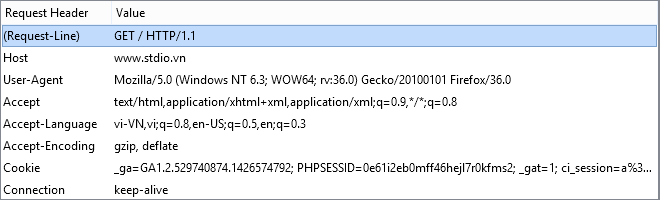
**ĐỀ CƯƠNG**

**Xây dựng ứng dụng Web an toàn**

# 1. Trình bày cấu trúc của truy vấn HTTP và phản hồi HTTP.

**HTTP Request**

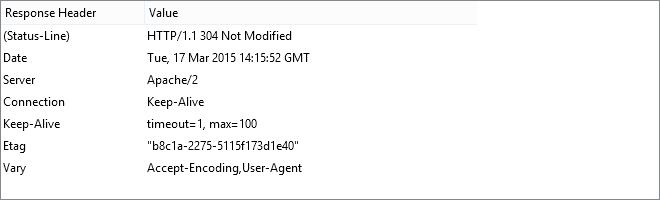
  
Ví dụ một HTTP Request

* Bắt đầu của HTTP Request sẽ là dòng Request-Line bao gồm 3 thông tin đó là:
* Method: là phương thức mà HTTP Request này sử dụng, thường là GET, POST, ngoài ra còn một số phương thức khác như HEAD, PUT, DELETE, OPTION, CONNECT. Trong ví dụ trên là GET
* URI: là địa chỉ định danh của tài nguyên. Trong tường hợp này URI là / - tức request cho tài nguyên gốc, nếu request không yêu cầu một tài nguyên cụ thể, URI có thể là dấu \*.
* HTTP version: là phiên bản HTTP đang sử dụng, ở đây là HTTP 1.1.
* Tiếp theo là các trường request-header, cho phép client gửi thêm các thông tin bổ sung về thông điệp HTTP request và về chính client. Một số trường thông dụng như:
* Accept: loại nội dung có thể nhận được từ thông điệp response. Ví dụ: text/plain, text/html…
* Accept-Encoding: các kiểu nén được chấp nhận. Ví dụ: gzip, deflate, xz, exi…
* Connection: tùy chọn điều khiển cho kết nối hiện thời. Ví dụ: keep-alive, Upgrade…
* Cookie: thông tin HTTP Cookie từ server.
* User-Agent: thông tin về user agent của người dùng.

**HTTP Response**

Cấu trúc HTTP response gần giống với HTTP request, chỉ khác nhau là thay vì Request-Line, thì HTTP có response có Status-Line. Và giống như Request-Line, Status-Line cũng có ba phần như sau:

* HTTP-version: phiên bản HTTP cao nhất mà server hỗ trợ.
* Status-Code: mã kết quả trả về.
* Reason-Phrase: mô tả về Status-Code.



# 2. Phân biệt hai phương thức POST và GET.

Hai phương thức được sử dụng nhiều nhất trong HTTP request là GET và POST

Với GET, câu truy vấn sẽ được đính kèm vào đường dẫn của HTTP request. Ví dụ: **/?username=”abc”&password=”def”**

Một số đặc điểm của phương thức GET:

* GET request có thể được cached, bookmark và lưu trong lịch sử của trình duyệt.
* GET request bị giới hạn về chiều dài, do chiều dài của URL là có hạn.
* GET request không nên dùng với dữ liệu quan trọng, chỉ dùng để nhận dữ liệu.

Ngược lại, với POST thì câu truy vấn sẽ được gửi trong phần message body của HTTP request, một số đặc điểm của POST như:

* POST không thể, cached, bookmark hay lưu trong lịch sử trình duyệt.
* POST không bị giới hạn về độ dài.

Các phương thức khác

Ngoài GET và POST, HTTP request còn có thể có một số phương thức khác như:

* HEAD:; giống như GET nhưng chỉ gửi về HTTP header.
* PUT: tải lên một mô tả về URI định trước.
* DELETE: xóa một tài nguyên định trước.
* OPTIONS: trả về phương thức HTTP mà server hỗ trợ.
* CONNECT: chuyển kết nối của HTTP request thành một kết nối HTTP tunnel.

# 3. Khái niệm và chức năng của cookie, session.

**Cookie**

Cookie là một đoạn văn bản ghi thông tin được tạo ra và lưu trên trình duyệt của máy người dùng. Cookie thường được tạo ra khi người dùng truy cập một website, cookie sẽ ghi nhớ những thông tin như tên đăng nhập, mật khẩu, các tuỳ chọn do người dùng lựa chọn đi kèm. Các thông tin này được lưu trong máy tính để nhận biết người dùng khi truy cập vào một trang web.

Khi người dùng truy cập đến một trang web có sử dụng cookie, web server của trang đó sẽ tự động gửi cookie đến máy tính của người dùng. Những cookie này tự động được tổ chức trong hệ thống máy tính. Khi truy cập đến các trang web sử dụng được cookie đã lưu, những cookie này tự động gửi thông tin của người dùng về cho chủ của nó (người tạo ra cookie). Tuy nhiên những thông tin do cookie ghi nhận không được tiết lộ rộng rãi, chỉ có website chứa cookie mới có thể xem được những thông tin này. Cookie được xem là một thành phần không thể thiếu được với những website có khối lượng dữ liệu lớn, có số lượng người dùng đông, và có những chức năng đi kèm với thành viên đăng ký. Phần lớn các website này là các website thương mại điện tử.

**Session**

Session là một khái niệm phổ biến được dùng trong lập trình các website có kết nối với cơ sở dữ liệu database. Đặc biệt các chức năng như đăng nhập, đăng xuất người dùng sẽ khó có thể thực hiện được nếu không sử dụng session.

Session đơn giản là 1 cách để chúng ta lưu lại dữ liệu của người dùng sử dụng website. Giá trị của session được lưu trong một tập tin trên máy chủ. Ví dụ khi bạn đăng nhập vào một trang web và đăng nhập với tài khoản đã đăng ký trước đó. Máy chủ sau khi xác thực được thông tin bạn cung cấp là đúng nó sẽ sinh ra một tập tin (hay chính là session của trình duyệt của bạn) chứa dữ liệu cần lưu trữ của người dùng.

Bạn có thể tuỳ ý quyết định xem nên lưu trữ những thông tin nào vào Session. Nhưng thông thường chúng ta chỉ nên lưu những thông tin tạm thời trong session ví dụ như số lượng sản phẩm người dùng đã thêm vào giỏ nhưng chưa mua, hay những nhật xét đang được viết và lưu dưới dạng nháp nhưng chưa gửi đi. Những dữ liệu sử dụng lâu dài như nội dung nhận xét đã được gửi đi hay số sản phẩm đã được mua thì nên được thực hiện ở máy chủ chứa cơ sở dữ liệu.

**So sánh**

| Cookie | Session |
| --- | --- |
| Cookie được lưu trữ trên trình duyệt của người dùng. | Session không được lưu trữ trong trình duyệt của người dùng. |
| Dữ liệu cookie được lưu trữ ở phía máy khách. | Dữ liệu session được lưu trữ ở phía máy chủ. |
| Dữ liệu cookie dễ dàng sửa đổi khi chúng được lưu trữ ở phía khách hàng. | Dữ liệu session không dễ dàng sửa đổi vì chúng được lưu trữ ở phía máy chủ. |
| Dữ liệu cookie có sẵn trong trình duyệt của chúng ta đến khi hết hạn. | Dữ liệu session có sẵn cho trình duyệt chạy. Sau khi đóng trình duyệt sẽ mất thông tin session. |

# 4. Phân loại hiểm họa an toàn ứng dụng web

1. Nhóm hiểm họa xác thực

Đó là những tấn công lên các phương pháp xác thực mà ứng dụng web sử

dụng để xác thực người dùng, dịch vụ hay ứng dụng khác.Trong trường hợp ứng

dụng web, việc xác thực người dùng thường được thực hiện bằng mật khẩu. Khi

đó, có những hiểm họa an toàn như sau:

* Hiểm họa dò mật khẩu
* Hiểm hoạ sử dụng lược đồ xác thực không đủ mạnh
* Hiểm hoạ khôi phục mật khẩu không an toàn

2. Nhóm hiểm họa trao quyền

Đó là những tấn công lên các phương pháp được web server sử dụng để xác

định xem người dùng có được trao quyền thực hiện một hành động nào đó hay

không. Các hiểm họa an toàn thông tin liên quan đến cơ chế trao quyền gồm:

* Hiểm họa bị đoán biết định danh phiên làm việc
* Hiểm hoạ sử dụng cơ chế phân quyền không an toàn
* Hiểm hoạ không giới hạn thời gian có hiệu lực của phiên làm việc
* Hiểm hoạ gán ghép định danh phiên làm việc

3. Nhóm hiểm họa tấn công lên máy khách

Những tấn công này chỉ tác động lên máy của người dùng web. Máy chủ không

thiệt hại gì trong trường hợp này. Khi người dùng duyệt một website anh ta thường

không nghĩ đến việc bị tấn công bởi chính website đó. Lợi dụng lòng tin này kẻ ác ý có thể sử dụng những phương pháp khác nhau để tấn công lên máy người dùng.

* Hiểm hoạ thực thi kịch bản liên trang
* Hiểm hoạ giả mạo nội dung
* Hiểm hoạ phân tách phản hồi HTTP

4. Nhóm hiểm họa thực thi câu lệnh

Đây là những tấn công khác nhau nhằm thực thi câu lệnh trên web server. Nếu

web server không có cơ chế an toàn đủ mạnh thì kẻ tấn công có thể làm thay đổi

những câu lệnh như thế nhằm đạt được những mục đích nhất định.

* Hiểm họa tràn bộ đệm
* Hiểm họa tấn công lên chuỗi định dạng
* Hiểm họa chèn câu lệnh LDAP
* Hiểm họa thực thi câu lệnh hệ điều hành
* Hiểm họa chèn câu lệnh SQL
* Hiểm họa chèn mã kịch bản phía máy chủ
* Hiểm họa chèn câu lệnh Xpath

5. Nhóm hiểm họa làm lộ thông tin

Các tấn công thuộc nhóm này được thực hiện nhằm mục đích thu thập thông tin

về ứng dụng web

* Hiểm họa duyệt thư mục
* Hiểm hoạ định danh các ứng dụng
* Hiểm hoạ rò rỉ thông tin
* Hiểm hoạ đường dẫn ngược
* Hiểm hoạ đoán biết vị trí tài nguyên

6. Nhóm hiểm họa tấn công logic

Các tấn công thuộc nhóm này nhằm vào việc khai thác các tính năng hoặc logic hoạt động của ứng dụng web.

* Hiểm họa sử dụng các tính năng 1 cách ác ý
* Hiểm họa từ chối dịch vụ
* Hiểm họa kiểm tra người/máy bất thành
* Hiểm họa kiểm tra tiến trình bất thành

# 5. Tấn công XSS: khái niệm, phân loại, hậu quả của tấn công XSS, cách thức tấn công và phòng chống.

XSS là một kỹ thuật tấn công bằng cách chèn vào các website động (ASP,

PHP, …) những thẻ HTML hay những đoạn mã nguy hiểm có thể gây nguy hại

cho những người sử dụng khác. Những đoạn mã nguy hiểm thường được viết bằng

Client site script như javaScript, Jscript

*Phân loại:*

* Stored XSS: là hình thức tấn công mà ở đó cho phép kẻ tấn công có thể chèn một đoạn script nguy hiểm (thường là Javascript) vào website của chúng ta thông qua một chức năng nào đó (vd: viết lời bình, guestbook, gửi bài..), để từ đó khi các thành viên khác truy cập website sẽ bị dính mã độc từ kẻ tấn công này, các mã độc này thường được lưu lại trong database của website chúng ta nên gọi là Stored. Stored XSS phát sinh do chúng ta không lọc dữ liệu do thành viên gửi lên một cách đúng đắn, khiến cho mã độc được lưu vào Database của website. Truy vấn để được lưu lâu dài trên server.
* Reflected XSS: được thực hiện và có hiệu lực trong một cặp truy vấn và phản hồi HTTP; Là dạng tấn công không có trạng thái, tức là kết quả truy vấn tiếp theo sẽ không phụ thuộc vào các truy vấn trước đó. Kịch bản của tấn công này thường yêu cầu nạn nhân phải nhấn chuột vào một đường liên kết được chuẩn bị trước bởi kẻ tấn công.

*Hiểm hoạ tấn công XSS:*

* Đánh cắp cookies
* Tấn công lừa đảo
* Sử dụng trình duyệt như là công cụ quét cổng trong mạng LAN

*Cách tấn công:*

Xét từ phía kẻ tấn công, quá trình thực hiện tấn công XSS sẽ trải qua các bước cơ bản như sau:

* B1:Mở website mục tiêu và xác định các vị trí có thể nhập dữ liệu
* B2:Xác định khả năng website mục tiêu có chứa lỗi XSS.
* B3:Chèn mã vào vị trí có khả năng chứa lỗi
* B4: Khai thác.

*Cách phòng chống:*

* Phát hiện và vô hiệu hóa các đoạn mã độc.
* Chỉ chấp nhận những dữ liệu hợp lệ.
* Từ chối nhận các dữ liệu hỏng.
* Liên tục kiểm tra và thanh lọc dữ liệu
* Tạo ra danh sách những thẻ HTML được phép sử dụng, xóa bỏ thẻ <script> hoặc đóng các thẻ Script trong thẻ <comment> coi đoạn Script đó như là một đoạn trích dẫn thôi.
* Lọc ra bất kì một đoạn mã JavaScript/Java/VBScript/ActiveX/Flash Related.
* Lọc dấu nháy đơn hay kép.
* Lọc ký tự Null.
* Xóa những kí tự “ > ”, “ < ”
* Ngoài ra, tấn công XSS có thể phòng tránh được nhờ vào công tác kiểm thử và kiểm duyệt mã nguồn.

# 6. Tấn công SQL injection: khái niệm, phân loại, hậu quả của tấn công SQL injection, cách thức tấn công và phòng chống.

SQL Injection là một kĩ thuật cho phép những kẻ tấn công thi hành các câu

lệnh truy vấn SQL bất hợp pháp (người phát triển không lường trước được) bằng

cách lợi dụng lỗ hổng trong việc kiểm tra dữ liệu nhập từ các ứng dụng web.

*Phân loại:*

* Kỹ thuật khai thác dữ liệu thông qua toán tử UNION
* Kỹ thuật khai thác thông qua các câu lệnh điều kiện
* Kỹ thuật khai thác dữ liệu với Blind SQL Injection
* Kỹ thuật vượt qua các bộ lọc tham số đầu vào

*Hậu quả:* SQL Injection có thể cho phép những kẻ tấn công thực hiện các thao tác như delete, insert, update… trên CSDL của ứng dụng, thậm chí là của server mà ứng dụng đó đang chạy. Lỗi SQL Injection thường xảy ra trên các ứng dụng web có dữ liệu được quản lý bằng DBMS như sql server, my Sql, DB2…

*Cách thức tấn công:*

*Cách phòng chống:*

* Làm sạch dữ liệu đầu vào: Lọc dữ liệu đầu vào sử dụng hai mô hình
* Mô hình danh sách cho phép- whitelist:liệt kê danh sách dữ liệu đầu vào hợp lệ
* Mô hình danh sách cấm-blacklist: mô hình xây dựng lên các mẫu đầu vào được cho là nguy hiểm và không chấp nhận các mẫu này (%,--,\\*,\*/,\_,[,@...).
* Xây dựng truy vấn cho mô hình tham số hóa: là mô hình xây dựng truy vấn có thể sử dụng để thay thế mô hình truy vấn động. Với mục đích xây dựng một lần, dùng nhiều lần (mỗi lần chỉ cần thay đổi tham số, tham số truyền vào lúc thực thi). Mô hình tham số hiện tại chỉ thực hiện trên các câu lệnh DML(select, insert, replace, update), create table còn các câu lệnh khác thì chưa được hỗ trợ.
* Chuẩn hóa dữ liệu: Mã hóa dữ liệu đầu vào dưới định dạng nào rồi gửi cho ứng dụng mà sau đó dữ liệu đó có thể được giải mã theo định dạng mà kẻ tấn công mong muốn.
* Sử dụng cơ chế lọc bổ sung như: Sử dụng tường lửa Web, sử dụng hệ thống phát hiện và ngăn chặn xâm nhập, sd tường lửa CSDL…Tăng cường cập nhật bản vá, hạn chế tối đa quyền người dùng, sử dụng mật mã để bảo vệ dữ liệu…

# 7. Tấn công CSRF: khái niệm, phân loại, hậu quả của tấn công CSRF, cách thức tấn công và phòng chống.

CSRF (Cross Site Request Forgery) là kĩ thuật tấn công bằng cách sử dụng quyền chứng thực của người sử dụng đối với 1 website khác. Các ứng dụng web hoạt động theo cơ chế nhận các câu lệnh HTTP từ người sử dụng, sau đó thực thi các câu lệnh này.

1. Tấn công DoS là gì? Chỉ ra một số dạng tấn công DoS.
2. Phân biệt DoS và DDoS. Tại sao khó chống lại được DDoS?
3. Tấn công Path Traversal là gì? Cách thức tấn công và phòng chống? Hậu quả của tấn công Path Traversal?
4. Quy trình xây dựng ứng dụng web an toàn.
5. Quy trình kiểm thử an toàn ứng dụng web.
6. Quy trình kiểm duyệt code an toàn ứng dụng web (Code review).
7. Xây dựng ứng dụng web an toàn: Các đầu mục công việc và cụ thể hóa trong Hệ thống quản lý (bài tập lớn).
8. Cho một phần nhỏ của hệ thống web, yêu cầu đưa ra các bước thực hiện để có thể khai thác một số lỗ hổng như SQL injection, XSS, CSRF, hiểu rõ bản chất được lỗ hổng và biện pháp khắc phục cho lỗ hổng của hệ thống đó.