

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



Nguyễn Cao Bảo Long

**CÀI ĐẶT, PHÁT TRIỂN THÊM TÍNH NĂNG
VÀ TẠO NỘI DUNG CHO MỘT MÔI TRƯỜNG THI ĐẤU
TÂN CÔNG VÀ PHÒNG THỦ MẠNG**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
Ngành: Công nghệ thông tin**

HÀ NỘI – 2022

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

Nguyễn Cao Bảo Long

**CÀI ĐẶT, PHÁT TRIỂN THÊM TÍNH NĂNG
VÀ TẠO NỘI DUNG CHO MỘT MÔI TRƯỜNG THI ĐẤU
TÂN CÔNG VÀ PHÒNG THỦ MẠNG**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
Ngành: Công nghệ thông tin**

Cán bộ hướng dẫn: TS. Nguyễn Đại Thọ

HÀ NỘI – 2022

TÓM TẮT

Tóm tắt: Các cuộc thi Capture The Flag (CTF) dạng tấn công phòng thủ là một nền tảng giảng dạy các kiến thức về lập trình an toàn rất hiệu quả do các kiến thức và kinh nghiệm được truyền tải cho các thí sinh thông qua các ví dụ thực tế và có sự tương tác cao. Tuy nhiên tồn tạo hai rào cản lớn ngăn cản việc tổ chức rộng rãi của các cuộc thi CTF như vậy ở Việt Nam đó là:

- Thứ nhất, hệ thống tổ chức của cuộc thi yêu cầu rất nhiều tài nguyên hệ thống và kèm theo đó là yêu cầu rất lớn về nguồn lực để giám sát hệ thống nhằm đảm bảo tính ổn định.
- Thứ hai, phần lớn các thí sinh tham dự đều cảm thấy lối chơi của cuộc thi phức tạp, yêu cầu quá nhiều kiến thức và kỹ năng cùng một lúc khiến cho các thí sinh chưa có kinh nghiệm cảm thấy khó khăn trong việc bắt kịp với các thí sinh khác.

Khoa luận tập trung hướng đến giải quyết hai vấn đề này bằng cách áp dụng một hệ thống framework có sẵn, cụ thể ở đây là framework InCTF sử dụng công nghệ application container – được cải tiến từ framework iCTF, nhưng được sửa đổi nhằm giảm thiểu gánh nặng lên ban tổ chức và điều chỉnh luật chơi cho phù hợp với các cuộc thi trước đây trong nước và trên thế giới. Đồng thời tạo một số kịch bản và thử thách để có thể đưa vào tổ chức thi đấu một cách nhanh chóng.

Từ khóa: Tấn công phòng thủ, CTF, framework, application container.

LỜI CAM ĐOAN

Em xin cam đoan rằng đề tài Cài đặt, phát triển thêm tính năng và tạo nội dung cho một môi trường thi đấu tấn công và phòng thủ mạng là do em thực hiện dưới sự hướng dẫn của TS. Nguyễn Đại Thọ.

Các số liệu và kết quả được đưa ra trong khóa luận tốt nghiệp này là trung thực và không sao chép hay sử dụng công trình nghiên cứu của người khác mà không chỉ rõ về tài liệu tham khảo.

Em xin hoàn toàn chịu trách nhiệm trước bộ môn, khoa và nhà trường về lời cam đoan này.

Hà Nội, ngày....., tháng....., năm 2022.

Sinh viên

Nguyễn Cao Bảo Long

LỜI CẢM ƠN

Đầu tiên, em xin gửi lời cảm ơn chân thành đến quý Thầy, Cô, Cán bộ công tác tại trường Đại học Công Nghệ - Đại học Quốc Gia Hà Nội. Đặc biệt là các Thầy, Cô trong khoa Công nghệ thông tin đã tận tình chỉ dạy và trang bị cho em nhiều kiến thức, kỹ năng cần thiết không chỉ về kỹ thuật, mà còn về các khía cạnh khác trong đời sống.

Em xin trân trọng cảm ơn TS. Nguyễn Đại Thọ đã tận tình chỉ bảo, định hướng, giúp em có thể hoàn thành khóa luận tốt nghiệp của mình một cách tốt nhất.

Em cũng xin bày tỏ lòng biết ơn sâu sắc đến gia đình, người thân, bạn bè và đồng nghiệp đã giúp đỡ động viên em hoàn thành khóa luận tốt nghiệp này.

Xin gửi tới mọi người lời chúc tốt đẹp nhất !

MỤC LỤC

TÓM TẮT	1
LỜI CAM ĐOAN.....	2
LỜI CẢM ƠN	3
MỤC LỤC.....	4
DANH SÁCH THUẬT NGỮ VÀ TỪ VIẾT TẮT	6
MỞ ĐẦU.....	2
Chương 1. TỔNG QUAN VỀ THI ĐẤU CAPTURE THE FLAG	5
1.1. Khái niệm về thi đấu CTF	5
1.2. Các thể thức thi đấu CTF.....	5
1.3. Vai trò và lợi ích thi đấu CTF.....	5
1.4. Các cuộc thi CTF tiêu biểu trong nước và trên thế giới	6
1.4.1. Các cuộc thi CTF trên thế giới	6
1.4.2. Các cuộc thi CTF trong nước	7
Chương 2. HÌNH THỨC THI ĐẤU TẤN CÔNG VÀ PHÒNG THỦ	10
2.1. Tổng quan.....	10
2.2. Luật chơi và cách tính điểm	10
2.3. Các mô hình tổ chức thi đấu CTF hình thức tấn công phòng thủ.....	11
2.3.1. Mô hình phân tán	11
2.3.2. Mô hình tập trung.....	12
2.4. Nền tảng thi đấu CTF tấn công và phòng thủ iCTF	14
2.4.1. Thiết kế của dịch vụ	14
2.4.2. Giả lập các dịch vụ máy chủ	18
2.4.3. Cơ sở dữ liệu trung tâm.....	18
2.4.4. Hệ thống tính điểm.....	19

2.4.5. Triển khai mô hình	19
2.5. Nền tảng thi đấu CTF tấn công và phòng thủ InCTF	20
2.5.1. Ý tưởng.....	20
2.5.2. Docker	20
2.5.3. Các thành phần	21
2.5.4. Vận hành các dịch vụ giả lập	22
2.5.5. Vận hành các mã khai thác	23
2.6. Khái niệm vòng đấu.....	23
Chương 3. THỬ NGHIỆM, NÂNG CẤP VÀ XÂY DỰNG KỊCH BẢN THI ĐẤU CHO NỀN TẢNG INCTF	25
3.1. Thử nghiệm đánh giá hiệu năng.....	25
3.1.1. Lý do	25
3.1.2. Kịch bản và các thông số thử nghiệm.....	25
3.1.3. Các kết quả đánh giá.....	26
3.2. Cải tiến giao diện	28
3.2.1. Ván đề đặt ra	28
3.2.2. Yêu cầu chức năng	30
3.2.3. Yêu cầu phi chức năng.....	31
3.2.4. Thiết kế logic.....	32
3.2.5. Thiết kế vật lý	33
3.2.6. Kết quả thử nghiệm	35
3.3. Cải tiến lối chơi	39
3.3.1. Ván đề đặt ra	39
3.3.2. Phương án giải quyết	39
3.4. Cải tiến cách tính điểm.....	39
3.4.1. Ván đề đặt ra	39

3.4.2. Phương án giải quyết	40
3.5. Thiết kế kịch bản dịch vụ phục vụ thi đấu	42
3.5.1. Vấn đề đặt ra	42
3.5.2. Các kịch bản	44
Chương 4. KẾT LUẬN	51
4.1. Kết quả đạt được.....	51
4.2. Các vấn đề còn tồn đọng	51
TÀI LIỆU THAM KHẢO.....	52

DANH SÁCH THUẬT NGỮ VÀ TỪ VIẾT TẮT

Thuật ngữ/Từ viết tắt	Chú giải
API	Application programming interface / Giao diện lập trình trình ứng dụng.
Appication Container	Kĩ thuật đóng gói ứng dụng.
Attack-defense	Thể thức thi đấu CTF theo dạng các dịch vụ được kiểm soát bởi các đội chơi.
CTF	Capture The Flag, một hình thức thi đấu về bảo mật và an toàn thông tin.
Flag	Là một chuỗi kí tự bí mật, mục tiêu cần đánh cắp khi khai thác thành công một dịch vụ.
Framework	Khung phát triển phần mềm.

MỞ ĐẦU

Chỉ trong năm 2021, nền công nghiệp an ninh mạng thế giới đã chứng kiến hơn 28000 lỗ hổng bảo mật được báo cáo, nằm trải dài từ các hạng mục phần mềm như quản lý nội dung trang web, trình duyệt, hay chính bản thân hệ điều hành. Do vậy việc giảng dạy các lập trình viên về vấn đề lập trình an toàn là một ưu tiên hàng đầu nhằm giảm thiểu mất mát và giữ an toàn cho cả người sử dụng và các tổ chức.

Các cuộc thi về bảo mật cung cấp một mô hình học tập thông qua các thử thách với mục đích cung cấp các kiến thức về an toàn thông tin và lập trình an toàn. Có rất nhiều cuộc thi đã được tổ chức trong nước như WhitehatCTF, VibloCTF, ISITDTU CTF, Sinh viên với An toàn thông tin hoặc trên thế giới như FBCTF, GoogleCTF, HITBCTF với nội dung phong phú và lượng kiến thức truyền tải cho các thí sinh là rất tốt.

Chính vì những lý do đó mà Đại học Công nghệ - Đại học Quốc gia Hà Nội đã và đang đi đầu trong lĩnh vực giảng dạy, tổ chức thi đấu bộ môn an toàn thông tin và đã đạt được những thành tựu nhất định như đạt giải Nhất vòng sơ khảo khu vực miền Bắc cuộc thi Sinh viên với an toàn thông tin năm 2015, 2017, 2018 và 2019, đạt giải Nhất vòng thi chung khảo toàn quốc cuộc thi Sinh viên với an toàn thông tin năm 2016, 2019 và 2021. Từ kết quả này em nhận thấy được có sự thiếu ổn định trong các vòng thi chung khảo của đội trường Đại học Công nghệ, mà theo em là do khó khăn trong việc tập luyện, tổ chức thi đấu theo thể thức của vòng thi chung khảo, mà cụ thể ở đây là thể thức Attack-defense. Khác với thể thức jeopardy sinh viên có thể tập luyện riêng rẽ mang tính chất cá nhân mà không cần sự can thiệp hay hướng dẫn và quản lý chung từ nhà trường và các giáo viên, việc tập luyện theo thể thức Attack-defense chỉ có thể thực hiện được thông qua một môi trường phần mềm chung, chịu sự quản lý và điều phối từ Trường Đại học Công nghệ.

Để nâng cao thành tích của trường, cũng như chuẩn hóa các quy trình và tạo tiền đề cho Đại học Công nghệ có thể tổ chức các cuộc thi CTF từ nội bộ cho tới trong nước, đồng thời giải quyết vấn đề tổ chức trong điều kiện Nhà trường không có kinh phí và trang thiết bị chuyên dụng, em xin đề xuất một số sửa đổi cũng như đưa ra một số kịch bản thi đấu dựa trên một nền tảng nguồn mở miễn phí trong khi hạn chế tối đa sử dụng các tài nguyên tính toán.

Trước đây có rất nhiều nền tảng mã nguồn mở được công bố và sử dụng rộng rãi để tổ chức thi đấu CTF, tuy nhiên nổi bật hơn cả là framework iCTF – một framework đã được sử dụng từ làm cơ sở để phát triển các hệ thống thi CTF theo thể thức Attack-

defense sau này. Tuy vậy iCTF sử dụng kiến trúc phân tán, yêu cầu lượng kiến thức và kĩ năng quản lý cũng như tài nguyên vô cùng lớn và khó đáp ứng. Do đó em quyết định sử dụng InCTF, một framework dựa trên iCTF nhưng sử dụng kiến trúc tập trung, giúp thoả mãn rất nhiều tiêu chí của một framework phù hợp với bài toán được đặt ra: nguồn mở, miễn phí, phô biến, được hỗ trợ tốt để nâng cấp phát triển mở rộng, vừa tiêu tốn ít tài nguyên. Mặc dù vậy, InCTF cũng có hai điểm yếu được nêu ra trong bài báo [2] đó là thiếu đi khả năng đánh cắp và nghe lén các cuộc tấn công của đội chơi khác, đồng thời hệ thống quản lý cuộc thi bị đình trệ khi có quá nhiều container được đưa lên trong cùng một khoảng thời gian. Ngoài ra trong quá trình thử nghiệm framework InCTF, em còn nhận thấy một số hạn chế khác như giao diện chưa được thực sự dễ sử dụng, không toát lên được sự chuyên nghiệp của một cuộc thi CTF; hệ thống tính điểm còn đơn giản, thiếu tính chiến thuật lẫn cạnh tranh; không có khả năng lựa chọn mục tiêu tấn công khiến cho việc tối ưu hệ thống tính điểm gần như vô nghĩa.

Hai hạn chế được nêu ra trong bài báo [2] khó có thể được khắc phục do đây là hạn chế của kiến trúc tập trung, gánh nặng tính toán được dồn vào một điểm nằm ngoài framework đó là Docker daemon và khó có thể được sửa đổi nếu không xây dựng lại kiến trúc của hệ thống. Tuy vậy những hạn chế mà em thấy được trong quá trình sử dụng như giao diện, hệ thống tính điểm hay lựa chọn các mục tiêu tấn công đều tác động trực tiếp tới trải nghiệm thi đấu của các đội chơi.

Mục tiêu của khoá luận là giải quyết những vấn đề sau của framework InCTF:

- Giao diện mới đảm bảo tiêu chí đơn giản nhưng thông nhất, bố cục rõ ràng, đẹp mắt.
- Hệ thống tính điểm đề cao khả năng phân loại các đội chơi cùng với việc đánh giá đúng mức độ kiến thức cũng như kĩ năng của các đội.
- Chức năng lựa chọn mục tiêu nhằm nâng cao tính chiến thuật, tận dụng tốt việc thay đổi hệ thống tính điểm ở trên.

Bên cạnh đó để có thể thử nghiệm một cách tron tru tất cả các tính năng vừa cũ vừa mới của inCTF và biến inCTF thành một hệ thống thực tế, không chỉ dừng ở một concept, một mô hình em cũng tạo ra nội dung, tức các kịch bản để các đội chơi có thể đăng ký tham gia và thi đấu như trong các cuộc thi thật. Kịch bản do em tạo ra đáp ứng các tiêu chí sau:

- Mỗi nội dung thi đấu sẽ bao gồm 2 dịch vụ, một dịch vụ với số lượng lõi hổng lớn và một dịch vụ với số lượng ít hơn nhưng đặt nặng tính kĩ thuật và khả năng giải quyết vấn đề của các đội.
- Đảm bảo các dịch vụ hoạt động hiệu quả, đúng đắn và kết hợp tốt với hệ thống framework có sẵn.
- Kịch bản dễ sửa đổi, cập nhật, đóng vai trò như tiền đề để các cuộc thi sau này có thể dựa vào để xây dựng, cải tiến thêm.

Kết quả của khoá luận là một framework, được dựa theo nền tảng có sẵn là InCTF, nhưng được sửa đổi để phù hợp với mô hình thực tế tại trường Đại học Công nghệ và phù hợp với các tiêu chuẩn về thi đấu CTF trong nước hay quốc tế. Bên cạnh đó, em còn tạo một số kịch bản thi đấu để có thể tận dụng vào các cuộc thi nội bộ do trường tổ chức, và làm tiền đề cho các cuộc thi sau đó mà áp dụng hệ thống framework này.

Khóa luận được tổ chức thành 4 chương với nội dung như sau:

Chương 1. Tổng quan về thi đấu Capture the flag: Trình bày vấn đề và giải pháp cho bài toán, giới thiệu chung về ứng dụng.

Chương 2. Hình thức thi đấu tấn công phòng thủ: Đi sâu hơn vào thiết kế của hệ thống, triển khai dịch vụ và các thành phần.

Chương 3. Mô hình tập trung: Trình bày thiết kế kiến trúc hệ thống, công nghệ sử dụng và cải tiến của mô hình.

Chương 4. Cải tiến và kết quả của khoá luận: Trình bày những vấn đề, giải pháp và kết quả sau khi giải quyết được những vấn đề đó.

Kết luận: Trình bày kết quả đạt được của khóa luận và định hướng phát triển trong tương lai.

Xin trân trọng cảm ơn !

Chương 1. TỔNG QUAN VỀ THI ĐẤU CAPTURE THE FLAG

1.1. Khái niệm về thi đấu CTF

Với sự phát triển không ngừng của mạng Internet, thế giới đang ngày càng phụ thuộc vào các cơ sở hạ tầng mạng. Tin tặc và tội phạm mạng đang khai thác và lợi dụng hệ thống này nhằm tấn công vào những cơ quan, tổ chức lớn trên thế giới nơi hệ thống bảo mật còn kém. Chính vì lý do đó mà bảo đảm an ninh và quyền riêng tư của ứng dụng và các hệ thống đang là ưu tiên hàng đầu của các tổ chức, và đào tạo học sinh, sinh viên về bộ môn an ninh mạng kết hợp với các cuộc thi bảo mật đang là giải pháp được áp dụng trên toàn thế giới.

Capture the flag (hay được gọi tắt là CTF) được biết tới như là tập hợp các cuộc thi tiêu biểu nhất trong lĩnh vực thi đấu an ninh mạng. Trong cuộc thi CTF, các đội chơi cần phải giải quyết một số thử thách được đặt ra bởi ban tổ chức, hoặc bảo vệ hệ thống mà được ban tổ chức cung cấp bằng cách kiểm tra, giám sát và loại bỏ các lỗ hổng trong hệ thống, đồng thời tấn công các hệ thống của những đội chơi khác. Do các cuộc thi CTF thường được tổ chức bởi những chuyên gia bảo mật, chúng đang trở thành một mô hình tiêu biểu cho việc giảng dạy và nghiên cứu an ninh mạng.

1.2. Các thể thức thi đấu CTF

Hiện nay có hai hình thức thi CTF phổ biến nhất đó là Jeopardy và Attack-defense.

Thể thức Jeopardy sẽ cung cấp cho các đội chơi các thử thách nằm trải dài các mảng khác nhau trong an toàn thông tin như mật mã học, pháp y kĩ thuật số, bảo mật hệ thống và an ninh hệ thống ứng dụng web. Các đội chơi được tự do giải quyết các thử thách theo bất cứ trình tự nào và làm việc một cách độc lập với các đội chơi khác.

Khác với thể thức Jeopardy, thể thức Attack-defense sẽ cung cấp cho các đội chơi các máy ảo mà trong đó chứa các dịch vụ hay ứng dụng tồn tại một hoặc nhiều lỗ hổng, thường là có chủ đích. Mục tiêu chính của các đội chơi là tìm ra các lỗ hổng này, và chúng và sử dụng lỗ hổng đã được tìm thấy nhằm tấn công các đội chơi khác. Hình thức thi này được cho là cung cấp kiến thức tốt hơn do bản chất của cuộc thi có độ tương tác cao và trên hết là có tính đối kháng giúp kích thích các đội chơi hơn là thể thức thi Jeopardy.

1.3. Vai trò và lợi ích thi đấu CTF

Với mục đích đánh giá khả năng xử lý tình huống và giải quyết vấn đề, các cuộc thi CTF được tổ chức bằng cách kết hợp học tập sáng tạo với giải trí. Sự thiếu hụt nhân sự trầm trọng trong lĩnh vực an toàn thông tin đang thúc đẩy việc tổ chức các cuộc thi CTF, bởi các cuộc thi này là một nền tảng để học tập và thu thập kiến thức về an toàn thông tin vô cùng hữu ích.

Những cuộc thi an toàn thông tin giúp cho thí sinh được tiếp xúc trực tiếp với các tình huống thực tế, kết hợp với trải nghiệm thi đấu CTF sẽ giúp thí sinh thu thập được các kiến thức và kỹ năng vô cùng hữu dụng sau này. Tuy nhiên có rất nhiều thí sinh thiếu kiến thức và kỹ năng phù hợp để tham gia thi đấu an toàn thông tin. Chỉ có một số cuộc thi CTF tiêu biểu như InCTF và MIT LL có chương trình giảng dạy và đào tạo thí sinh trước khi tổ chức thi đấu. Với việc huấn luyện và học tập như vậy, các thí sinh có thể tiếp thu kiến thức về ngôn ngữ lập trình, các kiến thức về lỗ hổng của phần cứng cũng như phần mềm và cách khắc phục. Các thí sinh sẽ được huấn luyện về cách đảm bảo tính an toàn và bảo mật của các dịch vụ, hệ thống của cá nhân cũng như của các bên thứ ba.

Hầu hết các vụ tấn công bảo mật xảy ra do sự thiếu hiểu biết về an toàn thông tin. Cần nâng cao tầm quan trọng của việc giảng dạy về bảo mật cho học sinh, sinh viên tuy nhiên không cần tổ chức các cuộc thi quá cầu kì để giảm thiểu gánh nặng cho ban tổ chức. Mặc dù vậy việc tổ chức một cuộc thi bảo mật đáp ứng đủ các tiêu chuẩn trong nước và quốc tế vẫn tồn tại nhiều trở ngại, trong đó là trở ngại về thời gian và giới hạn kỹ thuật nỗi bật hơn cả.

1.4. Các cuộc thi CTF tiêu biểu trong nước và trên thế giới

1.4.1. Các cuộc thi CTF trên thế giới

Trên thế giới có rất nhiều cuộc thi CTF đã, đang và sẽ được tổ chức. Tiêu biểu nhất là DEF CON CTF, được tổ chức thường niên từ năm 1993 tới nay, đón nhận rất nhiều sự quan tâm của cộng đồng an toàn thông tin trên thế giới. Ngoài ra còn có cuộc thi UCSB iCTF, với thí sinh phần lớn là học sinh sinh viên, là một trong những cuộc thi CTF tập trung vào giáo dục và đào tạo lớn nhất.

Một số cuộc thi CTF tiêu biểu khác:

Hack.lu CTF	Vòng đầu sẽ thi đấu theo thể thức Jeopardy và vòng sau sẽ thi đấu theo thể thức attack-defense. Được tổ chức trực tuyến và diễn ra thường niên. Trang chủ của cuộc thi: https://www.hack.lu/ctf/
-------------	--

CSAW CTF	Vòng đầu sẽ thi đấu theo thể thức Jeopardy và vòng sau sẽ thi đấu theo thể thức attack-defense. Trang chủ của cuộc thi: https://www.csaw.io/ctf
PICO CTF	Thi đấu theo thể thức Jeopardy, được tổ chức trực tuyến và diễn ra thường niên. Các đội chơi trên toàn thế giới có thể đăng ký. Trang chủ của cuộc thi: https://picoctf.com/

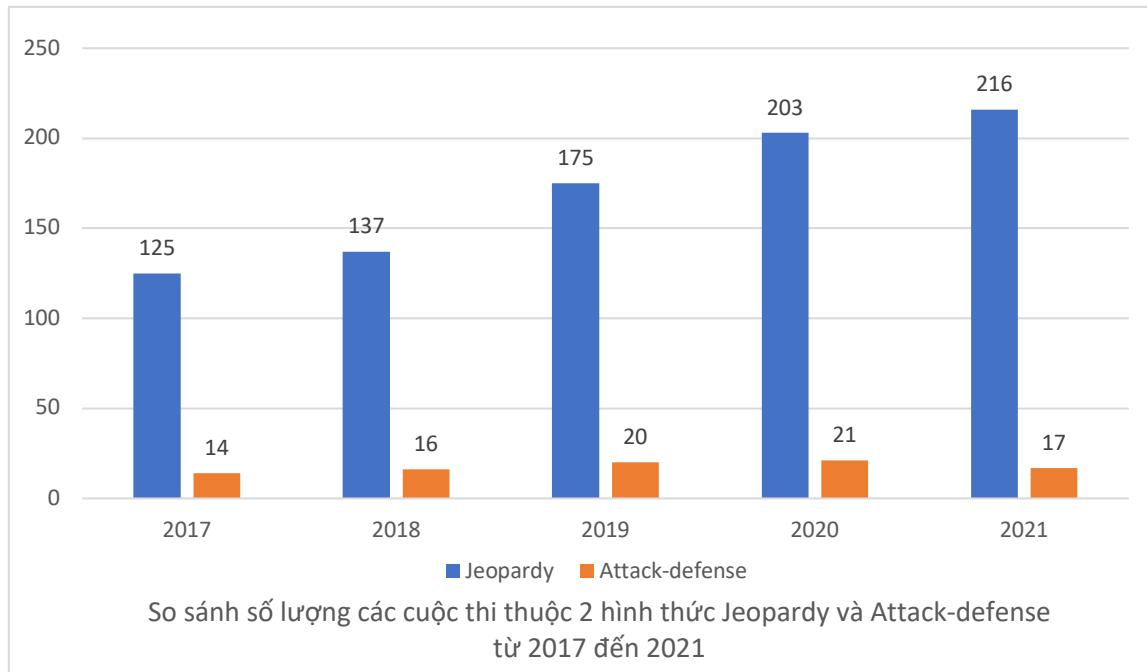
1.4.2. Các cuộc thi CTF trong nước

Bắt kịp xu thế tổ chức thi đấu bảo mật, rất nhiều đơn vị tại Việt Nam đã tổ chức các cuộc thi CTF, hội tụ và đáp ứng rất nhiều tiêu chuẩn của một cuộc thi tầm cỡ quốc tế. Một cuộc thi tiêu biểu, được tổ chức hàng năm dành cho đối tượng sinh viên cả nước, là cuộc thi Sinh viên với an toàn thông tin. Đại học Công nghệ, với truyền thống giảng dạy và học tập về bảo mật lâu đời, đã và đang đi đầu trong thành tích thi đấu của cuộc thi này và đã đạt được những thành tựu nhất định như đạt giải Nhất vòng sơ khảo khu vực miền Bắc cuộc thi Sinh viên với an toàn thông tin năm 2015, 2017, 2018 và 2019, đạt giải Nhất vòng thi chung khảo toàn quốc cuộc thi Sinh viên với an toàn thông tin năm 2016, 2019 và 2021.

Ngoài ra còn một số cuộc thi khác như cuộc thi Viblo CTF do Viblo tổ chức, kèm với hệ thống tập luyện và thi đấu thử theo hình thức Jeopardy; hay cuộc thi Whitehat Grand Prix quy tụ rất nhiều đội chơi trên toàn thế giới. Tuy vậy hai cuộc thi này đều thi theo thể thức Jeopardy, cho thấy được điều kiện về cơ sở hạ tầng lẫn khả năng tổ chức vẫn còn bị giới hạn rất nhiều.

Từ những điều trên trên thì em đã nhận thấy rằng thể thức thi Attack-defense được tổ chức rất ít so với thể thức Jeopardy do điều kiện cơ sở vật chất và rào cản địa lý cùng với số lượng đội chơi tham gia ít hơn rất nhiều so với thể thức Jeopardy. Đa số các cuộc thi dạng này đều được tổ chức dưới hình thức đối đầu trực tiếp chứ không phải trực tuyến – chỉ có khoảng 17 cuộc thi trực tuyến được tổ chức hàng năm trong các năm gần đây, ít

hơn rất nhiều so với số lượng cuộc thi theo thể thức Jeopardy.



Dựa vào trải nghiệm thực tế cùng với tìm hiểu các cuộc thi trước đây, em tin rằng có 2 lý do chính cho xu hướng này [1]:

- **Hệ thống và cơ sở hạ tầng phức tạp:** CTF hình thức Attack-defense yêu cầu một lượng tài nguyên lớn để đảm bảo các chức năng của cuộc thi hoạt động hiệu quả và mượt mà. Đây là lý do chính khiến cho các đơn vị tổ chức gặp khó khăn trong việc tổ chức thi, khi mà họ vừa phải thiết kế và chạy các dịch vụ của cuộc thi, vừa phải đảm bảo cho các tác vụ khác hoạt động đúng cách.
- **Yêu cầu các thí sinh có khả năng đa nhiệm cao:** Bên cạnh việc yêu cầu các thí sinh phải tìm lỗ hổng, khai thác và sửa chữa chúng thì hình thức thi này còn yêu cầu các thí sinh phải thực hiện một số tác vụ quản trị hệ thống khác. Công việc này bao gồm đảm bảo các dịch vụ của đội chơi và đường truyền tới chúng hoạt động ổn định, duy trì và quản lý các bản lưu trữ và theo dõi các kết nối tấn công tới dịch vụ của đội, v.v. Chính vì những điều này làm cho các đội chơi, nhất là những đội nhỏ và thiếu kinh nghiệm, lo ngại việc tham dự các cuộc thi theo thể thức này.

Nhóm tác giả của framework InCTF đã đưa ra một mô hình tấn công phòng thủ mới, sử dụng Docker thay thế cho các máy ảo. Việc sử dụng Docker giúp giảm thiểu gánh nặng

lên ban tổ chức do áp lực yêu cầu cơ sở hạ tầng và thiết kế hệ thống được giảm đi đáng kể. Hơn nữa, các thí sinh sẽ có thể tập trung vào tìm kiếm, sửa đổi và khai thác các lỗ hổng thay vì phải thực hiện các công việc quản trị hệ thống.

Chương 2.HÌNH THỨC THI ĐẤU TÂN CÔNG VÀ PHÒNG THỦ

2.1. Tổng quan

Trong các cuộc thi CTF với hình thức tấn công phòng thủ, các đội chơi được cung cấp các máy ảo giống hệt nhau và trong đó tồn tại các lỗ hổng phần mềm được tạo ra có chủ đích dưới dạng các dịch vụ mạng. Các đội sẽ phân tích các dịch vụ này, tìm kiếm các lỗ hổng và vá chúng, đồng thời sử dụng lỗ hổng đã tìm thấy để tấn công các đội chơi khác. Các cuộc thi CTF hình thức này giúp kiểm tra khả năng tấn công và phòng thủ trong lĩnh vực phần mềm cũng như quản trị hệ thống và bảo mật mạng. Chính vì những yêu cầu này mà các cuộc thi CTF dạng tấn công phòng thủ yêu cầu hệ thống cơ sở hạ tầng đặc biệt, khiến cho việc thiết kế, quản lý và duy trì gặp nhiều khó khăn. Càng nan giải hơn khi tổ chức các cuộc thi dưới dạng trực tuyến do các đội chơi và thí sinh nằm rải rác ở các vị trí địa lý khác nhau.

Các dịch vụ là các ứng dụng mạng, được lập trình và cung cấp bởi ban tổ chức. Những dịch vụ này có thể đơn giản như một hệ thống máy chủ chat trực tuyến, cung cấp không gian tương tác cho các máy con kết nối tới hay phức tạp hơn như dịch vụ của một ngân hàng hay mạng xã hội. Tất cả các dịch vụ này đều có khả năng chứa một lượng thông tin bí mật, ở đây cụ thể là flag, mà chỉ có thể đoạt được bằng cách khai thác, truy cập vào hệ thống (ví dụ như đánh cắp tài khoản và mật khẩu của quản trị viên). Các dịch vụ này đồng thời tồn tại một hoặc nhiều các lỗ hổng như SQLi, tràn bộ đệm hay sử dụng mã khoá yếu, v.v được đưa vào một cách có chủ đích. Với đặc điểm này nên các dịch vụ cần được thiết kế nhằm đảm bảo việc lấy flag chỉ có thể thực hiện được bằng cách cung cấp chính xác các thông tin mật hoặc khai thác được lỗ hổng trong ứng dụng. Do flag và các thông tin mật không thể bẻ khoá bằng cách bruteforce nên việc cung cấp flag của một đội chơi khác là minh chứng cho việc khai thác thành công dịch vụ của đội chơi đó.

2.2. Luật chơi và cách tính điểm

Các cuộc thi CTF dạng tấn công phòng thủ thường được tổ chức và kéo dài trong vòng 8 tới 12 tiếng. Trong khoảng thời gian đầu, không có flag nào được lưu trữ trong bất kì dịch vụ nào. Điều này tạo điều kiện cho các đội chơi có thời gian phân tích các dịch vụ và hệ thống được cung cấp, sửa chữa và khai thác các lỗ hổng mà đội tìm được trong giai đoạn sau của cuộc thi. Giai đoạn thứ hai là giai đoạn ghi điểm, gồm các vòng đấu có thời gian xấp xỉ bằng nhau. Bắt đầu mỗi vòng đấu thì một đoạn mã do ban tổ chức thực thi sẽ

lưu flag vào tất cả các dịch vụ và sẽ kiểm tra vào cuối mỗi vòng đấu. Đôi khi ban tổ chức sẽ tiến hành cập nhật một hay một số dịch vụ mà khi đó có thể sẽ thêm hoặc thay đổi các chức năng hiện có của chúng. Các đội chơi tiếp tục phân tích và giám sát các kết nối mạng để tìm kiếm các lỗ hổng mà họ có thể đã bỏ qua trong giai đoạn này. Có ba cách để một đội chơi có thể ghi điểm trong một vòng đấu đó là:

1. Điểm tấn công, được ghi nhận khi có thể đánh cắp flag của đội khác.
2. Điểm khả dụng, được ghi nhận khi giữ cho các dịch vụ của đội hoạt động và trực tuyến.
3. Điểm phòng thủ, được ghi nhận khi ngăn chặn thành công các đội chơi khác đánh cắp flag của đội mình.

Hệ thống luật chơi và ghi điểm mô phỏng lại thực tế một cách chặt chẽ:

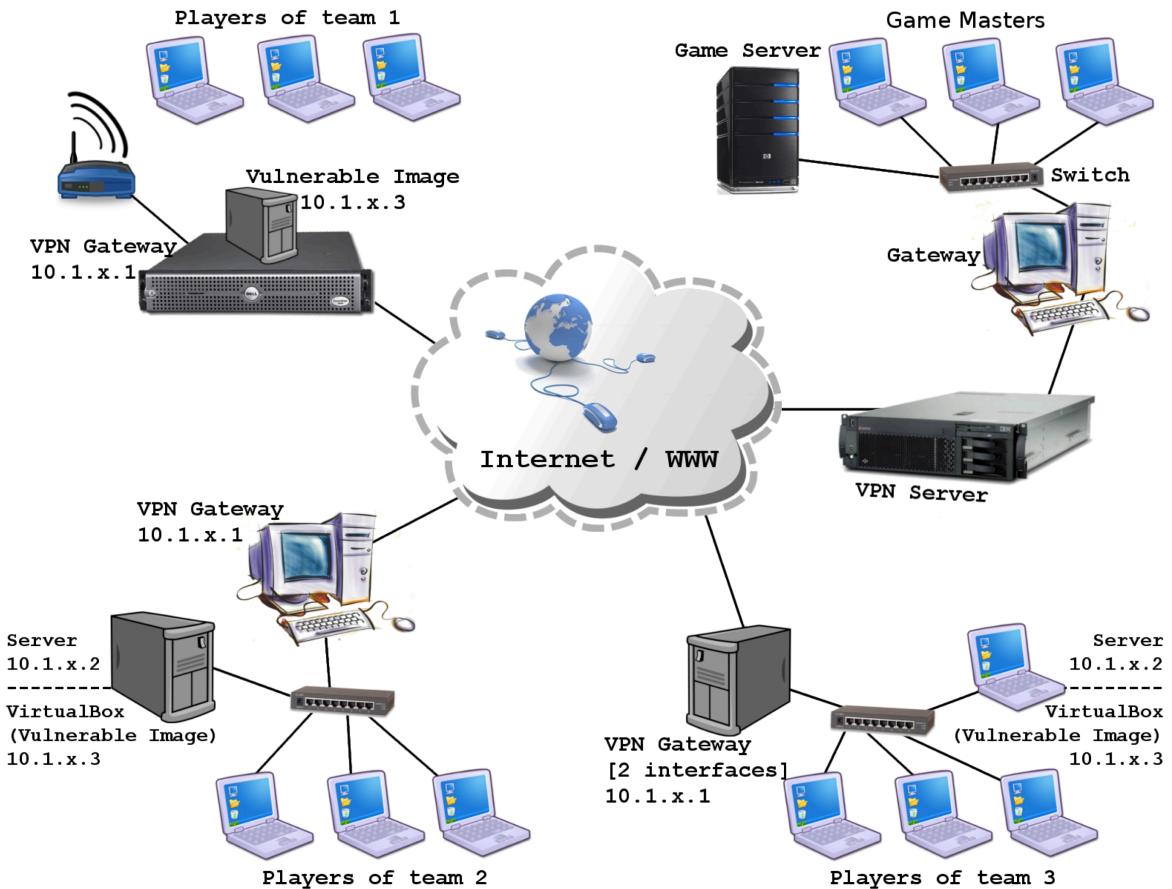
1. Yêu cầu của một dịch vụ là hoạt động đúng cách với các dữ liệu đầu vào hợp lệ và lọc bỏ, loại trừ các dữ liệu đầu vào độc hại mà không gây thất thoát thông tin.
2. Các dịch vụ có thể bị tấn công bất cứ lúc nào nên việc giám sát, kiểm tra thường xuyên là cần thiết.

Ngoài ra, việc bao gồm cả mảng tấn công trong các cuộc thi cũng giúp các thí sinh suy nghĩ theo lối của kẻ tấn công, đóng góp một phần không nhỏ trong việc thiết kế, đảm bảo và phát triển các giải pháp bảo mật cho các dịch vụ, dự án sau này của các đội chơi.

2.3. Các mô hình tổ chức thi đấu CTF hình thức tấn công phòng thủ

2.3.1. Mô hình phân tán

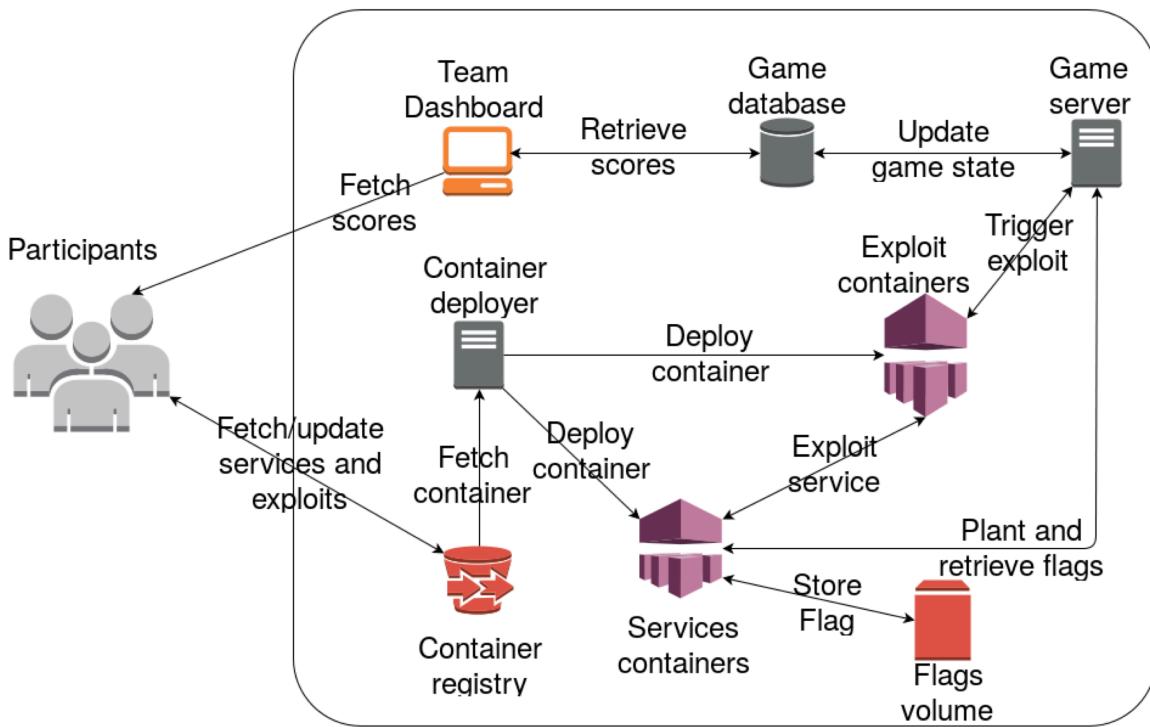
Đa phần các cuộc thi CTF tấn công phòng thủ sử dụng hình thức thi trực tuyến đều áp dụng mô hình này. Mô hình này sử dụng một tập hợp các mạng con mà mỗi mạng đó là một đội chơi, được kết nối tới hệ thống mạng của ban tổ chức thông qua hệ thống VPN. Do các mạng con đều có cấu trúc tương tự nhau cho nên các đội chơi có thể dễ dàng tìm kiếm và đánh cắp các flag từ các dịch vụ của đội chơi khác. Sau đó các đội chơi gửi flag lên hệ thống để lấy điểm, đồng thời theo dõi, giám sát hoạt động mạng trong mạng con của mình để phát hiện, ngăn chặn hành vi tấn công của các đội chơi khác.



Hình 1.1. Mô hình tổ chức thi đấu phân tán

2.3.2. Mô hình tập trung

Mô hình này được đề xuất bởi iCTF vào năm 2012, sau đó được sử dụng vào năm 2013. Cụ thể, ban tổ chức sẽ triển khai các máy ảo nằm trên hệ thống máy chủ của ban tổ chức, sau đó các đội sẽ kết nối tới các máy ảo này sử dụng SSH. Bằng cách này ban tổ chức có thể giảm thiểu được yêu cầu về mặt kĩ thuật khi tổ chức, đồng thời có thể quyết định các đội chơi có thể tự khai thác hay ban tổ chức sẽ nhận mã khai thác của đội chơi và thực thi nó. Những phần còn lại của mô hình đều tương tự với mô hình phân tán.



Hình 1.2. Mô hình tổ chức thi đấu tập trung

2.3.3. Những hạn chế của các mô hình hiện tại

Yêu cầu về tài nguyên, kĩ thuật cũng như giám sát của mô hình phân tán rất cao. Đồng thời, các đội chơi gặp nhiều vấn đề trong khâu chuẩn bị hệ thống mạng con do thiếu kinh nghiệm, phần cứng không đủ để đáp ứng các yêu cầu chạy nhiều máy ảo song song, thiếu trang thiết bị mạng và bị giới hạn bởi các chính sách bảo mật ở nơi mà các đội chơi tổ chức tham gia thi đấu.

Mô hình tập trung giảm thiểu gánh nặng về mặt thiết bị cũng như triển khai của các đội. Tuy vậy ban tổ chức sẽ là bên phải gánh vác số lượng máy ảo lớn hơn rất nhiều lần so với mô hình phân tán. Kèm theo đó là trở ngại về mặt địa lý khi các đội có vị trí địa lý xa so với vị trí đặt các máy chủ của ban tổ chức gấp phải vấn đề về độ trễ của kết nối mạng, điều mà có ảnh hưởng không nhỏ trong quá trình thi đấu.

Dù sử dụng mô hình nào đi chăng nữa thì các đội chơi thiếu kinh nghiệm đều gặp rất nhiều khó khăn trong thi đấu CTF dạng tấn công phòng thủ. Lý do chủ yếu là vì các đội chơi, ngoài việc phải tìm kiếm và khai thác các lỗ hổng trong dịch vụ, còn phải duy trì,

quản lý và theo dõi hệ thống mạng, trạng thái các dịch vụ, lưu trữ các bản lưu khẩn cấp và còn rất nhiều vấn đề khác. Nhiều đội không có kiến thức về những vấn đề này, cho nên gặp phải trải nghiệm không tốt khi dịch vụ của họ bị tấn công và ngừng hoạt động hoặc do các nguyên do khác. Tất cả các khó khăn này làm đội chơi bị phân tán, rời khỏi mục tiêu quan trọng nhất của cuộc thi đó là học tập và tìm hiểu các lỗ hổng phần mềm, đồng thời rèn luyện kỹ năng lập trình an toàn.

2.4. Nền tảng thi đấu CTF tấn công và phòng thủ iCTF

2.4.1. Thiết kế của dịch vụ

Mục tiêu của các đội chơi là đánh cắp được flag của đội khác và gửi lên hệ thống nhằm ghi điểm, đồng thời bảo vệ dịch vụ của đội mình. Các lỗ hổng tồn tại trong dịch vụ được viết một cách có logic, đồng thời một dịch vụ thường tồn tại nhiều lỗ hổng khiến cho việc một dịch vụ có khả năng bị tấn công là rất cao. Dịch vụ sẽ phải trải qua các bước kiểm tra để đảm bảo là các lỗ hổng có thể được khai thác bởi các đội chơi, tránh làm các đội gặp các trở ngại ngoài ý muốn.

Một dịch vụ có ba yêu cầu cơ bản như sau:

- Mỗi một vòng đấu thì flag sẽ được thay đổi, nhằm đảm bảo cho hệ thống có thể nhận biết được các tấn công thuộc các vòng đấu khác nhau.
- Đội phòng thủ không được thay đổi giá trị của flag mà bắt buộc phải và và bảo vệ dịch vụ của mình.
- Các đội chơi sẽ không được biết flag hiện tại của đội, tránh làm cho việc sử dụng hệ thống tường lửa một cách ngây thơ có thể bảo vệ thành công dịch vụ.

Để đạt được ba yêu cầu này, hệ thống sẽ được xây dựng như sau:

- Mỗi dịch vụ sẽ đi kèm thêm hai hệ thống là getflag và setflag. Hệ thống sẽ làm mới flag vào mỗi vòng đấu và kiểm tra flag trong quá trình thi đấu để đảm bảo các đội chơi không thay đổi flag.
- Hệ thống getflag và setflag sẽ được thiết kế để ẩn flag khỏi các đội, sử dụng những kĩ thuật khác nhau để các kết nối giống như các chức năng thông thường của ứng dụng.
- Một dịch vụ sẽ sử dụng một số bộ 3 tham số là flag_id, token và flag. Khi cần kiểm tra flag thì hệ thống getflag chỉ cần cung cấp flag_id và token là lấy được flag. Ví dụ như flag nằm trong một tài khoản người dùng, thì hệ thống getflag lúc này sẽ có tài khoản và mật khẩu (lần lượt tương ứng với flag_id và token) của

người dùng đó, còn các đội chơi phải khai thác một lỗ hổng nào đó để lấy được flag mà không biết mật khẩu.

```
2
3     class GetFlag():
4         def execute(self, ip, port, flag_id, token):
5             import socket
6             import random
7             import time
8             import datetime
9
10            flag = ''
11            error = -1
12            error_msg = ''
13
14            try:
15                s = socket.socket()
16                s.connect((ip, port))
17
18                msg = s.recv(1024)
19                s.send("1")
20                s.recv(1024)
21                s.send(flag_id)
22                s.recv(1024)
23                s.send(token)
24                msg = s.recv(1024)
25                flag = msg.split('\n')[-2]
26                error = 0
27
28            except Exception as e:
29                error = 42
30                error_msg = str(e)
31
32            self.flag = flag
33            self.error = error
34            self.error_msg = error_msg
35
36        def result(self):
37            return {'FLAG' : self.flag,
38                    'ERROR' : self.error,
39                    'ERROR_MSG' : self.error_msg,
40                    }
```

Hình 2.1. Ví dụ về nội dung của hệ thống getflag của một dịch vụ

```

13
14     def execute(self,ip,port,flag):
15
16         def set_flag(server_socket):
17             data = clientsock.recv(BUFSIZ)
18             globals()["flag_id"] = int(data.split(",")[0])
19             globals()["cookie"] = data.split(",")[1]
20             globals()["flag"] = data.split(",")[2]
21
22             error = -1
23             error_msg = ''
24             flag_id = datetime.datetime.fromtimestamp(random.randint(1,int(time.time())))
25                                         .strftime("%Y/%m/%d")
26
27             cookie = cities[random.randint(0,len(cities)-1)]
28
29             try:
30                 s = socket.socket()
31                 s.connect((ip,port))
32
33                 msg = s.recv(1024)
34                 s.send("2")
35                 msg = s.recv(1024)
36                 if msg.startswith("Insert a date"):
37                     s.send(flag_id)
38
39                 msg = s.recv(1024)
40                 if msg.startswith("Insert a location"):
41                     s.send(cookie)
42
43                 msg = s.recv(1024)
44                 if msg.startswith("Insert the recorded temperature:"):
45                     s.send(flag)
46
47
48             except Exception as e:
49                 error = 42
50                 error_msg = str(e)
51
52             self.flag_id = flag_id
53             self.cookie = cookie
54             self.error = error
55             self.error_msg = error_msg
56
57         def result(self):
58             return {'FLAG_ID' : self.flag_id,
59                     'TOKEN' : self.cookie,
60                     'ERROR' : self.error,
61                     'ERROR_MSG' : self.error_msg,
62                     }

```

Hình 2.2. Ví dụ về nội dung của hệ thống setflag của một dịch vụ

```

2
3  class Exploit():
4      def execute(self, ip, port, flag_id):
5          import socket
6          import cPickle
7          import base64
8          import marshal
9          import random
10         import time
11         import datetime
12
13         flag = ''
14         error = 0
15         error_msg = ''
16
17         try:
18             s = socket.socket()
19             s.connect((ip, port))
20
21             msg = s.recv(1024)
22             s.send("1")
23             msg = s.recv(1024)
24             s.send(flag_id)
25             msg = s.recv(1024)
26             s.send("lol|cat neverguess | grep \"%s\" | tail -n1"%flag_id)
27             msg = s.recv(1024)
28             flag = msg.split('\n')[-2]
29         except Exception as e:
30             error = 42
31             error_msg = str(e)
32
33         self.flag = flag
34         self.flag_id = flag_id
35         self.error = error
36         self.error_msg = error_msg
37
38     def result(self):
39         return {'FLAG' : self.flag,
40                 'ERROR' : self.error,
41                 'ERROR_MSG' : self.error_msg,
42             }

```

Hình 2.3. Ví dụ về nội dung của một chương trình khai thác dịch vụ

Nhằm giúp cho việc vận hành và kiểm tra các dịch vụ dễ dàng hơn thì một số thông tin sẽ được cấp vào dưới dạng JSON, bao gồm:

- Thông tin người viết dịch vụ.
- Địa điểm lưu các script của hệ thống getflag, setflag, hệ thống làm nhiễu thông tin,...
- Cổng kết nối được sử dụng (TCP hoặc UDP)
- ID của dịch vụ.
- Tên dịch vụ.
- Mô tả về dịch vụ.
- Mô tả về flag_id (Thông tin gì sẽ được sử dụng để định danh giá trị flag, cách mà thông tin đó được sử dụng bởi dịch vụ - như tên người dùng trong ví dụ ở mục 2.1.1.)

Bốn thông tin cuối sẽ được cung cấp cho cả thí sinh, giúp họ biết cách viết và triển khai mã khai thác sử dụng những thông tin nào, đồng thời cung cấp cho họ biết giá trị trả về nào sẽ là flag.

2.4.2. Giả lập các dịch vụ máy chủ

Để triển khai dịch vụ tới đội chơi thì có nhiều cách, trong đó có hai cách thường được sử dụng là một script cài đặt tự động hoặc sử dụng Debian package. Cách sử dụng script cài đặt thì có nhiều trở ngại, nhất là khi dịch vụ yêu cầu sử dụng các phần mở rộng không có sẵn trong hệ thống.

Đối với việc sử dụng Debian package thì lại có rất nhiều ưu điểm, phải kể đến tính ổn định, dễ dàng thêm các phần mở rộng, và có thể tự động khởi chạy dịch vụ. Bằng việc sử dụng Debian package, việc cài đặt và triển khai các dịch vụ sẽ được chuẩn hóa, đồng thời giúp ích cho ban tổ chức trong trường hợp cần cập nhật dịch vụ do phát hiện lỗi không có chủ đích vào thời điểm diễn ra cuộc thi.

Ngoài ra, việc sử dụng Debian package giúp ích cho việc phân quyền dịch vụ trong môi trường máy ảo. Mỗi dịch vụ sẽ có một người dùng với những quyền hạn nhất định, chỉ ảnh hưởng tới các tập tin của dịch vụ đó, giúp ngăn chặn việc cài cắm các backdoor của các đội chơi khác.

2.4.3. Cơ sở dữ liệu trung tâm

Cơ sở dữ liệu trung tâm dùng để quản lý tình trạng và giữ cho hoạt động của cuộc thi theo như kịch bản của ban tổ chức. Là trung tâm điều khiển nên tất cả các thành phần của cuộc thi đều lấy và ghi nhận tình trạng của cuộc thi vào hệ thống này.

Được chia làm hai phần chính, thứ nhất là API để các bộ phận khác của hệ thống tương tác, thứ hai là cơ sở dữ liệu, nơi lưu trữ các thông tin của cuộc thi. Hướng thiết kế này giúp phân tách cơ sở dữ liệu khỏi các phần khác của hệ thống, tránh việc xảy ra xung đột khi toàn bộ các thành phần đều có thể can thiệp trực tiếp vào cơ sở dữ liệu. Kèm theo đó, toàn bộ các thành phần đều cần log lại quá trình hoạt động vào cơ sở dữ liệu này để tiện cho quá trình phân tích, sửa lỗi và cải thiện hệ thống sau này.

2.4.4. Hệ thống tính điểm

Hệ thống tính điểm đảm nhiệm vai trò theo dõi và quản lý tình trạng của các dịch vụ bằng cách sử dụng hệ thống getflag và setflag ở phần 2.1.1. Hệ thống tính điểm sẽ dựa vào thiết lập của ban tổ chức, thường là vào đầu và cuối mỗi vòng đấu, để tạo ra một bộ ba tham số flag_id, token và flag cho mỗi dịch vụ. Các tham số này được sử dụng với các hệ thống setflag và getflag để tiến hành kiểm tra dịch vụ và ghi điểm cho các đội chơi.

Đồng thời hệ thống tính điểm cũng khởi chạy hệ thống làm nhiễu giúp ngăn cản các đội chơi có thể trích xuất flag từ các gói tin trong mạng, hay giả lập một số hành động tương tác với dịch vụ nhằm qua mặt các đội chơi.

Trạng thái của dịch vụ được xác định bằng kết quả của các hệ thống setflag, getflag và hệ thống giả lập hành động. Nếu giá trị trả về là thành công thì nghĩa là dịch vụ vẫn đang hoạt động. Nếu không có dữ liệu trả về thì dịch vụ được gắn nhãn là không hoạt động. Nếu tương tác với dịch vụ có dấu hiệu sai lệch (ví dụ như flag không đúng) thì hệ thống sẽ gắn nhãn cho dịch vụ này là hoạt động sai lệch.

Hệ thống setflag và getflag phải hoạt động theo một thứ tự định sẵn, không như hệ thống giả lập hành động có thể hoạt động một cách ngẫu nhiên. Hệ thống tính điểm sẽ tự động thêm bớt độ trễ của các hệ thống getflag và setflag, nhằm ẩn các gói tin này khỏi các đội và gây khó khăn cho các đội sử dụng phương pháp finger-print.

2.4.5. Triển khai mô hình

Mô hình có một số tùy chọn cho ban tổ chức dễ dàng triển khai hơn:

- Ban tổ chức có thể cung cấp kết nối SSH hoặc để các đội tự cài đặt máy ảo. Với cách làm cung cấp máy ảo cho các đội tự thiết lập thì sẽ giảm tải cho hệ thống, tăng khả năng phát triển và mở rộng hệ thống cho nhiều đội chơi với số lượng dịch vụ lớn hơn.
- Phân tán các dịch vụ của cuộc thi ra một hoặc nhiều máy ảo, giúp cho việc áp lực lên một hệ thống được san đều ra nhiều hệ thống hơn.

- Cấu trúc liên kết mạng cũng có thể được thay đổi, giữa việc sử dụng cấu trúc mặc định của Virtual Box là tạo các mạng LAN ảo, hoặc sử dụng chức năng sinh ra các máy ảo sử dụng chung một bridged networks nhằm giảm thiểu công đoạn chuẩn bị khi khởi tạo chúng trên các hệ thống host khác nhau.

Đơn giản nhất là cài đặt và khởi tạo toàn bộ các máy ảo trên hệ thống máy chủ của ban tổ chức, nhưng vấn đề lớn nhất đó chính là việc không thể mở rộng hay phát triển hệ thống cho nhiều đội chơi do tài nguyên của ban tổ chức có hạn.

2.5. Nền tảng thi đấu CTF tấn công và phòng thủ InCTF

2.5.1. Ý tưởng

Như đã đề cập ở phần 3.1.6, điểm yếu của mô hình này là việc khâu chuẩn bị rườm rà và đặt nặng yếu tố kĩ thuật, đồng thời hệ thống khó có thể triển khai và mở rộng thêm cho các cuộc thi lớn hơn.

Nhằm giải quyết các vấn đề của mô hình sử dụng máy ảo, framework sử dụng toàn bộ application container nhằm thay thế cho các máy ảo. Điều này khiến cho yêu cầu về tài nguyên được giảm đi đáng kể mà sẽ được trình bày ở phần sau của khoá luận.

Cách chơi của mô hình mô phỏng lại quá trình sửa lỗi trong việc phát triển phần mềm: Xác định vấn đề, sửa chữa và kiểm tra kĩ lưỡng sau đó commit các thay đổi hoặc đưa ra các bản vá. Chính vì lý do này mà các đội chơi được giảm bớt gánh nặng phải làm quen với công việc mới.Thêm vào đó, việc quản lý các phiên bản của image là một phần của quy trình làm việc, do đó các thí sinh không cần lo lắng về việc làm mất hay hỏng các dịch vụ và không thể đưa các dịch vụ về trạng thái hoạt động một cách nhanh chóng. Một ưu điểm khác đó là các dịch vụ có thể được chạy trong một môi trường tương tự với môi trường phân tích, giúp giảm thiểu độ trễ khi sử dụng mạng. Ngoài ra các thí sinh cũng không cần phải tự chạy các mã khai thác, không cần phải mua hay trang bị những thiết bị mạng đắt đỏ cũng như không phải chuẩn bị hệ thống mạng và VPN để kết nối tới cuộc thi. Ban tổ chức cũng dễ dàng để mở rộng hệ thống dựa vào các công cụ như Docker Swarm và Docker Cloud. Docker cũng có một cộng đồng người sử dụng vô cùng lớn và hoạt động sôi nổi, mà từ đó có thể sinh ra những công cụ mạnh mẽ giúp tối ưu việc cài đặt và triển khai của ban tổ chức trong tương lai.

2.5.2. Docker

Docker là một ứng dụng áp dụng kỹ thuật ảo hóa cấp độ hệ điều hành. Tương tự với ảo hóa cấp độ phần cứng, Docker cung cấp khả năng cô lập các tiến trình trong một môi trường riêng biệt được gọi là container, và hạn chế tài nguyên mà tiến trình đó có thể sử dụng. Các container này được tạo ra từ những container image, giống như cách mà các máy ảo được tạo ra từ các virtual disk image. Tuy nhiên, khác với ảo hóa phần cứng, Docker chỉ ảo hóa kernel, giúp giảm thiểu tài nguyên, cải thiện hiệu năng mà vẫn cung cấp đầy đủ các tính năng cô lập và bảo mật. Chính vì lý do này mà Docker có khả năng thay thế các hệ thống sử dụng máy ảo trong các cuộc thi CTF hình thức tấn công phòng thủ.

Đi kèm với Docker, framework sử dụng Gitlab với chức năng Container Registry để quản lý các bản vá và các đoạn mã khai thác được tải lên bởi các đội chơi. Bằng cách này việc lưu trữ, cập nhật và phân phát các Docker image trở nên đơn giản và dễ dàng hơn.

2.5.3. Các thành phần

Được cải tiến từ mô hình đề xuất bởi iCTF, có bốn thành phần mới được thêm vào để giúp cho mô hình có thể hoạt động với hệ thống Docker là: hệ thống quản lý container image, hệ thống lưu trữ các container của dịch vụ và mã khai thác, hệ thống lưu trữ và quản lý flag. Kèm theo đó hệ thống quản lý trò chơi đã được sửa đổi và thay thế nhằm cập nhật các image trên hệ thống theo thời gian các vòng đấu và sửa đổi thành phần vmcreator để tạo các container thay vì máy ảo, sửa đổi thành phần router do các đội chơi không được truy cập trực tiếp vào các container này. Các thành phần khác giống như các thành phần được iCTF đề xuất.

Hệ thống quản lý và phân phát container cung cấp một phương thức trao đổi và quản lý container dễ dàng bằng cách tối ưu hóa việc sử dụng Container Registry và Gitlab. Gitlab lưu trữ, quản lý và cung cấp các container image, đồng thời xác thực và quản lý quyền truy cập của các người chơi. Mỗi đội chơi được cung cấp một miền mà chỉ đội đó có quyền truy cập, giúp các đội chơi tải lên các container chứa mã khai thác và áp dụng các thay đổi với container gốc. Hệ thống gameserver sẽ đồng bộ các dịch vụ định kì vào đầu mỗi vòng đấu, phân tách nơi người chơi cập nhật các dịch vụ và nơi các dịch vụ này được khởi chạy giúp giải quyết vấn đề về độ trễ khi SSH.

Máy chủ nơi lưu trữ toàn bộ các container đồng thời là nơi khởi chạy các dịch vụ và các container chứa mã khai thác của các đội. Do giới hạn tài nguyên nên các dịch vụ này đều đang được triển khai trên cùng một máy chủ, tuy nhiên thiết kế của mô hình cho phép

triển khai trên nhiều máy chủ giúp cho việc cân bằng tải và chia sẻ gánh nặng xử lý tốt hơn. Kèm theo đó là việc ứng dụng các công cụ như Docker Swarm hay Docker Universal Control Plane giúp việc phát triển cơ sở hạ tầng lên nhiều hệ thống phần cứng dễ dàng hơn. Hệ thống quản lý container kết nối tới Docker thông qua API tồn tại ở một socket TCP, giúp cho việc khởi tạo, phá huỷ và khởi chạy dễ dàng. API này tách biệt với kết nối mạng bên ngoài và chỉ chấp nhận kết nối tới thông qua việc cung cấp mã bí mật, giúp giới hạn người có thể truy cập và trao đổi thông tin với API.

Để có thể đồng bộ các container với các image, hệ thống quản lý container sẽ thông báo tới hệ thống quản lý trò chơi khi nhận thấy có sự thay đổi trong các container. Vào vòng đấu mới, hệ thống sẽ xoá bỏ container cũ đi vào xây dựng lại container mới dựa vào image mới nhất. Do việc khởi tạo và xoá bỏ các container là một chu trình nhanh chóng, nên nó không gây ảnh hưởng quá lớn tới thời gian của các đội chơi. Có một cách khác đó là tạo một container mới mà không xoá container cũ đi, chỉ xoá sau khi container mới được tạo ra thành công giúp giảm thiểu tối đa tác động tới thời gian của các đội.

Một phản ứng phụ của việc tạo và xoá các container như thế là flag ở trong các dịch vụ cũng bị xoá bỏ. Việc này sẽ gây lỗi nếu như flag được đặt trong hệ cơ sở dữ liệu, khi mà nó yêu cầu khởi tạo lại sau mỗi lần xoá. Trong trường hợp đó Docker yêu cầu cung cấp một container riêng chỉ để chứa dữ liệu hoặc tạo ổ chứa dữ liệu để lưu trữ cơ sở dữ liệu.

2.5.4. Vận hành các dịch vụ giả lập

Tương tự với cách tiếp cận của iCTF, sử dụng một Debian image để đơn giản hoá việc tạo các dịch vụ. Sau đó các Docker image được tạo ra từ Debian image kèm với đó là việc sử dụng một ubuntu image nhỏ gọn. Toàn bộ quá trình đều được tự động hoá hoàn toàn bằng việc sử dụng một số file mẫu có sẵn có thể dễ dàng sửa đổi cho phù hợp với yêu cầu thực tế.

Để phân phát các image, đầu tiên các đội cần tạo tài khoản Gitlab trên hệ thống. Sau đó ban tổ chức sẽ đẩy các image này vào mỗi một miền để các đội có thể tải xuống. Hoặc ban tổ chức sẽ nén toàn bộ các image lại và cung cấp vào trước thời gian thi vì nghẽn mạng có thể xảy ra nếu số lượng đội chơi lớn đồng loạt yêu cầu tải về vào thời điểm diễn ra cuộc thi.

Tất cả dịch vụ đều được bind tới 1 port, mà sau đó được map tới 1 port của máy chủ. Cả hai port này đều có thể được cấu hình trong tệp tin cấu hình của hệ thống, đảm bảo việc

không có hai port nào trùng nhau. Vào thời điểm bắt đầu, tất cả các image đều được dán nhãn yêu cầu cập nhật, và từ đó hệ thống quản lý sẽ khởi chạy các image này. Quá trình này được tự động hoàn toàn và thời gian phụ thuộc vào số lượng image cũng như cấu hình và tốc độ xử lý của máy chủ.

2.5.5. Vận hành các mã khai thác

Tương tự như việc khởi chạy các dịch vụ, các container chứa mã khai thác có thể được dựng lên bằng bất cứ image nào, trên Docker hub hay là tự dựng từ đầu với yêu cầu duy nhất là sử dụng chung kiến trúc với hệ thống máy chủ. Đội chơi có thể tự do chọn phiên bản hệ điều hành, cài đặt môi trường với các thư viện tùy chọn, các công cụ, ngôn ngữ lập trình miễn sao nó thực thi câu lệnh khai thác khi container khởi chạy. Cải tiến này đáng giá so với iCTF khi ban tổ chức không cần cài đặt các công cụ chỉ để chạy được mã khai thác của các đội chơi. Sau khi hoàn tất việc khởi chạy container ở trên máy cá nhân, các đội chơi có thể tiến hành đẩy image lên hệ thống quản lý container. Khi đó hệ thống sẽ thông báo tới máy chủ quản lý trò chơi, lưu lại một chỉ mục yêu cầu cập nhật image vào vòng đấu tiếp theo.

Mục tiêu khai thác được truyền đi dưới dạng mảng JSON, thông qua giá trị biến môi trường TARGETS. Mỗi một giá trị của mảng bao gồm 3 cặp giá trị tương ứng với IP, cổng và mã định danh của flag. Sử dụng những giá trị này, các mã khai thác sẽ lấy tất cả flag ở các dịch vụ mà tấn công thành công để đưa ra stdout. Khi thành công thì flag sẽ được gửi lên hệ thống quản lý để ghi điểm cho đội chơi và container chứa mã khai thác sẽ bị phá huỷ. Việc phá huỷ là cần thiết vì chỉ có như vậy mục tiêu mới có thể được đưa vào biến môi trường của container. Mặc dù vậy thì việc ảnh hưởng tới thời gian của các đội chơi là rất ít do việc phá huỷ và khởi tạo lại container tiêu tốn ít thời gian.

2.6. Khái niệm vòng đấu

Một cuộc thi được chia ra làm nhiều vòng đấu, với lượng thời gian tương đương nhau và có thể thay đổi cho phù hợp với mong muốn của ban tổ chức. Mỗi vòng đấu sẽ bao gồm 5 giai đoạn, lần lượt theo thứ tự:

- Đồng bộ hoá container dịch vụ: Tất cả các dịch vụ mà được cập nhật sẽ được đồng bộ hoá với container trên server.

- Đồng bộ hoá container khai thác: Tất cả các container khai thác sẽ được đồng bộ hoá với container trên server.
- Lưu trữ flag: Hệ thống sẽ khởi tạo giá trị flag trong các container. Nếu không thể lưu trữ thì container sẽ được gắn nhãn là không hoạt động đúng cách.
- Khởi chạy mã khai thác: Hệ thống khởi chạy toàn bộ các đoạn mã khai thác tới tất cả các mục tiêu đang tồn tại và hoạt động. Đội chơi sẽ nhận được điểm với mỗi dịch vụ mà flag bị đội đó đánh cắp.
- Thu thập lại flag: Hệ thống kiểm tra và đánh giá trạng thái của dịch vụ dựa vào flag được lưu trữ từ trước.

Chương 3. THỬ NGHIỆM, NÂNG CẤP VÀ XÂY DỰNG KỊCH BẢN THI ĐẤU CHO NỀN TẢNG INCTF

3.1. Thử nghiệm đánh giá hiệu năng

3.1.1. Lý do

Do điều kiện thực tế ở trường Đại học Công nghệ không có thiết bị máy chủ chuyên nghiệp hiệu năng cao để đáp ứng những hệ thống CTF tấn công và phòng thủ thông thường do chúng đòi hỏi lượng tài nguyên tính toán và lưu trữ rất lớn nên cần phải thử nghiệm yêu cầu về hiệu năng của InCTF trên các thiết bị có khả năng tính toán yếu hơn hoặc trên các dịch vụ máy chủ đám mây.

3.1.2. Kịch bản và các thông số thử nghiệm

Hiệu năng của hệ thống được đánh giá thông qua việc so sánh lượng tài nguyên sử dụng bởi hệ thống áp dụng container và hệ thống sử dụng máy ảo trong 2 trường hợp:

- Thay đổi số lượng dịch vụ trong khi giữ nguyên số lượng đội tham gia.
- Thay đổi số lượng đội tham gia trong khi giữ nguyên số lượng dịch vụ.

Lượng tài nguyên sẽ được theo dõi trong 2 trường hợp ở trên, cụ thể:

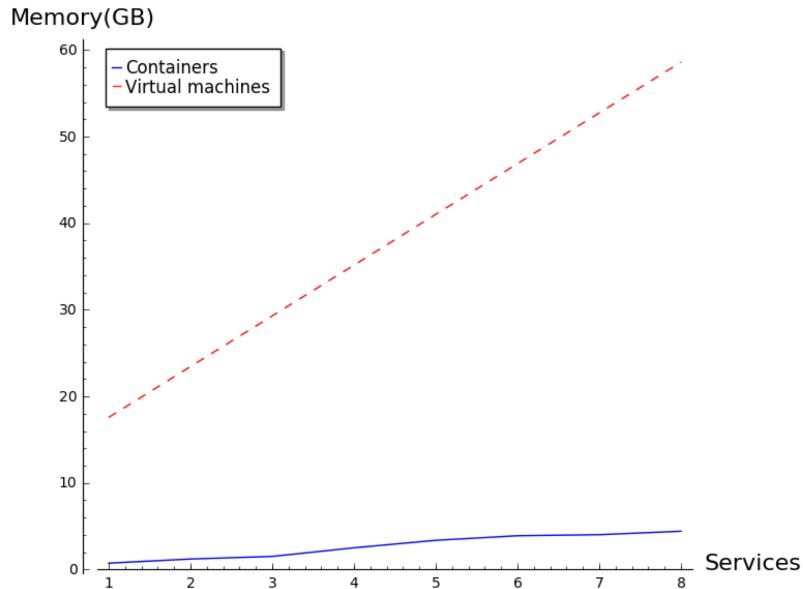
- Mô phỏng 8 cuộc thi với 3 dịch vụ cho mỗi đội. Bắt đầu bằng 5 đội chơi và thêm 5 đội chơi cho mỗi cuộc thi tiếp theo.
- Mô phỏng 8 cuộc thi với 30 đội trong mỗi cuộc thi. Bắt đầu bằng 1 dịch vụ cho mỗi đội và thêm một dịch vụ trong mỗi cuộc thi tiếp theo.

Tất cả các container được chạy trên hệ điều hành MacOS, sử dụng 16gb bộ nhớ Ram cùng với CPU M1 Pro chạy trên tiến trình armv8 với 10 nhân xử lý. Các thông số được lấy trong thời gian 1 vòng đấu, tương đương với 10 phút dựa theo thiết lập mặc định của framework.

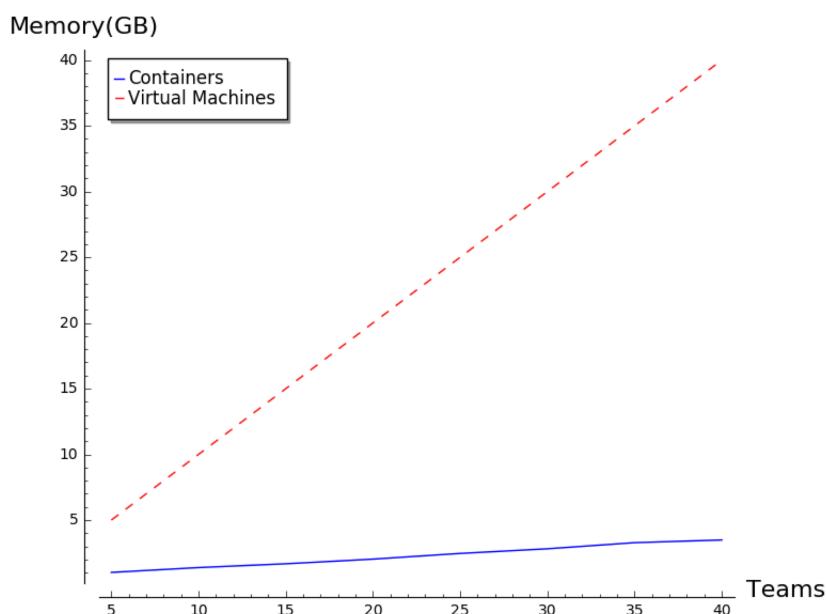
Do giới hạn về tài nguyên và kiến trúc armv8 của Apple chưa hỗ trợ máy ảo nên thông số sẽ được lấy từ dữ liệu của các cuộc thi khác đã sử dụng kiến trúc máy ảo là 1GB Ram và 2 nhân CPU cho các máy ảo, 200MB bộ nhớ cho 3 dịch vụ và phần còn lại là cho hệ điều hành đối với một cuộc thi có 3 dịch vụ. Trong các cuộc thi đó thì lượng tài nguyên như

này là vừa đủ để hệ thống hoạt động hiệu quả. Từ những thông số này chúng ta có thể dự đoán lượng tài nguyên yêu cầu theo kịch bản mô phỏng ở trong phần 2.4.1.

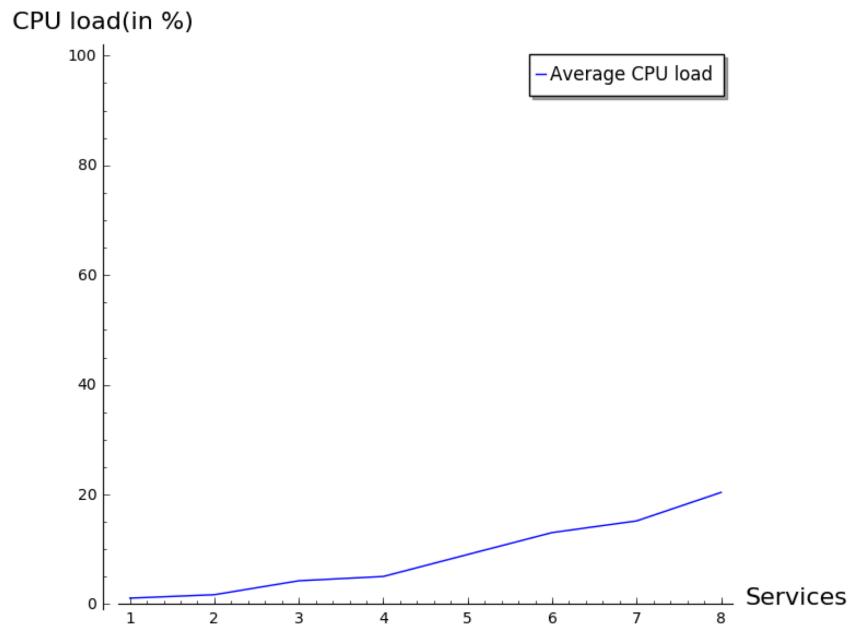
3.1.3. Các kết quả đánh giá



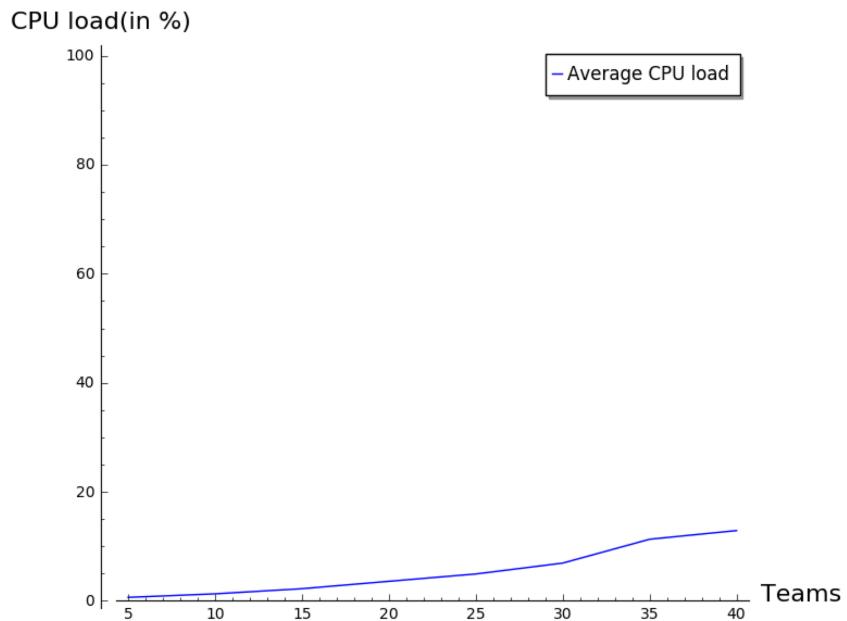
Biểu đồ 3.4. Lượng bộ nhớ sử dụng trong thử nghiệm mô phỏng số 1



Biểu đồ 3.5. Lượng bộ nhớ sử dụng trong thử nghiệm mô phỏng số 2



Biểu đồ 3.6. Lượng CPU sử dụng trong thử nghiệm mô phỏng số 1



Biểu đồ 3.7. Lượng CPU sử dụng trong thử nghiệm mô phỏng số 2

Dựa theo các biểu đồ 3.4, 3.5, 3.6 và 3.7, có thể thấy việc tổ chức một cuộc thi sử dụng framework áp dụng đóng gói ứng dụng là khả thi, đi kèm với lượng tài nguyên tiêu tốn là ít hơn rất nhiều so với framework sử dụng máy ảo. Tuy vậy kết quả chỉ mang tính tham khảo do Docker daemon bị quá tải với lượng container exploit khổng lồ cùng hoạt động một lúc. Có thể tối ưu framework bằng cách sử dụng nhiều Docker daemon hay thiết kế lại hệ thống khởi chạy các container khai thác, hoặc sử dụng nhiều máy chủ hơn thay vì chỉ một. Bằng cách này thì hệ thống máy chủ vừa có nhiều cổng hơn để cho nhiều dịch vụ và nhiều đội tham gia hơn, vừa có thể giảm gánh nặng lên Docker.

3.2. Cải tiến giao diện

3.2.1. Vấn đề đặt ra

Hiện tại giao diện người dùng của cuộc thi vô cùng đơn giản, không đem lại sự chuyên nghiệp hay cảm giác phấn khích khi các đội chơi giành được điểm.

Giao diện hiện tại của cuộc thi:

The screenshot shows a web-based interface for a CTF competition. At the top, there are navigation links: Show Services, Submit Flag, View Scoreboard, View status of exploits, and Pending updates. Below these, a timer displays "Current round will end in 04:23".

Challenge 1: driller (Port: 9091 - Status: Unknown)

In the awesome old days, everyone could freely setup their own uranium mine around the globe -- as long as they have enough money to bribe those governments. However, some great guys saw the great opportunity beneath, and they set up their own evil enterprise! Here it is, Global Drillers Enterprise Ltd. Their sweet dream, and your nightmare! With seamless cooperation with most countries, now you have to buy their service, select a spot and choose a driller, before you make any penny from uranium mines. To show their kind and mercy, you are offered an Autocratic version of their service, which is protected by the notorious cyber army named ***** (Access Denied). The binary is protected by some lame stuff. Anyways, you have no other choice. Get prepared, crack the service, and get their flags!

Flag identifier: Please use name of the spot as flag ID.

Challenge 2: poipoi (Port: 3335 - Status: Unknown)

Uranium mining is the process of extracting uranium ore from the ground. A prominent use of uranium from mining is as fuel for building your nuclear weapons and then terrifying your enemies. The POIPOI service collects information of all uranium mines in the world, so if you know of some new mines please insert related locations as soon as you can and help your country to dominate the world!

Flag identifier: the flag_id is the user_id returned by the server after successful registration.

Challenge 3: sillybox (Port: 4669 - Status: Unknown)

You enrich uranium. You need centrifuges. You need computers. Your computers need fans. Obvious solution is obvious. But now code runs on top of centrifuges, that's dangerous! So you put up this little silly sandbox that runs arbitrary code BUT allows reading files only if you know their password. But maybe some files are interesting... See the README for more info.

Flag identifier: The flag_id is the name of the file you have to read. It will contain the flag.

Challenge 4: tattletale (Port: 13007 - Status: Unknown)

You scored a couple contracting jobs at another team's intelligence agency. They have many secrets. You and your team must leak data to the public to achieve enough political impact to hinder their operations. You and your team must rendezvous with one of our agents on the inside. Show up for work and see if you can find a way to exfiltrate data!

Flag identifier: the flag_id for this service is the room you work

Challenge 5: temperature (Port: 56098 - Status: Unknown)

Hình 3.1. Giao diện mặc định của cuộc thi

Một số giao diện tham khảo của các cuộc thi CTF khác:

	ATTACKS	FS 0.00	CHESS 0.00	PBCSK7 2/round 17:50:00	FORVOLUTION 0.00	TIMECAPSULE 0.00	GENEALOGY 0.00	CELLS 0.00	AUTHME 0.00
1	Bushwhackers 10.60.1.0/24  192.643.91	<u>14 059.94</u> 2708 / -889 99% service disabled	0 0 service disabled	<u>57 448.02</u> 5958 / 94% service disabled	<u>4194.57</u> 1089 / -31 100% service disabled	<u>36 051.46</u> 5429 / -32 98% service disabled	0 / -72 100% service disabled	<u>35 347.4</u> 6087 / -420 99% service disabled	<u>8436.77</u> 2759 / -1634 99% service disabled
2	C4T BUT S4D EDU 10.60.8.0/24  137.274.11	<u>10 880.08</u> 5621 / -69 99% service disabled	<u>17 165.86</u> 3153 / -148 98% service disabled	0 / -407 48% down	<u>18 433.05</u> 4434 / -40 100% service disabled	<u>10 858.77</u> 2955 / -98 95% service disabled	<u>55 412.4</u> 8581 / -49 99% service disabled	0 / -1551 96% service disabled	<u>8883.53</u> 2758 / -1643 100% service disabled
3	RedRocket EDU 10.60.15.0/24  96.377.34	<u>11 214.83</u> 2518 / -70 98% service disabled	0 / -15 100% service disabled	<u>1701.88</u> 214 / -167 85% down	<u>29 008.55</u> 3729 / 0 100% service disabled	0 / -137 95% service disabled	<u>42 826.32</u> 7584 / -566 100% service disabled	0 / -822 51% service disabled	<u>9109.33</u> 2745 / -1484 100% service disabled
4	perfect blue 10.60.4.0/24  82.456.77	<u>4896.39</u> 1716 / -175 97% service disabled	<u>27 357.34</u> 4143 / 0 100% service disabled	0 / -449 54% down	0 / -46 100% service disabled	0 / -433 99% service disabled	0 / -56 100% service disabled	<u>41 018.03</u> 4926 / -261 100% service disabled	<u>2519.48</u> 2021 / -1616 100% service disabled
5	mhackeroni 10.60.3.0/24  68.466.21	<u>456</u> 563 / -946 88% service disabled	<u>54 895.55</u> 5478 / 0 100% service disabled	<u>1738.87</u> 168 / -6 74% down	0 / -361 71% service disabled	0 / -97 83% service disabled	<u>9.37</u> 3 / -26 78% service disabled	<u>215.32</u> 217 / -32 72% service disabled	<u>3068.22</u> 2177 / -1340 93% service disabled
6	organizers 10.60.12.0/24  67.074.18	0 / -2728 100% service disabled	0 / -17 100% service disabled	0 / -110 61% down	<u>1361.9</u> 1015 / -42 100% service disabled	<u>41 441.92</u> 4890 / 0 99% service disabled	0 / -1147 100% service disabled	<u>3460.32</u> 2142 / -298 99% service disabled	<u>1066.89</u> 1517 / -1585 97% service disabled
7	More Smoked Leet Chi... 10.60.2.0/24  54.687.3	<u>6924.52</u> 2786 / -103 100% service disabled	<u>731.69</u> 454 / -64 100% service disabled	<u>16 546.49</u> 1595 / -69 100% service disabled	0 / -31 84% down	<u>4513.78</u> 1094 / -40 100% service disabled	0 / -1148 99% service disabled	0 / -188 73% service disabled	<u>7593.73</u> 2434 / -1148 88% service disabled
8	Bulba Hackers ONLINE EDU 10.60.18.0/24  42.564.34	<u>6660.42</u> 2765 / -156 99% service disabled	0 / -26 100% service disabled	<u>223.69</u> 18 / -127 92% down	<u>19 869.15</u> 3996 / -49 100% service disabled	<u>12 106.95</u> 3206 / -387 99% service disabled	0 / -47 99% service disabled	0 / -403 57% service disabled	<u>2886.08</u> 2178 / -1492 100% service disabled
9	DiceGang 10.60.5.0/24  33.466.53	<u>3433.52</u> 1837 / -371 96% service disabled	0 / -0 100% service disabled	0 / -195 65% down	<u>5241.33</u> 1469 / -78 100% service disabled	0 / -1722 100% service disabled	0 / -1165 100% service disabled	<u>19 114.32</u> 3164 / -254 99% service disabled	0 / -1644 100% service disabled
10	Shellphish EDU 10.60.14.0/24  20.326.11	0 / -1508 99% service disabled	<u>19 532.22</u> 2941 / -17 100% service disabled	0 / -452 68% down	0 / -1471 100% service disabled	0 / -1701 100% service disabled	0 / -1134 100% service disabled	0 / -1440 93% service disabled	1 / 22 / -1567 100% service disabled

Hình 3.2. Giao diện bảng điểm của cuộc thi HITB+ CyberWeek PRO CTF 2021

CTFd
Users
Teams
Scoreboard
Challenges
Notifications
Team
Profile
Settings

Challenge
3 Solves
X

The Lost Park

50

What is the name of this monument?

Solve this challenge to unlock another one!

Multiple Choice



Forensics

The Lost Park



Programming

Squares



The answer is Major Mark Park

statue.jpg

Submit

Hình 3.3. Giao diện của CTFd

Cần cải tiến giao diện, làm cho bố cục dễ nhìn hơn, dễ dàng theo dõi tình trạng của dịch vụ mà không cần phải tải lại trang, hệ thống bảng điểm dễ nhìn và so sánh hơn.

3.2.2. Yêu cầu chức năng

Khoá luận sẽ liệt kê các chức năng của giao diện mới nhằm phục vụ đối tượng chính là các đội chơi và ban tổ chức. Các chức năng này sẽ được chia vào các nhóm chức năng, cùng với độ ưu tiên của chúng.

Bảng 3.1. Yêu cầu chức năng của ứng dụng

Chức năng	Mô tả	Độ ưu tiên
Chức năng chung		
Đăng nhập	Đội thi hoặc quản trị viên có thể đăng nhập vào ứng dụng bằng tài khoản và mật khẩu được ban tổ chức cung cấp.	Cao
Chức năng của đội thi		
Quản lý tình trạng của các dịch vụ	Các đội chơi có thể quan sát trạng thái của các dịch vụ, với thông tin như flag_id, tình trạng hoạt động.	Cao
Quản lý tình trạng của các mã khai thác	Các đội chơi có thể quan sát được trạng thái của các mã khai thác, với những thông tin như thành công, lỗi, flag trả về	Cao
Gửi flag	Các đội chơi có thể gửi flag lên hệ thống để kiểm tra nhằm ghi điểm cho đội của mình	Cao
Nhận thông báo của ban tổ chức	Các đội chơi có thể truy cập vào thông báo gửi đến từ ban tổ chức như thông báo lỗi dịch vụ, sự cố hay giải đáp các thắc mắc của các đội chơi.	Cao

Quan sát được thứ hạng của đội chơi	Các đội chơi có thể quan sát được thứ hạng của đội chơi, đồng thời trạng thái của các dịch vụ của đội địch và điểm hiện tại	Cao
Chức năng của quản trị viên		
Quan sát tình trạng hệ thống	Quản trị viên có thể quan sát được tình trạng của các dịch vụ và các thay đổi của hệ thống.	Cao
Quản lý các đội chơi	Quản trị viên có thêm, sửa, xoá các đội chơi và đặt lại mật khẩu cho các đội chơi.	Cao
Quản lý các dịch vụ	Quản trị viên có thể thêm, sửa, xoá các dịch vụ, thay đổi nội dung mô tả của từng dịch vụ.	Cao
Gửi thông báo tới các đội chơi	Quản trị viên có thể gửi thông báo gửi tới các đội chơi thông qua giao diện quản trị.	Cao
Kiểm tra kết nối tới hệ thống bằng điểm	Quản trị viên có thể kiểm tra, kiểm thử các chức năng của bảng điểm tương tác	Trung bình

3.2.3. Yêu cầu phi chức năng

a. Yêu cầu về giao diện:

Màu sắc đặc trưng của frontend là màu xám. Trong các giao diện (ngoại trừ giao diện đăng nhập) luôn có navbar để đội chơi có thể dễ dàng chuyển chức năng khi sử dụng.

b. Yêu cầu về thực thi

Sử dụng ngôn ngữ Vue.js để tối ưu tốc độ xử lý. Yêu cầu phải giao tiếp được với hệ thống API có sẵn của framework.

c. Yêu cầu về hiệu năng

Tốc độ phản hồi nhanh, hiển thị thông báo từ ban tổ chức và gửi flag lên hệ thống có độ trễ tối đa là 3s. Phải chịu được lượng kết nối lớn do trung bình một cuộc thi sẽ có khoảng 100 người dùng tại cùng một thời điểm.

d. Yêu cầu về bảo mật

Bảo đảm tính bảo mật không để lộ flag giữa các đội, đồng thời yêu cầu các cơ chế bảo vệ với giao diện của ban tổ chức.

3.2.4. Thiết kế logic



Biểu đồ 3.1. Biểu đồ Usecase tổng quát

Các tác nhân của hệ thống:

- Người dùng: Người dùng của hệ thống sẽ bao gồm các thành viên của đội thi và quản trị viên hay ban tổ chức. Người dùng hệ thống chỉ có thể đăng nhập bằng tài khoản và mật khẩu cho trước, nếu làm mất mật khẩu thì có thể yêu cầu quản trị viên cấp lại.

- Đội thi: Tập hợp những người dùng sau khi đăng nhập thành công qua cổng đăng nhập chung được gọi là đội thi và có quyền sử dụng những chức năng như gửi flag, xem danh sách và thứ tự các đội thi, xem thông tin chi tiết về các dịch vụ của cuộc thi.
- Quản trị viên: Người dùng sau khi đăng nhập qua giao diện quản trị thì được gọi là quản trị viên. Quản trị hệ thống có nhiệm vụ thông báo những thay đổi của dịch vụ cho toàn bộ các đội chơi, đặt lại mật khẩu của đội chơi, thông báo và giải đáp thắc mắc mà các đội chơi gặp phải trong quá trình thi đấu.

3.2.5. Thiết kế vật lý

Do yêu cầu xử lý một lượng truy cập lớn, đồng thời yêu cầu tốc độ phản hồi nhanh nên em quyết định sử dụng Vue.js để thiết kế frontend. Vue.js là một framework Javascript được tạo bởi Evan You, giúp chúng ta xây dựng giao diện người dùng cũng như xây dựng Single Page Application thân thiện với người dùng, chúng xây dựng từ các thư viện, cách triển khai component, các chức năng đặc trưng của nó như SFC (Single File Component) [1] Tham khảo các giao diện trên mạng có cùng sử dụng ngôn ngữ này thì có một giao diện thiết kế riêng cho các cuộc thi CTF là Cardinal, rất phù hợp để làm giao diện cho các cuộc thi dạng tấn công phòng thủ. [6]

Để giao diện có thể liên lạc với hệ thống backend phức tạp thì cần có sự giúp đỡ của API. Giao diện lập trình ứng dụng (API) là một sự kết nối giữa các thành phần của hệ thống với nhau. API được sử dụng ở hệ thống này với mục đích giúp máy chủ và hệ thống frontend liên lạc được với nhau. Swagger được lựa chọn để làm tài liệu mô tả API cho hệ thống. Bằng việc sử dụng Swagger, lập trình viên có thể gọi và kiểm thử các API ngay trên trình duyệt web mà không cần phải cài đặt thêm phần mềm nào khác. Ngoài ra, Swagger cho phép việc giả lập API, tính năng này giảm bớt sự phụ thuộc giữa các thành viên khi phải triển khai chức năng có sự liên kết giữa các thành phần. Tại phần này, khóa luận sẽ liệt kê danh sách API của hệ thống và mô tả những API chính mà frontend sử dụng.

Nhóm API xác thực tài khoản: Nhóm API này phục vụ mục đích xác thực tài khoản của người dùng. Do các tài khoản đều được quản trị viên cấp nên chỉ có duy nhất API login:

POST /login User login

Hình 3.4. Nhóm API xác thực tài khoản

Nhóm API quản lý dịch vụ: Nhóm API này phục vụ mục đích quản lý các dịch vụ chạy trong cuộc thi.

GET /state Get current state of the game

GET /getgameinfo Get Current Game Info (teams and services)

GET /getservicesstate Get the Service State

GET /setservicesstate Set a Service's State (up, up_non_functional or down)

Hình 3.5. Nhóm API quản lý dịch vụ

Nhóm API quản lý script: Nhóm API này phục vụ mục đích quản lý các script được thực thi ví dụ như setflag, getflag, các script khai thác của các đội.

GET /allscripts Get all scripts

GET /script Get specific script payload

GET /ranscript Result of running a script

Hình 3.6. Nhóm API quản lý các script

Nhóm API quản lý flag: Nhóm API này phục vụ mục đích quản lý các flag của các dịch vụ, cũng như quản lý các flag được gửi lên hệ thống.

GET /setcookieandflagid Set cookie and flag id

GET /getlatestflagandcookie Get current flag, flag_id, cookie

GET /getlatestflagids Get current flag_id for all teams and services

GET /submitflag Submit a flag from a team

Hình 3.7. Nhóm API quản lý các flag

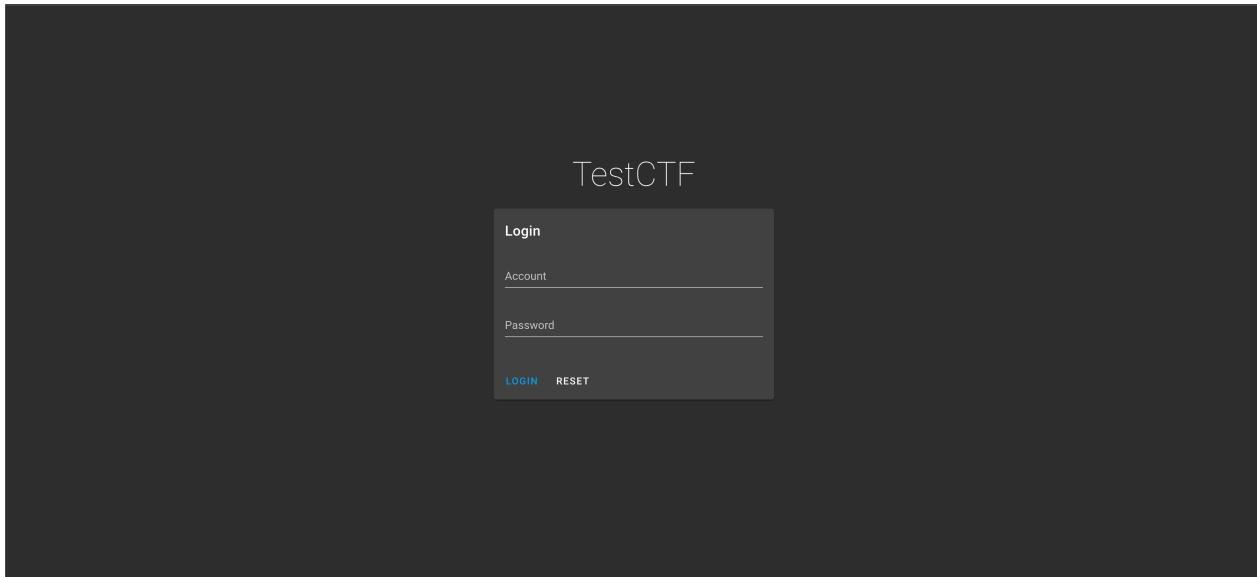
Nhóm API điều khiển trò chơi: Nhóm API này phục vụ mục đích điều khiển trò chơi như thời gian mỗi vòng, cập nhật container, lưu lại các lượt tấn công của các đội chơi.

GET	/services	Get service details	▼
POST	/container_changed	Mark container as requiring update	▼
GET	/ranexploit	Updating status of exploit	▼
GET	/exploitlogs	Exploit logs	▼  
GET	/changed_containers	List of changed containers	▼
GET	/tick_duration	Time to next tick	▼

Hình 3.8. Nhóm API điều khiển trò chơi

3.2.6. Kết quả thử nghiệm

Trang đăng nhập của các đội thi, tài khoản và mật khẩu sẽ được ban tổ chức sinh ngẫu nhiên và cấp cho các đội vào thời điểm diễn ra cuộc thi



Hình 3.9. Giao diện đăng nhập của đội thi

Giao diện chính của cuộc thi, nơi cập nhật trạng thái của các dịch vụ, điểm và thứ tự của đội chơi, log của các hành động mà đội chơi đã thực hiện và nơi gửi flag lên hệ thống.

The screenshot shows the main interface of the competition. At the top, there are navigation tabs: TestCTF, GAMEBOX STATUS (which is selected), RANK, BULLETIN, and LOGOUT. Below the tabs, it says "To Round 18: 0 Minute 33 Second".

Left Panel (Scoreboard):

- hihihoho (Token: 14bd2a75c23dd7cc8a0cfad6e7a643d1)
 - driller: 134.209.108.85.9091, 1000.00 (Online)
 - poipoi: 134.209.108.85.3335, 1000.00 (Online)
 - sillybox: 134.209.108.85.4669, 1000.00 (Online)
 - tattletale: 134.209.108.85.13007, 1000.00 (Online)
 - temperature: 134.209.108.85.56098, 1000.00 (Online)
- #1 / 5000.00

Right Panel (Live Log):

No Data

Bottom Panel (Flag Submission):

Submit Flag Please input your flag here... SUBMIT

POST /flag

Header

Content-Type: application/json
Authorization: 14bd2a75c23dd7cc8a0cfad6e7a643d1

Body

```
{"flag": "your_flag_here"}
```

curl -X POST http://134.209.108.85:19999/api/flag -H 'Authorization: 14bd2a75c23dd7cc8a0cfad6e7a643d1' -d '{"flag": "your_flag_here"}'

Hình 3.10. Giao diện chính của cuộc thi

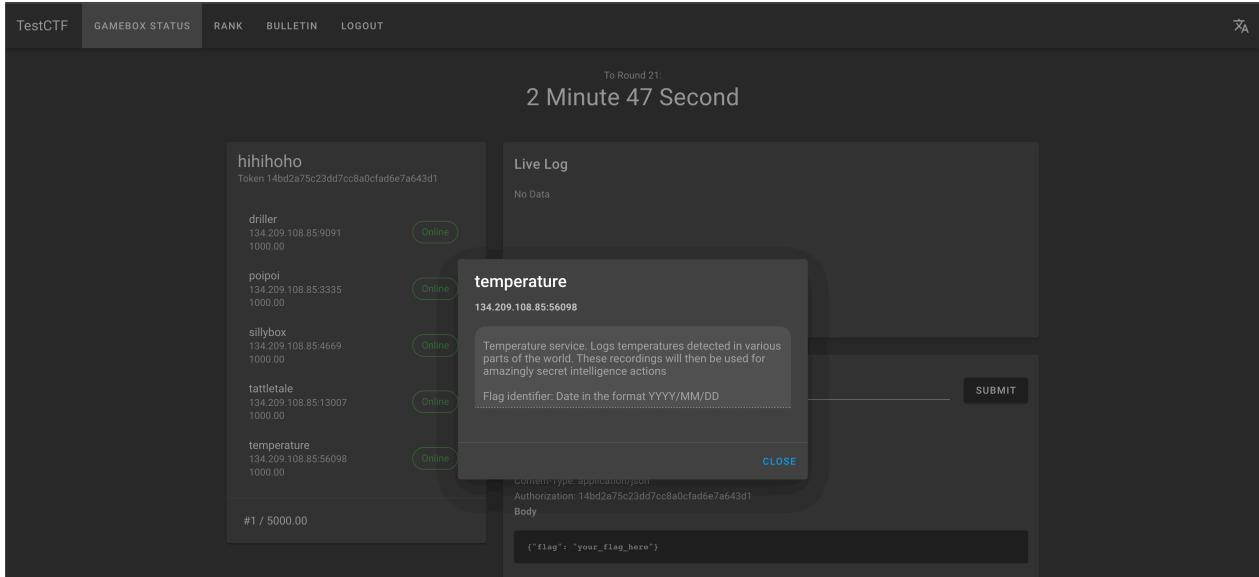
Giao diện bảng điểm của cuộc thi, nơi tất cả các đội đều có thể truy cập và xem trạng thái dịch vụ của các đội chơi khác, cùng với số điểm và xếp hạng của các đội.

The screenshot shows the scoreboard table. At the top, there are navigation tabs: TestCTF, GAMEBOX STATUS, RANK (selected), BULLETIN, and LOGOUT. The table has columns: #, Team, Score, driller, poipoi, sillybox, tattletale, and temperature.

#	Team	Score	driller	poipoi	sillybox	tattletale	temperature
1	hihihoho	5000.00	✓	✓	✓	✓	✓
2	Imaobaka	0.00					

Hình 3.11. Giao diện bảng điểm của cuộc thi

Khi chọn một dịch vụ ở giao diện chính, cửa sổ thông tin chi tiết của dịch vụ hiện lên, cung cấp mô tả về dịch vụ, flag_id của dịch vụ này và một số thông tin khác.



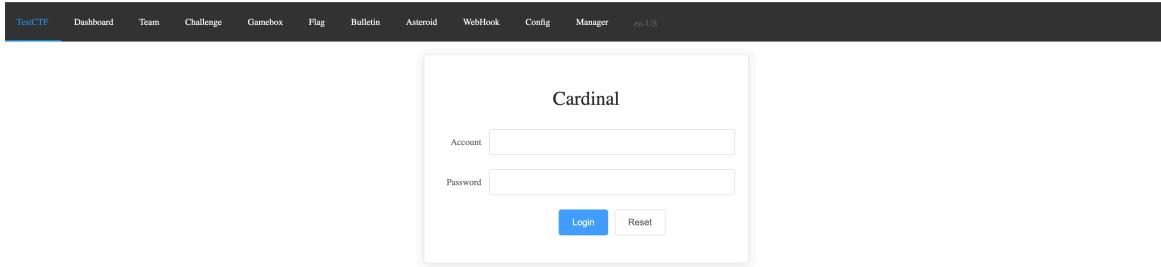
Hình 3.12. Giao diện thông tin dịch vụ của các đội chơi

Giao diện bảng điểm và tấn công trực tiếp giữa các đội được viết bằng Unity. Khi một đội đánh cắp được flag của đội khác, giao diện sẽ thể hiện bằng hoạt ảnh ngẫu nhiên.



Hình 3.13. Giao diện bảng điểm và tương tác giữa các đội

Giao diện đăng nhập của quản trị viên, nơi chứa các chức năng như quản lý dịch vụ, quản lý đội chơi, quản lý các tài khoản người dùng...



Hình 3.14. Giao diện đăng nhập của quản trị viên

1 Minute 33 Second to Round 129

#	Team	Score	driller	polpol	sillybox	tattletale	temperature
1	hihihoho	5000.00	0	0	5.1 MiB	56	

Flags Submitted: 0 | CheckDown: 0 | Memory Allocated: 5.1 MiB | Goroutines: 56

```

[7/28/2022, 11:00:01 PM] [WARNING] Round 128 Score Calculation Done!The process took 0.044201581 s.
[7/28/2022, 11:45:01 PM] [WARNING] Round 127 Score Calculation Done!The process took 0.042438956 s.
[7/28/2022, 11:40:01 PM] [WARNING] Round 126 Score Calculation Done!The process took 0.053401638 s.
[7/28/2022, 11:35:02 PM] [WARNING] Round 125 Score Calculation Done!The process took 0.060212675 s.
[7/28/2022, 11:30:02 PM] [WARNING] Round 124 Score Calculation Done!The process took 0.042497912 s.
[7/28/2022, 11:25:02 PM] [WARNING] Round 123 Score Calculation Done!The process took 0.050908921 s.
[7/28/2022, 11:20:02 PM] [WARNING] Round 122 Score Calculation Done!The process took 0.059849378 s.
[7/28/2022, 11:15:02 PM] [WARNING] Round 121 Score Calculation Done!The process took 0.059849378 s.
[7/28/2022, 11:10:01 PM] [WARNING] Round 120 Score Calculation Done!The process took 0.044571125 s.
[7/28/2022, 11:05:02 PM] [WARNING] Round 119 Score Calculation Done!The process took 0.061417722 s.
[7/28/2022, 11:00:02 PM] [WARNING] Round 118 Score Calculation Done!The process took 0.048562075 s.
[7/28/2022, 10:55:02 PM] [WARNING] Round 117 Score Calculation Done!The process took 0.191316711 s.
[7/28/2022, 10:50:01 PM] [WARNING] Round 116 Score Calculation Done!The process took 0.044604743 s.
[7/28/2022, 10:45:01 PM] [WARNING] Round 115 Score Calculation Done!The process took 0.054771433 s.
[7/28/2022, 10:40:02 PM] [WARNING] Round 114 Score Calculation Done!The process took 0.043381848 s.
[7/28/2022, 10:35:02 PM] [WARNING] Round 113 Score Calculation Done!The process took 0.045101825 s.
[7/28/2022, 10:30:02 PM] [WARNING] Round 112 Score Calculation Done!The process took 0.042570339 s.
[7/28/2022, 10:25:02 PM] [WARNING] Round 111 Score Calculation Done!The process took 0.043287019 s.
[7/28/2022, 10:20:01 PM] [WARNING] Round 110 Score Calculation Done!The process took 0.041480617 s.
[7/28/2022, 10:15:01 PM] [WARNING] Round 109 Score Calculation Done!The process took 0.038369982 s.
[7/28/2022, 10:10:02 PM] [WARNING] Round 108 Score Calculation Done!The process took 0.041480617 s.
[7/28/2022, 10:05:02 PM] [WARNING] Round 107 Score Calculation Done!The process took 0.061351103 s.
[7/28/2022, 10:00:02 PM] [WARNING] Round 106 Score Calculation Done!The process took 0.045564794 s.
[7/28/2022, 9:55:02 PM] [WARNING] Round 105 Score Calculation Done!The process took 0.039532519 s.
[7/28/2022, 9:50:01 PM] [WARNING] Round 104 Score Calculation Done!The process took 0.04477032 s.
[7/28/2022, 9:45:01 PM] [WARNING] Round 103 Score Calculation Done!The process took 0.074230107 s.
[7/28/2022, 9:40:01 PM] [WARNING] Round 102 Score Calculation Done!The process took 0.047716553 s.
[7/28/2022, 9:35:01 PM] [WARNING] Round 101 Score Calculation Done!The process took 0.0418794 s.
[7/28/2022, 9:30:02 PM] [WARNING] Round 100 Score Calculation Done!The process took 0.063601857 s.
[7/28/2022, 9:25:02 PM] [WARNING] Round 99 Score Calculation Done!The process took 0.045923113 s.

```

Hình 3.15. Giao diện chính của quản trị viên

TestCTF							
New Team		Team		Challenge		Gamebox	
ID	Logo	Team Name	Score	Token	Create At	Update At	Options
1		hihihoho	5000.00	j4bd2a75c23dd7cc8af0cfad6e7af643d1	7/28/2022, 2:04:25 PM	7/28/2022, 11:55:02 PM	<button>Edit</button> <button>Delete</button>
2		lmaobaka	0.00	c0633906c9ad7fad861ff621d3caa16	7/28/2022, 2:04:25 PM	7/28/2022, 3:28:33 PM	<button>Edit</button> <button>Delete</button>

Hình 3.16. Giao diện quản lý các đội chơi

ID	Challenge	Team	IP	Port	Score	Description	Down	Attacked	Create At	Options
1	driller	hihihoho	134.209.108.85	9091	1000.00	Please use name of the spot as flag ID.	false	false	7/28/2022, 2:34:25 PM	<button>Edit</button>
2	poipoi	hihihoho	134.209.108.85	3335	1000.00	Flag identifier: the flag_id is the user_id returned by the server after successful registration.	false	false	7/28/2022, 2:36:07 PM	<button>Edit</button>
3	sillybox	hihihoho	134.209.108.85	4669	1000.00	Flag identifier: The flag_id is the name of the file you have to read. It will contain the flag.	false	false	7/28/2022, 2:36:54 PM	<button>Edit</button>
4	tattletale	hihihoho	134.209.108.85	13007	1000.00	Flag identifier: the flag_id for this service is the room you work.	false	false	7/28/2022, 2:37:47 PM	<button>Edit</button>
5	temperature	hihihoho	134.209.108.85	56098	1000.00	Temperature service. Logs temperatures detected in various parts of the world. These recordings will then be used for amazingly secret intelligence actions Flag identifier: Date in the format YYYY/MM/DD	false	false	7/28/2022, 2:38:46 PM	<button>Edit</button>

Hình 3.16. Giao diện quản lý các dịch vụ

3.3. Cải tiến lối chơi

3.3.1. Vấn đề đặt ra

Hiện tại, framework không cung cấp bất cứ cách nào để lựa chọn mục tiêu tấn công mà sẽ thực thi các mã khai thác trên container với tất cả các dịch vụ của các đội chơi. Cách này sẽ làm giảm công việc của các đội chơi cũng như ban tổ chức, tuy nhiên lại làm cuộc thi đấu trở nên kém phần chiến thuật.

3.3.2. Phương án giải quyết

Để giải quyết vấn đề này, em thêm một giá trị truyền vào container khai thác khiến hệ thống truyền vào 4 giá trị chứ không phải là 3 như đã đề cập ở mục 3.2.3. Bốn giá trị đó lần lượt là TeamID, IP, Port và flag_id. Bằng cách này các đội chơi có thể chọn lọc các cuộc tấn công của mình, giúp nâng cao tính chiến thuật của cuộc thi.

3.4. Cải tiến cách tính điểm

3.4.1. Vấn đề đặt ra

Hiện tại, cách tính điểm của framework khá đơn giản. Một đội chơi ghi được điểm khi đánh cắp được flag của đội chơi khác, và đội bị đánh cắp sẽ mất cùng số điểm đó.

Cách tính điểm không đề cao được tính chiến thuật và không tận dụng được cải tiến về lối chơi được đề cập ở mục 4.2.1.

3.4.2. Phương án giải quyết

Tham khảo các cuộc thi gần đây, em nhận thấy chúng đều sử dụng các cách tính điểm khác nhau, nhưng tổng kết lại vẫn có sự tương đồng nhất định. Em đã tổng hợp và thêm bớt một số thành phần nhằm không làm phức tạp hoá cách tính điểm, gây khó khăn cho các đội chơi ít kinh nghiệm.

a) Ghi điểm

Thay vì các đội chỉ ghi được điểm của vòng đấu nếu các đội gửi lên hệ thống flag của vòng đấu đó, thì bây giờ một flag sẽ có giá trị trong vòng 15 vòng đấu kể từ khi flag được hệ thống setflag đặt cho dịch vụ. Điều này khiến tính chiến thuật cao hơn khi một đội chơi có thể giữ flag và không gửi lên hệ thống, vì lý do chiến thuật, giúp các đội phải tính toán kỹ lưỡng việc gửi flag hơn.

b) Tính điểm dịch vụ

Hệ thống hiện tại chỉ ghi điểm theo từng vòng đấu, mà không quan trọng việc dịch vụ tồn tại được bao nhiêu lâu trong quãng thời gian đã trôi qua. Điều này có thể khiến cho các đội chơi sử dụng chiến thuật tắt dịch vụ vào cuối cuộc thi nhằm bảo toàn điểm số mà không gặp bất cứ rào cản hay rủi ro nào.

Chính vì lý do này, em đề xuất thêm vào hệ thống một số giá trị sau:

- SLA: Điểm chất lượng dịch vụ. Một đội chơi không thể đảm bảo được dịch vụ của mình tồn tại càng lâu, thì điểm của đội chơi khi tấn công dịch vụ đó của đội chơi khác sẽ giảm đi. Ví dụ dịch vụ của đội chơi A tồn tại trong cả quá trình cuộc thi diễn ra thì điểm SLA của đội A là 1, nếu trong 4 tiếng mà chỉ có 1 tiếng dịch vụ của đội A hoạt động đúng cách thì SLA sẽ bằng $\frac{1}{4}$.
- Điểm của flag: Điểm này nhằm đánh giá mức độ am hiểu về dịch vụ của một đội chơi. Nếu đội chỉ tấn công mà không phòng thủ thì điểm này sẽ rất thấp và ngược lại. Khi tấn công thành công một dịch vụ thì điểm flag của đội tấn công sẽ tăng lên, đồng thời điểm flag của đội phòng thủ sẽ giảm đi.

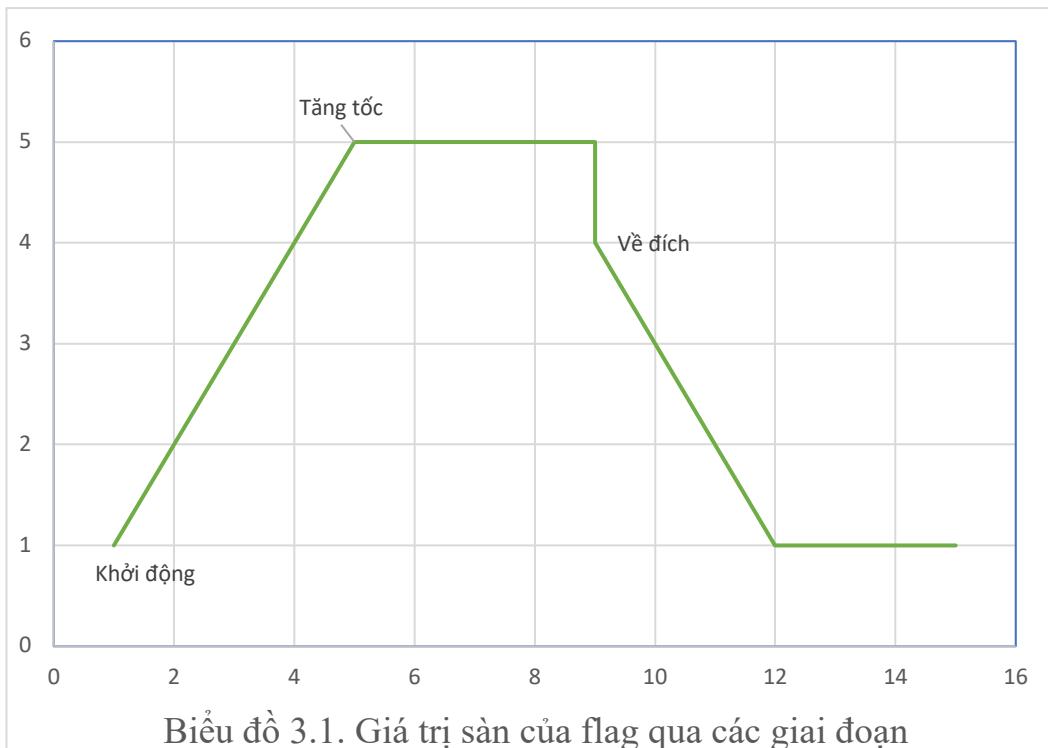
```

def get_score(team):
    score = 0
    for service in services:
        score += SLA(team, service) * FlagPoints(team, service)
    return score

```

Hình 3.14. Công thức tính điểm của một đội

- Mỗi flag sẽ có một giá trị sàn. Giá trị này sẽ được tính ở đầu mỗi vòng đấu và là giá trị chung của các đội. Một dịch vụ sẽ có 3 giai đoạn là khởi động, tăng tốc và về đích.
- Giá trị sàn sẽ tăng dần cho đến khi có một đội khai thác thành công một dịch vụ thì khi đó giá trị flag vẫn tăng trong một khoảng thời gian ngắn (khởi động).
- Khi quãng thời gian khởi động kết thúc, giá trị flag của dịch vụ đó sẽ được giữ nguyên (tăng tốc) cho đến khi hệ thống ghi nhận được một số lượng flag khác nhau nhất định. Giai đoạn này chính là lúc khai thác dịch vụ sẽ được điểm tối đa của dịch vụ đó
- Sau thời điểm này thì giá trị sàn của flag sẽ giảm dần về 1 và rồi giữ nguyên giá trị đó đến hết cuộc thi (về đích).
- Cơ chế này sẽ khiến các đội phải khẩn trương tấn công các dịch vụ để sớm vì sẽ nǎm được điểm tối đa của dịch vụ này, đồng thời giảm dần giá trị flag của dịch vụ đó về sau nhằm phân loại được mức độ khó, dễ của các dịch vụ và giảm gánh nặng bảo vệ các dịch vụ này của những đội top đầu.



3.5. Thiết kế kịch bản dịch vụ phục vụ thi đấu

3.5.1. Vấn đề đặt ra

Một cuộc thi CTF thường có 5 mảng chính là khai thác dịch vụ Web, khai thác lỗ hổng phần mềm, mật mã và mã khoá, dịch ngược và cuối cùng là điều tra kĩ thuật số. Thiết kế dịch vụ cho các đội, đồng thời chọn một tổ hợp các lỗ hổng để đưa vào các dịch vụ đó luôn là thử thách hàng đầu đối với ban tổ chức các cuộc thi. Nếu một lỗ hổng quá khó khai thác hay phát hiện sẽ khiến cho các đội thi chán nản, dễ gây tâm lý bỏ cuộc sớm. Chính vì lý do này nên em sẽ kết hợp một số lỗ hổng dễ dàng phát hiện, khai thác và sửa chữa với những lỗ hổng ở một mức độ cao hơn nhằm dễ dàng phân loại các đội chơi nhưng vẫn tạo sự hứng khởi cho những đội chơi ở trình độ thấp hơn.

Có hai cách đơn giản để đưa các lỗ hổng vào trong dịch vụ của các đội chơi. Cách thứ nhất là sử dụng những phiên bản phần mềm cũ, tồn tại lỗ hổng nhưng lại được sử dụng phổ biến khiến các đội có thể vận dụng kinh nghiệm trong quá khứ. Đồng thời có thể kết hợp với việc sử dụng một số ứng dụng hay các dự mã nguồn mở do học sinh, sinh viên hay các lập trình viên sơ đẳng tạo ra trước đó nhằm đảm bảo tính chất thực tế với những lỗi hay gấp của lập trình viên. Tuy vậy cách này lại có điểm hạn chế rất lớn là do những lỗ hổng này được công khai trên mạng nên có khả năng cao là đã có những đoạn mã khai thác

được công bố trước đó hoặc các đội có thể sửa chữa những lỗ hổng kiểu này bằng cách cập nhật phiên bản phần mềm mà dịch vụ sử dụng lên mới nhất.

Cách thứ hai, đó là sử dụng một dịch vụ có sẵn mà đảm bảo các yêu cầu về bảo mật, nhưng sau đó được sửa đổi để tạo ra lỗ hổng trong dịch vụ. Tuy nhiên dù lỗ hổng có được giấu kĩ đến mức nào thì vẫn có thể so sánh mã nguồn của dịch vụ với mã nguồn gốc. Vì vậy khi sử dụng cách này, ban tổ chức cần xoá hết dấu vết về phiên bản hay tên của dịch vụ gốc khỏi mã nguồn. Cách làm này gây tiêu tốn rất nhiều thời gian, đồng thời có thể khiến mọi công sức tiêu biến nếu không xoá sạch dấu vết.

Nhằm khắc phục nhược điểm của hai cách trên, ban tổ chức các cuộc thi thường chọn cách phát triển một dịch vụ mới. Nếu các thử thách của cuộc thi không yêu cầu việc học một ngôn ngữ lập trình mới thì các dịch vụ thường được viết bằng những ngôn ngữ phổ biến như JS, PHP, Python, C/C++. Bất kể lựa chọn là gì thì ban tổ chức hay người phát triển dịch vụ cần hiểu rõ các lỗ hổng cũng như cấu trúc dữ liệu của ngôn ngữ được sử dụng để đảm bảo được tiêu chuẩn của những lỗ hổng này. Dịch vụ nên có một phiên bản hoàn chỉnh không tồn tại lỗ hổng, theo sau đó là những phiên bản có chứa lỗ hổng có chủ đích. Bằng cách làm này, người phát triển dịch vụ sẽ đảm bảo được các lỗ hổng có mục đích rõ ràng, hướng đi cụ thể và giới hạn được khả năng khai thác dịch vụ của các đội chơi.

Mỗi một lỗ hổng đều có mục đích, hoặc là một mắt xích trong quy trình khai thác dịch vụ, hoặc cung cấp cho kẻ tấn công khả năng tiếp cận với flag. Trong mọi trường hợp thì việc khai thác một dịch vụ thành công không cung cấp khả năng xoá hay thay đổi flag, đồng thời không thể tồn tại lỗ hổng mà cho phép một dịch vụ có thể đọc được flag của dịch vụ khác. Những yêu cầu còn lại là tồn tại các đoạn script để hệ thống quản lý cuộc thi có thể thay đổi và lấy flag từ dịch vụ của các đội. Kèm theo đó là việc kiểm thử nghiêm ngặt để đảm bảo các dịch vụ tuân theo một kịch bản có trước. Việc tìm kiếm các lỗ hổng không có chủ đích không phải là yêu cầu bắt buộc, nhưng nên được áp dụng.

Để đảm bảo chất lượng cũng như đảm bảo về mặt thời gian đủ dài, em đề xuất mỗi mảng khai thác sẽ gồm 2 dịch vụ, một dịch vụ với các lỗ hổng đơn giản nhưng tồn tại nhiều lỗ hổng khác nhau để cao tốc độ khai thác và một dịch vụ sử dụng những kỹ thuật khó hơn, tập trung vào kiến thức và kỹ năng giải quyết tình huống của các đội chơi.

3.5.2. Các kịch bản

Do giới hạn về kiến thức cũng như kĩ thuật, em sẽ chỉ đưa ra 1 kịch bản khai thác đơn giản cho các đội chơi và thuộc mảng khai thác dịch vụ web:

- Tên dịch vụ: Website học tập - LMS
- Mã nguồn của dịch vụ: https://github.com/ngcaobaolong/web_demo
- Ngôn ngữ sử dụng: PHP
- Lỗi hỏng được thêm vào: login bypass, php type juggling, sql, php file inclusion.

3.5.2.1. Yêu cầu chức năng

Bảng 3.2. Yêu cầu chức năng của dịch vụ

Chức năng	Mô tả	Độ ưu tiên
Chức năng chung		
Đăng nhập	Học sinh hoặc quản trị viên có thể đăng nhập vào ứng dụng bằng tài khoản và mật khẩu.	Cao
Đăng ký	Học sinh có thể đăng ký tài khoản để sử dụng các chức năng của ứng dụng.	Cao
Chức năng của học sinh		
Xem danh sách các người dùng trên hệ thống	Học sinh có thể xem danh sách các người dùng trên hệ thống, bao gồm tên tài khoản, họ và tên, email, ảnh đại diện.	Cao
Xem danh sách các câu hỏi được đặt ra trên hệ thống	Học sinh có thể quan sát được những câu hỏi trên hệ thống và có thể gửi câu trả lời.	Cao
Xem danh sách bài tập được tạo ra trên hệ thống	Học sinh có thể quan sát được những bài tập được giao trên hệ thống, đồng thời có thể gửi file bài làm lên hệ thống để chấm điểm.	Cao

Nhắn tin với các người dùng khác	Học sinh có thể nhắn tin tới các học sinh hay quản trị viên khác trong khu vực thông tin chi tiết của user.	Cao
Xem được thông tin chi tiết của các người dùng trên hệ thống	Các đội chơi và quản trị viên có thể xem được thông tin chi tiết của một user trên hệ thống.	Cao
Thay đổi thông tin cá nhân của bản thân	Các đội chơi có thể thay đổi được thông tin cá nhân của bản thân, bao gồm email, số điện thoại, ảnh đại diện và mật khẩu	Cao
Đăng xuất	Đội thi có thể đăng xuất khỏi tài khoản hiện tại.	Cao
Chức năng của quản trị viên		
Tạo bài tập trên hệ thống	Quản trị viên có thể tạo bài tập, tải file bài tập lên hệ thống.	Cao
Quản lý các người dùng	Quản trị viên có thêm, sửa, xoá học sinh và thay đổi thông tin của các học sinh đó như tên người dùng, họ và tên, email, số điện thoại, ảnh đại diện, tài khoản và mật khẩu.	Cao
Tạo câu hỏi trên hệ thống	Quản trị viên có thể tạo các câu hỏi và đáp án trên hệ thống.	Cao
Đăng xuất	Quản trị viên có thể đăng xuất khỏi tài khoản hiện tại.	Cao

3.5.2.2. Yêu cầu phi chức năng

a. Yêu cầu về giao diện:

Màu sắc đặc trưng của frontend là màu trắng và cam. Trong các giao diện (ngoại trừ giao diện đăng nhập) luôn có navbar để đội chơi có thể dễ dàng chuyển chức năng khi sử dụng.

b. Yêu cầu về thực thi

Sử dụng ngôn ngữ PHP nhưng không sử dụng framework để dễ dàng thêm lõi một cách có chủ đích. Yêu cầu luồng của ứng dụng dễ hiểu, không gây khó khăn trong việc sử dụng các chức năng cơ bản của ứng dụng.

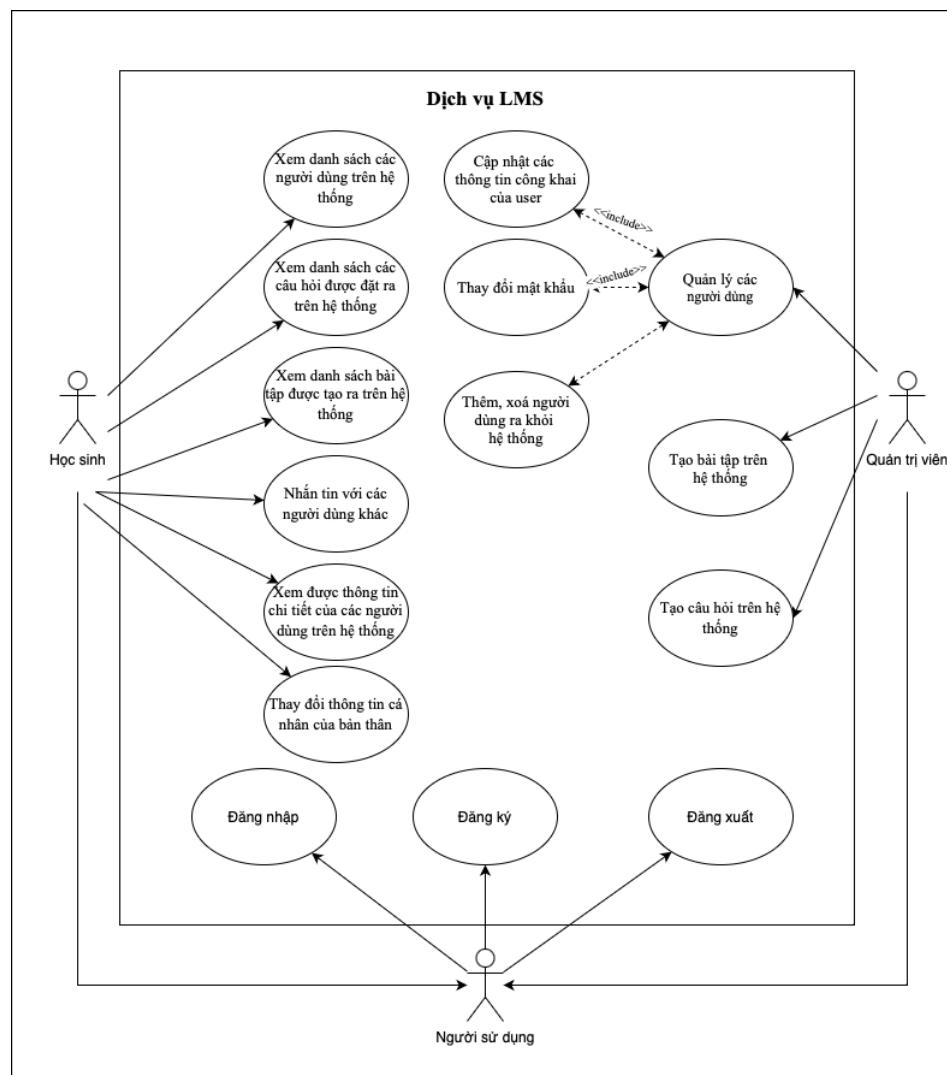
c. Yêu cầu về hiệu năng

Tốc độ phản hồi nhanh, hiển thị bài tập, các câu hỏi hay danh sách user trong 3s. Phải chịu được lượng kết nối lớn do trung bình một cuộc thi sẽ có khoảng 30 người dùng tại cùng một thời điểm.

d. Yêu cầu về bảo mật

Bảo đảm tính bảo mật không để lộ flag giữa các đội, đồng thời yêu cầu các cơ chế bảo vệ với hệ thống cơ sở dữ liệu và hệ thống.

3.5.2.3. Thiết kế logic



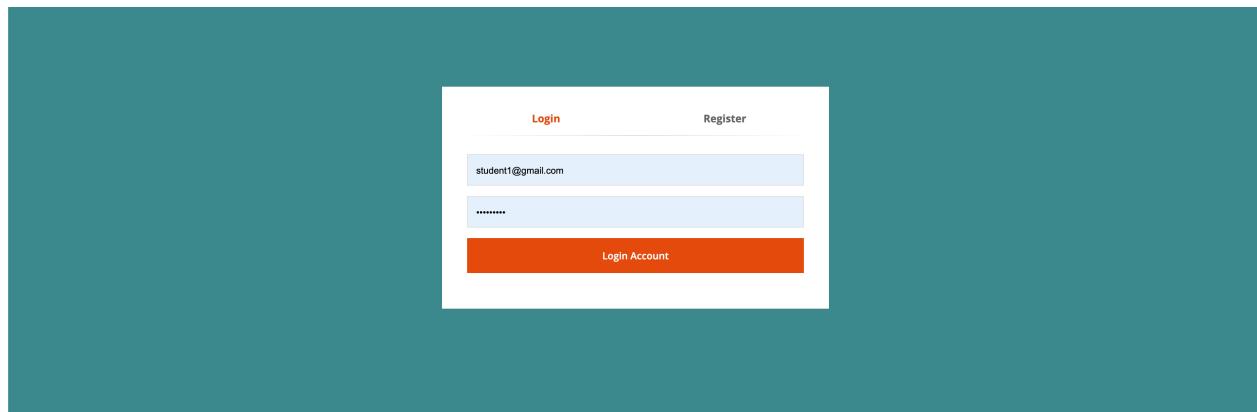
Biểu đồ 3.2. Biểu đồ Usecase tổng quát

Các tác nhân của hệ thống:

- Người dùng: Người dùng của hệ thống sẽ bao gồm các thành viên của đội thi và quản trị viên hay ban tổ chức. Người dùng quản trị chỉ có thể đăng nhập thông qua tài khoản và mật khẩu cho trước, người dùng có quyền hạn bình thường có thể đăng ký tại trang đăng ký tài khoản của ứng dụng.
- Học sinh: Tập hợp những người dùng sau khi đăng nhập thành công qua cổng đăng nhập được gọi là học sinh và có quyền sử dụng những chức năng như xem danh sách các học sinh, xem thông tin chi tiết các học sinh có trên hệ thống như email, ảnh đại diện, số điện thoại..., trả lời các câu hỏi và tải lên câu trả lời của các bài tập mà quản trị viên đưa ra.
- Quản trị viên: Người dùng sau khi đăng nhập qua giao diện đăng nhập với quyền quản trị thì được gọi là quản trị viên. Quản trị hệ thống có quyền hạn cao nhất trong hệ thống, có thể thay đổi, thêm bớt, sửa xoá các user và thông tin của các user đó, đồng thời có khả năng tải lên các bài tập, câu hỏi để các user có thể trả lời.

3.5.2.4. Kết quả

Trang đăng nhập của các học sinh, có thể đăng ký tài khoản với quyền học sinh tại đây:



Hình 3.15. Giao diện đăng nhập của dịch vụ



Hình 3.16. Giao diện chính của dịch vụ

Giao diện liệt kê danh sách người dùng trên hệ thống:

The screenshot shows the "User list" page. At the top, there is a navigation bar with links for "Home", "User list", "Quiz", and "Homework". On the right side of the header, there is a red button labeled "Hello, student1" and a "Logout" link. Below the header, the page title "User list" is displayed in bold. In the center, there is a table with two columns: "USER" and "EMAIL". The table contains one row with the user information: "a" (student1) and "ssss@gmail.com".

Hình 3.17. Giao diện danh sách người dùng trên hệ thống

The screenshot shows the "Course Quiz Page". At the top, there is a navigation bar with links for "Home", "User list", "Quiz", and "Homework". On the right side of the header, there is a red button labeled "Hello, student1" and a "Logout" link. Below the header, the page title "Course Quiz Page" is displayed in bold. In the center, there is a section titled "Quiz" with the sub-section "Guess the Liter". It includes a hint "Hint: aaaaaaaaaaa" and a text input field with the placeholder "Type your answer in here". Below the input field is a blue "Send answer" button.

Hình 3.18. Giao diện trả lời câu hỏi

Giao diện thông tin chi tiết người dùng và tin nhắn tới:

The screenshot shows a user profile editing interface. At the top, there is a placeholder for an avatar with a smiley face icon. Below it, the username 'student1' is displayed in a dark blue box. The main form area contains fields for 'Username' (student1), 'Full Name' (a), 'Email Address' (ssss@gmail.com), 'Phone number' (123), and 'New Password' (*****). There is also a 'Upload Avatar' section with a 'Choose File' button and a 'No file chosen' message. A 'Submit Changes' button is located below the password field. At the bottom, a message from others section is visible.

Hình 3.19. Giao diện bảng điểm và tương tác giữa các đội

Giao diện thêm người dùng của quản trị viên:

The screenshot shows an 'Add user' form. It includes fields for 'Username' (student1), 'Full name', 'Email Address', 'Phone number', and 'New Password'. A large orange 'Add an Account' button is at the bottom. The entire form is set against a teal background.

Hình 3.20. Giao diện thêm người dùng của quản trị viên

3.5.2.5. Đóng gói dịch vụ

Ban tổ chức có thể sử dụng Debian package như framework đã đề xuất, hoặc sử dụng Docker để dựng dịch vụ. Tuy cách dựng dịch vụ trên docker giúp người tạo có thể kiểm soát và đóng gói dễ dàng, nhưng bù lại không thể đảm bảo được tính chuẩn xác cũng như đồng nhất giữa các phiên bản vì có thể tồn tại những thay đổi không được ghi chép lại có thể khiến cho dịch vụ hoạt động không đúng như kịch bản. Đổi lại, với một dịch vụ phức tạp, việc viết và xây dựng một debian package tốn rất nhiều thời gian và công sức. Tạo một debian package tuy đơn giản, nhưng phụ thuộc vào độ phức tạp của dịch vụ thì việc tối ưu cũng như đảm bảo tính đúng đắn của Debian package cũng rất quan trọng.

Để đóng gói dịch vụ trên, sử dụng dpkg-deb với tham số build để tạo debian package. Sau đó thay vì cài đặt LAMP thủ công, em sử dụng Docker image có sẵn LAMP và đưa mã nguồn vào thư mục /var/www/html. Framework cũng cung cấp script create_container giúp tự động quá trình tạo các tệp tin cần thiết để ban tổ chức có thể tạo container chứa dịch vụ.

Chương 4. KẾT LUẬN

4.1. Kết quả đạt được

- Framework đã được cài đặt và hoạt động ổn định.
- Các yêu cầu về chức năng của ứng dụng đã hoạt động chính xác. Có thể tạo đội chơi và triển khai các dịch vụ lên hệ thống máy chủ.
- Đã áp dụng được một số cải tiến vào hệ thống và không gây mất ổn định của framework.

4.2. Các vấn đề còn tồn đọng

Khóa luận tốt nghiệp này vẫn còn rất nhiều điểm hạn chế cũng như tiềm năng để phát triển thêm những tính năng liên quan.

Tuy là điểm mạnh nhưng đồng thời cũng là một điểm yếu của hệ thống tổ chức thi CTF sử dụng đóng gói ứng dụng là các đội chơi không thể nghe lén các gói tin mạng trong hệ thống server được. Nguyên do là các đội chơi không thực sự kết nối tới hệ thống máy chủ mà chỉ đẩy các container chứa dịch vụ và mã khai thác lên hệ thống để hệ thống tự động khởi tạo và khởi chạy. Có thể cô lập toàn bộ hệ thống dịch vụ sang 1 server và cho các đội chơi có quyền truy cập từ xa nhưng điều đó lại làm hỏng cấu trúc tổng thể của framework.

Thứ hai là việc tồn tại nút nghẽn ở vị trí của Docker, khi mà khởi chạy quá nhiều container khai thác dẫn đến việc hệ thống xử lý bị đình trệ. Có thể giải quyết vấn đề này bằng cách sử dụng nhiều Docker daemon hơn, hoặc chia các dịch vụ sang nhiều hệ thống máy chủ mà mỗi máy có một Docker daemon. Hiện tại do chức năng sử dụng nhiều Daemon docker đang trong giai đoạn thử nghiệm, đồng thời do giới hạn về khả năng tiếp cận nhiều hệ thống máy chủ nên khóa luận này sẽ không đề cập chi tiết về vấn đề này.

TÀI LIỆU THAM KHẢO

Tiếng Việt:

- [1] pviethieu, "Một số kiến thức cơ bản về VueJS," Viblo, 11 September 2020. [Online]. Available: <https://viblo.asia/p/mot-so-kien-thuc-co-co-ban-ve-vuejs-yMnKMjpgZ7P>.

Tiếng Anh:

- [2] Arvind S Raj, Bithin Alangot, Seshagiri Prabhu, and Krishnashree Achuthan, Amrita Vishwa Vidyapeetham, "Scalable and Lightweight CTF Infrastructures Using Application Containers," *USENIX Workshop on Advances in Security Education*, 2016.
- [3] vidar-team, "Cardinal_frontend," [Online]. Available: https://github.com/vidar-team/Cardinal_frontend.
- [4] Giovanni Vigna, Kevin Borgolte, Jacopo Corbetta, Adam Doupe, Yanick Fratantonio, Luca Invernizzi, Dhilung Kirat, and Yan Shoshitaishvili, University of California, Santa Barbara, "Ten Years of iCTF: The Good, The Bad, and The Ugly," in *USENIX Summit on Gaming, Games and Gamification in Security Education*, San Diego, 2014.
- [5] inctf, "The InCTF Framework," [Online]. Available: <https://github.com/inctf/inctf-framework>.
- [6] vidar-team, "Cardinal frontend," [Online]. Available: https://github.com/vidar-team/Cardinal_frontend.
- [7] wuhan005, "Asteroid," [Online]. Available: <https://github.com/wuhan005/Asteroid>.
- [8] HITB, "HITB pro ctf 2021," [Online]. Available: <https://github.com/HITB-CyberWeek/proctf-2021>.
- [9] shellphish, "The iCTF Framework 3.0," [Online]. Available: <https://github.com/shellphish/ictf-framework>.