Applications of Machine Learning to Cybersecurity

Singhal, Nithin
nsinghal32@gatech.edu

Lingamaneni, Vamsi Krishna
vlingamaneni6@gatech.edu

Mulani, Nityam
nmulani3@gatech.edu

I. Introduction

Information Technology (IT) is playing an increasingly pivotal role in critical infrastructure systems, and for good reason. Modern critical infrastructure relies on network technologies, internet-connected devices, and wireless communications to enable the large-scale distribution of essential resources to consumers efficiently. The integration of IT enhances efficiency, reliability, and scalability, making it possible to monitor and manage complex systems effectively. Interconnected devices, for instance, enable rapid detection and resolution of individual component failures, reducing the risk of widespread disruptions. However, alongside these benefits, the adoption of IT introduces significant challenges and vulnerabilities.

The main challenge associated with IT is that connecting a system's devices to the internet gives cyber-criminals the opportunity to exploit vulnerabilities and gain unauthorized access to critical systems. This connectivity expands the attack surface, making essential infrastructure susceptible to a range of cyber threats, such as data breaches, ransomware attacks, and system sabotage. Cyber-criminals can disrupt operations, compromise sensitive information, and even endanger public safety by exploiting weaknesses in network security. As a result, safeguarding these systems against potential cyberattacks has become a critical priority in the design and management of modern infrastructure.

Machine Learning (ML) offers a promising solution for this challenge. Unlike traditional cyber security methods, ML-based systems are capable of adapting to evolving threats by analyzing vast amounts of data to identify patterns, detect any anomalies and predict potential vulnerabilities in any system. This paper proposes a three-layered security framework that utilizes ML techniques to enhance cyber security in IT systems present in critical infrastructures. The framework will have an combination of penetration testing, malware detection and anomaly detection, to provide robust, resilient, adaptive and layered defense mechanism.

A. AI threat

A problem emerges when considering Artificial Intelligence. AI will create chaos in the cybersecurity field as it will allow cybercriminals to target "attacks at unprecedented speed and scale while avoiding traditional, rule-based detection measures", meaning "a new era of scalable, custom-made, and human-like assaults"[1]. According to Guembe et al, the current day cybersecurity tool we have won't work against the advanced AI driven cyberattacks[1]. They say that it's a matter of "machine efficiency vs. human effort", especially as AI continues to improve, requiring advanced cybersecurity specialists to counter them, which would be expensive and difficult to find. This is because an AI could create an attack that is faster, more unpredictable, more sophisticated, and more adaptable through obfuscating algorithms that change identifiers leaving current day tools and cybersecurity specialists struggling to defend [1]. Systems not designed to defend against AI will struggle and fail under the onslaught of these attacks.

As shown by the study, there are many different ways of using AI to attack systems. Within reconnaissance, there are a multitude of ways that AI can improve cyberattacks chance of success, they include "intelligent target profiling, clever vulnerability detection / intelligent malware, intelligent collection / automatic learn behavior and intelligent vulnerability prediction / outcome" [1]. Even the human factor is affected by AI as it can generate synthetic phishing URLs that are significantly more effective than previous methods of randomly generated segments. DeepPhish is one that improves previously effective phishing links using "LSTM model to create a phishing URL classifier" to produce new links, raising success rates of two previous threat actors from .69

II. Penetration Testing

Penetration testing, or pen testing, is a cybersecurity measure taken to increase resilience and prevent attacks from happening. It is a "simulation of an attack to verify the security of a system or environment to be analyzed", with many avenues such as "physical means utilizing hardware, or through social engineering"[3]. The overall goal is to test under extreme circumstances to ensure that the system behaves normally under strain and to find security vulnerabilities before malicious actors can exploit them, with auxiliary goals of testing employee and company preparation[3].

According to Denis, Zena, and Hayajneh there are typically 4 types of penetration testing, "external testing, internal testing, blind testing, and double blind testing"[3]. External testing works by targeting externally visible servers or devices, to find if an attacker can gain access and to what level they could. Internal testing checks to see what a standard authorized user can do. A blind testing, having someone or a team attack the system with limited information to simulate an actual attacker. With a double blind working in much of the same way, except it is on both sides and only a few within the company know about the attack[3].

Fundamentally though the different types of tests all work by first gathering information about the system being targeted, attempting to get into the system whether for real or virtually, and then reporting back what was found[3]. There are currently both automated tools and manual tools out there that can aid

you in testing or do it for you[3]. Another popular way is by having certified ethical teams to probe your system for you[4].

There are some downsides to pen testing in that it is very cost and labor intensive, since it can require so much time and resources[4]. This means that companies only performed at milestones, meaning there could be unpatched errors.

There is some hope as AI doesn't only have to be used offensively, it can also be used to defend. We can enhance current day cybersecurity tools using machine learning and AI to detect attackers, and AI based attacks that humans would struggle to defend against.

Professor Confido et al was able to enhance PenBox, an automated pen testing framework with machine learning[5]. PenBox is originally a prototype developed by the European Space Agency to test space systems and is able to discover assets, find vulnerabilities, access these vulnerabilities, and then attack other systems. They enhanced it by reducing the number of steps it takes to access a system through reinforcement learning. Their aim was to reduce the high-cost and effort intensive problem of traditional pen testing, this would allow for better protection as companies would be able to perform pen testing much more often[5]. This is crucial as AI attacks will target any vulnerabilities in a system en mass.

Reinforcement learning works by having a set of rewards for the agent after reaching a certain state, then encouraging them to find a cumulative policy that maximizes the total reward[5]. In this case PenBox is rewarded after performing successful intrusion into the system, along with having costs for performing certain actions such as IP scanning to promote efficiency. To find an optimal policy it must find a balance between exploring previously undiscovered states and exploitation of already known high rewarding states. This balance of trying new actions or using already known actions is maintained through random selection called Epsilon-Greedy Action Selection[5].

They then used the Q-learning algorithm and Deep Q-Networks(DQN) to help with training[5]. A Q-value used in the Q learning algorithm gives the "estimated value for doing a certain action that will lead to the next state". Which then allows the identification of the best action for a given state by choosing the action with the highest Q-Value, allowing it to determine the best policy for that specific scenario. This is then represented using a table with pairs of state and action then estimate its value. The problem with this approach is limited in its generalization ability and does not scale well[5].

A Deep Neural Net, or DNN, which is called Deep Q-network, or DQN, when used to estimate Q-Values, is used to overcome the scaling problem of Q-Learning[5]. Using a technique called Deep Q-Learning, which is when a DQN is used for Q-Learning, works by having the Neural Net estimate the value instead of having a perfect value from the table[5].

The training was very successful as it was able to improve in its goal of compromising the machine, with lower steps after being trained[5]. With 35 seconds of training it was able to compromise the machine in 703 steps, and the best result being only 33 steps required. There are some negatives as the result was still worse than a human doing it, as it only took 4 steps when done manually. It is inherently computationally expensive as a machine learning based technique. This is again only a prototype made by the European Space Agency; it could be further improved, along with optimizations for cost and performance[5].

III. Malware Detection

The integrity and resilience of critical IT infrastructures are essential. As cyber threats continue to increase in complexity and volume, traditional methods of malware detection, such as signature-based systems, struggle to provide adequate protection. This is especially true in the case of polymorphic and zero-day malware, which evolve quickly and are often designed to bypass conventional security measures. Machine learning (ML) has emerged as a promising solution for overcoming these challenges. By leveraging large datasets and advanced algorithms, ML models can adapt to new threats, detect sophisticated malware, and significantly enhance the security of critical IT systems.

One of the primary benefits of machine learning in malware detection is its ability to detect previously unknown threats. Unlike traditional malware detection methods, which rely on predefined signatures, machine learning models can analyze vast amounts of system behavior and identify patterns that indicate malicious activity. This is crucial in sectors such as healthcare, energy, and finance, where malware attacks can cause significant disruptions and financial losses. This paper discusses the benefits, techniques, and real-world applications of machine learning for malware detection, specifically focusing on how these techniques can strengthen critical IT infrastructures.

Critical IT infrastructures are vital to the functioning of modern society, from ensuring continuous electricity supply in power grids to safeguarding sensitive health data in hospitals. Malware attacks targeting these infrastructures can have devastating consequences, including service outages, financial losses, and data breaches. Traditional methods of malware detection, such as signature-based systems, are limited by their reliance on known malware signatures and static detection models. As malware becomes more sophisticated, evolving rapidly to evade detection, these methods are no longer sufficient.

In response to these challenges, machine learning offers a more dynamic and adaptive approach. ML models can be trained to recognize malicious patterns in data, even if they have never been seen before. Unlike signature-based methods, which require constant updates to account for new malware variants, machine learning models can learn from data and improve over time, detecting novel malware as it evolves [6]. This makes ML an essential tool for defending critical IT infrastructures against the growing threat of advanced persistent threats (APTs) and zero-day vulnerabilities.

Machine learning encompasses a wide range of techniques, each offering different advantages depending on the problem at hand. In the context of malware detection, several ML

approaches have proven effective, including supervised learning, unsupervised learning, deep learning, and hybrid models. These techniques are designed to detect malware by analyzing different aspects of system behavior and file characteristics.

Supervised learning is one of the most common ML techniques used in malware detection. In this approach, the algorithm is trained using a labeled dataset that includes both benign and malicious samples. By learning from these examples, the model is able to make predictions about new, unseen samples. Some of the most commonly used supervised learning algorithms for malware detection include Support Vector Machines (SVMs), Random Forests (RF), and Artificial Neural Networks (ANNs).

Support Vector Machines (SVMs) are particularly effective for classifying malware based on feature sets extracted from system logs or file attributes. SVMs work by mapping the data into a higher-dimensional space and finding the optimal hyperplane that separates malicious from benign samples [7]. The experiments conducted [7] demonstrated that SVMs achieved high accuracy rates in distinguishing between benign and malicious files, particularly when the feature set included dynamic system behavior patterns, such as network traffic and system calls. These models were able to detect zero-day malware strains that had never been encountered before.

Random Forests (RF) are an ensemble learning method that constructs multiple decision trees and aggregates their predictions to arrive at a final classification decision. This approach is particularly effective in handling large and complex datasets, such as those generated by system logs or network traffic data [6]. In one experiment, the authors used Random Forests to classify malware based on a dataset containing over 5,000 samples, including both known and unknown malware. The results showed that Random Forests achieved a high detection rate while maintaining low false positives, making it a reliable choice for malware detection in critical IT infrastructures.

Artificial Neural Networks (ANNs) are deep learning models that are capable of learning complex, non-linear relationships within the data. In malware detection, ANNs are often used to recognize intricate patterns in system behavior or file structures [7]. One experiment [7] demonstrated that ANNs were particularly effective at identifying polymorphic malware, which constantly changes its code to evade detection. By learning from a large dataset of system behaviors, ANNs were able to classify previously unseen malware variants with high accuracy.

While supervised learning requires labeled data, unsupervised learning does not. In malware detection, unsupervised learning techniques can be used to identify anomalous behavior that deviates from normal patterns. Clustering algorithms, such as K-Means and DBSCAN, group similar data points together, allowing the model to identify patterns that may indicate malware activity [8].

Experiments involving unsupervised learning techniques focused on detecting unknown malware variants that did not match any known signatures [8]. The authors used clustering algorithms to group system behaviors based on characteristics like memory usage, file access patterns, and network traffic. The clustering model was able to identify groups of malicious activities that had not been previously labeled, showing that unsupervised techniques can effectively detect novel malware. While unsupervised learning is less common in malware detection, it holds significant promise for identifying new and unknown threats. It is particularly useful in environments where labeled data is scarce or where new malware strains emerge quickly.

Hybrid models combine multiple techniques to take advantage of the strengths of different approaches. For example, combining static analysis (analyzing the code structure of a file) with dynamic analysis (monitoring its behavior during execution) can provide a more comprehensive view of malware activity [8]. Hybrid models are particularly effective in detecting complex malware that uses both static and dynamic evasion techniques.

There are few experiments [8] which demonstrated the power of hybrid models in malware detection. The authors combined both static features (such as file headers and bytecode) and dynamic features (such as system calls and memory usage patterns) to train a hybrid classifier. This model outperformed single-model approaches in detecting sophisticated malware that used polymorphic techniques to alter its code at runtime. The hybrid model was able to identify malicious behavior even when the malware changed its signature to avoid detection.

Building on these hybrid approaches, there is a method that combines both static and dynamic analysis for malware detection in a single integrated framework [9]. The method combines static analysis, which focuses on examining binary code or executable files without running them, with dynamic analysis that involves monitoring the behavior of malware during execution [9]. This hybrid method has been shown to improve detection accuracy by leveraging both the code characteristics and the runtime behavior of the malware, offering a more comprehensive and robust approach to identifying malicious software. In this approach, static features include printable strings and metadata extracted from the binary files, while dynamic analysis involves observing API call sequences and interactions with the operating system, often using sandboxes or virtualized environments. The integration of these two methodologies allows for the identification of both known and unknown malware, addressing the limitations of each individual approach when used alone.

Deep learning has emerged as one of the most powerful techniques in malware detection due to its ability to analyze large and complex datasets. In particular, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have shown great promise in detecting malware by analyzing system behavior and file characteristics.

CNNs are particularly effective when applied to malware images. These images are generated by converting binary malware data into a visual format, allowing deep learning models to analyze the data at a pixel level [7]. Experiments [7] involved using CNNs to analyze grayscale images generated

from malware binaries. The results showed that CNNs were able to detect previously unseen malware variants with high accuracy, even those that used obfuscation techniques to evade detection by traditional methods.

RNNs, on the other hand, are well-suited for analyzing time-series data, such as sequences of system calls or network traffic. By processing sequences of events, RNNs can identify patterns that indicate malicious activity. Experiments [6] have demonstrated that RNNs were able to identify abnormal sequences of system calls that suggested malware activity. This ability to detect sequential patterns is crucial for identifying advanced malware that may attempt to mimic normal system behavior to avoid detection.

Transfer learning is a technique that allows models to leverage knowledge from one domain and apply it to another, reducing the need for large amounts of labeled data. In the context of malware detection, transfer learning can be used to adapt models trained on one type of malware (e.g., desktop malware) to detect malware on different platforms (e.g., mobile malware) [8].

Experiments [8] have demonstrated the effectiveness of transfer learning by training a model on desktop malware samples and then fine-tuning it with a smaller dataset of mobile malware samples. The results showed that transfer learning allowed the model to perform well on mobile malware detection despite the limited availability of labeled mobile malware data.

The integration of ML into malware detection systems offers significant benefits for critical IT infrastructures. One of the most important advantages is detection accuracy. ML models can recognize patterns in system behavior that are indicative of malicious activity, even when the malware is previously unknown. This capability is essential for detecting advanced threats that employ evasion tactics, such as zero-day malware or polymorphic malware that changes its appearance to avoid detection. ML-based detection systems can also enhance real-time response capabilities. By analyzing data continuously and detecting anomalies as they occur, these systems can alert security teams to potential threats immediately. This ability to respond quickly is crucial in preventing cyberattacks from causing widespread damage in critical infrastructures, such as power grids or hospitals.

Moreover, machine learning models can improve over time. As new malware samples are added to training datasets, ML models can adapt and learn to detect new threats more effectively. This continuous improvement makes ML-based systems more resilient to evolving cyber threats than traditional signature-based systems, which require constant manual updates.

Real-world applications of ML-based malware detection systems are already making a significant impact in various industries. For instance, in the healthcare sector, ML models are being used to protect patient data and prevent ransomware attacks. In the energy sector, ML models are helping secure power grid infrastructure from cyberattacks. In finance, machine learning is used to protect sensitive financial transactions from fraud and malware attacks. The increasing adoption of machine learning in these critical industries demonstrates its potential to strengthen the resilience of IT infrastructures against the growing threat of cyberattacks.

In conclusion, machine learning represents a powerful and adaptive solution for malware detection in critical IT infrastructures. By leveraging a variety of ML techniques, including supervised learning, unsupervised learning, deep learning, and hybrid models, organizations can detect and mitigate sophisticated malware threats that would otherwise evade traditional security measures. With its ability to continuously improve and adapt to new threats, machine learning will play a crucial role in securing the critical IT systems that underpin modern society.

IV. Anomaly Detection

The growing reliance on digital systems in critical infrastructure sectors such as energy, healthcare, and transportation has made their systems' networks prime targets for cyberattacks. Attackers aiming to disrupt or compromise these systems can cause significant harm by targeting these networks, ranging from service outages to catastrophic failures, with potentially disastrous consequences for public safety and national security. Detecting such intrusions early is crucial, as even a brief breach can lead to widespread damage. Network intrusion and anomaly detection algorithms are essential tools in identifying unauthorized access and abnormal behavior in real-time, enabling swift response and mitigation. These algorithms, especially those leveraging machine learning, can detect subtle deviations in network activity that may signal an attack, even if the threat is novel or sophisticated (e.g., zero-day exploits). By proactively monitoring and analyzing network traffic, these systems can safeguard critical infrastructure from increasingly complex cyber threats.

One notable example of a network breach in critical infrastructure is the 2015 Ukrainian power grid attack, which was attributed to a cyberattack by a Russian hacker group known as Sandworm. The attackers exploited a vulnerability in the remote control software used by Ukraine's power grid, allowing them to remotely disconnect over 230,000 people from their electricity supply for several hours. The attack was sophisticated, involving a combination of spear-phishing emails, malware, and the exploitation of outdated systems. Once inside the network, the attackers used the malware to manipulate the power grid's SCADA systems, which are essential for monitoring and controlling power grids. This breach highlighted the critical vulnerabilities in industrial control systems and raised alarms about the potential for similar attacks on power infrastructure worldwide. It also underscored the importance of network intrusion detection systems and anomaly detection algorithms to identify such sophisticated threats before they can cause widespread damage [10].

One early attempt at using machine learning for network intrusion was in 1999. Researchers at UT Austin used two different machine learning algorithms simultaneously: genetic algorithms and decision trees. In their system (called Network

Exploitation Detection Analyst Assistant), genetic algorithms are used to generate rules for identifying intrusive behavior in network traffic by evolving solutions iteratively. Each rule is encoded as a "chromosome," consisting of genes that represent various attributes of network connections, such as source IP, destination IP, ports, and protocols. The GA begins with a randomly initialized population of these rules and evaluates their performance against a pre-classified dataset. The evaluation (fitness) rewards rules that correctly identify anomalous connections and penalizes those that misclassify normal traffic. Through genetic operators like crossover (combining parts of two parent rules) and mutation (introducing random changes to genes), the algorithm refines the rule population over multiple generations. Niching techniques are applied to maintain diversity and discover multiple effective rules, ensuring the algorithm can detect a variety of intrusion patterns, including complex and distributed attacks [11].

Decision trees in the paper are employed to classify network events based on the same attributes (source IP, destination IP, ports, etc.). Using the ID3 algorithm, the training data is partitioned iteratively based on the attribute that maximizes information gain, creating a tree structure where nodes represent attributes and leaves represent classifications (e.g., "intrusion" or "normal"). Decision trees are pruned to simplify the generated rules and enhance their generalizability, avoiding overfitting to specific training instances. Unlike GAs, decision trees create a single comprehensive rule that encapsulates multiple clauses, allowing the intrusion detection system to evaluate events against a structured hierarchy. This structured rule set enables efficient filtering of network traffic and provides insights into patterns of anomalous activity, such as connections from high-risk IPs (e.g., "Hot IP" lists) [11].

By combining these two approaches, the system benefits from the exploratory power of GAs to discover diverse rule sets and the deterministic precision of DTs to create interpretable and hierarchical rules for intrusion detection [11].

Keep in mind the historical context of this approach. It was created in 1999, when computational power was far more limited. Although neural nets existed back then, they were less computationally feasible due to the less powerful machines of the time, so the designers in this paper chose more lightweight machine learning algorithms.

25 years later, in 2024, researchers from various universities published a paper entitled "Enhancing Critical Infrastructure Security: Unsupervised Learning Approaches for Anomaly Detection." This paper studies the performance of 6 unsupervised algorithms for detecting anomalies within a network of a critical infrastructure system. The main dataset being studied here is the SWAT (Secure Water Treatment) dataset, which contains network traffic information from 51 sensors, actuators, and PLC devices in an 11-day period. This period consisted of 7 days of normal behavior and 4 days of attack scenarios. The unsupervised learning models were trained on the normal behavior of the SWAT dataset and evaluated on the attack scenarios of the SWAT dataset as well as 9 other datasets, primarily containing network traffic data. While the 6 unsupervised algorithms are quite different, they all attempt to establish a baseline of "normal behavior" then use that to predict if some given activity is normal or anomalous [12].

The 6 models studied are: Isolation Forest, Local Outlier Forest, One-Class SVM, and 3 autoencoders (Vanilla Autoencoder, Variational Autoencoder, and VAE-LSTM). The 3 autoencoders are neural-net based models, while the other 3 are not. The Isolation Forest has much less computational cost than the others, so it was much faster to train and make predictions. The 3 autoencoder models were far slower but had much better accuracy at identifying attacks. In particular, the VAE-LSTM stood out as the best model, correctly classifying 23 of the 26 selected attack scenarios used for evaluation. The remaining two models, Local Outlier Forest, and One-Class SVM, had poor performance in terms of computational cost and accuracy when compared to the autoencoders [12].

The fact that these models are unsupervised makes them particularly applicable to critical infrastructure cybersecurity. Labeled data is often difficult to obtain, or only available in small amounts. These models work on unlabelled data, meaning they can learn from larger datasets where little effort is needed to construct this data. When given high frequency data, the VAE-LSTM can make a prediction in a matter of seconds [12], meaning there would only be seconds between a cyberattack occurring and an anomaly being reported. In addition, unsupervised models are good at detecting novel, previously unseen attacks, as long as they deviate from the baseline in some way. This means that zero-day exploits can potentially be secured using these types of models.

While there is a vast collection of research which uses machine learning models to detect anomalies in computer networks, some researchers approach the problem from a different angle. A paper written by researchers from the Air Force Institute of Technology entitled "Radio-frequency-based anomaly detection for programmable logic controllers in the critical infrastructure" explores a way of detecting anomalous operations in programmable logic controllers (PLCs), which are critical components in SCADA systems. These systems manage essential operations in critical infrastructure sectors like energy, water, and transportation. Traditional security measures, such as bit-level IT security protocols and intrusion detection systems, are not feasible to run on PLCs due to their limited computational resources and long operational lifetimes. This makes them particularly vulnerable to cyberattacks, as exemplified by high-profile cases like Stuxnet [13].

To solve this problem, the paper proposes an innovative radio-frequency (RF)-based anomaly detection methodology. This approach utilizes unique RF emissions produced by PLCs during their operation to monitor and detect deviations indicative of anomalies, whether caused by malicious activities or system failures. By leveraging physical-layer attributes instead of traditional bit-level data, the method enables effective anomaly detection without burdening the PLC's limited computing capabilities. The methodology demonstrates that RF signals can provide a robust foundation for identifying abnormal behaviors in SCADA systems, ensuring better pro-

tection of critical infrastructure [13].

The proposed system captures RF emissions emitted by PLCs during their normal operation. These emissions act as a unique ”fingerprint” of the system’s behavior, reflecting the internal activities of the PLC, which can be used to detect anomalies. The RF emissions are analyzed to extract relevant features, which represent the normal behavior of the PLCs. These features are used as inputs to machine learning models, where patterns in the RF data are learned to distinguish between normal and anomalous operations. While the paper does not have any experimental results regarding how machine learning models perform on RF data, it does refer to related research that uses machine learning algorithms to classify the RF emission data into ”normal” and ”anomalous” categories [13]. It is therefore reasonable to infer that machine learning models trained on RF emission data could be used for anomaly detection in a SCADA system.

A crucial property of machine learning is that it can be used to solve virtually any prediction-based problem as long as you have a sufficiently large and expressive dataset. This means that anomaly detection for critical infrastructure does not need to be restricted to analyzing network traffic data or RF emission data. It can learn from any dataset as long as it is representative of the system’s normal operating conditions and contains sufficient variation to capture potential anomalies. For example, machine learning models can be trained on datasets from sensor networks, operational logs, or even data generated by physical processes, such as temperature, pressure, or flow rates in industrial systems. The ability to process diverse types of data, from network traffic to sensor readings, allows machine learning-based anomaly detection to be applied to various aspects of critical infrastructure, including power grids, water treatment plants, and transportation networks. By learning the typical behavior of a system, these models can identify any deviations from normal operation, which might signal a cyberattack, malfunction, or failure, thus providing a robust, adaptable solution for ensuring the security and reliability of critical infrastructure.

V. Conclusion

We propose a three-layered security framework for critical infrastructure systems which uses ML to defend against cyber threats. The layers are:

1) Pen testing discover security flaws before deployment
2) Malware detection to identify attacks before they start
3) Anomaly detection to identify attacks shortly after they start

By incorporating these layers into a system, we propose that a minuscule portion of attacks will pass through all three layers, making the system highly robust.

For the first layer, pen testing should be done during the software development phase. Using ML models, we can simulate cyberattacks against the system, and if any are discovered, we can revise the system to be immune, or at least more secure against those vulnerabilities.

For the second layer, we can use malware detection ML models as part of a quarantining system for all newly added software. Before any program can run, it must be approved by the malware detection algorithm. If we use an accurate, generalizable model for this task, it will weed out a large percentage of malware that enters the system before it can cause damage or disruptions.

For the final layer, we use anomaly detection ML models to identify anomalous behavior of the devices in the system. If network intrusions are detected, we can disconnect the affected devices from the network. If some other data anomaly is detected, we can take analogous actions.

Having these three layers introduces redundancy to the system’s security. If these ML models were perfect, they would all stop 100% of attacks. Since we do not live in a perfect world, the models will never be 100% accurate, but potentially can still have high accuracy. Ideally, the models will usually misclassify different attacks. If this is true, it means that there will be very few attacks that bypass all 3 layers at the same time, and the system will be statistically safe. We hope that critical infrastructure engineers will utilize similar design philosophies as they implement their systems. While it is unlikely that our framework specifically will be used, the concept of redundancy between layers is likely to enhance security significantly while still being relatively simple to implement.

References

[1] B. Guembe, A. Azeta, S. Misra, V. C. Osamor, L. Fernandez-Sanz, and V. Pospelova, “The Emerging Threat of Ai-driven Cyber Attacks: A Review,” Applied Artificial Intelligence, vol. 36, no. 1, pp. 1–34, Mar. 2022, doi: https://doi.org/10.1080/08839514.2022.2037254.

[2] A. Correa Bahnsen, “DeepPhish Simulating Malicious AI,” Dec. 2018. Accessed: Dec. 06, 2024. [Online]. Available: https://i.blackhat.com/eu-18/Wed-Dec-5/eu-18-CorreaBahnsen-DeepPhish-Simulating-Malicious-AI.pdf

[3] M. Denis, C. Zena, and T. Hayajneh, “Penetration testing: Concepts, attack methods, and defense strategies,” 2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT), Apr. 2016, doi: https://doi.org/10.1109/lisat.2016.7494156.

[4] Y. Stefinko, A. Piskozub, and R. Banakh, “Manual and automated penetration testing. Benefits and drawbacks. Modern tendency,” 2016 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET), Feb. 2016, doi: https://doi.org/10.1109/tcset.2016.7452095.

[5] Alessandro Confido, E. V. Ntagiou, and M. Wallum, “Reinforcing Penetration Testing Using AI,” Mar. 2022, doi: https://doi.org/10.1109/aero53065.2022.9843459.

[6] A.-A. Mustafa Majid, A. J. Alshaibi, E. Kostyuchenko, and A. Shelupanov, “A review of artificial intelligence based malware detection using deep learning,” Materials Today: Proceedings, Jul. 2021, doi: https://doi.org/10.1016/j.matpr.2021.07.012.

[7] J. Singh and J. Singh, “A survey on machine learning-based malware detection in executable files,” Journal of Systems Architecture, vol. 112, p. 101861, Aug. 2020, doi: https://doi.org/10.1016/j.sysarc.2020.101861.

[8] A. Bensaoud, J. Kalita, and M. Bensaoud, “A survey of malware detection using deep learning,” Machine Learning with Applications, vol. 16, p. 100546, Jun. 2024, doi: https://doi.org/10.1016/j.mlwa.2024.100546.

[9] P. V. Shijo and A. Salim, “Integrated Static and Dynamic Analysis for Malware Detection,” Procedia Computer Science, vol. 46, pp. 804–811, 2015, doi: https://doi.org/10.1016/j.procs.2015.02.149.

[10] M. Pollard, ”A Case Study of Russian Cyber-Attacks on the Ukrainian Power Grid: Implications and Best Practices for the United States,” Pepperdine Policy Review, vol. 16, no.

1, Jan. 2024, Accessed: May 19, 2024. [Online]. Available: https://digitalcommons.pepperdine.edu/ppr/vol16/iss1/1/

[11] C. Sinclair, L. Pierce, and S. Matzner, "An application of machine learning to network intrusion detection," Proceedings 15th Annual Computer Security Applications Conference (ACSAC'99), 2020, doi: https://doi.org/10.1109/csac.1999.816048.

[12] A. Pinto, L.-C. Herrera, Y. Donoso, and J. A. Gutierrez, "Enhancing Critical Infrastructure Security: Unsupervised Learning Approaches for Anomaly Detection," International Journal of Computational Intelligence Systems, vol. 17, no. 1, Sep. 2024, doi: https://doi.org/10.1007/s44196-024-00644-z.

[13] S. Stone and M. Temple, "Radio-frequency-based anomaly detection for programmable logic controllers in the critical infrastructure," International Journal of Critical Infrastructure Protection, vol. 5, no. 2, pp. 66–73, Jul. 2012, doi: https://doi.org/10.1016/j.ijcip.2012.05.001.