ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA MẠNG MÁY TÍNH & TRUYỀN THÔNG

----------✸✸✸----------

BÁO CÁO CUỐI KỲ

Đề tài: Prevention Session Hijacking By One-Time Cookies

Giảng viên: **Nguyễn Ngọc Tự**

Nhóm: 12

Thành viên nhóm:

1. Nguyễn Đức Trung – 20520956
2. Nguyễn Trọng Nguyên – 20521677
3. Nguyễn Tú Ngọc – 20521665

Mã lớp: NT101.N11.ATCL

Thành phố Hồ Chí Minh, tháng 01 năm 2023

MỤC LỤC

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG

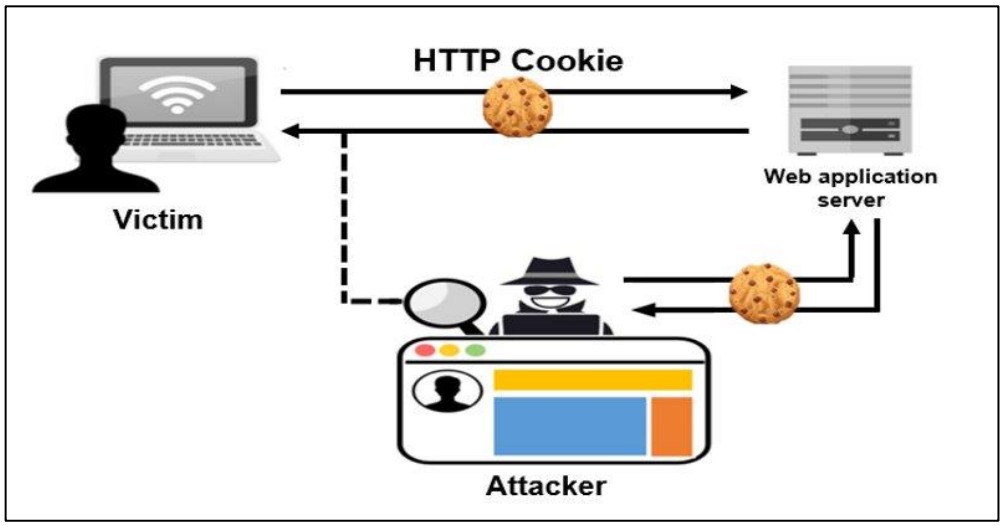
CHƯƠNG 1 MỞ ĐẦU

HTTP cookies (Hypertext Transfer Protocol Cookies) là một phần dữ liệu được chia sẻ giữa Web Application Server và người dùng để lưu trữ các thông tin trạng thái hoặc dữ liệu hoạt động của người dùng trên trình duyệt. Cookies thường được sử dụng trong các ứng dụng web để định danh người dùng và xác thực các phiên làm việc tương ứng. Vì thế việc đánh cắp cookies có thể dẫn đến việc người dùng bị chiếm quyền điều khiển phiên làm việc đã được xác thực.

Để chống lại hình thức tấn công này, nhóm chúng em đề xuất một hệ thống bảo vệ cookies an toàn và hiệu quả. Phương pháp mà nhóm sử dụng là One-Time Cookies nhằm chống lại mối đe dọa Cookies Hijacking và đồng thời thực hiện mã hóa các phần thông tin quan trọng trong cookies để đảm bảo tính toàn vẹn và bảo mật của phiên làm việc (session).

CHƯƠNG 2 TỔNG QUAN

2.1. Ngữ cảnh



Hình 1. Ngữ cảnh tấn công Session Hijacking

Khi nạn nhân thực hiện một phiên làm việc (session) với Web application server:

- Nạn nhân gửi yêu cầu (request) truy cập vào ứng dụng web và thực hiện thao tác login
- Server thực hiện việc xác thực danh tính người dùng và gửi trả về phản hồi (response) kèm theo cookies - được lưu trữ ở trình duyệt web. Cookies này sẽ chứa các thông tin quan trọng của nạn nhân.
- Khi này kẻ tấn công (attacker) có thể thực hiện đánh cắp cookies và mạo danh nạn nhân qua mặt server để đánh cắp thông tin cũng như chiếm phiên làm việc của nạn nhân.

2.2. Các bên liên quan

Mô hình gồm 3 bên liên quan

- **Người dùng:**

Người dùng (client) hoặc nạn nhân (victim) sẽ gửi các request lên web application server. Giả sử khi người dùng cần đăng nhập vào một ứng dụng web, các thông tin như username hay password sẽ được gửi lên server để xác thực. Sau khi xác thực

thành công, server sẽ gửi trả về cookies để tiếp tục phiên làm việc hiện tại của người dùng, không cần phải xác thực lại ở request tiếp theo.

- **Server:**

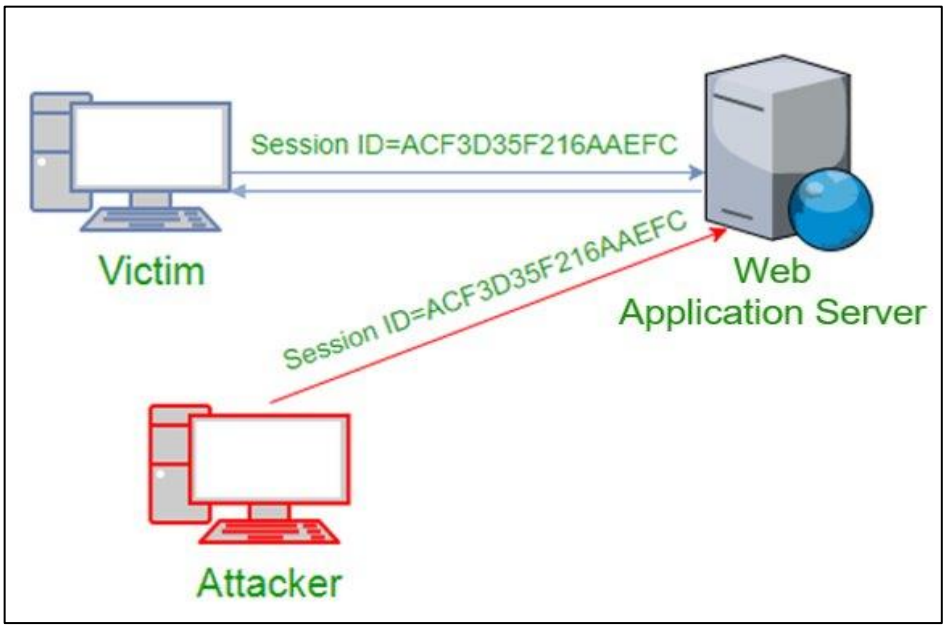
Server sẽ nhận các thông tin request từ phía người dùng, thực hiện xác thực và gửi trả về cookies kèm các thông tin dữ liệu cần thiết

- **Attacker:**

Là bên thứ 3, nằm giữa người dùng và server, thực hiện việc đánh cắp cookies của người dùng và chiếm phiên làm việc 1 cách trái phép

CHƯƠNG 3 TRIỂN KHAI MÔ HÌNH THỬ NGHIỆM

3.1. Mô phỏng tấn công



Hình 2. Mô hình tấn công Session Hijacking

Nhóm thực hiện mô phỏng lại quá trình Attacker thực hiện đánh cắp cookies của người dùng sau khi đã được xác thực từ phía Server và trả về cookies.

Quy trình triển khai như sau:

1. Nạn nhân thực hiện login vào web application server.
2. Server xác thực nạn nhân và gửi phản hồi về cho nạn nhân kèm theo cookies.

3. Attacker đánh cắp cookies bằng cách tấn công Session Hijacking, và mạo danh nạn nhân để login server.
4. Server lầm tưởng và thực hiện các request trong phiên làm việc đã được xác thực.

Demo phiên làm việc với cookies không được mã hóa:

Link video: https://drive.google.com/file/d/1ywdf4OzKgmpIpUcEixmJ9cD8A-18QxrE/view

3.2. Mô phỏng phòng chống

3.2.1. *Tài nguyên triển khai*

- **Server:**

Triển khai ứng dụng Web bằng ngôn ngữ Python với thư viện Flask cho phía backend

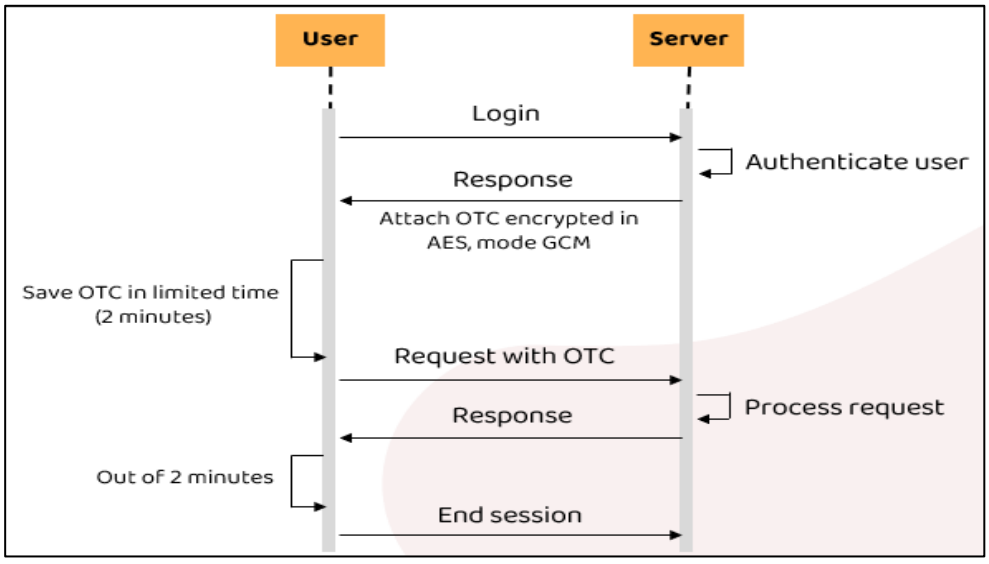
- **Client:**

Sử dụng như 1 máy Local đăng nhập vào Web application server

- **Attacker:**

Sử dụng các công cụ như Burpsuite và FoxyProxy để đánh cắp và chỉnh sửa cookies.

3.2.2. *Triển khai phòng chống Session Hijacking*



Hình 3. Mô hình phòng chống Session Hijacking

Nhóm triển khai mô hình phòng chống với với phương pháp **Generate One-time Cookies trong vòng 2 phút kết hợp Mã hóa Cookies**

Quy trình triển khai như sau:

1. Người dùng thực hiện thao tác login vào Web application server.
2. Server tiến hành xác thực các thông tin được gửi lên từ phía người dùng, và gửi lại cookies đã được mã hóa với thuật toán AES / mode GCM, ở đây phần key sẽ được lưu trữ bên phía Server.

```
def encrypt(self, raw):
    raw = self._pad(raw)
    iv = Random.new().read(AES.block_size)
    cipher = AES.new(self.key, AES.MODE_GCM, iv)
    return base64.b64encode(iv + cipher.encrypt(raw.encode()))
```

Hình 4. Mã hóa cookies bằng thuật toán AES / mode GCM

3. Khi đó cookies sẽ được generate sau mỗi 2 phút, buộc người dùng phải thực hiện thao tác login trở lại.
4. Điều này khiến cho attacker không đủ thời gian để giải mã cũng như sao chép cookies để giả mạo người dùng và chiếm quyền điều khiển phiên làm việc.

Demo phiên làm việc với cookies đã được mã hóa

Link Video: https://drive.google.com/file/d/13YcXaaxhnvPjgzeiBaNDkiiP-19s_g0W/view

CHƯƠNG 4 KẾT LUẬN

Với phương pháp nhóm đề xuất trên, hiện vẫn còn nhiều khuyết điểm. Bởi với các giao thức hiện tại như SSL/TLS đã mặc định mã hóa các phần thông tin cookies, các ứng dụng Web thường yêu cầu người dùng đăng nhập lại dù đã có đánh cắp được cookies. Vì thế việc đánh cắp cookies dường như không mang lại nhiều ý nghĩa. Ngoài ra việc cấu hình session time với khoảng thời gian 2 phút đem lại các trải nghiệm không tốt cho phía người dùng.

TÀI LIỆU THAM KHẢO

1. Prapty, R. T. , Md, S. A. , Hossain, S. , and Narman, H. S. (2020). **Preventing Session Hijacking using Encrypted One-Time-Cookies**. *2020 Wireless Telecommunications Symposium (WTS).* pp, 1-6. doi: 10.1109/WTS48268.2020.9198717. IEEE

2. Sathiyaseelan, A. M. , Joseph, V. , and Srinivasaraghavan, A. (2017). **A proposed system for preventing session hijacking with modified one-time cookies.** *2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC).* pp, 451-454. doi: 10.1109/ICBDACI.2017.8070882. IEEE

PHỤ LỤC

Bảng phân công công việc

Nguyễn Đức Trung 20520956	**Nguyễn Tú Ngọc 20521665**	**Nguyễn Trọng Nguyên 20521677**
- Lựa chọn đề tài - Tìm research paper - Source code web - Seedlab	- Source code web - Demo tấn công - Slide powerpoint - Viết report	- Demo phòng chống - Triển khai mã hóa cookie - Thuyết trình

Bảng 1. Bảng phân chia công việc