PHÁT HIỆN LỖI VÀ LỖ HỔNG PHẦN MỀM

Bài 2. Tổng quan về lỗi và lỗ hổng phần mềm

Tài liệu tham khảo

1. Đặng Vũ Sơn, Vũ Đình Thu, "**Phát hiện lỗi và lỗ hổng phần mềm**", Hv KTMM, 2013
2. Freitez, et al. "**Software vulnerabilities, prevention and detection methods: a review**" *SEC-MDA 2009: Security in Model Driven Architecture*. 2009.
3. Michael Howard, David LeBlanc, and John Viega, "**24 Deadly Sins of Software Security**", Mc Graw Hill, 2010

1. Khái niệm
2. Phân loại
3. Một số lỗ hổng phần mềm cụ thể

1. Khái niệm
2. Phân loại
3. Một số lỗ hổng phần mềm cụ thể

Phần mềm

- **Phần mềm** là một tập hợp những câu lệnh và các dữ liệu liên quan nhằm thực hiện một số chức năng hoặc giải quyết một vấn đề cụ thể nào đó.

Hiểm họa

- **Hiểm họa ATTT** của HTTT là những **khả năng tác động** lên TT, HTTT dẫn tới sự thay đổi, hư hại, sao chép, sự ngăn chặn tiếp cận tới TT; tới sự phá huỷ hoặc sự ngừng trệ hoạt động của vật mang TT.
- **Ví dụ**: virus, động đất, tấn công mạng

Lỗ hổng

- **Lỗ hổng** của HTTT là những **khiếm khuyết trong chức năng, thành phần** nào đó của HTTT mà có thể bị lợi dụng để gây hại cho hệ thống.
- **Ví dụ**:
  - Không có cơ chế ngăn chặn duyệt mật khẩu
  - Luật lọc của tường lửa không được cập nhật
  - Không có UPS

Lỗ hổng phần mềm

- **Lỗ hổng phần mềm** là một điểm yếu trong cách thức cài đặt hoặc là một lỗi trong phần mềm có thể bị khai thác bởi một kẻ tấn công để làm thay đổi hoạt động bình thường của phần mềm [1]
- **A software vulnerability** is a flaw or defect in the software construction that can be exploited by an attacker in order to obtain some privileges in the system.

Điểm yếu ≠ Lỗ hổng

❑Lỗ hổng (Vulnerability)

- thực tế đã bị khai thác
- https://cve.mitre.org/
- CVE = Common Vulnerabilities and Exposures

❑Điểm yếu (Weakness)

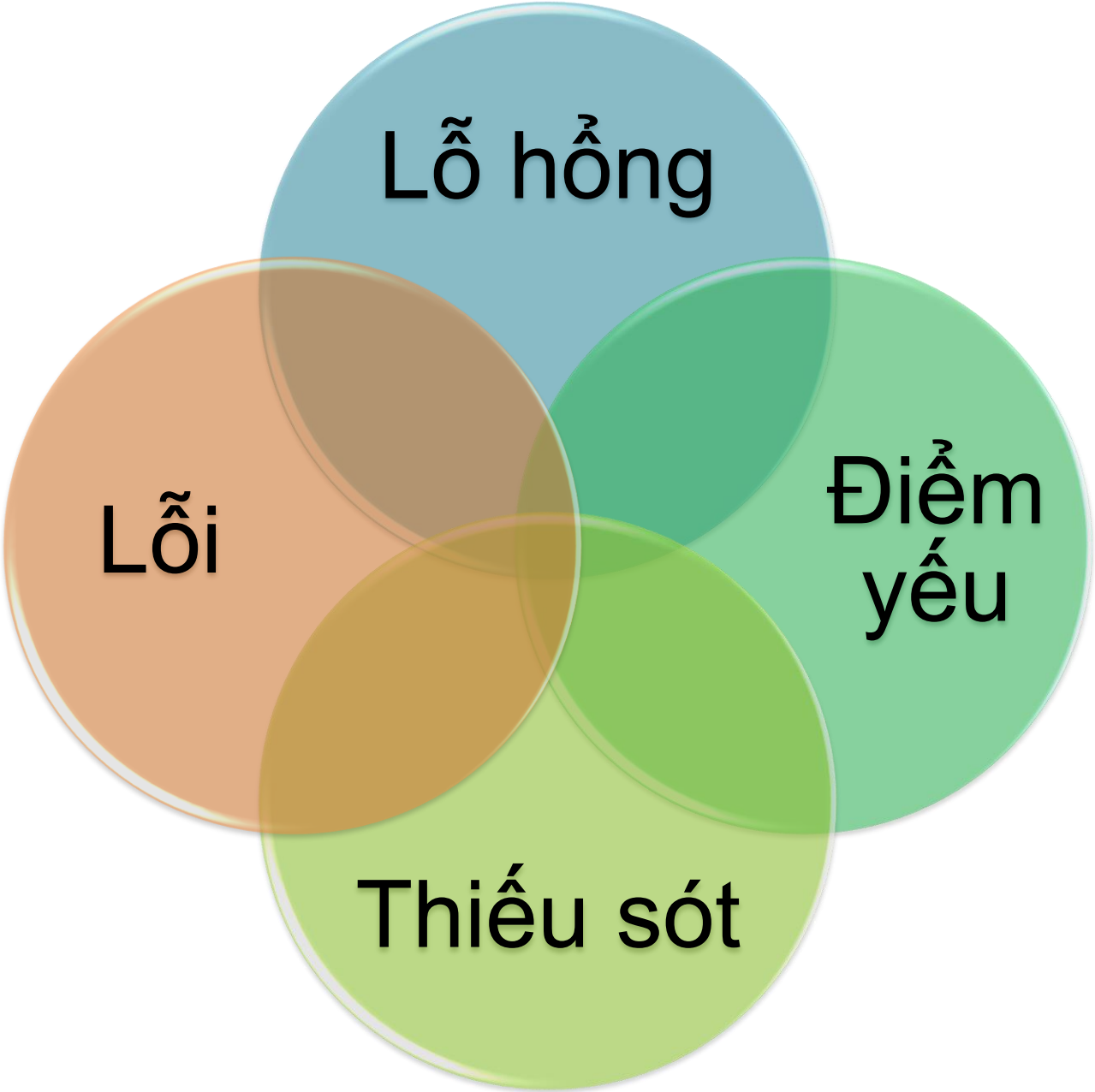
- Có thể bị khai thác
- https://cwe.mitre.org/
- CWE = Common Weakness Enumeration

Zero-day vulnerability?

Lỗi và lỗ hổng phần mềm

- Nguyên nhân của lỗ hổng là lỗi ở một công đoạn nào đó của quy trình phát triển phần mềm
- Có lỗi chưa chắc đã có lỗ hổng
- Có lỗ hổng thì nhất định là vì có lỗi

Thuật ngữ "lỗ hổng"



1 Khái niệm

2 Phân loại

3 Một số lỗ hổng phần mềm cụ thể

Phân loại

- **Phân loại**: là việc phân chia một tập hợp thành các tập hợp con theo một tiêu chí phân loại nhất định
- **Tiêu chí phân loại**: là một đặc điểm của các phần tử được chọn để phân biệt các phần tử với nhau
- **Ví dụ tiêu chí phân loại**: Giới tính, Điểm trung bình, Độ tuổi, Cân nặng,...

Phân loại lỗ hổng phần mềm

❑Tiêu chí phân loại

- Theo nguyên nhân xuất hiện
- Theo thời điểm xuất hiện (trong quy trình phát triển phần mềm
- Theo mức độ nguy hiểm
  - Định tính
  - Định lượng

Phân loại theo nguyên nhân xuất hiện

- In most of the cases vulnerabilities are caused by improper validation of the data supplied by the user [2]
- Lỗ hổng do kiểm tra dữ liệu: Tràn bộ đệm, Chuỗi định dạng, XSS, SQL Injection...
- Lỗ hổng khác: Race condition, Sử dụng các thành tố mật mã không an toàn...

Các giai đoạn phát triển phần mềm

Nghiên cứu sơ bộ (Preliminary Investigation)

Phân tích yêu cầu (Analysis)

Thiết kế hệ thống (Design of the System)

Xây dựng phần mềm (Software Construction)

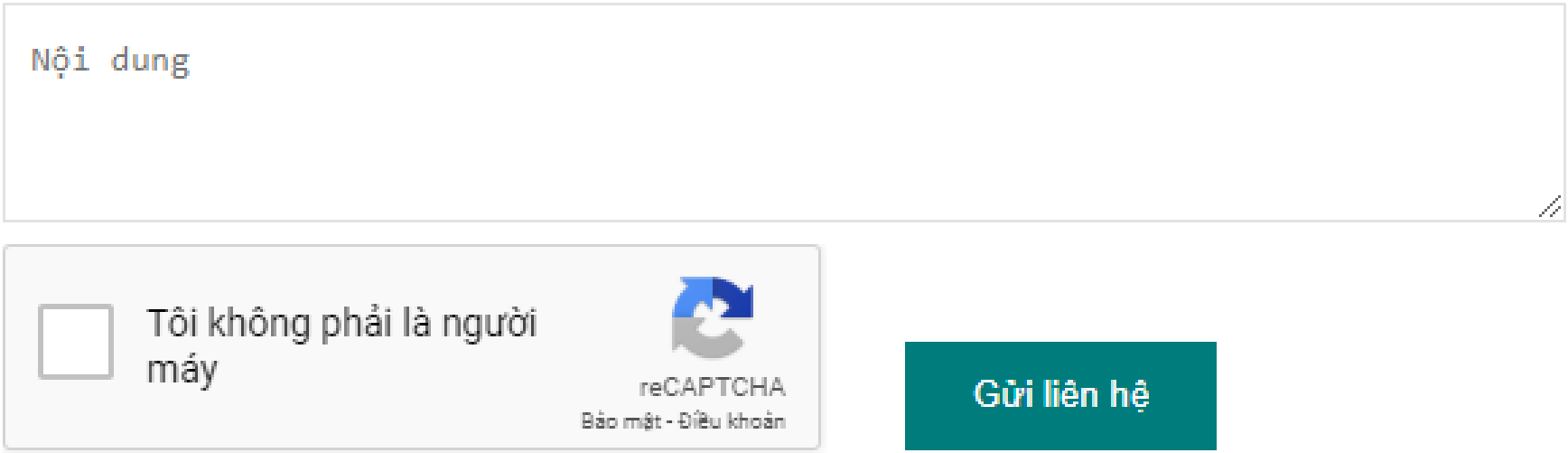
Thử nghiệm hệ thống (System Testing)

Thực hiện, triển khai (System Implementation)

Bảo trì, nâng cấp (System Maintenance)

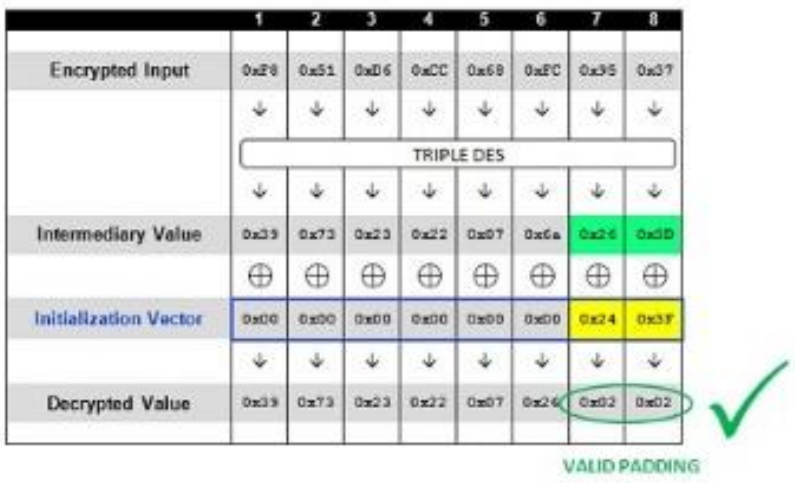
Giai đoạn Phân tích (đặc tả) yêu cầu

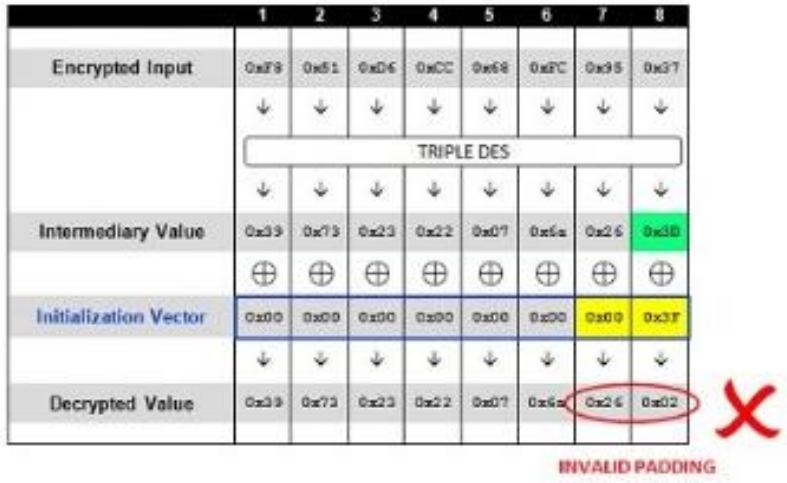
- Lỗ hổng xuất hiện do không có yêu cầu về tính năng an toàn
- Ví dụ: không yêu cầu cơ chế chống spam ở trang "Liên hệ"



Giai đoạn Thiết kế

- Lỗ hổng xuất hiện do:
  - Thiết kế luồng thực thi không an toàn
  - Lựa chọn hoặc cho phép lựa chọn sử dụng các thành tố không an toàn
- Ví dụ: tấn công Padding Oracle lên chế độ CBC của mã khối





Giai đoạn Xây dựng (lập trình)

- Lỗ hổng xuất hiện do sử dụng các hàm, các cấu trúc không an toàn, do không kiểm tra thỏa đáng dữ liệu đầu vào
- Ví dụ:
  - buffer overflow
  - format string
  - race condition
  - integer overflow...

Định tính mức độ nguy hiểm

- **Lỗ hổng loại C** (Mức thấp): cho phép tấn công từ chối dịch vụ (DoS)
- **Lỗ hổng loại B** (Mức trung bình): cho phép người dùng cục bộ leo thang đặc quyền hoặc truy cập trái phép.
- **Lỗ hổng loại A** (Mức cao): cho phép người dùng từ xa có thể truy nhập trái phép vào hệ thống

Định lượng mức độ nguy hiểm

- Common Vulnerability Scoring System, https://www.first.org/cvss/

- Có 3 nhóm đại lượng đặc trưng cho mỗi lỗ hổng

- Base Metric Group
- Temporal Metric Group
- Environmental Metric Group

Base Metric Group

Exploitability Metrics

- Attack Vector
- Attack Complexity
- Privileges Required
- User Interaction

Impact Metrics

- Confidentiality Impact
- Integrity Impact
- Availability Impact

- Scope

Temporal Metric Group

Exploit Code Maturity

Remediation Level

Report Confidence

Environmental Metric Group

Modified Base Metrics

Confidentiality Requirement

Integrity Requirement

Availability Requirement

Định lượng mức độ nguy hiểm

- Mỗi đại lượng đều có thể đo được và nhận một giá trị nhất định
- Có công thức để tính điểm chung cho lỗ hổng từ giá trị của các đại lượng, https://www.first.org/cvss/calculator/3.0
- Thang điểm: 0.0 đến 10.0; điểm càng cao càng nguy hiểm

Định lượng mức độ nguy hiểm

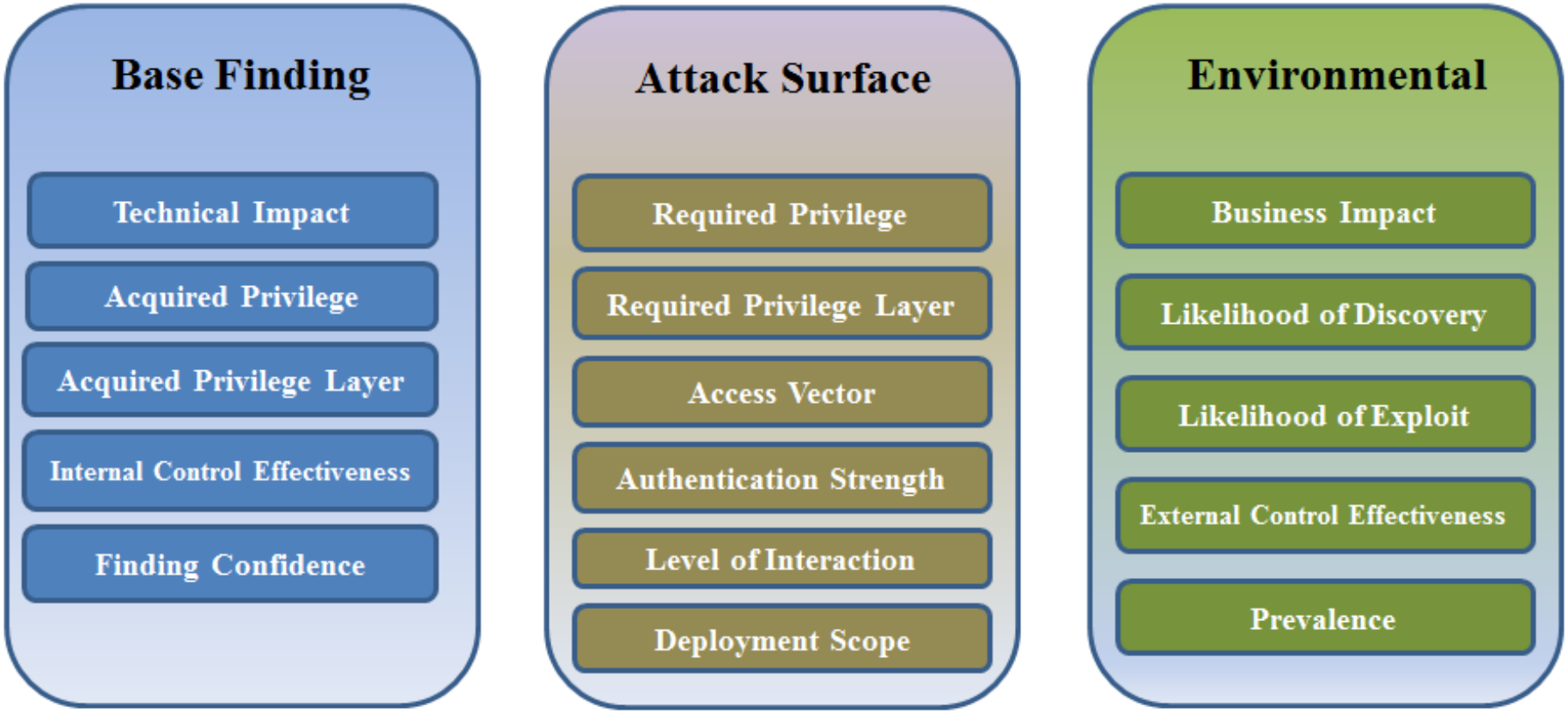
Rating	CVSS Score
None	0.0
Low	0.1 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0

Demo

Common Weakness Scoring System

- CWSS Metric Groups

  https://cwe.mitre.org/cwss/cwss_v1.0.1.html



1 Khái niệm

2 Phân loại

3 Một số lỗ hổng phần mềm cụ thể

Lỗ hổng do lập trình

Buffer Overflow	Format String	Integer Overflow
Off-by-One	Race Condition	●●●

Lỗ hổng do lập trình

Buffer Overflow	Format String	Integer Overflow
Off-by-One	Race Condition	●●●

Lỗ hổng khiến dữ liệu có kích thước lớn có thể tràn ra khỏi vùng đệm để chứa nó

```
char st[10];
gets(st);
```

Lỗ hổng do lập trình

Buffer Overflow	Format String	Integer Overflow
Off-by-One	Race Condition	●●●

Lỗ hổng khiến dữ liệu chuỗi bị diễn giải như một chuỗi định dạng

```
char st[10];
gets(st);
printf(st);
```

Lỗ hổng do lập trình

Buffer Overflow	Format String	Integer Overflow
Off-by-One	Race Condition	●●●

Lỗ hổng khiến kết quả phép toán trên số nguyên bị diễn giải sai khi vượt quá phạm vi giá trị

```
int a, b;
scanf("%d", &b);
if(a+b<a)
        printf("b<0");
```

Lỗ hổng do lập trình

Buffer Overflow	Format String	Integer Overflow
Off-by-One	Race Condition	●●●

Lỗ hổng khiến ký tự kết thúc chuỗi bị ghi đè

```
char *st1, *st2;
...
for(int i=0; i<=strlen(st1); i++)
        st1[i] =st2[i];
```

Lỗ hổng do lập trình

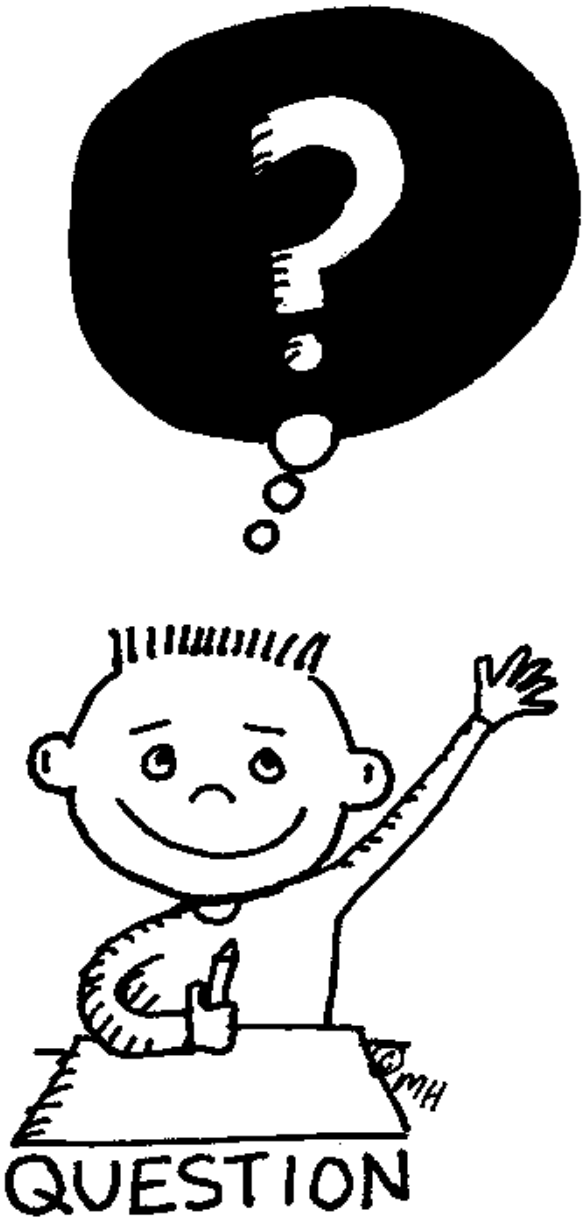
Buffer Overflow	Format String	Integer Overflow
Off-by-One	Race Condition	●●●

Lỗ hổng trong vấn đề đồng bộ dữ liệu khiến một tiến trình vẫn xử lý dữ liệu cũ, trong khi dữ liệu đã được cập nhật bởi một tiến trình khác

Lỗ hổng do lập trình

Buffer Overflow	Format String	Integer Overflow
Off-by-One	Race Condition	●●●

- Lỗ hổng web: SQL Injection, XSS, CSRF...
- Sử dụng các thành tố mật mã không tốt
- Giải phóng bộ nhớ 2 lần
- Sử dụng bộ nhớ sau khi đã giải phóng
- ...


QUESTION