Integrating EPSS-Based Vulnerability Risk Prediction with Genetic-Algorithm Scheduling for Enterprise Patch Management

Arsénio Ferraz
School of Engineering of the Polytechnic Institute of Porto (ISEP/IPP)
1010137@isep.ipp.pt

César Vieira
School of Engineering of the Polytechnic Institute of Porto (ISEP/IPP)
1241523@ isep.ipp.pt

Rui Soares
School of Engineering of the Polytechnic Institute of Porto (ISEP/IPP)
1221283@ isep.ipp.pt

Mário João
School of Engineering of the Polytechnic Institute of Porto (ISEP/IPP)
1221019@isep.ipp.pt

Gonçalo Jesus
School of Engineering of the Polytechnic Institute of Porto (ISEP/IPP)
1220822@isep.ipp.pt

Abstract—**Enterprise patch management requires prioritizing and scheduling remediation actions across heterogeneous IT infrastructures under strict operational constraints. This work presents an integrated decision support approach that surveys vulnerability scoring and exploitation risk signals using supervised machine learning approaches that use the Exploit Prediction Scoring System (EPSS) as the target label and evolutionary optimization, specifically genetic algorithms, for workforce and infrastructure-aware scheduling. We position these techniques in the context of realistic enterprise environments, highlighting the importance of heterogeneous Windows-based workloads, maintenance windows, and human resource availability (including staggered shifts and non-working intervals) when designing practical decision support systems for patch management.**

Keywords— Vulnerability Management; Patch Management; Exploit Prediction Scoring System (EPSS); Machine Learning; Genetic Algorithms; Scheduling; Decision Support Systems; Enterprise IT Infrastructures.

I. Introduction

Modern enterprises operate highly heterogeneous IT infrastructures composed of multiple application layers, platforms and technologies. These environments evolve organically over time and often combine legacy and modern systems, reflecting organizational, technical and business constraints. As a result, vulnerability and patch management in enterprise contexts is inherently complex, where remediation actions may affect different layers of the software stack, introduce inter-component dependencies, require service restarts and must be carefully coordinated to avoid unacceptable downtime.

In this work, we consider a representative enterprise baseline centered on Windows Server environments hosting business-critical workloads such as Active Directory, IIS/SharePoint, SQL Server, and Exchange. Servers are distributed across DEV, UAT, and PROD environments, each with distinct maintenance windows and recovery time objectives (RTOs). While this baseline reflects a common and realistic deployment scenario, the proposed EPSS-based learning and genetic-algorithm (GA) scheduling approach is not limited to this specific stack and can be generalized to more complex and heterogeneous enterprise infrastructures.

Crucially, effective patch management is not only a problem of vulnerability ranking, but also a resource and time-constrained scheduling problem. In practice, remediation plans must respect server-specific maintenance windows as well as workforce availability constraints, including staggered working hours, on-call rotations and non-working intervals, such as lunch breaks. To support this integration, vulnerability data must be carefully preprocessed and enriched to include both exploitation-risk signals and attributes relevant to operational planning.

II. From CVSS to EPSS as an Exploitation-Risk Signal

The Common Vulnerability Scoring System (CVSS) was introduced by Mell et al. [1] as a standardized approach for assessing vulnerability severity. Although CVSS has been widely adopted in industry, subsequent empirical studies by Allodi and Massacci [2] demonstrated that CVSS scores correlate only weakly with real-world exploitation, limiting their usefulness for operational prioritization.

To address this limitation, the Forum of Incident Response and Security Teams (FIRST) proposed the Exploit Prediction Scoring System (EPSS) [3], which estimates the probability that a vulnerability will be exploited in the near future. EPSS has emerged as a practical operational metric and is increasingly used as a supervised learning target, enabling predictive models to generalize exploitation risk across heterogeneous vulnerability features and software ecosystems.

III. Data Preparation

The proposed decision support framework relies on a dataset specifically prepared to support both vulnerability risk prediction and planning-oriented decision making. Rather than starting from raw vulnerability feeds, this work builds upon an existing vulnerability management dataset and extends it with additional information required for operational scheduling.

The primary dataset used in this study is the “CVE, CISA KEV & EPSS Dataset” [4] collection available on Kaggle. This dataset provides structured vulnerability information commonly used in enterprise vulnerability management contexts, including CVE identifiers, severity, exploitability, impact scores and EPSS score. The dataset reflects a realistic starting point for security teams, as it resembles data typically aggregated from scanning and asset management tools.

To enrich this dataset with authoritative vulnerability metadata and exploitation related signals, additional information was retrieved from the National Vulnerability Database (NVD) through its public API. Using CVE identifiers as primary keys, NVD data was integrated to obtain product and its corresponding versions affected and if a Proof of Concept (PoC) is available online. This integration step ensured consistency with standardized vulnerability definitions and enabled the inclusion of temporal and severity related features.

Temporal features, including vulnerability age and time since disclosure, were also derived, as exploitation risk and remediation urgency evolve over time.

Numerical features were normalized to ensure comparability, while categorical attributes related to software products and vulnerability types were encoded to support tree-based learning models.

Additional attributes were derived to represent characteristics relevant to patch management, such as estimated remediation effort, software criticality and associations between vulnerabilities and affected systems. These attributes enable the planning component to reason about resource allocation, sequencing constraints, and workload distribution.

Overall, the data preparation phase complemented the existing vulnerability dataset allowing for both machine learning based risk estimation and genetic algorithm based planning. This preprocessing step is therefore a prerequisite for integrating vulnerability intelligence with executable remediation schedules under real enterprise constraints.

IV. Supervised Learning Models and Validation

Early work by Bozorgi et al. [5] demonstrated that supervised machine learning models can outperform heuristic rules when predicting exploit availability. Building on this foundation, ensemble tree-based methods have become dominant for tabular security data.

Random Forests, introduced by Breiman [6], provide robustness and a degree of interpretability. Gradient boosting, formalized by Friedman [7], was later extended into scalable systems such as XGBoost by Chen and Guestrin [8] and LightGBM by Ke et al. [9]. Subsequent research further improved the efficiency and scalability of boosting methods, including communication efficient parallel tree construction [10], GPU acceleration [11] and quantized training techniques [12].

Distance-based and margin-based models remain relevant baselines. The k-Nearest Neighbors algorithm was formalized by Cover and Hart [13], while Support Vector Machines were introduced by Cortes and Vapnik [14]. These models are commonly evaluated together to assess trade-offs between predictive performance, scalability, and interpretability in EPSS-aligned risk prediction.

Model validation is conducted using metrics such has precision, recall, F1-score and ROC-AUC. To ensure robust generalization, cross-validation techniques are employed during training.

This validation approach provides a more realistic assessment of model performance in operational environments, where predictions must generalize to future vulnerabilities rather than historical data.

V. Planning and Scheduling with Genetic Algorithms

Automated planning and scheduling have long been central topics in artificial intelligence. Genetic algorithms (GAs), introduced by Holland [15] and later systematized by Goldberg [16], are evolutionary optimization techniques particularly well suited to complex, constrained search spaces.

In the context of enterprise patch management, scheduling must account for multiple interacting constraints: server availability (maintenance windows and blackout periods), application and service dependencies (e.g., databases preceding IIS- or SharePoint-based services), reboot requirements, limited technical parallelism, and human resource constraints such as staff calendars, skill coverage, staggered shifts, and non-working intervals. GAs naturally accommodate these requirements, as chromosomes can encode patch-to-time-slot and patch-to-technician assignments, while fitness functions penalize violations of infrastructure and workforce constraints. In this context, genetic algorithms are used as heuristic optimization methods, aiming to generate high-quality feasible schedules rather than guaranteed optimal solutions.

VI. Research Gap and Positioning

The literature shows substantial progress in both vulnerability risk prediction and optimization-based scheduling. However, many approaches stop at producing ranked lists of vulnerabilities without ensuring that remediation plans are operationally feasible. Conversely, classical planning and scheduling approaches often lack data-driven risk inputs grounded in real exploitation likelihood.

Hybrid approaches that combine EPSS-aligned supervised learning for risk estimation with workforce-aware genetic-algorithm scheduling offer a pragmatic pathway toward end-to-end decision support for enterprise patch management. By explicitly modeling heterogeneous Windows workloads, environment-specific maintenance windows, and realistic human availability constraints, such approaches bridge the gap between vulnerability intelligence and executable remediation plans.

VII. Conclusion

This paper presented an integrated approach to enterprise patch management that combines EPSS-based vulnerability risk prediction with genetic-algorithm scheduling under realistic operational constraints. Unlike traditional approaches that focus exclusively on vulnerability ranking, the proposed framework explicitly addresses workforce availability, maintenance windows, and system dependencies.

A key contribution of this work is the explicit separation between risk estimation and decision-making. Supervised machine learning models are used to estimate exploitation likelihood, while genetic algorithms translate these predictions into feasible remediation schedules. This separation improves modularity, interpretability, and practical applicability.

By aligning data-driven risk prediction with planning and decision support concepts, this work demonstrates how machine learning and optimization techniques can be combined to address complex real-world cybersecurity problems. Future work will focus on large-scale experimental validation and the incorporation of dynamic threat intelligence updates.

REFERENCES

[1] P. Mell, K. Scarfone, and S. Romanosky, “A Complete Guide to the Common Vulnerability Scoring System,” NIST, 2007.

[2] L. Allodi and F. Massacci, “Security Events and Vulnerability Data for Empirical Analysis,” IEEE Trans. Dependable Secure Comput., 2017.

[3] FIRST, “Exploit Prediction Scoring System (EPSS),” 2021.

[4] CVE, CISA, KEV & EPSS Dataset, “https://www.kaggle.com/datasets/francescomanzoni/vulnerability-management-datasets”

[5] M. Bozorgi et al., “Beyond Heuristics: Learning to Classify Vulnerabilities and Predict Exploits,” Proc. KDD, 2010.

[6] L. Breiman, “Random Forests,” Mach. Learn., 2001.

[7] J. H. Friedman, “Greedy Function Approximation: A Gradient Boosting Machine,” Ann. Stat., 2001.

[8] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” Proc. KDD, 2016.

[9] G. Ke et al., “LightGBM: A Highly Efficient Gradient Boosting Decision Tree,” Proc. NeurIPS, 2017.

[10] Q. Meng et al., “A Communication-Efficient Parallel Algorithm for Decision Tree,” Proc. NeurIPS, 2016.

[11] H. Zhang, S. Si, and C.-J. Hsieh, “GPU Acceleration for Large-scale Tree Boosting,” SysML, 2018.

[12] Y. Shi et al., “Quantized Training of Gradient Boosting Decision Trees,” Proc. NeurIPS, 2022.

[13] T. Cover and P. Hart, “Nearest Neighbor Pattern Classification,” IEEE Trans. Inf. Theory, 1967.

[14] C. Cortes and V. Vapnik, “Support-Vector Networks,” Mach. Learn., 1995.

[15] J. H. Holland, “Adaptation in Natural and Artificial Systems,” 1975.

[16] D. E. Goldberg, “Genetic Algorithms in Search, Optimization and Machine Learning,” 1989.