ỨNG DỤNG SANDBOX PHÂN TÍCH MÃ ĐỘC TRÊN MÔI TRƯỜNG PHÂN TÁN

APPLYING SANDBOX TO MALWARE ANALYSIS IN A DISTRIBUTED ENVIRONMENT

Nguyễn Tấn Khôi[1], Trần Thanh Liêm[2]

[1]Trường Đại học Bách khoa, Đại học Đà Nẵng; ntkhoi@dut.udn.vn

[2]Đại học Đà Nẵng; ttliem@ac.udn.vn

Tóm tắt - Hiện nay, mã độc phát sinh ngày càng nhiều và càng tinh vi, khó phát hiện. Việc phân tích theo cách truyền thống là không khả thi, do đó cần có các kỹ thuật hiệu quả để phát hiện và phân tích mã độc. Để phân tích lượng mã độc lớn, ta có thể phát triển một hệ thống phân tích mã độc động sử dụng kỹ thuật sandbox tạo ra môi trường an toàn. Hệ thống này tự động thực thi một chương trình dựa trên môi trường phân tán và cho kết quả báo cáo mô tả các hành vi của chương trình. Bài báo trình bày hướng nghiên cứu và xây dựng hệ thống sandbox trên môi trường phân tán MapReduce nhằm tự động phân tích các hành vi của mã độc. Giải pháp đề xuất cho phép giảm thời gian phân tích và phát hiện chính xác mã độc.

Từ khóa - sandbox; tính toán; song song; mã độc; phân tán; mạng; an toàn; bảo mật

Abstract - Nowadays, the number of malware programs has increased more and more, appearing to be more sophisticated and difficult to detect. The traditional way for analyzing these programs is no longer feasible; therefore, it is necessary to have effective techniques for detecting and analyzing malware. To analyze large quantities of malware, we can develop a dynamic malware analysis system using Sandbox technology, thereby creating a safe environment. This system automatically executes a program based on a distributed environment and produces a report describing the program's behaviours. This paper presents an approach to research and construct a sandbox system in the distributed environment of apReduce for the automatic analysis of malware behaviours. The proposed solution makes it possible to reduce the time for the analysis and to accurately detect malware.

Key words - sandbox; calculation; parallel; malware; distributed; network; safety; security

1. Đặt vấn đề

Theo kết quả thống kê từ Viện nghiên cứu độc lập về an toàn thông tin AV-TEST, kể từ khi mã độc đầu tiên xuất hiện vào năm 1984, cho đến nay đã có khoảng 150.000.000 mã độc được phát tán. Đặc biệt gần đây, số lượng mã độc phát triển nhanh chóng trên toàn thế giới đã đặt ra nhiều vấn đề về an ninh thông tin cho toàn bộ những người sử dụng Internet trên toàn cầu. Năm 2015, Việt Nam nằm trong danh sách các nước có tỉ lệ phát tán mã độc nhiều nhất thế giới.

Các mã độc lây lan ngày càng nhiều, quá trình phát hiện và xử lý mã độc rất phức tạp, do đó hướng ứng dụng hệ thống sandbox và tính toán phân tán để phân tích mã độc đang được quan tâm hiện nay. Sandbox là một kỹ thuật quan trọng trong lĩnh vực bảo mật có tác dụng tạo ra môi trường để các mã độc thể hiện hết các tính năng mà vẫn đảm bảo được tính an toàn cho hệ thống bên ngoài.

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. 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 là một phần mềm mã đóng và việc tùy chỉnh các kịch bản bên trong Buster Sandbox không được hỗ trợ nhiều [7]. Khả năng mã độc phát hiện môi trường phân tích của Zero Wine Sandbox rất cao, các tập tin trong Zero Wine Sandbox thường có dung lượng nhỏ nên bị hạn chế về phân tích các loại tập tin khác nhau [8]. Hệ thống Cukoo cung cấp các công cụ để phân tích mã độc trên môi trường sandbox, tuy nhiên quá trình phân tích được thực tuần tự nên hiệu quả không cao [9].

Để phân tích và xử lý lượng dữ liệu lớn có mã độc, ta có thể triển khai trên môi trường xử lý phân tán. Bài báo này trình bày hướng nghiên cứu, thiết kế và xây dựng hệ thống phân tích và xử lý mã độc trên môi trường điện toán đám mây sử dụng mô hình xử lý phân tán MapReduce để phát hiện sớm và ngăn chặn mã độc nhằm bảo vệ an toàn thông tin cho hệ thống mạng máy tính.

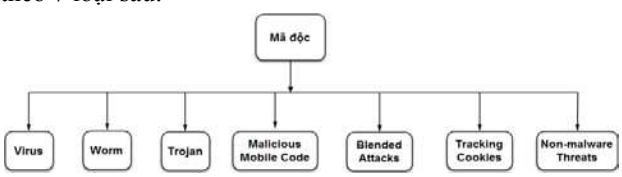
Bài báo này sẽ giới thiệu về mã độc và các phương pháp phân tích mã độc; trình bày về hệ thống sandbox; nghiên cứu xây dựng hệ thống tính toán xử lý sử dụng mô hình MapReduce để phân tích mã độc dựa trên cơ chế sandbox; đánh giá kết quả đạt được.

2. Cơ chế phân tích mã độc

2.1. Mã độc và phân loại mã độc

Mã độc là một chương trình được chèn một cách bí mật vào hệ thống với mục đích làm tổn hại đến tính bí mật, tính toàn vẹn hoặc tính sẵn sàng của hệ thống [5].

Mã độc chia theo các dạng có thể gây hại cho máy tính theo 7 loại sau:



Hình 1. *Phân loại mã độc*

2.2. Các hành vi của mã độc

Khi lây lan trong một máy tính, mã độc có thể có các hành vi sau:

- Thay đổi tập tin trong máy tính: Kiểm tra hành vi liên quan đến việc tạo những tập tin mới, xóa tập tin hoặc thay đổi nội dung của tập tin trên hệ thống.

- Thay đổi các giá trị trong Registry: Kiểm tra hành vi thay đổi trong hệ thống Registry như việc tạo ra các khóa Registry mới hoặc sửa đổi các giá trị trong khóa Registry.

- Cài đặt các phần mềm gián điệp: Thực hiện cài đặt các phần mềm khác liên quan đến hệ thống, tin tặc có thể dựa vào đó để ăn cắp thông tin hoặc thực hiện ý đồ khác mà chúng mong muốn.

- Thực hiện các hoạt động phá hoại: Thực hiện kết nối đến các địa chỉ IP khác, tên miền nào để cập nhật phiên bản mã độc khác hoặc nhận lệnh điều khiển tấn công theo chủ ý của tin tặc.

- Tạo hoặc thay đổi các dịch vụ của hệ điều hành: Các mã độc khi lây nhiễm vào hệ thống sẽ tương tác với registry để có thể ưu tiên khởi động trước lúc hệ thống khởi động.

- Tiêm nhiễm vào các tiến trình khác trên hệ thống: Để tránh bị các chương trình diệt virus phát hiện, các mã độc có thể hoạt động dưới một tiến trình khác hoặc giả mạo các tiến trình hợp pháp của hệ điều hành [5].

2.3. Các phương pháp phân tích mã độc

2.3.1. Phân tích mã độc thủ công

Phân tích thủ công bao gồm: phân tích sơ lược, phân tích hoạt động và phân tích bằng cách đọc mã thực thi. Cả ba bước phân tích trên đều cần thiết và bổ trợ cho nhau để có được kết quả chính xác nhất về hành vi của mã độc [3].

a. Phân tích sơ lược

Phân tích sơ lược là giai đoạn đầu của một quá trình phân tích mã độc và hầu như luôn luôn phải thực hiện. Một phân tích bề mặt thực hiện việc lấy các thông tin ban đầu về tập tin mã độc rồi xác nhận nó có phải mã độc hay không mà không cần thực thi nó.

Những thông tin có thể lấy được từ tập tin mã độc gồm có: loại tập tin, tên tập tin, kích thước tập tin, timestamp và hàm băm (MD5, SHA-1,...).

b. Phân tích hoạt động

Đây là kỹ thuật liên quan đến việc chạy mã độc và giám sát nó trên hệ thống phân tích nhằm bóc tách mã độc hoặc tạo ra dấu hiệu nhận dạng mạng hoặc cả hai. Tuy nhiên, trước khi thực thi mã độc, người phân tích cần thiết kế một môi trường để thực thi mã độc mà không làm ảnh hưởng đến hệ thống mạng thật của tổ chức.

Phương pháp này có thể sẽ không phát hiện hết các hành vi của mã độc. Phân tích hoạt động cần dùng các công cụ hỗ trợ như: Regshot, Sysanalyzer, Process Explorer, HijackThis, Fundelete,…

c. Phân tích bằng cách đọc mã thực thi của mã độc

Phân tích bằng cách đọc mã thực thi là một kỹ thuật sử dụng một bộ tách rời (disassembler) để dịch ngược các đoạn mã bên trong một mã độc thành dạng hợp ngữ để từ đó tìm hiểu các chỉ dẫn lệnh nhằm biết chính xác chương trình mã độc có thể làm những việc gì.

Những chỉ dẫn lệnh được thực thi bởi CPU, do vậy nó sẽ cho ta biết chính xác những gì chương trình mã độc thực hiện. Tuy nhiên để có thể thực hiện được phân tích tĩnh, đòi hỏi người phân tích phải am hiểu sâu về hợp ngữ, các mã chỉ dẫn lệnh và các khái niệm, các hàm API trong hệ điều hành.

3.3.2. Phân tích mã độc tự động

Số lượng mã độc sinh ra và được phát tán ngày càng nhiều và đều đặn mỗi ngày, thì việc phân tích thủ công coi như không thể thực hiện kịp với số lượng lớn mã độc. Do vậy cần phải có một hệ thống tự động phân tích để giúp người quản trị có thể phân tích số lượng lớn mã độc nhằm cung cấp những thông tin và dấu hiệu nhận dạng cụ thể từng mã độc trước khi thực hiện phân tích sâu hơn; có nghĩa là các công đoạn phân tích đều do hệ thống tự động thực hiện, từ công đoạn nhận mã độc cho đến 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.

2.4. Hệ thống sandbox

Hiện nay các hệ thống sanbox được sử dụng nhằm phân tích tự động một lượng lớn các mẫu mã độc và là một bước đầu tiên trong quá trình phân tích mã độc hoàn chỉnh. Cần phải có những cách đơn giản để gửi các tập tin mã độc vào sandbox, xác định các tùy chọn và tính năng cho người chạy phân tích và trích xuất kết quả phân tích.

Sandbox dễ sử dụng và cung cấp một bản tóm tắt ở mức cao nhất các hoạt động nguy hiểm mà mã độc thực hiện trong thời gian phân tích. Trong trường hợp kết quả phân tích cho thấy dấu hiệu nghi ngờ cần kiểm tra kỹ, thì sẽ được phân tích chuyên sâu hơn.

3. Phân tích mã độc trên môi trường phân tán

Mô hình phân tán là một hệ thống có chức năng và dữ liệu phân tán trên các máy trạm được kết nối với nhau bởi một mạng máy tính. Nếu các máy tính này cùng sử dụng chung trên một phần cứng thì được gọi là một cụm (cluster), ngược lại hoạt động riêng rẽ trên các phần cứng khác nhau thì chúng được gọi là một lưới (grid).

3.1. Mô hình xử lý phân tán MapReduce

3.1.1. Giới thiệu

MapReduce là mô hình xử lý phân tán cho phép các ứng dụng có thể xử lý lượng dữ liệu lớn. Các dữ liệu này được đặt tại các máy tính phân tán nhằm khai thác kinh nghiệm tính toán giúp rút ngắn thời gian xử lý toàn bộ dữ liệu [2].

Dữ liệu đầu vào có thể là dữ liệu có cấu trúc (dữ liệu lưu trữ dạng bảng quan hệ hai chiều) hoặc dữ liệu không cấu trúc (dữ liệu dạng tập tin hệ thống).

3.1.2. Nguyên tắc hoạt động của MapReduce

Quá trình MapReduce thực hiện hai hàm Map() và Reduce(). Hệt thống triển khai bao gồm máy master (máy chủ) và máy slave (máy trạm). Trong đó máy master làm nhiệm vụ điều phối sự hoạt động của quá trình thực hiện MapReduce trên các máy slave. Các máy slave làm nhiệm vụ thực hiện quá trình Map và Reduce với dữ liệu mà nó nhận được [4].

- Thực hiện hàm Map():

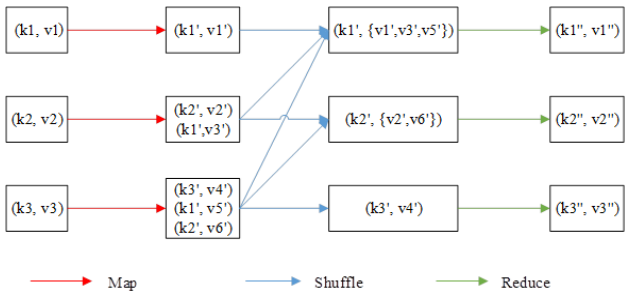
Máy master sẽ phân phối các tác vụ Map và Reduce vào các máy slave sẵn sàng. Các tác vụ này được master phân phối cho các máy dựa trên vị trí của dữ liệu liên quan trong hệ thống. Máy slave khi nhận được tác vụ Map sẽ đọc dữ liệu được nhận từ phân vùng dữ liệu đã gán cho nó và thực hiện hàm Map. Kết quả đầu ra là các cặp *<key, value>* trung gian. Các cặp này được lưu tạm trên bộ nhớ đệm của các máy.

Sau khi thực hiện xong công việc Map. Các máy slave làm nhiệm vụ chia các giá trị trung gian thành R vùng (tương ứng với R tác vụ Reduce) lưu xuống đĩa và thông báo kết quả, vị trí lưu cho máy master.

- Thực thi tác vụ Reduce():

Máy master sẽ gán các giá trị trung gian và vị trí của các dữ liệu đó cho các máy thực hiện công việc slave. Các máy slave làm nhiệm vụ xử lý sắp xếp các key, thực hiện hàm Reduce và đưa ra kết quả cuối.

Sơ đồ hoạt động của quá trình MapReduce được biểu diễn như trong Hình 2.



Hình 2. *Sơ đồ hoạt động của quá trình MapReduce*

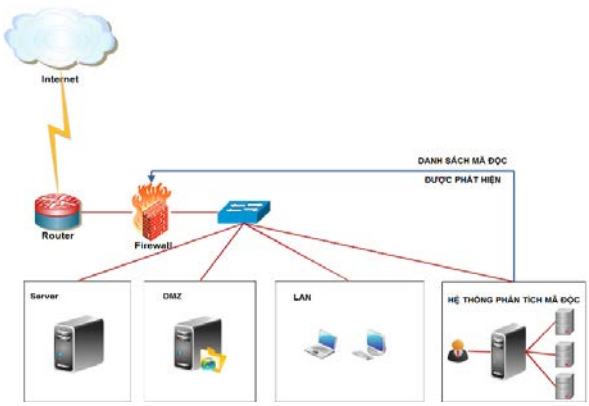
3.2. Nền tảng Hadoop

Apache Hadoop là một framework dùng để chạy những ứng dụng trên một cụm máy tính lớn được xây dựng trên những phần cứng thông thường. Hadoop hiện thực mô hình MapReduce, đây là mô hình mà ứng dụng sẽ được chia nhỏ ra thành nhiều phân đoạn khác nhau, và các phần này sẽ được chạy song song trên nhiều nút khác nhau. Nhờ cơ chế streaming, Hadoop cho phép phát triển các ứng dụng phân tán bằng cả java lẫn một số ngôn ngữ lập trình khác như C++, Python, Pearl.Các thành phần của Hadoop bao gồm: Core, HDFS, MapReduce, Hbase, Hive, Chunka, Pig.

3.3. Hệ thống phân tích mã độc

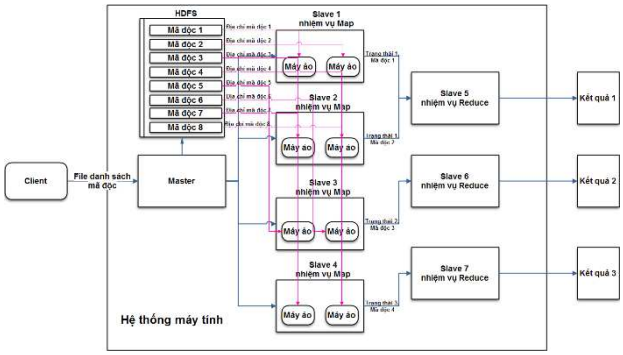
3.3.1. Mô hình phân tích mã độc

Với số lượng mã độc cần phân tích ngày càng nhiều thì việc phân tích bằng thủ công như hiện nay là một điều không thể do nhân lực có hạn cũng như mỗi công đoạn đều thực hiện bằng tay nên tốn rất nhiều thời gian. Để khắc phục vấn đề này, cần thiết phải xây dựng một hệ thống có khả năng tự động phân tích hành vi của mã độc để hỗ trợthêm cho phương pháp phân tích thủ công, đồng thời dựa vào hệ thống này có thể nhanh chóng đưa ra một số biện pháp xử lý kịp thời nhằm hạn chế táchại do mã độc gây ra.



Hình 3. *Mô hình tổng quan hệ thống*

Hệ thống được thiết kế theo Hình 3. Trong đó mô hình phân tích mã độc được thể hiện ở Hình 4.

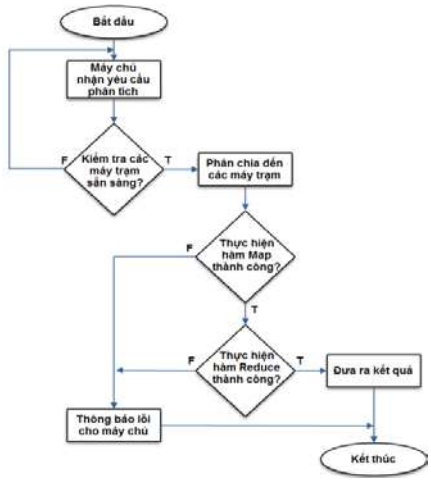


Hình4. *Mô hình hệ thống phân tích mã độc*

3.3.2. Quy trình thực hiện

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.



Hình 5. *Quy trình thực hiện phân tích mã độc*

Quy trình thực hiện phân tích mã độc được mô tả theo các bước tổng quan và thuật toán như sau (Hình 5):

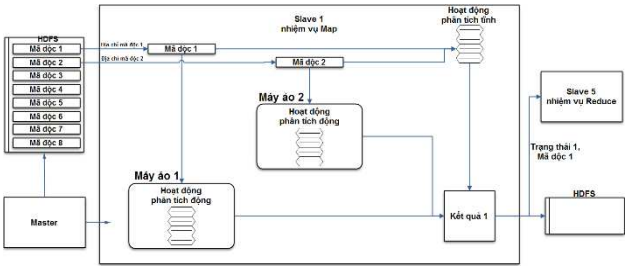
- Đầu vào:
  + Danh sách tập tin nghi ngờ có mã độc
  + Địa chỉ URL nghi ngờ mã độc
- Đầu ra:
  + Kết luận tập tin có nhiễm mã độc hay không
  + Thông tin mô tả hành vi mã độc
  + Thống kê các mã độc được phân tích
- Thuật toán:

3.3.3. 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.

Sau đó, sẽ thực hiện chạy hoạt động phân tích tĩnh. Tiếp theo, mã độc sẽ được chép vào máy ảo để thực hiện công việc phân tích động. Tại máy ảo, mã độc được thực thi và những hành vi của mã độc sẽ được ghi lại.

Sơ đồ hoạt động của các máy slave làm nhiệm vụ Map được mô tả bằng Hình 6:

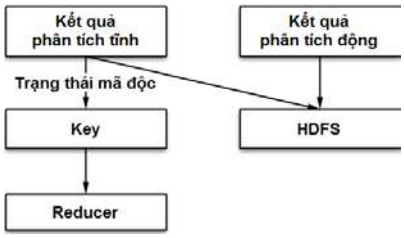


Hình6. *Sơ đồ hoạt động của máy làm nhiệm vụ Map*

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 trên Hình 7:



Hình 7. *Xử lý kết quả phân tích*

Kết quả 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. Trạng thái mã độc chính là đầu ra của quá trình Map và là đầu vào của quá trình Reduce.

Đoạn mã lệnh thể hiện cơ chế Map mã độc:

```
public void map(LongWritable key, Text value,
OutputCollector<Text, Text> output, Reporter
reporter) throws IOException {

// TODO Auto-generated method stub
Configuration conf = new Configuration();
conf.addResource(new Path(CORE_SITE_PATH));

//copy file from HDFS to Local
filename = value.toString();
FileSystem fs = FileSystem.get(conf);
fs.copyToLocalFile(new Path(COPY_DIR,filename), new
Path(LOCAL_DATA_DIR));

// Run script which implements the static analysis
runScript(filename);

// Run dynamic analysis
boolean succeed = runDynamicAnalysis(filename);
...
}
```

3.3.4. Cơ chế Reduce mã độc

Cơ chế Reduce mã độc được thể hiệnnhư sau:

```
public void reduce(Text key, Iterator<Text> values,
OutputCollector<Text, Text> output, Reporter
reporter) throws IOException {
// TODO Auto-generated method stub
StringBuilder sb = new StringBuilder();
while(values.hasNext()){
Text text = values.next();
sb.append(text);
if(values.hasNext()) sb.append(",");
}
output.collect(key, new Text(sb.toString()));
}
```

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.

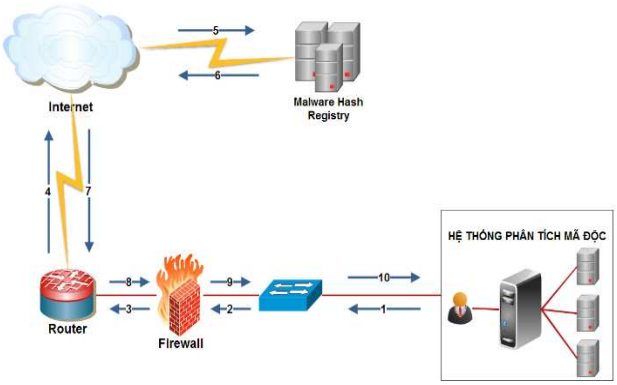
3.4. Các mô đun chức năng

3.4.1. Chức năng phân tích tĩnh

Để thực hiện phần này, ta cần phải có kết nối mạng. Với một đoạn Linux shell script sẽ tự động việc tính giá trị MD5 và gởi lên dịch vụ Malware Hash Registry của Team Cymru để kiểm tra[10]. Dịch vụ này sẽ phản hồi lại là mã độc đã được phát hiện trước đây hay chưa, khả năng các Antivirus phát hiện là bao nhiêu phần trăm.

Đoạn lệnh thực hiện chức năng phân tích tĩnh:

```
#!/bin/bash
MALWARE=$1
MD5='md5sum ${MALWARE} | awk '{print $1}''
whois -h hash.cymru.com ${MD5} > ${MALWARE}.static
```



Hình *8. Tổng quan hoạt động chức năng phân tích tĩnh*

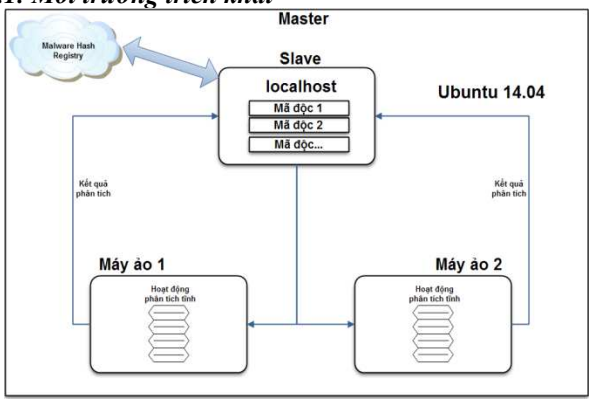
Kết quả nhận được báo cáo thống kê bao gồm: STT, tên mã độc, tình trạng mã độc (NOT OK, OK, N/A), phần trăm Antivirus phát hiện và báo cáo thống kê bao gồm: số lượng mã độc được phân tích, số lượng mã độc bị phát hiện, số lượng mã độc không bị phát hiện, số lượng tập tin xảy ra lỗi trong quá trình phân tích.

3.4.2. Chức năng phân tích động

Quá trình phân tích mã độc được thực thi trong môi trường sandbox, đây là môi trường độc lập để tránh ảnh hưởng đến các hệ thống khác. Trong hệ thống, chúng tôisử dụng hệ điều hành Ubuntu 14.04, môi trường thực thi mã độc là Windows XP. Chương trình sử dụng bộ thư viện VIX API cho phép viết các đoạn script tự động hóa các thao tác bật tắt máy ảo, truyền tập tin giữa máy thật và máy ảo. Chức năng snapshot của VMware giúp khôi phục lại nhanh chóng dữ liệu và cấu hình đã được đánh dấu trước đó, cũng có nghĩa là chúng ta có thể chọn môi trường để tiến hành phân tích cho phù hợp. Điều này rất thuận lợi cho việc phân tích mã độc.

4. Kết quả thử nghiệm và đánh giá

4.1. Môi trường triển khai



Hình 9. *Mô hình triển khai thử nghiệm*

Hệ thống phân tích mã độc được triển khai trên:

- Hệ điều hành Ubuntu 14.04

- Nền tảng Hadoop: 1.2.1

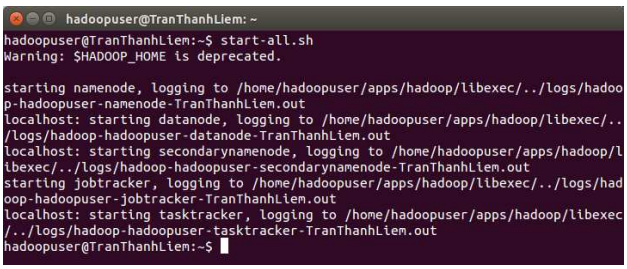
-Phiên bản máy ảo: VMware Workstation 10.0.3, phiên bản Linux

- Hệ điều hành máy ảo thực thi mã độc: Windows XP

- Thực hiện thử nghiệm khoảng 300 mã độc, nguồn từ VNCERT [11]forum BKAV [12] và virussign [13]

- Mô hình thử nghiệm gồm có một máy master và một máy slave

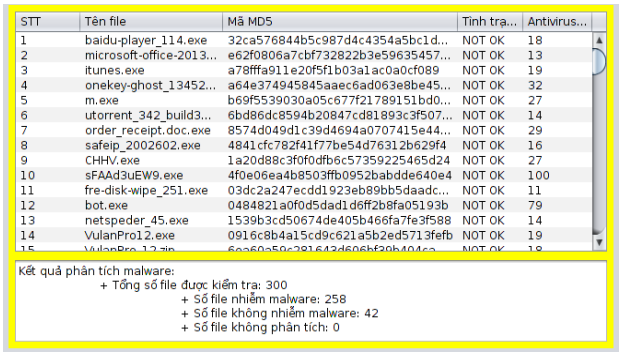
4.2. Kết quả triển khai



Hình 10. *Khởi chạy các dịch vụ của Hadoop*

Để khởi động các dịch vụ Hadoop, ta dùng lệnh *start-all.sh.* Có thể kiểm tra lại chương trình đã chạy hay chưa bằng lệnh *jps.*

Trong chương trình, ta chọn tập tin (*.txt) chứa tên các mã độc để phân tích. Sau đó kích vào nút Malware Analysis để chạy chương trình. Việc phân tích này được thực hiện trên nhiều mẫu thử lần lượt là 50, 100, 150, 200, 250 và 300 mã độc.



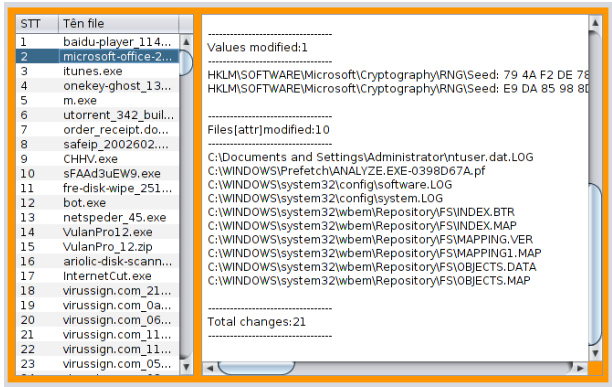
Hình 11. *Kết quả phân tích tĩnh 300 mã độc*

Kết quả phân tích tĩnh sẽ cung cấp thông tin bao gồm 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 1. *Thời gian phân tích tĩnh 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 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%.



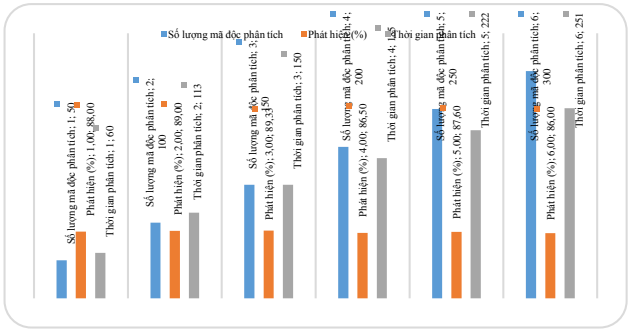
Hình 12. *Kết quả phân tích động*

Kết quả phân tích động sẽ hiển thị danh sách mã độc bên trái và khi kích vào mã độc sẽ hiển thị những hành vi của mã độc đó tác động vào hệ thống thông qua công cụ Regshot. Hình 12 minh họa kết quả phân tích động.

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. Các công đoạn phân tích đều được 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, 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 [6].



Hình 13. *Biểu đồ thời gian phân tích tĩnh mã độc trên một máy*

Bảng 1 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 13).

Khi triển khai 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: khoảng 150 giây. Các giai đoạn khác đã được kiểm chứng thực nghiệm nên kết quả sẽ < 155 giây.

- Mỗi mã độc mất trung bình 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.

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.

5. Kết luận

Bài báo nghiên cứu xây dựng mô hình tính toán phân tán để phân tích mã độc sử dụng. 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. Hệ thống sandbox trên môi trường phân tán giúp cho việc phân tích mã độc được thực hiện một cách an toàn, ngoài ra còn giúp cho việc giảm thời gian phân tích, tăng hiệu quả công việc. Hướng nghiên cứu tiếp theo sẽ xây dựng cơ sở dữ liệu mã độc, xây dựng các chức năng báo cáo, thống kê số liệu và cảnh báo sớm.

Tài liệu tham khảo

[1] Alexis Galarza (2011), Automated Malware Analysis using MapReduce and Virtualization, Universidad del Turabo
[2] Amol G. Kakade, Prashant K. Kharat, Anil Kumar Gupta (2013), Survey of Spam Filtering Techniques and Tools, and Map Reduce with SVM
[3] Dennis Distler (2007), Malware Analysis: An Introduction
[4] Kyuseok Shim (2012), Map Reduce Algorithms for Big Data Analysis, Seoul National University, Korea
[5] Peter Mell, Karen Kent, Joseph Nusbaum (2005), Guide to Malware Incident Prevention and Handling, America
[6] https://www.virustotal.com/
[7] http://bsa.isoftware.nl/
[8] http://zerowine.sourceforge.net/
[9] http://www.cuckoosandbox.org/
[10] https://www.team-cymru.org/Services/MHR/
[11] http://www.bkav.com.vn/
[12] http://vncert.gov.vn/
[13] http://virussign.com/

(BBT nhận bài: 22/08/2015, phản biện xong: 19/10/2015)