**Báo cáo IPDS**

1. **Một số loại tấn công mạng phổ biến**
   1. **Tấn công từ chối dịch vụ phân tán (DDos – Distributed reflection Denial of Service):**

* Mục tiêu là làm quá tải tài nguyên của hệ thống (máy chủ, dịch vụ, mạng) bằng cách gửi lượng lớn yêu cầu từ nhiều nguồn khác nhau, khiến hệ thống không thể phục vụ người dùng hợp lệ.
* Phát hiện: IDPS có thể nhận diện lưu lượng mạng bất thường, chẳng hạn như lưu lượng cao đột ngột từ nhiều nguồn khác nhau tấn công vào một mục tiêu.
* Phòng chống: Hạn chế băng thông cho các nguồn nghi vấn, chặn lưu lượng từ các IP có dấu hiệu bất thường, hoặc sử dụng kỹ thuật lọc lưu lượng.
  1. **Tấn công cơ sở dữ liệu (SQL Injection):**
     + Tấn công này khai thác lỗ hổng trong các ứng dụng web để chèn mã SQL độc hại vào các truy vấn cơ sở dữ liệu, cho phép kẻ tấn công truy cập, sửa đổi hoặc xóa dữ liệu.
     + Công cụ: sqlmap, sử dụng các payload đơn giản như ' OR 1=1 --.
     + Phát hiện: IDPS có thể phát hiện các chuỗi ký tự hoặc mẫu truy vấn SQL độc hại trong các yêu cầu HTTP đến máy chủ web.
  2. **Tấn công khai thác lỗ hỏng bảo mật - Exploit attack**
     + Đây là kiểu tấn công khai thác lỗ hổng bảo mật chưa được biết đến hoặc chưa được vá bởi nhà cung cấp phần mềm.
  3. **Tấn công mật khẩu – password attack**
     + **Brute force attack – tấn công dò mật khẩu:** Đây là một kiểu tấn công mà kẻ tấn công thử mọi kết hợp có thể của mật khẩu hoặc khóa để bẻ khóa tài khoản hoặc mã hóa.

Phát hiện: IDPS có thể giám sát các nỗ lực đăng nhập thất bại liên tiếp từ một địa chỉ IP cụ thể. Nếu phát hiện quá nhiều lần thử không thành công trong một khoảng thời gian ngắn, IDPS có thể đánh dấu đây là một cuộc tấn công brute force.

* + - **Dictionary Attack (tấn công từ điển**): là một biến thể của Brute Force Attack, tuy nhiên kẻ tấn công nhắm vào các từ có nghĩa thay vì thử tất cả mọi khả năng, nhiều người dùng có xu hướng đặt mật khẩu là những từ đơn giản VD: password, motconvit,… -> tỉ lệ thành công cao hơn.
    - **Key Logger Attack (tấn công Key Logger):** ưu lại lịch sử các phím mà nạn nhân gõ, bao gồm ID, password hay nhiều nội dung khác. Nguy hiểm vì dù có đặc mật khẩu phức tạp cũng không tác dụng.
  1. A diagram of a computer hacker

     Description automatically generated**Tấn công Man-in-the-Middle Attack:** Tấn công này xảy ra khi kẻ tấn công chen ngang vào giao tiếp giữa hai bên (ví dụ: giữa người dùng và trang web), cho phép họ đọc, thay đổi hoặc chặn dữ liệu trao đổi.

Phát hiện: IDPS có thể phát hiện các thay đổi bất thường trong lưu lượng mạng, chẳng hạn như các thông báo giả mạo ARP, DNS hoặc SSL/TLS.

* **ARP (Address Resolution Protocol) là giao thức được sử dụng để chuyển đổi địa chỉ IP thành địa chỉ MAC trên mạng nội bộ. Kẻ tấn công có thể giả mạo thông báo ARP để đánh lừa các thiết bị trong mạng gửi dữ liệu cho kẻ tấn công thay vì thiết bị hợp pháp.**
* **DNS (Domain Name System) là hệ thống dịch địa chỉ IP thành tên miền (ví dụ: chuyển 192.168.1.1 thành example.com). Kẻ tấn công có thể gửi các thông báo DNS giả mạo để chuyển hướng người dùng đến một trang web lừa đảo thay vì trang hợp pháp.**
* **SSL/TLS (Secure Sockets Layer/Transport Layer Security) là giao thức bảo mật được sử dụng để mã hóa dữ liệu khi truyền tải giữa người dùng và máy chủ. Kẻ tấn công có thể giả mạo hoặc thay thế chứng chỉ SSL/TLS để nghe lén hoặc chiếm quyền truy cập vào kết nối bảo mật.**
  1. A computer screen with text and icons

     Description automatically generatedTấn công giả mạo – Phishing attack: Phishing là hình thức tấn công hacker giả mạo thành một đơn vị/cá nhân uy tín để chiếm lòng tin của người dùng, thông thường qua email. Mục đích của tấn công giả mạo (phishing) là đánh cắp dữ liệu nhạy cảm như thông tin thẻ tín dụng, mật khẩu, đôi khi phishing là một hình thức để lừa người dùng cài đặt malware vào thiết bị của họ

Phát hiện: IDPS có thể nhận diện các URL hoặc nội dung trong lưu lượng mạng liên quan đến các trang web lừa đảo hoặc giả mạo.

* 1. A diagram of a computer network

     Description automatically generated**Tấn công chiếm đoạt phiên (Session Hijacking attack)** : Tấn công chiếm đoạn phiên (Session Hijacking) là một loại tấn công mạng trong đó kẻ tấn công chiếm quyền điều khiển một phiên kết nối đang diễn ra giữa hai hệ thống, chẳng hạn giữa một người dùng và máy chủ.

Mục tiêu của tấn công chiếm đoạn phiên là giành quyền truy cập vào thông tin, tài khoản, hoặc tài nguyên mà người dùng hợp pháp đang sử dụng.

* 1. **Social Engineering:** Đây là hình thức tấn công dựa vào việc lừa gạt con người, như dụ dỗ họ cung cấp thông tin nhạy cảm hoặc thực hiện hành động không mong muốn.

Phát hiện: IDPS có thể phát hiện các hoạt động lừa đảo hoặc tấn công xã hội thông qua giám sát nội dung email, liên kết, và yêu cầu truy cập.

Phòng chống: Cảnh báo người dùng về các liên kết hoặc email nguy hiểm, đồng thời ngăn chặn các truy cập từ các nguồn đáng ngờ.

* 1. Tấn công quét cổng (Port Scanning) là kỹ thuật mà kẻ tấn công sử dụng để khám phá các cổng mở hoặc các dịch vụ đang chạy trên máy chủ hoặc mạng mục tiêu. Quá trình này giúp kẻ tấn công xác định các điểm yếu trong hệ thống và từ đó có thể thực hiện các cuộc tấn công khác như xâm nhập trái phép hoặc khai thác lỗ hổng bảo mật.
  2. Tấn công bằng phần mềm độc hại (Malware Attack): tấn công bằng phần mềm độc hại (malware)

Phát hiện: IDPS có thể phát hiện các mẫu mã độc hại thông qua dấu vết của nó trong lưu lượng mạng hoặc trên các tệp tin truyền qua

* Các thuật toán học máy trong hệ thống phát hiện xâm nhập mạng
  + Học máy (Machine Learning - ML) là một tập hợp con của AI bao gồm tất cả các phương thức và thuật toán cho phép các máy để tìm hiểu tự động bằng các mô hình toán học để trích xuất thông tin hữu ích từ các bộ dữ liệu lớn.
  + ML phổ biến nhất (còn gọi là học tập nông) được sử dụng cho IDS là
    - cây quyết định,
    - K-nearest neighbors (KNN),
    - mạng thần kinh nhân tạo (ANN)
    - máy vectơ hỗ trợ (SVM)
    - cụm K-Mean
    - mạng học tập nhanh.
* Decision Tree

A diagram of a graph

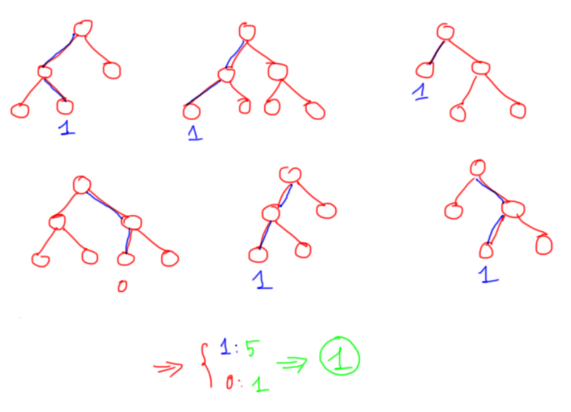
Description automatically generatedDT là một trong những thuật toán ML có giám sát cơ bản được sử dụng để phân loại và hồi quy bộ dữ liệu đã cho bằng cách áp dụng hàng loạt các quyết định (quy tắc). Mô hình có cấu trúc cây thông thường với các nút, cành và lá. Mỗi nút biểu thị một thuộc tính hoặc một tính năng. Một nhánh đại diện cho một quyết định hoặc quy tắc trong khi mỗi lá đại diện cho một kết quả hoặc nhãn lớp. Thuật toán DT sẽ tự động chọn các tính năng tốt nhất để xây dựng nên cây quyết định và sau đó thực hiện thao tác cắt tỉa để loại bỏ các nhánh không liên quan từ cây để tránh sự phù hợp. Các mô hình DT phổ biến nhất là Cart, C4.5 và ID3. Nhiều thuật toán học tập nâng cao như rừng ngẫu nhiên (Random Forest) và XGBoost (eXtreme Gradient Boosting) được xây dựng từ nhiều cây quyết định.

Có hai thông số cần chú ý:

* + - **Entropy (Độ hỗn loạn)**:
      * Entropy đo lường độ bất định hoặc hỗn loạn của một tập hợp dữ liệu. Một tập hợp dữ liệu càng "lẫn lộn" giữa các lớp, entropy càng cao.
      * A black background with white text

        Description automatically generatedCông thức tính entropy cho một tập dữ liệu S là:
    - Information Gain (Lợi ích thông tin):
      * Information gain đo lường sự giảm entropy khi sử dụng một thuộc tính để phân chia dữ liệu. Nó giúp ID3 chọn thuộc tính nào tốt nhất để phân chia dữ liệu tại mỗi bước.
      * A black background with white text

        Description automatically generatedCông thức tính information gain cho thuộc tính AA là:
* Random Forest

Thuật toán Random Forest sẽ bao gồm nhiều cây quyết định, mỗi cây được xây dựng dùng thuật toán Decision Tree trên tập dữ liệu khác nhau và dùng tập thuộc tính khác nhau. Sau đó kết quả dự đoán của thuật toán Random Forest sẽ được tổng hợp từ các cây quyết định.Khi dùng thuật toán Random Forest, ta hay để ý các thuộc tính như: số lượng cây quyết định sẽ xây dựng, số lượng thuộc tính dùng để xây dựng cây. Ngoài ra, vẫn có các thuộc tính của thuật toán Decision Tree để xây dựng cây như độ sâu tối đa, số phần tử tối thiểu trong 1 node để có thể tách.

Trong thuật toán Decision Tree, khi xây dựng cây quyết định nếu để độ sâu tùy ý thì cây sẽ phân loại đúng hết các dữ liệu trong tập training dẫn đến mô hình có thể dự đoán tệ trên tập validation/test, khi đó mô hình bị overfitting.

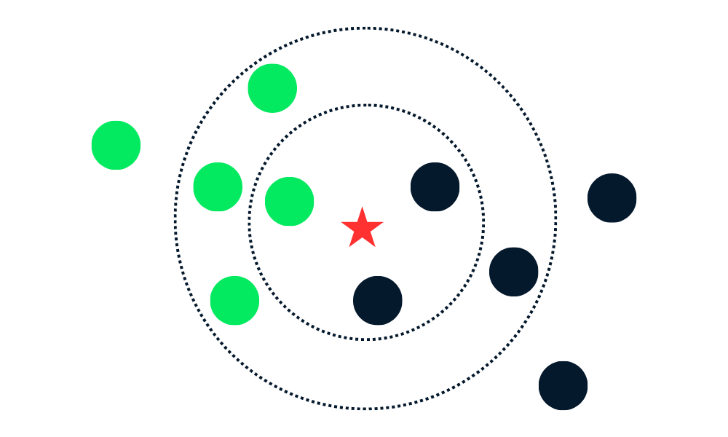
Thuật toán Random Forest gồm nhiều cây quyết định, mỗi cây quyết định đều có những yếu tố ngẫu nhiên:

Lấy ngẫu nhiên dữ liệu để xây dựng cây quyết định.

Lấy ngẫu nhiên các thuộc tính để xây dựng cây quyết định.

Do mỗi cây quyết định trong thuật toán Random Forest không dùng tất cả dữ liệu training, cũng như không dùng tất cả các thuộc tính của dữ liệu để xây dựng cây nên mỗi cây có thể sẽ dự đoán không tốt, khi đó mỗi mô hình cây quyết định không bị overfitting mà có thế bị underfitting, hay nói cách khác là mô hình có high bias. Tuy nhiên, kết quả cuối cùng của thuật toán Random Forest lại tổng hợp từ nhiều cây quyết định, thế nên thông tin từ các cây sẽ bổ sung thông tin cho nhau, dẫn đến mô hình có low bias và low variance, hay mô hình có kết quả dự đoán tốt.

* K*-Nearest Neighbor (KNN)*

KNN là một trong những thuật toán ML có giám sát đơn giản nhất sử dụng ý tưởng về "điểm dữ liệu tương tự tồn tại gần nhau trong một không gian" để dự đoán lớp của một mẫu dữ liệu nhất định. Nó xác định đầu ra một mẫu dựa trên thông tin của K điểm dữ liệu trong tập huấn luyện gần nó nhất (hay còn gọi là K-lân cận). Trong thuật toán KNN, tham số K ảnh hưởng đến hiệu suất của mô hình. Nếu giá trị của K nhỏ, mô hình có thể dễ bị overfitting khi đó mô hình không học được gì từ dữ liệu. Trong khi, khi lựa chọn giá trị K lớn có thể dẫn đến việc phân loại sai đối tượng mẫu. Karatas đã so sánh hiệu suất của các thuật toán ML khác nhau bằng cách sử dụng tập dữ liệu CSE-CIC-IDS2018. Họ giải quyết vấn đề về sự mất cân bằng dữ liệu bằng bằng cách sử dụng kỹ thuật tăng kích thước mẫu (Smote), dẫn đến việc cải thiện tỷ lệ phát hiện cho các mẫu thuộc lớp thiểu số.

A math equations and a green button

Description automatically generated with medium confidence

Công thức tính khoản cách giữa các điểm dữ liệu:

* Support vector machine

A diagram of a graph

Description automatically generatedSVM là một thuật toán ML có giám sát dựa trên ý tưởng về một siêu mặt phẳng (hay còn gọi là hyper lane) để phân tách các điểm của dữ liệu. Siêu mặt phẳng này sẽ chịu trách nhiệm chia không gian thành các miền khác nhau và mỗi miền này sẽ chứa một loại dữ liệu. SVM được sử dụng làm giải pháp của cả các vấn đề tuyến tính và phi tuyến tính. Đối với các vấn đề phi tuyến tính, các hàm kernel được sử dụng. Ý tưởng là đầu tiên ánh xạ một vectơ đầu vào có kích thước thấp thành không gian tính năng có kích thước cao bằng hàm kernel. Tiếp theo, một mặt phẳng cận biên tối đa và tối ưu thu được sẽ hoạt động như một ranh giới quyết định sử dụng các vectơ hỗ trợ.

* Artificial neural network

A diagram of a network

Description automatically generatedANN cũng là một thuật toán ML có giám sát và được truyền cảm hứng từ hệ thống thần kinh của bộ não con người. ANN được tạo thành từ các yếu tố xử lý gọi là tế bào thần kinh (nút) và các kết nối giữa chúng (cạnh). Các nút này được sắp xếp trong một lớp đầu vào, nhiều lớp ẩn và một lớp đầu ra. Thuật toán lan truyền ngược hay còn gọi là Backpropagation được sử dụng như một kỹ thuật học tập cho ANN. Ưu điểm chính của việc sử dụng kỹ thuật ANN là khả năng thực hiện mô hình hóa phi tuyến tính bằng cách học tập từ các bộ dữ liệu lớn. Tuy nhiên, vấn đề chính khi đào tạo mô hình với ANN là mức tiêu thụ thời gian cao do tính chất phức tạp của nó, làm chậm quá trình học tập và để đạt được một giải pháp dưới mức tối ưu.

* Một số mô hình IDS phổ biến
  + Snort
* Cơ chế hoạt động:
  + Chữ ký (Signature-based): Snort kiểm tra các gói dữ liệu mạng so với một bộ quy tắc (rules) để phát hiện các mẫu tấn công đã biết.
* Chế độ hoạt động: Có ba chế độ chính:
  + Sniffer: Nghe và hiển thị gói dữ liệu trên màn hình.
  + Packet Logger: Ghi lại toàn bộ các gói dữ liệu vào file log.
  + IDS/IPS: Kiểm tra gói dữ liệu theo bộ quy tắc để phát hiện tấn công.
* Triển khai Snort:
  + Cài đặt: Snort có thể được cài đặt trên nhiều hệ điều hành (Linux, Windows, Docker).
  + Cấu hình: Bạn cần cấu hình file snort.conf, thêm các quy tắc phát hiện tấn công trong file local.rules.
  + Chạy Snort: Ví dụ, bạn có thể sử dụng lệnh sau để chạy Snort trong chế độ IDS:

snort -A console -c /etc/snort/snort.conf -i eth0

* Suricata
  + Cơ chế hoạt dộng:
    - Dựa trên chữ ký và dị thường (Anomaly-based): Suricata phát hiện tấn công không chỉ dựa vào chữ ký mà còn sử dụng các kỹ thuật phát hiện bất thường, phân tích giao thức.
    - Hỗ trợ đa luồng: Điều này giúp xử lý khối lượng lớn dữ liệu mạng nhanh hơn so với Snort.
    - Phân tích giao thức: Suricata có thể phân tích chi tiết các giao thức như HTTP, TLS, DNS, cho phép phát hiện tấn công ở mức ứng dụng.
  + Triển khai Suricata:
    - Cài đặt: Tương tự như Snort, bạn có thể cài đặt Suricata trên nhiều hệ điều hành hoặc sử dụng Docker.
    - Cấu hình: File cấu hình suricata.yaml giúp bạn tùy chỉnh các giao thức, quy tắc và luồng xử lý.
    - Chạy Suricata:

suricata -c /etc/suricata/suricata.yaml -i eth0

* Zeek
  + Cơ chế hoạt động:
    - Event-based (Dựa trên sự kiện): Zeek không dựa trên chữ ký mà tập trung vào việc phát hiện hành vi mạng thông qua hệ thống event-driven. Nó phân tích các giao thức mạng và tạo ra các sự kiện để ghi lại những hành vi bất thường hoặc tấn công tiềm ẩn.
    - Kịch bản hóa (Scripting): Zeek sử dụng ngôn ngữ scripting mạnh mẽ để người dùng có thể tùy chỉnh việc phát hiện và ghi lại các sự kiện.
  + Triển khai Zeek:
    - Cài đặt: Cài đặt Zeek trên các hệ điều hành Linux hoặc sử dụng Docker.
    - Cấu hình: Tùy chỉnh các script để định nghĩa những sự kiện cần giám sát trong mạng. Ví dụ, cấu hình local.zeek giúp bạn tùy chỉnh log và giao thức.
    - Chạy Zeek:

zeek -i eth0 local.zeek