ĐẠI HỌC QUỐC GIA HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

HỒ CÔNG LONG
TRẦN VĂN THÁI
NGUYỄN LÊ THẢO NGỌC
NGUYỄN HOÀNG CÔNG THÀNH

ĐỒ ÁN AN TOÀN MẠNG

MỘT CÁCH TIẾP CẬN MỚI DỰA TRÊN PHÂN TÍCH TRỰC TIẾP THÍCH ỨNG CỦA LƯU LƯỢNG ĐƯỢC MÃ HÓA ĐỂ XÁC ĐỊNH PHẦN MỀM ĐỘC HẠI TRONG NGÀNH CÔNG NGHIỆP IOT

A NOVEL APPROACH BASED ON ADAPTIVE ONLINE ANALYSIS OF ENCRYPTED TRAFFIC FOR IDENTIFYING MALWARE IN IIOT

TP. Hồ Chí Minh, 2023

ĐẠI HỌC QUỐC GIA HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

HỒ CÔNG LONG - 21522297
TRẦN VĂN THÁI - 21522583
NGUYỄN LÊ THẢO NGỌC - 21521191
NGUYỄN HOÀNG CÔNG THÀNH - 21522598

ĐỒ ÁN AN TOÀN MẠNG

MỘT CÁCH TIẾP CẬN MỚI DỰA TRÊN PHÂN TÍCH TRỰC TIẾP THÍCH ỨNG CỦA LƯU LƯỢNG ĐƯỢC MÃ HÓA ĐỂ XÁC ĐỊNH PHẦN MỀM ĐỘC HẠI TRONG NGÀNH CÔNG NGHIỆP IOT

A NOVEL APPROACH BASED ON ADAPTIVE ONLINE ANALYSIS OF ENCRYPTED TRAFFIC FOR IDENTIFYING MALWARE IN IIOT

MÔN HỌC: AN TOÀN MẠNG
LỚP: NT140.O11.ANTT

GIÁO VIÊN LÝ THUYẾT
ThS. Nghi Hoàng Khoa

TP.Hồ Chí Minh - 2023

LỜI CẢM ƠN

Để hoàn thành đồ án môn học này, chúng tôi xin tỏ lòng biết ơn sâu sắc đến Thầy ThS. Nghi Hoàng Khoa đã hướng dẫn, cung cấp kiến thức, kinh nghiệm quý báu trong suốt quá trình thực hiện đồ án. Sự tận tâm và sự đồng hành của thầy đã đóng góp không nhỏ vào việc hoàn thành đồ án môn học của chúng tôi.

Mặc dù , nhóm nghiên cứu đã có nhiều cố gắng để thực hiện đồ án một cách hoàn chỉnh nhất song không thể tránh khỏi thiếu sót, rất mong nhận được được những đóng góp của thầy.

Sau cùng, nhóm chúng tôi xin gửi đến ThS. Nghi Hoàng Khoa, gia đình, bạn bè thân thiết và mọi người lời kính chúc sức khỏe và lời tri ân chân thành nhất.

TP. Hồ Chí Minh, 2023
Trân trọng./.
Nguyễn Hoàng Công Thành, Trần Văn Thái,
Nguyễn Lê Thảo Ngọc, Hồ Công Long

MỤC LỤC

DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

AARF: Advanced adaptive random forest module
IARF: Improved adaptive random forest
ARF: Adaptive random forest

DANH MỤC CÁC HÌNH VẼ

DANH MỤC CÁC BẢNG BIỂU

TÓM TẮT ĐỀ TÀI

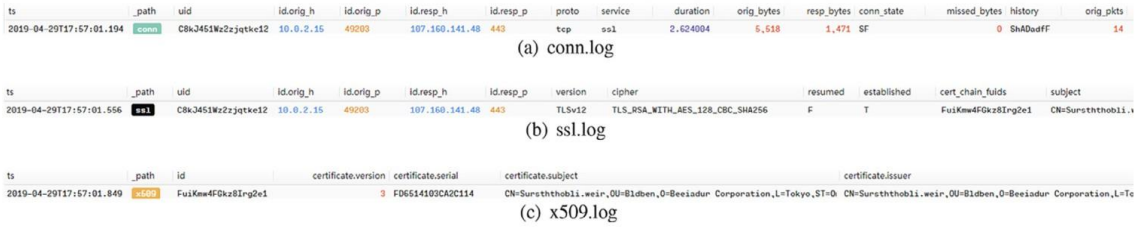
Với sự xuất hiện liên tục của phần mềm độc hại mới là mối de dọa nghiêm trọng đối với ngành công nghiệp iot (IIOT), đồng thời việc xác định các phần mềm độc hại thông việc phát hiện lưu lượng độc hại trong các luồng lưu lượng truy cập được mã hóa, trôi dạt là 1 thách thức. Những phương pháp như phát hiện gói sâu (DPI) thì không hiệu quả trong lưu lượng được mã hóa, đối với phương pháp man-in-the-middle có thể phát hiện lưu lượng được mã hóa thông qua việc giải mã và kiểm tra nội dung văn bản gốc, tuy nhiên việc này xung đột với mã hóa dữ liệu để bảo mật và tốn nhiều tài nguyên để tái mã hóa. Giải pháp được đưa ra là sử dụng các kĩ thuật học máy để phân tích tiêu đề không được mã hóa và các tính năng thống kê của dữ liệu lưu lượng mã hóa. Những thách thức khi triển khai 1 số mô hình thực tế là dữ liệu để huấn luyện không được lấy đầy đủ trong giai đoạn đầu mà được cung cấp liên tục dưới dạng dữ liệu truyền phát và phải thường xuyên cập nhật, dữ liệu mất cân bằng các mô hình dựa trên học máy sẽ thiên về lành tính , dẫn đến suy giảm hiệu suất phân loại. Giải pháp được bài báo nguyên cứu đề xuất một phương pháp mới dựa trên phân tích trực tiếp thích ứng để xác định chính xác các nhóm phần mềm độc hại bằng cách phân tích luồng lưu lượng được đề cập ở trên. Cách tiếp cận này thì được dựa trên improve adaptive random forest (IARF), dựa trên adaptive random forest (ARF) để có khả năng cập nhật thích ứng các tham số khi xử lý các loại lưu lượng phần mềm độc hại mới trong luồng lưu lượng truy cập và nhạy cảm với các nhóm phần mềm độc hại có ít mẫu trong lưu lượng không cân bằng. Hơn nữa với tính năng trích xuất các gói lưu lượng được mã hóa thông qua Brim, một công cụ phân tích mạng nguồn mở , bài báo đề xuất một phương pháp phát hiện lưu lượng độc hại được mã hóa

dựa trên Brim và IARF. Cách tiếp cận này nhằm mục đích xác định chính xác các họ phần mềm độc hại trong các luồng lưu lượng truy cập được mã hóa, trôi dạt và mất cân bằng

CHƯƠNG 1. KIẾN THỨC NỀN TẢNG

1.1. Brim

Là 1 công cụ phân tích mạng nguồn mở được xây dựng từ một số thành phần nguồn mở khác: Zed, Zeek,.. Brim đọc các tệp pcap và chuyển đổi chúng thành nhật ký Zeek sau đó xuất nhật ký dưới dạng csv. Các nhật ký cần trích xuất trong demo gồm conn.log ghi lại thông tin về chi tiết kết nối IP, TCP và ICMP; ssl.log ghi lại chi tiết thông tin bắt tay ssl; và x509.log ghi lại dữ liệu về phân tích chứng chỉ x509.



ts	_path	uid	id.orig_h	id.orig_p	id.resp_h	id.resp_p	proto	service	duration	orig_bytes	resp_bytes	conn_state	missed_bytes	history	orig_pkts
2019-04-29T17:57:01.194	conn	C8kJ451Wz2zjqtke12	10.0.2.15	49203	107.160.141.48	443	tcp	ssl	2.624004	5,518	1,471	SF	0	ShADadfF	14

(a) conn.log

ts	_path	uid	id.orig_h	id.orig_p	id.resp_h	id.resp_p	version	cipher	resumed	established	cert_chain_fuids	subject
2019-04-29T17:57:01.556	ssl	C8kJ451Wz2zjqtke12	10.0.2.15	49203	107.160.141.48	443	TLSv12	TLS_RSA_WITH_AES_128_CBC_SHA256	F	T	FuiKmw4FGkz8Irg2e1	CN=Sursththobli.…

(b) ssl.log

ts	_path	id	certificate.version	certificate.serial	certificate.subject	certificate.issuer
2019-04-29T17:57:01.849	x509	FuiKmw4FGkz8Irg2e1	3	FD6514103CA2C114	CN=Sursththobli.weir,OU=Bldben,O=Beeiadur Corporation,L=Tokyo,ST=O	CN=Sursththobli.weir,OU=Bldben,O=Beeiadur Corporation,L=To

(c) x509.log

Hình 1.1 1 bản ghi cùng phiên trong Brim

Trong conn.log, mỗi bản ghi tương ứng với một phiên mạng, tức là một quá trình giao tiếp giữa máy khách và máy chủ mở ra bằng ba cái bắt tay và đóng bằng một kết nối đóng. Đối với một phiên được mã hóa các nhật ký được liên kết với nhau bằng các khóa tương quan: “UID” của conn.log - “UID” của ssl.log và “cert_chain funds” của ssl.log – “ID” của x509.log.

1.2. ARF

Thuật toán ARF là sự chuyển thể của thuật toán của RF sang truyền dữ liệu. RF ban đầu phát triển các cây quyết định dựa trên base classifiers và quyết định cuối cùng bằng cách bo phiếu dựa trên cây quyết định . Cơ chế ngẫu nhiên trong rừng ngẫu nhiên làm cho nó có khả năng chống lại tình trạng quá khớp và không nhạy cảm với các giá trị mặc định của mẫu. Cơ chế ngẫu nhiên xuất phát từ 2 thành phần: (1) Nếu kích thước đặc điểm của mẫu là M, hằng số $m \ll M$, chọn ngẫu nhiên m đặc điểm từ M đặc điểm và lựa chọn tính năng tốt nhất từ bộ tính năng này trước khi cây chuẩn bị phân chia. (2) Nếu kích thước của mẫu huấn luyện là N, cho mỗi cây, n mẫu huấn luyện ($n<N$) được lấy từ tập huấn luyện theo cách bootstrap

để tạo thành tập con cho huấn luyện .

Điều kiện tiên quyết để cơ chế ngẫu nhiên này hoạt động hiệu quả là phải có bộ dữ liệu đầy đủ, điều này xung đột với dữ liệu truyền phát. Để thích ứng của RF với dữ liệu truyền phát yêu cầu quy trình tổng hợp bootstrap trực tuyến thích hợp và giới hạn quyết định phân chia lá. conn.log, đối với yêu cầu quy trình tổng hợp bootstrap trực tuyến, ARF áp dụng thuật toán online bagging với Poision (λ = 6). ARF bất cứ khi nào một nút được tạo ra trong cây cơ sở , một tập con tính năng sẽ được chọn và lựa chọn phân tách tiếp theo dựa trên tập con tính năng này.[2]

Algorithm 2 Adaptive random forests. **Symbols:** m: maximum features evaluated per split; n: total number of trees ($n = |T|$); δ_w: warning threshold; δ_d: drift threshold; $c(\cdot)$: change detection method; S: Data stream; B: Set of background trees; $W(t)$: Tree t weight; $P(\cdot)$: Learning performance estimation function.

1: **function** ADAPTIVERANDOMFORESTS(m, n, δ_w, δ_d)
2: $T \leftarrow CreateTrees(n)$
3: $W \leftarrow InitWeights(n)$
4: $B \leftarrow \emptyset$
5: **while** $HasNext(S)$ **do**
6: $(x, y) \leftarrow next(S)$
7: **for all** $t \in T$ **do**
8: $\hat{y} \leftarrow predict(t, x)$
9: $W(t) \leftarrow P(W(t), \hat{y}, y)$
10: $RFTreeTrain(m, t, x, y)$ ▷ Train t on the current instance (x, y)
11: **if** $C(\delta_w, t, x, y)$ **then** ▷ Warning detected?
12: $b \leftarrow CreateTree()$ ▷ Init background tree
13: $B(t) \leftarrow b$
14: **end if**
15: **if** $C(\delta_d, t, x, y)$ **then** ▷ Drift detected?
16: $t \leftarrow B(t)$ ▷ Replace t by its background tree
17: **end if**
18: **end for**
19: **for all** $b \in B$ **do** ▷ Train each background tree
20: $RFTreeTrain(m, b, x, y)$
21: **end for**
22: **end while**
23: **end function**

Hình 1.2: Thuật toán ARF

1.3 IARF

Một phương pháp mới được đề cập trong bài báo được cải tiến dựa trên ARF với việc thay đổi phương pháp lấy mẫu.Ý tưởng là làm cho bộ phân loại mẫu trước khi nhận được mẫu mới và nếu dự đoán đúng, mẫu sẽ bị loại bỏ trong quá trình huấn luyện bộ phân loại, chỉ sử dụng mẫu không chính xác để đào tạo.

Algorithm 1: Improved Adaptive Random Forests Algorithm

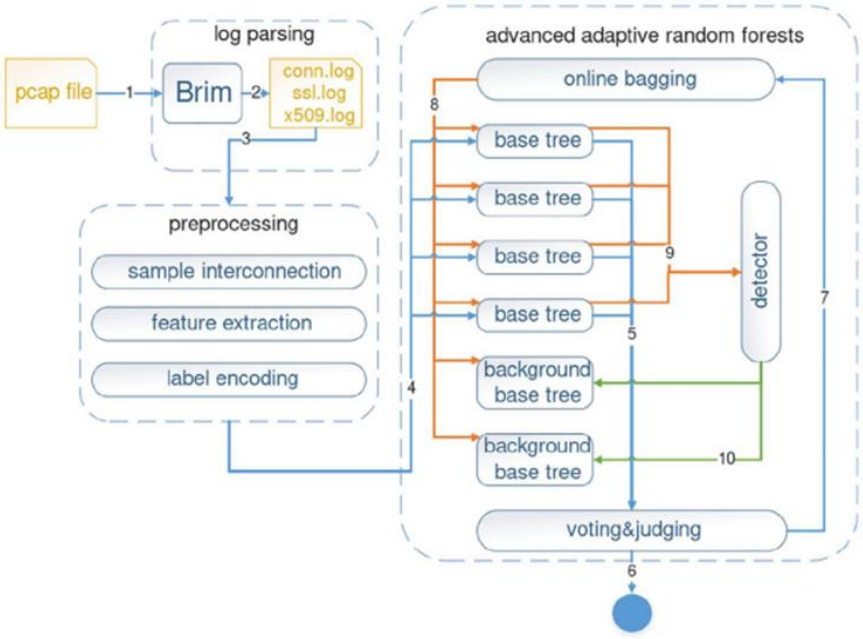
Symbols: **m**:maximum features for per split; **n**:number of base trees; **α**:warning threshold; **β**:drift threshold; **F**:forest of base trees; **D**:data stream;
BT:background trees; **W(t)**:weight of Tree(t) **P**: learning performance estimation function; **nt**:new background tree.

```
1: Function TrainImprovedAdaptiveRandomForests(m,n,α,β)
2: while HasNext(D) do
3:    (x,y) ← next(D)
4:    y^p ← F.predict(x)
5:    if NotSame(y, y^p) then
6:       for all t ∈ F do
7:          y' ← t.predict(x)
8:          W(t) ← P(W(t),y',y)
9:          BaseTreeTrain(x, y, t, m)
10:         if WarningDetected(x, y, t, α) then
11:            nt ← CreateTree()
12:            B(t) ← nt
13:         if DriftDetected(x, y, t, β) then
14:            t ← B(t)
15:         for all nt ∈ B bf do
16:            BaseTreeTrain(x, y, t, m)
17: End function
```

Hình 1.3 Thuật toán IARF

CHƯƠNG 2. PHƯƠNG PHÁP TIẾP CẬN

2.1. Mô hình



Hình 2.1: Kiến trúc mô hình dựa trên IARF

Mô hình gồm 3 giai đoạn: log parsing, preprocessing và AARF

Các bước tiến hành:

(1) Nhập tệp pcap vào brim

(2) Xử lý tệp pcap thông qua brim để trích xuất nhật ký zeek - Conn.log, ssl.log, x509. log

(3) Đưa các nhật ký vào preprocessing ,nhiệm vụ của nó là kết nối bản ghi, trích xuất tính năng và mã hóa nhãn

(4) Gửi mẫu vào từng base tree để kiểm tra

(5) Base tree gửi kết quả đến module voting&judging để đánh giá ,đưa ra kết quả dựa trên biểu quyết đa số

(6) Xuất kết quả dự đoán

(7) Kết quả dự đoán sai sẽ gửi lại về module online-baging để train lại mô hình với mẫu dự đoán sai

(8) Module online-baging sẽ phân phối mẫu đến các base tree và cả các background tree. Sau đó cây nhận mẫu sẽ được huấn luyện

(9) Thực hiện cảnh báo và phát hiện độ lệch

(10) Giá trị phát hiện độ lệch của base tree đến ngưỡng cảnh báo sẽ tạo 1 background tree; nếu đạt đến ngưỡng trôi thì thay thế base bằng background tree.

2.2 Chuyển đổi thành log và tiền xử lý

Học Sử dụng Brim để phân tích các tệp pcap và tạo nhật ký Zeek để trích xuất tính năng từ 3 file conn.log, ssl.log, x509.log. Đối với các phiên mã hóa các bản ghi nằm rải rác nhau trong 3 nhật ký này nên kết nối chúng lại thông qua các khóa được mô tả ở phần 1.1. Một số trang web độc hại có thể ngụy trang bằng cách đăng ký chứng chỉ hợp pháp miễn phí , vì vậy ta cần thêm tính năng let_encrypt để xem xác định xem chứng chỉ của phiên đã được đăng ký tại letencyprt.org hay chưa. Mã hóa nhãn các đặc điểm phân loại thành số nguyên. Sử dụng gồm 29 fearture để train và đánh giá mô hình:

Features extracted from conn.log, ssl.log and x509.log

Num	Feature name	Log	Description
1	proto	conn.log	The transport layer protocol of the connection
2	service	conn.log	An identification of an application protocol being sent over the connection
3	duration	Conn.log	How long the connection lasted
4	conn_state	conn.log	Connection state
5	history	conn.log	Records the state history of connections
6	orig_bytes	conn.log	The number of payload bytes the originator sent
7	resp_bytes	conn.log	The number of payload bytes the responder sent
8	orig_pkts	conn.log	Number of packets that the originator sent
9	orig_ip_bytes	conn.log	Number of IP level bytes that the originator sent
10	resp_pkts	conn.log	Number of packets that the responder sent
11	resp_ip_bytes	conn.log	Number of IP level bytes that the responder sent
12	missed_bytes	conn.log	Indicates the number of bytes missed in content gaps
13	version	ssl.log	SSL/TLS version that the server chose.
14	cipher	ssl.log	SSL/TLS cipher suite that the server chose
15	server_name_len	ssl.log	Length of the Server Name Indicator SSL extension
16	resumed	ssl.log	Flag to indicate if the session was resumed
17	established	ssl.log	Flag to indicate if this ssl session has been established
18	next_protocol	ssl.log	successfully

19	validation_status	ssl.log	Result of certificate validation for this connection
20	certificate.version	x509.log	The version of the encountered SCT
21	certificate.key_alg	x509.log	Name of the key algorithm
22	certificate.sig_alg	x509.log	Name of the signature algorithm
23	certificate.key_type	x509.log	Key type, if key parseable openssl
24	certificate.key_length	x509.log	Length of the key
25	curve	x509.log	Curve, if EC–certificate
26	san_dns_num	x509.log	Numbers of DNS entries in the SAN
27	san_ip_num	x509.log	Numbers of IP entries in the SAN
28	basic_constraints.ca	x509.log	Basic constraints extension of the certificate
29	lets_encrypt	x509.log	Flag to indicate if the certificate was registered at letsencrypt.org

Bảng 1: Danh sách Feartures

CHƯƠNG 3. DEMO

3.1. Kịch bản demo

So sánh mô hình áp dụng thuật toán IARF với các mô hình sử dụng thuật toán khác như ARF, RF, LRCV, LSVC, GBDT.

3.2. Thu thập dataset

Dữ liệu được nhóm em thu thập thủ công các file pcap và file csv kết hợp từ malware-traffic-analysis.net, MCFP dataset, và CTU-13 datase.Sau đó sử dụng code để tự động phân tích và tổng hợp dataset trên command line thay vì Brim desktop. Tất cả các dữ liệu này đều chứa dữ liệu lưu lượng được mã hóa. Vì trong môi trường thực tế có lưu lượng lành tính nhiều hơn so với lưu lượng độc hại nên để sát với thực tế chúng em giới hạn kích thước tối đa của mẫu độc hại so với mẫu lành tính khi train.

Family	Sessions	Tỉ lệ
Normal	171864	34.5%
Dridex	92233	18.5%
Dreambot	298	0.06%
Gootkit	408	0.08%
Hancitor	474	0.1%
Miuref	4626	0.9%
Trickbot	226436	45.4%
Vawtrak	594	0.12%
Zeus-panda	16	0.04%
zeus	964	0.3%
Tổng	498033	100%

Bảng 2 Danh sách dataset thu thập

3.3. Triển khai thuật toán IARF

Với thuật toán IARF, nhóm chúng em có tìm hiểu và phân tích function code của các thư viện river, sklearn của python. Các funtion code của AdaptiveRandomForestClassifier của thư viện river hỗ trợ khá giống với thuật toán ARF và IARF sau khi xét mẫu (Learn_one) được đề cập trong bài báo. Trong sklearn và 1 số code tìm hiểu, thấy được cách phân tích và cách dự đoán mẫu để train. Từ đó, nhóm tự triển khai thuật toán dựa trên những hiểu biết và kiến thức học được từ quá trình tìm hiểu ở trên.

3.4. Triển khai demo

3.4.1. Tài nguyên:

Môi trường phát triển:

- Thực hiện trên: Google Colab
- Ngôn ngữ lập trình: python3
- Thư viện sử dụng: numpy, pandas, mathplotlib, sklearn, river…

3.4.2. Thu thập dataset và tiền xử lý

Gồm 3 phần:

- Trích xuất nhật ký
- Kết nối nhật ký
- Clean dataset: loại bỏ các cột không cần thiết, xác định các chứng chỉ và mã hóa nhãn.

Link colab:
https://colab.research.google.com/drive/1bsRL1iJwCfJjjQYKpEkaLVxgHz7ymgnL

3.4.3. Training và so sánh các model

Link colab:
https://colab.research.google.com/drive/1YV1W1xJj2IHo-REbPrrIIG_6n-KTS000

3.4.4 Đánh giá kết quả so với bài báo:

	Precision	Recall	F1-score	Training Time	Learning Sample
IARF	99.68%	99.67%	99.66%	425.05s	3179
ARF	98.78%	98.95%	98.73%	2395.57s	497920
LRCV	84.78%	92.07%	88.26%	2688.92s	497920
LSVC	96.83%	97.46%	96.83%	1390.87s	497920
RF	99.96%	99.96%	99.96%	6.80s	497920
GBDT	99.36%	99.18%	99.24%%	1488.68s	497920

Bảng 3 Kết quả training của bài báo

	Precision	Recall	F1-score	Training Time	Learning Sample
IARF	99.97%	99.97%	99.97%	167.75s	35
ARF	99.87%	99.87%	99.87%	450.36s	137680
LRCV	75.40%	75.40%	75.40%	359.93s	96376
LSVC	89.44%	89.44%	89.44%	221.58s	96376
RF	99.93%	99.93%	99.93%	13.95s	96376
GBDT	99.93%	99.93%	99.93%	345.72s	96376

Bảng 4 Kết quả training của nhóm

	IARF			ARF			
Family	Precision	Recall	F1-score	Precision	Recall	F1-score	Support
Normal	99.85%	99.89%	99.87%	98.99%	99.99%	99.49%	50940
Dridex	99.24%	99.51%	99.38%	98.55%	99.28%	98.90%	1000
Dreambot	98.22%	94.61%	96.35%	59.13%	13.05%	17.88%	48
Gootkit	97.53%	88.72%	92.69%	58.75%	25.67%	31.81%	41
Hancitor	89.48%	94.82%	91.88%	72.19%	62.82%	66.53%	219
Miuref	94.19%	95.25%	94.66%	99.66%	70.83%	82.10%	718

Trickbot	99.52%	98.81%	99.16%	99.69%	96.63%	98.13%	1000
Vawtrak	99.97%	99.73%	99.85%	98.14%	99.98%	99.04%	1000
Zeus-panda	91.48%	73.64%	80.51%	78.80%	34.87%	44.31%	119
zeus	95.76%	90.51%	92.51%	99.92%	59.22%	71.42%	239

Bảng 5 Kết quả của bài báo so sánh IARF với ARF

	IARF			ARF			
Family	Precision	Recall	F1-score	Precision	Recall	F1-score	Support
Normal	100%	100%	100%	100%	100%	100%	100000
Dridex	100%	100%	100%	100%	100%	100%	15000
Dreambot	98%	98%	98%	99%	95%	97%	298
Gootkit	98%	99%	99%	98%	100%	99%	408
Hancitor	100%	100%	100%	99%	100%	99%	474
Miuref	100%	100%	100%	99%	100%	100%	4926
Trickbot	100%	100%	100%	100%	100%	100%	15000
Vawtrak	99%	99%	99%	98%	94%	96%	594
Zeus-panda	72%	81%	76%	100%	12%	22%	16
zeus	100%	100%	100%	98%	95%	96%	239

Bảng 6 Kết quả của nhóm khi so sánh IARF với ARF

- Kết quả của nhóm thực hiện gần như tương đồng vơi độ hiệu quả mà tác giả đề cập.

CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Ở chương này, chúng tôi đưa ra những kết luận về nghiên cứu, những hạn chế, và đồng thời đưa ra hướng cải thiện và phát triển.

4.1. Kết luận

Nghiên cứu này nhằm mục đích xác định phần mềm độc hại trong việc truyền dữ liệu lưu lượng truy cập được mã hóa, trôi dạt và mất cân bằng. Nghiên cứu góp phần nâng cao hiểu biết của chúng em về việc phát hiện lưu lượng độc hại trong môi trường phức tạp, gần với thực tế hơn. Dựa trên trên ARF, một thuật toán học trực tuyến, IARF được đề xuất để phát hiện phần mềm độc hại trong luồng lưu lượng được mã hóa. IARF có thể cập nhật thích ứng khi có mẫu mới, điều quan trọng là phải theo dõi kịp thời các loại phần mềm độc hại mới. Ngoài ra, IARF vẫn nhạy cảm cho các dòng phần mềm độc hại có ít mẫu, điều này rất quan trọng để phát hiện các cuộc tấn công mạng bí mật nhưng có mối đe dọa nghiêm trọng. Hơn nữa, bài báo đề xuất phương pháp phát hiện lưu lượng truy cập độc hại được mã hóa mới dựa trên IARF để phát hiện chính xác các dòng phần mềm độc hại trong lưu lượng được mã hóa, trôi dạt và mất cân bằng. Bài báo triển khai nguyên mẫu của phương pháp và tiến hành thử nghiệm để đánh giá Tiếp cận. Kết quả thử nghiệm cho thấy phương pháp của bài báo hoạt động tốt hơn đáng kể so với ARF ban đầu về mặt về độ chính xác và hiệu quả trong trường hợp lưu lượng mất cân bằng, trôi dạt và mã hóa. Ngoài ra, so với ba đại diện phương pháp, cách tiếp cận của bài báo có hiệu suất tốt hơn. Một hạn chế trong nghiên cứu của bài báo là việc trích xuất tính năng chỉ hoạt động trên một phiên hoàn chỉnh, có nghĩa là phiên đó đã được qua và tải trọng bao gồm cả tải trọng độc hại đã được chuyển đi. Tuy nhiên, hiệu suất tốt nhất của sự bất thường Hệ thống phát hiện là phát hiện và cảnh báo các phiên mã hóa độc hại trước khi tải trọng độc hại được chuyển đi, để hệ thống có thể ngăn chặn sự tương tác tiếp theo của phiên một cách kịp thời

4.2. Hướng phát triển

Bên cạnh những kết quả mà chúng em đã đạt được trong đồ án môn học này thì vẫn còn nhiều vấn đề có thể tiếp tục cải tiến, phát triển trong tương lai cho bài báo.

- Trích xuất tính năng có thể thực hiện trên một phiên không hoàn chỉnh để có thể kịp thời ngăn sự tương tác của các phiên kịp thời -> Trích xuất các tính năng chỉ từ phần bắt tay của phiên thay vì toàn bộ phiên
- Phát triển kiến trúc trong nhiều ngữ cảnh khác nhau

TÀI LIỆU THAM KHẢO

Tiếng Anh:

[1] Niu, Z., Xue, J., Qu, D., Wang, Y., Zheng, J., & Zhu, H. (2022). A novel approach based on adaptive online analysis of encrypted traffic for identifying Malware in IIoT. Information Sciences, 601, 162-174. Chicago.

[2] Gomes, H. M., Bifet, A., Read, J., Barddal, J. P., Enembreck, F., Pfharinger, B., ... & Abdessalem, T. (2017). Adaptive random forests for evolving data stream classification. Machine Learning, 106, 1469-1495.