BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC ĐÀ NẴNG

TRẦN THANH LIÊM

ỨNG DỤNG MÔ HÌNH MAPREDUCE PHÂN TÍCH VÀ XỬ LÝ MALWARE

Chuyên ngành: Khoa học máy tính
Mã số: 60.48.01

TÓM TẮT LUẬN VĂN THẠC SĨ KỸ THUẬT

Đà Nẵng - Năm 2015

Công trình được hoàn thành tại

ĐẠI HỌC ĐÀ NẴNG

Người hướng dẫn khoa học: TS. NGUYỄN TẤN KHÔI

Phản biện 1: TS. NGUYỄN VĂN HIỆU

Phản biện 2: TS. HOÀNG THỊ LAN GIAO

Luận văn được bảo vệ trước Hội đồng chấm Luận văn tốt nghiệp thạc sĩ Kỹ thuật họp tại Đại học Đà Nẵng vào ngày 10 tháng 01 năm 2015.

Có thể tìm hiểu luận văn tại:

- Trung tâm Thông tin - Học liệu, Đại học Đà Nẵng

MỞ ĐẦU

1. Tính cấp thiết của đề tài

Sự phát triển mạnh mẽ của công nghệ thông tin trên lĩnh vực mạng nói chung và Internet nói riêng đã giúp cho việc trao đổi thông tin nhanh chóng, dễ dàng. Dịch vụ trao đổi thông tin cho phép người ta nhận hay gửi dữ liệu ngay trên máy tính của mình; thương mại điện tử cho phép thực hiện các giao dịch trên mạng. Do vậy, thông tin có thể bị đánh cắp, làm sai lệch hoặc giả mạo ngày càng nhiều hơn. 64,2 triệu lượt máy tính tại Việt Nam bị nhiễm virus năm 2011, theo thống kê của Hệ thống giám sát virus BKAV. Trung bình một ngày đã có hơn 175 nghìn máy tính bị nhiễm virus. Năm 2012, Việt Nam bị xếp thứ 15 về phát tán mã độc, thứ 10 về tin rác, thứ 15 về zombie. Năm 2013, mỗi tháng có khoảng 300 website bị tấn công làm thay đổi giao diện, đăng các thông tin sai lệch, phá hoại hoạt động của website. Điều này làm ảnh hưởng lớn tới các tổ chức, các công ty hay cả một quốc gia [17].

Hiện nay, với sự phát triển của công nghệ, lượng dữ liệu lưu trữ càng ngày càng lớn. Điều này đặt ra nhiều thách thức. Để xử lý lượng dữ liệu khổng lồ đó, rất nhiều công nghệ đã ra đời. Trong đó phải kể đến công nghệ tính toán phân tán. Ý tưởng chính của việc tính toán phân tán là chia bài toán thành những bài toán con và giải quyết trên các máy riêng biệt nhau được kết nối trong một cluster. Chúng ta có thể thấy sự thành công của các công ty như Google, Facebook trong thời đại bùng nổ về công nghệ hiện nay. Đằng sau sự thành công đó có sự đóng góp không nhỏ của một mô hình lập trình được đưa ra bởi Google – đó là mô hình lập trình phân tán MapReduce [18].

MapReduce là một mô hình lập trình phân tán, bao gồm hai

giai đoạn chính là Map và Reduce. Mô hình lập trình MapReduce được dùng để xử lý dữ liệu lớn dựa trên lý thuyết của mô hình tính toán song song và mô hình xử lý dữ liệu phân tán trên những cụm máy tính. Ưu điểm của MapReduce là:

- Xử lý tốt bài toán với lượng dữ liệu lớn có các tác vụ phân tích và tính toán phức tạp không lường trước được.

- Có thể tiến hành chạy song song trên các máy phân tán một cách chính xác và hiệu quả. Không cần quan tâm đến sự trao đổi dữ liệu giữa các cluster với nhau vì chúng hoạt động một cách độc lập, không cần theo dõi xử lý các tác vụ, xử lý lỗi.

- Có thể thực hiện mô hình MapReduce trên nhiều ngôn ngữ (Java, C/ C++, Python, Perl, Ruby) với các thư viện tương ứng.

Với nhu cầu xử lý dữ liệu lớn, các mã độc trên mạng ngày càng nhiều, quá trình phát hiện và xử lý mã độc rất phức tạp, mã độc cần thời gian xử lý ngắn hơn, do đó hướng ứng dụng MapReduce để phát hiện mã độc đang được quan tâm hiện nay.

Vì những lý do như trên, tôi đã chọn đề tài luận văn cao học:

"Ứng dụng mô hình MapReduce phân tích và xử lý malware"

2. Mục tiêu của đề tài

Mục tiêu chính của đề tài là nghiên cứu ứng dụng mô hình MapReduce để thiết kế xây dựng hệ thống phân tích và xử lý mã độc.

Để thực hiện được yêu cầu trên, luận văn tập trung thực hiện những nhiệm vụ cụ thể như sau:

- Tìm hiểu các nguyên tắc an toàn và bảo mật thông tin

- Tìm hiểu hệ thống hỗ trợ phát hiện xâm nhập

- Tìm hiểu phương pháp lập trình phân tán MapReduce, phân tích và xử lý các mã độc

- Nghiên cứu ứng dụng mô hình xử lý phân tán MapReduce để xây dựng hệ thống phân tích và xử lý mã độc

- Triển khai thử nghiệm trên hệ thống mạng tại Đại học Đà Nẵng

- Nhận xét đánh giá và so sánh với các hệ thống khác

3. Đối tượng và phạm vi nghiên cứu

3.1. Đối tượng nghiên cứu

- Cơ chế an toàn thông tin mạng
- Hệ thống phát hiện xâm nhập
- Mô hình lập trình phân tán MapReduce
- Cơ chế phân tích và xử lý các mã độc

3.2. Phạm vi nghiên cứu

- Các nguy cơ xâm nhập hệ thống
- Nghiên cứu các thư viện hỗ trợ phát hiện virus có sẵn
- Môi trường lập trình phân tán theo mô hình MapReduce
- Áp dụng vào hệ thống mạng tại Đại học Đà Nẵng

4. Phương pháp nghiên cứu

4.1. Phương pháp lý thuyết

- Cơ chế an toàn và bảo mật thông tin mạng
- Cơ chế phát hiện xâm nhập của mã độc
- Nguyên lý lập trình theo mô hình xử lý phân tán MapReduce
- Phân tích và xử lý các mã độc dựa vào mô hình MapReduce

4.2. Phương pháp thực nghiệm

- Phân tích, thiết kế các chức năng phân tích và xử lý mã độc
- Xây dựng các mô-đun xử lý phân tán với mã độc
- Xử lý, hiển thị và đánh giá kết quả
- Kiểm thử, nhận xét và đánh giá kết quả của hệ thống

5. Ý nghĩa khoa học và thực tiễn

5.1. Ý nghĩa khoa học

- Cơ chế phân tích xử lý và phát hiện mã độc

- Áp dụng phương pháp lập trình phân tán theo mô hình MapReduce để xử lý bài toán phân tích và xử lý mã độc theo thời gian thực

5.2. Ý nghĩa thực tiễn

- Đề xuất giải pháp phát hiện và xử lý mã độc cho các tổ chức và doanh nghiệp

- Cung cấp một hệ thống hỗ trợ cho các nhà quản trị mạng khai thác và ứng dụng phục vụ công việc của cơ quan, doanh nghiệp

6. Bố cục luận văn

Bố cục của luận văn gồm các phần chính như sau:

Mở đầu

Chương 1. Tổng quan đề tài

Chương 2. Mô hình xử lý phân tán MapReduce

Chương 3. Xây dựng hệ thống phân tích và xử lý mã độc

Kết luận và hướng phát triển

CHƯƠNG 1
TỔNG QUAN ĐỀ TÀI

1.1. AN TOÀN THÔNG TIN

1.1.1. Giới thiệu

1.1.2. Các yêu cầu an toàn bảo mật thông tin

1.1.3. Các tiêu chuẩn đảm bảo an toàn thông tin

1.1.4. Tình hình an ninh mạng trong nước

1.1.5. Tình hình an ninh mạng quốc tế

1.2. CÁC HÌNH THỨC TẤN CÔNG MẠNG

1.2.1. Tấn công thăm dò

1.2.2. Tấn công sử dụng mã độc

1.2.3. Tấn công xâm nhập mạng

1.2.4. Tấn công từ chối dịch vụ

1.3. TẤN CÔNG MÃ ĐỘC

1.3.1. Khái niệm mã độc

1.3.2. Phân loại mã độc

1.3.3. Hành vi của mã độc

1.3.4. Các phương pháp phân tích mã độc

a. Phương pháp phân tích mã độc thủ công

- *Phân tích sơ lược*
- *Phân tích hoạt động*
- *Phân tích bằng cách đọc mã thực thi của mã độc*

b. Phương pháp phân tích mã độc tự động

1.4. CÁC CÔNG TRÌNH NGHIÊN CỨU LIÊN QUAN

Trong hầu hết các hệ thống sandbox miễn phí được cung cấp trên mạng như: Joe Sandbox, Threat expert, CW Sandbox chỉ hỗ trợ cơ chế cho phép người dùng nhập cùng lúc một mã độc lên cho hệ thống phân tích. Ngay cả với trường hợp người dùng đưa lên một tập

tin nén *.zip chứa các tập tin mã độc nhưng các hệ thống trên đều tách riêng ra để phân tích từng tập tin.

Bên cạnh đó những hệ thống sandbox cho phép phân tích hành vi mã độc tự động miễn phí như Cuckoo Sandbox, Buster Sandbox hay Zero Wine Sandbox đều có những hạn chế riêng.

Buster Sandbox phụ thuộc vào việc tải và cài đặt Sandboxie, đây là một phần mềm mã đóng và có khuynh hướng thu phí người dùng. Bên cạnh đó việc tùy chỉnh các kịch bản bên trong Buster Sandbox lại không được hỗ trợ nhiều, do vậy người dùng thường chỉ chấp nhận những gì mà Buster Sandbox cung cấp [32].

Zero Wine Sandbox cũng gặp phải một số vấn đề khi phân tích các mã độc đó là khả năng mã độc phát hiện môi trường phân tích của Zero Wine Sandbox rất cao, thông qua việc đọc giá trị registry hoặc thông qua việc so sánh dung lượng tập tin Windows thật và tập tin Windows trong Zero Wine Sandbox, do các tập tin trong Zero Wine Sandbox thường có dung lượng nhỏ. Và phiên bản mới nhất của Zero Wine Sandbox được cung cấp vào cuối năm 2010 nên không theo kịp với sự phát triển của các loại mã độc [33].

Cuckoo Sandbox được bắt đầu vào năm 2010, là một dự án trong Google Summer of Code được thiết kế và phát triển bởi Claudio nex Guarnieri. Cuckoo có thể tận dụng sự sáng tạo và đóng góp của toàn thể cộng đồng để tạo nên một hệ thống phân tích mã độc ổn định và hiệu quả hơn. Tuy nhiên hệ thống Cukoo vẫn chạy một cách tuần tự phân tích từng mã độc [34].

Mô hình xử lý phân tán MapReduce được ứng dụng trong các bài toán dữ liệu rất lớn (big data), phức tạp mà với một máy gần như là không thể xử lý được. Lập trình phân tán theo mô hình này giúp giảm đáng kể thời gian xử lý công việc. Dưới đây là một số nghiên

cứu và ứng dụng theo mô hình lập trình phân tán MapReduce như: Bài toán đếm từ xuất hiện [13], Khai phá luật kết hợp với dữ liệu phân tán dựa trên mô hình MapReduce (Luận văn thạc sĩ) [3], Theo dõi các trận động đất trên thế giới [18], Áp dụng Search Engine phân tán [2], Tính độ tương tự của tài liệu theo mô hình MapReduce,…

1.5. KẾT CHƯƠNG

Trong chương này đã giới thiệu tổng quan về an toàn thông tin. Khái quát một số khái niệm cơ bản về an toàn thông tin, các hình thức tấn công mạng, tấn công mã độc. Nêu ra các yêu cầu an toàn bảo mật thông tin, các tiêu chuẩn đảm bảo an toàn thông tin, tình hình an ninh mạng trong nước và quốc tế. Ngoài ra, trong chương này, đã trình bày cụ thể các hình thức tấn công mạng và tấn công mã độc, các tác hại của mã độc, phân loại mã độc, các phương pháp và công cụ hỗ trợ phân tích mã độc. Từ đó hình thành ý tưởng ứng dụng mô hình xử lý phân tán MapReduce vào phân tích và xử lý mã độc.

CHƯƠNG 2
MÔ HÌNH XỬ LÝ PHÂN TÁN MAPREDUCE

2.1. ĐIỆN TOÁN ĐÁM MÂY

2.1.1. Đặc điểm dịch vụ điện toán đám mây

2.1.2. Lợi ích của điện toán đám mây

2.1.3. Tính chất cơ bản của điện toán đám mây

a. Tự phục vụ theo nhu cầu (On-demand self-service)

b. Truy xuất diện rộng (Broad network access)

c. Dùng chung tài nguyên (Resource pooling)

d. Khả năng co giãn (Rapid elasticity)

e. Điều tiết dịch vụ (Measured service)

2.1.4. Các mô hình điện toán đám mây

a. Mô hình dịch vụ

- Infrastructure as a Service – IaaS
- Platform as a Service – PaaS
- Software as a Service – SaaS

b. Mô hình triển khai

- Public Cloud
- Private Cloud
- Hybrid Cloud

2.2. MÔ HÌNH XỬ LÝ PHÂN TÁN MAPREDUCE

2.2.1. Giới thiệu

MapReduce là mô hình xử lý phân tán giúp các ứng dụng có thể xử lý nhanh một lượng dữ liệu lớn. Các dữ liệu này được đặt tại các máy tính phân tán. Các máy tính này sẽ hoạt động song song độc lập với nhau. Điều này giúp rút ngắn thời gian xử lý toàn bộ dữ liệu.

Ưu điểm của MapReduce:

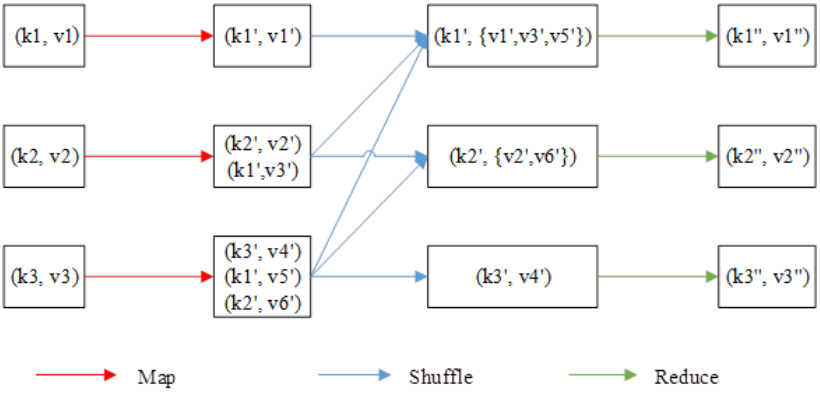
- Xử lý tốt bài toán về lượng dữ liệu lớn có các tác vụ phân

tích và tính toán phức tạp không lường trước được.

- Có thể tiến hành chạy song song trên các máy phân tán một cách chính xác và hiệu quả. Không phải quan tâm đến sự trao đổi dữ liệu giữa các cluster với nhau vì chúng hoạt động một cách độc lập, không phải theo dõi xử lý các tác vụ, xử lý lỗi.
- Có thể thực hiện mô hình MapReduce trên nhiều ngôn ngữ (Java, C/ C++, Python, Perl, Ruby) với các thư viện tương ứng [9].

2.2.2. Nguyên tắc hoạt động của MapReduce

MapReduce gồm hai quá trình thực hiện hai hàm Map và Reduce. Ý tưởng chính của MapReduce chính là thực hiện việc "Chia để trị" [13].



Hình 2.3. Sơ đồ hoạt động của quá trình MapReduce

2.2.3. Các hàm Map và Reduce

2.3. NỀN TẢNG HADOOP

2.3.1. Giới thiệu

Năm 2008, Hadoop đã phá kỷ lục thế giới về sắp xếp một terabyte dữ liệu. Chạy trên một cluster gồm 910 nút, Hadoop đã sắp

xếp một terabyte dữ liệu trong vòng 209 giây, phá kỷ lục cũ là 297 giây. Sau đó ít lâu, Google công bố ứng dụng chạy trên MapReduce của họ đã sắp xếp được một terabyte dữ liệu trong 68 giây. Vào tháng 5 năm 2009, một đội các nhà phát triển của Yahoo! đã dùng Hadoop để sắp xếp một terabyte dữ liệu trong vòng 62 giây.

a. Các thành phần của Hadoop

b. Tổng quan của một cụm Hadoop

c. Ứng dụng của Hadoop

2.3.2. Hệ thống tập tin phân tán Hadoop

a. Tổng quan thiết kế của HDFS

b. Các tính năng của NameNode

c. Khả năng chịu lỗi và chẩn đoán lỗi của HDFS

d. Các giao diện tương tác

e. Quản trị HDFS

2.3.3. Phát triển ứng dụng MapReduce trên nền tảng Hadoop

2.4. KẾT CHƯƠNG

Trong chương này đã trình bày về điện toán đám mây, mô hình lập trình MapReduce, chỉ ra được các ưu nhược điểm, nguyên tắc hoạt động, thực thi MapReduce. Ngoài ra, nội dung chương này đã giới thiệu về nền tảng Hadoop, trình bày tổng quan thiết kế của HDFS, các tính năng của NameNode, khả năng chịu lỗi và chẩn đoán lỗi của HDFS, các giao diện tương tác và quản trị HDFS. Từ đó làm nền tảng để xây dựng mô hình và quy trình thực hiện phân tích và xử lý mã độc dựa trên nền tảng Hadoop.

CHƯƠNG 3
THIẾT KẾ HỆ THỐNG PHÂN TÍCH VÀ XỬ LÝ MÃ ĐỘC

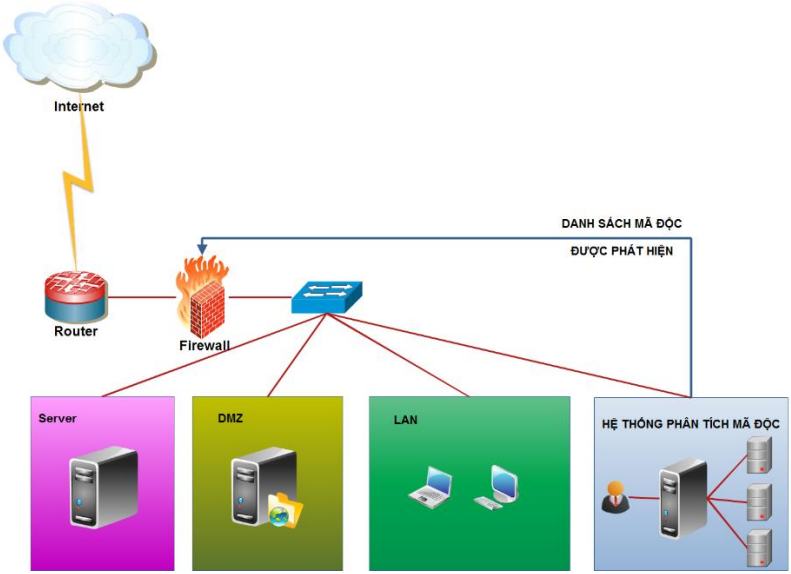
3.1. PHÁT BIỂU BÀI TOÁN

Với nhu cầu xử lý dữ liệu lớn, các mã độc trên mạng ngày càng nhiều, quá trình phát hiện và xử lý mã độc rất phức tạp, mã độc cần thời gian xử lý ngắn hơn, do đó hướng ứng dụng mô hình xử lý phân tán MapReduce để phân tích và xử lý mã độc đang được quan tâm hiện nay.

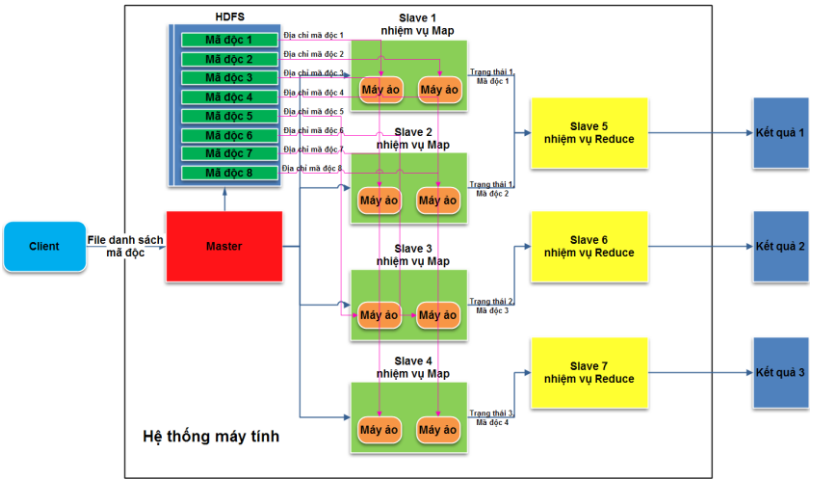
Yêu cầu của hệ thống phân tích và xử lý mã độc: *Tính phân tán, Tính an toàn, Tính tự động, Tính hiệu năng*

3.2. PHÂN TÍCH MÃ ĐỘC DỰA TRÊN MÔ HÌNH MAPREDUCE

3.2.1. Mô hình phân tích mã độc



Hình 3.1. Mô hình tổng quan hệ thống



Hình 3.2. Mô hình hệ thống phân tích mã độc

Hệ thống phân tích mã độc bao gồm một máy master (máy chủ) và nhiều máy slave (máy trạm). Hệ thống tập tin phân tán HDFS sẽ lưu trữ các mã độc cần phân tích. Với hệ thống này cho phép ta có thể chọn nhiều kiểu tập tin khác nhau và có thể đặt phân tán để phân tích. Máy client sẽ gởi danh sách tập tin các mã độc đến và yêu cầu thực hiện phân tích. Máy master sẽ xem xét những máy slave nào sẵn sàng và phân phối, gởi địa chỉ mã độc đến để làm nhiệm vụ Map. Các máy slave làm nhiệm vụ Map sẽ tải mã độc từ HDFS và tiến hành phân tích. Kết quả của quá trình Map sẽ được gởi đến các máy slave để làm nhiệm vụ Reduce. Kết quả phân tích cũng chính là kết quả của quá trình Reduce. Như vậy, ở các máy slave vừa làm nhiệm vụ Map, vừa làm nhiệm vụ Reduce.

Ở các máy slave đều được cài thêm một hoặc nhiều máy ảo. Tùy vào cấu hình máy chủ, việc cài đặt nhiều máy ảo sẽ giúp giảm thời gian phân tích, tăng hiệu quả xử lý. Mục đích của việc cài đặt

máy ảo là tạo ra môi trường an toàn để thực thi mã độc sau khi mã độc được tải về từ HDFS. Các máy ảo này được lập trình để có thể chạy tự động (tự khởi động, tự động khôi phục lại môi trường sạch, tự sao chép tập tin về phân tích, trả kết quả cho máy slave, tự động tắt máy ảo) mà không cần sự can thiệp của con người. Với chức năng Snapshot, sẽ giúp cho việc khôi phục lại môi trường, cấu hình phân tích trong máy ảo trở nên nhanh chóng hoặc có thể chọn lựa các môi trường phân tích khác nhau để phù hợp với các tập tin phân tích mã độc.

3.2.2. Quy trình thực hiện

a. Đầu vào

- Danh sách tập tin nghi ngờ có mã độc
- Địa chỉ mã độc

b. Đầu ra

- Kết luận tập tin có nhiễm mã độc hay không
- Kết quả báo cáo chi tiết về các hành vi của mã độc
- Thống kê các mã độc được phân tích
- Kết quả xử lý, ngăn chặn, gỡ bỏ mã độc

c. Sơ đồ thuật toán

d. Các bước thực hiện

3.2.3. Cơ chế Map mã độc

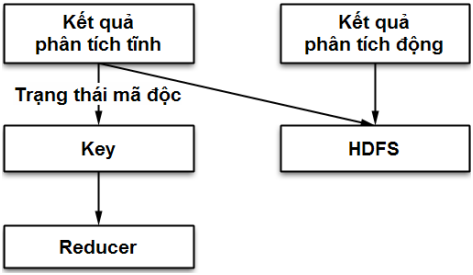
Các máy slave làm nhiệm vụ Map sẽ nhận đầu vào là một cặp <key, value> với key là tên các mã độc, value là địa chỉ của mã độc. Dựa vào địa chỉ này, các máy cục bộ sẽ tải các mã độc về phân tích.

Như vậy, quá trình hoạt động của máy slave làm nhiệm vụ Map có ba giai đoạn chính:

- Tải mã độc về máy slave để làm nhiệm vụ Map từ HDFS

- Thực hiện chạy hoạt động phân tích tĩnh
- Chép mã độc vào máy ảo, thực hiện chạy hoạt động phân tích động

Công việc xử lý kết quả phân tích được mô tả như sau:



Hình 3.5. Xử lý kết quả phân tích

Kết quả của quá trình phân tích được xuất ra tập tin. Kết quả phân tích sẽ được phân chia, một phần là đầu ra cho quá trình Map, một phần được lưu xuống HDFS để phục vụ cho việc thống kê.

Việc phân tích tĩnh gồm có các thông tin cơ bản như: tên mã độc, giá trị MD5, trạng thái mã độc, khả năng phát hiện mã độc này của các Antivirus. Trạng thái mã độc là NOT OK nếu như mã độc đó được phát hiện, OK nếu mã độc đó không được phát hiện, N/A nếu như có lỗi xảy ra trong quá trình phân tích. Trạng thái mã độc chính là đầu ra của quá trình Map và sẽ là đầu vào của quá trình Reduce.

Đoạn mã lệnh chương trình minh họa cơ chế Map mã độc:

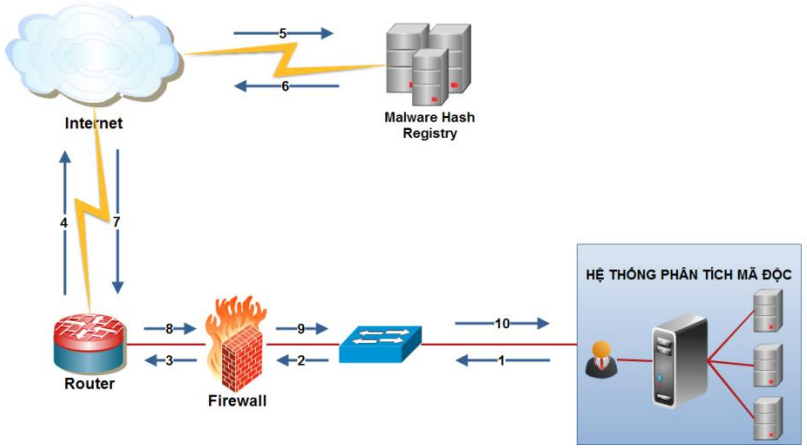
3.2.4. Cơ chế Reduce mã độc

Sau khi thực hiện xong nhiệm vụ Map, các máy slave sẽ thực hiện nhiệm vụ Reduce. Đầu vào của các máy Reduce sẽ là cặp các <key, value>, với key là trạng thái mã độc (NOT OK, OK, N/A) và value là tên của các mã độc. Các máy Reduce sẽ nhóm các mã độc có cùng trạng thái thành từng nhóm.

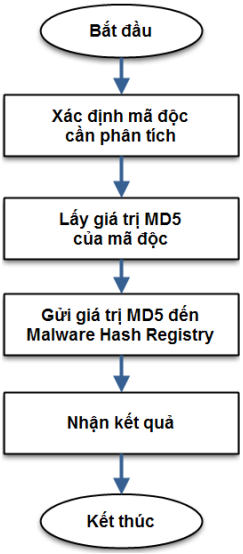
3.3. THIẾT KẾ XÂY DỰNG HỆ THỐNG

3.3.1. Phân rã chức năng

3.3.2. Chức năng phân tích tĩnh



Hình 3.7. Tổng quan hoạt động chức năng phân tích tĩnh



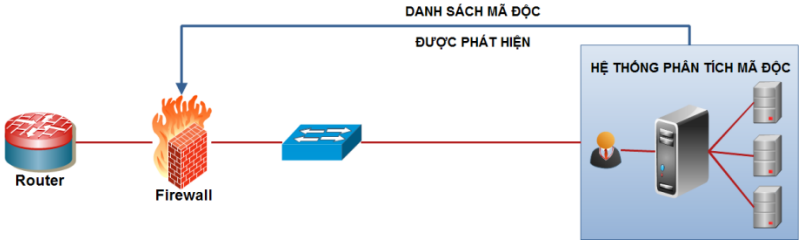
Hình 3.8. Sơ đồ hoạt động phân tích tĩnh

```
#!/bin/bash

MALWARE=$1

MD5=`md5sum ${MALWARE} | awk '{print $1}'`

whois -h hash.cymru.com ${MD5} > ${MALWARE}.static
```

3.3.3. Chức năng phân tích động

Hình 3.9. Sơ đồ hoạt động phân tích động

3.3.4. Chức năng ngăn chặn, gỡ bỏ



Hình 3.10. Chức năng ngăn chặn, gỡ bỏ

3.4. TRIỂN KHAI THỬ NGHIỆM

3.4.1. Môi trường triển khai

3.4.2. Công cụ sử dụng

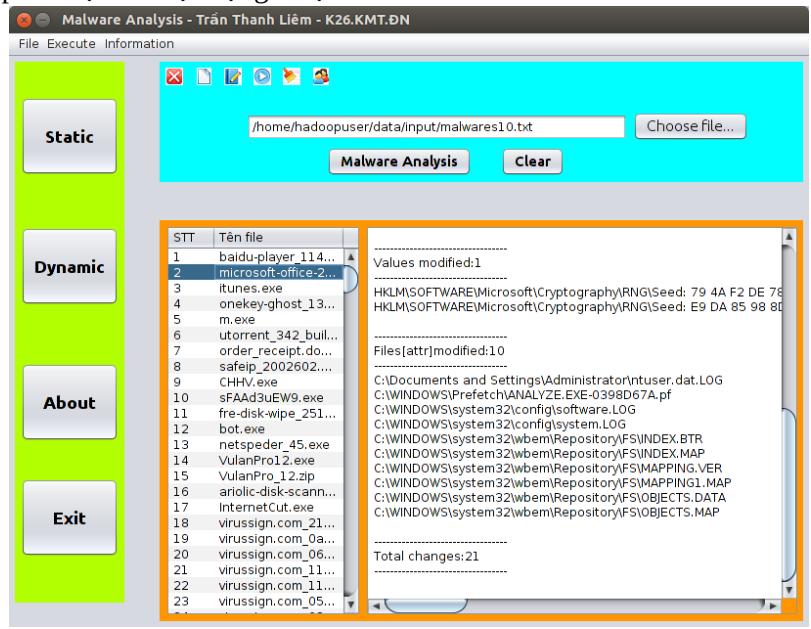
3.4.3. Kết quả thử nghiệm

Kết quả phân tích tĩnh sẽ hiển thị: STT, tên mã độc, giá trị MD5, tình trạng mã độc (NOT OK, OK, N/A), số lượng Antivirus phát hiện và thống kê số lượng mã độc đã phân tích, số lượng mã độc bị phát hiện, số lượng mã độc không bị phát hiện, số lượng lỗi trong quá trình phân tích.

Bảng 3.2. Thời gian phân tích tĩnh các mã độc trên một máy

STT	*Số lượng mã độc*	*Thời gian phân tích*	*Phát hiện*	*Tỉ lệ %*
1	50	60 giây	44	88.00%
2	100	113 giây	89	89.00%
3	150	150 giây	134	89.33%
4	200	185 giây	173	86.50%
5	250	222 giây	219	87.60%
6	300	251 giây	258	86.00%

Kết quả phân tích tĩnh trên 50, 100, 150, 200, 250 và 300 mã độc lần lượt là 44/50, 89/100, 134/150, 173/200, 219/250 và 258/300 với thời gian tương ứng là 60, 113, 150, 185, 222 và 251 giây, tỉ lệ phát hiện mã độc đạt giá trị từ 86.00% đến 89.33%.



Hình 3.19. Kết quả phân tích động

Tập tin *[thuthuat.chiplove.biz]---Keygen-IDM-6xx.exe* bị nhiễm mã độc đã làm thay đổi hệ thống. Hành vi của mã độc này đã được hệ thống ghi lại: số khóa bị xóa đi trong registry là 6, số lượng thêm vào là 2, số lượng chỉnh sửa là 11. Tổng số thay đổi trong registry là 19.

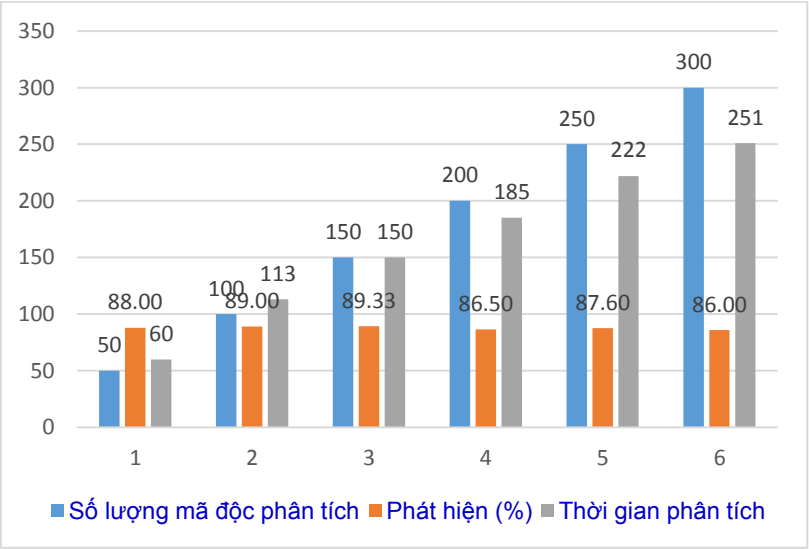
3.5. NHẬN XÉT ĐÁNH GIÁ

Chương trình ứng dụng được xây dựng trên cơ sở lý thuyết và mô hình đề xuất sắp tới sẽ thử nghiệm tại Đại học Đà Nẵng. Bước đầu, hệ thống phân tích và xử lý mã độc sẽ đem lại những thuận tiện trong việc đảm bảo an toàn thông tin mạng. Việc tận dụng năng lực của các máy chủ trong thời gian rỗi bằng cách đặt lịch thực hiện góp phần nâng cao năng suất trong sáng kiến, cải tiến kỹ thuật, rút ngắn được đáng kể thời gian phân tích mã độc.

Hệ thống phân tích và xử lý mã độc được ứng dụng từ mô hình MapReduce này được hoạt động một cách tự động. Tự động ở đây có nghĩa là các công đoạn phân tích đều do hệ thống tự động thực hiện, từ công đoạn nhận mã độc cho đến sao chép mã độc vào máy ảo, thực thi mã độc và cuối cùng là đưa ra bản báo cáo chi tiết về hành vi mã độc mà không cần con người tác động vào.

Qua các kết quả thực nghiệm ở trên cho thấy:

Tỉ lệ mã độc được phát hiện trên các bộ thử lần lượt là 44/50, 89/100, 134/150, 173/200, 219/250 và 258/300, tỉ lệ phát hiện đạt từ 86% trở lên. Kết quả của hệ thống phân tích tĩnh này phụ thuộc vào dịch vụ bên ngoài sử dụng, dịch vụ Malware Hash Registry của Team Cymru. Ngoài ra, ta cũng có thể sử dụng các dịch vụ khác như: Virus Total [20].



Hình 3.21. Biểu đồ thời gian phân tích tĩnh các mã độc trên một máy

Bảng 3.2 thể hiện thời gian phân tích tĩnh các mã độc trên một máy. Ta thấy rằng số lượng mã độc được phân tích càng tăng lên thì thời gian phân tích trên mỗi mã độc sẽ giảm lại. Nếu số lượng mã độc là 50 và thời gian phân tích là 60 giây thì trung bình thời gian phân tích mỗi mã độc là 1,2 giây. Nếu số lượng là 300 và thời gian phân tích là 251 giây thì thời gian phân tích trung bình chỉ còn lại là 0,84 giây (giảm 30% trên mỗi mã độc). Hình 3.21 minh họa rõ vấn đề này.

Nếu triển khai được trên các máy chủ IBM x3650 M4 (02 x Xeon 8C E5-2640v2 95W 2.0GHz, 32GB RAM, 02 x 300GB) tại Đại học Đà Nẵng thì:

- Thời gian phân tích trên 2 máy ảo: 150 giây + < 5 giây (các giai đoạn khác đã được kiểm chứng thực nghiệm) nên kết quả sẽ < 155 giây.

- Mỗi mã độc trung bình mất khoảng 1,2 giây phân tích nên thời gian phân tích 1000 mã độc là: 1,2 * 1000 / 240 + < 5 giây (các giai đoạn khác) nên kết quả sẽ < 10 giây.

Bảng 3.3. Dự tính thời gian phân tích 300 mã độc trên hệ thống

STT	*Số lượng máy chủ*	*Tổng số máy ảo*	*Thời gian (dự kiến)*
1	1	1	251 giây
2	1	2	< 155 giây
3	12	240	< 10 giây

Kết quả phân tích tĩnh và phân tích động được hiển thị rõ ràng, hợp lý. Hệ thống hoạt động ổn định, các chức năng đều cơ bản hoàn thành và cho kết quả chính xác, hệ thống tương đồng với tài liệu số [8].

So với các hệ thống sandbox miễn phí như Joe Sandbox, Threat expert, CW Sandbox, việc ứng dụng mô hình xử lý phân tán MapReduce có thể xử lý hàng loạt, tự động, trong khi các mô hình trên chỉ cho phép nhập một mã độc để phân tích và chưa tự động.

So với những hệ thống xử lý tự động như Buster Sandbox, Cukoo Sandbox hay Zero Wine Sandbox thì hệ thống này giúp cho việc phân tích được thực hiện song song, tăng hiệu năng, giảm thời gian trong việc phân tích số lượng lớn mã độc.

Như vậy, việc ứng dụng mô hình xử lý phân tán này giúp việc phân tích và xử lý mã độc được thực hiện một cách nhanh chóng, cơ sở dữ liệu được cập nhật kịp thời, có thể phân tích hàng loạt các tập tin tùy vào số lượng máy ảo, tính tự động và tính tương thích với hệ thống cao và dễ dàng, linh hoạt trong việc xử lý, khắc phục sự cố.

3.6. KẾT CHƯƠNG

Trong chương này đã phát biểu bài toán, đưa ra mô hình phân tích và xử lý mã độc, sơ đồ phân rã chức năng. Bên cạnh đó đã thiết kế hệ thống phân tích tĩnh, hệ thống phân tích động, hệ thống ngăn chặn, gỡ bỏ. Ngoài ra, trong chương này đã triển khai thử nghiệm trên môi trường máy thật Ubuntu 14.04 với công cụ triển khai như VMware, Eclipse, Plugin cho Hadoop trên Eclipse, AutoIT, Regshot, bộ thư viện JNI và đánh giá kết quả đạt được trên bộ thư viện mã độc thử nghiệm. Việc triển khai hệ thống phân tích và xử lý mã độc dựa trên mô hình xử lý phân tán MapReduce là nhu cầu thực sự cần thiết đối với Đại học Đà Nẵng và các cơ quan, doanh nghiệp. Hệ thống sẽ góp phần đáng kể trong việc phát hiện và ngăn chặn mã độc kịp thời, góp phần nâng cao hiệu quả làm việc.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. KẾT QUẢ ĐẠT ĐƯỢC

Trong thời gian tìm hiểu, nghiên cứu cơ sở lý thuyết và triển khai ứng dụng công nghệ, luận văn đã đạt được những kết quả sau:

Về mặt lý thuyết, luận văn đã đạt được một số kết quả sau:

- Hiểu về an toàn thông tin, các yêu cầu an toàn bảo mật thông tin, các tiêu chuẩn đảm bảo an toàn thông tin, nắm bắt tình hình an ninh mạng trong nước và quốc tế
- Hiểu được các hình thức tấn công hệ thống
- Hiểu được cách tấn công, tác hại và phân loại mã độc, các phương pháp phân tích và các công cụ phân tích mã độc
- Hiểu được mô hình xử lý phân tán MapReduce
- Hiểu được Hadoop và các thành phần của nó
- Tìm hiểu bộ thư viện JNI và công nghệ ảo hóa VMware

Về mặt thực tiễn ứng dụng, luận văn đã đạt được một số kết quả sau:

- Nghiên cứu ứng dụng mô hình xử lý phân tán MapReduce để thiết kế, xây dựng hệ thống phân tích và xử lý mã độc
- Đóng góp ý tưởng, mã nguồn chương trình
- Kiểm tra tính khả thi của hệ thống qua thư viện mẫu mã độc
- Triển khai hệ thống này chạy thử nghiệm tại Đại học Đà Nẵng
- So sánh và làm nổi bật tính hiệu quả của hệ thống này so với các hệ thống trước đây

Tuy nhiên, luận văn còn tồn tại các hạn chế như sau:

- Công cụ phân tích chưa đa dạng nên có thể chưa phát hiện hết hành vi của mã độc
- Quá trình phân tích mã độc bao gồm nhiều công đoạn phức tạp như là phân tích sơ lược, phân tích hoạt động và phân tích bằng cách đọc mã thực thi, vì thế hệ thống phân tích tự động này chỉ hỗ trợ một phần trong các công đoạn nói trên, không thể thay thế hoàn toàn yếu tố con người
- Công đoạn xử lý và phòng chống mã độc còn đơn giản, chưa có cơ chế ngăn chặn và loại bỏ hiệu quả

2. KIẾN NGHỊ VÀ HƯỚNG PHÁT TRIỂN

Kết quả của luận văn đặt ra nhiều vấn đề khoa học cần tiếp tục nghiên cứu giải quyết. Hướng phát triển tiếp theo của đề tài là:

- Bổ sung xây dựng hệ thống thực thi trên nền điện toán đám mây
- Xây dựng các thành phần bổ trợ nhằm thể hiện rõ hơn kết quả phân tích và xử lý, giúp cho nhiều đối tượng có thể sử dụng báo cáo một cách hiệu quả
- Phát triển các thành phần xử lý và phòng chống mã độc tự động dựa trên kết quả phân tích có được để tích hợp vào nhiều hệ thống nhằm hạn chế kịp thời các tác hại do mã độc gây ra