Secure Programming
Brute Force Attack and Secure Captcha

Tài liệu này nhằm mục đích trang bị cho lập trình viên ứng dụng kiến thức cơ bản về tấn công Brute force và một số lỗi sử dụng Captcha không an toàn. Bên cạnh đó, đưa ra những khuyến cáo khi lập trình để phòng ngừa nguy cơ tấn công Brute Force và sử dụng Captcha hiệu quả.

Chi tiết văn bản

Phiên bản	Ngày sửa	Người sửa	Mục đích
1.0	15/10/2014	Võ Đức Quang	Tạo văn bản

Hà Nội 10/2014

1. Tấn công Brute force

1.1. Giới thiệu

Brute force attack gọi nôm na là tấn công bạo lực, đoán mò. Là phương pháp sử dụng các phép thử đúng sai với các dữ liệu tạo sẵn để dò ra thông tin phù hợp. Các trường hợp áp dụng tấn công Bruteforce :

- Thu thập các thông tin chưa biết trước như user/pass (Form Login), thông tin email (Form lấy lại mật khẩu),…
- Tấn công mã hóa để thu chuỗi clear text

Tùy theo các trường hợp thực tế, phương pháp tấn công này sẽ sử dụng Từ điển có sẵn (Dictionary) hoặc vét cạn (Thử tất cả các tổ hợp).

Khi đi vào tấn công brute force một website cụ thể, ta phải xác định được các dấu hiệu để nhận biết được khi nào tìm ra được bộ giá trị đúng: Cái gì sẽ xảy ra khi login thành công, cái gì xảy ra khi login không thành công?

Trường hợp thông dụng thường gặp nhất là tấn công Form Login với từ điển user/pass có sẵn. Các yếu tố ảnh hưởng đến việc sử dụng tấn công Brute force dạng này:

- Cơ chế login của trang web
  - Có cơ chế chống brute force hay không?
  - Có khả năng bypass hay không?
- Tài nguyên hệ thống dùng để tấn công

Công cụ gửi request tấn công có thể tự code bằng các ngôn ngữ script hoặc các tools proxy như Burpsuite.

Demo

Một video demo tấn công Brute force Login trang web www.nhac.vui.vn .

http://youtu.be/qHvQ_Xkf4-g

Ở đây, website không có bất kỳ cơ chế chống tấn công Bruteforce nào.

Demo sử dụng công cụ Burpsuite tấn công Bruteforce với Username vét cạn 6 ký tự, Password là list đơn giản gồm 2 mật khẩu là: *"123456", "123123"*.

Dấu hiệu nhận biết đúng sai trong kết quả trả về thông qua *"Content-Length"* trong Response.

1.2. Cơ chế chống tấn công Brute force

- Đối với dạng tấn công dò tìm clear text từ thông tin mã hóa
    - Sử dụng kiểu mã hóa là hàm băm 1 chiều kết hợp với *Salt*
    - Mã hóa nhiều lần
    - Giữ bí mật cơ chế mã hóa
- Đối với tấn công dò tìm thông tin thông qua việc gửi request
    - Sử dụng captcha bảo vệ sau nhiều lần nhập thông tin sai
    - Block tài khoản, IP trong một khoảng thời gian tùy theo mức độ

2. Captcha trong các ứng dụng

2.1. Captcha

CAPTCHA là cụm từ viết tắt của "*Completely Automated Public Turing Test to Tell Computers and Humans Apart*" ("phép thử Turing để phân biệt giữa người và máy - Turing là một nhà khoa học người Anh"). Đây là phép thử để xác định xác suất đối tượng được thử là con người, cho phép quản trị viên các trang web phân biệt đâu là người, đâu là máy.

Việc sử dụng captcha thông qua ngữ cảnh một bài kiểm tra mà chỉ có con người dùng các giác quan, kiến thức của mình mới giải được, sau đó phía server kiểm tra đáp án này, nếu đúng sẽ xử lý yêu cầu.

Dạng quen thuộc nhất của Captcha là một hình ảnh gồm các ký tự bị bóp méo. Và công việc để làm là phải gõ lại đúng các ký tự này vào một ô vuông. Nếu những gì gõ vào khớp với hình ảnh trên thì mình đã hoàn thành được bài kiểm tra.

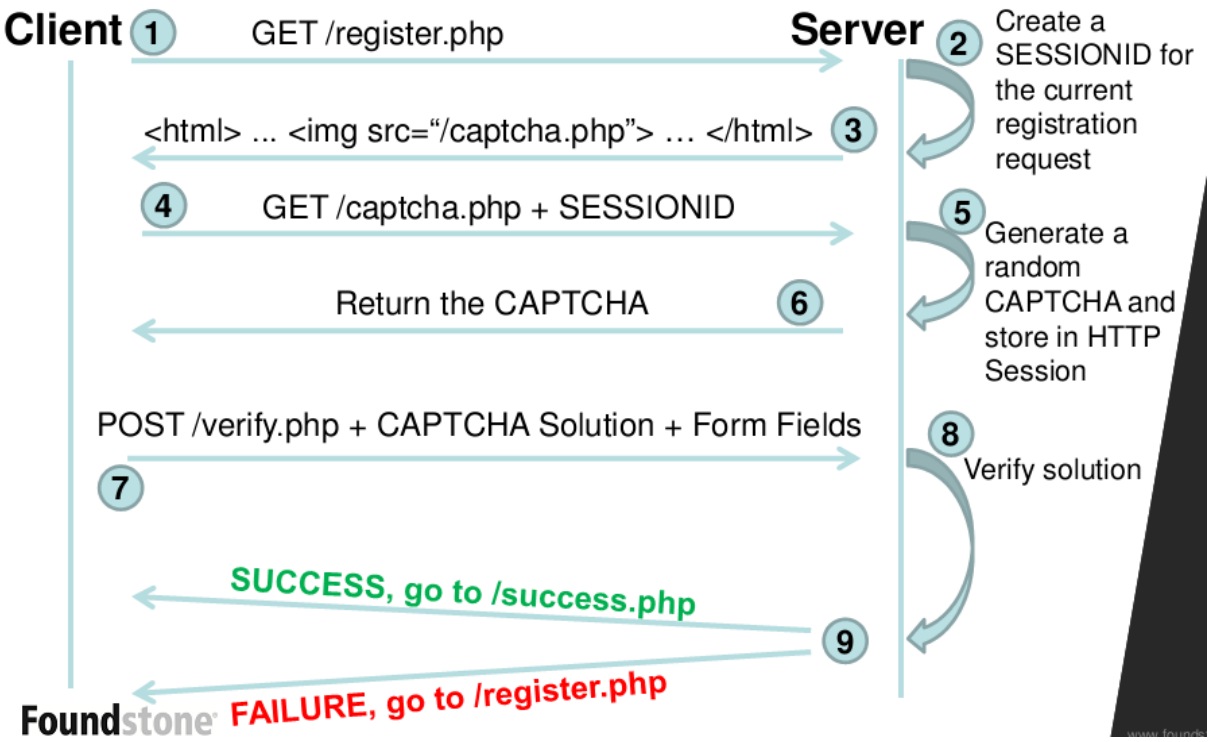
Các thành phần trong website thường được sử dụng CAPTCHA

- *Phần đăng ký thành viên*
- *Chức năng khôi phục mật khẩu*
- *Chức năng post bài viết*
- *Xác nhận thanh toán trong các web thương mại…*

2.2. Mô hình triển khai Captcha an toàn

Một mô hình Captcha được triển khai an toàn phải tuân thủ theo các bước như sau:

Trong hình là Captcha sử dụng cho form đăng ký.



Quá trình thực hiện như sau:

Bước 1: Truy cập site có sử dụng xác thực captcha (ở đây là trang đăng ký).

Bước 2: Server tạo một SessionID mới tương ứng với phiên làm việc hiện tại của Client.

Bước 3: Server gửi về Client đường dẫn để lấy về CAPTCHA

Bước 4: Client gửi yêu cầu Captcha cùng với SessionID

Bước 5: Tạo hình ảnh CAPTCHA lưu trữ

Bước 6: Server gửi Client một CAPTCHA

Bước 7: Client gửi các thông tin và lời giải CAPTCHA cho Server để xác thực

Bước 8: Server xác thực Captcha: phiên làm việc? CAPTCHA? Lời giải CAPTCHA? và sẽ gửi phản hồi về cho Client. Xóa Session Captcha.

Bước 9: Nếu thông tin phản hồi là thành công thì thực hiện bước tiếp theo. Thất bại thì yêu cầu thực hiện lại

2.3. Các lỗi thường gặp khi triển khai Captcha

Một số lỗi thường gặp khi triển khai Captcha:

- Tin tưởng phía người sử dụng
    - Captcha được tạo và Verify ngay bên phía Client

    - Captcha được tạo bên Server và Verify phía Client
    - Captcha được tạo bên Server nhưng việc sinh captcha phụ thuộc vào đầu vào từ Client. (Lựa chọn văn bản tấn công)
- Lỗi xác thực captcha
    - Thực hiện kiểm tra xác thực Captcha sau các việc kiểm tra khác
    - Không xóa session Captcha sau khi xác thực dẫn đến việc sử dụng lại
- Sai sót gửi đáp án Captcha về cho Client

3. Lập trình an toàn

3.1. Chống tấn công Brute Force

- Đối với dạng tấn công dò tìm clear text từ thông tin mã hóa
    - Sử dụng kiểu mã hóa là hàm băm 1 chiều kết hợp với Salt
    - Mã hóa nhiều lần
    - Giữ bí mật cơ chế mã hóa
- Đối với tấn công dò tìm thông tin thông qua việc gửi request như Form Login
    - Sử dụng captcha bảo vệ sau nhiều lần nhập thông tin sai
    - Triển khai mô hình Captcha an toàn (Mục 3.2)
    - Block theo tài khoản, IP trong một khoảng thời gian tùy theo chức năng và mức độ nguy hiểm

3.2. Triển khai Captcha an toàn

- Sử dụng loại Captcha đủ mạnh
    - Nội dung captcha tạo ra không được phụ thuộc vào các yếu tố tác động từ phía Client
    - Các văn bản Captcha phải được tạo ngẫu nhiên,
    - Các hình ảnh Captcha có thể nhận diện được bởi con người còn máy thì không
    - Thông tin Captcha phải được mã hóa
    - Captcha không được tái sử dụng
- Triển khai theo mô hình Captcha an toàn (Mục 2.2)
- Tạo Captcha bên phía Server, Xác thực Captcha bên phía server
- Không gửi đáp án Captcha về Client
- Về thứ tự kiểm tra: luôn kiểm tra xác thực Captcha trước khi kiểm tra các thông tin khác
- Luôn xóa session Captcha sau khi xác thực