Using an SBOM to Mitigate a Lemons Market

Peter J. Caven
Luddy School of Informatics, Computing, and Engineering
Indiana University
Bloomington, Indiana, United States
pcaven@iu.edu

Xinyao Ma
Luddy School of Informatics, Computing, and Engineering
Indiana University
Bloomington, Indiana, United States
maxiny@iu.edu

Vafa Andalibi
Luddy School of Informatics, Computing, and Engineering
Indiana University
Bloomington, Indiana, United States
vafandal@iu.edu

L. Jean Camp
Luddy School of Informatics, Computing, and Engineering
Indiana University
Bloomington, Indiana, United States
ljcamp@iu.edu

***Abstract*—The Software Bill of Materials (SBOM) has emerged as a possible tool to mitigate information asymmetry within the security market. By promoting transparency throughout the supply chain, stakeholders now have crucial information that can support decisions throughout a product's lifecycle. This pre and post-procurement decision support aligns with the evolving cybersecurity paradigm and supports well-established economic models. Our research identifies the need for more effective communication within the current security market. While SBOMs may present an effective and viable option, their current instantiation is not suitable for all consumers. We explore how SBOMs can be made more usable. This paper seeks to draw from our research to discuss the economic benefits of communicating security. Particularly, we focus on how both visualizing SBOMs and integrating SBOM information into labels can increase transparency, which increases consumers' willingness to pay.**[1]

***Index Terms*—Security, Labels, SBOM, Permissions, Secure Supply Chain**

I. Introduction

The concept of a *lemon's market* was introduced in 1970 to describe the U.S. second-hand car market [2]. The term *lemon* refers to a used vehicle with hidden issues or defects that may not be readily apparent to a buyer. Unscrupulous sellers may attempt to mask underlying issues by investing in cheaper superficial improvements (e.g., new exterior paint or interior detailing) [16].

A lemon market can be characterized by three key factors. First, information asymmetry exists, wherein the seller holds more information about the quality of the goods than the buyer. Second, the buyer rationally assumes that the goods offered for sale are of inferior quality since the seller has not sufficiently proved the quality. Finally, the development and sale of high-quality goods become financially impractical, as there are no reliable means for a buyer to assure quality [2], [16], [28], [31]. Essentially low and high-quality goods become indistinguishable. This results in a situation where buyers are unwilling to pay a premium for a potentially higher-quality good, fearing it may end up being low quality. This threat of inconsistent quality leads to a lack of consumer confidence and decreased demand across the entire market.

A lemons market is normally a two-sided market, where one person is selling a good and one person is buying a good. However, technology creates more complicated markets [23]. For example, the mobile app market is more intricate due to the existence of three major stakeholders: the developer, the buyer, and the marketplace competition (often in the form of a duopoly) [49]. Mobile app marketplaces need to simultaneously cater to the needs of developers, while at the same time instilling trust in buyers. More explicitly, a marketplace must appeal to developers, since app availability increases a smartphone's usability and functionality, which drives hardware purchases; simultaneously, the marketplace must maintain buyers' trust in the hosted applications to ensure continued marketplace utilization. This leads to the need for clear benefit communication [49]. Without it, consumers may lack confidence, especially when there are disparate ratings across multiple marketplaces (i.e., one has only positive reviews and the other has negative reviews).

Information asymmetry is the result of a communications problem. In this paper, we will use experiments designed to peel back the layers of the lemon market to foster better communication. We focus on how visualizing SBOMs and integrating SBOMs into labels can both be used to increase transparency and mitigate the lemons market in terms of security and privacy. In doing so, we demonstrate that while SBOMs may be complex, they hold the key to providing a more safe and secure technology market.

[1] This research was supported in part by CTIA and the Comcast Innovation Fund. We acknowledge support from the US Department of Defense [Contract No. W52P1J2093009]. This material is based upon work supported by the U.S. Department of Homeland Security under Grant Award Number 17STQAC00001-07-00. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security, Comcast, CTIA, Indiana University, nor the U.S. Department of Defense.

II. Background & Related Works

A challenge arises in how privacy and security risks are communicated to users. Various measures, such as runtime permissions, manifest presentations, Apple tracking transparency, and Apple privacy labels have been implemented to address this issue [6], [29], [34], [45]. The core argument is that providing more information to users enables them to make better decisions. When users have access to better risk communication, they can demand higher security and privacy, which increases their willingness to pay for secure products [12], [15], [22]. This creates an incentive for developers to develop better products. Without this signal, developers with superior products may never enter the marketplace, leading to a decrease in overall quality and the persistence of a security lemons market [2].

A. Risk Communication

To improve risk communication and change market behavior, we can start with some assumptions from the early days of risk communication – show users they are choosing a risky option and that it is better not to take risks. But if users have no choice (e.g., they have to share phone contacts in order for the application to work), they will acquiesce. Therefore, we need to create partnerships to encourage developers to request fewer permissions, thus building long-term trust in marketplaces. Additionally, it empowers users to select the most secure and privacy-preserving apps, in turn decreasing information exfiltration. This is, of course, the goal of risk communication.

Risk communication also tells us that security should be a gain. We can leverage prospect theory, which suggests that people prefer gains over the probability of loss [27]. Currently, all users see are gains. They are making decisions on marginal gains and losses, not the final outcome. No one is choosing to be vulnerable allowing hackers access to their data. However, they are making incremental gain and benefit choices that lead to this outcome. Therefore, we should present security and privacy as gains in our risk communication strategy. We need to communicate more effectively using models that are shown to work in benefit communication and apply them to risk communication.

Studies have shown that offering phone owners better choice points for permissions, and ways to prioritize higher-quality permissions, can lead to more informed decisions [5], [32], [34], [45]. This technical problem can be addressed through an economics-based solution. Currently, decisions are solely based on benefits, and developers have no incentive to have correct privileges or protect data. There is no cost for being risk-maximizing. By using simple indicators to communicate risks and benefits at the moment of decision, customers are more informed [5], [20]. Our underlying assumption is that app purchases drive developers' engineering choices. To achieve better risk communication, we must provide clear information about risks and benefits, empowering users to make secure decisions and prioritize their privacy.

B. Security Labels

To address these challenges, the National Institute of Standards and Technology (NIST) has proposed the use of security labels [37], [38]. In 2023, the Federal Communications Commission (FCC) revealed the U.S. Cyber Trust Mark, stemming from NIST's security labeling recommendations [18], [52]. On March 14, 2024, the FCC voted to use this label for wireless consumer Internet of Things (IoT) products [19].

Again, the same multiple decision-makers are playing a role in this process (i.e., developers, buyers, and marketplaces). And again, developers will not invest in creating secure, less privileged products if buyers do not prioritize security. If we want buyers to care more about security, labels and risk communication must be attention-grabbing, easy to comprehend, and aligned with users' mental models [6], [9]. Presenting this information at the time of decision-making is crucial.

A good example is the United States' *Smoking Kill* label, which is clear and straightforward, effectively conveying the risk without requiring extensive understanding of medical harms [21]. There are similar labels in Australia, which are even more effective by using graphic imagery [24], [50]. All these labels serve as a warning, conveying the message that smoking leads to death without the need for detailed or convoluted medical jargon. However, the field of computer science still struggles to achieve such clear and urgent risk communication; it expects users to understand technical terms or implicit impacts associated with risk.

C. Software Bill of Materials

Complementary to NIST's labeling effort, the National Telecommunications and Information Administration (NTIA) is working on a software listing called a Software Bill of Materials (SBOM). This has been described as an ingredient list for systems. NTIA defines an SBOM as a nested inventory of all the components, information, and supply chain relationships that contribute to a piece of software [41].

Bills of materials have long been standard practice in manufacturing environments to identify all materials used in the manufacture of a product. Similarly, an SBOM enables secure use by identifying all software components; thus, it can be used to trace vulnerabilities embedded in complex code packages [51]. The minimum level of information creates interoperability and allows for the traceability of components through a product's supply chain. While an SBOM is important, by itself it is not sufficient to create a market for safe, secure software and operations.

D. The SBOM Lifecycle

The SBOM lifecycle occurs over four cyclic phases: software evaluation, generation, operations, and verification. In the generation of an SBOM, we want to identify the components, represent them correctly, and prune them so that components that are of no concern are removed. For verification of the SBOM, we might do static analysis, configuration file analysis, real-time runtime analysis; we will want some cryptographic attestations of these. For using SBOMs in operation, we integrate it with our threat modeling, map vulnerabilities to service, and identify mitigations. Then we update the code and do software evaluation, where we map against current SBOMs, identify differentiations, look at how the dependencies occur for different services, and evaluate customer-specific use-cases.

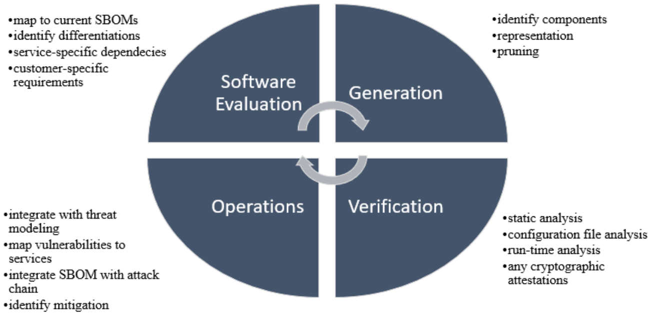

•map to current SBOMs
•identify differentiations
•service-specific dependecies
•customer-specific requirements
Software Evaluation
Generation
•identify components
•representation
•pruning
Operations
Verification
•integrate with threat modeling
•map vulnerabilities to services
•integrate SBOM with attack chain
•identify mitigation
•static analysis
•configuration file analysis
•run-time analysis
•any cryptographic attestations

Fig. 1: The four phases of the SBOM lifecycle.

In theory, this idea of a well-defined software lifecycle, consisting of requirements, planning, software design, testing, and release, is sound. In practice, software development and deployment can be more chaotic and iterative than this simplified model suggests. It requires constant vigilance to identify vulnerabilities and maintain a secure codebase throughout the software's lifecycle. Especially when we discover something wrong started in the requirements phase and we have to redevelop it, iteratively updating the code as we learn more. How then do we support users and developers in this constant feedback, nonlinear lifecycle? In this paper, we conduct experiments to identify ways of communicating these vulnerabilities. Applying the lessons learned can make unfriendly SBOMs more friendly to a variety of stakeholders.

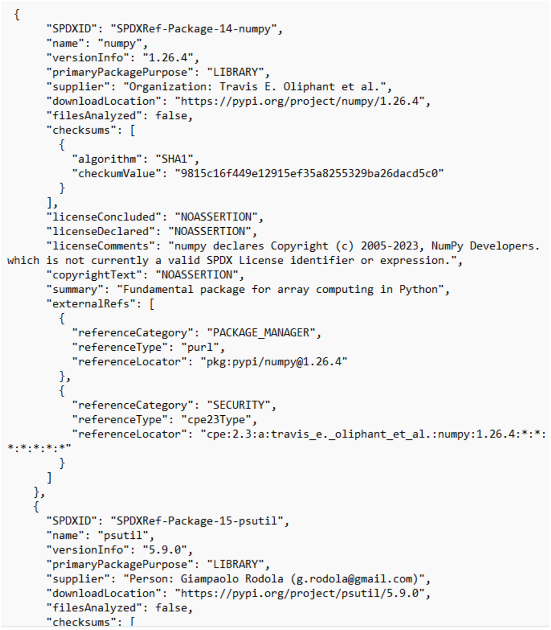
However, we need to maintain some level of complexity to have value for technical stakeholders. For example, security labels only serve the least expert and do not address operational concerns. They are inadequate for experts needing technical information. SBOMs address this requirement gap, though their current instantiation is not designed to be integrated into labeling efforts. Another consideration is SBOM's consistent update and validation, providing a more accurate representation of a product's current security posture. A static label, on the other hand, represents the validation within a moment in time, which may not accurately reflect the reality of a dynamically evolving security ecosystem.

E. Vulnerability Disclosures

The significance of patching can be traced back to events like the Therac-25, a medical linear accelerator used for cancer radiation therapy. Eleven machines were installed, and six people were seriously injured, with numerous other complaints about the device. Still, Therac did not fix it until the U.S. Food and Drug Administration (FDA) forced them to do so [30]. The lack of timely resolution resulted in severe injuries to patients, highlighting the critical need for efficient vulnerability disclosure and response processes.

While often a vendor will acknowledge and quickly patch a vulnerability, other times they may take months to respond, and sometimes outright deny the existence of said vulnerabilities. Still, vulnerability reporting and 3rd party analysis create tremendous value in the ecosystem [26]. However, the scale of vulnerabilities has surpassed human evaluation capacity. In 2020, the National Vulnerability Database identified 18,349 vulnerabilities; in 2023, it had grown to 28,819 a year [39]. It is no longer possible for humans to adequately evaluate all these issues, and in 2024, NIST paused its enrichment efforts of Common Vulnerabilities Exposures (CVEs) [40]. This highlights the importance of automation in assessing, framing, and prioritizing vulnerabilities.

The discovery of a vulnerability is often challenging, but once it is identified, applying that vulnerability and determining the scope and impact on systems is a bigger challenge. And when we introduce bugs into this ecosystem, the complexity increases again. The SBOM is critical in managing increasingly complex software ecosystems [51]. And it is a critical tool for future automation efforts. Currently, SBOMs are represented by a machine-readable file structure such as JavaScript Object Notation (JSON). This allows machines to quickly parse and build nested inventories so that when a vulnerability is identified it can be properly managed. SBOMs can be used to determine when to invest based on the status, technical impact, level of access, and likelihood of an attack.



```
 {
      "SPDXID": "SPDXRef-Package-14-numpy",
      "name": "numpy",
      "versionInfo": "1.26.4",
      "primaryPackagePurpose": "LIBRARY",
      "supplier": "Organization: Travis E. Oliphant et al.",
      "downloadLocation": "https://pypi.org/project/numpy/1.26.4",
      "filesAnalyzed": false,
      "checksums": [
        {
          "algorithm": "SHA1",
          "checkumValue": "9815c16f449e12915ef35a8255329ba26dacd5c0"
        }
      ],
      "licenseConcluded": "NOASSERTION",
      "licenseDeclared": "NOASSERTION",
      "licenseComments": "numpy declares Copyright (c) 2005-2023, NumPy Developers.
which is not currently a valid SPDX License identifier or expression.",
      "copyrightText": "NOASSERTION",
      "summary": "Fundamental package for array computing in Python",
      "externalRefs": [
        {
          "referenceCategory": "PACKAGE_MANAGER",
          "referenceType": "purl",
          "referenceLocator": "pkg:pypi/numpy@1.26.4"
        },
        {
          "referenceCategory": "SECURITY",
          "referenceType": "cpe23Type",
          "referenceLocator": "cpe:2.3:a:travis_e._oliphant_et_al.:numpy:1.26.4:*:*:
*:*:*:*:*"
        }
      ]
    },
    {
      "SPDXID": "SPDXRef-Package-15-psutil",
      "name": "psutil",
      "versionInfo": "5.9.0",
      "primaryPackagePurpose": "LIBRARY",
      "supplier": "Person: Giampaolo Rodola (g.rodola@gmail.com)",
      "downloadLocation": "https://pypi.org/project/psutil/5.9.0",
      "filesAnalyzed": false,
      "checksums": [
```

Fig. 2: An example of what a few packages in an SBOM look like in JSON format.

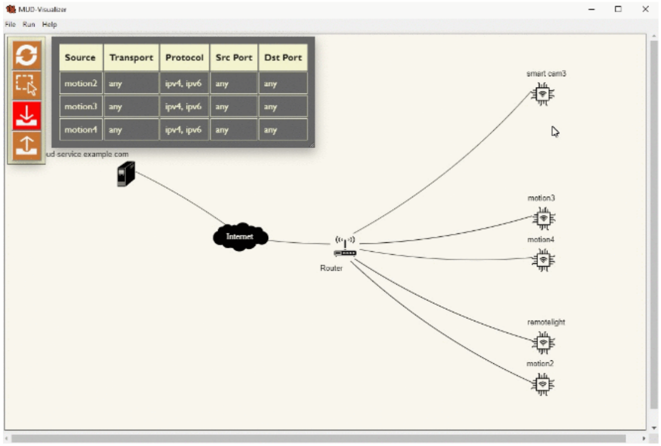
III. Method & Experiments

In this section, we discuss how SBOMs can be used to mitigate a *lemons market*. First, we draw a similarity to a visualizer we developed for Manufacturer Usage Description and how SBOM data can be represented to provide a more complete understanding of its interaction. Next, we will determine which security factors users may find the most salient within the presented SBOM. Finally, we seek to determine if consumers would be willing to pay for more secure products. These experiments demonstrate that while SBOMs in their current form may not be usable, they have significant value if they can be communicated in the right way to the right users.

A. Visualizing SBOMs

We can gain valuable insights from app permissions, software development practices, and access control through the Manufacturer Usage Description (MUD). MUD is a machine-readable access control list designed for IoT devices, ensuring secure device-specific access control without the need for customizing hardware controls. The idea is to simplify onboarding and facilitate device integration by sharing MUD files. The goal is to enable easy plug-and-play usage for all users [3].

1) Method: To assess the usability of the MUD-Visualizer, we conducted a comparison study where 52 participants were divided into two groups: a control group using standard textual MUD files and an intervention group using the MUD-Visualizer, shown in Figure 3. We use the control to evaluate the effectiveness of visualization. Both groups were tasked with answering 23 questions regarding the information presented in the MUD. For example, participants were tasked with identifying which remote servers or local devices were allowed to interact with other devices on the network based on their MUD file. Additionally, we asked basic questions related to the protocols permitted by the devices, such as IP version, port numbers, and whether TCP or UDP was used. The MUD-Visualizer has been developed to simplify information presentation, allowing increased usability and understanding of interconnected information.



Source	Transport	Protocol	Src Port	Dst Port
motion2	any	ipv4, ipv6	any	any
motion3	any	ipv4, ipv6	any	any
motion4	any	ipv4, ipv6	any	any

Fig. 3: The MUD-Visualizer can help visualize the connected nodes as well as display the traffic data.

2) Results: Using the Software Usability Scale (SUS) for both the control and intervention, we can determine the groups' usability; an aggregate score of 68 is considered to have average usability [4]. Participants in the control group scored 55.19, while those in the intervention scored 77.02. This higher score is indicative of increased usability of complex information. We used the Mann-Whitney rank sum test to determine that this difference between the groups was statically significant, $p < .001$. When analyzing the time to selection, as well as the accuracy of selection, we again see the intervention outperforming the control. The MUD-Visualizer took significantly less time to use, almost a third less time than the control. Additionally, the accuracy difference was statistically significant, where participants using the MUD-Visualizer had a median accuracy of 100.00%, while the control only had a median accuracy of 78.26%.

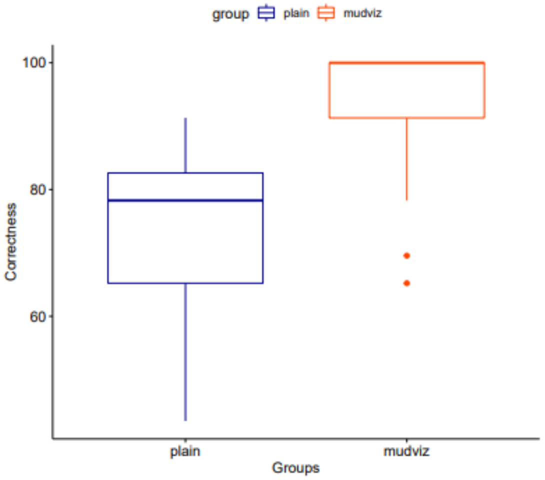


Fig. 4: Median accuracy between the control (i.e., plain) and the intervention (i.e., mudviz).

3) Implications: This visualization approach has the potential to expand to be used for SBOMs. When users receive the manufacturer's usage device access control list, they gain insight into the components of their device. Similar to an SBOM, the MUD is machine-readable but not easily accessible to humans. To support developers in generating and visualizing MUDs, they require proper support. As demonstrated in our study, an effective method to do this was the MUD-Visualizer. Simply allowing visual manipulation instead of relying solely on written access control rules increased accuracy. This visualizer concept aligns with SBOM generation; creating a map of dependencies, where all other dependencies naturally flow as part of the overall SBOM's upstream connections. Adding or removing components becomes intuitive, as users do not have to review the entire SBOM for each change. This not only makes it computationally efficient but also useful to those interacting with the code.

B. SBOM Visualization Tools

While automation will drive the initial vulnerability identification, human interaction is important to determine the criticality of the vulnerability, scope, and impact. Organizations have finite resources and cannot remediate every risk. Visualization tools help map complex interdependencies, allowing analysts to assess the broader implications beyond automated outputs. Combining automation for detection and human expertise for contextual analysis creates a more comprehensive approach to managing security risks. However, for human expertise to be valuable, decisions must be accurate. Recall, that the MUD-Visualizer demonstrated a high correlation between accuracy and accessibility, and we can apply this finding to SBOMs.

1) Method: In this study, we compared two open-source SBOM visualization tools (i.e., It-Depends and DeepBits) against a machine-readable JSON file generated with the Software Package Data Exchange (SPDX) standard. It-Depends focuses on identifying dependencies and flags packages with vulnerabilities, as shown in Figure 5. This tool lacks detailed information, such as the source of the vulnerability [53]. In Figure 6, we show the use of DeepBits, a commercial AI-based SBOM generation suite [13]. This tool provides source information on package vulnerabilities, including CVE identifiers. For our study's control, we use JSON files using the SPDX standard, an open standard for representing SBOMs [48]. We use the JSON file format to provide a more *traditional*, machine-readable SBOM to compare against the other two visualization tools. An excerpt of the JSON file format is shown in Figure 2.

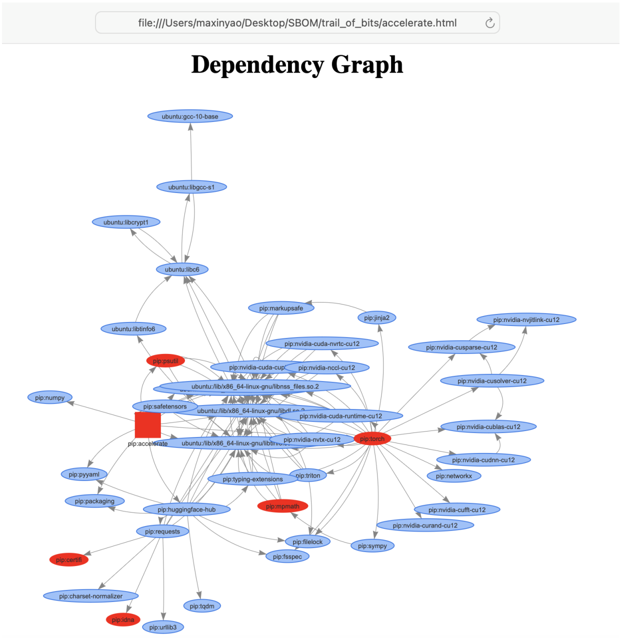


Fig. 5: It-Depends shows the interconnected dependencies between vulnerable packages.

This study was based on vulnerability identification and mitigation tasks. Specifically, we evaluated SBOM's acceptability and accuracy by randomly distributing participants into one of the three conditions (i.e., It-Depends, DeepBits, JSON). Within each condition, participants were presented with a series of code components and asked to determine the existence of a vulnerability, any dependencies, and any mitigation steps.

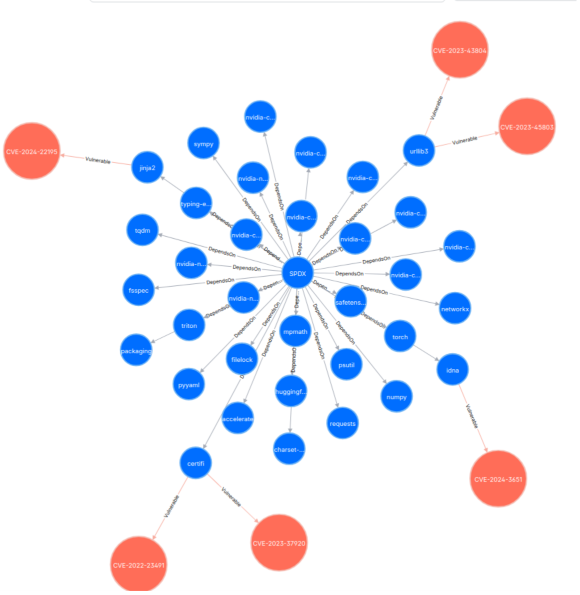


Fig. 6: A DeepBits generated graphic can show CVEs in code packages

2) Preliminary Results: This project is currently ongoing; however, a primary objective of this study is to determine whether visualizations enable SBOM usability. We have recruited 70 participants, who were randomly distributed to one of the three groups. All participants assigned to It-Depends (19) and DeepBits (22) completed the task. However, of the 29 participants assigned to the JSON file, only 17 completed the tasks. Both visualizations demonstrate how users can more efficiently engage with vulnerability information. Further analysis is needed to assess how these relate to performance and user experience. Additionally, the current results are based on 70 participants, where the majority of participants lack experience with SBOMs. As SBOMs become more common in software development, we anticipate more familiarity with SPDX's JSON file format.

3) Implications: The lower completion rate from the JSON condition indicates that users experience more cognitive load and frustration when working with less visual tools. In practice, this may delay the assessment of a vulnerability's scope and impact, prolonging exposure to an organization. However, while visualizations are a solution, this integration is still a challenge. With the complexity and necessity of tools to make SBOMs *usable*, is it realistic that SBOMs will ever be used? Only if they can be readily identified and integrated. When users are burdened with a vast number of fragmented SBOMs, regardless of format, they will never use them. Instead, having a comprehensive view of all the components and their dependencies creates usability.

C. Integrating SBOMs into Labels

Labels are designed to be used at the point of purchase, as it synthesizes technical information into a simple, graphical design. On the other hand, SBOMs, in their current form, are not user-friendly, as they are designed for longer-term use to track vulnerabilities throughout the supply chain. Ideally, combining SBOMs with labels will strengthen the security of products, encouraging consumers to make more security-conscious choices over a product's lifecycle. The efficacy of both labels and SBOMs will be a function of their reliability and relevance. In this experiment, we explore which security features are most important to consumers and how they can be used to convey implicit aspects of an SBOM [8], [10].

1) Method: This experiment began with a study of security guidelines to gauge their efficacy in practice [14], [33], based on sources from the Federal Trade Commission (FTC) [11], National Highway Traffic Safety Administration (NHTSA) [35], Federal Bureau of Investigation (FBI) [17], Online Trust Alliance (OTA) [42], NIST [47], and Open Web Application Security Project (OWASP) [43]. The resulting union of the 131 best practices was 56 unique recommendations. As federal labeling efforts would primarily impact users within the federal acquisition system (i.e., suppliers, vendors, and buyers), we include additional federal guidelines [36]–[38], [46] and AI & ML factors [48].

We used these 73 security factors in a virtual card sorting exercise to identify insights into participant decision-making. The 66 participants were recruited and asked to identify the relative importance of a given factor for a specific purpose (e.g., supports transport encryption as a component of secure operations). Specifically, participants were presented with a security feature that was associated with one of five security categories (e.g., authentication, secure onboarding). Each feature was placed into one of four categories (i.e., very important, important, less important, or not important) according to the degree to which they would influence a participant's purchasing decision.

2) Results: As we assume consumers of SBOMs would have some technical literacy, we selected participants with some expertise in coding. In addition, previous work had found that only the more technically literate users would engage with the technical information presented on labels. We identify the attitudes towards a specific category using measures of central tendency and dispersion to rate the distribution of responses. Results from the study are aggregated into five separate security categories and represented in Figure 7. The initial identification of *top ten* items for label design are:

1) Sensitive personal information
2) Two-factor authentication
3) Brute force protection
4) Transport encryption
5) Standards compliance
6) Vulnerability process
7) Specialized hardware requirements
8) Encryption at rest
9) Required consent of data sharing
10) System backups

The study also aimed to identify the relationship between security factors and different groups. The results show only education and technical acumen had any significant correlation with any security factors or categories. And even then, it was only marginally influential between variables.

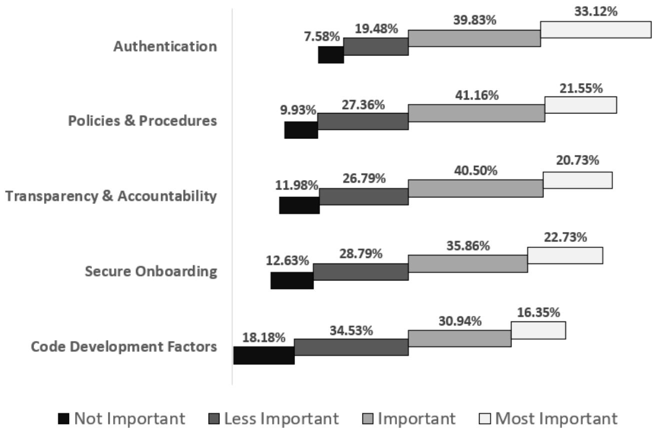


Fig. 7: Aggregate responses are categorized into security categories.

3) Implications: From this study, we can see that security labels, and to a higher degree, SBOMs, are a significant research challenge. However, SBOMs have significant upside if leveraged properly, extending beyond purchasing decisions to secure operations post-purchase. Ensuring secure operations involves integrating with threat modeling, mapping vulnerabilities to services, linking with the attack chain, and identifying mitigation options. A critical question then emerges: will companies invest in security? Knowledge of vulnerabilities is essential to evaluate appropriate return on investment, and sharing this information across organizations could increase investment within the ecosystem. The existence of SBOMs, as a mechanism of information sharing, increases investment simply by decreasing uncertainty. SBOMs should ideally enhance the situation by improving information flow. Next, we will determine if consumers would be willing to pay for more transparent security.

D. Willingness to Pay for Security

Willingness-to-pay is the maximum amount of money a consumer is willing to spend to acquire a good or service, denoting the value they place on a particular item [25]. Empirical results of multiple laboratory investigations illustrate that consumers will pay for security and demonstrate the importance of quality and brand to consumers [1], [7], [8], [12], [15], [22]. Additional research in willingness to pay for security and privacy indicates a greater willingness to pay for privacy [1], [22], [44]; however, security and privacy are complex, ever-changing, and often intertwined. Modern privacy includes dimensions that are clearly aligned with security: risk, integrity, and trust. Moreover, if privacy is the goal, security is the enabler. In this experiment, we seek to understand if consumers will pay more for increased security.

1) Method: To determine consumers' willingness to pay, we conducted a simulated purchasing experiment. We surveyed 599 participants and gave each $15, informing them they would receive their product and any remaining funds at the end of the study. Participants were then divided into six experimental groups: five interventions based on different security indicators and one control with no indicators. Products were labeled with these security indicators to determine if they would influence choice. We utilized Amazon marketplace listings to provide external validity since these listings already have built-in economic and product trade-offs. Additionally, participants are more likely to authentically engage with these listings to process information across product descriptions, pricing structure, customer reviews, product ratings, and product features/designs. Since these listings mirror actual economic influences consumers face, representing more multi-dimensional decision dynamics, it allows us to better determine the impact of security in the marketplace.

2) Results: A key research question centers around whether security labels can increase a consumer's willingness to pay for a product. To assess this we evaluated the pricing for each label compared to the control and employed a Mann-Whitney U Test, which allowed us to compare the distributions of our samples to determine whether they were statistically different from the control. Given the design of our experiment, any deviation in spending between the labeled groups and the control group could reasonably be theorized to reflect more security-conscious decision-making, as security labeling was the only variable introduced. Our results indicate that in terms of general consumers, there are no statistically significant differences when compared to the control. This suggests that, overall, security labels do not drive a collective increase in participants' willingness to pay for more secure products.

However, when focusing on consumers who are already predisposed to caring about security, we saw an increase of 16.5% above non-security-aware consumers and 11.3% above the control. These results were statistically significant, indicating a willingness to pay among Security-Aware participants. The increase was less pronounced within the Privacy-Aware participants, and not statistically significant. This suggests that privacy concerns alone do not substantially elevate willingness to pay for security features.

Willingness to Pay

	SA	S&PA	PA
Non-Security-Aware	+16.5%	+22.0%	+4.4%
Control	+11.3%	+20.7%	+4.9%
Intervention	+10.4%	+19.3%	+3.4%

TABLE I: Percent increase for Security-Aware (SA), Privacy-Aware (PA), and Security&Privacy-Aware (S&PA) participants vs. Non-Security-Aware, Control, and Intervention participants. For example, SA participants were willing to pay +16.5% more for security compared to non-security-aware consumers.

3) Implications: We can apply three lessons learned from this experiment – the importance of simplicity, the need to communicate economic value to the consumer, and the role of familiarity. An effective awareness campaign could not only emphasize the significance of security but also provide clear and accessible information about the benefits and implications of cost savings. Over time, increased awareness and trust in security could significantly enhance the overall security market, as individuals would come to expect it. Communicating security's value proposition is critical as general consumers often do not understand underlying benefits and the economic trade-offs (i.e., higher upfront investment in security may save more money in the long term).

IV. Conclusion

In this paper, we reviewed how information asymmetry can lead to market failure. We explored the historical underpinnings of a lemons market, used to describe a used car market, and we applied a similar framework for new technologies. Studying this phenomenon highlighted that even in the digital age, where a vast amount of information is easily accessible, information asymmetry can persist. This persistence of a lemons market results in adverse consumer interaction, due to hidden or obscured information, ultimately deteriorating the overall quality of goods in the market. We reviewed two methods that could mitigate this: visualizing SBOMs and using SBOM information in security labels. In doing so, it should drive an increase in consumers' willingness to pay.

Ultimately an SBOM, supported by user-friendly interactions and the right tools, has the potential to foster a market for safe and secure software across its lifecycle, from development and verification to operations and purchase support. By providing transparent and easily consumable information about complex software, the SBOM becomes a catalyst for informed decision-making. This peels back the layers of the lemons market for any device that is reliant on software code as a foundation. Overall, the SBOM can correct how the marketplace interacts with stakeholders, aligning itself with established mental models.

Acknowledgment

We would like to thank Atmikha Jeeju for her assistance in reviewing this paper.

References

[1] Alessandro Acquisti, Leslie K. John, and George Loewenstein. What Is Privacy Worth? *The Journal of Legal Studies*, 42(2):249–274, 2013.

[2] George A Akerlof. The Market for "Lemons": Quality Uncertainty and the Market Mechanism. In *Uncertainty in Economics*, pages 235–251. Elsevier, 1978.

[3] Vafa Andalibi, Jayati Dev, DongInn Kim, Eliot Lear, and L Jean Camp. Is Visualization Enough? Evaluating the Efficacy of MUD-Visualizer in Enabling Ease of Deployment for Manufacturer Usage Description (MUD). In *Annual Computer Security Applications Conference*, pages 337–348, 2021.

[4] Aaron Bangor, Philip T Kortum, and James T Miller. An Empirical Evaluation of the System Usability Scale. *Intl. Journal of Human–Computer Interaction*, 24(6):574–594, 2008.

[5] Kevin Benton, L. Jean Camp, and Vaibhav Garg. Studying the Effectiveness of Android Application Permissions Requests. In *2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pages 291–296, March 2013.

[6] L. Jean Camp, Shakthidhar Gopavaram, Jayati Dev, and Ece Gumusel. Lessons for Labeling from Risk Communication. In *Workshop and Call for Papers on Cybersecurity Labeling Programs for Consumers: Internet of Things (IoT) Devices and Software*, pages 1–3, Washington D.C., September 2021. NIST.

[7] Peter Caven, Zitao Zhang, Jacob Abbott, Xinyao Ma, and L. Jean Camp. Comparing the Use and Usefulness of Four IoT Security Labels. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA, 2024. Association for Computing Machinery.

[8] Peter J. Caven, Jacob Abbott, and L. Jean Camp. Towards a More Secure Ecosystem: Implications for Cybersecurity Labels and SBOMs. In *TPRC 51*, pages 1–15, Rochester, NY, 2023. SSRN.

[9] Peter J Caven, Shakthidhar Gopavaram, Jayati Dev, and L Jean Camp. *SoK: Anatomy of Effective Cybersecurity Label Development*. SSRN, Rochester, NY, 2023.

[10] Peter J. Caven, Shakthidhar Reddy Gopavaram, and L. Jean Camp. Integrating Human Intelligence to Bypass Information Asymmetry in Procurement Decision-Making. In *MILCOM 2022 - 2022 IEEE Military Communications Conference (MILCOM)*, pages 687–692, Rockville, MD, 2022. IEEE.

[11] Federal Trade Commission. FTC Report on Internet of Things Urges Companies to Adopt Best Practices to Address Consumer Privacy and Security Risks. January 2015.

[12] Lorrie Faith Cranor, Yuvraj Agarwal, and Pardis Emami-Naeini. Internet of Things Security and Privacy Labels Should Empower Consumers. *Communications of the ACM*, 67(3):29–31, 2024.

[13] Deepbits Technology. Generate, Distribute and Monitor SBOMs in One Platform. https://www.deepbits.com/sbom.

[14] Andrew Dingman, Gianpaolo Russo, George Osterholt, Tyler Uffelman, and L. Jean Camp. Good Advice That Just Doesn't Help. In *2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 289–291, Orlando, FL, 2018. IEEE.

[15] Pardis Emami-Naeini, Janarth Dheenadhayalan, Yuvraj Agarwal, and Lorrie Faith Cranor. Are Consumers Willing to Pay for Security and Privacy of IoT Devices? In *In Proceedings of the 32nd USENIX Security Symposium*, pages 1–18, Anaheim, CA, 2023. USENIX Association.

[16] Winand Emons and George Sheldon. The Market for Used Cars: New Evidence of the Lemons Phenomenon. *Applied Economics*, 41(22):2867–2885, 2009.

[17] Federal Bureau of Investigation. IoT Poses Opportunities for Cyber Crime, September 2015. https://www.ic3.gov/media/2015/150910.aspx.

[18] Federal Communications Commission. Rosenworcel Announces Cybersecurity Labeling Program for Smart Devices, 2023.

[19] Federal Communications Commission. FCC Creates Voluntary Cybersecurity Labeling Program for Smart Products, 2024.

[20] Baruch Fischhoff. *Communicating Risks and Benefits: An Evidence Based User's Guide*. Government Printing Office, 2012.

[21] Sabine Glock, Simone Maria Ritter, RCME Engels, AJ Dijksterhuis, Rick Bart van Baaren, and Barbara Caterina Nadine Müller. 'Smoking kills' vs.'Smoking Makes Restless': Effectiveness of Different Warning Labels on Smoking Behavior. 2013.

[22] Shakthidhar Gopavaram, Jayati Dev, Sanchari Das, and L. Jean Camp. IoT Marketplace: Willingness-To-Pay vs. Willingness-To-Accept. In *20th Annual Workshop on the Economics of Information Security*, 2021.

[23] Andrei Hagiu and Julian Wright. Multi-Sided Platforms. *International journal of industrial organization*, 43:162–174, 2015.

[24] David Hammond, Geoffrey T Fong, Ann McNeill, Ron Borland, and K Michael Cummings. Effectiveness of Cigarette Warning Labels in Informing Smokers About the Risks of Smoking: Findings from the International Tobacco Control (ITC) Four Country Survey. *Tobacco control*, 15(suppl 3):iii19–iii25, 2006.

[25] W Michael Hanemann. Willingness to Pay and Willingness to Accept: How Much Can They Differ? *The American Economic Review*, 81(3):635–647, 1991.

[26] Allen D Householder, Garret Wassermann, Art Manion, and Chris King. The CERT Guide to Coordinated Vulnerability Disclosure. *Software Engineering Institute (Carnegie Mellon University). Disponible en https://bit. ly/3CSCaz5*, 2017.

[27] Daniel Kahneman and Amos Tversky. Prospect Theory: An Analysis of Decision Under Risk. In *Handbook of the fundamentals of financial decision making: Part I*, pages 99–127. World Scientific, 2013.

[28] Jae-Cheol Kim. The Market for "Lemons" Reconsidered: A Model of the Used Car Market with Asymmetric Information. *The American Economic Review*, 75(4):836–843, 1985.

[29] Konrad Kollnig, Anastasia Shuba, Max Van Kleek, Reuben Binns, and Nigel Shadbolt. Goodbye Tracking? Impact of iOS App Tracking Transparency and Privacy Labels. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, page 508–520, New York, NY, USA, 2022. Association for Computing Machinery.

[30] Nancy Leveson et al. Medical Devices: The Therac-25. *Appendix of: Safeware: System Safety and Computers*, 1995.

[31] Jonathan Levin. Information and the Market for Lemons. *RAND Journal of Economics*, pages 657–666, 2001.

[32] Behnood Momenzadeh and Jean Camp. Peeling the Lemons Problem with Risk Communication for Mobile Apps.

[33] Behnood Momenzadeh, Helen Dougherty, Matthew Remmel, Steven Myers, and L. Camp. Best Practices Would Make Things Better in the IoT. *IEEE Security & Privacy*, PP, May 2020.

[34] Behnood Momenzadeh, Shakthidhar Gopavaram, Sanchari Das, and L Jean Camp. Bayesian Evaluation of User App Choices in the Presence of Risk Communication on Android Devices. In *International Symposium on Human Aspects of Information Security and Assurance*, pages 211–223. Springer, 2020.

[35] National Highway Traffic Safety Administration. Cybersecurity Best Practices for Modern Vehicles. *Report No. DOT HS*, 812:333, 2016.

[36] National Information Assurance Partnership. Protection Profile for Application Software. Technical report, NIAP, October 2021.

[37] National Institute of Standards and Technology. Recommended Criteria for Cybersecurity Labeling for Consumer Internet of Things (IoT) Products. Technical report, U.S. Department of Commerce, February 2022.

[38] National Institute of Standards and Technology. Recommended Criteria for Cybersecurity Labeling of Consumer Software. Technical report, U.S. Department of Commerce, February 2022.

[39] National Institute of Standards and Technology. National Vulnerability Database (NVD) Statistical Summary. Technical report, 2024.

[40] National Institute of Standards and Technology. NVD News. Technical report, 2024.

[41] National Telecommunications and Information Administrator. Framing Software Component Transparency: Establishing a Common Software Bill of Materials (SBOM). 2020.

[42] Online Trust Alliance. OTA Internet of Things, accessed April 2019. https://www.internetsociety.org/ota/.

[43] Open Web Application Security Project. OWASP IoT Project, accessed April 2019. https://www.owasp.org/index.php/IoT_Security_Guidance.

[44] Prashanth Rajivan and Jean Camp. Influence of Privacy Attitude and Privacy Cue Framing on Android App Choices. In *Authentication Workshop of the 12th Symposium on Usable Privacy and Security*, Denver, CO, 2016. USENIX Association, USENIX Association.

[45] Prashanth Rajivan and L. Jean Camp. Influence of Privacy Attitude and Privacy Cue Framing on Android App Choices. In *Twelfth Symposium on Usable Privacy and Security*. USENIX Association, 2016.

[46] Secure Communications Alliance and IoT PP Working Group. IoT Secure Element Protection Profile (IoT-SE-PP). Technical report, Smart Communication Alliance, December 2019.

[47] Adam Sedgewick. Framework for Improving Critical Infrastructure Cybersecurity, Version 1.0. 2014.

[48] SPDX AI ML Working Group. SPDXv3 AI SBOM, 2022. https://lists.spdx.org/g/spdx-ai.

[49] Lara Stocchi, Naser Pourazad, Nina Michaelidou, Arry Tanusondjaja, and Paul Harrigan. Marketing Research on Mobile Apps: Past, Present and Future. *Journal of the Academy of Marketing Science*, pages 1–31, 2022.

[50] Kamala Swayampakala, James F Thrasher, David Hammond, Hua-Hie Yong, Maansi Bansal-Travers, Dean Krugman, Abraham Brown, Ron Borland, and James Hardin. Pictorial Health Warning Label Content and Smokers' Understanding of Smoking-Related Risks—a Cross-Country Comparison. *Health education research*, 30(1):35–45, 2015.

[51] National Telecommunications and Information Administration. Software Suppliers Playbook: SBOM Production and Provision. 2021. https://www.ntia.doc.gov/report/2021/minimum-elements-software-bill-materials-sbom.

[52] The White House. Biden-Harris Administration Announces Cybersecurity Labeling Program for Smart Devices to Protect American Consumers.

[53] Trail of Bits. It-Depends. https://github.com/trailofbits/it-depends.