



BÁO CÁO ĐỒ ÁN

Đánh giá hiệu năng hệ thống mạng với giao thức SNMP và công cụ Observium

Môn học: Đánh giá hiệu năng hệ thống mạng máy tính

Lớp: NT531.O21.MMCL

Giảng viên: PGS.TS Lê Trung Quân, Ths. Lê Anh Tuấn

THÀNH VIÊN THỰC HIỆN (Nhóm 4):

STT	Họ và tên	MSSV
1	Đặng Trung Hậu	21520833
2	Nguyễn Gia Quân	21521327

ĐÁNH GIÁ KHÁC:

Tổng thời gian thực hiện	
Phân chia công việc	<p>Đặng Trung Hậu:</p> <ul style="list-style-type: none">- Tìm hiểu lý thuyết SNMP.- Tìm hiểu và cấu hình Observium.- Xây dựng mô hình, cấu hình SNMP và Observium trên mô hình thực nghiệm.- Viết báo cáo word, powerpoint. <p>Nguyễn Gia Quân:</p> <ul style="list-style-type: none">- Viết báo cáo word.
Ý kiến (nếu có) + Khó khăn + Đề xuất, kiến nghị	

Phần bên dưới của báo cáo này là báo cáo chi tiết của nhóm thực hiện

MỤC LỤC

A.	LỜI MỞ ĐẦU	2
1.	Giới thiệu về đề tài	2
2.	Mục tiêu của đề tài	3
B.	BÁO CÁO CHI TIẾT	3
3.	Tổng quan về giám sát hiệu năng mạng	3
3.1.	Một số khái niệm cơ bản	3
3.2.	Các yếu tố cơ bản trong hệ thống giám sát hiệu năng	4
4.	Cơ bản về giao thức Simple Network Management Protocol (SNMP)	5
4.1.	Khái niệm về giao thức	5
4.2.	Các thành phần chính của giao thức SNMP	5
4.3.	Phương thức hoạt động của giao thức SNMP	6
4.4.	So sánh các phiên bản SNMP	7
5.	Phát triển hệ thống giám sát dựa trên công cụ Observium	7
5.1.	Tổng quan về Observium	7
5.2.	Phân tích source code	9
5.3.	So sánh với các công cụ khác	11
6.	Triển khai thực nghiệm và kết quả	13
6.1.	Mô hình triển khai	13
6.2.	Kết quả thực nghiệm	14
7.	Đánh giá, kết luận	14
7.1.	Kết quả đạt được	14
7.2.	Hướng phát triển	15
C.	TÀI LIỆU THAM KHẢO	15

A. LỜI MỞ ĐẦU

1. Giới thiệu về đề tài

Trong môi trường mạng ngày nay, việc quản lý và giám sát hệ thống mạng trở nên ngày càng phức tạp và quan trọng. Đối với các tổ chức và doanh nghiệp, việc duy trì hiệu suất ổn định và tính khả dụng của hệ thống mạng là yếu tố quyết định đến sự thành công và an toàn của họ. Trong bối cảnh này, công cụ giám sát mạng trở thành một phần không thể thiếu để giúp các quản trị viên mạng theo dõi, phát hiện và giải quyết các vấn đề mạng một cách hiệu quả.

Đồng thời, việc lựa chọn một công cụ giám sát phù hợp có thể đáp ứng các yêu cầu đa dạng của mạng ngày nay cũng đang trở thành một thách thức đối với các nhà quản trị mạng. Trong bối cảnh này, Observium, một công cụ giám sát mạng mã nguồn mở, đã nổi lên như một giải pháp hữu ích và hiệu quả.

Đề tài nghiên cứu này tập trung vào việc triển khai và áp dụng công cụ Observium vào môi trường mạng cụ thể. Bằng cách nghiên cứu và thực hiện triển khai thực tế, chúng tôi nhằm mục đích hiểu rõ hơn về khả năng và ưu điểm của Observium trong việc giám sát hệ

thống mạng, từ đó đưa ra những đề xuất và khuyến nghị về việc sử dụng và tối ưu hóa công cụ này trong các môi trường mạng thực tiễn.

2. Mục tiêu của đề tài

Mục tiêu chính của đề tài này là tìm hiểu, phân tích và triển khai công cụ giám sát mạng Observium trong một môi trường mạng cụ thể.

Cụ thể, các mục tiêu của nghiên cứu bao gồm:

Hiểu rõ về Observium: Nắm vững kiến thức về tính năng, cấu trúc và hoạt động của Observium. Điều này bao gồm việc tìm hiểu về cách cài đặt, cấu hình và sử dụng các tính năng quan trọng của công cụ này.

Triển khai thực tế: Thực hiện triển khai Observium trong một môi trường mạng thực tế, bao gồm việc cài đặt, tích hợp và cấu hình để có thể giám sát các thiết bị và hệ thống mạng.

Đánh giá và So sánh: Đánh giá hiệu suất và tính năng của Observium trong môi trường triển khai so với các công cụ giám sát mạng khác. So sánh ưu và nhược điểm của Observium so với các giải pháp khác nhau.

Tối ưu hóa và Đề xuất: Đề xuất các biện pháp tối ưu hóa và cải thiện sự hiệu quả của Observium trong môi trường mạng, dựa trên kết quả đánh giá và phân tích.

Đề xuất phát triển: Xây dựng hướng dẫn về việc sử dụng Observium và đề xuất hướng phát triển tương lai cho công cụ này, để hỗ trợ các tổ chức và doanh nghiệp trong việc quản lý và giám sát hệ thống mạng.

Trong bài báo cáo này chúng tôi sẽ trình bày với các thành phần:

- ✚ Tổng quan về giám sát hiệu năng mạng
- ✚ Cơ bản về giao thức Simple Network Management Protocol
- ✚ Phát triển hệ thống giám sát dựa trên công cụ Observium
- ✚ Triển khai thực nghiệm và kết quả
- ✚ Đánh giá, kết luận

B. BÁO CÁO CHI TIẾT

3. Tổng quan về giám sát hiệu năng mạng

3.1. Một số khái niệm cơ bản

- Giám sát hiệu năng mạng

Giám sát hiệu năng mạng là việc thu thập các thông tin trên các thành phần của hệ thống, phân tích các thông tin, dấu hiệu nhằm đánh giá và đưa ra các cảnh báo cho người quản trị hệ thống.

Đối tượng của giám sát hiệu năng mạng là tất cả các thành phần, thiết bị trong hệ thống mạng:

- Các máy trạm
- Cơ sở dữ liệu
- Các ứng dụng
- Các server

- Các thiết bị mạng
- Các thành phần trong hệ thống mạng

Để một hệ thống mạng hoạt động tốt nó bao gồm rất nhiều thành phần, hoạt động trên các nền tảng và môi trường khác nhau

- Các máy trạm
- Các máy chủ
- Các thiết bị hạ tầng mạng: Router, switch, Hub...
- Các thiết bị, hệ thống phát hiện và phòng chống xâm nhập: IDS/IPS, Snort, FireWall...
- Các ứng dụng chạy trên các máy chủ và máy trạm.
- Log hệ thống

Là một thành phần quan trọng của hệ thống mạng. Nó lưu lại một cách chính xác mọi hoạt động của hệ thống, tình trạng hoạt động của hệ thống, các ứng dụng, các thiết bị đã và đang hoạt động trong hệ thống.

- Các loại log chính trong hệ thống:
 - Log Access: Là log ghi lại toàn bộ thông tin truy cập của người dùng tới hệ thống, truy cập của các ứng dụng tới cơ sở dữ liệu...
 - Log Event: là log ghi lại chi tiết những sự kiện mà hệ thống đã thực hiện. Log ứng dụng, log của hệ điều hành...
 - Log Device: là log ghi lại tình trạng hoạt động của các thiết bị phần cứng và phần mềm đang được sử dụng: Router, Switch, IDS, IPS...

Log là một thành phần cực kỳ hữu hiệu cho việc giám sát cũng như khắc phục các sự cố trong hệ thống mạng. Tuy nhiên, với những hệ thống lớn, chạy nhiều ứng dụng, lượng truy cập cao thì công việc phân tích Log thực sự là một điều vô cùng khó khăn.

3.2 Các yếu tố cơ bản trong hệ thống giám sát hiệu năng

Để công tác giám sát hiệu năng mạng đạt hiệu quả cần phải xác định được các yếu tố cốt lõi, cơ bản nhất của giám sát như:

- Xác định các đơn vị, hệ thống, thiết bị, dịch vụ cần giám sát.
- Xác định trạng thiết bị, giải pháp phần mềm thương mại phục vụ giám sát.
- Xác định phần mềm nội bộ và phần mềm nguồn mở phục vụ giám sát.
- Xác định các thiết bị, công cụ, giải pháp hỗ trợ phân tích kết quả giám sát: công cụ NMAP, TCPDUMP, Wireshark, Nessus...

Ngoài các trang thiết bị, công cụ, giải pháp hỗ trợ thì yếu tố con người và đặc biệt là quy trình phục vụ giám sát là vô cùng quan trọng.

4. Cơ bản về giao thức Simple Network Management Protocol (SNMP)

4.1 Khái niệm về giao thức

Simple Network Management Protocol (SNMP) hoạt động ở lớp ứng dụng, chạy cổng 160, 161, 162 trên UDP. SNMP dùng để quản lý các thiết bị trong mạng như router, server, ... SNMP không chỉ cho phép kiểm soát việc hoạt động của các thiết bị mạng mà còn có thể quản lý các thiết bị mạng từ xa. SNMP có khả năng theo dõi, lấy thông tin, đưa ra thông báo và tác động đến các hoạt động hệ thống mạng theo ý muốn của người quản trị.

Ví dụ: SNMP có thể theo dõi tốc độ đường truyền, tự động gửi thông báo cho người quản trị khi có một cổng trên router bị tắt, xem thông tin ổ cứng của server, ...

4.2 Các thành phần chính của giao thức SNMP

- Một hệ thống sử dụng SNMP bao gồm 2 thành phần cơ bản: NMS (Network Management Station) và NE (Network Element)
- NMS (SNMP manager) là một máy tính chạy phần mềm quản lý SNMP (PRTG, Solarwinds) để thực hiện giám sát và điều khiển tập trung các SNMP agent.
- NE (SNMP agent) là các thiết bị mạng hỗ trợ SNMP và chịu sự quản lý của các SNMP manager.
- Một SNMP manager có thể quản lý nhiều SNMP agent và ngược lại một SNMP agent có thể chịu sự quản lý của nhiều SNMP manager.

Một số khái niệm cơ bản của giao thức SNMP:

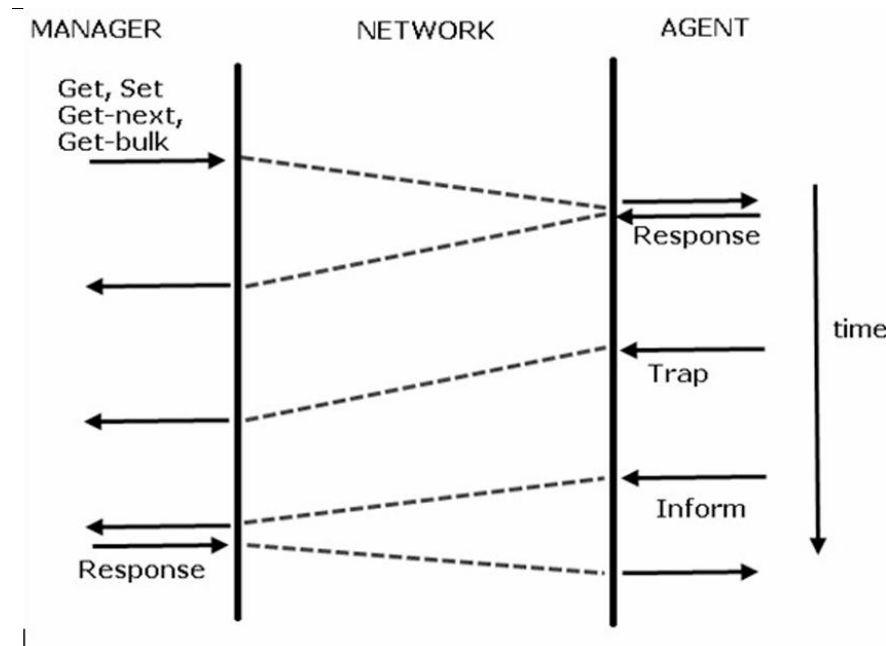
- Object:
 - Mỗi thiết bị có hỗ trợ SNMP cung cấp nhiều thông tin khác nhau, mỗi thông tin đó được gọi là object. Ví dụ: router có các thông tin về tổng số card, tổng số cổng, tổng số bit đã truyền/nhận, tên router, trạng thái tắt/mở của các cổng.
 - Mỗi object có một tên gọi riêng và một mã số Object ID (OID) để nhận dạng object đó. Ví dụ: tên thiết bị được gọi là sysName, OID là 1.3.6.1.2.1.1.5.
 - Mỗi object chỉ có duy nhất một OID nhưng có thể có nhiều tên gọi nên người ta sử dụng một chỉ số sub-id để phân biệt các object. Ví dụ: Tên thiết bị được gọi là sysName, OID là 1.3.6.1.2.1.1.5; nếu thiết bị có 2 tên thì chúng sẽ được gọi là sysName.0 và sysName.1 và có OID lần lượt là 1.3.6.1.2.1.1.5.0 và 1.3.6.1.2.1.1.5.1.
 - Một số object phổ biến thì được chuẩn hóa OID, riêng các object mới được tạo ra theo yêu cầu của cá nhân thì phải được mô tả OID. Để lấy một thông tin có OID đã chuẩn hóa thì ứng dụng SNMP phải gửi một gói tin SNMP có chứa OID của object đó cho SNMP agent, SNMP agent khi nhận được thì nó phải trả lời bằng thông tin ứng với OID đó.
 - Object access quy định quyền truy cập của mỗi object là READ_ONLY (chỉ cho phép đọc object) hoặc READ_WRITE (cho phép đọc và thay đổi giá trị object). Ví dụ: tên của một thiết bị (sysName) thì ta có thể thay đổi nên có quyền READ_WRITE, còn giá tổng số cổng của thiết bị (ifNumber) là READ_ONLY thì không thể đổi.
 - MIB (Management Information Base) là một cấu trúc dữ liệu gồm các object được quản lý, được dùng cho việc quản lý các thiết bị chạy trên nền TCP/IP. MIB được thể hiện thành 1 tập tin (MIB file) và có thể biểu diễn thành 1 cây (MIB tree). Muốn

hiểu được một OID thì cần có tập tin MIB mô tả OID đó. Các thiết bị được quản lý bằng SNMP chỉ khi các ứng dụng SNMP manager và SNMP agent cùng hỗ trợ một MIB. Các ứng dụng này cũng có thể hỗ trợ cùng lúc nhiều MIB.

4.3 Phương thức hoạt động của giao thức SNMP

- **GetRequest:** Manager gửi GetRequest (chứa một hoặc nhiều OID) đến agent để lấy một hoặc nhiều giá trị của object trong MIB.
- **GetNextRequest:** Manager gửi GetNextRequest đến agent dùng để lấy giá trị của object nằm kế tiếp object được chỉ ra trong MIB.
- **GetBulkRequest:** Manager gửi GetBulkRequest đến agent để lấy nhiều giá trị của nhiều object.
- **InformRequest:** Manager gửi InformRequest đến manager nhằm trao đổi thông tin với nhau.
- **SetRequest:** Manager gửi SetRequest đến agent để thiết lập giá trị cho object dựa vào OID. Chỉ những object có quyền READ_WRITE mới có thể thiết lập giá trị được.
- **SetResponse:** Sau khi Agent nhận được các thông điệp GetRequest, GetNextRequest hay SetRequest thì nó sẽ gửi GetResponse để trả lời. Trong GetResponse có chứa OID của object được yêu cầu và giá trị của object đó.
- **Trap:** Trap được agent tự động gửi đến manager để thông báo trong agent có sự kiện hay biến cố xảy ra.

Ví dụ: khi có một công bị tắt, người dùng đăng nhập thất bại. Việc gửi hay không gửi khi biến cố xảy ra do hãng sản xuất thiết bị agent quy định. SNMP request/reponse dùng để quản lý còn SNMP trap dùng để cảnh báo. Nguồn gửi trap được gọi là Trap Sender và nơi nhận trap được gọi là Trap Receiver. Mỗi trap sender có thể gửi nhiều trap đến nhiều trap receiver cùng lúc. Trap gồm 2 loại chính: generic trap (được qui định trong các chuẩn SNMP) và specific trap (do hãng sản xuất tự định nghĩa). Có thể phân biệt loại trap dựa vào mã số là một số nguyên chứa trong gói tin trap. Theo SNMPv1, generic trap có 7 loại sau: coldStart(0), warmStart(1), linkDown(2), linkUp(3), authenticationFailure(4), egpNeighborloss(5), enterpriseSpecific(6).



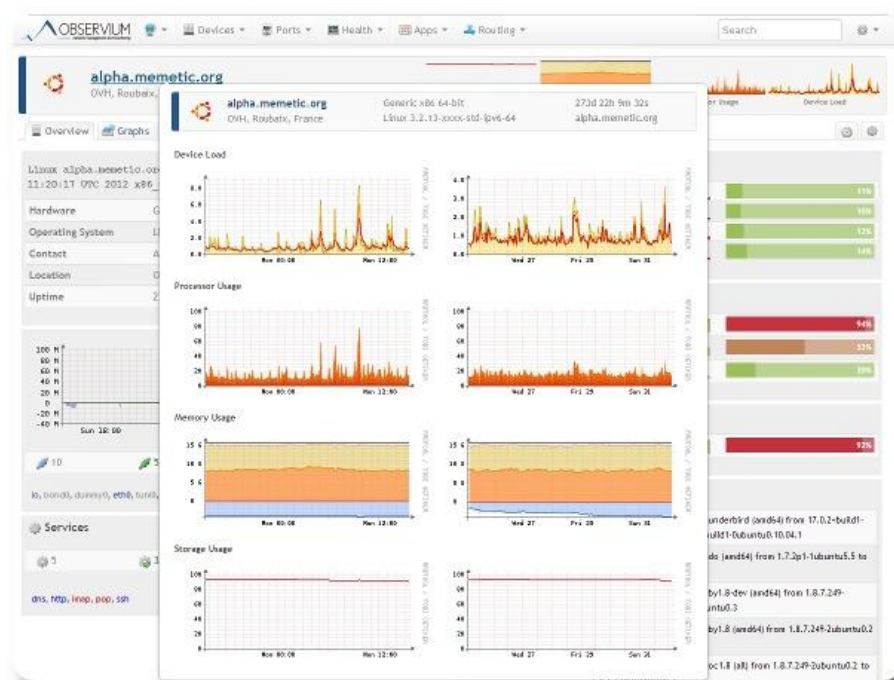
4.4 So sánh các phiên bản SNMP

- Tính đến năm 2022, SNMP có 3 phiên bản khác nhau bao gồm:
 - **SNMP phiên bản 1 (SNMPv1):** Triển khai đầu tiên, hoạt động trong đặc tả thông tin quản lý cấu trúc và được mô tả trong tài liệu RFC 1157.
 - **SNMP phiên bản 2 (SNMPv2):** Phiên bản được cải tiến để hỗ trợ xử lý lỗi hiệu quả hơn và được mô tả trong RFC 1901. Lần đầu tiên được giới thiệu trong RFC 1441 hay còn thường được gọi là SNMPv2c.
 - **SNMP phiên bản 3 (SNMPv3):** Phiên bản này cải thiện tính bảo mật, quyền riêng tư và được giới thiệu trong RFC 3410.
- SNMPv2 là phiên bản giao thức SNMP phổ biến nhất trong thời điểm hiện tại. Trong khi đó, phiên bản SNMPv3 là gần nhất và được bổ sung hỗ trợ xác thực, mã hóa tập tin và các tin nhắn SNMP cũng như bảo vệ các gói tin trong quá trình truyền đi.

5. Phát triển hệ thống giám sát dựa trên công cụ Observium

5.1 Tổng quan về Observium

- Giới thiệu:
 - Observium là một nền tảng giám sát mạng đầy đủ tính năng với giao diện trang nhã và mạnh mẽ, mạnh mẽ nhưng đơn giản và trực quan. Nó hỗ trợ một số nền tảng bao gồm Linux, Windows, FreeBSD, Cisco, HP, Dell và nhiều nền tảng khác, và bao gồm tính năng tự động phát hiện thiết bị. Nó giúp người dùng thu thập số liệu mạng và cung cấp đồ thị trực quan về số liệu thiết bị từ dữ liệu hiệu suất được thu thập.



- **Tính năng:**
- **Khám phá thiết bị tự động** – Một trong những tính năng nổi bật của Observium là khả năng tự động khám phá các thiết bị trên mạng của bạn (thông qua tập lệnh). Điều này có nghĩa là quản trị viên trung tâm dữ liệu không phải thêm mọi thiết bị họ muốn giám sát theo cách thủ công.
- **Giám sát thời gian thực** – Observium liên tục thu thập dữ liệu từ các thiết bị được giám sát, cung cấp thông tin chi tiết theo thời gian thực về hiệu suất mạng. Quản trị viên có thể xem dữ liệu như mức sử dụng băng thông, mức sử dụng CPU và bộ nhớ cũng như trạng thái thiết bị trong một trang tổng quan duy nhất. Khả năng hiển thị thời gian thực này cho phép xác định nhanh chóng các vấn đề và xử lý sự cố chủ động.
- **Khả năng mở rộng** – Môi trường trung tâm dữ liệu rất năng động và có thể phát triển nhanh chóng. Observium Community Edition có thể mở rộng quy mô theo cơ sở hạ tầng của bạn, hỗ trợ nhiều loại thiết bị và giao thức. Cho dù bạn đang quản lý một trung tâm dữ liệu nhỏ hay mạng doanh nghiệp lớn, Observium đều có thể thích ứng với nhu cầu của bạn.
- **Cảnh báo và Thông báo** – Để đảm bảo rằng quản trị viên được thông báo về các sự kiện quan trọng, Observium cung cấp khả năng cảnh báo và thông báo mạnh mẽ. Các

quy tắc cảnh báo có thể tùy chỉnh có thể được định cấu hình để kích hoạt thông báo qua email, SMS hoặc các kênh liên lạc khác, đảm bảo rằng các vấn đề tiềm ẩn được giải quyết kịp thời.



























- **Dữ liệu lịch sử** – Observium không chỉ tập trung vào giám sát thời gian thực; nó cũng lưu trữ dữ liệu lịch sử. Dữ liệu lịch sử này là vô giá cho việc phân tích xu hướng, lập kế hoạch năng lực và kiểm tra tuân thủ. Quản trị viên có thể dễ dàng truy cập dữ liệu hiệu suất lịch sử để đưa ra quyết định sáng suốt về phân bổ và tối ưu hóa nguồn lực.
- **Đồ thị thiết bị và giao diện** – Trực quan hóa dữ liệu là điều cần thiết để hiểu hiệu suất mạng. Observium tạo ra các đồ thị và biểu đồ chi tiết cho các thiết bị và giao diện, giúp quản trị viên dễ dàng xác định xu hướng, điểm bất thường và tắc nghẽn.





5.2 Phân tích source code

Phân tích open source code dựa trên phiên bản community. Hầu hết các tính năng được viết bằng ngôn ngữ PHP

Chi tiết về source code tham khảo tại link github: [Observium-community-edition](https://github.com/observium/observium-community-edition)

```
> .github
> html
> includes
> libs
> logs
> mibs
> rrd
> scripts
> templates
> tests
```

- >  update
 -  .gitignore
 -  .phpcs.xml
 -  INSTALL
 -  LICENSE.COMMUNITY
 -  README
 -  VERSION
 -  add_device.php
 -  adduser.php
 -  alerter.php
 -  check-errors.php
 -  config.php.default
 -  config_to_json.php
 -  delete_device.php
 -  discovery.php
 -  housekeeping.php
 -  irc.php
 -  notifications.php
 -  observium-wrapper
 -  poller-wrapper.py
 -  poller.php
 -  rename_device.php
 -  snmp.conf.example
 -  snmpd.conf.example
 -  snmptrap.php
 -  syslog.php

 -  test_alert.php
 -  test_code.php
 -  test_db.php
 -  test_geo.php
-

5.3 So sánh với các công cụ khác

Công cụ	Phương thức hoạt động	Nhược điểm	Tính năng
Observium	<ul style="list-style-type: none"> - Mã nguồn mở, được viết dựa trên ngôn ngữ chính là PHP, CSDL chính là MySQL, hoạt động trên giao diện Web - Thu thập dữ liệu định kì thông qua Net-SNMP. - Hoạt động dựa trên nền tảng RRDTool (Round-Robin Database tool) - một nền tảng cho phép theo dõi hệ thống tuy nhiên lại không có giao diện đồ họa. 	<ul style="list-style-type: none"> - Phiên bản mã nguồn mở thì có ít tính năng giám sát nâng cao, các phiên bản nâng cao phải trả phí. - Chưa có giao diện mobile web 	<p>Khám phá thiết bị tự động – Một trong những tính năng nổi bật của Observium là khả năng tự động khám phá các thiết bị trên mạng của bạn (thông qua tập lệnh).</p> <p>Giám sát thời gian thực – Observium liên tục thu thập dữ liệu từ các thiết bị được giám sát, cung cấp thông tin chi tiết theo thời gian thực về hiệu suất mạng.</p> <p>Khả năng mở rộng – Môi trường trung tâm dữ liệu rất năng động và có thể phát triển nhanh chóng.</p> <p>Cảnh báo và Thông báo - Các quy tắc cảnh báo có thể tùy chỉnh có thể được định cấu hình để kích hoạt thông báo qua email, SMS hoặc các kênh liên lạc khác, đảm bảo rằng các vấn đề tiềm ẩn được giải quyết kịp thời.</p> <p>Dữ liệu lịch sử – Observium không</p>

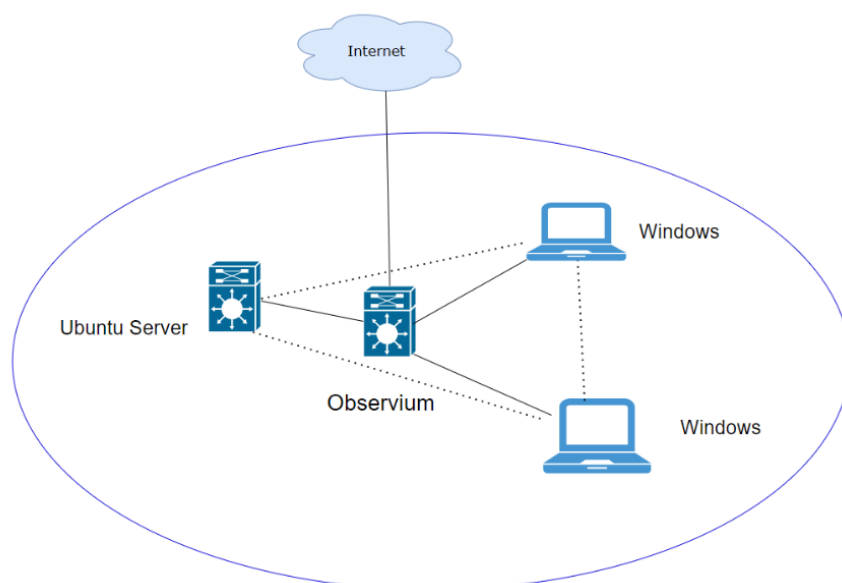
			<p>chỉ tập trung vào giám sát thời gian thực; nó cũng lưu trữ dữ liệu lịch sử.</p>
Zabbix	<p>Zabbix bao gồm các thành phần sau: Zabbix Server, Zabbix Proxy, Zabbix Agent và Web Interface.</p> <p>Zabbix Server sẽ chịu trách nhiệm cho các hoạt động kiểm tra dịch vụ mạng từ xa, thu thập thông tin, lưu trữ, hiển thị, cảnh báo,...</p> <p>Proxy là phần tùy chọn của Zabbix. Proxy có chức năng thu nhận dữ liệu, lưu trong bộ nhớ đệm và được chuyển đến Zabbix server.</p> <p>Zabbix agent là chương trình Zabbix dùng để cài đặt lên các máy chủ hoặc thiết bị phía Client. Từ đó, Zabbix Server hoặc Proxy có thể lấy các thông tin cần thiết từ Client</p> <p>Web Interface là một phần của Zabbix Server. Thông thường Web Interface được khởi chạy trên cùng một máy chủ vật lý nơi Zabbix Server đang chạy</p>	<p>Không có giao diện web mobile hỗ trợ.</p> <p>Không phù hợp với hệ thống mạng lớn hơn 1000+ node thiết bị client cần giám sát. Lúc này phát sinh vấn đề hiệu suất về PHP và Database.</p> <p>Thiết kế template/alerting rule đôi khi khá phức tạp.</p>	<ul style="list-style-type: none"> - Kiểm tra hiệu suất và tính khả dụng - Hỗ trợ theo dõi qua SNMP, IPMI, JMX, giám sát VMware - Tùy chỉnh kiểm tra hệ thống - Tùy chỉnh khoảng thời gian thu thập dữ liệu - Thực hiện với Zabbix server, Zabbix proxy và Zabbix agents - Giám sát trang web - Lưu trữ lịch sử dữ liệu - Và gồm nhiều tính năng vượt trội hơn các phần mềm giám sát mã nguồn mở khác.

Cacti	<ul style="list-style-type: none"> - Mã nguồn mở, được viết dựa trên ngôn ngữ chính là PHP, CSDL chính là MySQL, hoạt động trên giao diện Web - Thu thập dữ liệu định kì thông qua Net-SNMP. - Hoạt động dựa trên nền tảng RRDTool (Round-Robin Database tool) - một nền tảng cho phép theo dõi hệ thống tùy nhiên lại không có giao diện đồ họa. 	<p>Là phần mềm mã nguồn mở nên các chức năng và độ hoàn thiện không bằng các phần mềm trả phí doanh nghiệp</p>	<ul style="list-style-type: none"> - Thu thập dữ liệu: thu thập dữ liệu bằng cách sử dụng poller được lập lịch trong hệ điều hành. - Lưu trữ dữ liệu: Cacti sử dụng RRDTool để lưu trữ dữ liệu. - Biểu diễn dữ liệu: Một trong những tính năng được đánh giá cao nhất của RRDTool là chức năng vẽ đồ thị được tích hợp sẵn. Đồ thị có thể được thực hiện theo nhiều cách khác nhau. Có thể dễ dàng tùy chỉnh biểu đồ.
--------------	--	--	---

6. Triển khai thực nghiệm và kết quả

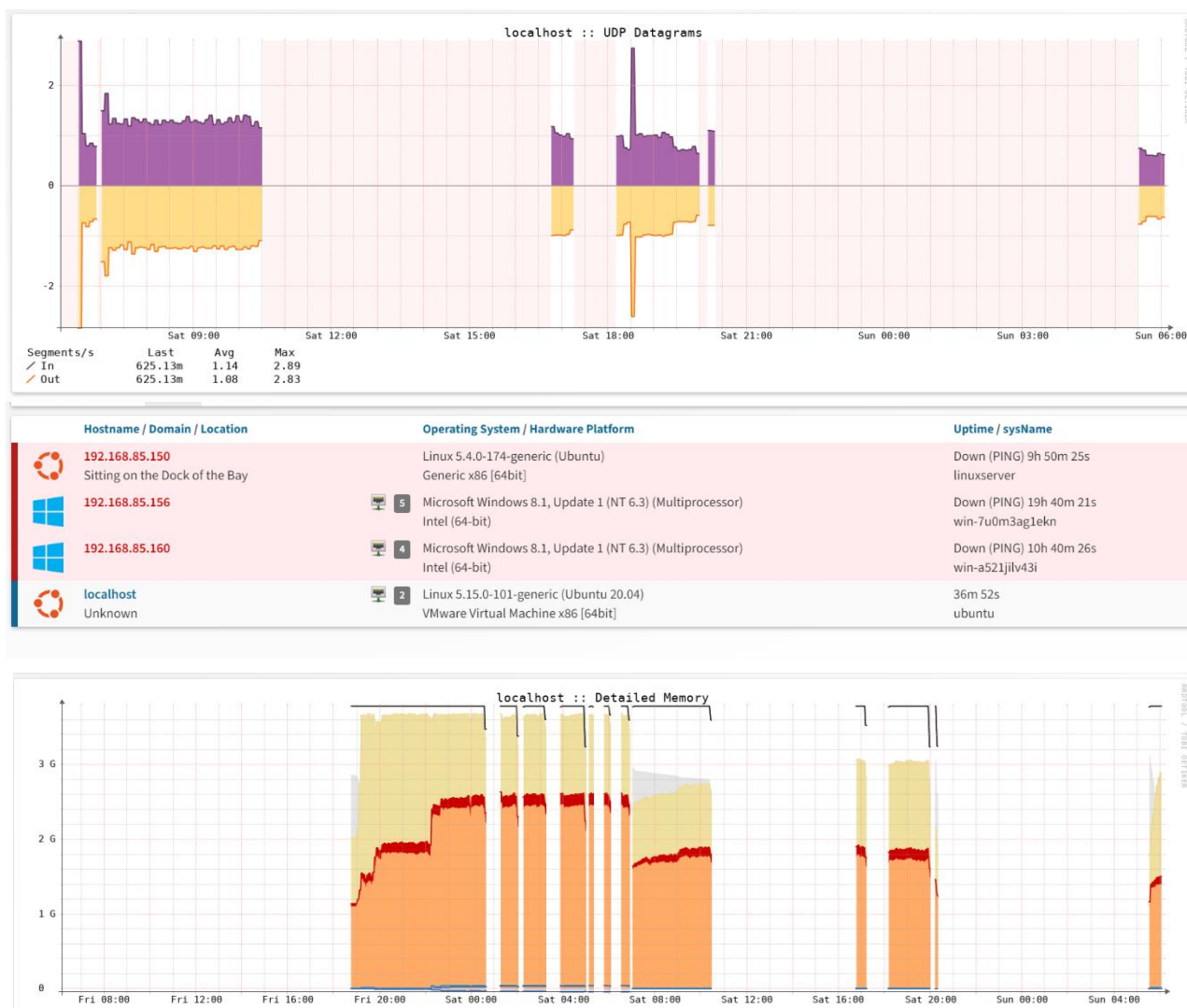
6.1 Mô hình triển khai

- Một máy chủ đóng vai trò là server giám sát, được cài công cụ Observium để giám sát các thiết bị khác.
- Các máy còn lại bao gồm máy trạm(workstation) Windows, máy chủ Ubuntu. Các máy nằm trong một mạng LAN và có khả năng đi ra ngoài mạng để có thể giám sát thông lượng.



6.2 Kết quả thực nghiệm

Một số kết quả sau quá trình cài đặt và giám sát thực nghiệm.



Demo chi tiết quá trình triển khai và kết quả: [Liên kết tới video demo](#)

7. Đánh giá, kết luận

7.1 Kết quả đạt được

Báo cáo đã nghiên cứu, triển khai và hoàn thành những vấn đề sau:

✚ Lý thuyết:

- Về vấn đề giám sát: Đi sâu phân tích về giám sát hệ thống và tầm quan trọng của việc giám sát hệ thống trong môi trường mạng.
- Về giao thức quản lý mạng: Trình bày về giao thức quản lý mạng đơn giản (Simple Network Management Protocol) bao gồm: khái niệm giao thức quản lý mạng, các thành phần trong giao thức quản lý mạng, và cách hoạt động của giao thức quản lý mạng.
- Nắm được phương pháp hoạt động và lấy thông tin hệ thống của phần mềm mã nguồn mở Observium.

✚ Thực nghiệm:

- Báo cáo đưa ra mô hình triển khai và trình bày toàn bộ các bước cấu hình các hệ thống giám sát theo mô hình triển khai đã đề ra.
- Về cơ bản đã khai thác được các chức năng chính của phần mềm mã nguồn mở Observium.

✚ Những kết quả đạt được:

- Có các kiến thức về giám sát hệ thống, các giao thức quản lý mạng.
- Triển khai thành công mô hình giám sát hệ thống bằng phần mềm mã nguồn mở Observium. Có thể cấu hình Router, Switch, ASA, IPS, Windows, Linux phục vụ cho quá trình giám sát.
- Tích lũy kinh nghiệm trong việc cấu hình các công nghệ trên.

7.2 Hướng phát triển

- Tích hợp các giải pháp giám sát khác như Snort (IPS, Firewall) và các plugins vào hệ thống giám sát đã có sẵn nhằm tối ưu hóa hệ thống Observium.
- Nâng cấp thiết bị để tăng cường phát hiện, xử lý các sự cố trên hệ thống.

C. TÀI LIỆU THAM KHẢO

1. <https://4sysops.com/archives/network-monitoring-with-the-open-source-tool-observium/>
2. <https://infosecmonkey.com/setting-up-alerts-in-observium-community-edition/>
3. <https://www.observium.org/>
4. [Step-by-Step Tutorial: Setting Up Observium Linux Agent](#)
5. [How to add a device to the Device Monitoring Software \(Observium\)](#)
6. <https://github.com/DanielleHuisman/observium-community-edition/tree/main/mibs>

Hết.