# BÀI 6.
# TẤN CÔNG TỪ CHỐI DỊCH VỤ

Nguyễn Đức Toàn,
Viện Công nghệ thông tin và Truyền thông,
Đại học Bách khoa Hà Nội

1

1

## Nội dung

- Khái niệm chung
- Một số kỹ thuật tấn công DoS điển hình
- Phòng chống tấn công DoS

2

2

# 1. DoS LÀ GÌ?

Nguyễn Đức Toàn,
Viện Công nghệ thông tin và Truyền thông,
Đại học Bách khoa Hà Nội



3

## DoS là gì?

- Denial of Service: Ngăn cản dịch vụ cung cấp tới người dùng bình thường
- Cách thức thực hiện: Gửi lượng dữ liệu đủ lớn làm quá tải nút thắt cổ chai (bottleneck) của hệ thống
  - Lưu lượng tấn công lớn hơn băng thông của mục tiêu, hoặc
  - Số lượng gói tin lớn hơn khả năng xử lý của mục tiêu
  - Có thể kết hợp khai thác lỗ hổng phần mềm cung cấp dịch vụ

→ Các kỹ thuật DoS tìm cách khuếch đại lưu lượng

- DDoS-Distributed DoS: tấn công được thực hiện bởi nhiều nguồn khác nhau:
  - Thường sử dụng botnet
- Có thể xảy ra trên tất cả các tầng của hệ thống mạng



4

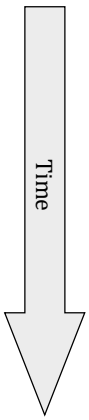
# Phân loại

- Tấn công vật lý: gây ra sự cố nguồn điện, kết nối mạng
- Tấn công băng thông: gửi liên tục một lượng lớn các gói tin làm tràn ngập băng thông của nạn nhân
  - Thường sử dụng các kỹ thuật khuếch đại
  - Ví dụ: Ping of Death, Smurf attack, DNS Amplification, UDP Flood
- Tấn công tài nguyên hệ thống: Gửi một lượng lớn yêu cầu làm cạn kiệt tài nguyên của nạn nhân
  - Thường khai thác điểm yếu của giao thức
  - Ví dụ: Tear drop, TCP SYN Flood, HTTP Flood, DHCP Starvation
- Tấn công dựa trên khai thác lỗ hổng phần mềm
  - Buffer Overflow
  - Integer Overflow
  - Format String

5

5

# Sự phát triển của các hình thức DoS




- Point-to-point DoS
  - TCP SYN floods, Ping of death, etc..
- Reflection/Amplification DoS
- Coordinated DoS
- Multi-stage DDoS
- P2P botnets
- Amplification attacks

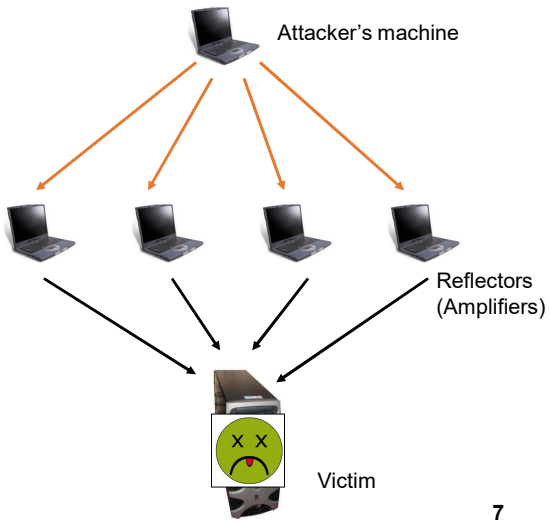
Sử dụng botnet

6

6

# Reflection/Amplification DoS

1. Kẻ tấn công gửi các gói tin giả mạo nạn nhân tới mạng khuếch đại
2. Mạng khuếch đại gửi dữ liệu trả lời cho nạn nhân
3. Nạn nhân bị đánh sập do phải xử lý lượng dữ liệu cực lớn

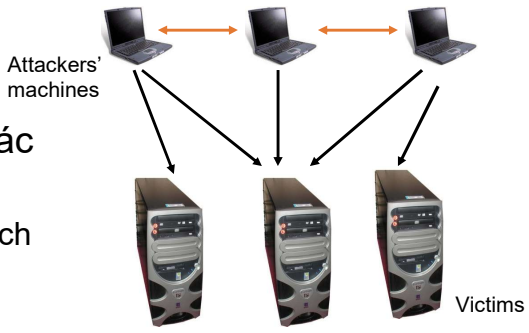



7

7

# Coordinated DoS

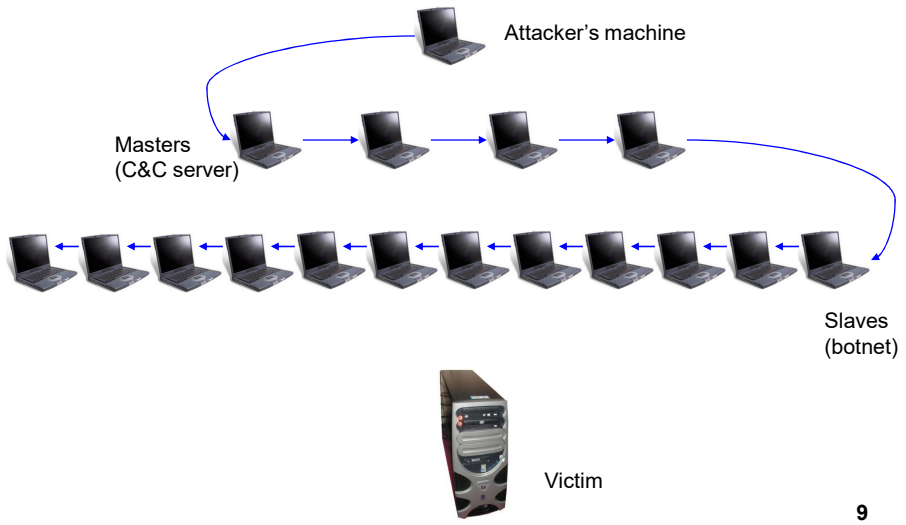
- Hình thức mở rộng của DoS
- Phối hợp nhiều nguồn tấn công khác nhau
  - Thường sử dụng dịch vụ IRC




8

8

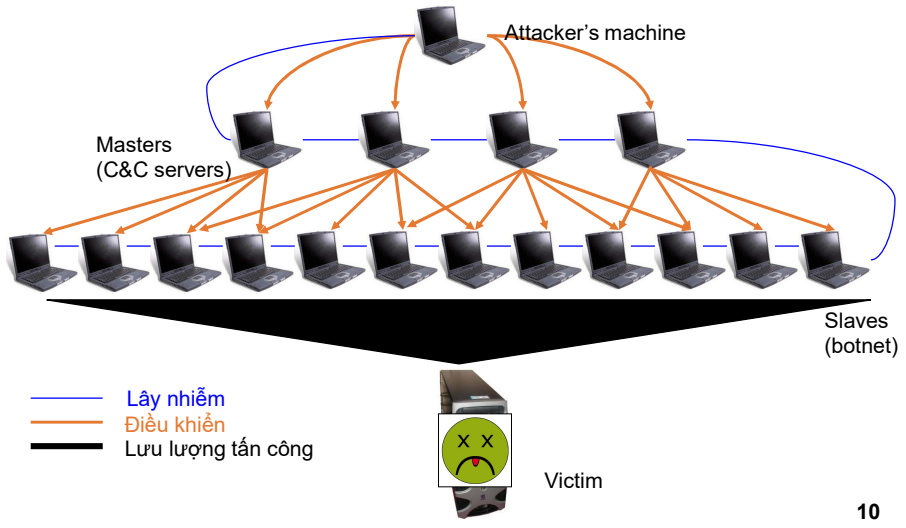
# Triển khai DDoS




9

9

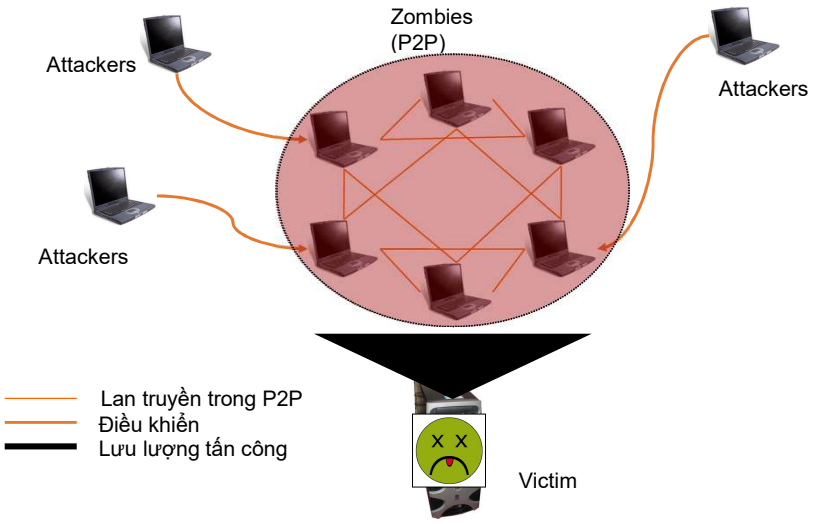
# Triển khai DDoS




10

10

# Triển khai DDoS sử dụng P2P botnet




11

11

# Distributed Reflection DoS (DRDoS)

- Relector: nút mạng có khả năng gửi hồi đáp khi nhận được thông điệp yêu cầu
  - Trên lý thuyết, tất cả các giao thức có hồi đáp đều có thể lợi dụng
- DRDoS gửi yêu cầu tới reflector với địa chỉ nguồn là địa chỉ nạn nhân
  - Reflector gửi thông điệp hồi đáp cho nạn nhân
- DrDoS thường sử dụng các giao thức mà thông điệp hồi đáp có kích thước lớn hơn nhiều thông điệp yêu cầu

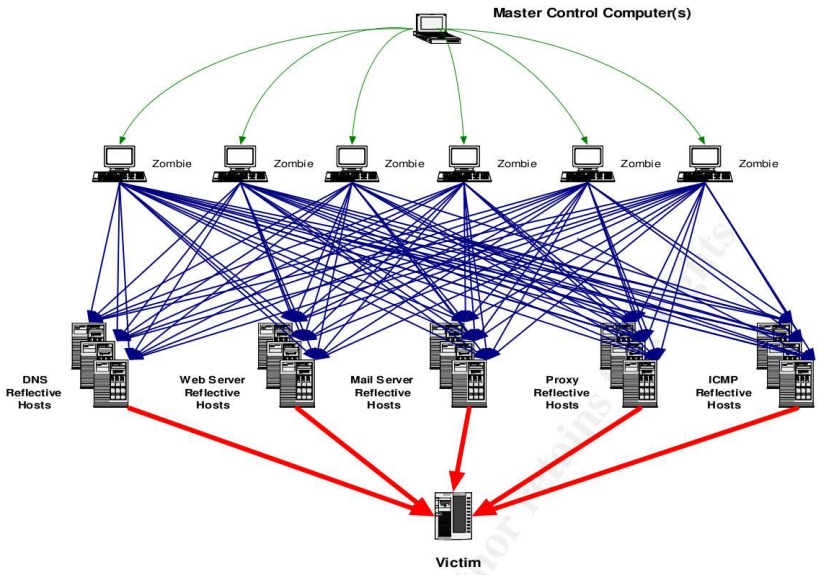
→khuếch đại lưu lượng

- Tại sao DRDOS nguy hiểm?
  - Che giấu nguồn tấn công
  - Không cần đòi hỏi số lượng bot lớn

12

12

# DRDoS




13

13

# Tại sao DoS rất khó phòng chống?

- Kỹ thuật tấn công đơn giản
- Mạng Internet không được thiết kế để chống lại tấn công DoS
- Dễ dàng để xâm nhập và điều khiển máy tính của người dùng đầu cuối
  - 2010: BredoLab(30tr. bot), Mariposa(12tr.), Conficker(10tr.)
  - Xu hướng mới: sử dụng các thiết bị IoT (VD: Mirai-300K)
- Rất khó phân biệt lưu lượng tấn công và lưu lượng người dùng thông thường
- Thiếu sự phối hợp giữa các ISP
- Rất khó để triển khai các biện pháp phòng chống

14

14

# Một số cuộc tấn công DoS điển hình

| Thời gian | Mục tiêu | Lưu lượng | Kỹ thuật |
|---|---|---|---|
| 03/2013 | Spamhaus | ~300Gbps | DNS Amp. Attack |
| 02/2014 | Một công ty Châu Âu | ~400Gbps | NTP Amp.Attack |
| 11/2014 | Một số website tại Hồng Kông | ~500Gbps | DNS Amp. Attack |
| 12/2015 | BBC | ~600Gbps | DNS Amp. Attack |
| 09/2016 | Blog của Brian Krebs | 620-660Gbps | SYN, GET, POST Flooding |
| 09/2016 | OVH Công ty hosting tại Pháp | ~1Tbps | Multiple type |

Xem thêm:

- pic.twitter.com/XmlwAU9JZ6
- Kaspersky DDOS intelligence report
- Verisign DDoS Trends Report

15

15

# 2. MỘT SỐ KỸ THUẬT TẤN CÔNG DOS

Nguyễn Đức Toàn,
Viện Công nghệ thông tin và Truyền thông,
Đại học Bách khoa Hà Nội

16

16

## 2. MỘT SỐ KỸ THUẬT TẤN CÔNG DOS

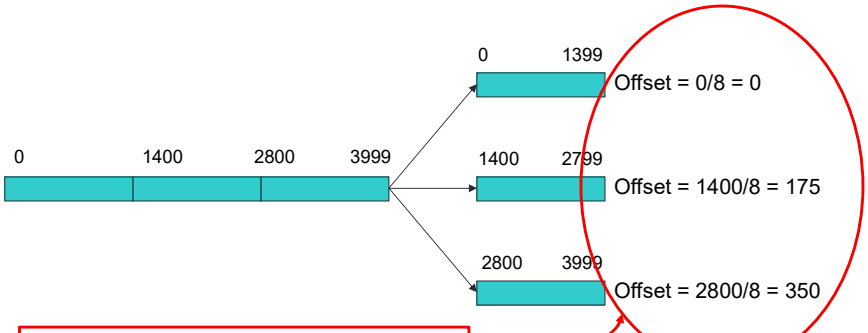
Nguyễn Đức Toàn,
Viện Công nghệ thông tin và Truyền thông,
Đại học Bách khoa Hà Nội

17

17

## Teardrop

- Lợi dụng cơ chế phân mảnh của giao thức IP
  - Offset cho biết vị trí của mảnh tin trong gói tin ban đầu

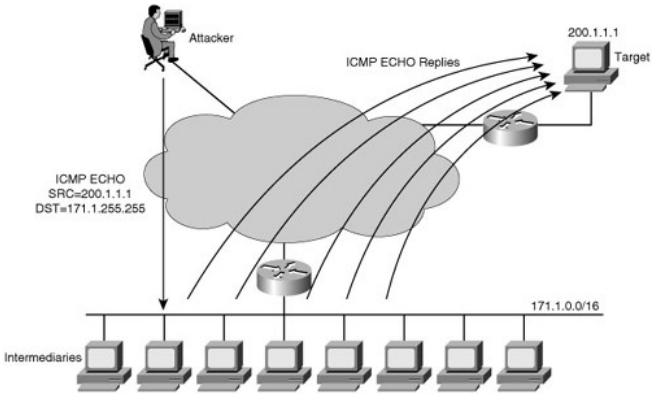



18

18

# Tấn công lợi dụng giao thức ICMP

- Ping of Death: gửi liên tục các gói tin ICMP có kích thước tối đa (xấp xỉ 64 KB)
- Smurf attack



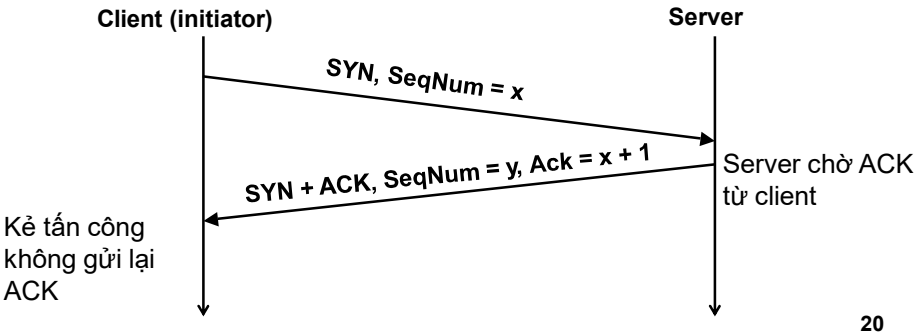

19

19

# TCP SYN Flooding

- Kẻ tấn công gửi hàng loạt gói tin SYN với địa chỉ nguồn là các địa chỉ IP giả
- Server gửi lại SYN/ACK, chuẩn bị tài nguyên để trao đổi dữ liệu, chờ ACK trong thời gian time-out

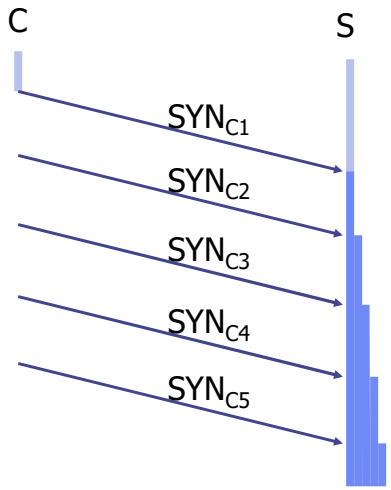
→ tấn công thành công nếu trong thời gian time-out làm cạn kiệt tài nguyên của ứng dụng, máy chủ vật lý




20

20

# Low rate TCP SYN Flood




- Lợi dụng: sau khi gửi gói tin SYN/ACK để chấp nhận kết nối, nạn nhân lưu giữ trạng thái kết nối trong hàng đợi
  - Đặt time-out
- Kỹ thuật tấn công: gửi số lượng gói tin SYN đủ lớn trong thời gian time-out để làm đầy hàng đợi. Kích thước hàng đợi mặc định:
  - Linux 1.2.x: 10
  - WinNT 4.0: 6
  - FreeBSD 2.1.5: 128

21

21

# Phòng chống LR SYN Flood

- Giải pháp tồi: tăng kích thước hàng đợi, giảm thời gian chờ time-out
- Giải pháp khác: lọc theo địa chỉ IP
  - Nếu tồn tại nút mạng đang sử dụng địa chỉ IP giả mạo trong gói tin SYN, khi nhận gói tin SYN/ACK, nút mạng này gửi gói tin RST tới nạn nhân

  → nạn nhân xóa kết nối khỏi hàng đợi

  →kẻ tấn công sử dụng địa chỉ IP chắc chắn chưa được sử dụng

  →Giải pháp phòng chống: bỏ qua các gói tin SYN có địa chỉ IP nguồn là địa chỉ chưa được sử dụng
- Đánh giá tính hiệu quả?

22

22

# SYN cookie

- Giải pháp hiệu quả hơn để chống LR SYN Flood
- Server không lưu giữ trạng thái kết nối
- Gửi gói tin SYN/ACK với giá trị Seq# là SYN cookie
  - T (5bit): bộ đếm 5 bit tăng sau mỗi 64s(chống tấn công phát lại)
  - MSS(3 bit): giá trị MSS được sử dụng
  - L = $MAC_{key}$(SAddr, SPort, DAddr, DPort, $SN_C$, T)
  - $SN_C$ : Seq# trong gói tin SYN gửi từ client
- Client thông thường gửi lại: Ack# = SYN Cookie + 1
- Server xác thực: (SYN Cookie) xor (SYN Cookie + 1)

→chỉ khởi tạo socket và cấp phát tài nguyên để xử lý kết nối khi xác thực thành công

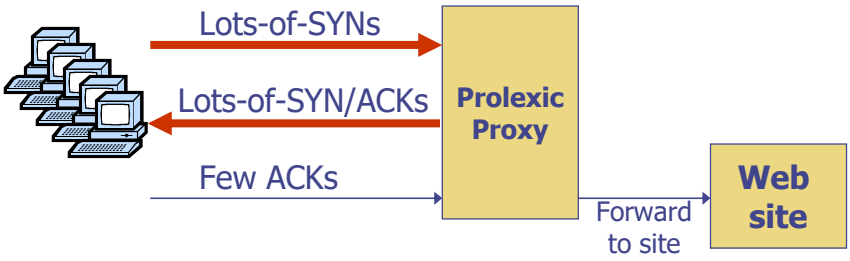
- Không hiệu quả để chống Massive Connection Flood
- Seq# dễ đoán hơn

23



# Phòng chống Massive SYN Flood

- Sử dụng Content Delivery Network(Prolexic, CloudFlare)
- Ý tưởng: chỉ chuyển tiếp các kết nối TCP đã thiết lập tới hệ thống

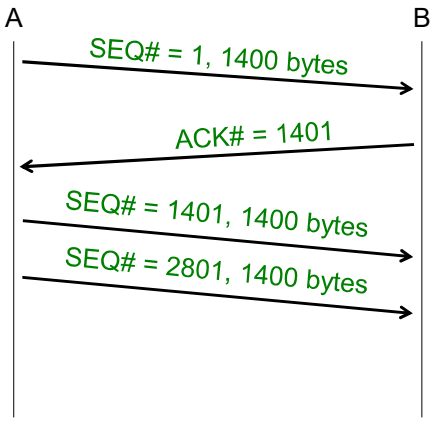



- Tấn công TCP Connection Flood?

24

# Kỹ thuật tấn công opt-ack

- Lợi dụng cơ chế kiểm soát tắc nghẽn của TCP: "Càng nhiều ACK báo nhận, tốc độ gửi càng cao"

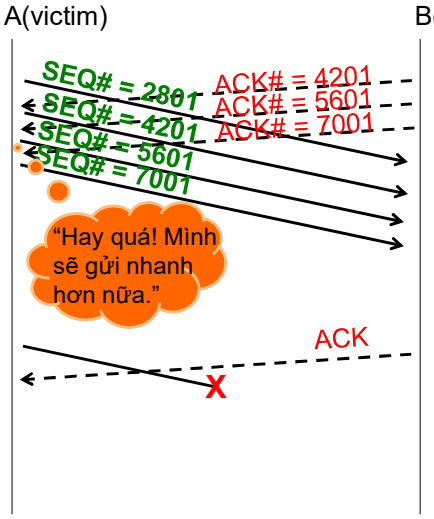



Nếu B "thuyết phục" được A gửi với tốc độ cao nhất có thể thì A tự tấn công DoS chính mình

25

25

# Kỹ thuật tấn công opt-ack




Nếu B đoán được giá trị SEQ mới nhất và thời điểm A sẽ gửi, B có thể gửi ACK sớm hơn.(Tại sao cần đoán đúng?)

Thậm chí, gói tin bị mất do tắc nghẽn cũng không ảnh hưởng tới kịch bản tấn công, miễn là B có thể gửi đúng ACK

26

26

# Hệ số khuếch đại

- Kỹ thuật tấn công opt-ack có khả năng khuếch đại lưu lượng tấn công do gói tin ACK có kích thước nhỏ hơn nhiều so với gói tin mang dữ liệu mà nạn nhân phải gửi
- Hơn nữa, TCP sử dụng ACK tích lũy
- Số lượng gói tin ACK lớn nhất mà kẻ tấn công có thể gửi:

$$\frac{Băng\ thông\ của\ attacker(bps)}{8 \times (14\ + 20 + 20)}$$

Tiêu đề Ethernet ← 14

Tiêu đề IP ← 20

Tiêu đề TCP → 20

- Lưu lượng lớn nhất nạn nhân gửi khi nhận được 1 ACK

$$\frac{Kích\ thước\ cửa\ sổ\ gửi}{MSS} \times (14 + 20 + 20 + MSS)$$

27



# Hệ số khuếch đại – Ví dụ

- Mỗi gói tin ACK kẻ tấn công gửi: 54 byte
- Lưu lượng nạn nhân phải gửi:
  - Kích thước cửa sổ mặc định: 65.536 byte
  - Kích thước MSS mặc định: 1460

$$\frac{65536}{1460} \times (14 + 20 + 20 + 1460) \approx 68.000\ bytes$$

- Hệ số khuếch đại

$$\frac{Kích\ thước\ cửa\ sổ\ gửi \times (14 + 20 + 20 + MSS)}{MSS \times (14 + 20 + 20)}$$

- Với thông số trên, hệ số khuếch đại ≈ 1258 lần
- Hệ số khuếch đại tăng lên nếu hai bên thỏa thuận sử dụng giá trị MSS nhỏ nhất có thể là 88
- Hệ số khuếch đại tăng nếu nạn nhân hỗ trợ cơ chế mở rộng cửa sổ(window scaling)

28

# Phòng chống tấn công opt-ack

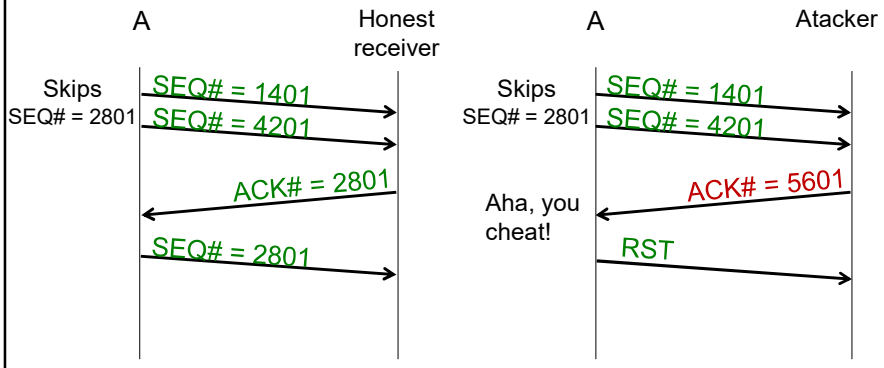
- Sử dụng giá trị thử thách ngẫu nhiên(challenge nonces) trong mỗi gói tin gửi đi. Yêu cầu client phải gửi ACK với giá trị đáp ứng
  - Không khả thi(Tại sao?)
- Hạn chế băng thông cho mỗi liên kết TCP
  - Không hiệu quả(Tại sao?)
- Thiết lập lại kết nối nếu gửi ACK ngoài cửa sổ
  - Làm tăng nguy cơ tấn công RTS Injection → không phù hợp
- Tạm giữ, không gửi đi một gói tin ngẫu nhiên

29

29

# Phòng chống tấn công opt-ack

- Tạm giữ, không gửi đi một gói tin ngẫu nhiên
  - Nếu bên gửi là bình thường, không gửi ACK báo nhận
  - Nếu bên gửi là tấn công, gửi ACK báo nhận thành công
  - Đánh giá giải pháp?

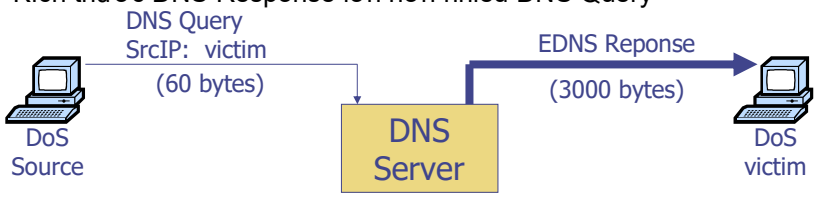



30

30

# DNS Amplification

- Lợi dụng:
  - DNS sử dụng giao thức UDP không cần thiết lập kết nối
  - Kích thước DNS Response lớn hơn nhiều DNS Query




- 2006: 580 nghìn DNS resolver miễn phí trên Internet
- 2013: 21.7 triệu DNS resolver miễn phí
- Thực hiện tương tự với các dịch vụ: NTP(x557), SNMPv2(x6.3), NetBIOS(3.8), SSDP(x30.8)…

31

31

# Phát hiện và giảm thiểu

- Hạn chế hoạt động Open DNS Resolver
  - Chỉ trả lời các truy vấn xuất phát từ trong mạng
  - Hạn chế số lượng thông điệp DNS Response gửi tới 1 client
- Chặn các truy vấn có địa chỉ IP không nằm trong mạng

32

32

# Các kỹ thuật thực hiện HTTP Flood

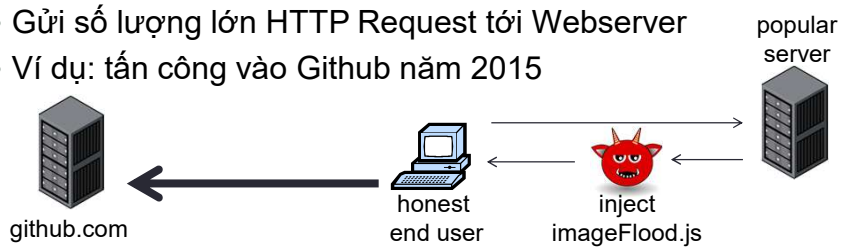
- Basic HTTP Floods: gửi yêu cầu truy cập liên tục tới các trang giống nhau
- Randomized HTTP Floods: gửi yêu cầu truy cập tới các trang một cách ngẫu nhiên
  - Cache-bypass HTTP Floods: sử dụng các kỹ thuật vượt qua các cơ chế cache trên máy chủ
  - WordPress XMLRPC Floods: lợi dụng có chế pingback trên WordPress để thực hiện kỹ thuật tấn công phản hồi

33

33

# Randomized HTTP Flood

- Gửi số lượng lớn HTTP Request tới Webserver
- Ví dụ: tấn công vào Github năm 2015




imageFlood.js

```
function imgflood() {
  var TARGET = 'victim-website.com/index.php?'
  var rand = Math.floor(Math.random() * 1000)
  var pic = new Image()
  pic.src = 'http://'+TARGET+rand+'=val'
}
setInterval(imgflood, 10)
```

34

34

# Randomized HTTP Floods – Ví dụ

```
75.118.29.205 - - [20/Jan/2014:19:32:06 -0500] "GET /?458739416183768700 HTTP/1.1" 2
173.245.56.201 - - [20/Jan/2014:19:32:06 -0500] "GET /?458726993617499500 HTTP/1.1"
79.19.41.22 - - [20/Jan/2014:19:32:06 -0500] "GET /?458741338856272200 HTTP/1.1" 20
190.121.64.3 - - [20/Jan/2014:19:32:06 -0500] "GET /?458722169268652700 HTTP/1.1" 2
186.54.141.146 - - [20/Jan/2014:19:32:06 -0500] "GET /?458741274224646000 HTTP/1.1"
```

```
83.21.152.106 - - [20/Jan/2014:19:40:08 -0500] "GET /?s=wikipedia HTTP/1.1"
179.9.92.22 - - [20/Jan/2014:19:40:08 -0500] "GET /?s=joy HTTP/1.1" 200 440
186.105.238.41 - - [20/Jan/2014:19:40:08 -0500] "GET /?s=santander HTTP/1.1
188.49.39.166 - - [20/Jan/2014:19:40:08 -0500] "GET /?s=news HTTP/1.1" 200
190.67.180.172 - - [20/Jan/2014:19:40:08 -0500] "GET /?s=gov HTTP/1.1" 200
200.94.168.60 - - [20/Jan/2014:19:40:08 -0500] "GET /?s=faith HTTP/1.1" 200
```

35

35

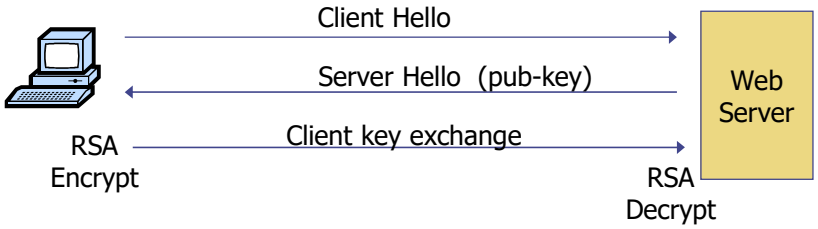
# Tại sao Layer 7 DoS khó bị phát hiện

- Layer 7 DoS dựa trên giao thức HTTP, thực hiện tấn công qua các kết nối TCP đã được thiết lập:
  - Không thể sử dụng các kỹ thuật phát hiện và ngăn chặn của TCP
  - Địa chỉ IP tồn tại thực sự và các gói tin IP hợp lệ → không thể phát hiện dựa trên phân tích bất thường các gói tin IP
- Lưu lượng tấn công thấp
- Truy cập tấn công không khác biệt nhiều với truy cập thông thường.

36

36

# SSL/TLS handshake




Lợi dụng đặc điểm quá trình bắt tay trong SSL/TLS: tốc độ mã hóa nhanh gấp 10 lần tốc độ giải mã

- Liên tục gửi thông điệp yêu cầu nạn nhân thực hiện lại giai đoạn thỏa thuận giao thức(Client key exchange)
- Phòng chống: tắt tính năng hỗ trợ thỏa thuận lại giao thức

37

37

# 3. PHÒNG CHỐNG VÀ GIẢM THIỂU TẤN CÔNG DoS

Nguyễn Đức Toàn,
Viện Công nghệ thông tin và Truyền thông,
Đại học Bách khoa Hà Nội

38

38

## Phòng chống tấn công DoS

- Chống tấn công DoS vào phần cứng
  - Hệ thống cất giữ: Phòng máy, tủ mạng, camera…
- Chống tấn công DoS khai thác lỗ hổng phần mềm:
  - Kiểm thử xâm nhập (Penetration Testing)
  - Cập nhật, vá lỗ hổng phần mềm
- Chống tấn công DoS vào tài nguyên tính toán:
  - Triển khai firewall, IDPS
  - Thiết lập thông số cấu hình hệ thống
  - Sử dụng các kỹ thuật thách đố (Ví dụ: CAPTCHA)

39

39

## Phòng chống tấn công DoS

- Chống tấn công DoS vào băng thông:
  - Mở rộng băng thông
  - Cân bằng tải (Load Balancing)
  - Đối với ISP: Chống tấn công từ nguồn
  - Triển khai firewall, IDPS
- Phát hiện nguồn tấn công:
  - Truy vết nguồn tấn công
  - Phát hiện và ngăn chặn mã độc botnet: triển khai IDPS, firewall

40

40

# Kỹ thuật thách đố client

- Ý tưởng: làm chậm lưu lượng tấn công
- Server yêu cầu client thực hiện giải đố một vấn đề tương đối khó. Ví dụ:
  - Tìm X với C cho trước sao cho:

$$LSB_n ( SHA\text{-}1( C \,||\, X ) ) = 0^n$$

  - Client phải vét cạn $2^n$ giá trị
  - Ví dụ với n = 16, mất 0.3 giây để tìm ra X với CPU có tốc đọo 1GHz
- Khi phát hiện bị tấn công, yêu cầu tất cả client kết nối tới phải giải đố

41

41

# Kỹ thuật thách đố client – Ví dụ

- Chống tấn công TCP connection floods
  - C là giá trị Seq trong gói tin SYN/ACK
  - Gói dữ liệu đầu tiên của client gửi tới phải trả lời giá trị X
- Chống tấn công vào quá trình bắt tay SSL/TLS
  - C là SSL/TLS session ID
  - Server kiểm tra X do client gửi tới trước khi giải mã
- Tương tự,…

42

42

# Kỹ thuật thách đố client – Hạn chế

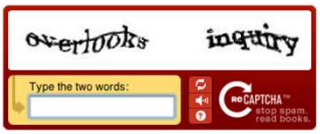
- Khó xác định giá trị n phù hợp do ứng dụng client có thể chạy trên nhiều nền tảng khác nhau:
  - PC
  - Smartphone
  - Thin PC (VD: các thiết bị cảm biến)
  - Giải quyết: sử dụng các bài toán cần tài nguyên bộ nhớ thay vì tài nguyên CPU
- Yêu cầu phải thay đổi ứng dụng ở cả 2 phía(client-server)

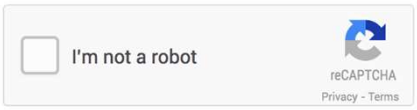
43

43

# Sử dụng CAPTCHAs

- Ý tưởng: xác định yêu cầu kết nối có thực sự do người dùng khởi tạo không




- Ứng dụng: chống tấn công DoS trên tầng ứng dụng (L7 DoS)

44

44

## Kiểm tra địa chỉ nguồn

- Hầu hết các kỹ thuật tấn công sử dụng địa chỉ giả mạo làm địa chỉ nguồn
- Giải pháp tại ISP: chỉ chuyển tiếp các gói tin có địa chỉ nguồn trong mạng do ISP quản lý
- Khó khăn:
  - Tất cả ISP phải cùng triển khai giải pháp
- Thực tế:
  - 2014: 25% AS bị giả mạo, 13% không gian địa chỉ IP bị giả mạo
  - Tấn công vào Spamhaus(03/2013): kẻ tấn công chỉ cần giả mạo 3 dải địa chỉ IP

45

45

## Truy vết tấn công

- Mục tiêu: Dựa trên lưu lượng tấn công để xác định đường đi của lưu lượng
- Ý tưởng: ghi nhớ tuyến đường(các router đã chuyển tiếp) vào gói tin
- Cơ sở:
  - Router khó bị tấn công hơn → có thể tin tưởng hơn
  - Tuyến đường ít thay đổi
- Khó khăn: yêu cầu gói tin sử dụng nhiều byte để lưu trữ thông tin

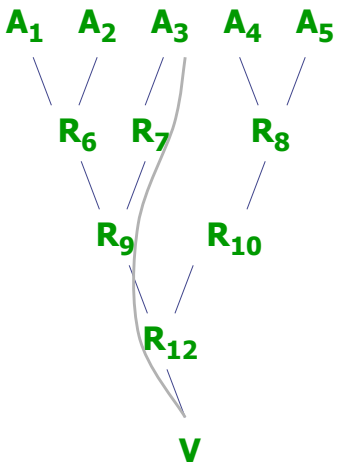
46

46

# Cách thức giải quyết

- Mỗi gói tin tấn công lưu thông tin 1 liên kết
- Thông tin cần lưu:
  - Liên kết: start IP và end IP
  - Khoảng cách: số hop tính từ địa chỉ startIP của liên kết
- Xử lý của router R: tạo sự kiện với xác suất p

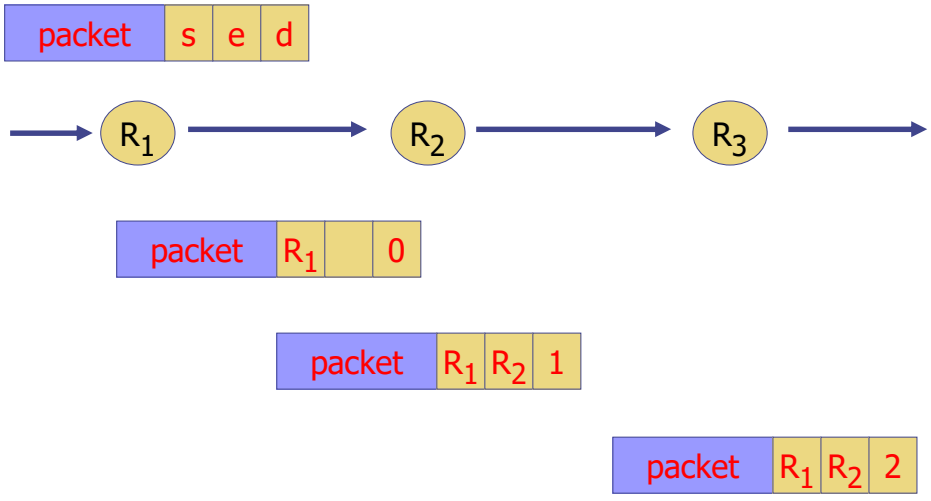
```
if (xảy ra sự kiện)
  startIP ← R
  distance ← 0
else
  if distance = 0
    endIP ← R
  distance ← distance+1
```




47

47

# Truy vết tấn công – Ví dụ




48

48

# Xây dựng đường đi của tấn công

- Truy xuất thông tin từ các gói tin
- Xây dựng đồ thị bắt đầu từ nạn nhân
  - Mỗi bộ giá trị (start, end, distance) cung cấp thông tin 1 chặng
- Vấn đề: không có tác dụng với các dạng tấn công refletion

49

49

Bài giảng sử dụng một số hình vẽ và ví dụ từ các bài giảng:

- Computer and Network Security, Stanford University
- Computer Security, Berkeley University
- Introduction to Computer Security, Carnegie Mellon University

50

50