**Giả mạo yêu cầu phía máy chủ (SSRF)**

**Khái niệm:**

SSRF là một lỗ hổng web cho phép kẻ tấn công làm máy chủ thực thi yêu cầu đến 1 vị trí không mong muốn.

Trong 1 cuộc tấn công SSRF, kẻ tấn công có thể làm cho máy chủ chỉ được kết nối với những dịch vụ bên trong cơ sở hạ tầng. Trong trường hợp khác, hắn có thể bắt máy chủ kết nối đến hệ thống bất kì từ bên ngoài, làm rò rỉ dữ liệu quan trọng cũng như thông tin đăng nhập.

**Tác động của SSRF:**

1 cuộc tấn công SSRF thành công có thể dẫn đến những hành động trái phép hoặc truy cập dữ liệu trong tổ chức, ứng dụng có lỗ hổng hoặc hệ thống phía sau mà ứng dụng giao tiếp.

Cho phép kẻ tấn công thực thi lệnh

1 cuộc khai thác SSRF có thể làm cho những kết nối đến hệ thống thứ 3 từ bên ngoài bị tấn công nguy hiểm.

**Những cuộc tấn công SSRF phổ biến:**

Những cuộc tấn công SSRF thường khai thác những mối quan hệ đáng tin cậy để leo thang 1 cuộc tấn công từ ứng dụng có lỗ hổng và thực hiện hành động trái phép. Là mối quan hệ với máy chủ, hoặc những hệ thống phía sau trong cùng 1 tổ chức.

**Những cuộc tấn công SSRF chống lại máy chủ:**

Trong cuộc tấn công SSRF, kẻ tấn công làm cho ứng dụng yêu cầu http đến máy chủ lưu trữ ứng dụng, sẽ liên quan đến việc cung cấp 1 URL với máy chủ là 127.0.0.1 hay localhost.

Xem xét 1 ứng dụng cho phép người dùng xem trạng thái kho hàng của 1 sản phẩm đó trong cửa hàng. Để cung cấp thông tin kho, ứng dụng bắt buộc phải lấy những API liên quan đến sản phẩm và cửa hàng. API là dữ liệu chung để các hệ thống tương tác với nhau. Chức năng được thực thi bằng việc gửi URL đến máy chủ để lấy API qua yêu cầu http.

POST /product/stock HTTP/1.0

Content-Type: application/x-www-form-urlencoded

Content-Length: 118

stockApi=http://stock.weliketoshop.net:8080/product/stock/check%3FproductId%3D6%26storeId%3D1

Điều này sẽ làm cho máy chủ xử lí yêu cầu đối với URL chỉ định, truy suất trạng thái kho hàng và trả về cho người dùng.

Trong tình huống này, kẻ tấn công có thể điều chỉnh yêu cầu đến 1 URL cục bộ đến máy chủ.

POST /product/stock HTTP/1.0

Content-Type: application/x-www-form-urlencoded

Content-Length: 118

stockApi=http://localhost/admin

Ở đây máy chủ sẽ lấy API liên quan đến URL admin và trả về cho người dùng.

Tất nhiên kẻ tấn công chỉ có thể thăm URL: /admin trực tiếp mà không tìm thấy điều gì thú vị vì chức năng này yêu cầu xác thực.

Nhưng việc yêu cầu url /admin từ chính máy cục bộ sẽ dễ dàng leo thang đặc quyền. Ứng dụng sẽ cấp toàn quyền truy cập vào chức năng quản trị vì yêu cầu có nguồn gốc từ 1 vị trí đáng tin cậy.

Tại sao ứng dụng lại hành xử theo cách này và tin tưởng ngầm những yêu cầu đến từ máy cục bộ?

Việc kiểm soát truy cập chỉ thực thi trước máy chủ của ứng dụng

Để khắc phục sự cố, ứng dụng có thể cho phép truy cập trang quản trị không cần xác thực với người dùng đến từ máy cục bộ. Điều này giúp cho nhà quản trị khắc phục hệ thống khi bọn họ mất thông tin đăng nhập. Giả định này chỉ áp dụng cho người dùng đáng tin cậy nhất.

Giao diện trang quản trị có thể lắng nghe cổng khác ứng dụng chính, vì vậy người dùng không thể trực tiếp truy cập.

🡪 Rất nhiều mối quan hệ đáng tin cậy, trong đó có những yêu cầu từ máy cục bộ sẽ được xử lí khác với những yêu cầu thông thường, có thế gây ra lỗ hổng SSRF nghiêm trọng.

**Những cuộc tấn công SSRF chống lại những hệ thống phía sau khác:**

Một mối quan hệ tin cậy khác để làm giả yêu cầu là máy chủ ứng dụng có thể tương tác với những hệ thống phía sau khác mà người dùng không vào trực tiếp được.

Những hệ thống này thường có địa chỉ IP riêng. Vì chúng được bảo vệ bởi cấu trúc liên kết mạng, nên có lớp bảo mật yếu hơn.

Trong rất nhiều trường hợp, những hệ thống này có thể truy cập chức năng nhạy cảm mà không cần xác thực.

Giả sử trang quản trị ở URL sau: https://192.168.0.68/admin. Kẻ tấn công có thể khai thác lỗ hổng ssrf bằng việc gửi di yêu cầu sau:

POST /product/stock HTTP/1.0

Content-Type: application/x-www-form-urlencoded

Content-Length: 118

stockApi=http://192.168.0.68/admin

**Phá vỡ hệ thống phòng thủ SSRF thông thường:**

**SSRF lọc đầu vào dựa trên blacklist:**

Một số ứng dụng chặn đầu vào chứa tên máy chủ như 127.0.0.1 và localhost hoặc Urls nhạy cảm như /admin. Bạn có thể bypass bằng những kĩ thuật sau:

* Dùng 1 địa chỉ IP thay thế cho 127.0.0.1 như 2130706433, 017700000001, or 127.1
* Đăng kí tên miền của bạn trỏ đến 127.0.0.1. Bạn có thể sử dụng spoofed.burpcollaborator.net
* Làm rối loạn những chuỗi bị chặn bằng URL mã hóa (double encoding) hoặc những biến thể khác.

**SSRF lọc đầu vào dựa trên whitelist:**

Một vài ứng dụng chỉ cho phép đầu vào phù hợp, bắt đầu với hoặc chứa những giá trị được phép. Bạn có thể phá vỡ nó bằng việc khai thác sự mâu thuẫn trong phân tích cú pháp Url.

* Bạn có thể nhúng thông tin đăng nhập trong Url trước tên máy chủ bằng việc sử dụng @.

https://expected-host@evil-host

* Bạn có thể sử dụng kí tự # để làm gián đoạn Url.

https://evil-host#expected-host

* Bạn có thể mã hóa Url hoặc kết hợp các kĩ năng với nhau.

**Bỏ qua bộ lọc SSRF bằng chuyển hướng:**

Chúng ta có thể bỏ qua bất cứ bộ lọc nào bằng việc khai thác lỗ hổng chuyển hướng.

Trong ví dụ ssrf trước, giả sử url do người dùng cung cấp được xác thực để ngăn chặn tấn công ssrf. Tuy nhiên, những urls của ứng dụng chứa lỗ hổng chuyển hướng. Bạn có thể tạo 1 url phù hợp với bộ lọc nhưng lại chuyển hướng đến hệ thống bạn muốn.

Ví dụ: /product/nextProduct?currentProductId=6&path=http://evil-user.net

Url sau sẽ chuyển hướng đến http://evil-user.net

Bạn có thể tận dụng lỗ hổng này để bỏ qua bộ lọc và khai thác ssrf như sau:

POST /product/stock HTTP/1.0

Content-Type: application/x-www-form-urlencoded

Content-Length: 118

stockApi=http://weliketoshop.net/product/nextProduct?currentProductId=6&path=http://192.168.0.68/admin

**Lỗ hổng SSRF mù:**

Lỗ hổng này sinh ra khi ứng dụng yêu cầu 1 url nhưng không trả về phản hồi.

Tác động của nó thường nhỏ hơn lỗ hổng thường vì không thể truy cập vào dữ liệu nhạy cảm từ hệ thống phía sau, nhưng trong 1 số tình huống nó có thể được khai thác để kích hoạt đoạn mã từ xa.

Cách tốt nhất để phát hiện lỗ hổng mù là dùng phương pháp OAST. Nó liên quan đến việc gửi 1 yêu cầu đến 1 hệ thống bên ngoài bạn kiểm soát được để theo dõi sự tương tác mạng với hệ thống đó.

**Những cách tấn công SSRF khác:**

Lỗ hổng SSRF rất dễ nhận ra vì những yêu cầu của ứng dụng có chứa Url.Một vài ví dụ sẽ khó nhận ra:

**Url rút ngắn trong yêu cầu:**

Thỉnh thoảng ứng dụng chỉ đặt tên máy chủ hoặc 1 phần của Url trong yêu cầu. Giá trị được gửi đi sẽ kết hợp với url đầy đủ trong máy chủ. Đây có thể là 1 lỗ hổng rất tiềm năng nhưng việc khai thác ssrf sẽ bị giới hạn vì ta không kiểm soát được toàn bộ Url.

**Url trong những định dạng dữ liệu:**

Một số ứng dụng xử lí những url trong những định dạng dữ liệu. Ví dụ rõ ràng nhất là định dạng XML được sử dụng rộng rãi trong ứng dụng web để truyền dữ liệu từ máy khách sang máy chủ. Khi ứng dụng chấp nhận dữ liệu ở định dạng xml và phân tích nó, có thể dính lỗ hổng XXE hoặc SSRF qua XXE.

**SSRF qua Referer header:**

Một số ứng dụng lấy Url từ Referer để yêu cầu.

🡪 Lỗ hổng SSRF.