AI based Intrusion Detection Systems

Course 2023-2024 Q1

19/12/2023

Group 3:

Natalia Dai

Clàudia Giró Figueras

Mario Martín Sola

Javier Villarreal Arias

TABLE OF CONTENTS

1. Introduction

Network security in information systems has always been an important aspect to have in mind when we talk about computers and how we use them. Threats are increasing day by day and it's crucial to know how to defend ourselves against them. One of the many tools we have to achieve that is Intrusion Detection Systems (IDS), a system designed to encounter intruders that want to have access to our data and resources for malicious intents.

And as all technology advances, IDS are not the ones to be left behind. Artificial Intelligence (AI) is starting to be implemented in this software to add an extra layer of security. It helps with tasks that can be hard to do or repetitive and to avoid human errors that can cause big catastrophes.

This project is dedicated to explain the implementation of AI in IDS technologies, explaining its mechanisms, different types of implementation and applications.

2. IDS

2.1. What is an IDS

Intrusion Detection System, or IDS, is a software application or a hardware device that inspects and monitors the content of the network traffic for signs of malicious activity, anomalies, violations of security, or possible attacks. These systems or products notify any activity that might compromise your data or network with alarms or automatic actions. It is akin to a high-tech burglar alarm for your network.

IDS is a specialized tool that operates by analyzing network packets and traffic patterns reading the content of logs files from routers, firewalls, servers and other network devices.

There are different types of IDS, depending on its functionality or its architecture that will be explained in the following sections.

2.2. Types based on the method

IDS are normally classified as two types: Signature-based IDS or Anomaly-based IDS:

Signature-based IDS

This kind of IDS maintains a database of known attacks based on predefined patterns for malicious network activities. It is compared against the traffic activity (logs), file hashes, known byte sequences or even email subject headings in order to detect suspicious data. In case there is a close match between a known attack and a recent online behavior, an alert will be triggered to notify the system administrator.

It is an effective and high accuracy system. The only issue would be the unawareness of new attacks since they would not be in the database or zero-day attacks.

Anomaly-based IDS

The latter type of IDS aims to identify unfamiliar attacks based on the definition of ordinary and anomalous behavior patterns. AI techniques (such as neural networks, machine learning or other mathematical methods) are used to create a model of a normalized baseline, being this the representation of how the system normally behaves, and then all network activity is compared to that baseline. When a deviation is detected an alert is activated.

This is useful for zero-day exploits which signature-based detection is not. However, it lacks high accuracy and might generate false positives since whatever not aligning with the "normal" behavior it would detect it as a suspicious activity.

2.3. Types based on the deployment

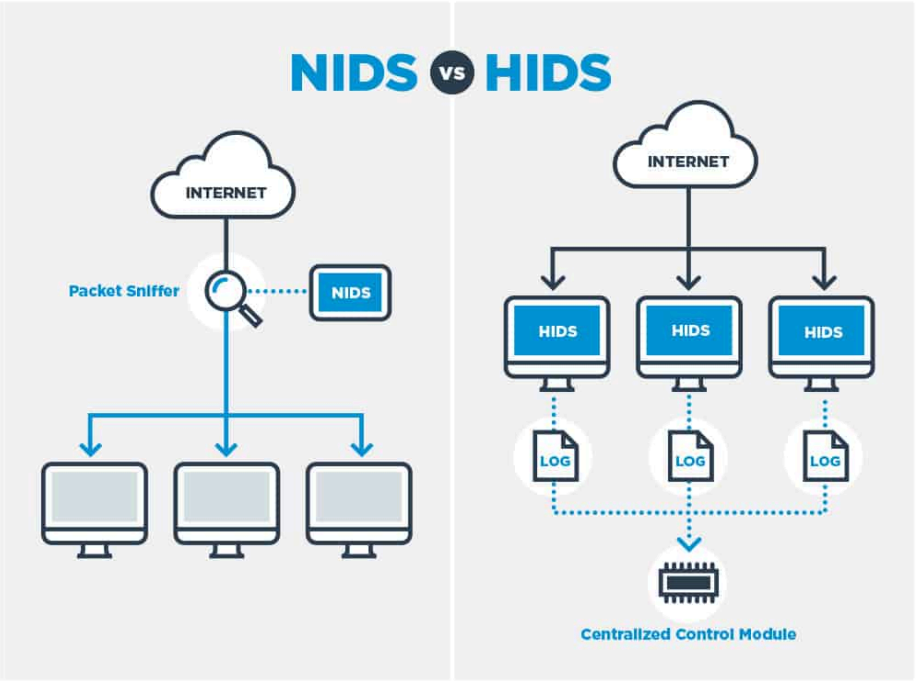
The most general ones are classified as Network IDS and Host IDS:

Network IDS

A NIDS strategically places sensors in various points across the network (the most vulnerable subnets). These would monitor and examine all traffic flowing to and from devices on the network without creating bottlenecks, and comparing it to known attack patterns. The Transport and applications protocol are also analyzed. Once an attack or suspicious behavior is observed, an alert is triggered and sent to the administrator.

Host IDS

A host-based IDS runs on hosts on which it is installed and it is only capable of monitoring the traffic passing through those devices, so that it can protect them against internal and external threats. It works by taking snapshots: it compares the most recent snapshot to previous ones, in order to identify differences or malicious activity. If there is something suspicious, the administrator will be notified.



3. AI and its evolution

3.1. Origins

The history of artificial intelligence dates back to ancient times, where myths and stories often featured the idea of crafted beings endowed with self intelligence or consciousness. These narratives laid to humanity's fascination with the idea of creating machines that could mimic cognitive processes. In fact, philosophers planted the seeds of modern AI by exploring the nature of human thought and attempting to understand cognition as a manipulable process and describing it as a mechanical manipulation of symbols.

The first very big step occurred in the middle of the 1940s with the invention of the *programmable digital computer*, a machine based on abstract mathematical reasoning. The period between these dates and the 1960 was strongly influenced by a conjunction of new technological developments, with the Second World War as catalyst for innovation. Visionaries and inventors such as John von Neumann and Alan Turing were considered the fathers of the technology behind it transitioning 19th century's decimal logic computers to binary logic machines. This shift allowed the manipulation of chains of 0s and 1s,

providing a significant evolution of the development of early computing systems and the subsequent evolution of artificial intelligence.

In 1957 it was predicted that the AI would surpass a human within a decade in chess. It proved correct, but took more than 20 years. Despite the initial enthusiasm and investments in AI, a phenomenon known as "*AI winter*" occurred: all the euphoria almost disappeared due to lack of objective completion.

It was with the advent of the first microprocessors at the end of 1970 that AI entered into the golden age of expert systems. The path was opened at MIT in 1965 and 1972, with an expert system specialized in molecular chemistry called *DENDRAL* and another one specialized in the diagnosis of blood diseases and prescription drugs, respectively. These were based on an "inference engine" programmed to be a logical mirror of human reasoning and generating high-level expertise answers based on input data.

Despite that, all this massive development fell again at the end 80s and early 90s, due to the significant efforts needed to understand the complex rules of AI, leading to a "*black box*" effect where *machine reasoning* was unclear. Maintenance became really problematic, and there were faster, less complex and cheaper ways of developing solutions. By the end of the decade, the term *artificial intelligence* became a taboo and started being substituted by the term "***advanced computing***".

Around the year 2010, AI experienced a resurgence driven by the availability of large volumes of data and the efficiency of graphics card processors, which occupied much less space compared to older machinery. Additionally, there was a shift in AI towards inductive approaches, leading to the development of what is nowadays known as "***deep learning***": the discovery of rules focused on classification and correlation using *neural networks*. This marked the beginning of a departure from manually coding rules.

Some of the driving forces behind these new techniques were IBM, with its *IBM Watson* machine, which outperformed two top contestants on the *Jeopardy!* quiz show, and Google, which successfully identified kittens in a video for the first time.

3.2. Use cases

Nowadays, AI is deeply embedded in our lives, powering a wide range of technologies and applications. From self-driving cars and virtual assistants to personalized recommendations and automated data analysis, AI is shaping the future of how we interact with the world around us.

The actual concept of AI revolves around its ability to learn and adapt, mimicking the human brain's ability to process information and make decisions. This is achieved through advanced algorithms and techniques, such as machine learning and deep learning, which enable AI systems to analyze vast amounts of data and extract patterns and insights.

AI's impact is evident across various domains. There are a lot of fields that have integrated the usage of this technology to increase productivity and data/results analysis. Some of the most populars use cases are:

1. Natural Language Processing (NLP)
2. Image and Speech recognition
3. Autonomous Vehicles
4. Customer service operations. Virtual Assistants and Chatbots
5. Healthcare Diagnosis
6. Robotics
7. Predictive maintenance
8. Process automation
9. Financial reporting and accounting
10. Cybersecurity

4. AI based IDS

4.1. What is it?

An AI IDS, or Artificial Intrusion Detection system, is a network security system that uses artificial intelligence to detect malicious activity on a network. The principal difference with a conventional IDS, is that it relies on machine learning algorithms to learn normal network traffic patterns and detect any deviations from them.

The AI IDS focuses on improving how the traditional IDS work. These are some concepts that want to be achieved:

- **Reduction of False Positives:** The quantity of false positives that occur in IDS can be unbearable for a human to deal with. With AI there will be a better distinction between normal and potentially malicious activities.
- **Anomaly Detection:** AI will be trained to detect new anomalies that have not been seen before by establishing a normal behavior pattern, helping to identify activities that may indicate a security breach.
- **Better Incident Response:** Some AI-based IDS include automated incident responses. These can be isolating affected devices, blocking suspicious network traffic, etc.
- **Adaptability:** AI enables us to adapt to changes in the threat landscape. Traditional IDS rely on signatures or static rules, leaving us vulnerable to new attacks that we are not prepared to face. The system can learn and improve its ability to detect and respond to threats, learning that can be used in the future if the same attack occurs.

4.2. How do AI IDSs work

4.2.1. Current datasets

Data is the core component when training any IDS that uses AI to detect attacks. Data can be collected from different sources, including host logs, network traffic, and application data. These are some of the most common datasets used for training current models:

- **KDDCup99** is the standard and most user dataset when it comes to train models for IDSs. The dataset is synthetic, which means that it was not gathered from real network traffic. It holds different types of attacks like DoS, User to Root, Remote to Local and Probing. Network logs show features as duration of connections, protocols, flags, login attempts. This dataset is quite old and has some important problems.

- **NSL-KDD** is an improved version of the KDDCup99 dataset to address the problems that this one has. Solves problems like the redundancy and duplication of records, tries to eliminate ambiguous records, etc.

- **ISCXIDS2012** is a dataset developed in 2012 which contains real network traffic captured in a controlled environment. Different attacks can be found like DoS, probing, malware. The features found are IP addresses, ports, protocols, packet size, duration, etc.

- **CICIDS2017** is another dataset, pretty similar to the previous one in terms of types of attacks and features, but in this case more modern and with higher volume of data.

There are more datasets that try to represent each of them the real state of the network in the year they were made, implementing different attacks like web attacks, port scans, brute-force, botnets… Some of these datasets are: CSE-CIC-IDS2018, UNSW-NB15, CIDDS…

As we can see, there are a bunch of different attacks that more or less try to represent some of the most common attacks. However, the training performed from until this year has some problems:

1. Most of the models trained to this day (around 70% of them) have been trained with KDDCup99 and NSL-KDD. The first one is the oldest we have mentioned since it is from 1999 and, of course, represents a pretty old context of the network. Models trained with this dataset will not be able to handle the current traffic in order to detect attacks. The dataset also has another problems regarding the balance. NSL-KDD tries to fix some of these problems, but the context of the traffic is still old.
2. The data from these two mentioned datasets is synthetic, which means that was not captured from real network traffic.
3. The method used to train these models was through the training of static datasets gathered (or not) from logs, traffic, etc. Since an IDS will detect attacks in real time. It would be interesting to try to train models exposing them to attacks in real time.

Since there seems to be a problem with datasets used. One of the uses we can give to AI to help the development would be using it to help us building these new datasets.

4.2.2 AI Labeling

It is part of the preprocessing in the development of a model of a Machine Learning model. It identifies the data without being processed to label it to specify its context for the models, to ensure the machine learning model makes precise predictions.

Companies use software, processes and data loggers to clean, structure and label data. Labels or annotations can take various forms depending on the task. Some common types are:

- **Image Classification:** assigning a label to an entire image.
- **Object Detection:** assigning a label to a specific object within an image.
- **Semantic Segmentations:** assigning a label to each pixel in an image, delineating object boundaries.
- **Text Classification:** assigning a label to a text.

- **Named Entity Recognition:** assigning a label and classifying entities (names, locations…) within a text.
- **Speech Recognition:** transcribing spoken words into text.

Labeled data is used in supervised learning, while unlabeled data is used in unsupervised learning. Labeled data is more difficult to obtain, as it is ponderous and more expensive to obtain.

Having labeled data allows AI to make more precise predictions, as it ensures a better quality control within the machine learning algorithms. Bad quality data will result in bad quality outputs. It also improves the usability of the data: classifying the data can be used to reduce the number of variables in a model. It has to be careful with human errors, as it can reduce the quality of the data.

4.2.3. Machine Learning models

Naive Bayes

The Naïve Bayes classifier is based on Bayes' theorem, which describes the probability of an event based on prior knowledge of conditions. It assumes that all features are independent from each other given the class label. It allows us to categorize data into classes by firstly examining the characteristics of this instance that we want to classify. Then, based on those features, Naive Bayes calculates the probability of that case belonging to each possible class and selects the one with the highest probability as the predicted category for that instance.

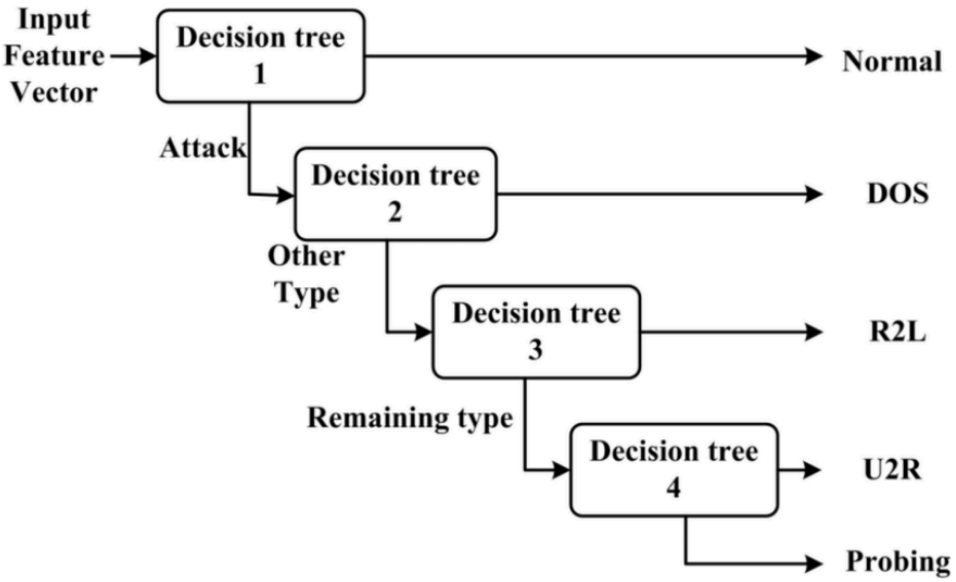
This model can be applied to AI-based IDS by analyzing all network traffic and behavior and can serve as the basis for differentiating between a normal network conduct and a potential anomalous one. It would calculate the probability of each feature occuring into those 2 options, gaining more efficiency and adaptability.

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

Decision trees

A decision tree is a flow-chart-like tree structure that uses a branching method to illustrate every possible outcome of a decision. Each node of the tree represents a question about a specific aspect, and each leaf leads to a different outcome to these questions, which means the final prediction in each case. The algorithm works by recursively splitting the dataset into subsets based on the most significant feature that best separates the data into homogeneous groups. They are really easy to interpret, however, decision trees are prone to overfitting, so the tree complexity must be controlled.

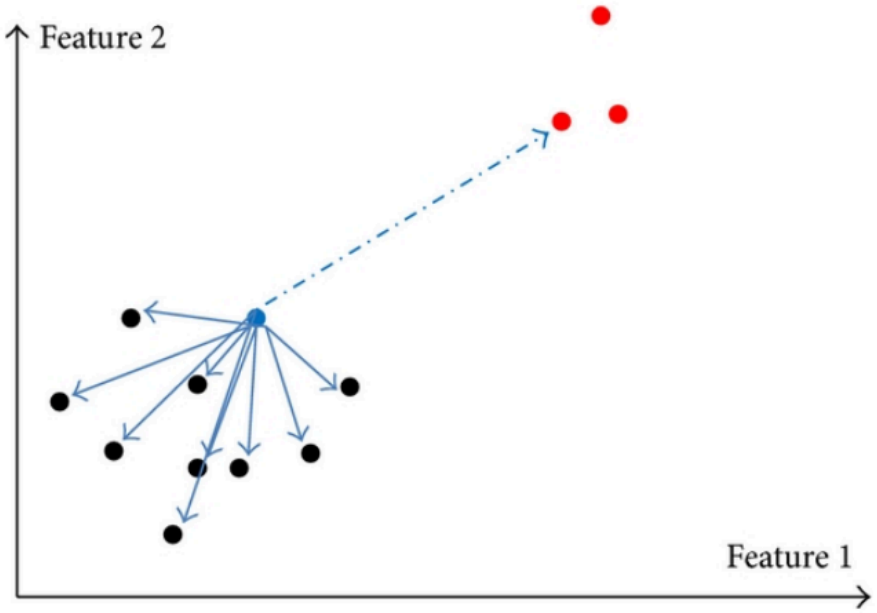
As for the application in this case, various network data attributes can be used as features and these would be the input for the decision tree algorithm. The two branches would be labeled as normal and anomalous network behaviors.



K-Nearest Neighbour

KNN classification algorithm makes predictions based on the similarity of the data from each other. It assumes that if most of the K nearest neighbors samples belong to a category, then the sample belongs to the same category. This model depends on the selected K, so it would be more robust if this value was larger in order to have more samples, but not too large to avoid increasing the computation time.

For the classification of the new instances in an AI-based IDS, KNN calculates the similarity (distance) between this instance and the stored instances in the training set and would be classified accordingly to a normal-based network system or an anomalous one.



4.2.4 Deep Learning models

Artificial Neural Networks

Inspired by how human brains work, these types of algorithms are the fundamental piece from all kinds of deep learning algorithms. The idea is to process the input data through elements called “neurons” organized in layers. Each neuron holds a weight and a bias, which gives information about how “important” this specific neuron is within the network. A specific operation with these values is performed inside the neuron and the output of all the neurons of a layer are transmitted throughout a full connection to the neurons of the next layer. Then, a final layer performs a classification task (could be, for example, classification of a virus). These NNs are also commonly called “feedforward” NNs.

These kind of algorithms are not that recent, since we can find first attempts of using this technique from 2010. Different improvements have been designed during these years to improve the results of feedforward NNs. For example, an ELM (Extreme Learning Machine) is a kind of feedforward NN suitable for data-based modeling in complicated processes and has an extremely fast learning capability.

But the best results are seen when other kind of NNs aside from this are used. The following DL techniques will explain these NNs.

Convolutional Neural Networks

The purpose of CNNs is to try to learn the relevant features of the input data. For the preprocessing for this model, the data (network packets) is segmented, breaking it down into smaller manageable units that can be fed into a CNN. Then, different processes like PCA can be used to extract the relevant features to reduce the size of the dimensionality.

After transforming the data, normalizing for preventing feature dominating and encoding some categorical variables are good practices before feeding the model.

CNN-MCL (mean convolution layer) is a CNN architecture developed for learning anomalies of the dataset. This method goes well since we want to detect strange or different behavior within the normal traffic of a network.

Some studies have also affirmed that modeling the kernel could be enough to detect most of the abnormalities of the incoming network.

Recurrent Neural Networks

RNNs are a type of neural network that work particularly well for sequential analysis, which fits perfectly with the idea of having to monitor the incoming traffic in real time. These models would analyze traffic over time and detect anomalies or malicious patterns. Variants of RNNs like LSTM-RNN and GRU-RNN address the vanishing gradient problem and are capable of retaining information over sequences of time.

Other approaches that have been studied are the combination of some of these models, from ensemble methods to networks being a combination of different architectures.

4.2.5. Generative AI for datasets

Generative AI refers to a class of artificial intelligence algorithms capable of generating new data similar to a given dataset. It can generate synthetic data based on patterns and relationships learned from actual data.

In the context of IDS it can be used to create synthetic network traffic data to be used as a dataset to train the ML models. This can help us to augment the available data to train our models, providing additional examples. Another advantage of synthetic data is privacy preservation: it can be generated without containing sensitive information.

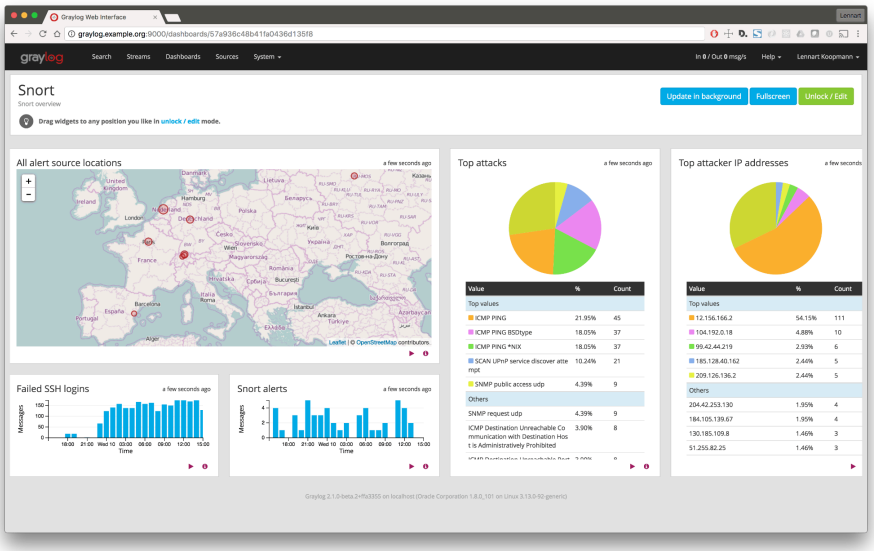
This type of data is very useful when we have an already big pool of real information and we want to extend it. Otherwise, using AI to generate more data from an already AI-generated dataset will not generate better results.

4.3. Examples

These AI-enabled IDSs represent a new generation of security solutions that seek to dynamically adapt to evolving threats and provide more efficient and accurate detection.

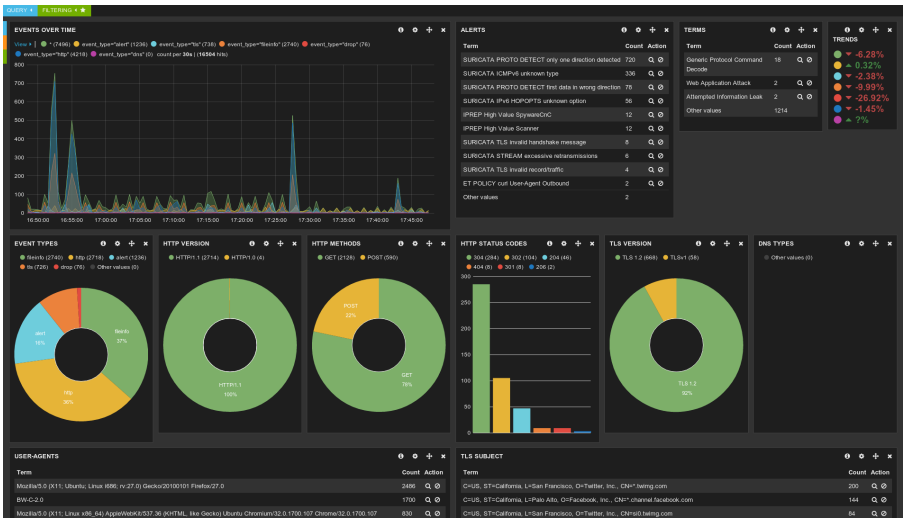
4.3.1. Snort

Snort is an open-source IDS that uses machine learning techniques to detect signature-based, anomaly-based, and behavior-based attacks. Being open-source, it is highly customizable, and its active community contributes rules and updates.



4.3.2. Suricata

Suricata is another open-source IDS that employs machine learning techniques similar to Snort, but it also offers machine learning-based detection capabilities. It stands out for its high performance and multi-threaded processing support.

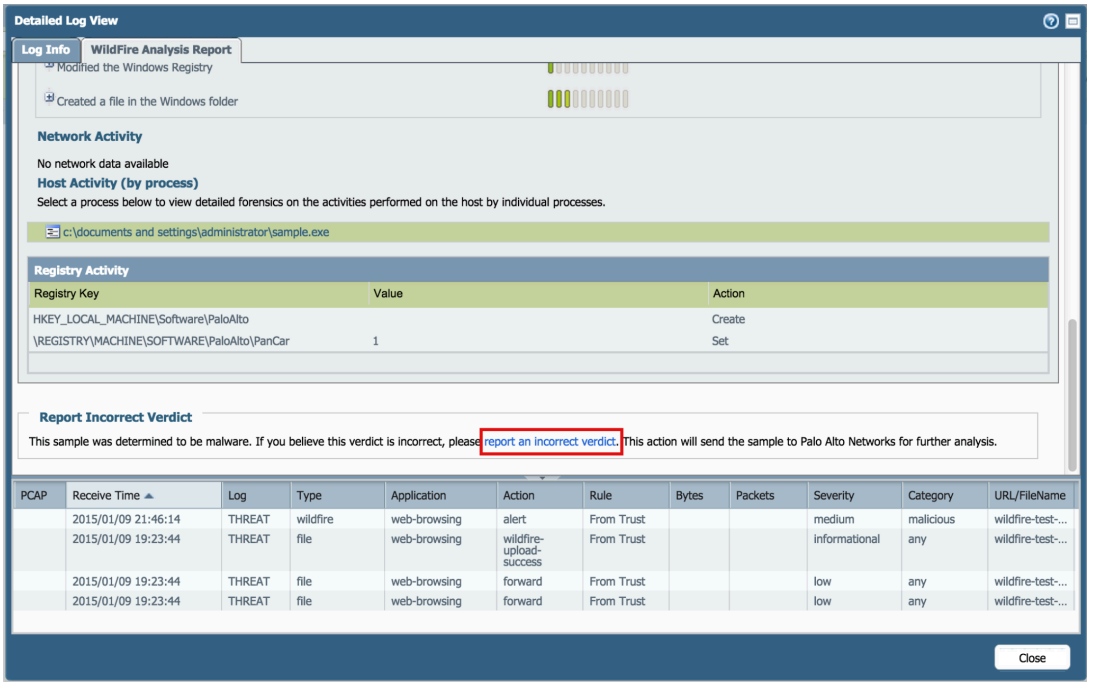


4.3.3. AlienVault USM

AlienVault USM is a commercial IDS that uses machine learning techniques to also offer a comprehensive approach by detecting signature-based, anomaly-based, behavior-based, and machine learning-based attacks. Additionally, it provides a Unified Security Management (USM) platform that integrates IDS, SIEM, and other security tools, along with event correlation capabilities.

4.3.4. Palo Alto Networks WildFire

WildFire by **Palo Alto Networks** is a commercial IDS that excels in an all-encompassing approach by detecting signature-based, anomaly-based, behavior-based, and machine learning-based attacks. It offers dynamic malware analysis to identify unknown threats and seamlessly integrates with the Palo Alto Networks security platform.



4.4. Future applications

In the future, it is expected that AI IDSs will be used for a lot of new and emerging applications. Some of these applications include:

- **Protecting IoT devices**: IoT devices are becoming increasingly vulnerable to attacks, and AI IDSs can help protect them by analyzing data traffic and device behavior to detect signs of malicious activity.

- **Preventing internal attacks**: AI IDSs can help detect internal attacks by analyzing user and device behavior to identify unusual activity.

- **Protecting mobile networks**: AI IDSs can help protect these networks by analyzing data traffic and uncommon patterns.

In addition to these specific applications, AI IDSs also have the potential to improve network security in general. For example, these can help organizations and users to gain a better visibility of their network environment, as they can analyze large amounts of data quickly and efficiently. This can lead to a better understanding of the environment and the threats to which they can be exposed.

5. Conclusions

Most of the models we have seen ensure that they have an accuracy of around 90-99% with the tests performed with their datasets, and that are, of course, quite good news. Signature based IDSs are getting obsolete. Attacks are evolving way too fast and it's impossible to try to detect all of them using this system so it is compulsory to find newer methods to detect future attacks.

One of the main problems regarding the training of IDS models is that there aren't many datasets published that are useful in representing the real scenario of cyberattacks in the present. Most of the models out there have proven their efficiency using these datasets for the training and testing. The majority of them use the KDD CUP 99 dataset, which has the problems presented before.

The cybersecurity field evolves fast and this includes the way attackers try to perform their attacks. The AI tools are extremely powerful and without any doubt we will start to see in the nexts years how these two fields start to work together.

6. Bibliography

IDS & AI IDS:

- https://www.sciencedirect.com/science/article/pii/S2665917423001630
- https://adyraj.medium.com/application-of-ai-in-intrusion-detection-system-9705d2efe050
- https://www.linkedin.com/pulse/autonomous-systems-power-ids-ia-madjiquene-ndong/
- https://www.iotworldtoday.com/security/tapping-ai-for-intrusion-detection-systems#close-modal
- https://www.n-able.com/blog/intrusion-detection-system
- https://www.geeksforgeeks.org/intrusion-detection-system-ids/
- https://www.paloaltonetworks.com/cyberpedia/what-is-an-intrusion-detection-system-ids
- https://www.helixstorm.com/blog/types-of-intrusion-detection-systems/

History of AI:

- https://sitn.hms.harvard.edu/flash/2017/history-artificial-intelligence/
- https://en.wikipedia.org/wiki/History_of_artificial_intelligence
- https://www.coe.int/en/web/artificial-intelligence/history-of-ai

AI Machine learning:

- https://en.wikipedia.org/wiki/Naive_Bayes_classifier
- https://www.sas.com/en_gb/insights/articles/analytics/machine-learning-algorithms.html#:~:text=There%20are%20four%20types%20of,%2Dsupervised%2C%20unsupervised%20and%20reinforcement.
- https://www.hindawi.com/journals/jece/2014/240217/

AI IDSs examples and use cases:

- https://www.snort.org/
- https://suricata.io/documentation/
- https://cybersecurity.att.com/products/usm-anywhere
- https://www.paloaltonetworks.es/network-security/advanced-threat-prevention

AI-generated datasets:

- https://garystafford.medium.com/unlocking-the-potential-of-generative-ai-for-synthetic-data-generation-f42907cf0879

- https://www.sangfor.com/blog/cybersecurity/what-is-generative-ai-cybersecurity

Crèdits transparències GCS i TXC Universitat Politècnica de Catalunya