BÁO CÁO BÀI TẬP

**Môn học: Cơ chế hoạt động của mã độc**

**Kỳ báo cáo: Buổi 04 (Session 04)**

**Tên chủ đề:**

*GV: Nghi Hoàng Khoa*

*Ngày báo cáo: 08/05/2023*

1. **THÔNG TIN CHUNG:**

*(Liệt kê tất cả các thành viên trong nhóm)*

Lớp: NT230.N21.ANTN

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Võ Anh Kiệt | 20520605 | 20520605@gm.uit.edu.vn |

1. **NỘI DUNG THỰC HIỆN:[[1]](#footnote-1)**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Công việc** | **Kết quả tự đánh giá** | **Người đóng góp** |
| 1 | Kịch bản 1 đến kịch bản 5 | 100% |  |
| 2 | Kịch bản 6 |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.**

BÁO CÁO CHI TIẾT

## Kịch bản 01

Graphical user interface, text, application, email

Description automatically generated

Đầu tiên ta thấy được là trong b4rn init có 5 function chính

Misc\_register

Init\_override

Init\_proc\_mods

Init\_proc\_maps

Init system calls

Sau khi được gọi insmod thì chương trình sẽ thực hiện b4rn\_init



Hàm Misc\_register

Thực hiện việc đăng ký thiết bị vào trong đó chính là b4rn\_dev để thực hiện các cấp quyền như read, write và các function có trong root kit

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

Ngoài ra khi thực hiện đăng ký thiết bị thì đã set up thêm các cơ chế đọc ghi và các function có sẵn của chương trình.

Hàm Init\_override

Graphical user interface, text, application, email

Description automatically generated

Thực hiện việc việc tìm kiếm set\_memory ở dạng read write hay là read only để thực hiện thao tác cho các function bên dưới.

Hàm Init\_proc\_mods

Graphical user interface, text, application, email

Description automatically generated

Với hàm này thì chương trình thực hiện việc việc che giấu các modules có trong /proc/modules

Với thao tác thực hiện việc ghi đè ở trong đường dẫn file bên trên bằng địa chỉ của modules sao cho có thể che giấu đi được các modules của root kit nhằm thực hiện quá trình ẩn thân của root kit

Hàm Init\_proc\_maps

Graphical user interface, text, application, email

Description automatically generated

Với hàm này thì tư tưởng sẽ giống với hàm proc\_mods bằng việc là che giấu đi các tiến trình có bên trong đường dẫn /proc/self/maps nhưng khác với proc\_mods thì proc\_maps thực hiện việc xoá đi những tiến trình có liên quan đến prefix b4rnd00r thì chương trình sẽ thực hiện thao tác xoá đi thày vì thực hiện thao tác ghi đè như proc\_mods

Hàm init\_system\_call

Graphical user interface, text, application, email

Description automatically generated

Với hàm này thì chương trình sẽ thực hiện việc chặn hiển thị những file có prefix là b4rnd00r bằng cơ chế sau

Đầu tiên người dùng sẽ thực hiện lệnh ls

Sau đó lệnh ls truyền đến hệ thống

Hệ thống sẽ gửi kết quả đến người dùng là các file đang có nhưng trong trường hợp này trước khi gửi đến người dùng thì chương trình sẽ thực hiện chặn thông tin và thực hiện chỉnh sửa bằng tìm kiếm các hiện thị có từ khoá là b4rnd00r và xoá đi trước khi hiển thị lên. Từ đó người dùng sẽ không thể nhìn thấy các file có prefix là b4rnd00r

Demo trên vm ware

Graphical user interface, text, application

Description automatically generated

Demo trên virtual box

A screenshot of a computer

Description automatically generated

## Kịch bản 02

Câu trả lời là có vì chương trình được cài đặt dưới dạng module kernel (có thể chạy ở phân quyền nhân – kernel priviledge level) từ đó có thể mở được các port TCP dùng để backdoor. Qua thao tác trên có thể cấp quyền truy cập cho kẻ tấn công từ xa thông qua bind shell hoặc reverse shell

## Kịch bản 03

Ngoài ra thay vì add trên thiết bị ta có thể add trên master boot loader giúp cho việc có thể chạy trước khi bật hệ điều hành. Ngoài ra có thể thực hiện chiếm quyền kiểm soát hệ thống sau đó tạo bind shell và cấu hình nó như việc 1 chương trình chạy nền, từ đó rootkits có thể che giấu thông tin bind shell.

## Kịch bản 04

1. Nếu chương trình muốn sử dụng các certain routines trong kernel thì nó phải gọi API kallsyms\_\*() và gọi tên symbol để lấy các con trỏ của hàm đó. Vì vậy để thực hiện điều trên thì nó cần phải có con trỏ hàm
2. Để thực hiện quá trình che giấu, chúng ta cần đến việc khả năng đọc ghi và thực hiện, nhưng khi làm việc này vấn đề phát sinh là kernel sẽ không cho phép ta thực hiện bằng cách kernel sẽ thực hiện read only ở kernel mode. Như vậy sẽ có 1 exception được thành lập nếu như cố sửa.

Vậy để vượt qua việc chặn này (page protection) của kernel ta sẽ thực hiện

Bước 1 tắt tính năng chống ghi (thực hiện chuyển từ read only sang read write)

Bước 2 thực hiện thao tác ghi đè

Bước 3 bật tính năng chống ghi (thực hiện chuyện lại read only)

Trong bài này bit chống ghi có trong cr0

Do đó ta cần thực hiện các thao tác liên quan đến gỡ bảo vệ trang và thực hiện thao tác trên bit cr0 để thực hiện ghi đè hàm.

1 hình ảnh hàm kallsysm

A screenshot of a computer code

Description automatically generated with medium confidence

2 hình ảnh hàm protect và unprotect page và thao tác cr0

A screenshot of a computer program

Description automatically generated with medium confidence

## Kịch bản 05

Đầu tiên ta thực hiện kiểm tra code thì thấy được thông tin có phần reboot trong hệ thống

A screenshot of a computer code

Description automatically generated with low confidence

Vì vậy thay vì add như một chương trình thông thường ta có thể add trên master boot loader giúp cho việc có thể chạy trước khi bật hệ điều hành. Ngoài ra có thể thực hiện chiếm quyền kiểm soát hệ thống sau đó tạo bind shell và cấu hình nó như việc 1 chương trình chạy nền, từ đó rootkits có thể che giấy thông tin bind shell.

## Kịch bản 06

---

***Sinh viên đọc kỹ yêu cầu trình bày bên dưới trang này***

# **YÊU CẦU CHUNG**

* Sinh viên tìm hiểu và thực hành theo hướng dẫn.
* Nộp báo cáo kết quả chi tiết những việc (**Report**) bạn đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
* Sinh viên báo cáo kết quả thực hiện và nộp bài.

**Báo cáo:**

* File .PDF. Tập trung vào nội dung, không mô tả lý thuyết.
* Nội dung trình bày bằng Font chữ Times New Romans/ hoặc font chữ của mẫu báo cáo này (UTM Neo Sans Intel/UTM Viet Sach)– cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.
* Đặt tên theo định dạng: [Mã lớp]-SessionX\_GroupY. (trong đó X là Thứ tự buổi Thực hành, Y là số thứ tự Nhóm Thực hành/Tên Cá nhân đã đăng ký với GV).

*Ví dụ: [*NT101.K11.ANTT*]-Session1\_Group3.*

* Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
* Không đặt tên đúng định dạng – yêu cầu, sẽ **KHÔNG** chấm điểm.
* Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

**Đánh giá**: Sinh viên hiểu và tự thực hiện. Khuyến khích:

* Chuẩn bị tốt.
* Có nội dung mở rộng, ứng dụng trong kịch bản/câu hỏi phức tạp hơn, có đóng góp xây dựng.

*Bài sao chép, trễ, … sẽ được xử lý tùy mức độ vi phạm.*

**HẾT**

1. Ghi nội dung công việc, các kịch bản trong bài Thực hành [↑](#footnote-ref-1)