Intrusion Detection Systems

Computers and electronics have slowly been consuming our lives. With each passing year, we humans place more responsibility on computer systems, causing us to house increasingly sensitive data on our machines and networks. This is a problem because there is always someone else who wants our information, and our information is not secure just by itself. Many software systems back at the conception of Intrusion Detection Systems (IDS) are greatly flawed with security issues. It is not as simple as locating the security flaws and fixing them because that method is not economically viable, but also virtually impossible to have one-hundred percent security coverage. Even if a system was one-hundred percent secure from a software standpoint, the users of the systems, humans, pose some of the greatest risks. Computer experts at the time needed a way to observe any and all activity on their local machines and networks and be alerted when intruded upon. In order to combat this issue, two computer security pioneers in the 1980's, Dorothy Denning and Peter Nuemann, created a model detection system which was first termed, Intrusion Detection Expert System, which is the basis of what we use today [Denning p1].

Intrusion Detection Systems (IDS) are just like they sound. The system alerts an entity, usually an administrator and a logging software, when an intrusion or any abnormal activities are detected. A pure IDS does not take any steps to mitigate the risks of the intrusion, the system only reports the matter. The basis of any Intrusion Detection System is as follows [Stallings]:

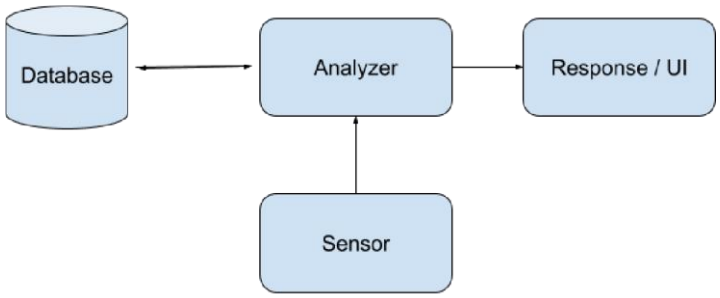
- Detect an intrusion as early as possible to reduce the chance of prolonged damage; the faster detection occurs, the faster the intruder can be kicked from a system.
- Collect any and all information on the intrusion; this allows for more information on intrusion techniques, and subsequently the information can be used to strengthen future security methods.
- Alert the appropriate administrators that an intrusion has been detected along with any collected information

An *intrusion* can be represented as many things. Broadly, an intrusion can be described as any event or activity that is unauthorized or viewed as suspicious by the system. A few examples of detected abnormal behavior listed in the model IDS are as follows:

[Denning]

- *Brute force attacks:* When a user is attempting to brute force their way into an account, the system will detect an abundance of failed login attempts and alert the administrator.
- *Abnormal behavior across users*: in the event that a user was successful in accessing an authorized users account, the system can compare past user actions to current user actions. This means if the intruder is performing activities not consistent with the past activities of the authorized user, the IDS can alert an administrator.

- *Legitimate users putting the system at risk:* a system is only as secure as its least secure individual, so it is important to monitor our own as well. Legitimate users may attempt to access security methods in an operating system, or click links to possible viruses in which the IDS would be able to detect if previously known. Intrusion Detection Systems, whether physical or software, are usually made up of four basic components: *sensors, analyzers, knowledge database,* and a *user-interface. Sensors* can only interpret data that passes through it. The sensors collect information, like network packets and system call traces, as inputs and send them off to be reviewed by an *analyzer*. The analyzer is able to receive information from sensors and other analyzers as input. The analyzer takes the input, represented in binary code, and compares it to a knowledge database housing a wide range of known suspicious and malicious byte sequences[Tawatia]. If there is a match between the knowledge database and the analyzer input, the analyzer will send a response saying that there



Basic Function of IDS Components
Based on diagram from (Tewatia, Mishra)

has been an intrusion. If there is no match then the system perceives the action as being authorized and will allow it to continue without alarm.

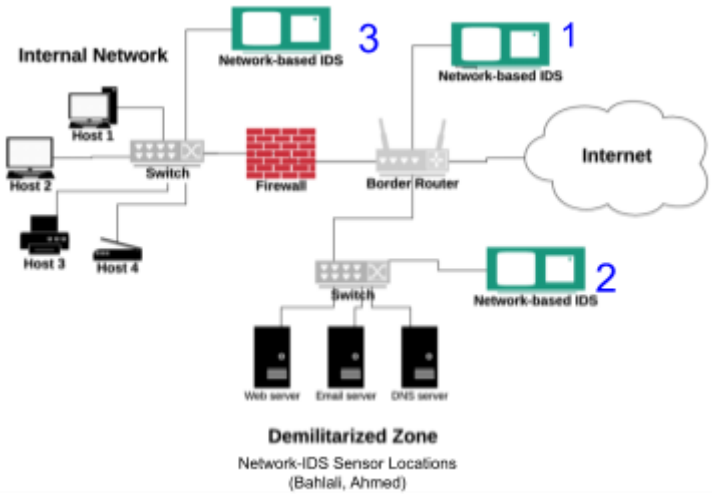
Intrusion Detection systems vary in the scope of their detection. Two main implementations of Intrusions Detection Systems are *Host-based* and *Network-based*. Before networks were as prevalent as they are now, administrators needed to keep their local machines safe. Host-based Intrusion Detection Systems are the earliest implementation of Denning's IDS model which allowed for the administrators to detect any unauthorized changes to a machine. When first configured on the machine the host-IDS takes an image of the local machine. The IDS watches inbound and outbound packets from the local computer. This is the main advantage of a host-based IDS, as they are able to monitor both incoming and outgoing packets, unlike a network based IDS [Stallings]. The IDS then will compare the image to the current state of the local machine and determine if any critical information or crucial operating systems files were manipulated or deleted. If so, the administrator will be alerted. Host-based IDS's are used on any machine that is not supposed to be manipulated and are only supposed to do specific activities. Essentially a host-based IDS is an inventory of system files that are not supposed to be messed with. This allows for administrators to know when an individual creates a backdoor in a system, changes operating system files, or when a user performs an action many times, like attempting to login. Since computers are often connected to multiple networks at a time, only having host-based IDS would not be sufficient to keep an organization safe. This is when network-based IDS come into account.

Network-based intrusion detection systems analyze network traffic. The area of the network it is analyzing depends on the placement of the Network-IDS sensors. There should be multiple network-IDS sensors in a system and each analyze different

subnets. The function of the network-based IDS is the same as the host-based except its input is from the network packets that travel through its sensor. The sensor observes packet traffic across the subnet, compares traffic information to a database of signatures, and if the signatures match the system alerts the administrator. It is important that the network-IDS sensors are placed in strategic positions across the network because the sensors can only gather information that crosses their path. This is why it is important for most subnets on the network to have a sensor. The most important locations to include network sensor would be [Bahlali]:

1. *Outside of the organization's firewall:* This not only allows for the administrator to see if anyone is attempting to break in from outside of the organization, but also gives the organization an idea of expected attacks and the amount of possible attacks to suspect. If there are a lot of intrusion attempts on the system it will tell the admins and the organization owners how at risk their system is.
2. *Inside the firewall but before the network's internal firewall:* This is considered to be the “Demilitarized Zone,” or DMZ. The DMZ houses the external servers used by an organization like DNS, web servers, and email servers . The DMZ sensor sits between the main firewall and the Internet Service Provider router.
3. *Behind internal firewalls:* When an IDS sensor is placed behind an internal firewall, administrators can detect any attempts that are made to crack the firewall. In some cases, there are many internal firewalls and the firewalls usually have different purposes. Placing a network-based IDS behind

internal firewalls ensures that the system is monitoring its own users actions.



Most organizations will aim to use a Hybrid Intrusion Detection which combines the Network and Host IDS approaches. This is the recommended set-up so that the detection systems have the best chance at catching suspicious or unauthorized activity across the subnets and the local machines as well. Intrusion detection systems are best used in tandem to broaden the scope of the coverage, but also to maximize the detection methods. There are two standard detection methods used by most systems, *Signature-based* and *Anomaly-based*.

Signature-based detection methods use a series of pattern matching algorithms to compare current activity patterns to a large knowledge database of signatures. These signatures can be represented as strings, frequently targeted port numbers, or inconsistent header conditions [Figueroa]. When current activity signatures match known malicious signatures an intrusion is detected. This method can only detect

previously known attack procedures and is susceptible to unknown or innovative attacks. To counteract this disadvantage, anomaly detection was invented.

Anomaly detection methods aim to adapt and detect previously unknown attacks. This method looks to find suspicious activity within the system based on a set of rules called trust models [Veeramreddy]. If activity is found to be out of the bounds of the trust model, the action will be alerted. These trust models can be set by an administrator and then adjusted over time using machine learning techniques. This allows the system to learn after each attack and make adjustments based on the intrusion information. Since an anomaly based system is attempting to predict authorized behavior versus suspicious behavior, this method can detect many *false positives*, behavior that is flagged as suspicious, but it is actually authorized activity. If an administrator experiences many false positives, the trust model may be too strict. On the other hand, if the trust model is too lenient there may be an increase in *false negatives*, suspicious activity that is recognized as authentic. A way to decrease false responses in the system is to enact a hybrid detection system using both *signature-based* and *anomaly-based* techniques.

Although Intrusion Detection Systems are incredibly useful tools that provide a great line of defense, they only intend to alert administrators about intrusions that have already occurred, there are no measures in place for the system to mitigate any harm done to the system. In some cases the detection system might report the best ways to reduce risk after an intrusion, but it is still up to the administrators to take counteractive measures rather than the system itself. This can leave the system vulnerable for extended periods, or at least until an administrator notices the alert and takes action

against it. Luckily, there are current systems in place that take preventative measures when an intrusion is detected. Intrusion Prevention Systems (IPS), today sometimes referred to as Intrusion Detection and Prevention Systems (IDPS), were created to address this issue.

IDPS follow the same basic principles and detection methodology that intrusion detection systems follow, with a few added concepts. The prevention system will not only alert an administrator, but it will also use a range of response techniques to prevent the attack from progressing [Base, et-al]. A few techniques include cutting off any network connection that is being used as an attack vector, reconfigure security environments to allow firewalls to block access from the attackers location, and even change the content of the attack itself. When the system can detect harmful content or attachments contained in a file or email, the system can change the content, allowing the attack to go through, but only the content that is determined safe will be allowed [Base, et-al]. IDPS use both signature and anomaly based detection and an additional, stateful protocol method.

The stateful protocol is similar to anomaly based methods, but it is not based on any host or network rulesets, instead the protocol uses universal profiles supplied by large technology vendors to compare observed user activity against. The universal profiles are all considered to be safe and allow for the system to notice any behavior digression in the system. The protocol is considered stateful because the system can follow the condition of the system which allows it to keep track of actions and can recognize when certain commands are being used too frequently [NIST p4]. Profiles can change depending on the vendor, as can results. The vendor sets the rules of the

profiles causing there to be different outcomes per vendor. State tracking offers a great deal of security, but it also uses a lot of resources and can be a burden to the performance of the system.

Today, organizations use intrusion detection and prevention systems to best secure their networks. The duality of the two systems allows for the most coverage of a network, but the IDPS cannot stop every attack. Experienced attackers use different evasion techniques to subvert the system and fly under the radar. Two main techniques are Distributed Denial of Service (DDOS) attacks and fragmentation efforts. DDOS attacks occur when an attacker sends an abnormal amount of traffic through a network, using up all the network's resources causing it to shutdown. This can be executed by sending a large amount of harmless actions through a network. Due to the actions being harmless the system will not suspect any bad behavior, allowing for the packets to be accepted, resulting in a DDOS attack [NIST p.5]. Another well known technique known as *fragmentation* involves the attacker splitting the packets into unrecognizable portions that will not be matched to any signature in the knowledge database [Tsung-Huan, et-al p.2]. IDPS can prevent this attack from happening by joining the fragmented packets together upon arrival, but if the system cannot due to limited resources, the packets will be perceived as safe.

Intrusion detection systems have come a long way and will continue to progress with the help of machine learning. When paired with machine learning these systems can adapt on the fly and alter trusted behavior models based on perceived activity. The systems can also determine patterns of actions based on the classification of users. An administrator will perform different actions than a regular user and the system can

determine the differences between the two. The adaptation of trust models in real time should allow the system to respond to less false-positives and decrease false-negatives. While the advancements in IDPS revolve around machine learning, scientists are also using the human immune system as a model to create algorithms that replicate our immune system. These algorithms will be used in detection systems to detect new patterns and attempt to find a solution [Hooks, et-al]. Just like the human immune system the artificial immune system will be able to determine whether a byte sequence is dangerous, and it will also need the ability to refer to previous attacks as to best classify the event and form a solution. This method is only theoretical and being experimented on Local Area Networks and there is some debate on the scalability of this venture [Hooks, et-al]. Replicating the human immune system is an onerous task with an unbelievable amount of challenges ahead of itself, but the payoff would be spectacular.

Intrusion detection systems have come a long way, but they still have a ways to go in order to adapt to the ever changing computer security environment. Intrusion detection systems are best implemented with the use of prevention systems, so that when the administrator is alerted there are also preventative measures happening in the background to keep the network safe. It is important that modern networks take all necessary security measures to protect the system. Intrusion detection systems are extremely effective, offer a level of deterrence, and provide crucial information on the health of the system. The future of IDPS is expansive and relatively demanding, but if artificial immune system-based detection systems prove to be possible, the payoff would be indescribable.

Bibliography

- Denning, Dorothy E. “An Intrusion-Detection Model.” *1986 IEEE Symposium on Security and Privacy*, Vol., no. SE-13, ser. No.2, Feb. 1987. *No.2*, https://doi.org/10.1109/sp.1986.10010 ; https://www.cs.colostate.edu/~cs656/reading/ieee-se-13-2.pdf

- Stallings, William. “21.2 Intrusion Detection Systems Chapter.” *Cryptography and Network Security: Principles and Practice*, 8th ed., Pearson Education, Inc., Hoboken, NJ, 2019, pp. Chapter 21.2.

- Tewatia, Rajni, and Asha Mishra. “Introduction to Intrusion Detection System: Review.” International Journal of Scientific & Technology Research Volume 4 Issue 5, May 2015. https://www.ijstr.org/final-print/may2015/Introduction-To-Intrusion-Detection-Syste m-Review.pdf

- [8] Bahlali, Ahmed Ramzi. (2019). Anomaly-Based Network Intrusion Detection System: A Machine Learning Approach https://www.researchgate.net/publication/339032478_Anomaly-Based_Network_Intrusion_Detection_System_A_Machine_Learning_Approach

- [9] Aldwairi, M., Abu-Dalo, A.M. & Jarrah, M. Pattern matching of signature-based IDS using Myers algorithm under MapReduce framework. EURASIP J. on Info. Security 2017, 9 (2017). https://doi.org/10.1186/s13635-017-0062-7

- Figueroa, Chris. *Intrusion Detection Systems Overview*, http://www.infosecwriters.com/articles/2015/11/09/intrusion-detection-systems - overview Accessed 15 Nov. 2021.

- Veeramreddy, Jyothsna & Prasad, Koneti. (2019). Anomaly-Based Intrusion Detection System. 10.5772/intechopen.82287. https://www.researchgate.net/publication/333874867_Anomaly-Based_Intrusion_Detection_System

- Bace, R. and Mell, P. (2001), Intrusion Detection Systems, Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD (Accessed November 15, 2021) https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=901146

- Tsung-Huan Cheng, Ying-Dar Lin, Senior Member, IEEE, Yuan-Cheng Lai, and Po-Ching Lin, Member, IEEE http://speed.cis.nctu.edu.tw/~ydlin/pdf/Evasion_Techniques_Sneaking_through_Your_Intrusion_Detection_Prevention_Systems.pdf

- Aickelin, Uwe, et al. "Immune System Approaches to Intrusion Detection - A Review." *SSRN Electronic Journal*, 2004, https://doi.org/10.2139/ssrn.2832021 https://arxiv.org/ftp/arxiv/papers/1305/1305.7144.pdf

- D. Hooks, X. Yuan, K. Roy, A. Esterline and J. Hernandez, "Applying Artificial Immune System for Intrusion Detection," *2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService)*, 2018, pp. 287-292, doi: 10.1109/BigDataService.2018.00051. https://ieeexplore.ieee.org/document/8405726