

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP. HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN CHUYÊN NGÀNH
ĐỀ TÀI ATTT015: NGHIÊN CỨU ỨNG DỤNG THUẬT TOÁN C4.5
DECISION TREE TRONG PHÁT HIỆN TẤN CÔNG MẠNG

Giảng viên hướng dẫn: Nguyễn Thị Hồng Thảo

Sinh viên thực hiện: Phạm Văn Minh

Mã số sinh viên: 2033216488

Lớp: 12DHBM02

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP. HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN CHUYÊN NGÀNH
ĐỀ TÀI ATTT015: NGHIÊN CỨU ỨNG DỤNG THUẬT TOÁN C4.5
DECISION TREE TRONG PHÁT HIỆN TẤN CÔNG MẠNG

Giảng viên hướng dẫn: Nguyễn Thị Hồng Thảo

Sinh viên thực hiện: Phạm Văn Minh

Mã số sinh viên: 2033216488

Lớp: 12DHBM02

TP.HCM, ngày 19 tháng 12 năm 2023

LỜI MỞ ĐẦU

Lời đầu tiên, em xin chân thành cảm ơn Quý giảng viên khoa Công nghệ thông tin - Trường Đại học Công thương Thành phố Hồ Chí Minh, và giảng viên hướng dẫn Nguyễn Thị Hồng Thảo đã giúp đỡ em hoàn thành bài báo cáo đồ án chuyên ngành về đề tài “NGHIÊN CỨU ỨNG DỤNG THUẬT TOÁN C4.5 DECISION TREE TRONG PHÁT HIỆN TẤN CÔNG MẠNG”.

Hiện nay, nhu cầu sử dụng internet trên toàn cầu đang có xu hướng ngày càng phát triển và ngày càng có nhiều công nghệ mới được phát minh. Đồng thời cũng kéo theo nhiều vấn đề về các yếu tố bảo mật, rủi ro an ninh thông tin trên môi trường không gian mạng. Tấn công mạng có thể kể đến nhiều hình thức, nhiều phương pháp tấn công. Và trong đề tài này, em đã tập trung nghiên cứu ứng dụng thuật toán cây quyết định C4.5 để phát hiện tấn công từ chối dịch vụ (DDoS).

Đồ án này là thành quả của quá trình học tập, nghiên cứu và tích lũy kinh nghiệm của cá nhân em. Chính vì thế em rất trân trọng thành quả mà trong suốt quá trình thực hiện bản thân mình đã đạt được.

Cuối cùng, kính mong Quý giảng viên cùng khoa Công nghệ thông tin trường Đại học Công thương Thành phố Hồ Chí Minh đánh giá khách quan và đưa ra nhận xét để em ngày càng phát huy tốt năng lực của bản thân mình.

Em xin chân thành cảm ơn!

TP.HCM, ngày 19 tháng 12 năm 2023

Sinh viên thực hiện

Phạm Văn Minh

MỤC LỤC

LỜI MỞ ĐẦU	3
PHÂN CÔNG NHIỆM VỤ VÀ MỨC ĐỘ HOÀN THÀNH.....	6
ĐÁNH GIÁ VÀ NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN	7
ĐÁNH GIÁ VÀ NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN	8
PHẦN I: NỘI DUNG VÀ MỨC ĐỘ HOÀN THIỆN	9
<i>1. Nghiên cứu về tấn công từ chối dịch vụ (DDoS)</i>	<i>9</i>
<i>1.1. Định nghĩa về một cuộc tấn công từ chối dịch vụ</i>	<i>9</i>
<i>1.2. Số liệu về quy mô và tần suất tấn công DDoS.....</i>	<i>9</i>
<i>2. Mức độ hoàn thiện trong quá trình nghiên cứu</i>	<i>11</i>
<i>2.1. Phân tích cách thức tấn công</i>	<i>11</i>
<i>2.2. Mức độ hoàn thiện trong quá trình kiểm thử tấn công</i>	<i>11</i>
PHẦN II: THUẬT TOÁN C4.5 DECISION TREE	12
<i>2.1. Định nghĩa về cây quyết định ^[2]</i>	<i>12</i>
<i>2.2. Nghiên cứu lý thuyết về C4.5 Decision Tree</i>	<i>13</i>
PHẦN III: NGHIÊN CỨU LÝ THUYẾT VÀ THỰC HÀNH VỀ TẤN CÔNG MỘT HỆ THỐNG MẠNG	15
<i>3.1. Nghiên cứu lý thuyết về tấn công một hệ thống mạng</i>	<i>15</i>
<i>3.2. Nghiên cứu thực hành tấn công hệ thống mạng</i>	<i>16</i>
<i>3.2.1. Tấn công TCP Flood (TCP Floot Attack).....</i>	<i>16</i>
<i>3.2.2. Tấn công ICMP Flood (ICMP Flood Attack)</i>	<i>18</i>
PHẦN IV: ỨNG DỤNG CỦA THUẬT TOÁN C4.5 DECISION TREE.....	20
<i>4.1. Thuật toán phát hiện tấn công DDoS sử dụng C4.5 Decision Tree</i>	<i>20</i>
<i>4.2. Thuật toán C4.5 Decision Tree để phân loại các cuộc tấn công theo giao thức </i>	<i>23</i>
<i>4.2.1. Xây dựng và phân loại tấn công</i>	<i>23</i>
<i>4.2.2. Phân loại tấn công dựa trên giao thức TCP</i>	<i>25</i>

4.2.3. Phân loại tấn công dựa trên giao thức ICMP.....	26
4.3. Xây dựng C4.5 Decision Tree dựa trên các phân loại:	26
4.3.1. Xây dựng và phân loại tấn công	26
4.3.2. Phân loại tấn công dựa trên giao thức TCP	27
4.3.3. Phân loại tấn công dựa trên giao thức ICMP.....	28
4.3.4. Xây dựng C4.5 Decision Tree dựa trên các phân loại	29
PHẦN V: CHƯƠNG TRÌNH ỨNG DỤNG PHÁT HIỆN TẤN CÔNG TỪ CHỐI DỊCH VỤ PHÂN TÁN	30
5.1. Phân tích thuật toán	32
5.2. Kiểm thử chương trình.....	35
5.3. Đánh giá.....	37
PHẦN VI: KẾT LUẬN	37
6.1. Về nội dung đồ án.....	37
6.2. Về quá trình thực hiện đồ án	38
TÀI LIỆU THAM KHẢO.....	39

PHÂN CÔNG NHIỆM VỤ VÀ MỨC ĐỘ HOÀN THÀNH

1. Phạm Văn Minh – 2033216488

Nhiệm vụ	Mức độ hoàn thành
1. Nghiên cứu nội dung về tấn công hệ thống mạng	100%
2. Nghiên cứu và phân tích số liệu thống kê	100%
3. Tìm hiểu và vận dụng thuật toán C4.5 vào an toàn thông tin	100%
4. Xây dựng thuật toán, phương pháp giải thuật	100%
5. Nghiên cứu lý thuyết và thực hành tấn công theo giao thức (TCP và ICMP)	100%
5. Xây dựng code C++ về phát hiện tấn công mạng	100%
6. Kiểm thử và hoàn thiện đồ án	100%
Tổng:	100%

ĐÁNH GIÁ VÀ NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Giảng viên hướng dẫn
(Kí và ghi rõ họ tên)

ĐÁNH GIÁ VÀ NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Giảng viên phản biện

(Kí và ghi rõ họ tên)

PHẦN I: NỘI DUNG VÀ MỨC ĐỘ HOÀN THIỆN

1. Nghiên cứu về tấn công từ chối dịch vụ (DDoS)

1.1. Định nghĩa về một cuộc tấn công từ chối dịch vụ

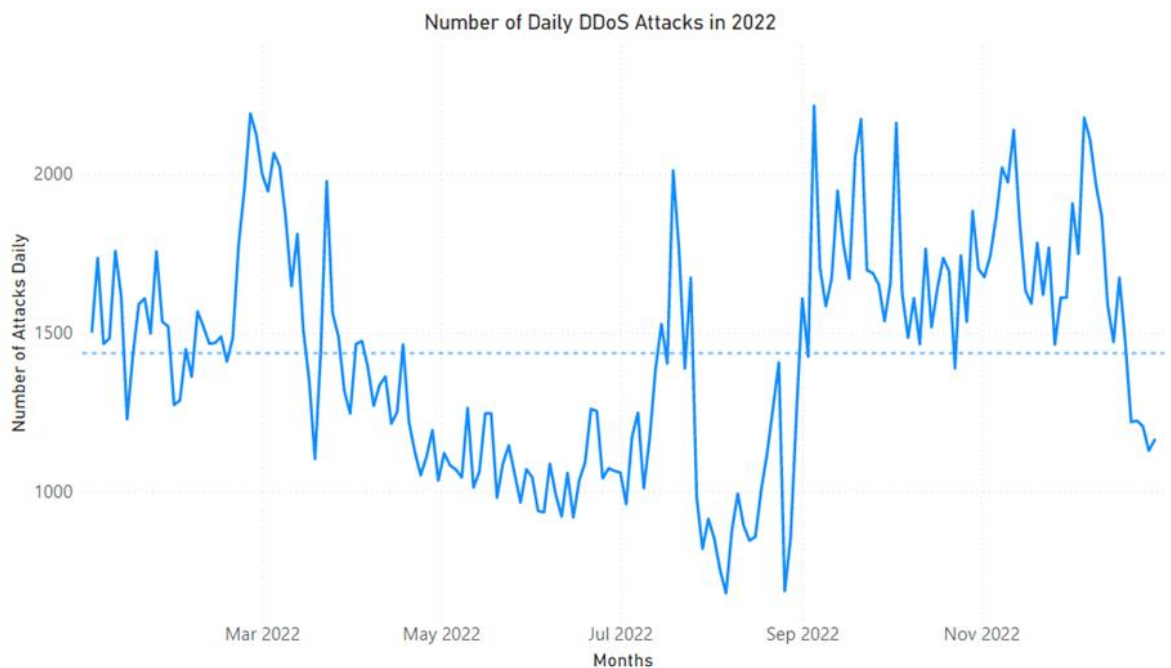
“Cuộc tấn công DDoS nhằm mục tiêu đến các website và máy chủ bằng cách làm gián đoạn dịch vụ mạng nhằm tìm cách làm cạn kiệt tài nguyên của ứng dụng. Thủ phạm đứng đằng sau các cuộc tấn công này sẽ gây tràn site bằng lưu lượng truy nhập lỗi, làm website hoạt động kém đi hoặc khiến website bị ngoại tuyến hoàn toàn. Những loại hình tấn công như này đang gia tăng.

Cuộc tấn công DDoS có phạm vi mục tiêu rộng, nhắm mục tiêu tới mọi loại ngành và quy mô công ty trên toàn cầu. Một số ngành như trò chơi, thương mại điện tử và viễn thông bị nhắm mục tiêu nhiều hơn các ngành khác. Cuộc tấn công DDoS là một trong số các mối đe dọa trên mạng phổ biến nhất và chúng có thể xâm phạm tới doanh nghiệp, bảo mật trực tuyến, doanh thu và danh tiếng của bạn.” (Theo Microsoft Security).

1.2. Số liệu về quy mô và tần suất tấn công DDoS

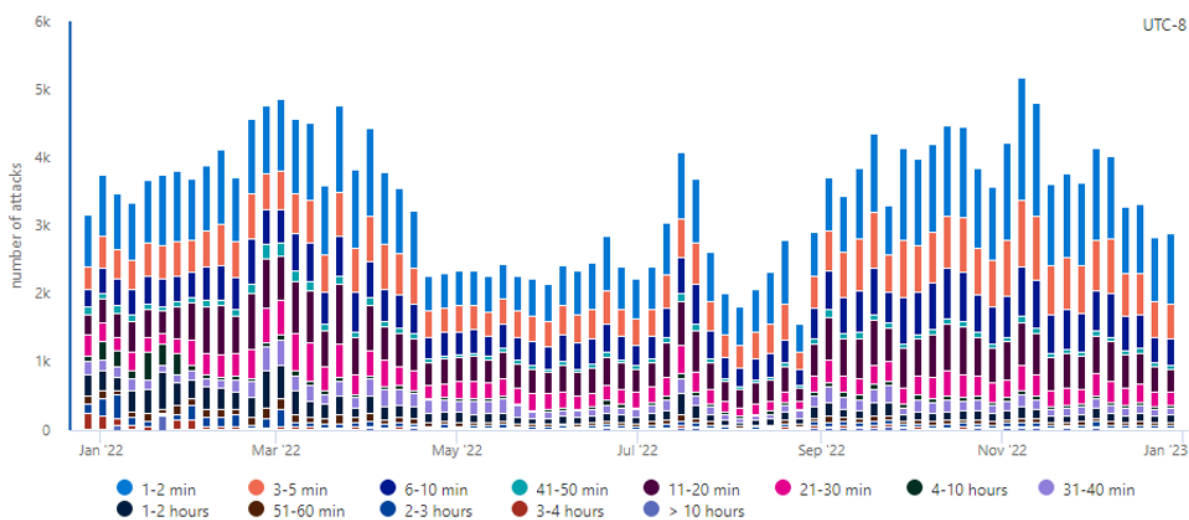
Trong quy mô thực tế, tấn công DDoS diễn ra với nhiều các máy tấn công đến một máy chủ (server) gây cạn kiệt tài nguyên hệ thống. Dẫn đến những rủi ro, thất thoát cho công ty, doanh nghiệp, nguy hiểm hơn là ảnh hưởng rất lớn nếu tin tặc thực hiện tấn công vào một công ty, doanh nghiệp đa quốc gia. Trang thông tin của Chính phủ.

Việc phát hiện và ngăn ngừa tấn công DDoS là tương đối khó, cần yêu cầu phải phân tích tình hình lưu lượng mạng thực tế và phải đảm bảo các truy cập an toàn vẫn luôn được thông suốt. Bằng việc phân tích dữ liệu về lưu lượng truy cập giúp ta phát hiện và ngăn chặn kịp thời, từ đó giảm thiểu tối đa nguy cơ rủi ro về tấn công DDoS.



Hình 1.2a: Số liệu hằng ngày về tấn công DDoS năm 2022^[1]

Theo dữ liệu thống kê về thời gian tấn công của một cuộc tấn công DDoS của Microsorf Security vào năm 2022, có khoảng 89% các cuộc tấn công có thời lượng dưới một phút. Và khoảng 26% các cuộc tấn công có thời lượng trung bình từ một đến hai phút.



Hình 1.2b: Thời lượng trung bình của một cuộc tấn công DDoS được ghi nhận

2. Mức độ hoàn thiện trong quá trình nghiên cứu

2.1. Phân tích cách thức tấn công

Trong quá trình phân tích dữ liệu và hiểu rõ quá trình tấn công từ chối dịch vụ, ta có thể ghi nhận rằng việc đánh giá cuộc tấn công từ chối dịch vụ phụ thuộc vào các yếu tố như:

- + Giao thức tấn công (TCP, UDP, ICMP)
- + Địa chỉ nguồn (tức địa chỉ của máy tấn công)
- + Địa chỉ đích (Tức địa chỉ của máy nạn nhân)
- + Cổng nguồn (Source Port)
- + Cổng đích (Destination Port)

2.2. Mức độ hoàn thiện trong quá trình kiểm thử tấn công

Trong quá trình thực hiện tấn công để phân tích rủi ro, hệ thống sẽ gặp một lượng lớn các gói tin yêu cầu và phản hồi lại. Gây ách tắc truyền thông dữ liệu, cá biệt có thể gây sập server nếu trong quá trình thực hiện tấn công mà server quá tải.

Ghi nhận mức độ rủi ro trong quá trình kiểm thử được ghi nhận lại như sau:

- + Quá trình tấn công TCP, server phải phản hồi liên tục các gói tin RST và ACK với một lưu lượng vô cùng lớn. Kết quả là server shutdown và phải restart lại để hoạt động bình thường.

493...	90.369484	192.168.223.128	192.168.223.130	TCP	60	35156 → 0	[<None>]	Seq=2680732398	Win=512	Len=0		
493...	90.369484	192.168.223.128	192.168.223.130	TCP	60	35155 → 0	[<None>]	Seq=197738959	Win=512	Len=0		
493...	90.369484	192.168.223.128	192.168.223.130	TCP	60	35154 → 0	[<None>]	Seq=4137745652	Win=512	Len=0		
493...	90.369484	192.168.223.128	192.168.223.130	TCP	60	35153 → 0	[<None>]	Seq=4169651572	Win=512	Len=0		
493...	90.369484	192.168.223.128	192.168.223.130	TCP	60	[TCP Previous segment not captured]	35152 → 0	[<None>]	Seq=1160470356	Win=512		
493...	90.369387	192.168.223.130	192.168.223.128	TCP	54	0 → 35143	[RST, ACK]	Seq=1	Ack=3995005595	Win=0	Len=0	
493...	90.369358	192.168.223.130	192.168.223.128	TCP	54	[TCP ACKed unseen segment]	0 → 35142	[RST, ACK]	Seq=1	Ack=1461480263	Win=0	Len=0
493...	90.369339	192.168.223.130	192.168.223.128	TCP	54	0 → 35141	[RST, ACK]	Seq=1	Ack=4291288590	Win=0	Len=0	
493...	90.369310	192.168.223.130	192.168.223.128	TCP	54	0 → 35140	[RST, ACK]	Seq=1	Ack=4208469274	Win=0	Len=0	
493...	90.369289	192.168.223.130	192.168.223.128	TCP	54	0 → 35139	[RST, ACK]	Seq=1	Ack=2217022587	Win=0	Len=0	
493...	90.369257	192.168.223.130	192.168.223.128	TCP	54	0 → 35138	[RST, ACK]	Seq=1	Ack=3450776300	Win=0	Len=0	
493...	90.369229	192.168.223.130	192.168.223.128	TCP	54	[TCP ACKed unseen segment]	0 → 35137	[RST, ACK]	Seq=1	Ack=115517486	Win=0	Len=0
493...	90.369196	192.168.223.130	192.168.223.128	TCP	54	0 → 35136	[RST, ACK]	Seq=1	Ack=4282337114	Win=0	Len=0	
493...	90.369177	192.168.223.130	192.168.223.128	TCP	54	[TCP ACKed unseen segment]	0 → 35135	[RST, ACK]	Seq=1	Ack=470253385	Win=0	Len=0
493...	90.369147	192.168.223.130	192.168.223.128	TCP	54	0 → 35134	[RST, ACK]	Seq=1	Ack=4147173445	Win=0	Len=0	
493...	90.369129	192.168.223.130	192.168.223.128	TCP	54	0 → 35133	[RST, ACK]	Seq=1	Ack=3194913488	Win=0	Len=0	
493...	90.369098	192.168.223.130	192.168.223.128	TCP	54	0 → 35132	[RST, ACK]	Seq=1	Ack=3614633365	Win=0	Len=0	
493...	90.369080	192.168.223.128	192.168.223.130	TCP	60	35151 → 0	[<None>]	Seq=1323058572	Win=512	Len=0		
493...	90.369080	192.168.223.128	192.168.223.130	TCP	60	35150 → 0	[<None>]	Seq=521337600	Win=512	Len=0		
493...	90.369080	192.168.223.128	192.168.223.130	TCP	60	[TCP Previous segment not captured]	35149 → 0	[<None>]	Seq=1021497175	Win=512		

Hình 2.2a: Ghi nhận quá trình tấn công TCP Flood

+ Quá trình tấn công ICMP, server phải phản hồi liên tục các echo request và reply. Kết quả là với lưu lượng phản hồi lớn và liên tục, server quá tải dẫn đến kết nối không còn được ổn định.

No.	Time	Source	Destination	Protocol	Length	Info
145...	282.719229	192.168.223.130	192.168.223.128	ICMP	98	Echo (ping) reply id=0x0001, seq=55787/60377, ttl=128 (request in 1456
145...	282.719464	192.168.223.128	192.168.223.130	ICMP	98	Echo (ping) request id=0x0001, seq=55788/60633, ttl=64 (reply in 1456899
145...	282.719485	192.168.223.130	192.168.223.128	ICMP	98	Echo (ping) reply id=0x0001, seq=55788/60633, ttl=128 (request in 1456
145...	282.719717	192.168.223.128	192.168.223.130	ICMP	98	Echo (ping) request id=0x0001, seq=55789/60889, ttl=64 (reply in 1456901
145...	282.719738	192.168.223.130	192.168.223.128	ICMP	98	Echo (ping) reply id=0x0001, seq=55789/60889, ttl=128 (request in 1456
145...	282.719940	192.168.223.128	192.168.223.130	ICMP	98	Echo (ping) request id=0x0001, seq=55790/61145, ttl=64 (reply in 1456903
145...	282.720057	192.168.223.130	192.168.223.128	ICMP	98	Echo (ping) reply id=0x0001, seq=55790/61145, ttl=128 (request in 1456
145...	282.720321	192.168.223.128	192.168.223.130	ICMP	98	Echo (ping) request id=0x0001, seq=55791/61401, ttl=64 (reply in 1456905
145...	282.720382	192.168.223.130	192.168.223.128	ICMP	98	Echo (ping) reply id=0x0001, seq=55791/61401, ttl=128 (request in 1456
145...	282.720676	192.168.223.128	192.168.223.130	ICMP	98	Echo (ping) request id=0x0001, seq=55792/61657, ttl=64 (reply in 1456907
145...	282.720708	192.168.223.130	192.168.223.128	ICMP	98	Echo (ping) reply id=0x0001, seq=55792/61657, ttl=128 (request in 1456
145...	282.720987	192.168.223.128	192.168.223.130	ICMP	98	Echo (ping) request id=0x0001, seq=55793/61913, ttl=64 (reply in 1456909
145...	282.721014	192.168.223.130	192.168.223.128	ICMP	98	Echo (ping) reply id=0x0001, seq=55793/61913, ttl=128 (request in 1456
145...	282.721333	192.168.223.128	192.168.223.130	ICMP	98	Echo (ping) request id=0x0001, seq=55794/62169, ttl=64 (reply in 1456911
145...	282.721371	192.168.223.130	192.168.223.128	ICMP	98	Echo (ping) reply id=0x0001, seq=55794/62169, ttl=128 (request in 1456
145...	282.723174	192.168.223.128	192.168.223.130	ICMP	98	Echo (ping) request id=0x0001, seq=55795/62425, ttl=64 (reply in 1456913
145...	282.723206	192.168.223.130	192.168.223.128	ICMP	98	Echo (ping) reply id=0x0001, seq=55795/62425, ttl=128 (request in 1456
145...	282.727731	192.168.223.128	192.168.223.130	ICMP	98	Echo (ping) request id=0x0001, seq=55796/62681, ttl=64 (reply in 1456915
145...	282.727828	192.168.223.130	192.168.223.128	ICMP	98	Echo (ping) reply id=0x0001, seq=55796/62681, ttl=128 (request in 1456
145...	282.730848	192.168.223.128	192.168.223.130	ICMP	98	Echo (ping) request id=0x0001, seq=55797/62937, ttl=64 (reply in 1456917
145...	282.730913	192.168.223.130	192.168.223.128	ICMP	98	Echo (ping) reply id=0x0001, seq=55797/62937, ttl=128 (request in 1456
145...	282.735415	192.168.223.128	192.168.223.130	ICMP	98	Echo (ping) request id=0x0001, seq=55798/63193, ttl=64 (reply in 1456919
145...	282.735484	192.168.223.130	192.168.223.128	ICMP	98	Echo (ping) reply id=0x0001, seq=55798/63193, ttl=128 (request in 1456
145...	282.740077	192.168.223.128	192.168.223.130	ICMP	98	Echo (ping) request id=0x0001, seq=55799/63449, ttl=64 (reply in 1456921

Hình 2.2b: Ghi nhận quá trình tấn công ICMP Flood

PHẦN II: THUẬT TOÁN C4.5 DECISION TREE

2.1. Định nghĩa về cây quyết định ^[2]

Trong lĩnh vực khai phá dữ liệu, cây quyết định (Decision Tree – DT) là một mô hình dự đoán (predictive model) thuộc lớp các bài toán phân lớp (classification problem) dùng để xác định lớp của các đối tượng cần dự đoán.

Cây quyết định dựa vào dãy các luật để dự đoán lớp của đối tượng. Mỗi một nút trong (internal node) của DT tương ứng với một biến, đường nối giữa nó với nút con của nó thể hiện một giá trị cụ thể cho biến đó. Mỗi nút lá (leaf) đại diện cho giá trị dự đoán của biến phân loại.

Cây quyết định học để dự đoán giá trị của các biến phân loại bằng cách dựa vào tập dữ liệu huấn luyện (training data) để chọn ra nút gốc (root node) để phân tách cây bằng cách tính độ lợi thông tin (Information Gain - IG), quá trình này được lặp lại một cách đệ qui cho đến khi không thể tiếp tục thực hiện.

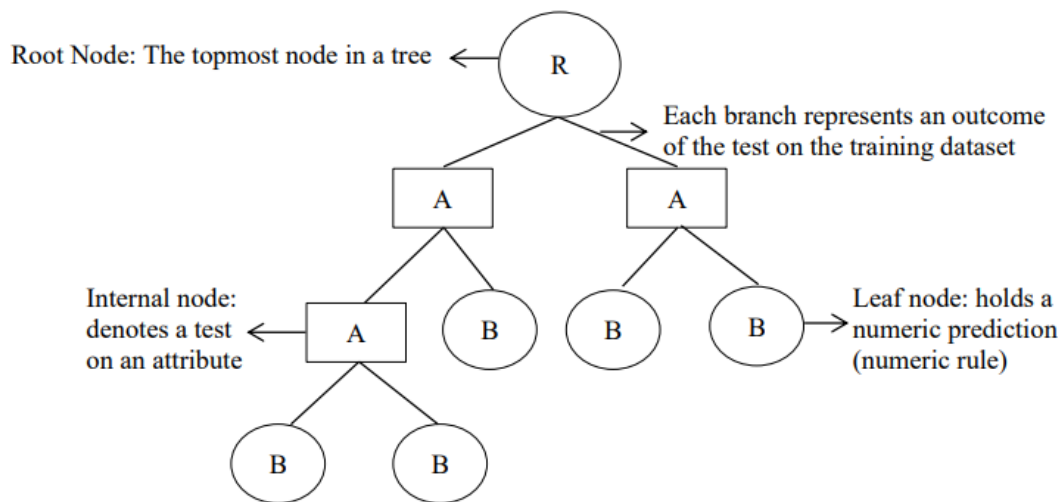
2.2. Nghiên cứu lý thuyết về C4.5 Decision Tree

Với sự phát triển của khoa học công nghệ hiện nay, lượng dữ liệu người dùng ngày càng lớn và kèm theo đó là yêu cầu bảo mật dữ liệu, phòng chống tin tặc tấn công đánh cắp dữ liệu của chúng ta thông qua việc sử dụng mạng thường ngày.

C4.5 Decision Tree là một trong những thuật toán hiệu quả được phát triển mở rộng dựa trên thuật toán ID3 (Iterative Dichotomiser 3), tiết kiệm chi phí hơn so với một số các giải pháp khác, thông qua các thuật toán như sử dụng Entropy để xây dựng mô hình, thuật toán tiết kiệm chi phí, thuật toán ngưỡng thích ứng,...

Thuật toán C4.5 Decision Tree có những ưu điểm như:

- + Là thuật toán phân loại và dự đoán dữ liệu bằng cây quyết định.
- + Kỹ thuật có ưu điểm: Xử lý các dữ liệu rời rạc và dữ liệu liên tục, đề ra các quy tắc dễ dàng để giải thích và khả năng sử dụng bộ nhớ máy tính để hoàn tất nhanh nhất nhằm tiết kiệm chi phí.
- + Mô hình dễ hiểu và dễ giải thích: Được ứng dụng dựa trên môn học cơ sở ngành “Cấu trúc dữ liệu và giải thuật”, những lập trình viên đều có thể dễ dàng học hỏi và phát triển cây theo cá nhân.
- + Độ chính xác cao: Việc xây dựng các node được dựa trên độ biến thiên về Entropy và độ khuếch đại GainRatio. Giúp định vị các node lá theo đúng vị trí và không bị trùng lặp dữ liệu.
- + Áp dụng trong nhiều lĩnh vực: Phân tích dữ liệu, phân tích thiết kế hệ thống, an ninh thông tin.



Hình 1.1: Sơ đồ giải thuật C4.5 Decision Tree

Công thức tính *Entropy* được định nghĩa như sau:

$$Entropy(S) = \sum_{i=1}^n (-p_i) * \log_2(p_i)$$

Trong đó:

+ E : Entropy

+ S : Bộ trường hợp

+ n : Số lượng phân vùng S

+ p_i : $\frac{\text{Số lượng mẫu cho loại } (i)}{\text{Tỉ lệ } S_i \text{ so với } S}$

Độ khuếch đại (Gain) là thước đo tính hiệu quả của một thuộc tính trong việc phân loại dữ liệu, sau giá trị entropy thu thập được cho một tập dữ liệu. Có công thức như sau:

$$Gain_{(S,A)} = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} * Entropy(S_i)$$

Trong đó:

- + G : Độ khuếch đại
- + S : Bộ trường hợp
- + A : Thuộc tính
- + n : Số thuộc tính thuộc phân vùng A
- + $|S_i|$: Số lượng phân vùng mẫu S của i
- + $|S|$: Số lượng mẫu trong S

Các yếu tố để đánh giá một thuật toán hiệu quả và tiết kiệm chi phí có thể kể đến như:

- + Decision Tree có thể tạo ra các quy tắc dễ hiểu.
- + Thực hiện phân loại mà không cần áp dụng quá nhiều tính toán.
- + Có thể xử lý dữ liệu liên tục và phân loại.
- + Phát hiện được nghi vấn truy cập trong thời gian nhanh nhất

PHẦN III: NGHIÊN CỨU LÝ THUYẾT VÀ THỰC HÀNH VỀ TẤN CÔNG MỘT HỆ THỐNG MẠNG

3.1. Nghiên cứu lý thuyết về tấn công một hệ thống mạng

Trong quá trình nghiên cứu đề tài, sinh viên nhận thấy rằng tấn công vào một hệ thống mạng bao gồm rất nhiều các yếu tố, các phương pháp tấn công. Có thể kể đến một số các phương pháp tấn công phổ biến thường gặp hiện nay như:

- + Tấn công từ chối dịch vụ (DoS Attack)
- + Tấn công từ chối dịch vụ phân tán (DDoS Attack)
- + Tấn công giả mạo (Phishing Attack)
- + Tấn công bị động (Passive Attack)
- + Tấn công Man in the middle (Man-in-the-Middle Attack)
- + Tấn công mật khẩu (Password attack)

- + Tấn công nội bộ (Insider attack)
- + Tấn công phá mã khóa (Compromised-key attack)
- + Tấn công cơ sở dữ liệu (SQL Injection)

3.2. Nghiên cứu thực hành tấn công hệ thống mạng

Dựa vào cơ sở lý thuyết về tấn công một hệ thống mạng, sinh viên tập trung vào nghiên cứu thực hành tấn công bằng phương pháp tấn công từ chối dịch vụ (DoS) và tấn công từ chối dịch vụ phân tán (DDoS). Tiến hành tấn công kiểm thử trên hạ tầng ảo hóa, danh sách thiết bị cần chuẩn bị gồm:

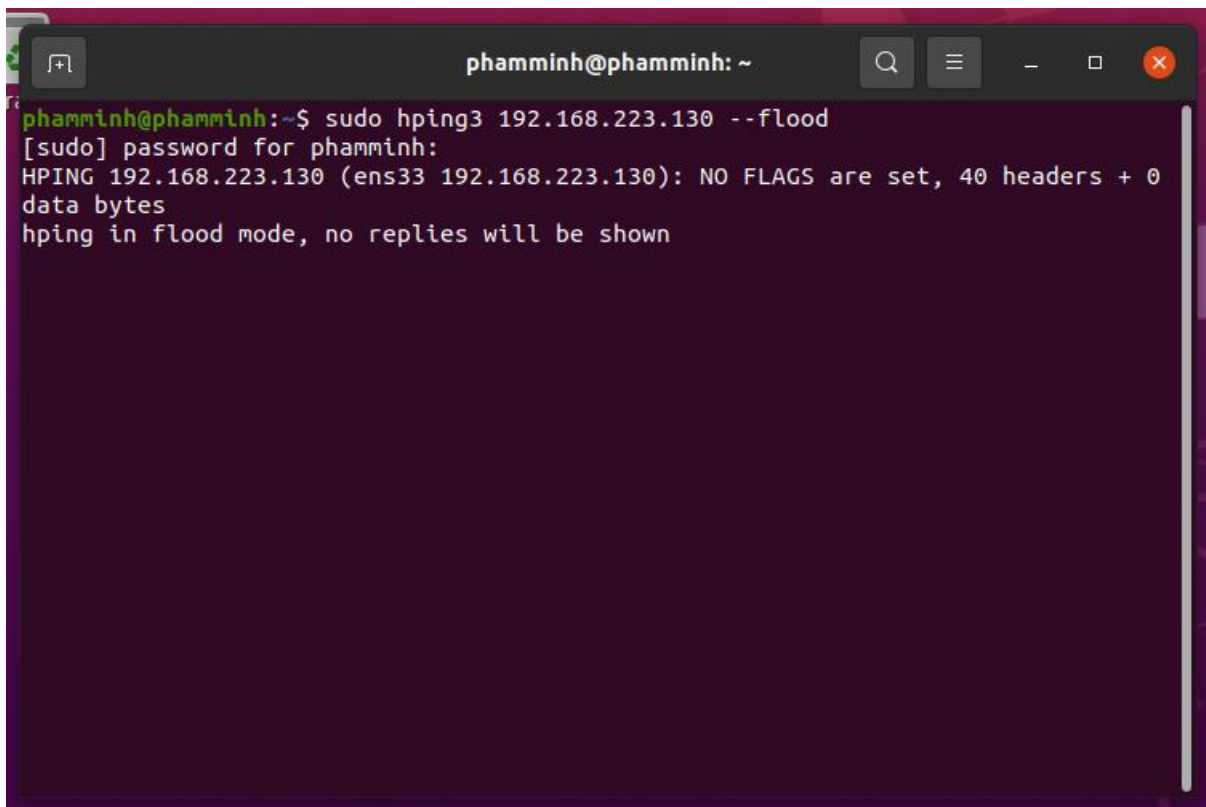
Thiết bị (Device)	Địa chỉ IP (IP Address)	Tường lửa (Firewall)
Windows Server 2019	192.168.223.130	No
Ubuntu 22.04	192.168.223.128	No

Bảng 3.2: Danh sách thiết bị thực hành tấn công

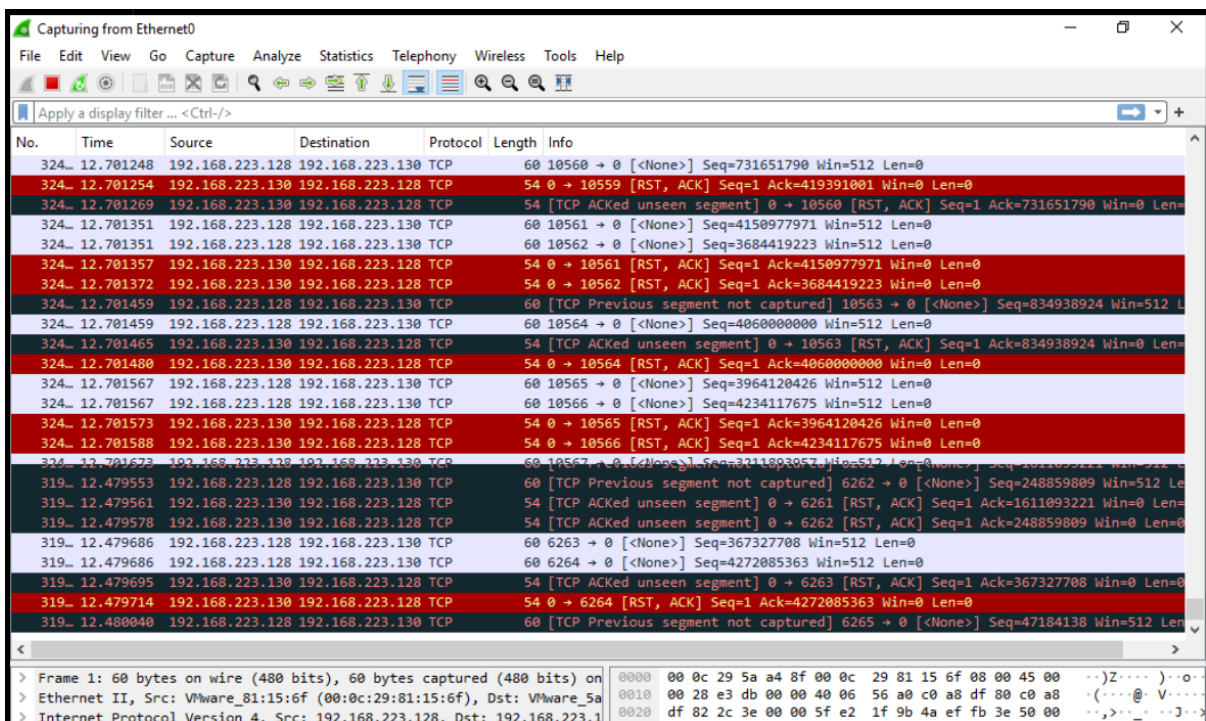
3.2.1. Tấn công TCP Flood (TCP Floot Attack)

Kỹ thuật thực hiện: Sử dụng Ubuntu/Linux để tấn công tràn SYN trên Windows Server 2019 bằng **Hping3**.

- + Trên máy thực hiện tấn công sử dụng lệnh “hping3 <IP victim> --flood”



Hình 3.2.1a: Thực hiện tấn công TCP Flood



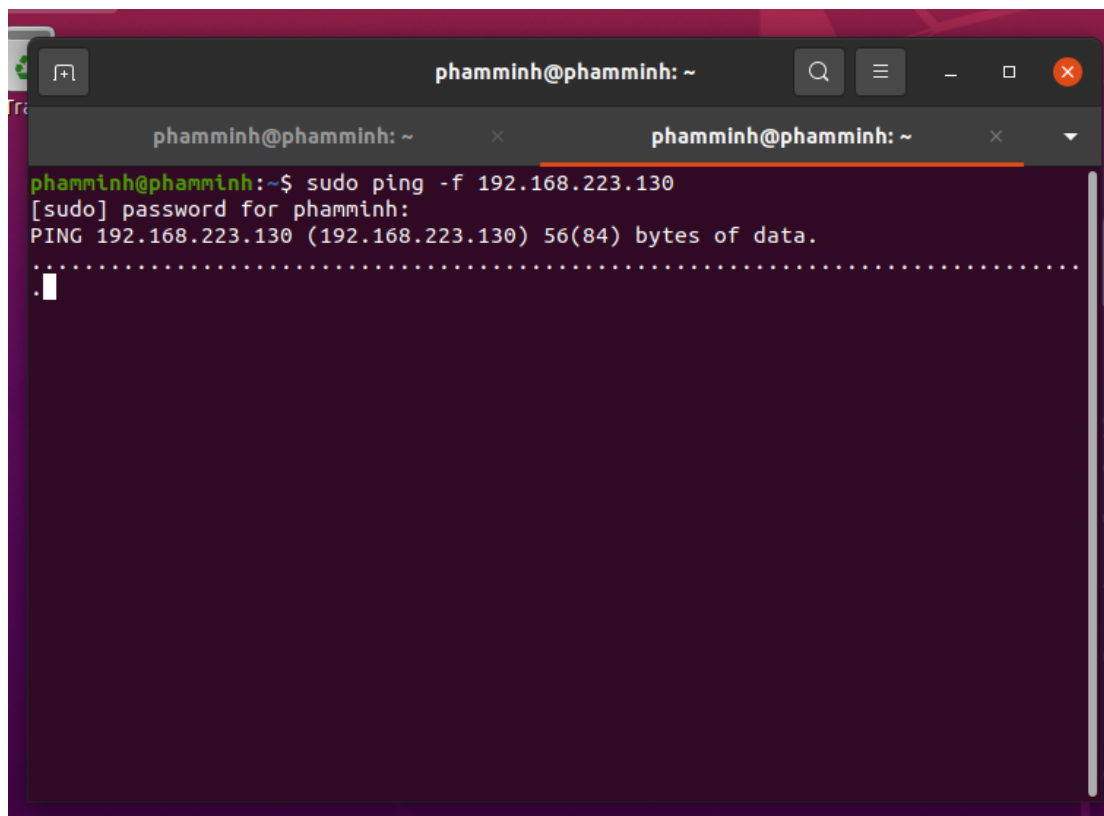
Hình 3.2.1b: Kết quả trả về từ công cụ phân tích trên Windows Server 2019

3.2.2. Tấn công ICMP Flood (ICMP Flood Attack)

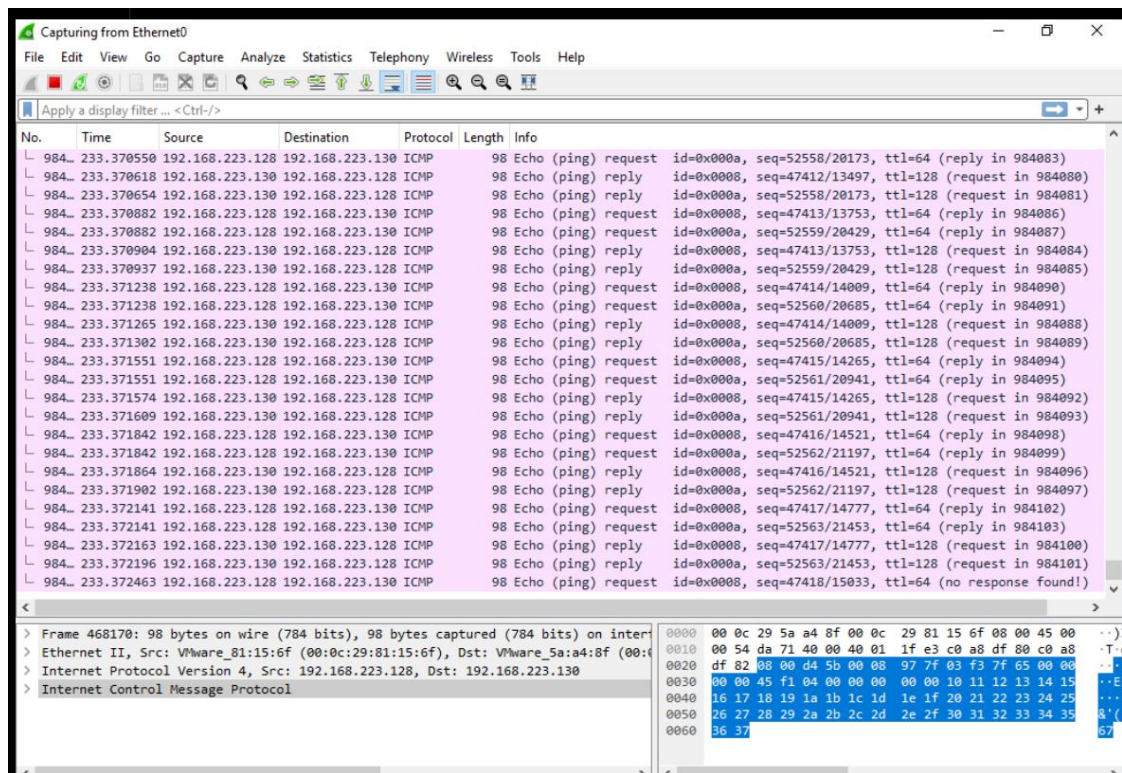
Kỹ thuật thực hiện: Sử dụng Ubuntu/Linux để tấn công tràn SYN trên Windows Server 2019 bằng **ping**.

+ Trên máy tấn công, sử dụng lệnh “ping -f <IP victim>”

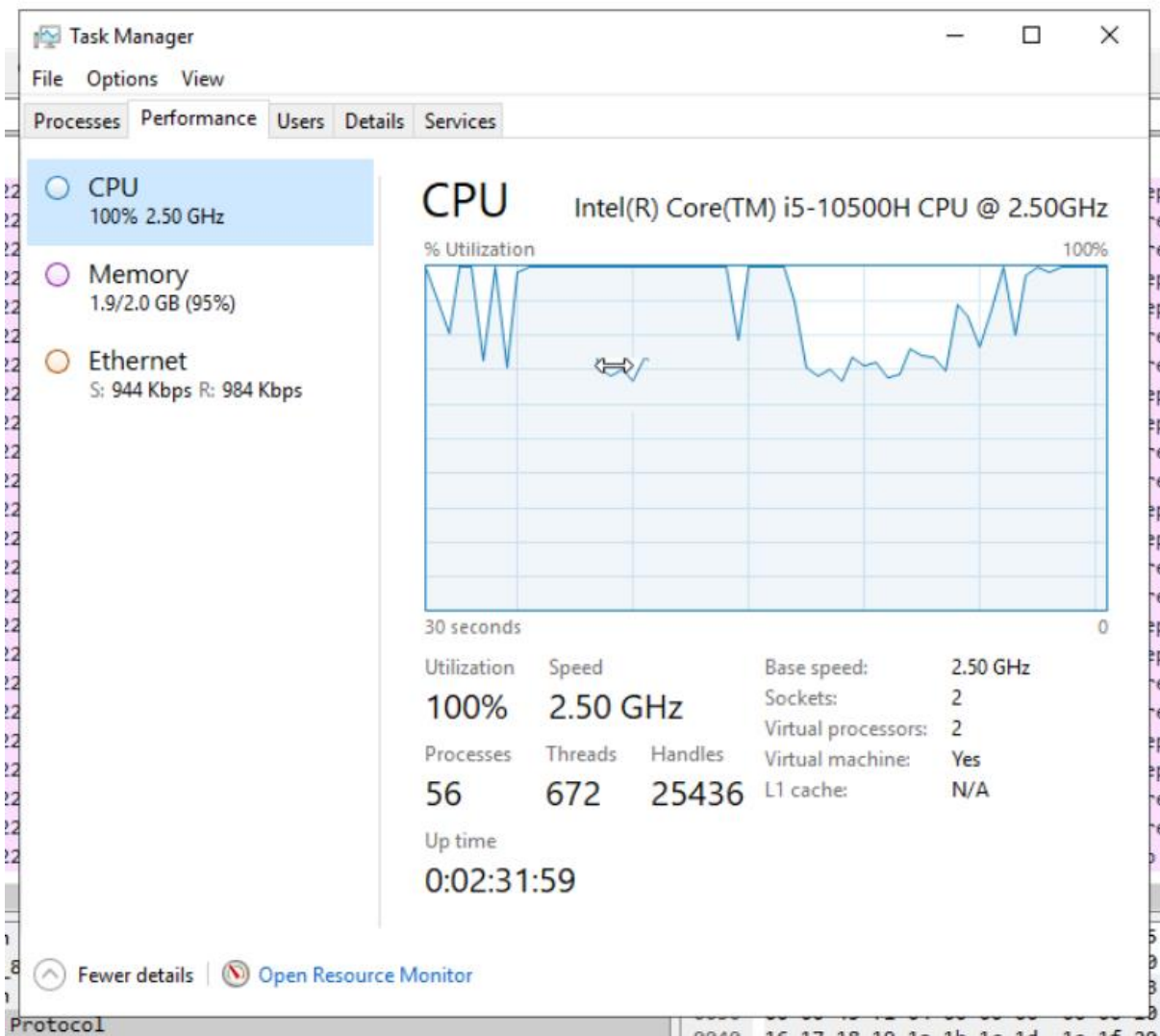
+ Với option “-f” là tấn công flood.



Hình 3.2.2a: Thực hiện tấn công ICMP Flood



Hình 3.2.2c: Kết quả từ công cụ phân tích



Hình 3.2.2c: Server trong cuộc tấn công bị quá tải

PHẦN IV: ỨNG DỤNG CỦA THUẬT TOÁN C4.5 DECISION TREE

4.1. Thuật toán phát hiện tấn công DDoS sử dụng C4.5 Decision Tree

Trong quá trình phân tích dữ liệu và xác định đâu là một cuộc tấn công DDoS. Ta cần kiểm tra tất cả các địa chỉ IP giao tiếp với server. Từ đó thống kê và xác định đâu là những IP đang truy cập bình thường, đâu là những IP đang thực hiện tấn công DDoS vào hệ thống mạng.

Thuật toán xây dựng cây quyết định C4.5 để phân loại sẽ dựa vào Entropy, Gain, và GainRatio để xác định độ biến thiên của những truy cập này. Từ đó xác định đúng đắn và nhanh chóng để bảo vệ hệ thống mạng khỏi những rủi ro đáng tiếc.

Dưới đây là một ví dụ về một cuộc tấn công DDoS được ghi lại dựa trên xác thực truy cập và số lần truy cập. Đây là một ví dụ đơn giản để nghiên cứu thuật toán, trong thực tế, số lượng truy cập có thể rất lớn.

Bảng dữ liệu về sự truy cập

No.	IP Address	Auth	Times
1	199.25.27.82	Yes	12
2	49.119.10.120	Yes	2
3	101.188.10.120	Yes	3
4	87.116.83.101	No	30
5	101.188.67.137	Yes	5
6	87.116.83.101	No	25
7	87.116.83.101	No	10
8	101.32.189.92	Yes	4

Giải thuật:

- Tính Entropy về tình trạng xác thực của các IP Address

$$\begin{aligned}
 Entropy(Auth) &= \sum -p(Yes) * \log_2 p(Yes) - p(No) * \log_2 p(No) \\
 &= -\frac{5}{8} * \log_2 \frac{5}{8} - \frac{3}{8} * \log_2 \frac{3}{8} = 0.954
 \end{aligned}$$

- Lấy Số lần truy cập (Times) để tiến hành kiểm tra việc truy cập của các IP

+ Xét: $Times \leq 10$

$$\begin{aligned}
 Entropy(Auth | Times \leq 10) &= -p(Yes) * \log_2 p(Yes) - p(No) * \log_2 p(No) \\
 &= -\frac{1}{5} * \log_2 \frac{1}{5} - \frac{4}{5} * \log_2 \frac{4}{5} = 0.722
 \end{aligned}$$

+ Xét: $Times > 10$

$$\begin{aligned} \text{Entropy}(\text{Auth} \mid \text{Times} > 10) &= -p(\text{Yes}) * \log_2 p(\text{Yes}) - p(\text{No}) * \log_2 p(\text{No}) \\ &= -\frac{2}{3} * \log_2 \frac{2}{3} - \frac{1}{3} * \log_2 \frac{1}{3} = 0.918 \end{aligned}$$

- Tính Độ khuếch đại (Gain) để thống kê sự thay đổi tăng hay giảm của Entropy.

$$\text{Gain}(\text{Auth}, \text{Times})$$

$$= \text{Entropy}(\text{Auth}) - \sum [p(\text{Auth} \mid \text{Times}) * \text{Entropy}(\text{Auth} \mid \text{Times})]$$

$$\text{Gain}(\text{Auth} \mid \text{Times} <> 10)$$

$$= 0.954 - \frac{5}{8} * 0.722 - \frac{3}{8} * 0.918 = 0.159$$

- Tính Tỷ lệ khuếch đại (GainRatio)

$$1. \text{GainRatio}(\text{Auth}) = \frac{\text{Gain}(\text{Auth})}{\text{Entropy}(\text{Auth})} = 0$$

$$2. \text{GainRatio}(\text{Auth} \mid \text{Times} <> 10) = \frac{\text{Gain}(\text{Auth} \mid \text{Times} <> 10)}{\text{Entropy}(\text{Auth})}$$

$$= \frac{0.159}{0.954} = 0.167$$

Nhận xét:

+ Nhận thấy RatioGain tăng từ 0 lên 0.167 đồng nghĩa với việc đã có sự biến thiên về Entropy.

+ Khi Entropy tăng tức là sự truy cập đã xuất hiện gia tăng khi ta lấy mốc số lần truy cập là 10.

=> **Kết luận được rằng** những truy cập lớn hơn 10 là những truy cập bất thường.

- Lập bảng để dễ dàng nhận thấy **Kết luận** trên là đúng:

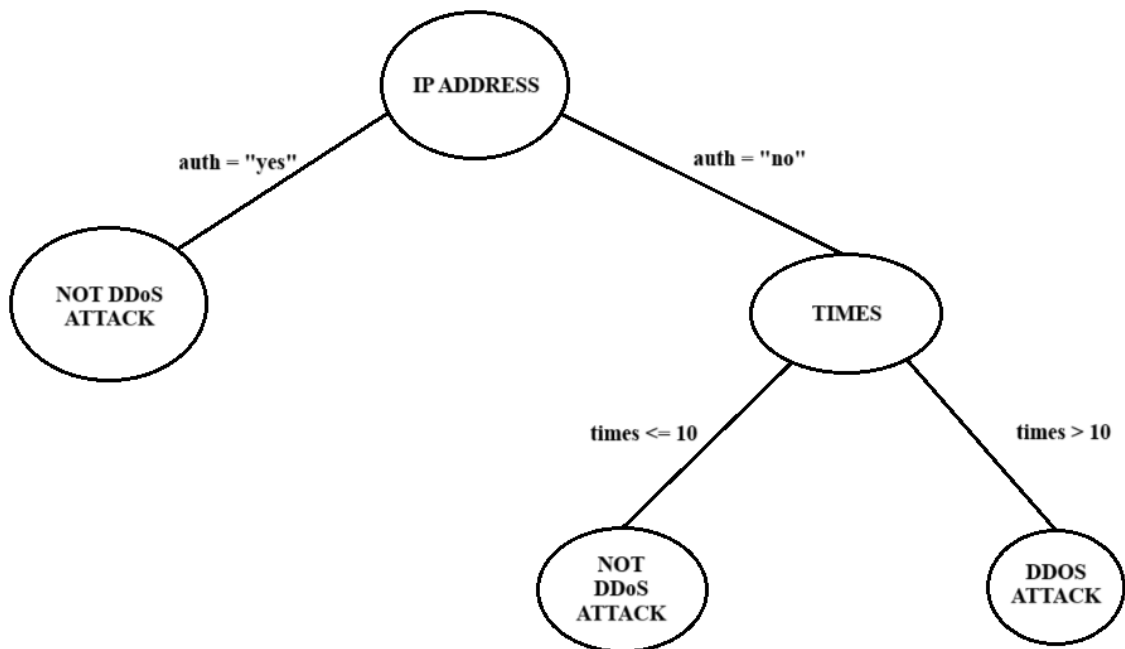
IP Address	Auth	Times (>10)
199.25.27.82	Yes	12
87.116.83.101	No	30
87.116.83.101	No	25

- Phân tích:

+ Số lần truy cập của địa chỉ IP 199.25.27.82 là 12 lần nhưng xác thực là truy cập an toàn.

+ Số lần truy cập của địa chỉ IP 87.116.83.101 lần lượt là 30 lần và 25 lần, tuy nhiên là truy cập không xác thực.

=> **Kết luận được rằng** việc áp dụng thuật toán C4.5 Decision Tree trong phát hiện tấn công Ddos là hoàn toàn khả thi vì thông qua sự chênh lệch Entropy dễ dàng liệt kê và phân tích những truy cập bất thường và đưa ra cảnh báo.



Hình 4.1: Cây quyết định C4.5 xác định tấn công DDoS

4.2. Thuật toán C4.5 Decision Tree để phân loại các cuộc tấn công theo giao thức

4.2.1. Xây dựng và phân loại tấn công

Để tiến hành đánh giá cuộc tấn công thuộc phân loại nào, ta cần đánh nhãn cho loại cho từng hình thức tấn công.

Protocol	Label
ARP_Spoof Attack	A
ARP_Sniffing Attack	B
SSH Brute force Attack	C
HTTP DOS Attack	D
HTTP SQL Injection	E
MAC Flooding	F
ICMP Flooding	G
HTTP Brute force Attack	H

Bảng 4.2.1a: Bảng phân loại tấn công theo nhãn

Attribute	Total cases	Type							
		A	B	C	D	E	F	G	H
Protocol:									
ARP	4	2	2	0	0	0	0	0	0
NBNS	1	0	1	0	0	0	0	0	0
TCP	11	0	0	5	1	0	1	0	4
SSH	6	0	0	6	0	0	0	0	0
HTTP	9	0	0	0	1	3	0	0	5
ICMP	2	0	0	0	0	0	0	2	0
Total:	33	2	3	11	2	3	1	2	9

Bảng 4.2.1b: Bảng thống kê số lần tấn công theo nhãn

- Tạo bảng tổng hợp tất cả các trường hợp và số lần ghi nhận tấn công, tính toán được thống kê dựa trên:

+ Thời gian (Time): 49 ms

+ Địa chỉ Source được hỗ trợ (Support_SourceAddress): 60%

Để xác định chắc chắn rằng đó là một cuộc tấn công thì ta cần xét các yếu tố dựa trên hai tiêu chí đã đặt ra ở trên, và bên cạnh đó cần xác định rằng gói tin có độ dài là bao nhiêu thì xác định đó là dấu hiệu của cuộc tấn công.

4.2.2. Phân loại tấn công dựa trên giao thức TCP

Để xác định rõ loại tấn công, ta cần phải đặt ngưỡng giới hạn độ dài của một gói tin được gửi đến. Cụ thể:

Time (ms)	Suport Source Address	Protocol	Length	Info	Class label
49	60%	TCP	74	ssh [SYN]	SSH brute force attack
49	60%	TCP	66	ssh [ACK]	SSH brute force attack
49	60%	TCP	66	ssh [FIN]	SSH brute force attack
49	60%	TCP	66	ssh>... [ACK]	SSH brute force attack
49	60%	TCP	66	ssh>... [FIN]	SSH brute force attack
49	60%	TCP	294	[TCP segment of reassembled PDU]	HTTP DOS attack
49	60%	TCP	54	[Malformed Packet]	MAC Flooding
49	60%	TCP	74	http [SYN]	HTTP brute force attack
49	60%	TCP	74	http>... [SYN]	HTTP brute force attack
49	60%	TCP	66	http [ACK]	HTTP brute force attack
49	60%	TCP	66	http>... [FIN]	HTTP brute force attack

Bảng 4.2.2: Ngưỡng phát hiện tấn công giao thức TCP

4.2.3. Phân loại tấn công dựa trên giao thức ICMP

Tương tự xác định tấn công qua giao thức TCP, ta đặt ngưỡng giới hạn như sau:

Time (ms)	Support Source Address	Protocol	Length	Info	Class label
49	60%	ICMP	98	Echo (ping) request	ICMP Flooding
49	60%	ICMP	98	Echo (ping) reply	ICMP Flooding

Bảng 4.2.3: Ngưỡng phát hiện tấn công giao thức ICMP

4.3. Xây dựng C4.5 Decision Tree dựa trên các phân loại:

4.3.1. Xây dựng và phân loại tấn công

- Để tiến hành đánh giá cuộc tấn công thuộc phân loại nào, ta cần đánh nhãn cho loại cho từng hình thức tấn công.

Protocol	Label
ARP_Spoof Attack	A
ARP_Sniffing Attack	B
SSH Brute force Attack	C
HTTP DOS Attack	D
HTTP SQL Injection	E
MAC Flooding	F
ICMP Flooding	G
HTTP Brute force Attack	H

Bảng 4.3.1a: Bảng phân loại tấn công theo nhãn

Attribute	Total cases	Type							
		A	B	C	D	E	F	G	H
Protocol:									
ARP	4	2	2	0	0	0	0	0	0
NBNS	1	0	1	0	0	0	0	0	0
TCP	11	0	0	5	1	0	1	0	4
SSH	6	0	0	6	0	0	0	0	0
HTTP	9	0	0	0	1	3	0	0	5
ICMP	2	0	0	0	0	0	0	2	0
Total:	33	2	3	11	2	3	1	2	9

Bảng 4.3.1b: Bảng thống kê số lần tấn công theo nhãn

Tạo bảng tổng hợp tất cả các trường hợp và số lần ghi nhận tấn công, tính toán được thống kê dựa trên:

+ Thời gian (Time): 49 ms

+ Địa chỉ Source được hỗ trợ (Support_SourceAddress): 60%

Để xác định chắc chắn rằng đó là một cuộc tấn công thì ta cần xét các yếu tố dựa trên hai tiêu chí đã đặt ra ở trên, và bên cạnh đó cần xác định rằng gói tin có độ dài là bao nhiêu thì xác định đó là dấu hiệu của cuộc tấn công.

4.3.2. Phân loại tấn công dựa trên giao thức TCP

Để xác định rõ loại tấn công, ta cần phải đặt ngưỡng giới hạn độ dài của một gói tin được gửi đến. Cụ thể:

Time (ms)	Support Source Address	Protocol	Length	Info	Class label
49	60%	TCP	74	ssh [SYN]	SSH brute force attack
49	60%	TCP	66	ssh [ACK]	SSH brute force attack
49	60%	TCP	66	ssh [FIN]	SSH brute force attack
49	60%	TCP	66	ssh>... [ACK]	SSH brute force attack
49	60%	TCP	66	ssh>... [FIN]	SSH brute force attack
49	60%	TCP	294	[TCP segment of reassembled PDU]	HTTP DOS attack
49	60%	TCP	54	[Malformed Packet]	MAC Flooding
49	60%	TCP	74	http [SYN]	HTTP brute force attack
49	60%	TCP	74	http>... [SYN]	HTTP brute force attack
49	60%	TCP	66	http [ACK]	HTTP brute force attack
49	60%	TCP	66	http>... [FIN]	HTTP brute force attack

Bảng 4.3.2: Ngưỡng phát hiện tấn công giao thức TCP

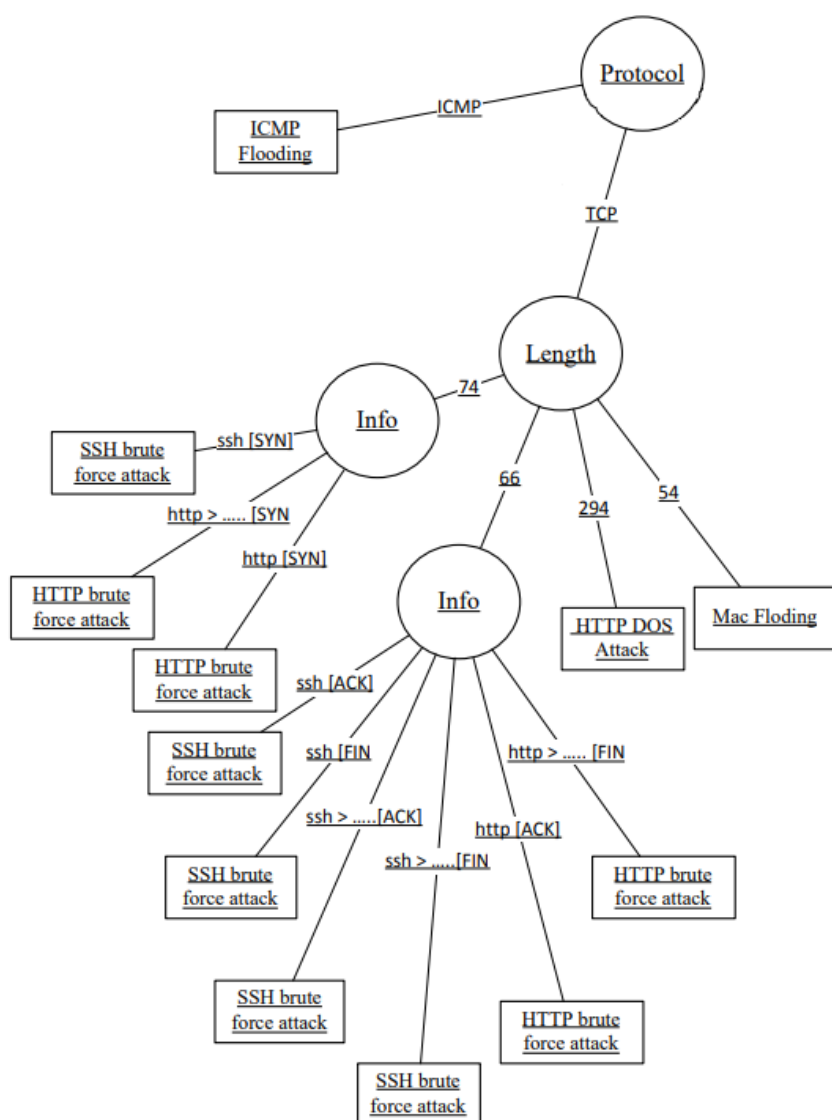
4.3.3. Phân loại tấn công dựa trên giao thức ICMP

Tương tự xác định tấn công qua giao thức TCP, ta đặt ngưỡng giới hạn như sau:

Time (ms)	Support Source Address	Protocol	Length	Info	Class label
49	60%	ICMP	98	Echo (ping) request	ICMP Flooding
49	60%	ICMP	98	Echo (ping) reply	ICMP Flooding

Bảng 4.3.3: Ngưỡng phát hiện tấn công giao thức ICMP

4.3.4. Xây dựng C4.5 Decision Tree dựa trên các phân loại



Hình 4.3.4: Cây quyết định C4.5 phân loại tấn công

PHẦN V: CHƯƠNG TRÌNH ỨNG DỤNG PHÁT HIỆN TẤN CÔNG TỪ CHỐI DỊCH VỤ PHÂN TÁN

Chương trình C++ triển khai thuật toán phát hiện tấn công từ chối dịch vụ phân tán (DDoS) dựa trên việc phân tích lưu lượng mạng.

`ddosAttack.cpp`

```
#include<bits/stdc++.h>
#include<fstream>
#include<sstream>
#include<cmath>
using namespace std;

int main(int argc, char *argv[])
{
    string line, key;
    int t = 0;
    double value, mean, sum, variance, stdDeviation, H, B = 300;
    map<string, int> premapper;
    ifstream myfile (argv[1]);
    cout << "If a DDoS attack is detected the output is specified as follows"
    << endl;
    cout << "output: (Portocol, Source IP, Destination IP, Source Port,
    Destination Port)" << endl;
    if (myfile.is_open())
    {
        while(getline(myfile, line) && (t < 500))
        {
            stringstream lineStream(line);
            string cell;
            vector<string> result;
            while(getline(lineStream, cell, ','))
                result.push_back(cell);
            if (!lineStream && cell.empty())
                result.push_back("");
            key = result[2] + ", " + result[11] + ", " + result[12] + ", " +
            result[13] + ", " + result[14];
            if(premapper.find(key) == premapper.end())
                premapper.insert(make_pair(key, 1));
            else
                premapper[key] += 1;
            t++;
        }
        int check;
        cout << "Number of Iterations: ";
        cin >> check;
        int count = 0;
        while(check > 0)
```

```

{
    cout << "Iteration " << ++count << ": " << endl;
    t = 0;
    map<string, int> mapper;
    while(getline(myfile, line) && (t < 500))
    {
        stringstream lineStream(line);
        string cell;
        vector<string> result;
        while(getline(lineStream, cell, ','))
            result.push_back(cell);
        if (!lineStream && cell.empty())
            result.push_back("");
        key = result[2] + ", " + result[11] + ", " + result[12] + ", "
+ result[13] + ", " + result[14];
        if(mapper.find(key) == mapper.end())
            mapper.insert(make_pair(key, 1));
        else
            mapper[key] += 1;
        t++;
    }
    sum = 0;
    for(auto x : mapper)
        sum += x.second;
    mean = sum / mapper.size();
    for(auto x : mapper)
        variance += pow(x.second - mean, 2);
    variance = variance / mapper.size();
    stdDeviation = sqrt(variance);
    for(auto x : mapper)
    {
        double temp;
        if(premapper.find(x.first) != premapper.end())
        {
            temp = premapper[x.first];
            if(x.second > temp)
                H = log(temp / x.second) - log(x.second / sum);
            else
                H = log(x.second / temp) - log(x.second / sum);
            if (H > 1.5 * mean)
                B++;
            if (H < 0.5 * mean)
                B--;
            if(fabs(mean - H) > B * stdDeviation)
                cout << x.first << endl;
        }
    }
    premapper = mapper;
    check--;
}

```

```

        myfile.close();
    }
    else
        cout << "Unable to open file!";
    return 0;
}

```

Full code Phát hiện tấn công DDoS

5.1. Phân tích thuật toán

Khai báo thư viện:

```

#include<bits/stdc++.h> // Thư viện C++
#include<fstream>       // Thư viện đọc, xuất file
#include<sstream>       // Thư viện thực hiện các chuỗi
#include<cmath>         // Thư viện toán học

```

Khai báo các biến:

```

string line, key;
int t = 0;
double value, mean, sum, variance, stdDeviation, H, B = 300;
map<string, int> premapper; // Sử dụng vector để xác định vị trí
ifstream myfile (argv[1]); // Đọc dữ liệu từ file *txt

```

- + line: Chuỗi để lưu mỗi dòng của tệp đầu vào.
- + key: Chuỗi để đại diện cho một định danh duy nhất cho mỗi bản ghi lưu lượng mạng.
- + t: Biến đếm để theo dõi số lượng bản ghi được xử lý.
- + value, mean, sum, variance, stdDeviation, H, B: Biến dùng cho các tính toán thống kê.
- + premapper: Bản đồ để lưu trữ tần suất của các bản ghi duy nhất trong tập dữ liệu ban đầu.
- + myfile: Dòng dữ liệu đầu vào để đọc dữ liệu lưu lượng mạng.

Vòng lặp xử lý dữ liệu – Phần 1

```

while(getline(myfile, line) && (t < 500))

```


+ Đọc từng dòng từ file *txt cho đến hết file txt và xử lý từ hết 500 bản ghi đầu tiên.

```
stringstream lineStream(line);
    string cell;
    vector<string> result;
    while(getline(lineStream, cell, ','))
        result.push_back(cell);
```

+ Chia mỗi dòng thành các phần bằng cách sử dụng ',' làm dấu phân cách và lưu chúng vào vector “result”.

```
key = result[2] + ", " + result[11] + ", " + result[12] + ", " + result[13] +
", " + result[14];
```

+ Tạo một khóa bằng cách nối các trường cụ thể từ bản ghi để xác định luồng dữ liệu cần xuất ra màn hình.

```
if(mapper.find(key) == mapper.end())
    mapper.insert(make_pair(key, 1));
else
    mapper[key] += 1;
t++;
```

+ Cập nhật tần suất của mỗi luồng lưu trữ sau mỗi lần duyệt và tăng biến đếm t lên 1 lần để duyệt luồng tiếp theo.

Vòng lặp dữ liệu – Phần 2

Sau khi lặp dữ liệu vòng lặp 1 để duyệt hết toàn bộ file, vòng lặp 2 sẽ tiến hành phân tích dữ liệu (data) theo số lần yêu cầu của người dùng.

Với số lần lặp càng nhiều, khả năng phát hiện các tấn công sẽ hiển thị nhiều và đầy đủ nhất.

```
int check;
cout << "Number of Iterations: ";
cin >> check;
```

+ Khởi tạo biến “check” để người dùng nhập số lần lặp.

```
stringstream lineStream(line);
string cell;
vector<string> result;
while(getline(lineStream, cell, ','))
    result.push_back(cell);
```

+ Tương tự vòng lặp ở Phần 1, ta tiếp tục xử lý từng dòng data trong *txt và xuất thông tin ra màn hình.

```
sum = 0;
for(auto x : mapper)
    sum += x.second;
```

+ Tính tổng các tần suất lặp lại của dữ liệu trong file dữ liệu đầu vào.

```
mean = sum / mapper.size();
```

+ Tính trung bình của các tần suất lặp lại

```
for(auto x : mapper)
    variance += pow(x.second - mean, 2);
variance = variance / mapper.size();
```

+ Tính phương sai của các tần suất lặp lại

```
stdDeviation = sqrt(variance);
```

+ Tính độ lệch chuẩn

Vòng lặp phân tích luồng dữ liệu

```
for(auto x : mapper)
```

+ Duyệt qua mỗi luồng trong tập dữ liệu hiện tại

```
double temp;
if(premapper.find(x.first) != premapper.end())
{
    temp = premapper[x.first];
    if(x.second > temp)
```

```

        H = log(temp / x.second) - log(x.second / sum);
    else
        H = log(x.second / temp) - log(x.second / sum);
    if (H > 1.5 * mean)
        B++;
    if (H < 0.5 * mean)
        B--;
    if(fabs(mean - H) > B * stdDeviation)
        cout << x.first << endl;
}

```

+ Phân tích mỗi luồng đang sử dụng để tính toán. Thuật toán sẽ in ra những thông tin về những IP đang thực hiện tấn công DDoS nếu phát hiện độ nhảy Entropy có sự biến thiên.

```

premapper = mapper;
check--;

```

+ Cập nhật bản đồ phân tích cho lần lặp tiếp theo

=> Cuối cùng đóng tập tin và kết thúc chương trình phân tích.

5.2. Kiểm thử chương trình

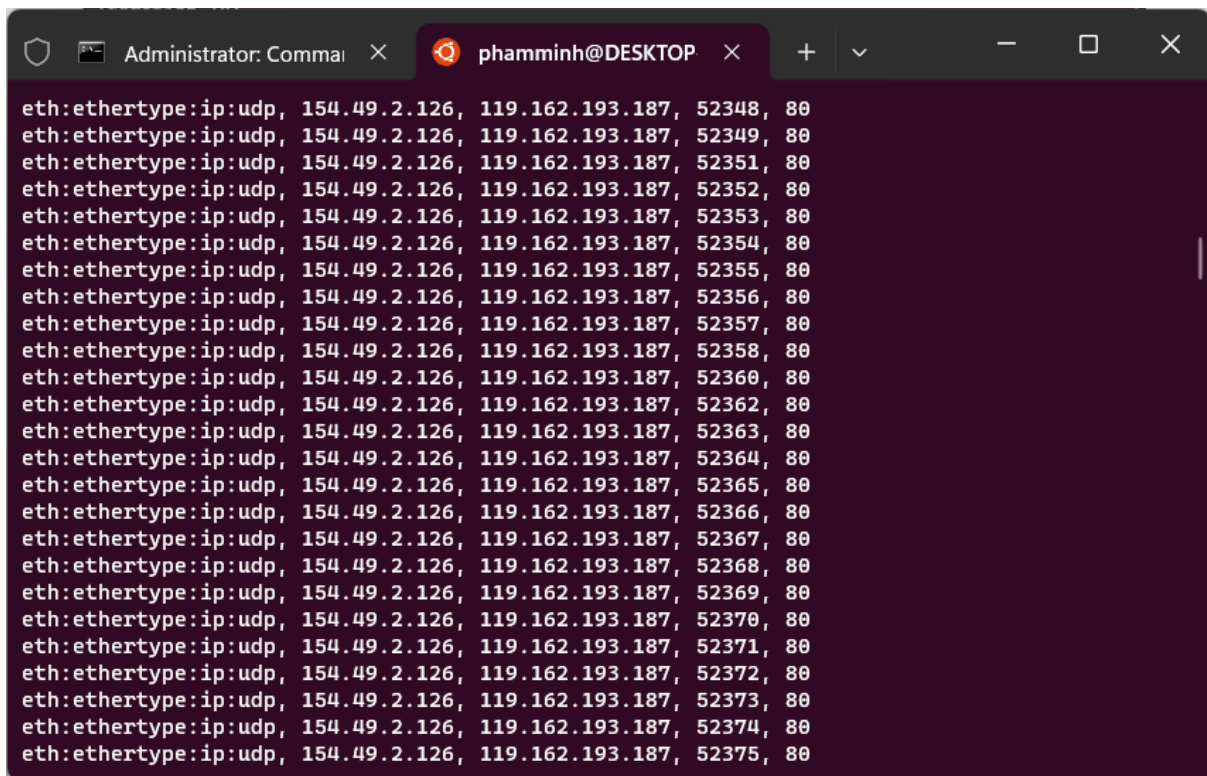
Ta sẽ kiểm tra file data.txt ghi nhận tất cả các địa chỉ IP đã truy cập và được hệ thống ghi nhận lại. Và duyệt với số lần lặp là 100 lần

```
Administrator: Comm... x phamminh@DESKTOP x + v - □ x
phamminh@DESKTOP-1SMS1CH:~$ cd DDoS_Attack
phamminh@DESKTOP-1SMS1CH:~/DDoS_Attack$ ./ddosAttack data.txt
If a DDoS attack is detected the output is specified as follows
output: (Portocol, Source IP, Destination IP, Source Port, Destination Port)
Number of Iterations: 100
Iteration 1:
Iteration 2:
Iteration 3:
Iteration 4:
Iteration 5:
Iteration 6:
Iteration 7:
Iteration 8:
eth:ethertype:ip:icmp, 120.215.126.182, 119.225.16.20, 3389, 80
Iteration 9:
eth:ethertype:ip:icmp, 120.215.126.182, 119.225.16.20, 3389, 80
Iteration 10:
eth:ethertype:ip:icmp, 120.215.126.182, 119.225.16.20, 3389, 80
Iteration 11:
eth:ethertype:ip:icmp, 120.215.126.182, 119.225.16.20, 3389, 80
Iteration 12:
eth:ethertype:ip:icmp, 120.215.126.182, 119.225.16.20, 3389, 80
Iteration 13:
eth:ethertype:ip:icmp, 120.215.126.182, 119.225.16.20, 3389, 80
Iteration 14:
```

Hình 5.2a: Quá trình kiểm thử

```
Administrator: Comm... x phamminh@DESKTOP x + v - □ x
eth:ethertype:ip:icmp, 120.215.126.182, 119.225.16.20, 3389, 80
Iteration 16:
Iteration 17:
Iteration 18:
eth:ethertype:ip:udp, 154.49.2.126, 119.162.193.187, 52568, 80
Iteration 19:
eth:ethertype:ip:udp, 154.49.2.126, 119.162.193.187, 52471, 80
Iteration 20:
eth:ethertype:ip:udp, 154.49.2.126, 119.162.193.187, 52570, 80
Iteration 21:
Iteration 22:
eth:ethertype:ip:tcp, 119.162.193.187, 154.49.2.126, 80, 52358
eth:ethertype:ip:tcp, 119.162.193.187, 154.49.2.126, 80, 52368
eth:ethertype:ip:tcp, 119.162.193.187, 154.49.2.126, 80, 52378
eth:ethertype:ip:tcp, 119.162.193.187, 154.49.2.126, 80, 52391
eth:ethertype:ip:tcp, 119.162.193.187, 154.49.2.126, 80, 52417
eth:ethertype:ip:tcp, 119.162.193.187, 154.49.2.126, 80, 52426
eth:ethertype:ip:tcp, 119.162.193.187, 154.49.2.126, 80, 52434
eth:ethertype:ip:tcp, 119.162.193.187, 154.49.2.126, 80, 52442
eth:ethertype:ip:tcp, 119.162.193.187, 154.49.2.126, 80, 52450
eth:ethertype:ip:tcp, 119.162.193.187, 154.49.2.126, 80, 52458
eth:ethertype:ip:tcp, 119.162.193.187, 154.49.2.126, 80, 52465
eth:ethertype:ip:tcp, 119.162.193.187, 154.49.2.126, 80, 52472
eth:ethertype:ip:tcp, 119.162.193.187, 154.49.2.126, 80, 52479
eth:ethertype:ip:tcp, 119.162.193.187, 154.49.2.126, 80, 52486
```

Hình 5.2b: Quá trình kiểm thử



```
eth:ethertype:ip:udp, 154.49.2.126, 119.162.193.187, 52348, 80
eth:ethertype:ip:udp, 154.49.2.126, 119.162.193.187, 52349, 80
eth:ethertype:ip:udp, 154.49.2.126, 119.162.193.187, 52351, 80
eth:ethertype:ip:udp, 154.49.2.126, 119.162.193.187, 52352, 80
eth:ethertype:ip:udp, 154.49.2.126, 119.162.193.187, 52353, 80
eth:ethertype:ip:udp, 154.49.2.126, 119.162.193.187, 52354, 80
eth:ethertype:ip:udp, 154.49.2.126, 119.162.193.187, 52355, 80
eth:ethertype:ip:udp, 154.49.2.126, 119.162.193.187, 52356, 80
eth:ethertype:ip:udp, 154.49.2.126, 119.162.193.187, 52357, 80
eth:ethertype:ip:udp, 154.49.2.126, 119.162.193.187, 52358, 80
eth:ethertype:ip:udp, 154.49.2.126, 119.162.193.187, 52360, 80
eth:ethertype:ip:udp, 154.49.2.126, 119.162.193.187, 52362, 80
eth:ethertype:ip:udp, 154.49.2.126, 119.162.193.187, 52363, 80
eth:ethertype:ip:udp, 154.49.2.126, 119.162.193.187, 52364, 80
eth:ethertype:ip:udp, 154.49.2.126, 119.162.193.187, 52365, 80
eth:ethertype:ip:udp, 154.49.2.126, 119.162.193.187, 52366, 80
eth:ethertype:ip:udp, 154.49.2.126, 119.162.193.187, 52367, 80
eth:ethertype:ip:udp, 154.49.2.126, 119.162.193.187, 52368, 80
eth:ethertype:ip:udp, 154.49.2.126, 119.162.193.187, 52369, 80
eth:ethertype:ip:udp, 154.49.2.126, 119.162.193.187, 52370, 80
eth:ethertype:ip:udp, 154.49.2.126, 119.162.193.187, 52371, 80
eth:ethertype:ip:udp, 154.49.2.126, 119.162.193.187, 52372, 80
eth:ethertype:ip:udp, 154.49.2.126, 119.162.193.187, 52373, 80
eth:ethertype:ip:udp, 154.49.2.126, 119.162.193.187, 52374, 80
eth:ethertype:ip:udp, 154.49.2.126, 119.162.193.187, 52375, 80
```

Hình 5.2c: Quá trình kiểm thử

5.3. Đánh giá

Thuật toán dựa trên độ nhảy và số dựa vào lần học đầu tiên là 500 lần duyệt đầu, sau đó khi người dùng nhập số lần duyệt vào vòng lặp 2, những IP nào có tần suất truy cập nhiều bất thường sẽ khiến độ lệch chuẩn thay đổi. Sau đó xuất ra màn hình những thông tin liên quan đến địa chỉ IP nghi vấn tấn công đó như: *giao thức tấn công, địa chỉ IP nguồn, địa chỉ IP đích, địa chỉ Port nguồn, địa chỉ Port đích*.

PHẦN VI: KẾT LUẬN

6.1. Về nội dung đồ án

Thông qua quá trình thực hiện đồ án chuyên ngành với đề tài “NGHIÊN CỨU ỨNG DỤNG THUẬT TOÁN C4.5 DECISION TREE” bản thân sinh viên đã tích lũy thêm những kinh nghiệm thực tế về ngành học An toàn thông tin tại trường Đại học Công thương Thành phố Hồ Chí Minh. Kết hợp với những nội dung yêu cầu tìm hiểu, nghiên cứu về đề tài. Cá nhân sinh viên nhận thấy đây là một đề tài bổ ích, có tính thực

nghiệm cao, tương đối dễ tiếp cận và không yêu cầu quá nhiều những phân tích phức tạp.

Tuy nhiên, bản thân sinh viên chưa có nhiều cơ hội được áp dụng nghiên cứu vào thực tiễn để đánh giá mức độ rủi ro bảo mật của một hệ thống. Song việc phân tích dữ liệu (data) mẫu vẫn còn là một thách thức đối với cá nhân sinh viên thực hiện vì chưa có nhiều kiến thức về các file log hệ thống. Việc phân tích chỉ dựa trên data mẫu ít nhiều vẫn còn tồn đọng những mặt khuyết điểm như: không đa dạng về môi trường phân tích hệ thống, không nắm rõ được hết các nhật kí của các loại thiết bị bảo mật như tường lửa, nhật kí hệ thống của server,...

Trong tương lai, ở học phần “Khóa luận tốt nghiệp”, bản thân sinh viên sẽ tiếp tục phát triển đề tài này một cách chi tiết hơn, đầy đủ hơn và dữ liệu đa dạng hơn.

6.2. Về quá trình thực hiện đồ án

Trong quá trình thực hiện đồ án, bản thân sinh viên đã học được tính tự lập, tính sáng tạo, tinh thần áp dụng kiến thức đã học được vào phát triển đồ án đến tiến độ hoàn thiện đúng với mục tiêu đề ra.

Mặc dù vẫn còn nhiều khuyết điểm, những nội dung tuy chưa hoàn tất vì vốn hiểu biết vẫn còn hạn chế. Song, cá nhân sinh viên đã vượt qua và khắc phục những tồn đọng tiêu cực đấy.

Đánh giá những hạn chế nhận ra trong quá trình thực hiện đồ án:

- + Chưa thể khai thác hết tất cả những cuộc tấn công hệ thống mạng.
- + Chưa thể phân tích các file nhật kí hệ thống của thiết bị tường lửa, thiết bị bảo mật,...
- + Chỉ thực hiện được trong phạm vi hạ tầng mạng nhỏ, cần thực hiện trên hạ tầng mạng rộng hơn để đánh giá khách quan tính thiết thực của nghiên cứu.

TÀI LIỆU THAM KHẢO

- [1] Microsoft Security, “2022 in review: DDoS attack trends and insights”.
- [2] Nguyễn Văn Chức, “*applying decision tree technique in data mining to build a consultant system for choosing majors for university entrance examination*”, (trích mục 2 Giới thiệu về kỹ thuật phân lớp dữ liệu dựa vào cây quyết định, Dec 2013).
- [Alka Gangrade, Ravindra Patel], “*building privacy-preserving c4.5 decision tree classifier on multi-parties*”.
- [Made Sudarma, Dandy Pramana Hostiadi], “*The Establishment of Decision Tree Model in Network Traffic Incident Using C4.5 Method*”, (trích. International Journal of Informatics and Communication Technology (IJ-ICT) Vol.3, No.1, February 2014, pp. 23~29).