Université Libre de Bruxelles

Master in cybersecurity

Network security

Intrusion Detection Systems

C. Louis
Cochez Benjamin
Croche Loïc
Hanquin Benjamin
January 11, 2022

Contents

1 Introduction

Today, computer systems are more and more confronted to new threats due to the advent of new technologies. For this reason, it's important to implement new detection mechanisms or adapt existent mechanisms. For example, with new IoT devices present in several domains like the health or the automobile, the lack of IDS could lead to some undetected intrusions. To carry out these mechanisms, the usage of IDS has become essential. The world of IDS is varied, their nature differ completely in function of the context, adapting to new threats.

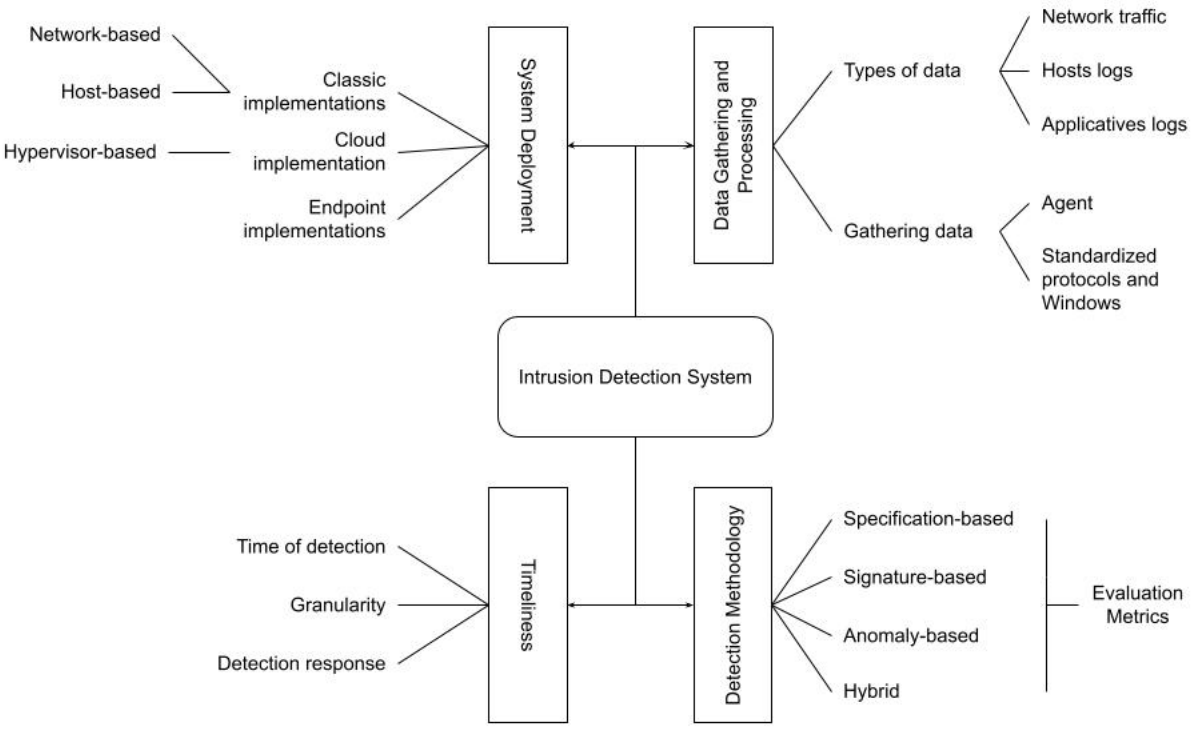
Firstly, we could define what's an intrusion as an unauthorized attempt, successful or not, trying to break, access or misuse property. The goal of an IDS is to detect and report these intrusions. Let's quote Ant Allen, research director at Gartner (firm providing information, advice, and tools for leaders in IT) "For an enterprise to protect itself from abuse of its information, it must monitor the events occurring in its computer system or network and analyze them for signs of intrusion. To do this, the enterprise must install an Intrusion Detection System (IDS)." So an IDS is a passive device that monitors the medium on which it has been installed (host/network) and once it detects abnormal behaviors, reports or alarm to an administrator or to an active device, following the level of risk evaluated by the IDS.

Secondly, the notions of detection and prevention are blurred when talking about intrusion systems. Sometimes, the notion of IPS (Intrusion Prevention System) is related to the IDS in the literature. IDS were suppose to detect intrusion only while IPS also had to prevent, react using automated response. Nowadays there is almost no distinction anymore, IDS point to a mix of detection and prevention. The term fell into disuse in profit of more specific terms like NIDS (network intrusion detection system), HIDS (host-based intrusion detection systems), WIPS (wireless intrusion prevention system).

In this paper we will entertain multiple aspects of the intrusions detection systems that we can see in the IDS taxonomy picture:

- The current state of the art for implementing an IDS.
- The type of data sources processed.
- The temporal aspect of the processing.
- The different methods of detection.

1.1 IDS Taxonomy



2 System deployment

2.1 Classic implementations

2.1.1 Network-based

Network intrusion detection systems (NIDS) is a type of IDS analyzing the entire flowing traffic on the belonging network. Its goal is to detect anomalous, inappropriate, or any data that is considered unauthorized and harmful on the company network. Like DoS attacks, port scanning, and botnets.

Once an attack has been identified, or that an intrusion behavior is noticed, the NIDS triggers an alert which must be catched by an administrator or by an IPS (intrusion protection system). Common attacks spotted by NIDS are : IP address spoofing, media access control (MAC) address spoofing, Address Resolution Protocol (ARP) cache poisoning and DNS name corruption. A very popular NIDS is Snort.

Wired On a wired topology, it is easy to take all the traffic passing through the firewall and send it for analysis to NIDS. The medium is easy to protect since the DMZ and LAN area is inside the building and the access is restricted to allowed people.

Wireless The wireless evolution has opened an exciting world for mobility, but security is a harder concern than on a wired network. On the contrary of the wired topology, the medium is really difficult to control. The range of the wireless network often reaches outside of the company buildings. Which means that any attacker standing near the building could send packets into the network. A lot of attack can occurs on this kind of medium : “Rogue devices, incorrect

configurations, connectivity problems, jamming, man-in-the middle attacks, wardrivers, scanning, RF interference, MAC spoofing, DoS attacks, attempts of brute force to get pass 802.1x, strong RFI, or use of traffic injection tools".(14)

The goal of a NIDS is to analyze the whole network traffic in wireless, sensors can catch it and send it to NIDS but they must be strategically positioned to catch the whole traffic. IEEE 802.11 doesn't help on that side, since it uses 2 frequency bands (2.4GHz and 5GHz) and each band is separated into channels. But each sensor can monitor a single channel on a specific band at a time. There are therefore two options, the sensor does a channel scanning (change of channel very quickly to scan each channel a few times per second) but it can miss some malicious activity. Or use a sensor for each channel all time long, but it requires a lot more sensors. The location is also really important to catch the whole organization WLAN range, but also in regions where WLAN is not accessible, for detecting and monitoring rogue AP as well as ad-hoc WLANs. Without forgetting that range of sensors can vary a lot following the location of the employees in the building. So it is recommended to overlap the range of sensors of at least 20%. And last but not least, if the sensor is open to physical threats (physically accessible to some attacker) an anti-tamper sensor should be used.

Implementation NIDS are obviously analysing both wired and wireless traffic.

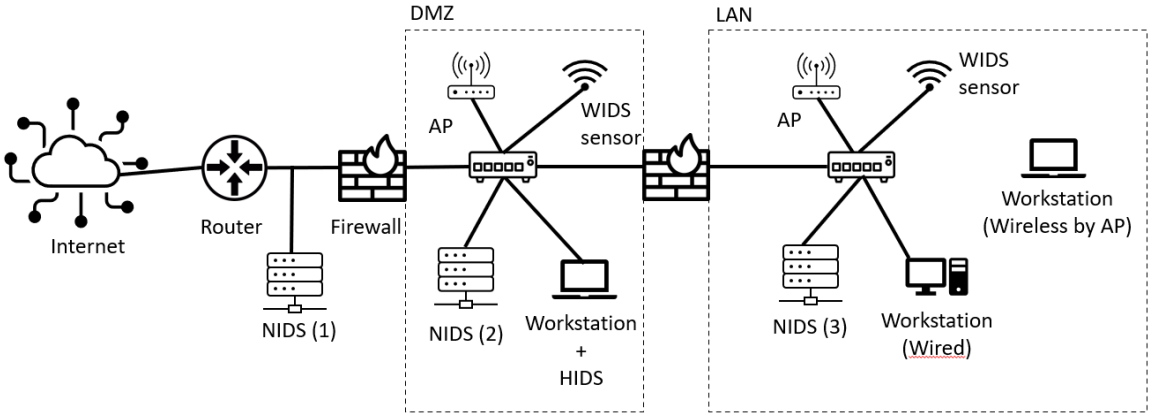


Figure 1: Architecture of a network-based intrusion detection system

There are 3 common places to install a NIDS :

1. Before the DMZ firewall : for analysis purpose, to count how many times the company is getting attacked, port scanning, ... But it generates an insane amount of useless logs since most attacks are blocked by the firewall.
2. In DMZ : Most useful place, since attacks passed the first firewall and that attacks usually occur on DMZ machines as pivot point.
3. In LAN : In case that a LAN machine gets corrupt and acts maliciously, the DMZ NIDS could not log network traffic in LAN.

2.1.2 Host-based

An host-based IDS (HIDS) runs, on the contrary of the network-based one, on hosts and devices. It means that if a company wants to have all its hardware protected, they are going to implement an HIDS on each sensible hardware device. The HIDS is a software on a computer monitoring OS logs checking for malicious behaviour (i.e. security/configuration file modified, a lot of files modified in a short lap of time meaning ransomware encrypting all files, ...).

HIDS has been growing since a study showed that the main part of malicious activities in a company network is coming from the employees. And a NIDS can not spot such malicious activities. Common HIDS is Splunk (which has a paying option for NIDS).

2.2 Cloud implementation

The emergence of clouds has created new issues. It uses many virtual machines that are constantly changing location or owner. And there are then two "informed users" that want to know if their machine is clean, the client but also the supplier of the VM (18) .

2.2.1 Hypervisor-based

In cloud computing the IDS can be at the hypervisor level, cloud computing is managing a lot of Virtual Machines (VMs) at the same time, so it became a must to defend against attacks on hypervisor, protecting all VMs running on it. The hypervisor and VM-Dependent Intrusion Detection and Prevention System (VMIDPS (15)) for a virtualized cloud environment has been proposed to protect the cloud computing. Which is composed of 4 components that are collaborating together. A management unit, a VMIDPS server, an IDPS core, and obviously the hypervisor.

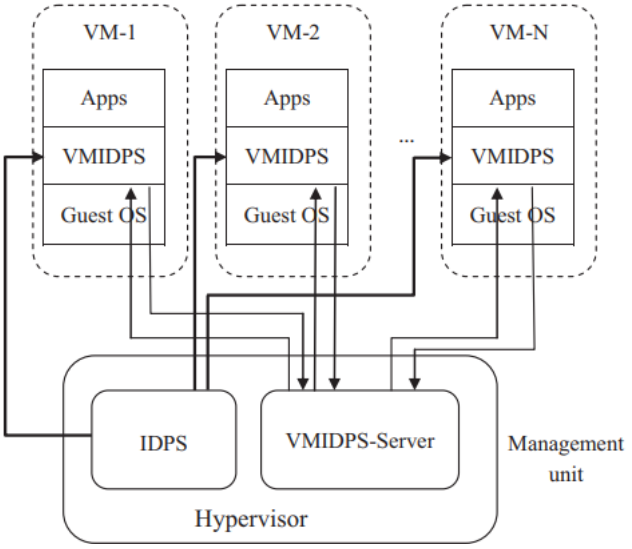


Figure 2: Architecture of VMIDPS

2.3 Endpoint implementations

In endpoint security there are also Protocol-based IDS (PIDS) and Application protocol-based IDS (APIDS) both protocols are used for enforcing the correct and legal use of a protocol. Typically they are on a web server monitoring HTTP(S) streams. And offer a better protection than filtering but at a trade-off of more computation on the web-server.

3 Data gathering and processing

An IDS, even the most powerful and advanced, needs something to fuel its algorithms. This something is obviously data coming from all over the IT ecosystem. These data come in plenty of different types and shapes, and from various hosts. This variety is what we are going to explain in this chapter !

3.1 Types of data

We will first begin by enumerating the biggest types of data that an IDS has to process and how it processes it. An IT ecosystem can produce hundreds of gigabytes of information that require automated processes if we want to be able to understand them and grasp all their potential. It is heterogeneous and thus produces multiple types of data, the more types an IDS can process the better it is. The core data types we are reviewing here are Network traffics, core hosts logs and applicative/customs logs.

3.1.1 Network traffic

In order to fully monitor an IT network, one should monitor its traffic. The traffic flowing through routers, switches, hosts and servers is very important to understand your network and detect early network attacks, intrusions, etc. All these complex networking data are represented via NetFlow on Cisco machines and via IP flow on other machines. Netflow allows you to capture traffic entering and leaving an interface and determine all sorts of networking specificities like congestion cause, traffic type, etc. Monitoring this protocol can detect DDoS attacks such as SYN-flood amongst many other attacks .
To go a bit further, we could take advantage of Software Defined Networking (SDN) for network monitoring. Instead of having to gather Netflow from all the routers and switches individually, we could gather this information from the SDN implementation directly to our IDS.

3.1.2 Hosts logs

Windows and Linux offer plenty of information on the system via their native logging mechanism. This logging mechanism produces logs called “Windows logs” in Windows, they are stored under the ”Event Viewer” app. In Linux, these logs are stored, in plaintext, under the “/var/log” folder.

There are three main logs (for clients) categories in the “Windows logs”. Linux logs are spread across multiples files (18) but we can map the more granular logs into theses three categories too :

- “Application logs” : Here are logged all the events created by an application running on the system or a service. In windows these logs are managed by the developers of the application. In linux these categories would contain httpd, mail, mysql, ...
- “Security Logs” : Here are logged events that are, by a way or another, related to the security of the system. These are defined in Windows by Microsoft and cover all the aspects ranging from logon attempts to file deletion. In Linux, this would engage logs files like faillog, auth.log, secure ...
- “System Logs” : Here are logged all the events generated by the core system itself. These events refer to drivers, the kernel, NTFS and all the low level protocols/components of the system. In linux log files like kern, cron, boot.log, ...

There are some windows logs categories specific to some server implementations too. These categories are Directory Service logs, DNS Server logs, file replication service logs.

In windows all these logs are marked with a certain severity level, this severity allows the user, or the IDS, to better understand the impact of the log on the system. In linux, the logs are more granular, so it is up to the user/IDS to mark specific logs files that are more critical than others.

Information logs are, as their names may say, only information about successful operations, when a task happened without raising error. This type of log is the most common on a system and it’s not necessarily a good idea to send all these data to your IDS. This amount of data may impact network performance badly for a small beneficial impact on security.

Warning logs, this severity announces that currently the event may not be a problem but this could be a problem in the future if this is not properly handled. This level becomes relevant because we better act when a warning occurs than when it is too late.

Error logs, this is obviously bad to have an error event because it has been brought up because an error actually happened. This kind of log is important and has to be sent to the IDS in order to raise alarms if a security threat is the cause of the error. Nonetheless, errors may not impact badly the system or the security of the OS and that is why we have to process this info and not raise an alarm for each error.

Critical logs, on the other hand means that there is really a big problem, that something crashed or has been compromised. This level should always be monitored and sent to the IDS.

As we know, a host is not only a compilation of software. It is a complex machinery, an agglomerate of hardware pieces. Monitoring these components can be very useful to detect certain kinds of attacks (than software can't). Metrics that can be gathered are for example CPU consumption, RAM consumption, Network usage, Disk usage or GPU consumption. This monitoring can be used in order to detect, for example, the over-expanding cryptojacking threat. This threat can be monitored by detected high usage of the CPU and GPU on the host. Ransomware could be detected as well by monitoring disk usage on the host, because the malware has to encrypt the whole disk the quicker it can, it will most likely use a hundred percent of the disk write speed.

3.1.3 Applicative logs

In an Enterprise IT network there is a need for additional tools in order to protect and monitor it ! All these tools produce logs and they are very useful in order to better understand the network. That's why an IDS must have compatibility to a vast majority of these applications. The two main applications protecting the network and producing useful logs are :

- Antivirus : When an antivirus produces a result it logs its findings into a proprietary language format. A result can be produced when a system scan is done (by logging the time it took, the number of files it scanned, if it found something, ...) or when a virus is detected (location, type or severity of the virus) more globally each time the AV produce an alert, this can be logged then sent to the IDS.
- Firewalls : When a threat is detected, the firewall logs the incident. This threat can be a remote connection from an unknown host, outgoing data to an unwanted network, DDoS, overall all kinds of network attacks/exfiltration.

3.2 Gathering of data

All this data comes from various hosts, and each host has one or multiple methods available to gather these logs. We will cover the gathering of logs from Windows hosts, with their proprietary protocols, to more open source ideas and protocols used, for example, in linux. All of the aforementioned require no additional softwares to gather information, but in some scenarios we want more powerful possibilities and that's where agents come into action.

3.2.1 Standardized protocols and Windows

In Linux and a vast majority of devices, the logs are created via a mechanism called Syslog. This mechanism writes by default the logs into plaintext files like explained before. But Syslog has a built-in mechanism to directly transfer (write) the logs remotely. Syslog not only defines a storing and writing mechanism but also defines the data structure of the so called "syslog logs". This protocol allows the host to send his logs to a remote server, in this case a compatible IDS, for further analysis.
In order to monitor the hardware configuration of the host, SNMP is an open source protocol that comes to mind. This protocol allows centralized host resource management. In the current scenario, the manager would be the IDS and the agent would be the hosts. SNMP works with a "Community" system, defined in the agent, that allows the authorized machines (manager/IDS) to poll authorized data (CPU, RAM consumption, etc.). SNMP is a non-secure protocol in it's primal state, that's why SNMPv3 have to be used in order to ensure confidentiality of the monitored host ressources.

This monitoring is done via WMI in windows and windows servers. WMI can query almost the same host resource as SNMP. WMI also allows, among other things, to detect the services running on the machine which can be interesting to know if a malicious service has been installed. WMI works through the use of a windows user with "COM" access rights (containing machine metrics). The IDS must then have the credits of this user to be able to connect to the server and retrieve the information. WMI does not only replace SNMP in windows but also forward the logs like Syslog would do it. In order to do that, the Windows user has to be granted the right to read the logs.

3.2.2 Agent

More advanced IDS can allow the user to implement a lightweight application on each host in order to gather information in a "All in one" type. These apps are also called agent, remote agent, and send data to the IDS in a secure way. This kind of agent simplifies the ecosystem by gathering and agglomerating all the information required into one unique secure mechanism. In the aforementioned case we would not have SNMP, Syslog, WMI on the network but we could only have HTTPS traffic for example.

4 Timeliness

Time is of critical importance for detecting and preventing intrusions on a network, some attacks are on a small time scale and others will spread to minimize detection. It's important to detect both overwhelming attacks and more insidious intrusions.

4.1 Time of detection

The detection can be done in real time (on-line) or not (off-line).

For real time detection, Ethernet packets are analysed as they come and apply some rules/filters or compared to previous models of normal traffic.

Similarly, off-line detection means the data is passed to a process that will do the same on stored traffic data.

In most cases, we want the system administrator to be informed of ongoing attacks so real time detection is the objective. For real time detection to be possible, the IDS needs enough resources to process the necessary data as fast as it's generated.

4.2 Granularity

It is possible to run continuous analysis of the data but the available resources will be a limiting factor. In addition, some attacks/intrusions that are made on a big time scale may be easier to detect in batch, depending on the size of the analysis window in continuous analysis. Batch processing is the sequential and automated processing of data batches.

4.3 Detection response

Depending on the gravity of the alert/warning, the response may vary. Warning sended by email to a system administrator is most common. In some cases immediate action could be beneficial, banning IP addresses linked to clear violation for exemple.

5 Detection Methodology

Among the detection methodologies, there exist three different ways to detect anomalies namely the signature-based, specification-based detections and anomaly-based.

5.1 Specification-based detection

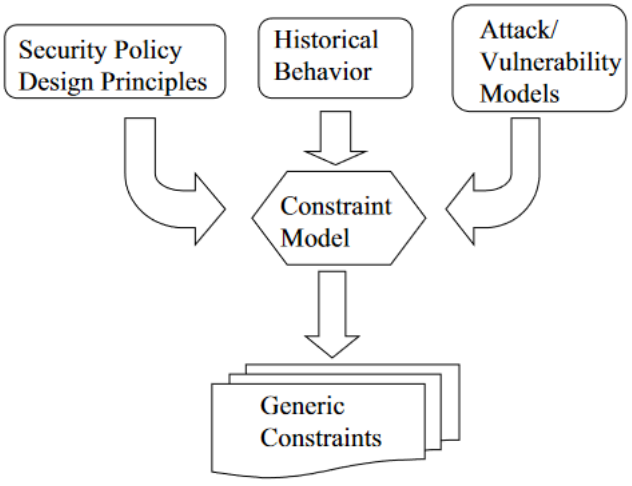
This methodology, also called "expert system" which is a system containing a set of rules, is more efficient in terms of memory because it must just contain the rules which will trigger an alert if a threshold is reached. Therefore, it works with a system of specification and, more precisely a hierarchy of specifications where we find generic specification and then, specific specification for certain protocols, applications or other things. For example, for applications accessible by a lot of people, the specifications are more precise to ensure to cover a bigger part of the attack surface. The architecture of the specification-based IDS is similar to the anomaly-based IDS (see later) but instead of the automatic model learning, all training part is manual.

To find efficient specifications, the human expert must collect previous data thanks to the monitoring to make a current network behavior profile and by this way, create adequate specifications in line with this profile. To help the human expert to find specifications, it can use Finite States Machines (FSM) methodology or some description languages like the Unified Modeling Language (UML).

We can see specifications as several types of constraint:

- Operational constraint: this constraint is based on the behavior of application, network, ... For example, if a user prints a document with a printer which should not print anything, an alert can be triggered. Or if a user wants to sign in to an application and fails more than three times its credentials, an alert is triggered.
- Access constraint: it restricts the access for files, directories, network ports, or on any object from users or programs.
- Resource usage constraint: this constraint limits the bandwidths, connections, memory, ...

We can see that this methodology is similar to ACL (Access Control List) and firewall rules present in network devices.



By this usage of specification, specification-based detection models could detect unknown and known anomalies. Known anomalies are covered because specifications are adapted for those there and unknown anomalies could be covered because generic specifications could mitigate them. Obviously, they don't mitigate all unknown anomalies, so we can say about the accuracy that the true positive is relatively good but not very high and the false negative is not negligible. By this way, we can conclude that the accuracy is medium.

However, the main problem of this methodology is that all of the constraints must be manually implemented and regularly updated to deal with new potential attacks, so it takes time but it's a good compromise between anomaly-based and signature-based detections if we want to minimize

the memory and detect unknown anomalies. Furthermore, this methodology needs human experts with a high level of knowledge.

5.2 Signature/misuse/knowledge-based detection

In the literature we can have 3 names for this type of detection namely misuse, knowledge and signature-based detection (SIDS). We will employ the signature name because it's the most common appellation and to stay coherent with our explanations. This methodology is based on pattern recognition of known indicators of compromise (IOC) using a database which contains all of them. These indicators are patterns of known attacks or intrusions detected in the past and they are also called "signatures". The idea is relatively simple, the signature-based IDS inspects the network traffic and compares the analyzed packets with the signatures database. If a correspondence with a signature occurs, an alert is triggered and in the case of an IPS, this one can respond by rejecting the specific malicious packets.

Therefore, the signature-based detection is simple to set up and does not require a lot of memory but it's really important to keep up-to-date the signature database because all of the security is based on this. Moreover, we can note about the accuracy that the false positive rate is rather low and the accuracy for true positives is perfect.

Unfortunately, this methodology is not adaptive in front of new types of attacks or intrusions not discovered yet (zero-day attacks) and also in some cases where it's possible that a known attack which is altered is not detected. Another important aspect is that this methodology is not scalable because, if we have a lot of traffic, inspection of all packets can be really complicated and sometimes impossible with the resource capabilities. For these reasons, the exclusive usage of SIDS is today less effective because of the number of attack variants and new attacks. To increase the effectiveness it would be a good idea to combine this methodology with another one like anomaly-based detection to detect a new attack and directly create a new signature based on this new attack. The figure 3 show the general process of a signature-based detection.

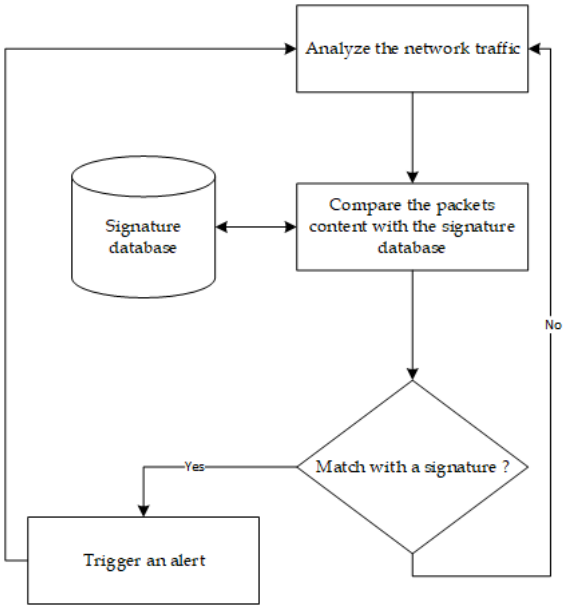


Figure 3: General process of a signature-based detection

We can also see an example of a signature (among a suite of signatures) that can be configured in Snort IDS proposed by the FBI to prevent of a possible attack on an OpenSSL heartbeat extension vulnerability:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET [25,443,465,636,992,993,995,2484] (msg:"SERVER-
OTHER OpenSSL TLSv1.2 heartbeat read overrun attempt"; flow:to_server,established; content:"|18 03
03|"; depth:3; detection_filter:track by_src, count 3, seconds 1; metadata:policy balanced-ips drop, policy
security-ips drop, ruleset community, service ssl; reference:cve,2014-0160; classtype:attempted-recon;
sid:30513; rev:5;)
```

Figure 4: https://us-cert.cisa.gov/sites/default/files/documents/FBI%20Private%20Industry%20Notice-140416-002.pdf

5.3 Anomaly-based detection

In this type of methodology, often abbreviated by AIDS, we can find a lot of possible models. Among these models we find two main types of models based on Statistics and Machine Learning techniques. All of these models are based on the fact that they learn patterns of normal behavior (training phase) to create a model and trigger an alert in the case of the behavior would be abnormal thanks to this model (detection phase), because this abnormal activity could be an anomaly and so, an intrusion.

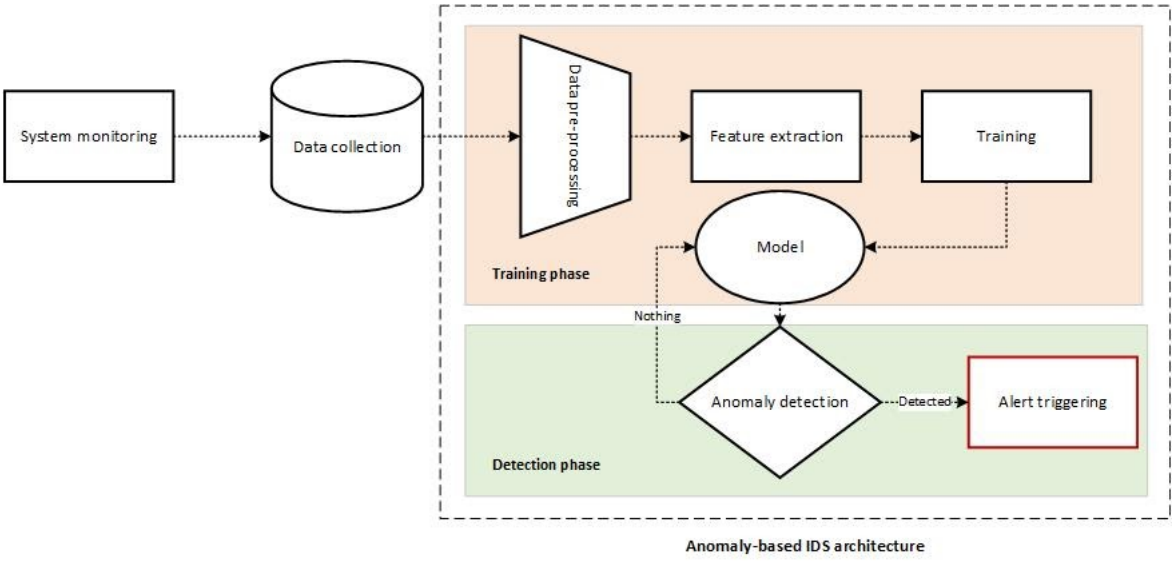


Figure 5: General process of an anomaly-based detection

As explained before, there are two types of models namely:

1. Statistical-based

   A statistical model is based on a distribution model to build the normal stochastic behavior profile. Statistical-based AIDS uses some statistical metrics like the mean, the median, the standard deviation and the mode to determine a score. The principal advantage of this model is that it can detect anomalies in real time. The idea is that the model compares the current behavior with the previous behavior and if a score exceeds a threshold, an alert is triggered. The statistical-based models often uses 3 types of models:

   (a) Univariate model: this type of model focuses on one variable in the profile behavior. It uses univariate distributions like a Gaussian.

   (b) Multivariate model: this type of model focuses on several variables in the profile behavior. In this case, the model will analyze relationships between at least 2 variables based on correlation measures. The big issue with this type of model is to build a reliable model when there are a lot of variables. Sometimes, to reduce the number of

variables, the usage of Principal Component Analysis can be efficient because it reduces the dimension of input vectors (less variables).

(c) Time series model: here this model focuses a series of observations based on several time intervals. Based on time series, the model will estimate the future behavior pattern in function of the past behavior profile and, if this pattern is different with a certain probability, an alert will be triggered.

2. Machine Learning

There are a lot of Machine Learning models. The idea of Machine Learning is to build a model thanks to the data learning and, more data is used during the training, more the model will correspond to the target normal behavior. Machine Learning models learn themselves and without explicit programmation. As explained before, these models are used in the training phase. During the training phase, the important variables (features) which influence the most the profile behavior are extracted. To select the best features it's relatively difficult and, for this reason, machine learning models aren't perfect but it is possible to use some algorithms to try to reach the best solution like PCA (already mentioned) or t-SNE. Once the features are extracted, you must select a Machine Learning algorithm which will learn the normal behavior of the system. You can decide to use 3 main approaches namely Unsupervised learning, Supervised learning and Ensemble Methods. There exists also Semi-Supervised learning but this kind of model is less popular. In fact, if your data are labelled, you can use the Supervised learning models and if it's not the case, you may use Unsupervised learning models. Ensemble methods use a combination of same models or different models to increase the accuracy taking advantage of the law of large numbers but it takes more time and more resources to train. In the Supervised Learning, we find Neural Network or SVM models and in the Unsupervised Learning, we find Clustering or Auto Encoders (AEs) models.

To improve the learning in Supervised Learning, and then the model, you can use some statistical tools like entropy or estimators (Mean Square Error) combined with some algorithms like the Gradient Descent or the Least Square method to make a model which should correspond at best at the normal behavior. In the case of Unsupervised Learning, you must have human feedback.

The main principal advantage of Machine Learning models is that they can be more adapted for complexe behavior and they are more powerful than other methods if there are a lot of data.

The main advantages of AIDS are the possibility to detect new kinds of attacks like zero-day attacks or modified known attacks and the capability to create new intrusion signatures which can be used in combination with SIDS to optimize the detection system. Furthermore, AIDS doesn't require a lot of knowledge compared to Specification-based IDS. AIDS can also highlight some complexe patterns of intrusion undetectable by SIDS and Specification-based IDS.

However, we can also notice some issues about AIDS. The most difficult thing with AIDS is to create a good model. We can say that it's impossible to make a perfect model capable to detect all intrusion without errors because these models are built with a dataset reflecting a specific behavior at a specific moment of the network traffic. It means that it could integrate some mistakes and noises which can lead to some biases. Moreover, these datasets are often forged and don't correspond to the real behavior of the target network traffic. These are generic dataset and, often attack patterns present in these datasets are not specific for this network traffic or just avoid to train the model with some other kind of known attacks. By this way, we can say that the generated false positive alarms will be relatively high compared to other methodologies like SIDS and Specification-based. Secondly, because of the previous reason, we can also say that these models are not very adaptative in dynamic systems. Thirdly, some kinds of attacks seem legitimate and in this situation the AIDS cannot detect any anomalies. Therefore, we can say that the rate of false negatives could be potentially high. Fourthly, for resource consumption, AIDS consumes more memory, and it could take more time to train models and detect some intrusion compared to the two other methodologies. Because of the resources consumption, AIDS models will not

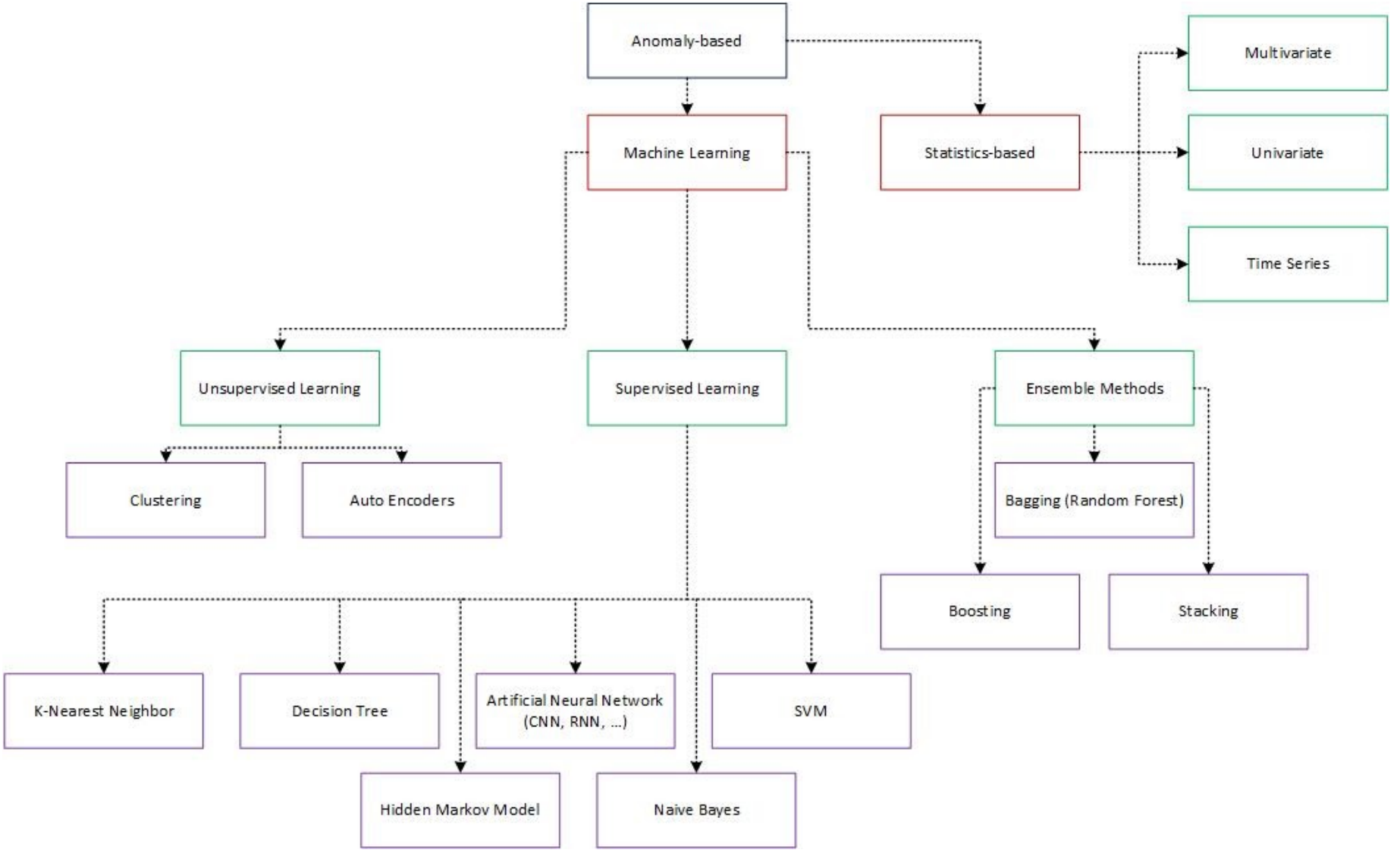


Figure 6: Types of anomaly-based models

be adapted for IoT. Fifthly, most of the time, the models will not be adapted for the real time detection. Finally, if the traffic is encrypted, it could be really difficult to detect any anomalies (in the case of NIDS implementation for example).

5.4 Hybrid detection

Firstly, we could combine anomaly-based and signature-based detections. This approach allows to recognize known normal behavior and in the case where the pattern is unknown, the anomaly-based detection can classify it. By this way, it allows to adapt and automatically create new specifications thanks to the Machine Learning algorithm.

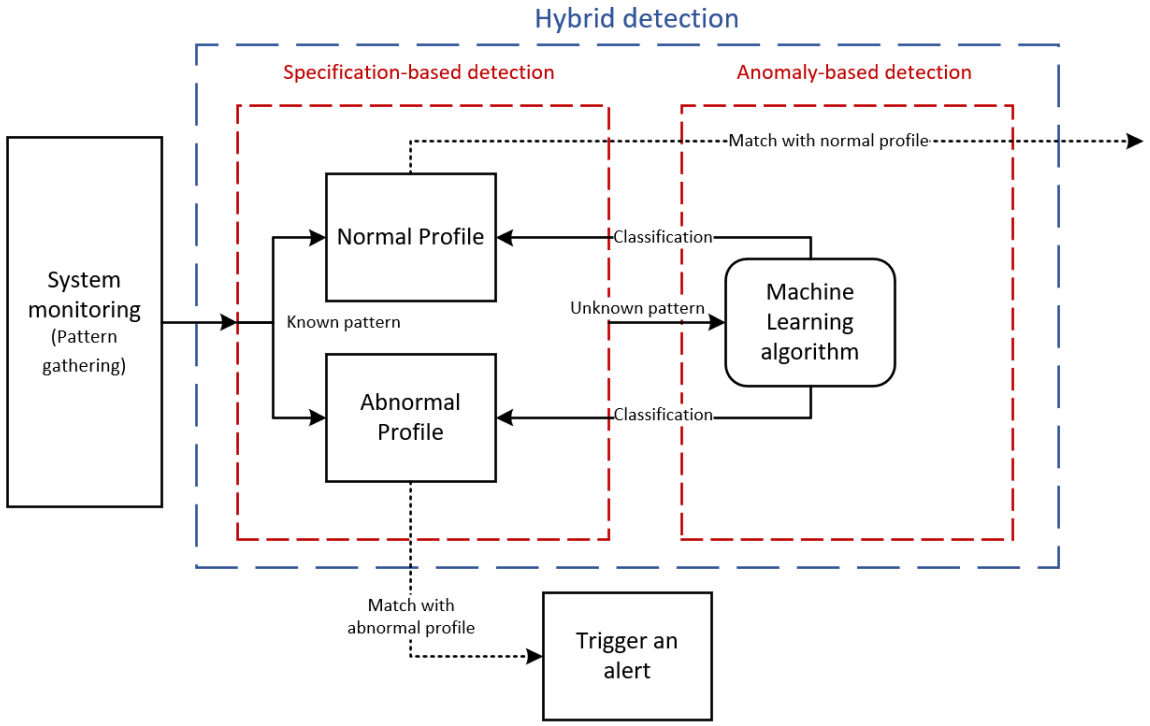


Figure 7: Architecture of hybrid detection methodology

Secondly, it is also possible to combine anomaly-based detection with signature-based detection. The interesting thing with this combination is that we can directly generate a new signature if an anomaly is detected into the anomaly-based detection.

5.5 Comparative table

Methodology	Memory Consumption	Accuracy	Unknown anomaly detection	Need constant updates	Network traffic adaptation
Specification-based	Less	Medium	Could	Sometimes manually	Yes
Signature-based	Depends of the database size	Perfect	No	Yes	Yes
Anomaly-based	High	Depends (not perfect)	Yes	No	No

5.6 Performance Metrics

All IDS models can be analyzed in function of their performance. To evaluate their performance, we can use some different evaluation metrics. All of them are based on the following confusion matrix:

Classified as	Malicious	Normal
Actual Class		
Malicious	True positive	False negative
Normal	False positive	True negative

- True Positive (TP) : it means that the IDS evaluate the pattern as malicious and that this pattern is effectively malicious.
- False Negative (FN): it means that the IDS evaluate the pattern as normal and that this pattern is malicious. That's what we want to avoid.
- False Positive (FP): it means that the IDS evaluate the pattern as malicious and that this pattern is normal. It's not really a problem because it's not a malicious pattern but it will generate more alerts.
- True Negative (TN): it means that the IDS evaluate the pattern as normal and that this pattern is effectively normal.

Once all of the behavior responses are classed, it is possible to determine evaluations metrics. Among these one, the most used and useful are :

1. The True Positive Rate (TPR), also called "detection rate", is the percentage of the patterns classified as malicious and which are indeed malicious. Detection rate = $\frac{TP}{(TP+FN)}$
2. The True Negative Rate (TNR), also called "Recall", is the percentage of the pattern correctly classified as normal compared to the total of normal classified patterns.TNR = $\frac{TN}{(TN+FN)}$
3. The False Positive Rate (FPR), also called "false alarm rate", is the percentage of the patterns classified as malicious while the patterns are a normal behavior. FPR = $\frac{FP}{(FP+TN)}$
4. The False Negative Rate (FNR) which represents the percentage of patterns identified as normal while the patterns are malicious. FNR = $\frac{FN}{(FN+TP)}$
5. The Accuracy which represents the percentage of patterns correctly classified. It's the total of correct predictions divided by the total of predictions. Accuracy = $\frac{(TP+TN)}{(TP+TN+FP+FN)}$
6. The Precision represents the percentage of relevant results instead of the irrelevant results. Precision = $\frac{TP}{(TP+FP)}$
7. The F-measure, also called "F1 score", gives the performance of the combined Recall and Precision evaluation metrics, it's the harmonic mean of both. It provides the system capacity to give relevant results and refuse the others. F-measure = $\frac{(2*Recall*Precision)}{(Recall+Precision)}$
8. The Error Rate expresses the rate of error about the intrusion detection. Err = $\frac{(FP+FN)}{(FP+FN+TP+TN)}$
9. The Receiver Operating Characteristic (ROC) Curve is a graphic which shows the performances of a classification model for all of classification thresholds. So it takes into account the TPR and the FPR. Ideally, we want to have a ROC value at 1. It's a good evaluation metric to evaluate a specific model using only one dataset but it's less relevant if we want to compare several datasets.

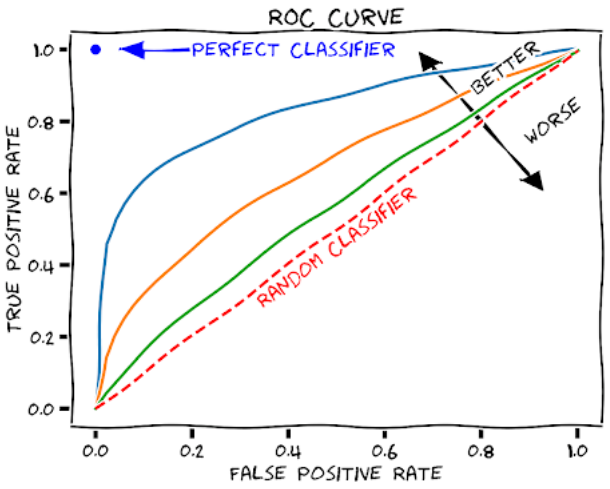


Figure 8: Graph of Receiver Operating Characteristic (ROC) Curve

6 Legal aspects

Today, a lot of countries have specific legislation concerning data protection and personal data. Personal data mean "any information relating to an identified or identifiable natural person ('data subject')" but this definition can change depending on the country's legislation. For example, in the world, we find Canada's Privacy Act and Personal Information Protection and Electronic Documents Act (Canada), Australia's Privacy Act, the United States' Safe Harbor Program or the United Kingdom's Data Protection Act. At the European level, data protection and personal privacy are ensured by the GDPR regulation, the NIS directive and the Cybersecurity Act. At the National level, some additional laws exist in some European countries. To improve the ethical aspect of artificial intelligence, the European Commision has recently proposed a new regulation to mitigate the possible high risk about the fundamental rights of natural people when a system uses an artificial intelligence. This regulation is therefore mainly destined for Machine Learning models and then, in our case, applicable for some anomaly-based detection systems. Due to the fact that IDS can treat a lot of data like pictures, sounds, biometric information, sensitive information or just personal information like IP address, it means that these intrusion detection information fall under the legislation on data protection and personal privacy.

6.1 Data exfiltration

These rules about data privacy are sometimes restrictive for enterprises which want to export their data to be analyzed by other businesses present in another country, especially outside of Europe for European countries. For this reason, it's often difficult to collect IDS data collections generated by some enterprises victim of attacks in order to create new IDS models adapted for new types of current attack.

However, the main aspect for IDS development is the information sharing to get quickly all information about new intrusions methodology and to adapt all IDS strategies for technologies which use them, like SIEM.

7 Conclusion

The evolution of technologies evolves quickly and that the reason why IDS must adapt more and more. A lot of new IDS models are created to respond to the new attack surfaces in various domains like IoT. We also have seen that there exists a lot of IDS types, by their processing strategy, their implementation, their temporal treatment and their data type.

Furthermore, the lack of real data slows down the development of powerful IDS to respond to the new attacks that evolve rapidly. This lack of data is due to the fact that enterprises don't want to give their information about internal operations, but also because of the existing laws on the personal data as explained in the chapter on the legal aspect. For this reason, the majority of IDS are modelised thanks to simulated attacks through forged packets. It is an important challenge that, in the future, communication and information sharing between enterprises are more present to protect all of us ensuring the respect of data privacy. In this optic, the homomorphic encryption could help to ensure the data privacy during IDS processing.

We can also say that IDS are not infallible and then, de facto, they can't certify 100% of protection against potential intrusions.

References

[1] Johnston S.R. (2001) The Impact of Privacy and Data Protection Legislation on the Sharing of Intrusion Detection Information. In: Lee W., Mé L., Wespi A. (eds) Recent Advances in Intrusion Detection. RAID 2001. Lecture Notes in Computer Science, vol 2212. Springer, Berlin, Heidelberg.

[2] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance)

[3] Sgaglione, Luigi Coppolino, L. D'Antonio, Salvatore Mazzeo, Giovanni Cotroneo, Domenico Scognamiglio, Andrea. (2019). Privacy Preserving Intrusion Detection Via Homomorphic Encryption. 321-326. 10.1109/WETICE.2019.00073.

[4] https://digital-strategy.ec.europa.eu/en/library/proposal-regulation-laying-down-harmonised-rules-artificial-intelligence

[5] Uppuluri P., Sekar R. (2001) Experiences with Specification-Based Intrusion Detection. In: Lee W., Mé L., Wespi A. (eds) Recent Advances in Intrusion Detection. RAID 2001. Lecture Notes in Computer Science, vol 2212. Springer, Berlin, Heidelberg.

[6] Ko C., Brutch P., Rowe J., Tsafnat G., Levitt K. (2001) System Health and Intrusion Monitoring Using a Hierarchy of Constraints. In: Lee W., Mé L., Wespi A. (eds) Recent Advances in Intrusion Detection. RAID 2001. Lecture Notes in Computer Science, vol 2212. Springer, Berlin, Heidelberg.

[7] Khraisat, A., Gondal, I., Vamplew, P. et al. Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecur 2, 20 (2019).

[8] A Survey on Anomaly Based Host Intrusion Detection System, Shijoe Jose1, D. Malathi1, Bharath Reddy1 and Dorathi Jayaseeli1. Journal of Physics: Conference Series, Volume 1000, National Conference on Mathematical Techniques and its Applications (NCMTA 18) 5–6 January 2018, Kattankulathur, India

[9] Intrusion Detection Systems for IoT: opportunities and challenges offered by Edge Computing, Pietro Spadaccino, Francesca Cuomo.

[10] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, E. Vázquez, Anomaly-based network intrusion detection: Techniques, systems and challenges

[11] Otoum, Y., Nayak, A. AS-IDS: Anomaly and Signature Based IDS for the Internet of Things. J Netw Syst Manage 29, 23 (2021).

[12] Stakhanova, Natalia Basu, Samik Wong, Johnny. (2010). On the symbiosis of specification-based and anomaly-based detection. Computers Security. 29. 253-268. 10.1016/j.cose.2009.08.007

[13] Kumar Ahuja, Dr. Gulshan. (2015). Evaluation Metrics for Intrusion Detection Systems-A Study. International Journal of Computer Science and Mobile Applications. 11

[14] Snehal Boob, Priyanka Jadhav; Wireless Intrusion Detection System. 2010, International Journal of Computer Applications (0975 – 8887)

[15] M.A. Kumara, C.D. Jaidhar, Hypervisor and virtual machine dependent Intrusion Detection and Prevention System for virtualized cloud environment, in: 2015 1st International Conference on Telematics and Future Generation Networks (TAFGEN), IEEE, May 2015, pp. 28-33.

[16] ManageEngine Windows Log type explanation

[17] Ubuntu Linux Log type explanation

[18] John Vacca, Computer and Information Security Handbook (3rd Edition), May 2017