**BỘ THÔNG TIN TRUYỀN THÔNG**



**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

----------

**Kỹ thuật giấu tin trong văn bản**

**Sử dụng CFG để giấu tin vào văn bản và truyền tin**

**LAB: cfg-lab-bruteforce**

**Sinh viên:** Lê Văn Tuấn

**Mã sinh viên:** B21DCAT208

**Bộ môn:** Kỹ thuật giấu tin

**Giảng viên hướng dẫn:** PGS.TS. Đỗ Xuân Chợ

**Hà Nội – 2025**

1. **Mục đích**

Giúp sinh viên hiểu hơn về thuật toán sử dụng phương pháp giấu tin CFG (Context Free Grammar), và vận dụng các thức tấn công người đứng giữa (MITM) để thực hiện bắt tấn công và giải mã thông điệp

1. **Yêu cầu đối với sinh viên**

Có kiến thức về mật mã học cơ sở và kĩ thuật giấu tin, sử dụng thành thạo linux

1. **Nội dung lý thuyết**

Văn phạm phi ngữ cảnh là một tập hợp hữu hạn các biến (còn được gọi là các ký hiệu chưa kết thúc), mỗi biến biểu diễn một ngôn ngữ. Ngôn ngữ được biểu diễn bởi các biến được mô tả một cách đệ quy theo thuật ngữ của một khái niệm khác gọi là ký hiệu kết thúc. Quy tắc quan hệ giữa các biến gọi là luật sinh. Mỗi luật sinh. Mỗi luật sinh có dạng một biết, ở vế trái sinh ra một chuối có thể gồm biến lẫn các ký hiệu là văn phạm G (V, T, P, S), trong đó:

* V là tập hợp hữu hạn các biến
* T là tập hợp hữu hạn các ký tự kết thúc V ∩ T = ∅
* P là tập hợp hữu hạn các luật sinh mà mỗi luật sinh có dạng A → α (với A là biến và α là chuỗi các ký hiệu ∈ (V∪T)\*)
* S là một biến đặc biệt gọi là ký hiệu bắt đầu văn phạm.

Các quy tắc sau chỉ ra cách để tạo ra một chuỗi (gồm các kí hiệu kết thúc) từ một CFG cho trước:

* Sử dụng 1 kí hiệu bắt đầu để khởi đầu. Chọn một luật có kí hiệu khởi đầu ở bên trái và chọn một kí hiệu ở phần bên phải của luật để thay thế kí hiệu khởi đầu này. Kí hiệu được sử dụng sẽ là một thành tố của văn bản sẽ được tạo sao này.
* Chọn 1 kí hiệu chưa kết thúc trong văn bản, tìm một luật có kí hiệu chưa kết thúc này ở bên trái và thay thế kí hiệu này bằng một kí hiệu ở phần bên phải của luật.
* Lặp lại bước 2 cho đến khi văn bản chỉ còn lại toàn các kí hiệu kết thúc.

Sau khi giấu tin xong sẽ gửi văn bản đến một máy client sau đó client giải mã thông điệp.

1. **Nội dung thực hành**

Add module file lab:

Khởi động bài lab:

Vào terminal gõ:

labtainer -r cfg-lab-bruteforce

(Chú ý: Sinh viên sử dụng mã sinh viên của mình để nhập thông tin email người thực hiện bài lab khi có yêu cầu để sử dụng khi chấm điểm)

**Nhiệm vụ 1: Giấu tin**

Nhiệm vụ của user1 là sẽ giấu mật khẩu file secret và gửi văn bản đã giấu tin sang cho máy user2. Để giấu tin, chạy lệnh:

encrypt

**Nhiệm vụ 2: Cài đặt môi trường**

Bây giờ ta cần chạy dịch vụ ftp server để user1 có thể gửi thông điệp cho user2. Ở máy user2 chạy:

python3 ftp\_server.py

Đồng thời cùng lúc này attacker sẽ khởi động dịch vụ arpspoof để thực hiện ARP spoofing, giả mạo địa chỉ MAC của user2. Ở máy attacker chạy đồng thời arpspoof và attack.py:

sudo arpspoof -i eth0 -t 192.168.0.10 192.168.0.20

sudo python3 attack.py

**Nhiệm vụ 3: Nghe lén**

Ở máy user1, tiến hành gửi file message.txt cho user2 (username là user và password là 123456):

ftp 192.168.0.20

put message.txt

Cùng lúc đó ở bên máy attacker, ta có thể thấy được thông tin của user2 đã được ghi lại ở giao diện attack.py

**Nhiệm vụ 4: Nhận file**

Sau khi có được thông tin của user2, tại máy attacker ta tiến hành ftp và get các file cần thiết về để giải mã thông điệp. Do người dùng đã ẩn đi các file cấu hình cfg và file giấu tách tin nên hacker chỉ có thể lấy văn bản giấu tin.

ftp 192.168.0.20

get message.txt

**Nhiệm vụ 5: Thống kê**

Như đã giải thích ở trên, file cấu hình cfg và file giấu tách tin đã được ẩn đi, nên hacker không thể sử dụng file cấu hình để thực hiện truy xuất mật khẩu, thay vào đó ở bài này, hacker sẽ sử dụng phương pháp thống kê và vét cạn để lấy mật khẩu.

Cụ thể, nếu cấu trúc cfg đơn giản. Các từ trong từng câu bị lặp đi lặp lại nhiều lần, thì kẻ tấn công có thể dễ dàng khoanh vùng được phạm vi các từ có thể có trong cấu hình. Nếu câu đủ dài hoặc số lượng văn bản giấu tin nhiều. Ví dụ với một văn bản như dưới đây:

*“Dog jumps slow slowly. Cat runs quick slowly. Dog jumps slow fast.”*

Với một file cấu hình đơn giản, attacker có thể thống kê được:

A1: Dog, Cat

A2: jumps, runs

…

Chỉnh sửa file grammar\_stats.py trên máy attacker để tự động quá trình thống kê, sau đó chạy:

python3 grammar\_stats.py

**Nhiệm vụ 6: Bruteforce**

Sau khi đã thống kê được các từ xuất hiện trong từng non-terminal (a1, a2, a3, a4) ta sẽ thực hiện bruteforce để tìm được danh sách từ khả dụng từ file message.txt

Khi bruteforce, chương trình sẽ lần lượt gán các kí tự kết thúc cho các giá trị nhị phân 0 và 1, sau đó đối chiếu với message.txt và ghép lại, nếu là từ hợp lệ thì lưu vào wordlist.txt

Sửa các quy tắc ở file bruteforce.py để đảm bảo cấu hình đúng cho việc bruteforce và chạy:

python3 bruteforce.py

**Nhiệm vụ 7: Wordlist**

Sau khi bruteforce xong, các từ hợp lệ sẽ được ghi vào file wordlist.txt, ở đây sẽ chứa từ mà văn bản giấu

cat wordlist.txt

**Nhiệm vụ 8: Bruteforce password**

Sử dụng file wordlist vừa thu được để bruteforce mật khẩu file secret

chmod +x secret

python3 bruteforce\_secret.py

**Nhiệm vụ 9: Secret**

Thu được mật khẩu sau khi bruteforce xong, sử dụng để mở file secret

Bài thực hành hoàn thành khi sinh viên chạy thành công file secret

**Kết thúc bài thực hành**

* Kiểm tra checkwork

checkwork

* Trên terminal đầu tiên sử dụng câu lệnh sau để kết thúc bài lab

stoplab

Khởi động lại bài lab

labtainer -r cfg-lab-bruteforce