provide more precise control over structure and content, leading to different outcomes.

9.0.4. Small Dataset. The small dataset of 86 common vulnerabilities between NCSC and APT limits the ability to draw meaningful conclusions, weakening comparative risk analysis and ontological evaluations. A larger dataset could provide deeper insight into vulnerability connections and improve result reliability.

10. Conclusion

This research aimed to investigate the alignment and consistency of threat intelligence data between the NCSC and APT datasets, focusing on representing uncertainty through two main research questions. The key findings demonstrate a strong alignment between the APT reserved and exploitation times and NCSC publications, although gaps in data completeness and consistency remain.

Justified updates, which include changes to both the description and likelihood of a vulnerability, show greater variability. Additionally, while high-likelihood threats are reported more quickly, this urgency often results in increased variability in publication times.

The ontological analysis found that although both datasets cover some common technical details for most vulnerabilities, significant differences remain in affected products, operating systems, and version details. Although both datasets prioritize high-likelihood threats, the lack of statistical significance in some areas limits firm conclusions.

Several limitations may have affected the study results, including missing likelihood data, difficulties in manually standardizing technical details, and the relatively small number of shared vulnerabilities, which limit broader conclusions.

References

[1] W. Tounsi and H. Rais, "A survey on technical threat intelligence in the age of sophisticated cyber attacks," *Computers & security*, vol. 72, pp. 212–233, 2018.

[2] S. de Smale, R. van Dijk, X. Bouwman, J. van der Ham, and M. van Eeten, "No one drinks from the firehose: How organizations filter and prioritize vulnerability information," in *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 1980–1996, IEEE, 2023.

[3] A. D. Jenkins, L. Liu, M. K. Wolters, and K. Vaniea, "Not as easy as just update: Survey of system administrators and patching behaviours," in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pp. 1–17, 2024.

[4] S. R. B. Mohd Kassim, S. Li, and B. Arief, "Understanding how national csirts evaluate cyber incident response tools and data: Findings from focus group discussions," *Digital Threats: Research and Practice*, vol. 4, no. 3, pp. 1–24, 2023.

[5] D. Schlette, P. Empl, M. Caselli, T. Schreck, and G. Pernul, "Do you play it by the books? a study on incident response playbooks and influencing factors," in *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 3625–3643, IEEE, 2024.

[6] S. Elder, M. R. Rahman, G. Fringer, K. Kapoor, and L. Williams, "A survey on software vulnerability exploitability assessment," *ACM Computing Surveys*, vol. 56, no. 8, pp. 1–41, 2024.

[7] J. Jacobs, S. Romanosky, B. Edwards, I. Adjerid, and M. Roytman, "Exploit prediction scoring system (epss)," *Digital Threats: Research and Practice*, vol. 2, no. 3, pp. 1–17, 2021.

[8] J. Jacobs, S. Romanosky, O. Suciu, B. Edwards, and A. Sarabi, "Enhancing vulnerability prioritization: Data-driven exploit predictions with community-driven insights," in *2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pp. 194–206, IEEE, 2023.

[9] F. Massacci, "The holy grail of vulnerability predictions," *IEEE Security & Privacy*, vol. 22, no. 1, pp. 4–6, 2024.

[10] L. Allodi and F. Massacci, "Comparing vulnerability severity and exploits using case-control studies," *ACM Transactions on Information and System Security (TISSEC)*, vol. 17, no. 1, pp. 1–20, 2014.

[11] M. Vasek, J. Wadleigh, and T. Moore, "Hacking is not random: a case-control study of webserver-compromise risk," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 2, pp. 206–219, 2015.

[12] S. Tajalizadehkhoob, R. Böhme, C. Ganán, M. Korczyński, and M. V. Eeten, "Rotten apples or bad harvest? what we are measuring when we are measuring abuse," *ACM Transactions on Internet Technology (TOIT)*, vol. 18, no. 4, pp. 1–25, 2018.

[13] G. Di Tizio, M. Armellini, and F. Massacci, "Software updates strategies: A quantitative evaluation against advanced persistent threats," *IEEE Transactions on Software Engineering*, vol. 49, no. 3, pp. 1359–1373, 2022.

[14] F. Massacci and G. Di Tizio, "Are software updates useless against advanced persistent threats?," *Communications of the ACM*, vol. 66, no. 1, pp. 31–33, 2022.

[15] S. Lipner and J. Pescatore, "Updates, threats, and risk management," *Communications of the ACM*, vol. 66, no. 5, pp. 21–23, 2023.

[16] F. Minna and F. Massacci, "Sok: Run-time security for cloud microservices. are we there yet?," *Computers & Security*, vol. 127, p. 103119, Apr 2023.

[17] D. Schatz, R. Bashroush, and J. Wall, "Towards a more representative definition of cyber security," *The Journal of Digital Forensics, Security and Law*, 2017.

[18] E. Casey, S. Barnum, R. Griffith, J. Snyder, H. van Beek, and A. Nelson, "Advancing coordinated cyber-investigations and tool interoperability using a community developed specification language," *Digital Investigation*, vol. 22, p. 14–45, 2017.

[19] V. Dragos, J. Ziegler, J. de Villiers, A. de Waal, A.-L. Jousselme, and E. Blasch, "Entropy-based metrics for urref criteria to assess uncertainty in bayesian networks for cyber threat detection," in *2019 22th International Conference on Information Fusion (FUSION)*, p. 1–8, Jul 2019.

[20] V. Dragos, J. Ziegler, and J. Pieter De Villiers, "Application of urref criteria to assess knowledge representation in cyber threat models," in *2018 21st International Conference on Information Fusion (FUSION)*, p. 664–671, Jul 2018.

[21] C. C. Insaurralde, E. P. Blasch, P. C. Costa, and K. Sampigethaya, "Uncertainty-driven ontology for decision support system in air transport," *Electronics*, vol. 11, no. 3, p. 362, 2022.

[22] C. C. Insaurralde, P. C. Costa, E. Blasch, and K. Sampigethaya, "Uncertainty considerations for ontological decision-making support in avionics analytics," in *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, p. 1–9, IEEE, 2018.

[23] C. Laudy and V. Dragos, "Use cases for social data analysis with urref criteria," in *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, p. 1–8, IEEE, 2020.

[24] K. Vasnier, A. I. Mouaddib, S. Gatepaille, and S. Brunessaux, "Multi-level information fusion and active perception framework: towards a military application," *Procedings of NATO SET-262 RSM on Artificial Intelligence for Military Multisensor Fusion Engines*, 2018.

[25] J. Freudiger, S. Rane, A. Brito, and E. Uzun, "Privacy preserving data quality assessment for high-fidelity data sharing," vol. 2014-November, p. 21–29, 2014.

[26] P. Kuehn, M. Bayer, M. Wendelborn, and C. Reuter, "Ovana: An approach to analyze and improve the information quality of vulnerability databases," in *Proceedings of the 16th International Conference on Availability, Reliability and Security*, ARES 21, (New York, NY, USA), p. 1–11, Association for Computing Machinery, Aug 2021.

[27] G. Gross, R. Nagi, and K. Sambhoos, "Situation assessment: Uncertainty representation in inexact graph matching," in *IIE Annual Conference. Proceedings*, p. 1, Institute of Industrial and Systems Engineers (IISE), 2010.

[28] G. Gross, R. Nagi, and K. Sambhoos, "Soft information, dirty graphs and uncertainty representation/processing for situation understanding," in *2010 13th International Conference on Information Fusion*, p. 1–8, IEEE, 2010.

[29] H. Liu, C. Zhong, A. Alnusair, and S. Islam, "Faixid: A framework for enhancing ai explainability of intrusion detection results using data cleaning techniques," *Journal of Network and Systems Management*, vol. 29, no. 4, 2021.

[30] B. E. Mullins, *The Integration of Artificial Intelligence Techniques to Improve the Effectiveness of Electronic Countermeasure Strategies in a Tactical Environment*. 1987.

[31] M. Oredsson, *Bridging the gap between information security risk assessments and enterprise risk management*. Master's thesis, University of Stavanger, Norway, 2018.

[32] T. Schaberreiter, V. Kupfersberger, K. Rantos, A. Spyros, A. Papanikolaou, C. Ilioudis, and G. Quirchmayr, "A quantitative evaluation of trust in the quality of cyber threat intelligence sources," in *Proceedings of the 14th International Conference on Availability, Reliability and Security*, ARES '19, (New York, NY, USA), p. 1–10, Association for Computing Machinery, Aug 2019.

[33] L. Sikos, "Handling uncertainty and vagueness in network knowledge representation for cyberthreat intelligence," vol. 2018-July, 2018.

[34] J. Cha, S. K. Singh, Y. Pan, and J. H. Park, "Blockchain-based cyber threat intelligence system architecture for sustainable computing," *Sustainability*, vol. 12, no. 16, p. 6401, 2020.

[35] M. Guarascio, N. Cassavia, F. S. Pisani, and G. Manco, "Boosting cyber-threat intelligence via collaborative intrusion detection," *Future Generation Computer Systems*, vol. 135, p. 30–43, Oct 2022.

[36] G. Di Tizio, M. Armellini, and F. Massacci, "Software updates strategies: a quantitative evaluation against advanced persistent threats," *IEEE Transactions on Software Engineering*, vol. 49, p. 1359–1373, Mar. 2023. arXiv:2205.07759 [cs].

Appendix A.
Data Availability Statement

All tools and data used in this article are available in the following GitHub repository: `https://anonymous.4open.science/r/ThreatIntelligenceAlignment-1E7C/README.md`. The repository includes:

Class files:

- `DataLoaderSaver.py`: functions to load and save datasets.
- `DataAnalyzer.py`: functions to analyse datasets.
- `DataProcessor.py`: functions to pre-process datasets.
- `ClassificationEngineering.py`: functions to create new classification columns.
- `URREFHelper.py`: functions to populate the URREF ontology.
- `ChartCreator.py`: functions to create different result charts.
- `CompAnalyzer.py`: functions to perform the comparative risk analyses.

Pipeline files:

- `exploration_NCSC.py`: exploratory data analysis of the NCSC dataset.
- `exploration_APT.py`: exploratory data analysis of the APT dataset.
- `processing_NCSC.py`: pre-processing of the NCSC dataset.
- `processing_APT.py`: pre-processing of the APT dataset.
- `feature_engineering_NCSC.py`: engineering of new attributes to the NCSC dataset.
- `classification_engineering.py`: engineering of a new dataframe including four classification columns for the update justification in NCSC descriptions.
- `RQ1.py`: temporal analysis of threats.
- `RQ2.py`: ontological analysis of threats.
- `RQ2_charts.py`: developing charts for ontological analysis.
- `RQ2_comp_analysis.py`: performing comparative risk analyses.

Input files:

- `NCSC_advisories_initial.csv`: initial NCSC dataset.
- `URREF.owl`: original URREF ontology.

Directories:

- `SPARQL Queries`: SPARQL queries for ontological analysis, including results for comparative risk analyses.
- `Extra Files`: excluded from this article.

Appendix B.
Replication Guide

B.1. Literature Review

Initially, three keyword-based queries are executed to retrieve relevant sources. Keyword searches are listed in Table 15, along with the total results and the number of relevant sources that are ultimately selected. Due to the large number of sources from Google Scholar, only the sources displayed on the first ten pages of each query are included, facilitated by the Google Scholar search result method, which prioritizes relevance[7]. A refined keyword search, found at the bottom of Table 15, was applied later to focus more on the representation of uncertainty. The sources retrieved from this refined search are screened similarly based on titles, abstracts, and selection criteria.

Keyword Search	Google Scholar		Scopus		ACM DL	
[*threat intelligence* OR *security intelligence* OR *cyber intelligence*] AND [*AI* OR *ML**] AND [*representation* OR *encoding*] AND [***uncertainty* OR *bias* OR *error***]	3,940	2	237	2	97	1
[*threat intelligence* OR *security intelligence* OR *cyber intelligence*] AND [*AI* OR *ML**] AND [*representation* OR *encoding*] AND [***fusion* OR *aggregation* OR *merge***]	2,240	4	242	4	71	0
[*threat intelligence* OR *security intelligence* OR *cyber intelligence*] AND [*AI* OR *ML**] AND [*representation* OR *encoding*] AND [***predictive policing* OR *prediction***]	2,510	0	376	3	78	1
[*threat intelligence* OR *security intelligence* OR *cyber intelligence*] AND [***uncertainty representation* OR *representing uncertainty***]	38	7	6	4	1	0

*[*AI* OR *ML*] and [*Artificial Intelligence* OR *Machine Learning*] used

TABLE 15. Total results found (left) and total sources selected (right) per keyword-based search and per digital library, including duplicates.

B.2. NCSC Data Pre-processing

B.2.1. Exploratory Data Analysis. The initial step is to understand the dataset by analyzing all variables, including the number of rows and columns, values, datatypes, duplicates, and missing values, which are reported in Table 2.

The attribute names were inconsistent with the terminology used on the NCSC website. Therefore, a data description file was generated to help understand all 15 variables. It was discovered that a potentially significant attribute, the likelihood description (`Kansomschrijving`), was absent. This attribute directly influences the likelihood of a threat occurring. However, `Kansomschrijving` is not a primary focus of this investigation, as the emphasis lies on assessing the likelihood and description of each threat. With the exception of the 95 missing values mentioned previously, all other data points are intact.

B.2.2. Data Selection. In the initial stages, only essential data are retained to minimize noise. This process involves removing duplicate entries and excluding irrelevant attributes. The column `NCSC inschaling` is eliminated because it aggregates the likelihood and impact indicators into a single string. Separating this information into distinct variables enhances its utility. These components are already present in the dataset, making `NCSC inschaling` redundant.

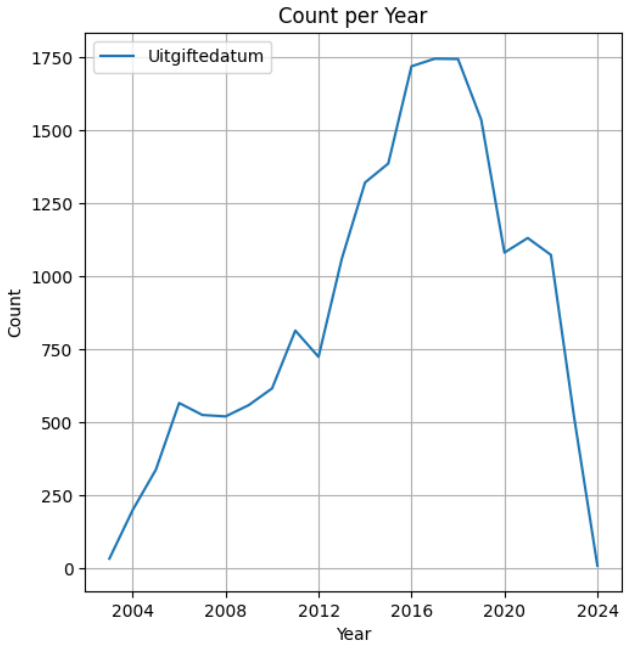
Furthermore, the column `Versie` is removed due to inconsistency. Some advisories use the notation 1.01 for the first updated version of a threat, while others denote subsequent publications as 2.00. Instead, the unique `NCSC ID` combined with `Uitgiftedatum` could effectively represent this information.

Other attributes removed from the NCSC dataset include `Titel`, `Schade`, `Schadeomschrijving`, and `Mogelijke oplossingen` as they are not relevant to this research.

The study analyses changes in likelihood classification, necessitating each threat to have at least two entries. Instances of NCSC IDs appearing only once are removed, since they fall outside the scope of this investigation. The line plot in Figure 11 illustrates that this removal does not create data gaps. It shows that even after filtering out single-entry vulnerabilities, a substantial number of instances remain consistent throughout all years in the dataset, particularly between 2015 and 2017, where the

7. Google Scholar Search Help: https://scholar.google.com/intl/en/scholar/help.html

majority of advisories are concentrated. This suggests that the removal of single-entry threats does not compromise the temporal coverage of the data set or the validity of the analysis. To develop the line plot, the data type of the column `Uitgiftedatum` was converted from a string object to a datetime format. After the data selection process, the dataset comprises a total of $19,215$ instances and 13 attributes.



Vulnerabilities that include updates are published consistently between 2002 and 2023, with a peak in threat advisories occurring from 2015 to 2017.

Figure 11. Line plot of `Uitgiftedatum`.

B.2.3. Data Cleaning. The next step is data cleaning, which involves identifying and correcting errors and inconsistencies to enhance the quality and reliability of the data. In addition to converting the `Uitgiftedatum`, the data types of the attributes `CVE-ID`, `Toepassingen`, `Versies` and `Platformen` are rectified by converting their string representation of lists into actual lists of strings.

During this process, missing data points were represented by the string *[]*. Additionally, instances of *['niet beschikbaar']* and *['-']* frequently appeared in the `CVE-ID` attribute. These instances are cleaned by imputing a $None$ value. Further inconsistencies were found among the CVE-IDs. After cleaning these instances by removing extraneous punctuation and spaces, the CVE-IDs were validated against the correct format. A CVE-ID is correctly formatted if it is a string that starts with three letters, followed by a hyphen, and then contains only digits and hyphens, with a total length of either 13 or 14 characters. Instances with incorrectly formatted CVE-IDs are removed from the dataset. If the list of the CVE-IDs becomes empty, this consequently results in new missing values. Ultimately, the `CVE-ID` attribute contains $1,025$ missing values. These instances are subsequently deleted from the dataset, as the `CVE-ID` is essential to serve as the unique common identifier with the APT reference dataset. The Common Vulnerabilities and Exposures (CVE)[8] program provides a reference method for publicly known cyber security vulnerabilities and exposures. The CVE-IDs are listed on MITRE, the American National Vulnerability Database (NVD) and the Dutch NCSC platform.

Another important task is the normalization of `Toepassingen`, `Versies` and `Platformen` for consistency and effective analysis. A mapping dictionary was created to standardise product names by merging various entries into a unified format. For example, names like *"Adobe Acrobat XI<11.0.03"* and *"Adobe Acrobat X<10.1.7"* are mapped to *"Adobe Acrobat"*. Operating systems follow a similar mapping process to ensure uniformity. Variations such as *"Apple Mac"*, *"Apple Mac OS"*, *"Mac OSX"* are all normalized to *"Apple Mac"*. For normalization of the versions, a function is implemented to

8. CVE: https://www.cve.org/

find the first digits before the dot. For example, entries such as *"15.7.2"*, *"15.1."* and *"15.5.9."* are transformed into *"15"*. If extraction was unsuccessful, the new entry is labeled as *"Unrecovered"*.

B.2.4. Outliers & Missing Values. The subsequent pre-processing step usually involves outlier detection and missing value imputation to improve data accuracy. However, since the research focuses on identifying uncertainty in advisory descriptions, imputing these anomalies would be counterproductive, as they may represent uncertainty in the data input itself.

B.2.5. Feature Engineering. The next phase involves leveraging the available dataset to generate new features. Previously, the attribute `Versie` was removed from the dataset due to inconsistencies. Given the importance of this information for the research, a new column, denoted as `Update`, is introduced. This attribute assigns an integer value to each update of every NCSC advisory in chronological order based on the corresponding `NCSC ID` and `Uitgiftedatum`.

B.3. APT Data Pre-processing

B.3.1. Exploratory Data Analysis. An exploratory data analysis was performed to better understand the dataset and assess its relevance to the research questions. The results are shown in Table 4.

B.3.2. Data Selection. The first step is to select relevant data from the APT dataset. The exploratory data analysis revealed that the `APT` attribute, which contains APT names based on MITRE ATT&ck, can be dropped as it falls outside the investigation's scope.

In addition, the `update` attribute is removed due to its high proportion of missing values and inconsistencies, with $62,801$ entries denoted as *"*"*. The remaining values are ambiguous, including entries like *"sp1", "gold", "beta_4", "update11_b03", "unknown"*. After these selections, the APT dataset contains $65,552$ instances and 9 attributes.

B.3.3. Data Cleaning. In the data cleaning phase, the time-related columns `exploited_time`, `reserved_time` and `published_time` are converted from a string object to a uniform datetime format. The conversion aligns the columns with the datetime format used for the `Uitgiftedatum` attribute of the NCSC, ensuring consistency between datasets.

In addition to converting the data types of the time-related attributes, data cleaning is also applied to the variables `product`, `version` and `os`. Similarly to the previously removed `update` attribute, the `version` and `os` columns contain entries marked with an asterisk (*"*"*) when version numbers or operating system specifications are unavailable. These asterisk entries, along with instances marked with a hyphen (*"-"*), are cleaned by imputing a $None$ value to ensure uniformity and accuracy throughout the dataset. Furthermore, the same normalization technique as for the NCSC dataset is applied to the `product`, `version` and `os` attributes. For example, the product entries *"ie"* and *"internet_explorer"* are standardized to *"Microsoft Internet Explorer"* using the same mapping as the NCSC.

B.3.4. Outliers & Missing Values. As with the NCSC dataset, the presence of outliers and missing values in the APT dataset could indicate uncertainty regarding threats. The previously mentioned 209 instances with missing data points could still represent some level of uncertainty. For example, these instances will have more uncertainty around the corresponding threats compared to other data in the APT dataset, but less uncertainty than instances that are completely absent from the dataset. However, data from the three available attributes are insufficient for this research, because the common identifier to connect the instance to the correct NCSC threat is missing. Imputing new values is also not viable as it would counterproductively affect the representation of uncertainty. Therefore, the 209 instances with missing values are deleted from the dataset.

B.4. Temporal Analysis

The process begins by merging the NCSC, NCSC Update Justification, and APT datasets, linked by the common unique identifier; `CVE-ID`. The data sets are expanded on the basis of this unique identifier, resulting in a merged dataset with 86 unique CVE-IDs. Time attributes are rounded to months for consistency, forming the basis for further analysis, including correlation, statistical tests, and time difference calculations.

B.4.1. Reserved Time to Initial Release. The first analysis examines the alignment between the `reserved_time` of the APT dataset and the `Uitgiftedatum` of initial NCSC releases. The analysis involves calculating the Pearson correlation and the p-value for significance. The Pearson correlation was chosen to detect linear relationships between continuous variables, such as timestamps. A strong correlation would indicate that the initial NCSC publications closely follow the reservations of the APT, suggesting timely sharing of threat information.

Additionally, the Mean Absolute Error (MAE) is calculated to measure the average time, providing a straightforward interpretation of delays in months between the initial NCSC publications and APT reservations. A lower MAE reflects better alignment and faster communication of threat information.

B.4.2. Exploited Time to Subsequent Releases. The second temporal analysis investigates the alignment between the `exploited_time` attribute from the APT dataset and the `Uitgiftedatum` of all subsequent NCSC updates. Again, the Pearson correlation and MAE are calculated. A strong correlation would indicate that subsequent updates of the NCSC are closely related to the actual exploitation events recorded in the APT dataset, suggesting responsive and timely updates following exploitation. A lower MAE would reflect prompt updates and effective communication about threat exploitation.

B.4.3. Update Justification on Timing. This analysis examines the time differences between NCSC publications and APT exploitation times using Equation 1. Positive values indicate that NCSC published after exploitation, while negative values indicate publication before APT exploitation.

$$Difference = exploited\_time - Uitgiftedatum \tag{1}$$

Using these time differences, the relationship between the justified and non-justified NCSC updates can be investigated. Justified updates, as detailed in Table 3, include updates of which the changes in the description are justified and important. Unjustified updates include updates with no changes, unjustified changes, and unimportant changes in the description.

Pearson correlation and the T-statistic are calculated to assess whether justified NCSC updates are more timely and responsive than unjustified updates, offering insight into the effectiveness of threat updates.

B.4.4. Likelihood on Timing. The final temporal analysis investigates the relationships between the three likelihood classifications (`Low`, `Medium`, and `High`) with the time difference from equation 1 between the exploitation by the APT and the publication by the NCSC.

Pearson correlation and the ANOVA F-statistic are used to quantify the strength of the relationship and to compare the variance between the likelihood classifications. Unlike the T-test, which is limited to comparing two groups, the ANOVA F-test evaluates the variance between multiple groups.

B.5. Ontological Analysis

B.5.1. URREF Ontology. To use the Resource Description Framework (RDF) schema for ontology description, the URREF ontology is converted from OWL/XML to RDF/XML syntax using the open-source editor `Protégé` for semantic interoperability[9]. The RDF schema outlines the structure of the URREF ontology, connecting various classes and properties using triples (subject, predicate, object) to represent the complex relationships within threat intelligence data. Next, we develop the classes and properties that represent the core concepts of the threat intelligence data. The classes structure the ontology, and the properties reflect the relationships between them.

B.5.2. Ontology Population. The ontology is populated by incrementally processing the merged NCSC and APT dataset. Each vulnerability is assigned a unique Uniform Resource Identifier (URI) based on its CVE-ID, ensuring distinct identification.

Instances are linked to dataset-specific URIs (NCSC or APT), and vulnerabilities are classified by likelihood (`Low`, `Medium`, or `High`). Additionally, the justification for likelihood changes, classified as `Justified`, `No Change`, `Unjustified`, and `Unimportant`, is also recorded for each vulnerability. In addition, the technical attributes (product, version, OS) are represented as instances of their respective classes, linked to the vulnerabilities to provide full technical context. Finally, the

9. RDF Schema: https://www.w3.org/TR/rdf-schema/

TABLE 16. Ontology classes and their descriptions.

Classes	Description
`Threat Intelligence`	Serves as the top-level class, containing all aspects of threat intelligence data.
`Vulnerability`	Represents individual vulnerabilities identified using the CVE-IDs employed.
`Product`	Captures the specific software or hardware product linked to the identified vulnerability.
`Version`	Captures the particular version number of the product affected by the vulnerability.
`OS`	Captures the operating system on which the affected product version is capable of running.
`Dataset`	Distinguishes between the sources of data, namely the NCSC and APT datasets.
`Likelihood`	Classifies the probability of the vulnerability being exploited, categorized as `Low`, `Medium`, or `High`.
`Justification`	Represents whether a change in likelihood is justified or not by an update of the vulnerability's description.

TABLE 17. Ontology properties and their descriptions.

Properties	Description
`hasCVEID`	Links a vulnerability to its unique identifier (CVE-ID).
`fromDataset`	Indicates the dataset from which a particular instance originates.
`affectsProduct`	Describes the relationships between a vulnerability and the affected product.
`affectsVersion`	Describes the relationships between the affected product and the affected version of the vulnerability.
`runsOn`	Describes the relationships between the affected version and the operating system of the vulnerability.
`hasLikelihood`	Connects vulnerabilities to their assessed likelihood classification.
`hasDescriptionChange`	Captures whether the update in the description of the vulnerability is justified by a change in likelihood.

ontology is serialized and stored in RDF/XML format, enabling structured analysis through SPARQL queries.

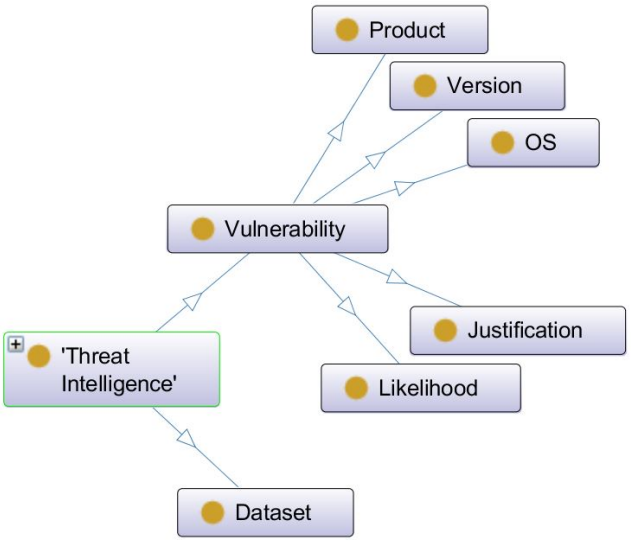


Figure 12. Ontology graph showing the class hierarchy.

B.5.3. SPARQL Queries. All queries used in this analysis are available in the GitHub repository. The experiments are structured in six sub-analyses.

Vulnerability Analysis. Compares unique vulnerabilities across both datasets. The SPARQL queries focus on the alignment and divergence of vulnerabilities related to affected products, operating systems, and versions.

1.1 Vulnerabilities with **any** product in common between NCSC and APT.
1.2 Vulnerabilities with **all** products identical for NCSC and APT.
1.3 Vulnerabilities with **no** NCSC products present in APT.
1.4 Vulnerabilities with **no** APT products present in NCSC.
1.5 Vulnerabilities with **all** products & **any** operating system in common between NCSC and APT.
1.6 Vulnerabilities with **all** products & **all** operating systems identical for NCSC and APT.
1.7 Vulnerabilities with **all** products & **any** version in common between NCSC and APT.

1.8 Vulnerabilities with **all** products & **all** versions identical for NCSC and APT.

Product Analysis. Investigates similarities and differences of the affected products in the NCSC and APT datasets, assessing the alignment of their product focus.

2.1 Products NCSC and APT share the most.
2.2 Products the most in NCSC.
2.3 Products the most in APT.
2.4 Products exclusively in NCSC.
2.5 Products exclusively in APT.

Operating System Analysis. Examines the alignment and divergence of operating systems related to vulnerabilities in both datasets.

3.1 Operating systems NCSC and APT share the most.
3.2 Operating systems the most in NCSC.
3.3 Operating systems the most in APT.
3.4 Operating systems exclusively in NCSC.
3.5 Operating systems exclusively in APT.

Update Justification Analysis. Focuses on the distribution of updates by justification type, such as justified or unjustified, and their relationship to the frequently shared products.

4.1 Vulnerabilities per justification; `Justified`, `Unjustified`, `Unimportant`, `No change`.
4.2 Vulnerabilities with more than one justification.
4.3 Vulnerabilities per justification for products most shared between NCSC and APT.

Likelihood Analysis. Categorises vulnerabilities by likelihood, and identifies trends with respect to the affected products and update justifications.

5.1 Vulnerabilities per likelihood classification; `Low`, `Medium`, `High`.
5.2 Vulnerabilities with more than one likelihood.
5.3 Vulnerabilities per likelihood for products most shared between NCSC and APT.
5.4 Vulnerabilities with high likelihood per update justification.

Comparative Risk Analysis. Investigates relationships between the products most frequently affected, likelihood classifications, and update justifications to identify risk factors by means of odds ratio reductions and chi-squared tests.

6.1 `Justified` updates for products most shared between NCSC and APT.
6.2 `High` likelihood for products most shared between NCSC and APT.
6.3 `High` likelihood per update justification.